A Methodology for Data-Informed Process Control in Progressive Die Sheet Metal Forming

by

Dylan Budnick

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2022

Author’s Declaration

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

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

Abstract

This thesis investigates the coupled relationship between the strip transfer and forming operations in progressive die sheet metal forming, including the effects of the strip layout geometry, and its effect on the process speed and accuracy. Servo-actuated strip lifters and feeder are considered to assist in minimizing the dynamic response of the strip during the transfer process. A methodology is proposed for identifying suitable trajectories to prescribe the motion of active strip lifters and feeder to obtain consistent part quality without risk of process failures for a progressive die operation.

Multiple iterations of a finite element (FE) model were constructed in LS-DYNA to simulate a progressive die operation. Various FE analysis techniques were used to reduce the computational cost of the simulations to allow for enough data to be generated for machine learning applications. Both explicit and implicit time-integration schemes were considered in iterations of the FE model.

Both single and dual carrier strip layouts were considered. The results of the FE simulations suggest that the single carrier strip layouts produce larger predicted dynamic displacements and rotations of the work-piece as compared to the dual carrier strip layouts during strip transfer. Furthermore, the single carrier strip layout is shown to be susceptible to strip misalignment.

The final version of the FE model utilized geometry based on a demonstrator tool being deployed at the Technische Universität München. A total of 1000 simulations were generated, 500 each for the 'I' and 'O' stretch-web types using a single carrier strip layout. Each simulation considered a unique permutation of control inputs sampled from the set of possible strokes rates and trajectories for the lifters and feeder. Cubic splines were used to generate the trajectories for the strip lifter and feeder by varying the position of two knots used to define the shape of the spline.

The results from the 1000 simulations indicate that in general the 'S' stretch-web produces a larger variance in the predicted dynamic response and 'work-piece placement as compared to the 'I' stretch-web. Furthermore, the stroke rate and lifter trajectory were shown to have a large influence on the overshooting of the work-pieces during strip transfer and the probability of whether tooling collisions occurred.

Multiple machine learning models were trained on the data generated by the final FE model. Two types of classifiers were constructed using neural network and XGBoost architectures. The first classifier predicts whether the clearance between the strip and binder are within a specified tolerance

(to prevent collision with the tooling) during strip transfer. The second classifier predicts whether the placement accuracy of the work-piece on the forming die after strip transfer is within a specified tolerance. A range of tolerances were considered when labeling the data for both classifiers. Nested-cross fold validation was used to select the hyperparameter tuning and model selection.

The machine learning classifiers were used to test all possible control inputs using a ‘minimum feed clearance’ of 10 mm and a maximum ‘work-piece placement error of 0.11 mm. The maximum stroke rate at which a given pair of lifter and feeder trajectories can operate was identified for all permutations. Five permutations that achieved the highest predicted stroke rate were simulated for an additional five strokes. The classifiers showed a reasonable ability to predict the ‘minimum feed clearance’ and ‘work-piece placement in the extended FE simulations for the selected trajectories, but, was unable to account for the strip misalignment which occurred after several strokes in all simulations.

This research successfully demonstrates a methodology for using machine learning models trained on FE simulations to predict process outcomes of a progressive die operation with variable feeder and lifter trajectories. The FE simulations used to train the machine learning models were generated by adopting computationally-effective FE modelling techniques in a single press stroke model. The machine learning models were shown to reasonably predict the process outcomes of novel input permutations in a multi-stroke FE simulation. One of the largest constraints in this research is the FE simulation time which limited the model complexity that could be considered in the training set generation. Furthermore, the demonstration of the machine learning predictions for a multi-stroke process was limited due to the susceptibility of the single carrier strip layout to misalign after strip progression. Future work should consider the use of dual carrier strip layouts for the generation of the training data. Alternative approaches may also be considered, such as a machine learning framework for directly predicting the forward dynamics of the progressive die operation or a co-simulation approach in which a robust controller interacts directly with the FE simulation.

Acknowledgements

I would like to thank my supervisors, Professor Michael Worswick and Professor William Melek for taking a chance on me with this ambitious project and the invaluable insights and guidance they have provided me with throughout my research. This project has been an amazing learning experience and I owe that to them.

Support and sponsorship from the Technische Universität München, Forming Technologies (part of Hexagon Manufacturing Intelligence), and Men at Work GmbH is gratefully acknowledged. The financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), Canadian National Research Council (NRC), Industrial Research Assistance Program (IRAP), Ontario Advanced Manufacturing Consortium (AMC), German Aerospace Center (DLR), and German Federal Ministry of Education and Research (BMBF) is highly appreciated and recognized.

I would like to thank Abdulrahman Ghannoum for his great feedback and the discussions we had both related and unrelated to my research. Thanks to Florian Steinlehner for his excellent teamwork and being a great person to have a drink with. I would to give a shout-out to Cameron Tolton for his friendship and the occasional chat about LS-DYNA.

Finally, I would like to thank my girlfriend, Shirley Wang, my parents, Ute Linder and Allan Budnick, and my uncle, Stephen Lindner, for their unconditional support and encouragement.

Dedication

To all my ancestors, whose small and large choices precipitated the life I am able to live

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Research in sheet metal forming has seen many advances over the past several decades which is embodied in the ability of industry to manufacture ever increasingly complex components with materials tailored to the specific needs of an end-product. The majority of this research has focused on the design of the sheet metal forming process, addressing the characterization of material formability or applying advanced material models in finite element analysis (FEA) to simulate forming processes, for example [1]. While the forming process design phase has seen an abundance of attention from the research community, production efficiencies in the manufacturing stage of sheet metal components have received comparatively little attention and largely are left in the hands of tool makers to realize.

The largest source of cost in sheet metal forming is often the cost of the sheet metal itself [2], which is normally delivered in a coil form and often referred to as the "strip" which is fed through a forming press in a progressive stamping operation. Therefore, the largest source of production efficiency can be found by increasing the material utilization of the strip. The most important factor in determining the material utilization is the strip layout (Figure 1) where small inefficiencies can accumulate to large sources of waste when components are produced in large quantities [3], [4]. However, a strip layout cannot be designed solely on the basis of material utilization since the geometry of the strip layout has a large influence on the dynamic response of the strip which can negatively impact the product quality and process outcome. In particular, the placement and selection of the stretch web connector geometry, which connects parts being formed to the sheet strip being fed through the forming stations (see Figure 1), has a significant impact on the dynamic response of strip during the rapid feeding and lifting of the strip in progressive die operations. Additionally, the selection of stroke rate is integral to the productivity of a process with faster stroke rates increasing the product through-put, but at the risk of increased dynamic excitation of the sheet metal strip which can result in reductions in part quality and even collisions between the strip and tooling [5], [6]. There has been limited research addressing this issue and therefore tool designers have relied on prior experience to deal with these trade-offs. The challenge of finding an optimal balance between production stability which minimizes risk of tooling collisions and maximizing material utilization is exacerbated by the lack of control inputs during progressive die operation. With the exception of stroke-rate, tool makers have limited flexibility in fine-tuning the actuation of the tooling components and material feeding to accommodate the unique

characteristics of a given strip layout. By introducing novel forms of actuation that can be controlled by an operator, a more optimal trade-off between production stability and cost becomes tractable for tool makers when designing strip-layouts, since the strip progression could be tailored to the dynamic response of strip. More specifically, active actuation could be introduced to the strip lifters which are traditionally actuated by passive springs. Additionally, a strip feeder could be introduced that allows for adjustable inputs by the tool operator instead of a fixed input based on the stroke rate. The introduction of these forms of actuation would be primarily limited by the implementation of their respective control strategies and the identification of suitable reference input trajectories for the controller.

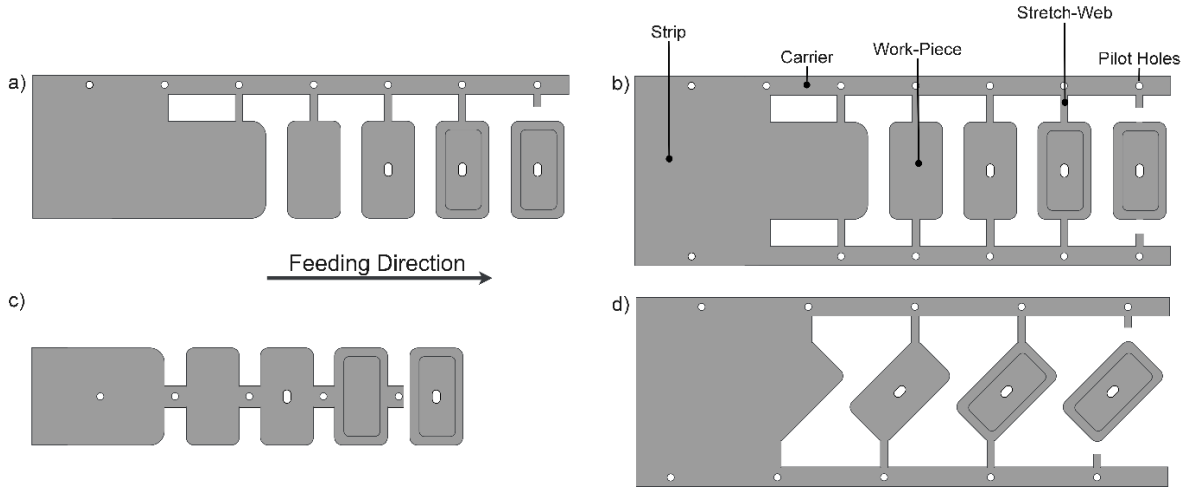

Figure 1: Examples of strip layouts used to produce the identical parts with (a) 1 row, 1 side carrier, (b) 1 row, 2 side carrier, (c) 1 row, 1 center carrier, (d) 1 row, 2 carrier, 45° nesting.

1.2 Problem Statement

Identifying suitable input trajectories for the control system requires modeling of the system to understand the cause and effect principles (Figure 2). However, a progressive die process is a highly non-linear system which has coupled elements of dynamics and forming. Formulating a general analytical expression to describe a progressive die operation is exceedingly difficult and therefore modelling such a system is best approached using numerical techniques, such as FEA or machine learning algorithms. Both of these techniques offer their own relative merits. FEA models have been shown to accurately predict dynamical systems and material deformation, for example, when compared to experimental results, but, are limited by their substantial computational cost and processing times

[7]. On the other hand, machine learning techniques can rapidly make predictions due to their low computational cost to execute but require vast and rich data sets to be trained on to properly learn the response of a non-linear system [8]. Such data sets do not exist currently for progressive die operations. One approach, and the focus of the current research, is to combine these two modeling techniques. This strategy has opened up an emerging area of research in which FEA can be used to construct datasets that would be difficult to gather experimentally to train a machine learning model to explore the system with the goal of optimizing a parameter or parameters [9].

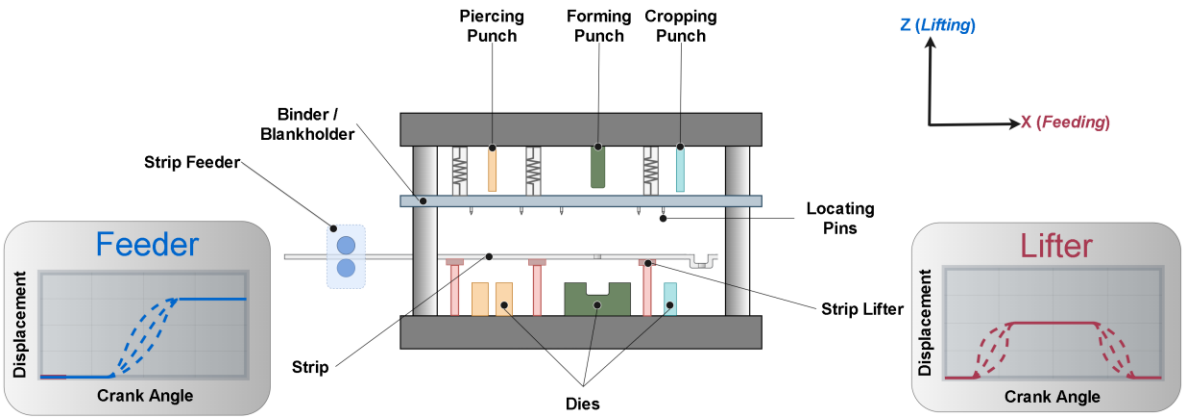

Figure 2: Labeled diagram of a progressive die operation used to form a basic cup with arrows indicating the feeding and lifting directions and multiple feeder and lifter input trajectory permutations.

1.3 Thesis Contributions

The objective of this research is to construct a methodology for identifying suitable input trajectories to actuate the strip lifter and feeders for a given progressive die operation. A hybrid FE-machine learning strategy was pursued which leverages the modeling fidelity of FEA and computational speed of machine learning. To overcome the absence of experimental data, models were constructed in the commercial FE software LS-Dyna to simulate a progressive die operation with varying inputs [10]. The FE predictions were used as a training set for a machine learning model to learn the behaviour of the progressive die operation. The machine learning model can then be utilized to explore the input permutation space more readily with the goal of finding process inputs for the feeder, lifter and stroke rate that optimize the performance of the progressive die operation. Another objective of this work was to identify appropriate methods for measuring the performance of progressive die operations. Furthermore, datasets for multiple strip layouts were constructed with the goal of understanding the

influence of the stretch-web on the dynamic response of the strip and demonstrating how the proposed methodology generalizes.

This research was undertaken as part of a larger collaboration between the University of Waterloo, the Technische Universität München (TUM), Hexagon Manufacturing Intelligence and Men At Work GmbH (MaW), under the 2+2 Canada-Germany International Collaboration Framework. As part of the collaboration, a progressive die tool was constructed by MaW and implemented at TUM. The goal for this tooling is to gather *in-situ* experimental data from a progressive die operation. The tooling includes a high-speed camera and proximity sensors to monitor the position and dynamic motion of the sheet metal strip. Actively controllable lifters and a strip feeder were designed and implemented by the TUM to test various inputs to the progressive die operation. The component of the research undertaken at the University of Waterloo, presented in this thesis, was to construct a cyclical data loop in which a tool designer can learn how to operate their tooling in the virtual world and apply this knowledge when operating their experimental tooling. Sensor data from the experimental tooling can then be used to inform the virtual model to be more representative of the experimental results to enhance its predictive capability and provide higher resolution insights to the tool designer.

The research goals addressed in this thesis are as follows:

- Identify metrics for evaluating the performance of a progressive die operation.
- Construct a FE model to simulate a progressive die operation with varying inputs
- Construct a Machine Learning model trained on the FE data to predict the performance of a given set of inputs to a progressive die operation
- Assess the predictive ability of the Machine Learning model

1.4 Thesis Organization

The remainder of the thesis is structured as follows. Chapter 2 provides a literature review of related work and the state-of-the-art. Chapter 3 describes the methodology implemented in this research. Chapter 4 details the iterations of the FE model developed to simulate the progressive die operation. It describes the variables that influence progressive die operations used in industry and suggests means for measuring the outcome of the process. Chapter 5 describes the machine learning component of this research with the so-called ‘Smart Trajectory Generator’. This chapter describes how the FE model was used to generate data and what insights can be gain from this data. Ultimately, Chapter 5 shows the

work-flow for identifying suitable process inputs for operating a progressive die based on virtual modeling. Chapter 6 presents conclusions from this research and recommendations for future work.

Chapter 2
Background

This chapter presents a review of published literature relevant to this research. This includes a review of sheet metal forming research, focused on simulation of formability and progressive die operations, as well as relevant research on machine learning. Previous studies in which machine learning is applied to sheet forming, though limited in number, are also discussed.

2.1 Progressive Sheet Metal Forming

2.1.1 Introduction

Sheet metal forming has had a very long history, dating back thousands of years. Gold, silver, and meteoric iron were used in a decorative capacity in the Neolithic era (>3500 BC). Two thousand years later in the Iron Age, sheet metal forming began to emerge as a common practice where craftsmen cold worked copper and gold using stone hammers and anvils. Some of the first reported mass production of sheet metal is from the Roman army where each soldier wore a belt decorated with formed brass plates. It was not until the 19th century that high-speed mass production became available through the invention of the first crank press by Root, followed shortly after other designs [11]. In the early days of high-speed production, sheet metal products were primarily manufactured for military purposes, similar to the Romans. By World War 2, sheet metal manufacturing was common practice, but, still limited to single operation presses where a pre-cut section of sheet metal was hand-fed by an operator into a press. To improve operator safety and production efficiencies, Stouten founded Capitol Engineering Company in 1953 with the goal of designing and selling single multi-station dies [12]. The novel concept was first received with skepticism but once demonstrated successfully, was soon sought after by many die designers. Over the next 30 years, progressive sheet metal forming would become spread around the world and start to become the staple in the electronics and automotive industry that it is today.

Progressive sheet metal forming has seen ubiquitous use because of its high-production capacity and relatively low cost while still producing complex sheet metal components. What distinguishes progressive die operations from other forms of sheet metal forming is that a press will perform sequential operations on a continuous strip of sheet metal that is moved through the tooling. Other

forms of sheet metal forming may still utilize multiple operations within a press, such as transfer stamping. However, in transfer stamping the strip of sheet metal will be sheared into discrete sections prior to entry or inside the tooling itself which are then individually transferred between die operations. An example of a sheet metal strip undergoing multiple operations within the same press can be seen in Figure 3. The operations performed in progressive dies can be generalized to shearing operations or bending and forming operations which plastically deform the material. Shearing operations include; *Shearing*, *Blanking*, *Punching/Coining*, *Parting*, *Lancing*, and, *Shaving*. Plastic deformation operations include *Bending*, *Twisting*, *Curling*, *Deep Drawing*, *Spinning*, *Stretch Forming*, *Necking*, *Bulging*, and, *Flanging* [2], [14].

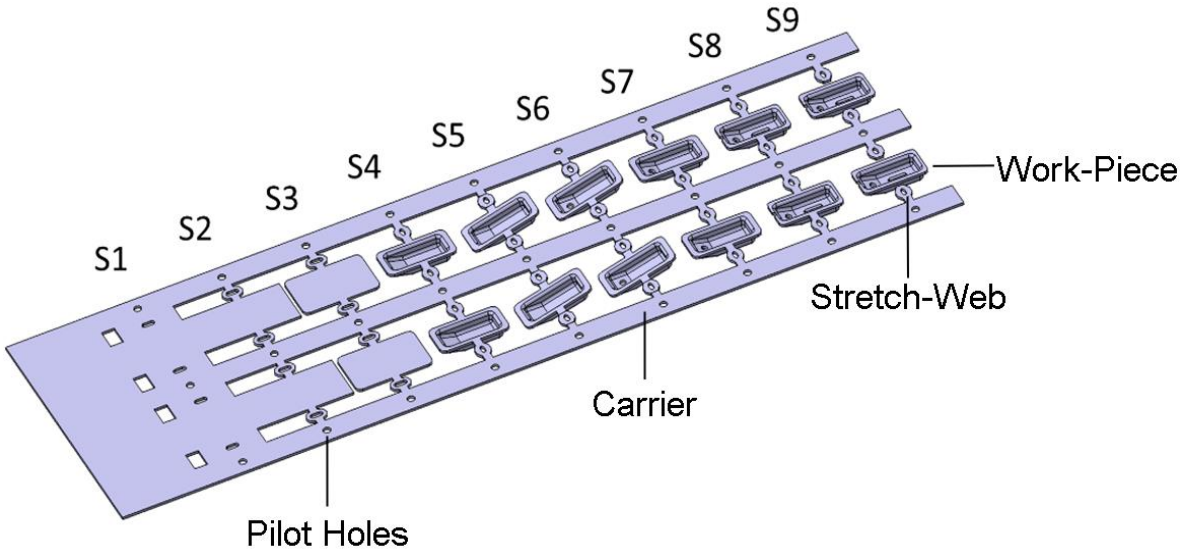

Figure 3: Example of sheet metal strip in a progressive die operation with 9 stages. Stage 1 through 3 are blanking operations that remove unnecessary sections of the sheet metal. Stage 4 is a cup forming operation. Stage 5 through 7 rotate the part to coin a hole in the wall of the cup and subsequently rotate the cup back to its original configuration. Stage 8 embosses a bead in the bottom of the cup. Finally, stage 9 crops/shears the cup from the rest of the strip [13].

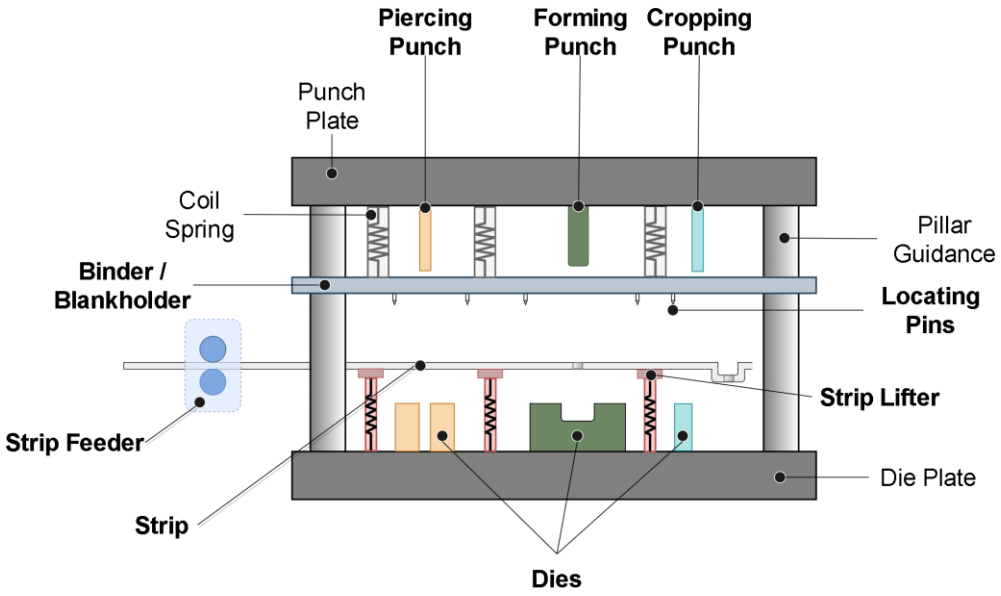

Figure 4: Progressive die tooling with primary components indicated with bold text; Binder, Strip, Die, Punch, Lifters, Locating Pins and Strip Feeder.

Referring to Figure 4, there are five key components within progressive dies used for sheet forming: locating pins, punch, binder/blank holder, die block, and, sheet metal strip [15]. Additionally, a strip feeder is required to transfer the strip through the tooling and strip lifters are required to lift the strip prior to the strip transfer to provide clearance between the formed work-piece and the die block. The tooling punches are responsible for shearing, forming, bending, and, any other operation(s) that plastically deforms or shears the sheet metal. The binder is used to prevent wrinkling of the sheet metal during drawing operations and used to release the sheet metal from the punch after forming or shearing operations. The locating pins are used to engage the pilot holes (Figure 5) to position the work piece prior to each tooling operation. The punch and die are the components that directly contact and press against the work-piece to shear or plastically deform it to the desired shape [2].

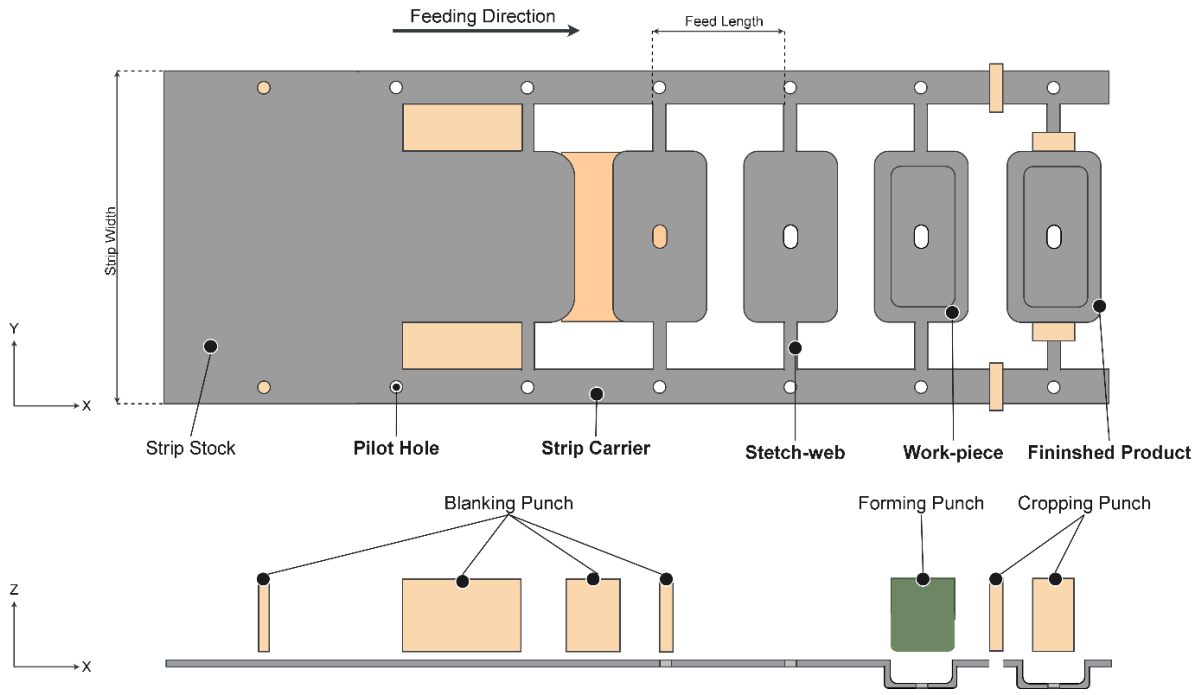

Figure 5: Top and side view of a sheet metal strip undergoing multiple operations in a progressive die operation. First, the blanking punches removes scrap material from the strip and create pilot holes. The remaining sheet metal 'blanks' are formed into cups with the forming punch. Finally, the finished product is removed from the strip by the cropping punch.

Currently, the design of strip layouts is largely an experience-based effort, resembling more of an art than a science. As a result, the terminology and design of the strip-layout lacks industry standardization [6]. Figure 5, shows an example of a strip layout undergoing a basic deep-drawing operation with the general sections of the strip identified. Throughout this thesis, the basic components of the strip will be identified using the terminology shown in Figure 5. The ‘strip stock’ is the unaltered sheet metal being fed from the coil. The work-piece is the portion of the strip being operated by the press and will leave the tooling as the finished product. The carrier strip is the section which runs the length of the press and connects all the individual work-pieces together. The ‘pilot holes’ or locating holes are used to locate the strip via locating pins during each stage of the tooling. Finally, the web or ‘stretch-web’ connects the work-piece to the carrier strip and supports it during the strip progression. The stretch web elongates as the blank is pulled into the die during the forming operation.

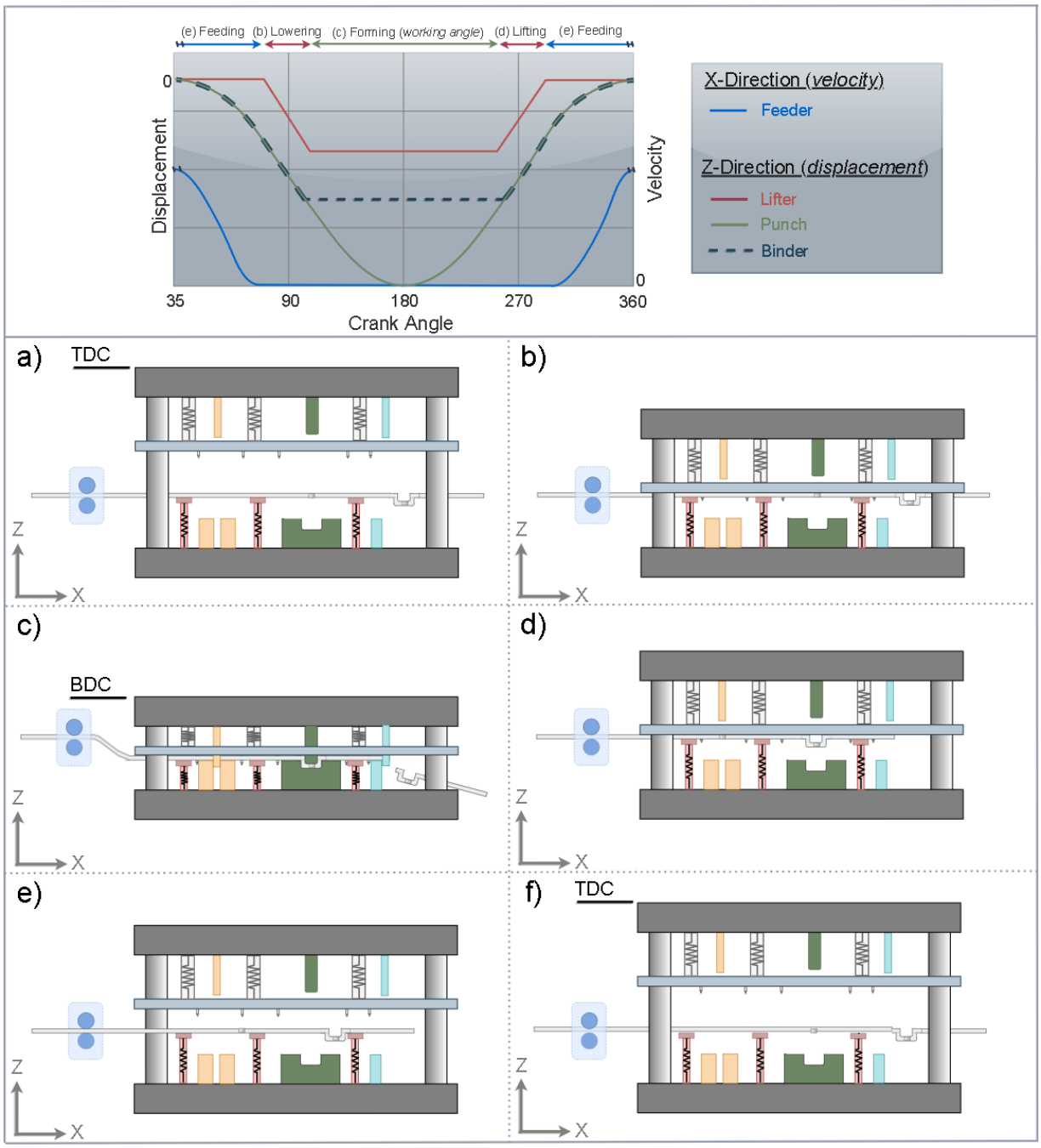

Figure 6: Standard motion of a single press stroke for a progressive die operation (top) and visualization of press stroke (bottom a-f). The vertical displacement of the tooling is measured in terms of displacement whiles the strip feeder is measured in terms of velocity.

Figure 6 shows the typical motion seen in a progressive die operation. The cycle starts with the tooling fully open and the upper portion of the tooling in the top-dead-centre position when the crank angle is at 0°. The lower tooling is fixed while the upper tooling is lowered by the press and moves the binder, locating pins, and punch with the same trajectory. As the binder approaches the strip, the locating pins thread the pilot holes and locate the strip position. Once the binder contacts the strip which is resting on the lifter in their upper position, the passive springs supporting the lifters being to compress and the lifter and strip begin to lower with the upper tooling. As the upper tooling continues to lower, the binder forces the strip and lifters to their lower position. Once the lifters are in their lower position, the strip will be flush with the die face and compressed by the binder. The binder spring will now begin to compress as the binder applies pressure to the strip against the die face and the upper tooling continues to lower. First the shearing punches engage with the strip, followed by the forming punches which begin to plastically deform the clamped work-piece. The punches continue to form the work-piece, which pulls material into the die cavity, taking the shape of the partially or fully formed component. At its lowest position (bottom dead-center), the press begins to open. The punches leave the strip and the binder spring expands as the upper tooling travels away from the lower die. The compressed lifter springs force the lifters and strip to follow the binder and upper tooling as they continue to rise until the lifters reach their upper position and the binder and locating pins disengage with the strip. The upper tooling finishes the stroke once it reaches the top dead-center position which signals to the strip feeder to progress the strip to the next stage of the tooling.

2.1.2 Stretch-Web Connectors

Stüber and Böttcher [6] wrote one of the very few pieces of literature providing a review of stretch-web connectors and their relevance in industry. The goal of their work was to categorize and define the selection of stretch webs for progressive sheet metal forming. Figure 7 shows the most commonly used stretch-web connectors used in progressive die operations. The selection of a stretch-web has a large influence on the retention force which restricts the flow of material during deep drawing operations. The ‘O’ stretch-web imparts higher rigidity and retention on the work-piece during forming operations whereas the ‘S’ stretch-web is more compliant and will induce less retention on the work-piece. Selecting a stretch-web with a suitable compliance for a specific forming operation is necessary since an excessive retention force can inhibit the work-piece from forming or result in the fracture of the stretch-web. On the other hand, the stretch-web influences the dynamic response of the sheet metal strip during progression between tooling stations since the stretch-web and work-piece act as spring-

mass system. Excessive compliance (low stiffness) of the stretch-web will make the work-piece more susceptible to oscillations during strip progression. Furthermore, the limiting factor of the stroke-rate and therefore the production rate is the oscillation of the strip during strip progression since increased strip vibrations can reduce component quality from sub-optimal strip transfers and potentially lead to collisions between the strip and tooling

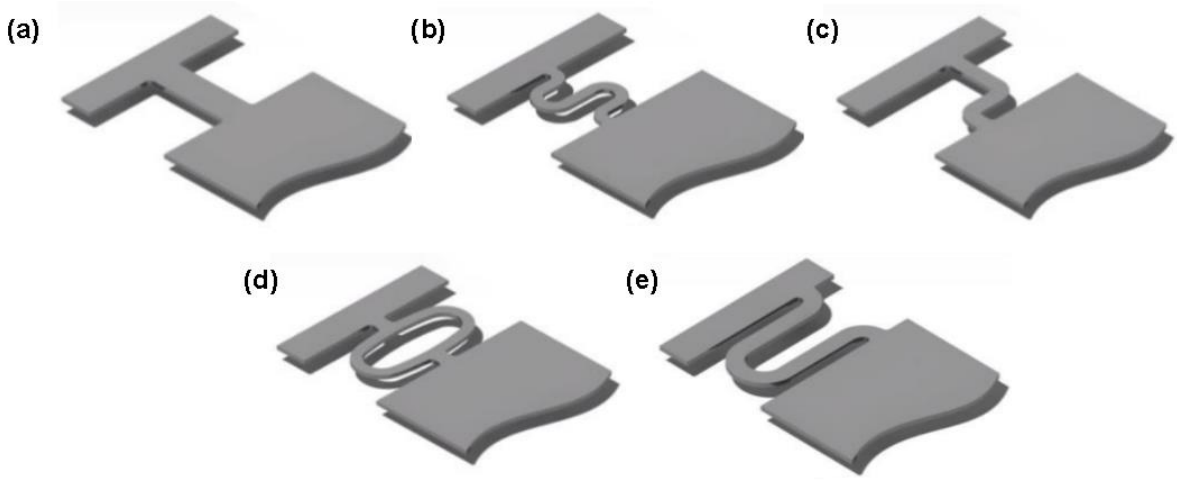

Figure 7: Example of five most common stretch-web connectors used in industry. (a) I-web (b) S-web (c) L-web, (d) O-web, (e) U-web [6].

2.1.3 Design

Production rate, or in other words, the number of parts produced per unit time is generally one of the most important metrics of any manufacturing process. Particularly in industrialized nations where manufacturing constitutes 15 to 25% of gross national product, manufacturing productivity has an integral role on the standard of living [14]. In progressive sheet metal forming, one of the main determinants of production rate, cost and part quality is the strip layout [16]. A strip layout that produces two parts simultaneously will double the production rate as compared to a layout which produces one part per stroke (Figure 8). However, producing more than one component per stroke will require a wider sheet metal strip and a larger press with increased tonnage. Furthermore, similar to the selection of stretch webs, the geometry of the strip layout has an intimate effect on the dynamic response of the strip during strip lifting and progression. A strip layout with a double-row of parts and a single carrier, as seen in Figure 8, will be more susceptible to dynamic oscillations than using two carriers since each

work piece will have an unsupported side, creating a cantilevering structure which is inherently less stiff than a structure supported on both sides.

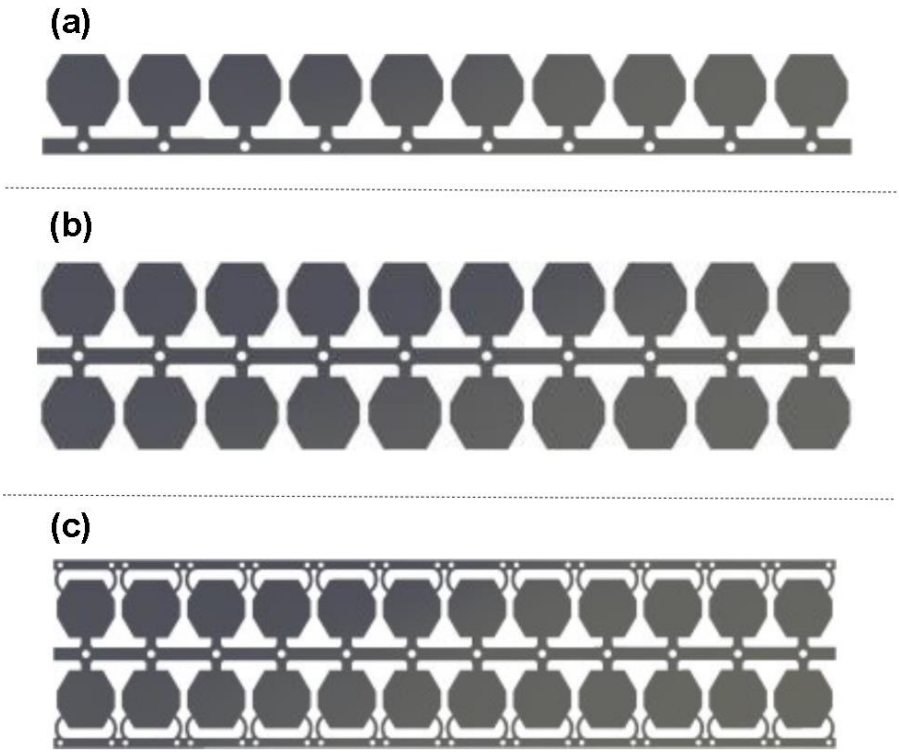

Figure 8: Examples of strip layouts with varying material utilization. (a) Single-row layout with one carrier, (b) double-row layout with single carrier, (c) double-row layout with two carriers [6].

Almost every design decision, including the press selection is downstream of the strip layout and therefore a tool designer always starts with unfolding the final formed shape of a component into a strip layout and working backwards to design the necessary tooling [2], [17]–[19]. While the desired stroke rate plays an integral role in designing a strip layout, the most important aspect is the material utilization [2], [3], [20]. Raw material costs often represent around 75% of total costs in stamping facilities [4]. With the high production rate of progressive dies, incremental improvements in material utilization can lead to large cost savings. For example, a tooling design that is adjusted to save a modest 10 grams of material per part and operates at 100 strokes per minute will save 1 tonne of material roughly every 16 hours of operation. Further justification for reducing material utilization beyond cost efficiencies is found in the environmental factors. Research conducted by Horton *et al.* [21], [22] found that yield losses amount to 44% of total sheet metal used which can be reduced in the strip design and layout

phase. Their study noted that across 46 passenger vehicles, the average material utilization for sheet metal components was 56% which results in 18 million tonnes of scrap metal per year. The study found that improving average material efficiency for passenger vehicles to 70% could reduce the global annual embodied emissions from sheet metal components by 26% (25 million tonnes of CO_2) and reduce material costs by 24% (£8 billion). Framing this in a global context, a study by Allwood *et al.* [23] estimated that steel and aluminum usage account for 10% of global anthropogenic carbon dioxide emissions. Additionally, about one fifth of aluminum and steel produced is cold-rolled for use in sheet metal products [24], [25].

2.1.4 Influencing Variables

Designing, operating, or researching sheet metal forming processes requires a fundamental understanding of the process and the physical phenomena involved. Sheet metal forming is inherently comprised of a multitude of complex interactions that should be understood as a system of interactions. Varying one process parameter such as the tooling or component geometry can have a cascading effect. For deep drawing operations, both die and punch radius influence the behaviour of the sheet metal and required forces [26]. A sharper punch radius will require higher forces to fold the sheet metal around the punch nose and can result in thinning or tearing. For V-bending, increasing the die radius will required higher punch forces during the initial stage of bending and decreased forces in the final stages [27]. Furthermore, increases in punch and die radii result in increased springback after forming [28]. The relative size of components is also important, such as the ratio of the blank and punch diameter (draw ratio) which informs the number of required drawing steps [3]. Finding optimal binder pressure is a challenging process that is well researched [29]–[31]. A sufficient binder force is required to prevent wrinkling along the sheet metal surface, but, an excessive binder force can lead to tearing [29]. An overview of the major process parameters that influence a sheet metal forming operation can be seen in Table 1. There are numerous interrelated variables that influence the sheet metal forming process and are often addressed in design handbooks. However, key aspects that must be taken into account are the mechanical properties of the sheet metal strip itself [14]. Understanding the behaviour of the flow stress for various strain and stress levels, strain rate dependencies, temperature sensitivities and formability limits is required to properly deform sheet metal into a desired shape without material failure.

Table 1: Overview of the significant variables in sheet metal forming and which aspect they influence [14].

Aspect	Variables
Sheet material or blank	Flow stress as a function of strain, strain rate, temperature and microstructure
	Formability as a function of strain, strain rate, temperature, strain path and microstructure
	Surface texture
	Thermal properties (density, melting point specific heat, thermal conductivity and expansions)
	Hardening anisotropy
	Geometric anisotropy
	Blank size, location and thickness
Deformation zone	Deformation mechanics
	Metal flow, velocities, strain, strain rate
	Stresses (variation during deformation
	Temperature (heat generation and transfer
	Damage accumulation
Tooling	Tool geometry and forces
	Surface conditions
	Material/heat treatment/hardness
	Temperature (heat generation and transfer)
Equipment used	Speed/production rate
	Binder design and capabilities
	Force/energy capacity
	Rigidity and accuracy
	Flexibility
Product	Geometry
	Dimensional accuracy/tolerances
	Surface finish
	Microstructure, metallurgical, and mechanical properties
Condition at tool/material interface	Lubricant type and temperature
	Insulation and cooling characteristics of the interface layer
	Lubricity and frictional shear stress
	Characteristics related to lubricant application and removal
Environment	Available manpower
	Air, noise and wastewater pollution
	Plant and production facilities and control

2.2 Sheet Metal Formability

Formability is arguably the most important characteristic of a material used in sheet metal forming. While the term is commonly used, formability is often understood contextually. On the press shop floor, formability will refer to a materials ability to pass through all the forming operations without presenting any problems [32]. However, in the research community, in which quantitative measurements are

necessary, formability refers to the amount of material strain prior to fracture [32]–[35]. Forming limit diagrams (FLDs) are one of the most commonly used methods for measuring the formability of sheet metal components. Generally, an FLD comprises a plot of the failure locus at the onset of necking, known as the forming limit curve (FLC), using the major strain (e_1) and minor strain (e_2) as coordinates. The FLD was first introduced by Keeler [36] for the positive minor strain region (tension-tension domain) and extended by Goodwin [37] for the negative minor strain region (tension-compression domain). The two most common approaches to constructing an FLD are the Marciniak test [38] (in-plane) and Nakazima test [39] (out-of-plane), also known as the Limiting Dome Height (LDH) test. The basic difference between the two test setups is the punch shape; the Nakazima test uses a hemispherical punch whereas the Marciniak uses a cylindrical punch (Figure 9). As a result of using a hemispherical, the failure location is more controlled and likely to occur at the apex of the nose and therefore has received wider adoption as compared to the Marciniak geometry which has less predictable failure.

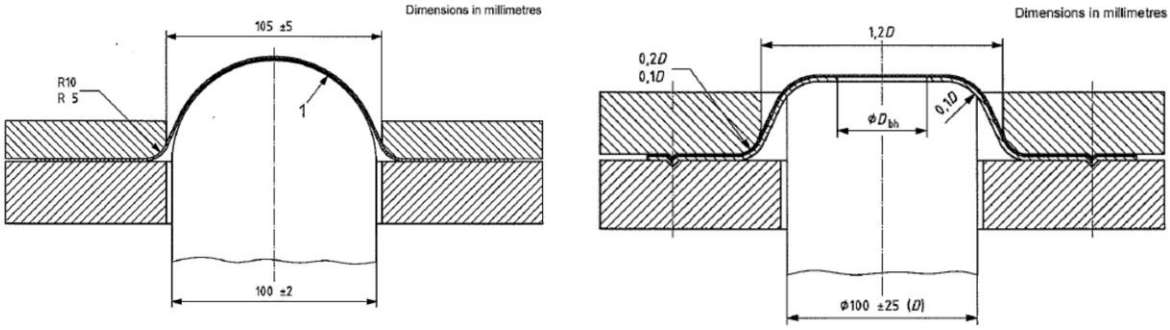

Figure 9: Experimental test set-up used in ISO 12004-2 (2008) for Nakazima (left) and Marciniak (right) tests [40].

There are three fundamental testing conditions to construct a basic FLC; uniaxial tension, plane strain tension and equi-biaxial tension which all correspond to unique strain paths on the FLD (Figure 10). The formation of FLC are highly strain path dependent [41]. During a one-step forming operation, strain paths are nearly proportional and maintain a nearly constant ratio between the major and minor strain. As a result, a given FLC is strictly valid only for processes which have linear loading paths [42]. However, stamping operations in industry often involve multi-stage operations in which secondary operations may involve drawing operations with principal strain increments which differ from the initial operation. To address this limitation of FLDs, a forming limit stress diagram (FLSD) which is independent of strain path was proposed by Kleemola and Pelkkikangas [43] and independently by

Arrieux *et al* [44]. Unlike FLCs, which can be constructed directly from experimental data by measuring the strain distribution, FLSDs cannot be constructed directly from experimental results since they use principal stress as coordinates which are not easily obtained in experiments. Stoughton [45] showed that a FLD can be transformed into an FLSD by selecting and applying a yield criterion and hardening law.

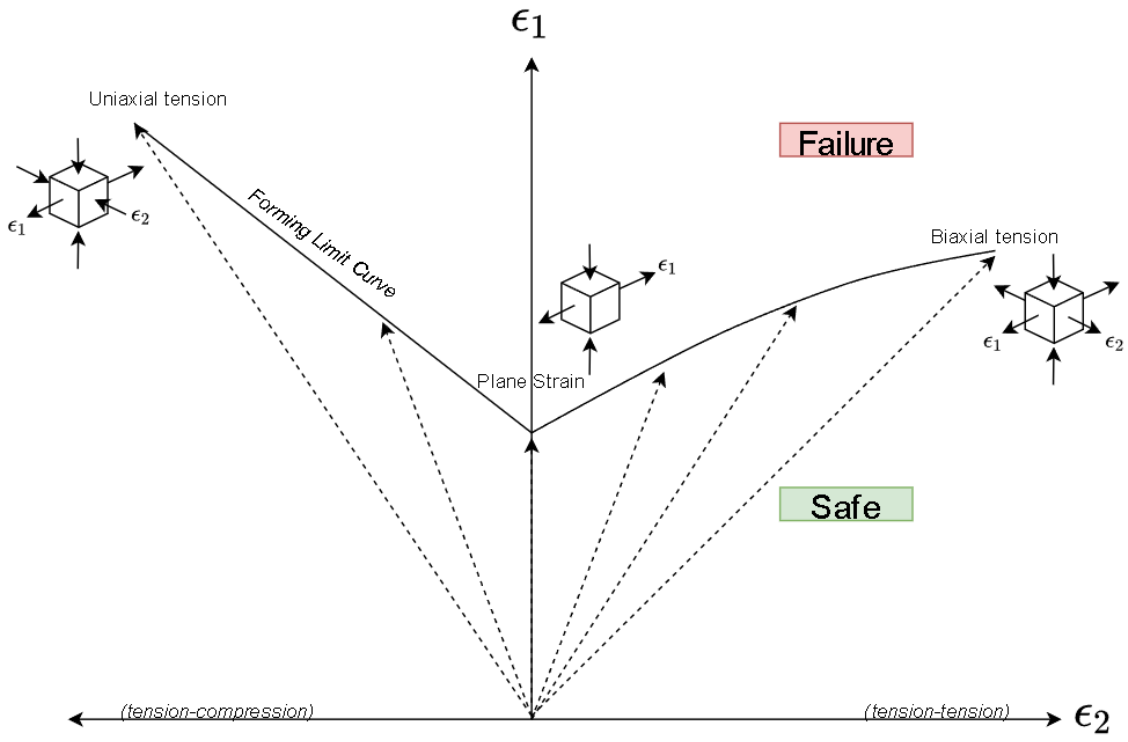

Figure 10: Forming limit diagram with basic strain paths.

Many theoretical models have been developed to predict the formability of sheet metal to address the time-consuming nature and high cost of experimental testing. Zhang *et al.* [46] categorized most of the numerical and theoretical formability models as being based on either: bifurcation theory; geometrical imperfection; or continuum damage approaches. Classical models used to predict sheet metal formability were based on bifurcation theory and proposed by Swift [47] and Hill [48]. One of the most widely used methods is the M-K model, first proposed by Marciniak and Kuczynski [38] which is based on the assumption of a pre-existing geometrical imperfection. The M-K model assumes the existence of a defect in the material such as a variation in the material thickness which allows for the derivation of system of equations which can be solved numerically to predict onset of localization within the defect

and determine an FLC. While there is plenty of research into formability models, they still do not obfuscate the use of experimental techniques such as the Nakazima test.

2.3 Control in Progressive Sheet Metal Forming

The actuation in progressive sheet metal forming operations is largely controlled by two components; the press and the strip feeder. The press is responsible for all actuation prescribed in the vertical direction, including the punches, binder, locating pins, and strip lifters. The punches and locating pins are fixed to either the upper or lower half of the press and follow the exact relative displacement of the press stroke. The binder and strip lifters are both fixed to passive springs which are actuated by the press stroke. The binder follows the same trajectory as the press, but, remains fixed when the punch is in contact with the strip. Similarly, the lifters follow the press trajectory when rising and lowering, but, remain fixed in their upper position during strip feeding and fixed in their lower position when the punch is in contact with the strip. There are three types of press used in industrial stamping applications; mechanical, mechanical servo press, and hydraulic presses [49]. Mechanical presses are the oldest and most common press type and are actuated by an electric motor which powers a fly-wheel based drive system. The major benefits of mechanical presses include their relatively lower initial cost, high accuracy and repeatability, and most importantly, their high stroke rate. Mechanical servo press differ from mechanical presses by their use of servo motors to directly power the vertical translation of the press without a flywheel. The use of servo motors allows for the press to have precise control over the press motion throughout the entire stroke, allowing for a high degree of flexibility in prescribing the press motion, but, come with a high initial cost [50]. Hydraulic presses are actuated by hydraulic cylinders and therefore have lower cycling speeds than mechanical presses but offer flexibility in their stroke length and pressure and are often used for deep drawing applications with complex geometries [51].

Similar to presses, there are multiple options for strip feeding units used in progressive die operations. The two types of strip feeders used in industry are grip feeders and roll feeders. Grip feeders use a mechanical clamps to grip and translate the strip whereas roll feeders translate the strip by pulling or pushing the strip between two spinning wheels [52]. Conventional feeders are driven directly by the press and are best suited for translating heavy strips. More modern feeders are servo driven and allow for flexibility in selecting the time allotted to feeding during the press stroke (feed angle), but, are

restricted by the mass of the strip [53]. Roll feeders, and more specifically, servo roll feeders are most commonly used and preferred in industry to their high performance and flexibility [54], [55].

Both the press and strip feeder use closed-loop control to follow precise tooling paths, but, these control systems exclude measurements of the product properties in their control loop [56], [57]. In the case of progressive die operations, tool operators have limited inputs to control the product properties once a tool is designed and fabricated. The primary control input from tool operators is the stroke rate, which informs the press speed at which it operates. Additional fine tuning of the strip feeding duration and the displacement profile of the press are possible with servo-actuated tooling. However, the aforementioned control inputs do not account for the final geometry or properties of the final sheet metal product in their control loops and are identified through experimental testing during die try-out. An experience tool operator will have developed an intuition for selecting the stroke rate and other control inputs, but, there is lack of tools that can be utilized to identify stroke rates and other control inputs prior to experimental testing.

With the adoption of Industry 4.0, tool makers are increasingly interested in incorporating novel sensing and actuating technology to directly measure and control product properties. The majority of research into control of sheet metal forming applications looks to either avoid failure or control product properties [58]. The introduction of these sensors to monitor the state of the work piece requires active actuation to control the behaviour of work-piece in response to its state. Active actuation in progressive die operations is largely limited too more advanced tooling systems which utilize servo strip feeders, servo or hydraulic presses and in some circumstances, actuated binders. The most common research areas in control of sheet metal forming operations is adapting the binder pressure to control the material flow during drawing operations to minimize wrinkling and tearing [59], [60]. Endelt *et al.* [61] were able to prevent tearing and minimizing wrinkling by controlling material flow in a deep drawing operation by using laser displacement sensors to measure the edge draw-in and applying hydraulic pressure to increase or decrease the blank holder pressure. Fischer *et al.* [62], [63] developed a control system to account for the variability in material properties during deep drawing operations. They employed eddy-current sensors to measure variations in thickness and yield strength of incoming sheet metal to be used in a closed–loop control of the forming process. An optical measurement system was used to measure edge draw-in of the forming process and the local blank holder force was adapted by the use of four hydraulic cylinders. As with any closed-loop control system, these approaches, that seek to minimize wrinkling and tearing, can be generalized as using some method for sensing or observing

the state of the system (*forming operation*), and a means of actuating and controlling the state of the system (*binder pressure*). Methods that have been applied to measure the material flow include using laser displacement sensors [64], inductive coils [65], eddy current sensors [66], and optical sensors [62] (Figure 11). Furthermore, controlled actuation of the binder force can be realized by independent hydraulic [67], piezoelectric [68], or gas spring [69] actuators. The other major application for adapting binder pressure is for minimizing spring back after forming operations [70]–[72]. Ultimately, controlling the strain distribution in the sheet metal work-piece is essential during forming operations to minimize part defects and the primary method for control actuation is a distributed blank holder force. A less common alterative to adapting the blank holder force is the use of adjustable draw-beads which can restrict material flow based on the bead depth [73], [74].

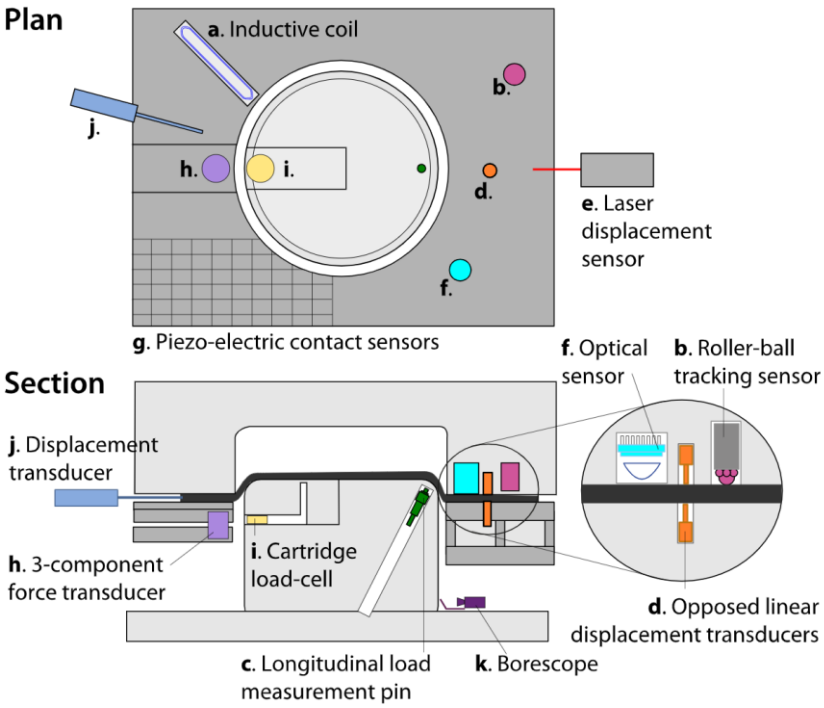

Figure 11: Example sensors and their placement for measuring material draw-in during a forming operation [58].

With the exception of the binder, applications of closed-loop control in progressive sheet metal forming is limited to servo driven presses and servo driven strip feeders which receive considerably less attention. Servo driven presses provide an interesting opportunity for tuning the displacement profile of the punch to the specific geometry being formed. While there is limited research on the topic,

adjusting the displacement profile of the punch has been shown to reduce thinning [75], [76], tearing [77], and a reduce process noise which results in extended tooling life [78], [79]. These methods typically reduce the speed of the punch immediately before contacting the sheet metal to reduce the punch force and velocity and subsequently speed up once being removed from the material to maintain a high production rate. Unlike the approaches which adapt the binder pressure, there is currently no research into actively adapting the punch displacement in response to the product behaviour and, instead, the same selected trajectory is followed for each stroke which is optimized experimentally or with the use of FEA.

While servo feeders have the ability to adjust the duration of feeding during the crank angle (feed angle), the author is unaware of any research with respect to optimizing servo driven feeders in progressive die operations. However, research has been conducted into optimizing collision-free tool paths for mechanical grippers used in transfer die sheet metal forming operations [80], [81]. Some research even incorporates the elastic response of the deformable sheet metal when optimizing the gripper's trajectory [82], [83]. However, the cases which incorporate the elastic response of the sheet metal only model the response of a single work-piece which is considerably simpler than predicting the response of an entire sheet metal strip which includes multiple work-pieces. The lack of research conducted into optimizing the feeding process in progressive die operations likely stems from the difficulty of predicting the response of an entire sheet metal strip and the cost and risk associated with experimental testing. Hence, this avenue of research, the focus of the current thesis, is ripe for new work.

2.4 Machine Learning

2.4.1 Overview

Machine learning (ML) has received wide adoption in the research community and commercial applications in an array of industries due to convergence of big data and sufficient computational power [84]. ML is often used synonymously with artificial intelligence (AI) in the media, but, is in fact a subset of AI that uses algorithms which can automatically improve or 'learn' through data or 'experience'. A more clear definition of ML which originates with Samuel [85] defines ML as a computational method which builds a model from a sample of data (training data) in order to make decisions or predictions *without being explicitly programmed to do so* [85]. There are three paradigms to machine learning; supervised learning, unsupervised learning, and reinforcement learning [86]. A

task is considered as supervised learning if both the input and output data have known labels such as classifying handwritten digits or predicting housing prices. Unsupervised learning consists of tasks in which a set of input data does not have any corresponding output labels. The goal in unsupervised learning is to identify the structure of the data such as groups of similar examples (clustering), determining the distribution of the input data (density estimate), or projecting high-dimensional data to lower dimensional data for visualization purposes. Finally, reinforcement learning requires the algorithm to find suitable actions or inputs for a given situation which maximize a reward. Unlike supervised learning, reinforcement learning is required to learn suitable inputs through a trial and error process that is guided by a reward function. Perhaps the most famous example of reinforcement learning is *Alpha Go*, a program developed by DeepMind which learned to play the game of Go, a game which is orders of magnitude more complex then chess, and beat the highest rank player multiple times by playing [87].

The most common form of ML is supervised learning which has two main approaches: regression and classification [88]. Classification algorithms seek to find the output which is restricted to a finite set of discrete values (*classes*). Examples of classification problems include image recognition or fraud detection in which the outputs of the algorithm are discrete and often qualitative such as predicting whether the image is a cat or a dog. Regression algorithms attempt to find an output which is continuous within a range and is typically quantitative. Examples of regression problems include predicting housing prices given a set of parameters that describe the house, such as the number of rooms and size of land.

2.4.2 Neural Networks

One of the most widely used machine learning algorithms is the artificial neural network (ANN) since there are numerous architectures that perform exceptionally well on various specific applications [89]. McCulloch and Pitts [90] were the first to formulate a mathematical model for a neuron, but, without the intention of it 'learning'. Widrow and Hoff [91] were the first to apply a learn rule known as the *delta rule* which used gradient descent to update the weights of the neuron. Several decades later in 1991, Hornik [92] introduced the *universal approximation theorem* which postulates that a Feedforward Neural Network (FNN) with a single hidden layer and a finite number of neurons can approximate any continuous function on a compact set of R^n using non-constant, monotonically-increasing activation functions. In 2006, Hinton *et al.* [93] introduced a fast learning algorithm which used a greedy layer-

wise strategy to train a type of ANN called Deep Belief Networks. The work by Hinton *et al.* helped popularize the concept of deep learning and initiate the explosion of deep learning that has taken place for nearly the past two decades. A multitude of ANN architectures have emerged over the past several decades that now offer researchers a wealth of tools to tackle complex pattern recognition problems, but, also proves tantalizing to the domain scientist or engineer whom seeks to apply these architectures to their speciality.

At the foundation of each ANN architecture is the concept of the multi-layer perceptron (*MLP*), also known as a feedforward neural network (FFNN) (Figure 12), which is often used ambiguously and in reference to any type of feedforward ANN or colloquially as 'vanilla' neural networks [86]. The most basic MLP will have at least three layers; an input layer, hidden layer, and an output layer. The input and output layers simply consist of vectors which correspond to the labelled input and output data are taken from a given sample in the dataset. The hidden layer consists of one or more neurons, also sometimes called perceptron's, which attempt to learn the mapping from the input vector to the output vector, Z, by using the product between the input vector X, and learned weights, α, with a learned bias term, α_0, in a non-linear activation function ϕ, for M neurons (Equation 1).

$$Z_m = \phi(\alpha_m^T X + \alpha_{0m}), \qquad m = 1, \dots, M \tag{1}$$

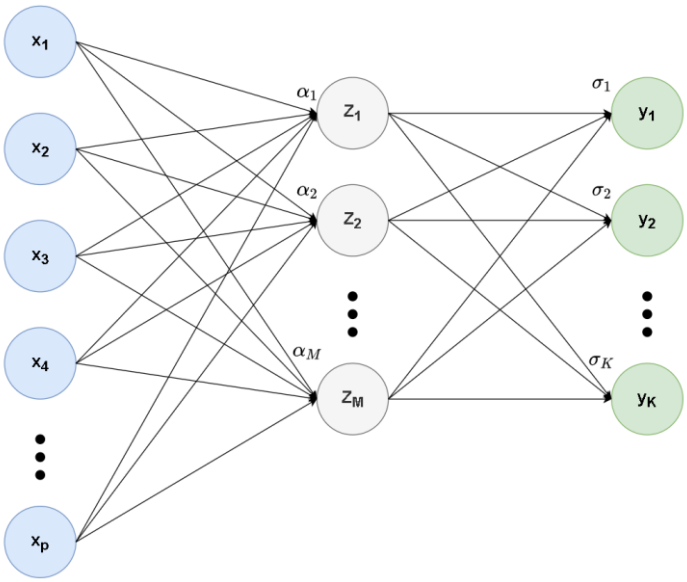

Figure 12: Basic MLP with p inputs, k outputs, and a single hidden layer.

A variety of activation functions are used in practice, such as *tanh*, *sigmoid*, *softmax*, with the most popular activation function for deep learning being the *Rectified Linear unit* (ReLU) [94]. However, the original perceptron introduced by Rosenblatt [95] used a *Heaviside* activation function. To obtain the output of the network, the output from the hidden layer is multiplied by another set of weights, β and summed with a bias term β_0 and transformed through the output activation function, σ, for K outputs. In other words, the output y_k can be expressed as:

$$y_k = \sigma(\beta_k^T Z + \beta_{0k}), \qquad k = 1, \dots, K \tag{2}$$

For MLPs with multiple hidden layers, the output of the previous hidden layer becomes the input to the subsequent hidden layer. In the case of regression, there is no output activation function. The final component to any ANN is the learning rule which adjusts the weights to successfully map the input vector, X, to the output vector, Y. For regression, a common error function is the mean squared error (MSE) which evaluates the error between the true output, y_i, and the predicted output, $\hat{y}_i$, for N samples and K outputs (Equation 3).

$$E = \frac{1}{n}\sum_{i=1}^{N} ||y_i - \hat{y}_i||^2 \tag{3}$$

For classification problems, a common error function is the average cross-entropy which can be defined as:

$$E = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} log(\hat{y}_{ik}) \tag{4}$$

In both cases, the goal is find the weights of the ANN which minimize the selected error function. The most common approach used for this optimization problem is the gradient descent algorithm which updates the weights by the product of the change in the error with respect to the weights and a scalar value known as the *learning rate*, γ_r [86], [96]. The update rule for gradient descent can be written using the following equations where E is the error function, β_{pm} is the learned weights between pth input and mth hidden neuron, α_{mk} is the learned weights between the mth neuron and the kth output, and γ_r is the learning rate for the $(t+1)$st iteration.

$$\beta_{pm}^{(t+1)} = \beta_{pm}^{(t)} - \gamma_r \frac{\delta E}{\delta \beta_{pm}^{(t)}} \tag{5}$$

$$\alpha_{mk}^{(t+1)} = \alpha_{mk}^{(t)} - \gamma_r \frac{\delta E}{\delta \alpha_{mk}^{(t)}} \quad (6)$$

There exists a multitude of ANN architectures, with widely used ones being the feed-forward neural network (FFNN), convolutional neural network (CNN), and recurrent neural network (RNN) [89].

2.4.3 Decision Trees

While ANNs receive a considerable amount of attention in supervised learning, other ML architectures can achieve comparable performances to ANNs, including support vector machines (SVM), decision trees, naïve Baye's classifiers, and logistic regression [86], [96]. From these alternative architectures, decision trees are particularly noteworthy because of their explainability and performance [97].

Decision trees use a multistage decision process to partition the space of inputs feature space into a set of rectangular regions and fit a constant output response in each region [96]. Figure 13 shows an example of a decision tree with features x_1 and x_2 and categorical outputs of {'small', 'medium', 'large'}. Decision trees are comprised of root nodes, internal nodes, and leafs. For a basic decision tree, each node considers a single feature and splits the tree into two branches or edges based on evaluation of the considered feature. Nodes are split until they contain a homogenous subsets of the data and become a leaf; beyond which additional splitting is not possible. In classification, each leaf is associated with a specific output or class label. In the case of Figure 13, the root node, which is located at the top of the decision tree, considers whether the feature x_2 is greater or less than two. If x_2 is less than two, then the left branch is followed, otherwise the right branch is followed. A given data sample will follow a path through the decision tree by assessing the value of the feature considered at each node until arriving at a leaf, which corresponds to the output of the decision tree. If a sample is passed to the decision tree where $x_1 = 7$, and $x_2 = 1$, then the path would lead to 'Leaf 2' and the data sample would be classified as 'Medium'.

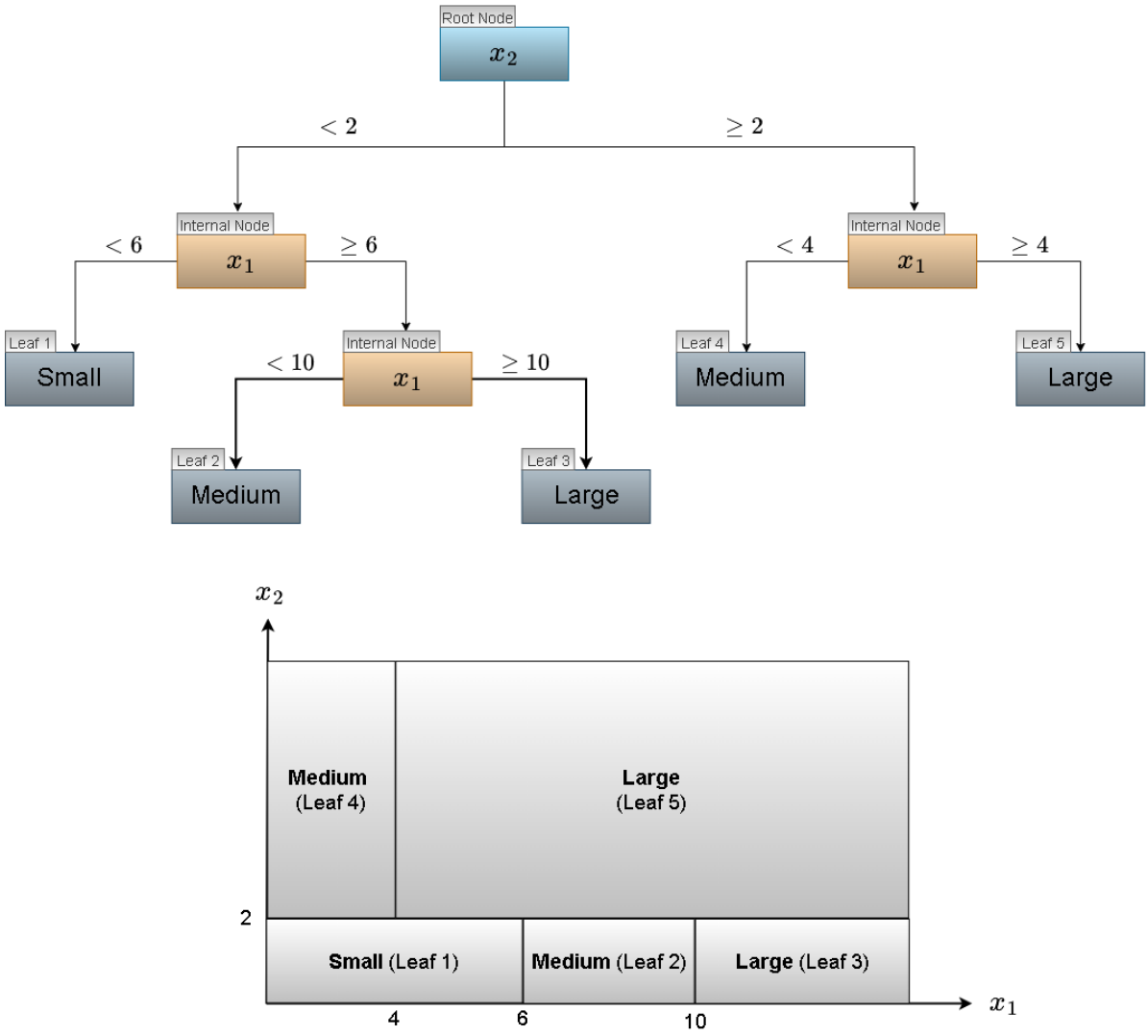

Figure 13: Example of (top) a decision tree with two features and three categorical outputs, and (bottom) the rectangular partitioning of the feature space associated with the decision tree.

Depending on the type of feature considered, the split may be expressed differently. For categorical features, there could be as many branches and child nodes as there are possible values the feature could take and therefore a unique path for each feature value. Alternatively, the feature values may be grouped together to reduce the number of branches and paths. For continuous features, the split uses a threshold on the coordinates of the vector space of the feature as demonstrated in Figure 13. Alternatively, a linear combination of continuous features may also be used to define a split.

To grow a decision tree, a means of generating node splits is required. This may be done through the definition of node impurities which measures the homogeneity of the labels at a given node. For classification, the two most common measure of node impurity are the *Gini criterion* and *cross-entropy* shown in Equation 7 and 8 respectively [86].

$$Q_m = \sum_{k=1}^{K} f_k(1 - f_k) \tag{7}$$

$$Q_m = -\sum_{k=1}^{K} f_k log f_k \tag{8}$$

In Equation 7 and 8, Q_m is the measure of impurity of node m and f_k is the frequency of class k in node m. For regression problems, the most commonly used measure of node impurity applies the sum of squares as seen in Equation 9.

$$Q_m = \sum_{i=1}^{N_m} (y_i - \mu_m)^2 \tag{9}$$

N_m is the number of observations in node, m, y_i is the true value of a observation i, and μ_m is the average value of all the observations in node m. A tree is grown by recursively splitting the terminal nodes, also known as leafs. The input feature and the split value which minimize the node impurity measure, Q_m, are selected to create the split. A tree may be grown until every unique observation in the training data is associated with a single terminal node leading to a very large and complex tree. This approach of fitting every observation to a leaf would lead to overfitting of the training data and would hinder the models generalizability. Therefore, decision tree algorithms often implement either a stopping rule, or, a means of pruning the tree to prevent overfitting.

A stopping rule prevents a node from being split unless the impurity function changes by a specified threshold. However, this approach is limited by the requirement of manually specifying a threshold. Furthermore, the current split may result in a minor decrease in impurity, while subsequent splits may lead to more drastic decreases in impurity. The more commonly adopted approach to handling overfitting is pruning.

A basic approach to pruning is to first create a large tree, T_0, and then prune it using *cost-complexity pruning* to find a subtree, $T \subset T_0$, which minimizes a cost complexity criterion, $C_\alpha(T)$ (Equation 10).

$$C_\alpha(T) = \sum_{m=1}^{|T|} Q_m(T) + \alpha|T| \tag{10}$$

In Equation 10, $|T|$ is the number of terminal nodes in tree T, $Q_m(T)$, is the impurity measure of node m in tree T, and, $\alpha \geq 0$ is a tuning parameter which controls the trade-off between the size of the tree and the ability to predict the training data. A large value of α, produces a small tree, while a value of $\alpha = 0$, would produce T_0. Brieman *et al.* showed that for every α, there is a unique subtree T_α which minimizes $C_\alpha(T)$ by using *weakest link pruning*. The value of α is often found using five or ten fold cross-validation by selecting the value of α which minimizes the cross-validated prediction error.

Decision trees may be augmented with techniques such as *ensembles* to reduce their prediction error. Ensembles use multiple 'weak' learners which by themselves can only produce accurate predictions for a subset of training data, but when multiple 'weak' learners are aggregated together and 'vote' on the predicted output, can outperform any individual learner [86]. The two most popular ensemble techniques are known as *bagging*, and *boosting.* Bagging is used to reduce the variance in machine learning models and help prevent overfitting by training multiple weak learners on various subsets of the training data which are then aggregated to produce a prediction. The various subsets of data are randomly sampled with replacement from the training data through a technique known as *bootstrapping*. A bagging model used for regression averages the prediction over the collection of weak learners and is defined by Equation 11, where there are *B* bootstrap subsets of data and a weak learner, f_b, associated with each subset.

$$f_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{11}$$

Boosting contrasts bagging as it does not use bootstrapping to generate subsets of data for each weak learner. Instead, bootstrapping sequentially grows the weak learners sequentially, allowing each one to focus on the samples that the previous leaner found difficult. A boosted model may be written as:

$$f(x) = \sum_{b=1}^{B} \lambda f_b(x) \tag{12}$$

where each f_b corresponds to an independent tree structure, and the shrinkage parameter, λ, is a small positive number which controls the rate of which the boosting model learns. Amongst one of the most popular boosted decision trees algorithms is the eXtreme Gradient Boosting (XGBoost) algorithm [98]. The XGBoost algorithm introduces multiple novelties, including a sparsity-aware algorithm for proposing candidate node splits, and a regularization term in the objective function to prevent overfitting of the boosted model. The objective function, $\mathcal{L}$, at iteration b may be written in the general form as:

$$\mathcal{L}^b = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{b-1} + f_b(x_i)\right) + \Omega(f_b) \tag{13}$$

The first term in Equation 13, $l\left(y_i, \hat{y}_i^{b-1} + f_b(x_i)\right)$, is a differentiable convex loss function that measures the difference between the target output, y_i, and the prediction from the previous iteration $\hat{y}_i^{b-1}$ combined with the prediction of the candidate tree f_b. The second term, Ω is the regularization term which penalizes the complexity of the model. At each iteration, the XGBoost algorithm adds the tree, f_b, which most improves the objective function $\mathcal{L}^b$. The XGBoost algorithm has gained popularity for its scalability to large datasets and its performance in machine learning competitions.

2.4.4 Applications in Sheet Metal Forming

While a variety of machine learning architectures are available and can often be used interchangeably to achieve similar results, the vast majority of applications of machine learning in sheet metal forming adopt ANN architectures, as indicated by several review papers [9], [99]. Machine learning is a very general tool that is akin to function estimation and therefore covering the depth and breath of applications in a given field of research is challenging. However, there are some key areas of research which utilize machine learning techniques in sheet metal forming, such as, for springback prediction, material modeling, and parameter identification. Several methods use the forming conditions and material properties to successfully predict springback in forming operations by using ANN trained on a limited amount of experimental data [100]–[104]. The use of ANN to predict springback can bypass the complexity of developing material models and allow for quicker optimization of forming parameters by using the ANN as an objective function. Other approaches use ANN to more directly replace constitutive models using experimental data such as the temperature, strain rate and strain to predict the flow stress for aluminum alloys [105], magnesium alloys [106], [107], and low alloy steels

[108]. While the ANNs used to predict flow stress showed relatively low error, these implementations likely would not generalize well due to the small size of the datasets used for training the networks.

While experimental data is inherently more desirable, it is costly and time consuming to generate and therefore other approaches seek to generate additional data from FE simulations which have been validated against experimental results. Manoocherhi and Kolahan [109] created a FE model of a cup drawing operation which was experimentally validated by comparing the force-displacement curves for conditions with and without wrinkling. The FE simulations were used to train an ANN to predict the minimum section thickness in the cup drawing operation using the inputted blank holder force, coefficient of friction, and anisotropic material parameters. In a similar manner, Chamekh *et al.* [110] proposed a method to identify anisotropic Hill parameters [111] using an ANN trained on data generated from a FE model. The FE model consisted of a cup drawing operation which is performed with varying material properties. The ANN uses the coefficient of friction, hardening parameters and the minimum thickness from the FE simulations to predict anisotropic Hill parameters. Abbasi *et al.* [112] developed a FE model of a notched tensile test which was validated against experimental results using DIC strain measurements. The FE model was then used to generate data for tensile tests with different parameters for the ductile damage model which was used to train an ANN which predicts the ductile damage parameters given the elongation of the specimen at fracture. In general, these techniques treat the system they are interested in as a black-box and use machine learning as a method for parameter identification [9], [113], [114].

More recent approaches seek to utilize RNNs instead of FFNN to replace standard phenomenological models. Bonatti and Mohr [115] developed a variation of RNNs they call ‘Minimal State Cells’ to represent the state-space of the material with the minimal number of meaningful state variables. The network uses the previous state of the material and strain increment to predict the stress tensor at the next time increment. The Minimal State Cell was shown to predict the three-dimensional stress-strain response accurately for arbitrary loading paths on a variety of materials. Bonatti and Mohr [116] extend their work to introduce the Linearized Minimal State Cell (LSMC) which apply the self-consistency update rule to the Minimal State Cell. By introducing self-consistency, their network overcomes the short-fall of a standard RNN which would predict different stress responses for identical strain-paths when the different sampling rates are used. The Linearized Minimal State Cell was implemented in a commercial FE software as a user-defined material and was shown to provide excellent agreement with an elastoplastic model when comparing the von Mises equivalent stress for a shear test, tensile test, and

three-point bending test (Figure 14). Kohar *et al.* [117] used a CNN-autoencoder with a well known type of RNN called Long Short-Term Memory (LSTM) to predict the force-displacement response and mesh deformation in axial crush FE simulations. The CNN-encoder is used to embed geometric data of the mesh, along with the mass and variation in element thickness into a latent space. Separate LSTM neural networks use the latent space from the autoencoder to predict the mesh deformation and the force-displacement response. An order of magnitude reduction in computational cost was realized while maintaining high predictive capability when compared to the results from a FE simulation. While machine learning frameworks used as phenomenological models or as surrogates to FE simulations have made considerable progress in terms of speed and accuracy, their application has still been limited to relatively simple material models and geometries and have yet to be applied to modelling an entire manufacturing process.

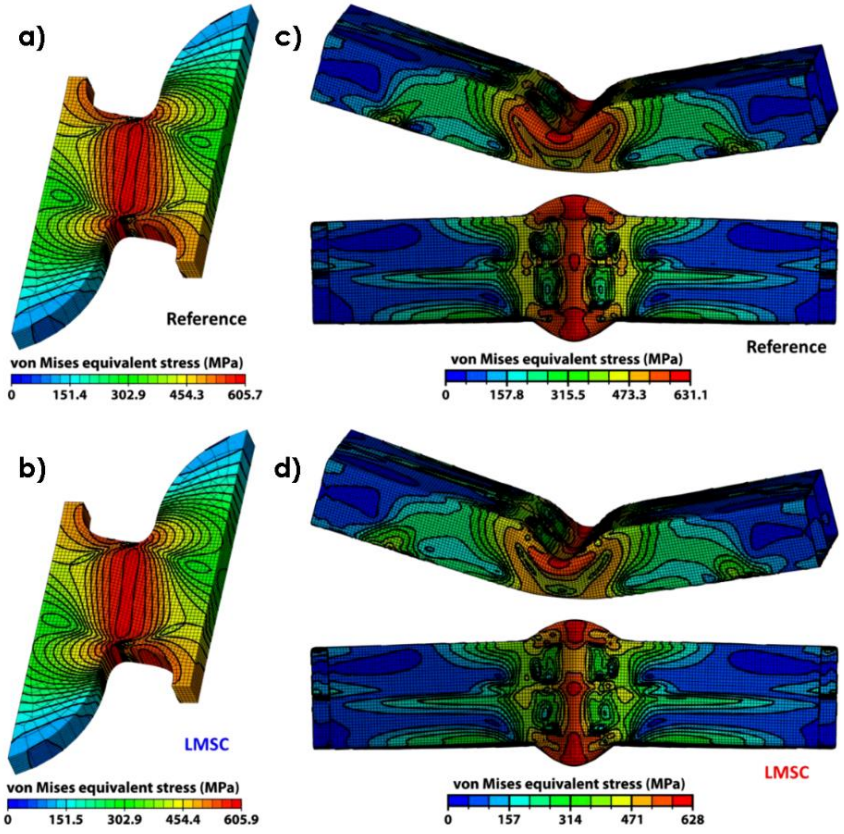

Figure 14: von Mises stress contours from shear test for the reference FEM (a) and LSMC model (b) and from the three-point bending for the reference FEM (c) and LSMC model (d) [116].

2.5 Digital Twins

The concept of a Digital Twin was originally envisioned by Grieves in the context of product life-cycle management and formalized in a 2014 paper [118]. Grieves introduced the idea of the Digital Twin as the synchronization between the data about a physical product and the information contained on the virtual product. More specifically, Grieves envisioned the merger of how a product *should* be manufactured and how a product *is* being manufactured and the instantaneous evaluation of whether the product is meeting its design specifications. In a later paper, Grieves [119] further expanded the idea of the Digital Twin to several subcategories including, Digital Twin Prototype, Digital Twin Instance, Digital Twin Aggregate, and Digital Twin Environment. The fundamental concept of a Digital Twin is a system which has physical and virtual counterparts which can interact and leverage both the physical and virtual worlds to the benefit of the entire system. For example, the virtual environment can use data from the physical environment to learn and adapt to behave in a manner more similar to its physical counterpart. As well, the physical environment can use the virtual environment to predict future behaviour and assist in providing informed decisions. An example of a Digital Twin in sheet metal forming could be implemented by simultaneously monitoring a sheet metal forming process using inline sensors to measure the contact pressure, deformation, and temperature while simulating the process in real-time to evaluate the current state of the work-piece formability in a multi-step forming process. The information from the virtual simulation can be used to inform the control inputs for the following forming operations [120].

Nearly two decades later from the initial proposal, the concept of the Digital Twin is now starting to see a larger emergence in the research community and in industry. A systematic review by Jones *et al.* [121] shows that the papers related to or about Digital Twins have seen a large acceleration in the past few years. The emerging realization of Digital Twins can be attributed to the growth and convergence in the surrounding technology that allow for its implementation such as the internet-of-things (IoT), cloud computing, and machine learning and AI. Jones *et al.* point out that Digital Twin use cases include product design, model-based engineering, health monitoring, smart vehicles, farming, but, the vast majority of applications are for manufacturing. Tao *et al.* [122] describe the use of Digital Twins in a manufacturing setting using four components, the physical shop-floor (PS), virtual shop-floor (VS), shop-floor service system (SSS), and shop-floor Digital Twin Data (SDTD) (Figure 15). The PS encompasses all the physical components involved in production such as the equipment, materials, environment, and humans. An important factor in the successful implementation of a Digital Twin is

the selection and use of the sensors and actuators used to monitor and control the PS and the technology and protocols used to communicate with the VS and SDTD. In the context of sheet metal forming, these technologies and related applications were described in Section 2.3. The VS is the virtual counterpart to the PS and incorporates virtualization tools. The physical states from the PS are used to update the models of the VS in real-time to evaluate their accuracy and provide feedback to PS to achieve nominal operating condition. The VS requires subcomponents to model the geometric, physical, behavioural and rule-based factors. Commonly used CAD tools such as NX, SolidWorks, or, Catia already provide excellent solutions to modelling the geometry of the PS. The physical, behavioural, and rules modelling describe the physical phenomenon, the machine behaviour, and the reasoning and rules behind the VS respectively. Of particular interest to researchers in the sheet metal industry would be the physical models used in a VS. Volk *et al.* [120] reviews various phenomenon and their respective models used in sheet metal forming including necking, wrinkling, tool wear, fracture, dimensional accuracy and residual stresses. For most of these phenomenon, FEA is an inexhaustible design tool for assisting in optimizing the design of the process prior to operation to account for any undesirable outcomes, but, is too slow for real-time use as is needed in the case of a VS used in a Digital Twin. For some phenomena such as wrinkling and necking, FEA can be used to establish process limits for various physical parameters, such as binder pressure and edge draw-in, and be used in control systems as mentioned in Section 2.3. However, these process limits cannot assist in identifying the future outcome of the process such as in multi-stage operations, which is a critical difference between control systems and Digital Twins. As pointed out by Volk *et al.* [120] and described in Section 2.4, machine learning models provide a computationally efficient alternative to FEA for modeling sheet metal forming processes which have the potential for inline control and use in Digital Twins, but, have yet to widely applied.

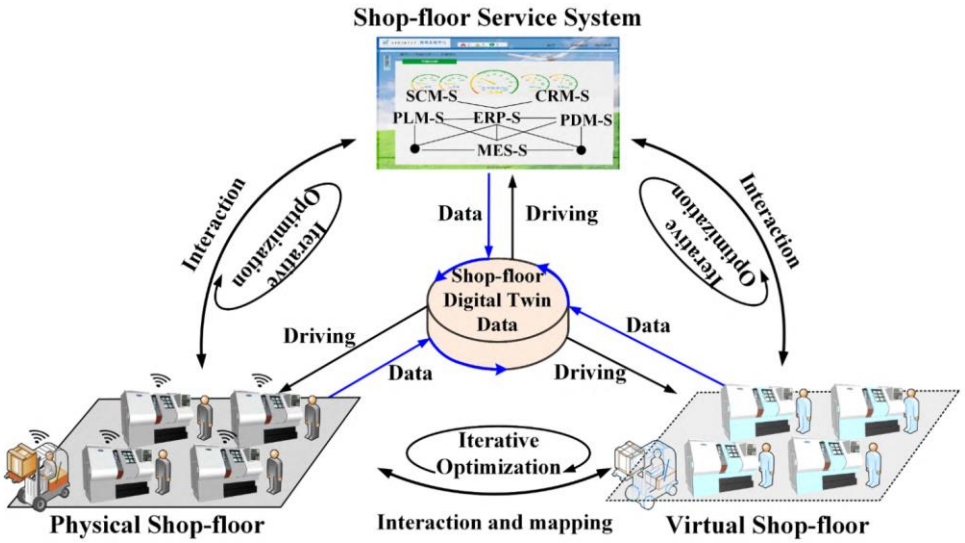

Figure 15: Interactions between the four major components of the Digital Twin shop-floor as described by Tao *et al*. [122].

The remaining components of the Digital Twin for manufacturing as described by Tao *et al.* [122] is the SSS and STDT. The SSS includes a variety of services that assist and manage the control of the PS and VS. Services provided by SSS for PS include production planning, scheduling and quality testing while services for the VS include calibration and testing of models used in VS. Finally, the STDT is the conglomeration of data from the PS, VS, and SSS. Data fusion and cleaning techniques such as Bayesian inference, Kalman filters or ANN may be applied for fusing the data from PS and VS which would make up the bulk of the STDT.

2.6 Proposed Area of Research

A considerable amount of research has gone into modelling and controlling the behavior of sheet metal during forming and shearing operations. Techniques such as FLDs provide a convenient way of predicting the onset of necking in forming operations and may inform the limitations for control strategies. Furthermore, several previous implementations of closed-loop control of the forming process have successfully control the properties of the formed part. However, an implicit assumption typically made when studying sheet metal formability and control strategies is the idealized placement of the work-piece inside the tooling. This assumption negates the possibility of errors in the placement of the work-piece or collisions during strip transfer in progressive die operations. When the strip transfer is too rapid, excessive motion of the strip (Figure 16) can lead to collisions or poor placement of the strip,

which can prevent the locating pins from aligning the strip, or result in the work-pieces being poorly positioned on the die face, leading to poorly formed components [5].

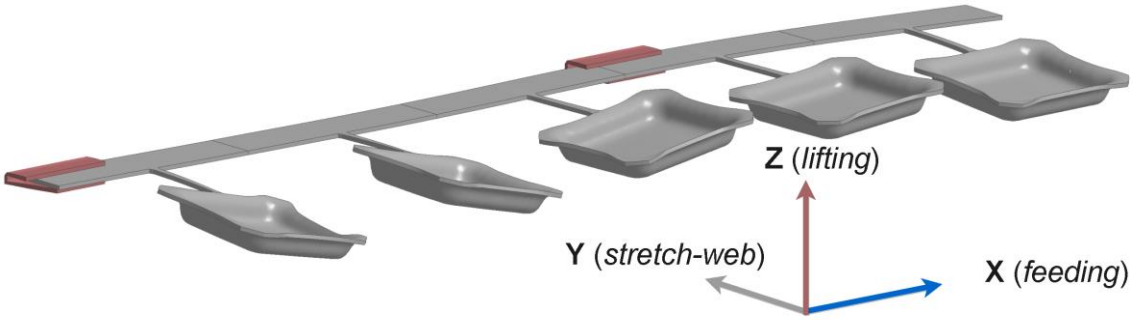

Figure 16: Excessive motion of the work-pieces as a result of rapid strip transfer.

Several factors influence the dynamic response of the strip including the selection of stretch-webs, the number of carriers, the part being produced, and the stroke rate. Progressive dies used in industry have passive spring-actuated strip lifters and tool-operators have limited ability to alter the strip progression. Therefore, in current progressive die systems, the main method for mitigating undesirable dynamic responses from the strip is to lower the stroke rate. By introducing active control to the strip lifters and feeder with adjustable input trajectories, tool-operators will have another tool to mitigate undesirable dynamic effects and can potentially maintain higher stroke rates. The current research is focused on developing a methodology for selecting suitable trajectories used to prescribe the motion of servo-actuated strip lifters and feeders that reduce or limit dynamic effects. More specifically, the current research investigates the coupled nature of the strip transfer and forming operations and the influence the strip layout has on identifying inputs to the feeder and lifter. The goal is to identify the set of input permutations of the stroke rate and trajectories for the feeder and lifter for which a given strip layout can achieve consistent part quality with limited risk of strip collisions or misalignment due to an excessive dynamic response of the strip.

To predict the coupled behaviour of the forming operation and strip transfer in progressive die operations with high fidelity, a structural-dynamic FEA is adopted. While, the FEA can be used to explore the various permutations of feeder and lifter input trajectories and their affect on specific strip layouts, FE simulations are computational expensive. Therefore, similar to the approaches described in Section 2.4.4, in the current work, the data generated from the FE simulation is used to train a machine learning model which can predict the process outcome given a set of process inputs. The trained

machine learning model is then used to readily explore the entire permutation space. The FE-ML framework captures both the modeling fidelity of FEA and the computational efficiency of ML to identify permissible process inputs for progressive die operations.

The remainder of the thesis is structured as follows. Chapter 3 outlines the methodology of this thesis, including an overview of the FE and ML modeling strategies and the demonstrator tool on which the models are based. Chapter 4 details the development of the FE models and provides some results showing the influence of the stroke rate and strip layout on the dynamic response of strip. Chapter 5 explores the data generated from the final version of the FE model and the training and testing process for the ML model which is used to find suitable inputs for the demonstrator tool. Finally, Chapter 6 provides conclusions from this research thesis and explores recommendations for possible future work.

Chapter 3
Physical Problem & Modelling Strategy

3.1 Demonstrator Tool (Description of the Physical Domain)

The methodology introduced in this thesis is meant to be generalizable to different strip layouts and progressive die operations, but, still requires a specific experimental design to be tested on. For this purpose a demonstrator tool was modelled, as shown in Figure 16. This tool was developed by TUM and MaW and is currently being implemented at TUM; access to the tool is made possible as part of the 2+2 collaborative project described in Section 1.3 of this thesis. The demonstrator tool was designed around the implementation of custom servo-actuated strip lifters which, which, to the author's knowledge, have not been used in previous industrial or research applications for progressive sheet metal forming. Additionally, a custom servo-actuated strip feeder was designed and incorporated into the tooling system. The design of the strip lifters and strip feeder will not be detailed in this work, but, their implementation in the demonstrator tool allows for adjustable control inputs to be used in the strip progression which satisfies the research goal of identifying inputs for the strip lifting and feeding that can minimize the dynamic response of the strip and permit higher stroke rates without adverse influences on the product quality.

The demonstrator tool has three primary operations: blanking, forming, and cropping. Additionally, the demonstrator tool has the ability to emboss the stretch-web to induce additional stiffness, although embossment was not considered in the current thesis research. The tooling incorporates up to sixteen stations in which tooling elements or sensors can be installed, allowing considerable flexibility. A gap, comprising empty stations (yellow in Figure 17), extends on either side of the forming operation where the work-pieces are unsupported to intentionally provide the work-pieces the opportunity to exhibit an unfettered dynamic response. To excite additional dynamic response, lifters can be removed from individual stations to create unsupported sections of the strip carrier. The dynamic response of the strip can be measured by a high speed stereo imaging or other sensors mounted inside the tooling to observe the strip between the blanking and cropping operations. It is recognized that such empty (unsupported) stations do not represent common industrial practice, however, the intent in the current research was to approximate a "worst case" scenario that will excite undesirable dynamic response of the strip so that mitigation strategies, the focus of the current research, can be assessed.

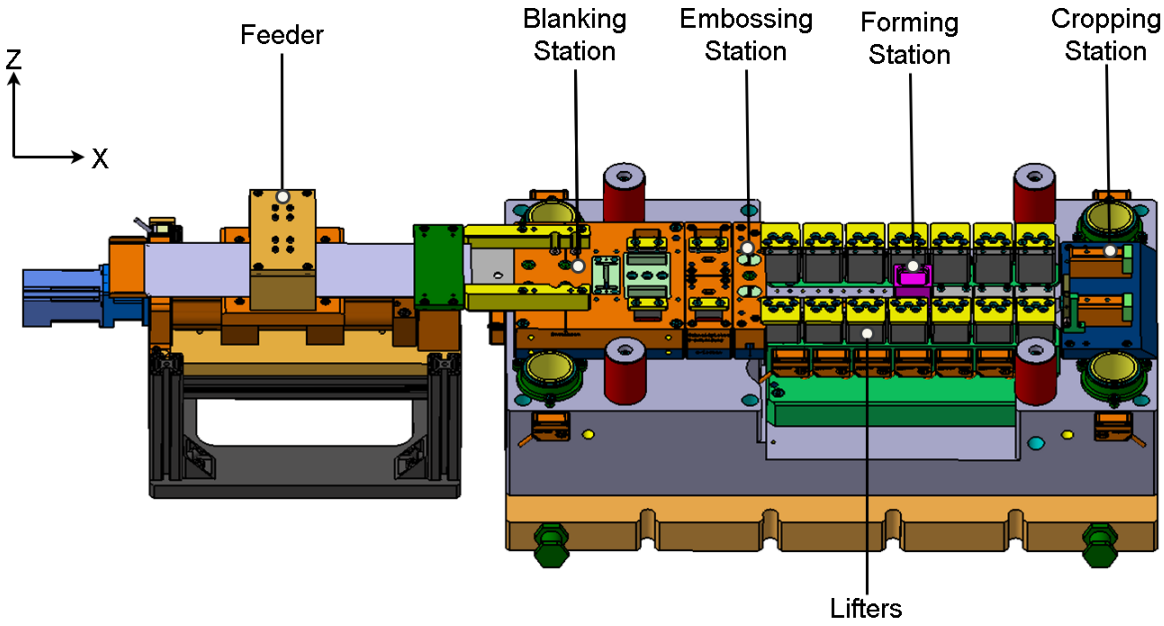

Figure 17: Overview of the CAD for the demonstrator tool showing the lower half of the tooling [123].

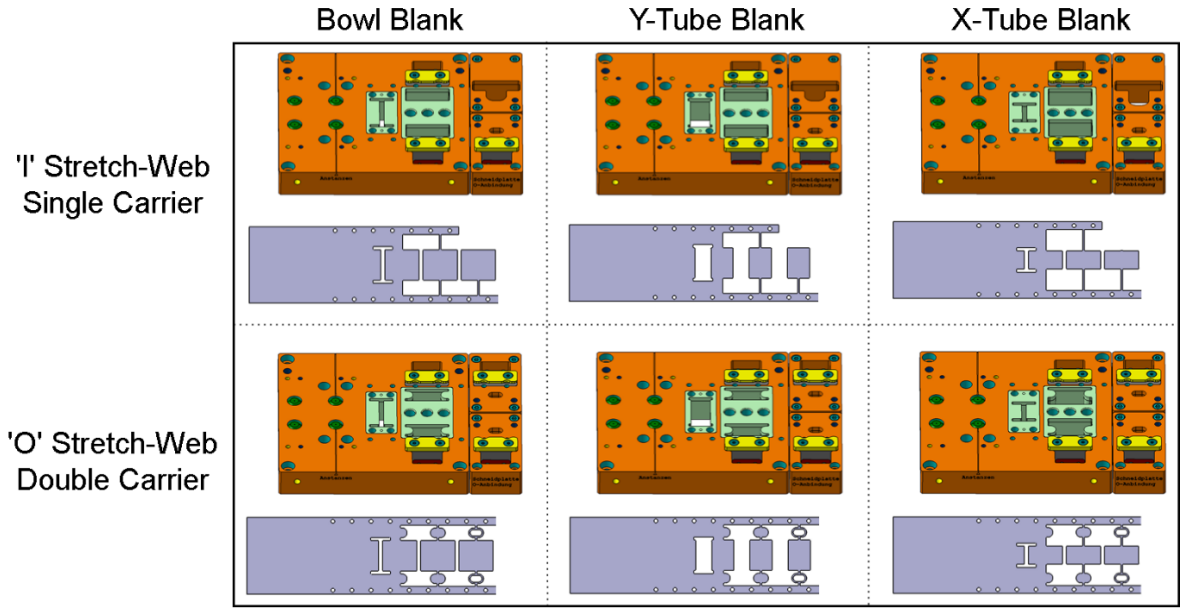

Figure 18: Strip layouts and blanking station insert modules for different strip layout configurations [123].

The blanking and forming operations use a modular design to allow for multiple strip layouts to be produced. Die and punch inserts can be replaced to create different strip layouts and formed parts. Blanking operations can produce strip layouts with either an ‘I’ or ‘O’ stretch-web with a single or double carrier configuration (Figure 18). Furthermore, three different formed parts can be produced using the same forming die (Figure 19). The depth of all of the cups is 6 mm (Figure 20), but can be adjusted using shims below the forming die to reduce the depth. To ensure sufficient clearance during the strip progression, the lifters raise the strip to a nominal height of 18 mm which aligns the strip in plane with the feeder. The length of each strip carrier which supports a given work-piece is 50 mm (Figure 21), which is also the distance the strip is progressed during the feeding phase of the stroke.

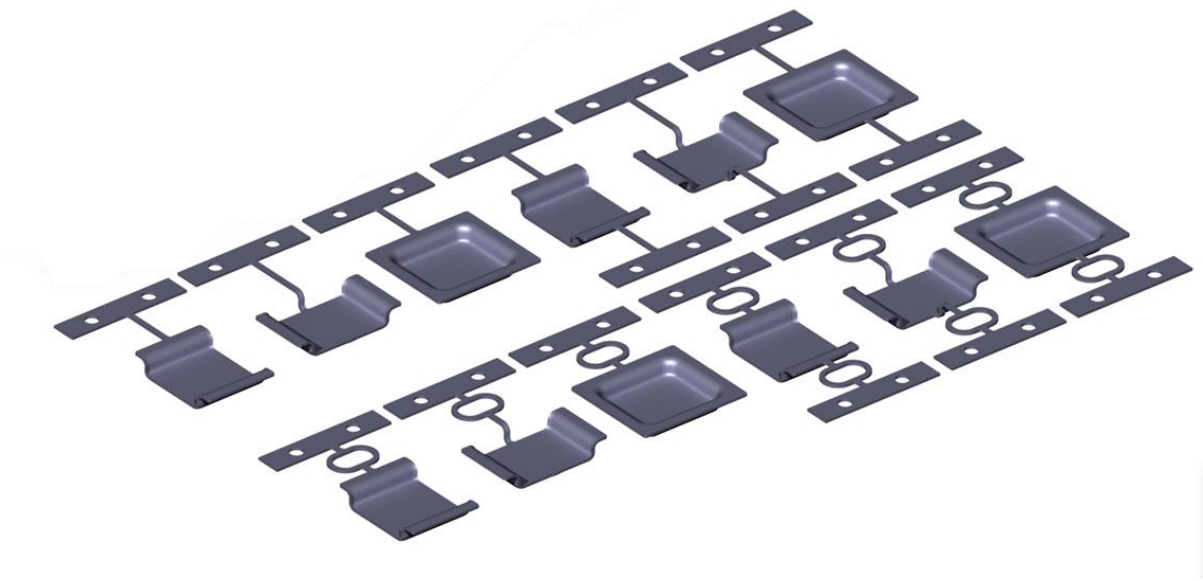

Figure 19: Overview of all stretch-web, strip carrier, and forming combinations [123].

By using various strip layouts in the demonstrator tool, the coupled relationship between the forming process and strip progression can be studied. The number of strip carriers and selection of stretch-webs will influence the stiffness of the strip layout structure which in turn affects the dynamic response of the strip. Furthermore, the selected geometry of the formed part will change the inertial properties of the work-piece which also affects the dynamic response of the strip. However, in the case of the models in this thesis, only the formed cup (Figure 20) with either an ‘I’ or an ‘S’ stretch-web configuration are considered.

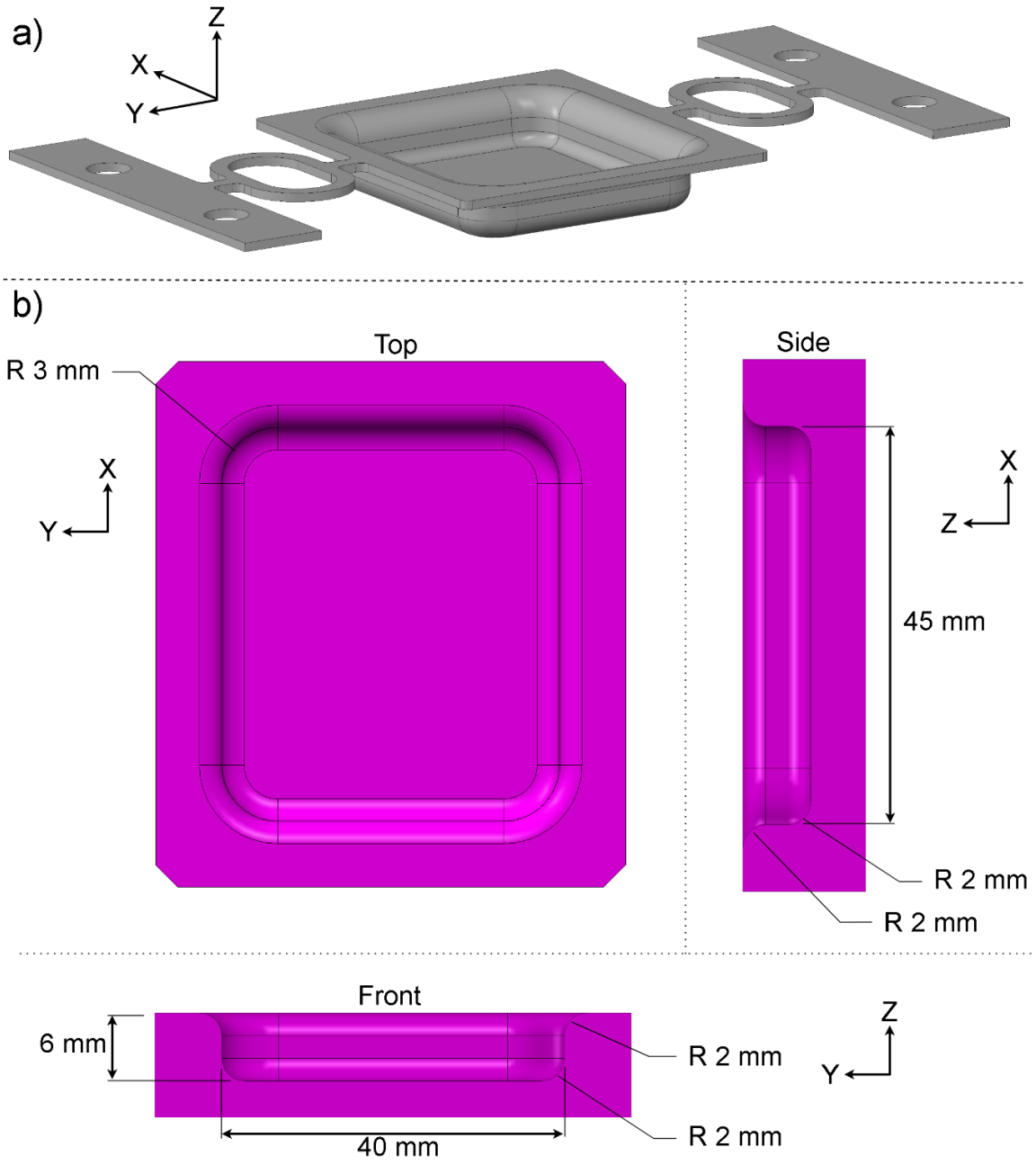

Figure 20: Overview of (a) formed cup with 'O' stretch-webs, and (b) the forming die with the nominal dimensions of the formed cup [123].

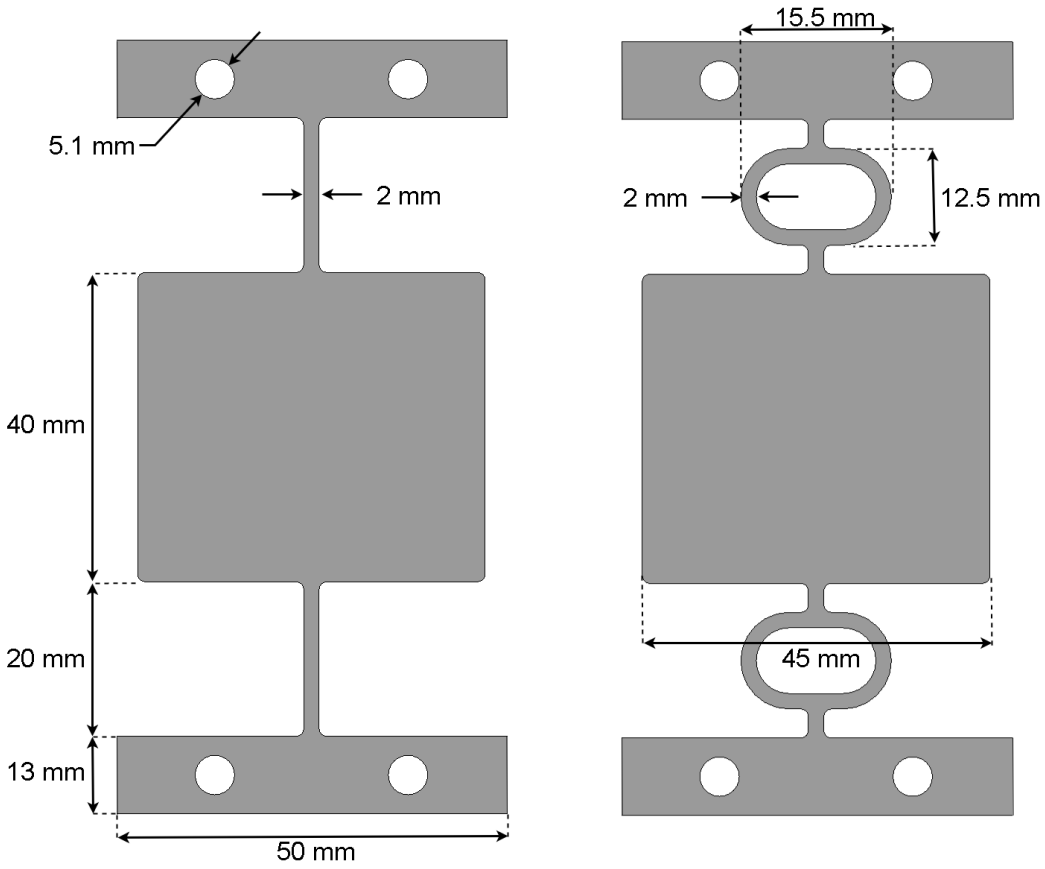

Figure 21: Dimensions of the blank geometry for the 'I' and 'O' stretch-webs [123].

3.2 Finite Element Analysis

The FE models developed in this research attempt to capture the coupled behaviour between the dynamic transfer and forming processes in a progressive die operation under different operating conditions by simulating an entire strip of sheet metal as it progresses through the tooling to form finished sheet metal components. The FE simulation, which is detailed in Chapter 4, includes the sheet metal strip and the main tooling, for example, the binder, punch, lifters, feeder, locating pins, and die (Figure 22). The operating conditions are varied by testing different permutations of feeder and lifter inputs with various stroke rates Furthermore, the influence of the strip layout on the dynamic response is explored by performing similar FEA with a variety of strip layouts and stretch webs.

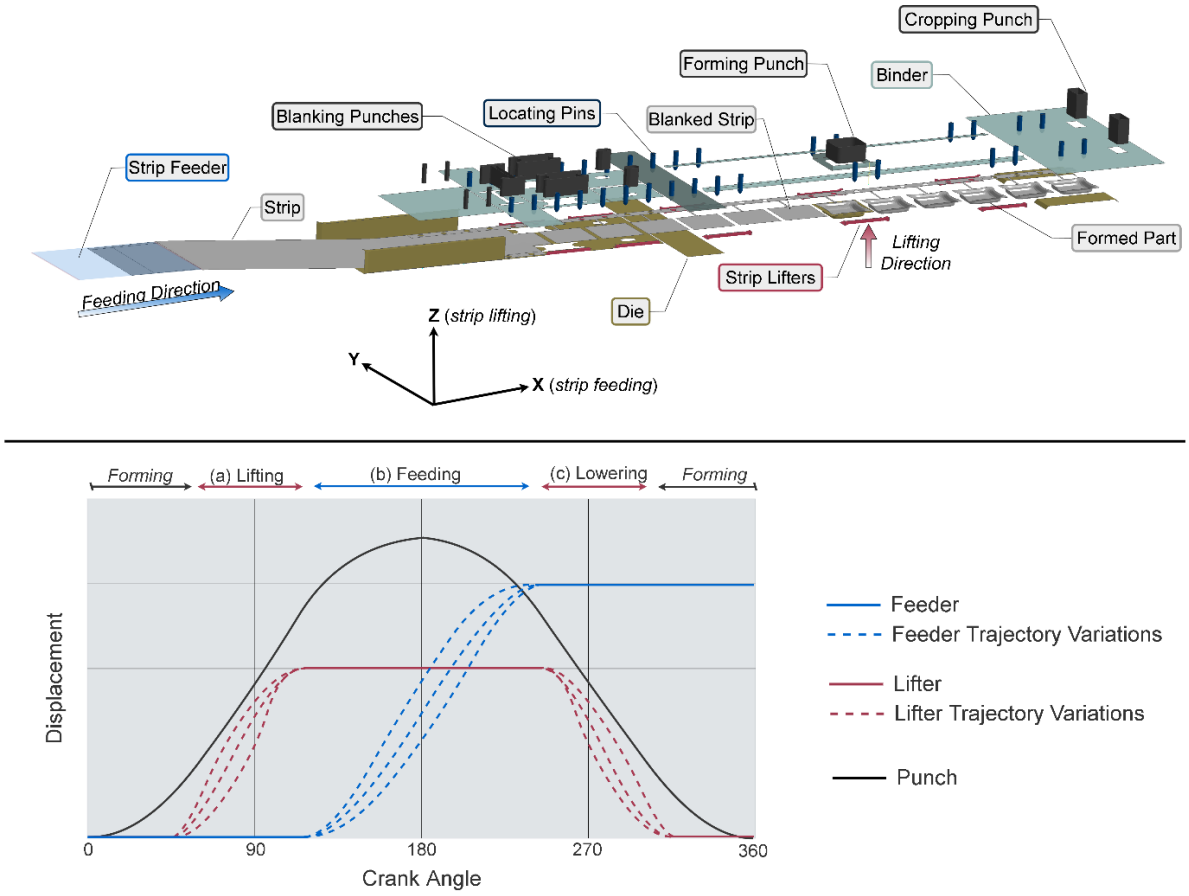

Figure 22: Example of the (top) FE model developed using the CAD from the demonstrator tool and (bottom) potential process inputs (trajectories) used in the FE simulations.

High degrees of model fidelity typically come at the cost of increasing model complexity and computational expense [124]. While the trade-off between model fidelity and computation expense is manageable in situations for which only a handful of simulations are required, this trade-off becomes a predicament when the FE model is being used to generate a dataset to train a ML model that inherently requires a large number of FE simulations to adequately learn the system. As an additional complexity, the current model's targeted prediction of the dynamic response of the strip; this requirement prevents the use of so-called "mass scaling" to reduce simulation time. In the current research, multiple revisions of FE models were developed in an effort to refine the balance between model fidelity and computational time, starting from a highly detailed model that was then simplified in order to enable use within a coupled ML framework. Part of this process entailed consideration of which aspects of the predictions were key to include in the FE model and whether there were components that could be

removed. To properly study the interactions between the dynamic strip movement and forming processes, the sheet metal components being produced in the study were of relatively simple complexity in terms of the formability challenge they represented (*i.e.* they were "easy to form"). Furthermore, the progressive die operations studied in this research were at times designed in a way which contrasts 'best practice' used in industry to intentionally induce excessive oscillations which could be studied to enable advancement of control mitigation strategies. Throughout the development of the FE model, components of the forming problem were simplified, as will be described in Chapter 4, with the intention of maintaining the quality of the dynamic response predictions.

LS-Dyna was selected to perform the FE simulations since it has strong contact algorithms for metal forming and both explicit- and implicit-dynamic algorithms to capture dynamic effects. Furthermore, this software is widely used in academia and industry to simulate impact and forming problems. LS-Dyna has a Massively Parallel Processing (MPP) version which was important in the current work to reduce simulation-time by allowing for efficient scaling of the number of CPU cores [10].

3.3 Machine Learning

The machine learning approach adopted in the current work treats the FEM as a "black box" and attempts to predict the system outcome given a set of control inputs. This approach is coined the name 'Smart Trajectory Generator' and is illustrated in Figure 23. The constructed FE model simulates a progressive die process using a subset of control inputs for the feeder, lifter and stroke rate. A dataset is generated from the completed FE simulations by compiling the control inputs and their corresponding outputs from the FE model. A machine learning architecture is then selected and trained, followed by testing on the generated test set. The ML model can be trained to map the control inputs of the lifter, feeder and stroke rate to the outcome of the process operating under those specific input conditions. Once trained, the ML model can be used to rapidly explore the entire permutation space of possible lifter, feeder, and stroke rate combinations to predict the outcome of every permutation and catalog the permutations which produce desirable outcomes. Defining what outcomes are desirable will be discussed in Chapter 4. The cataloged control inputs which produce desirable outcomes can be used to create a look-up table for tool operators to allow them to select input trajectories for the feeder and lifter which produce safe and desirable operating conditions. In the context of a virtual shop, as defined by Tao *et al.* [122] and discussed in Section 2.5, the Smart Trajectory Generator would be categorized as a production planning tool in the shop-floor service system since its usage allows tool-operators to plan

production runs by using the look-up table of permissible operating conditions for a given progressive die operation. The details of the Smart Trajectory Generator will be described in Chapter 5.

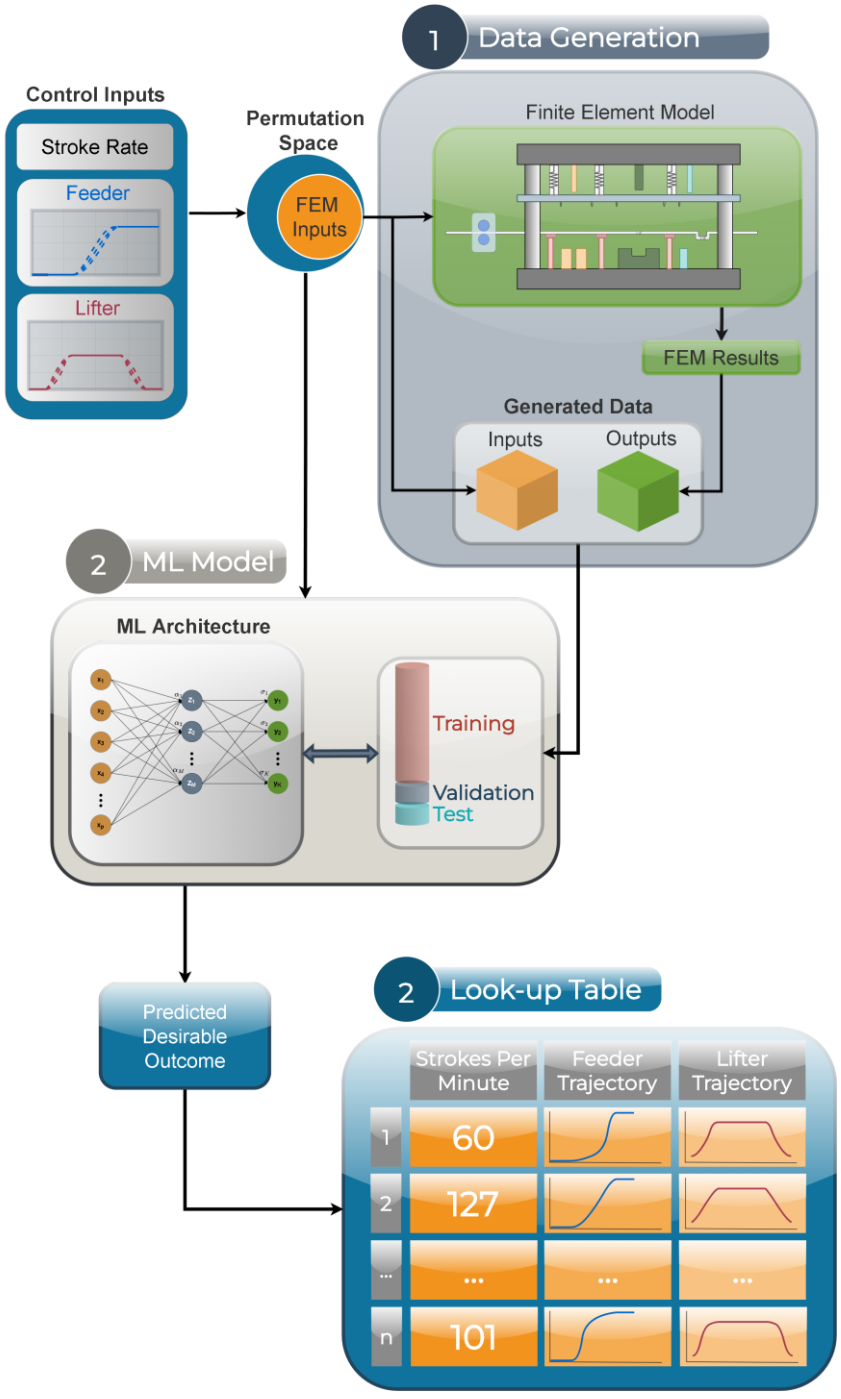

Figure 23: Overview of Smart Trajectory Generator methodology.

Chapter 4
FE Simulations

4.1 Modeling Overview

The goal of the FE simulation was to develop a virtual model that can be used to: (i) predict the mechanical behavior of the progressive die tooling under development at TUM; and (ii) generate sufficient data for training a machine learning model in the time-frame of weeks and not months. These two requirements, that the FE model be capable of experimental validation against its physical counterpart and have the capacity to generate training data for a machine learning model, are at odds since the first requires high model fidelity while the second requires lower computational expense to execute. As a result, the development of a FE model to simulate a progressive die process was by no means a trivial task. Multiple iterations of the FE model were developed in an attempt to find an appropriate balance between model fidelity and computational expense. The most significant changes seen throughout the model iterations were to the components that required the largest computational resource, such as, the mesh density and geometry. The description of the various models is given in this chapter. Section 4.2 describes the general finite element formulation using a detailed model of the progressive die operation. Section 4.3 provides an overview of the early modelling attempts which were not used to generate data, but, were necessary FE model iterations that were fundamental to the model refinement process. Section 4.4 outlines what predicted outcomes in the progressive die process model should be used to evaluate the outcome of the process. Section 4.5 details the FE models which were sufficiently computationally efficient to produce data. Section 4.6 presents the final FE model adopted to generate the data for the machine learning models. All models used an explicit dynamic formulation with the exception of the ‘Implicit’ model discussed in Section 4.3. Finally, it is noted that the experiments utilizing the progressive die tooling at TUM are still under development, such that experimental validation of the models in this thesis is still pending.

4.2 Detailed Model

4.2.1 Overview

The first model developed of the progressive die forming operation (Figure 24 and Figure 25) was also the most detailed FE model iteration. This model was developed using the CAD for the first design iteration of the demonstrator tool and therefore the mesh geometry for the ‘Detailed’ model vary somewhat from the final CAD introduced in Section 3.1 (Figure 21). While the geometry of the stretch-web and work-piece remain the same, the length of the strip carrier is 60 mm in the ‘Detailed’ model, compared to 50 mm in the final design, which consequently results in an increased strip progression (Figure 26). The original demonstrator tool design included the capacity to produce two rows of work-pieces in the tooling, which was removed from following iterations of the tooling design. The ‘Detailed’ model includes blanking, forming and cropping operations.

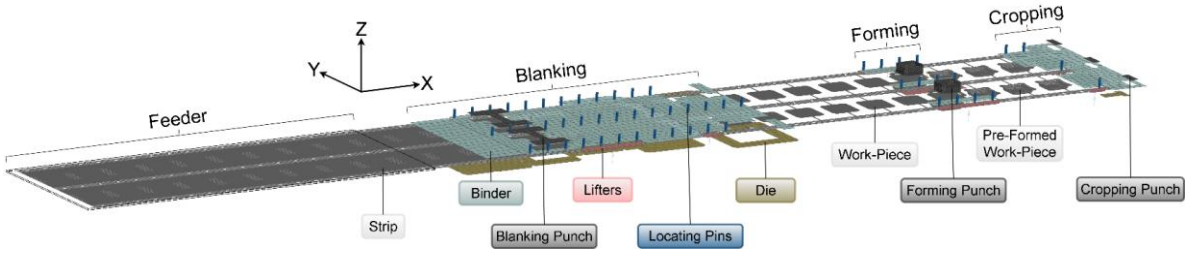

Figure 24: Overview of the 'Detailed’ model.

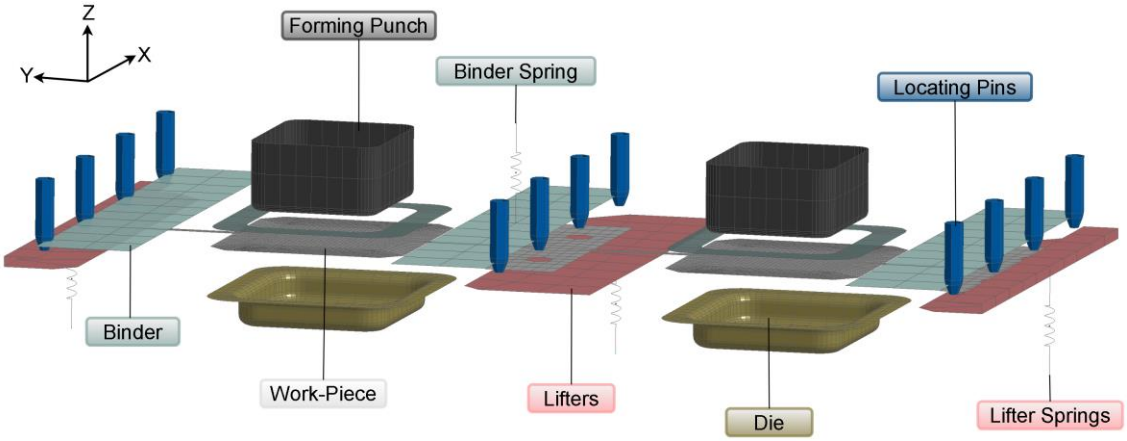

Figure 25: Forming operation of the 'Detailed’ model.

The strip layout (Figure 26) includes two rows of work-pieces and ‘I’ stretch-webs with two carriers each and a total of 30 work-pieces. The strip was meshed to include pre-blanked pilot holes and regions of the sheet that are punched out during the blanking process, which are simply removed from the simulation at the blanking station. The strip was initialized with several pre-formed work-pieces to reduce start-up transients.

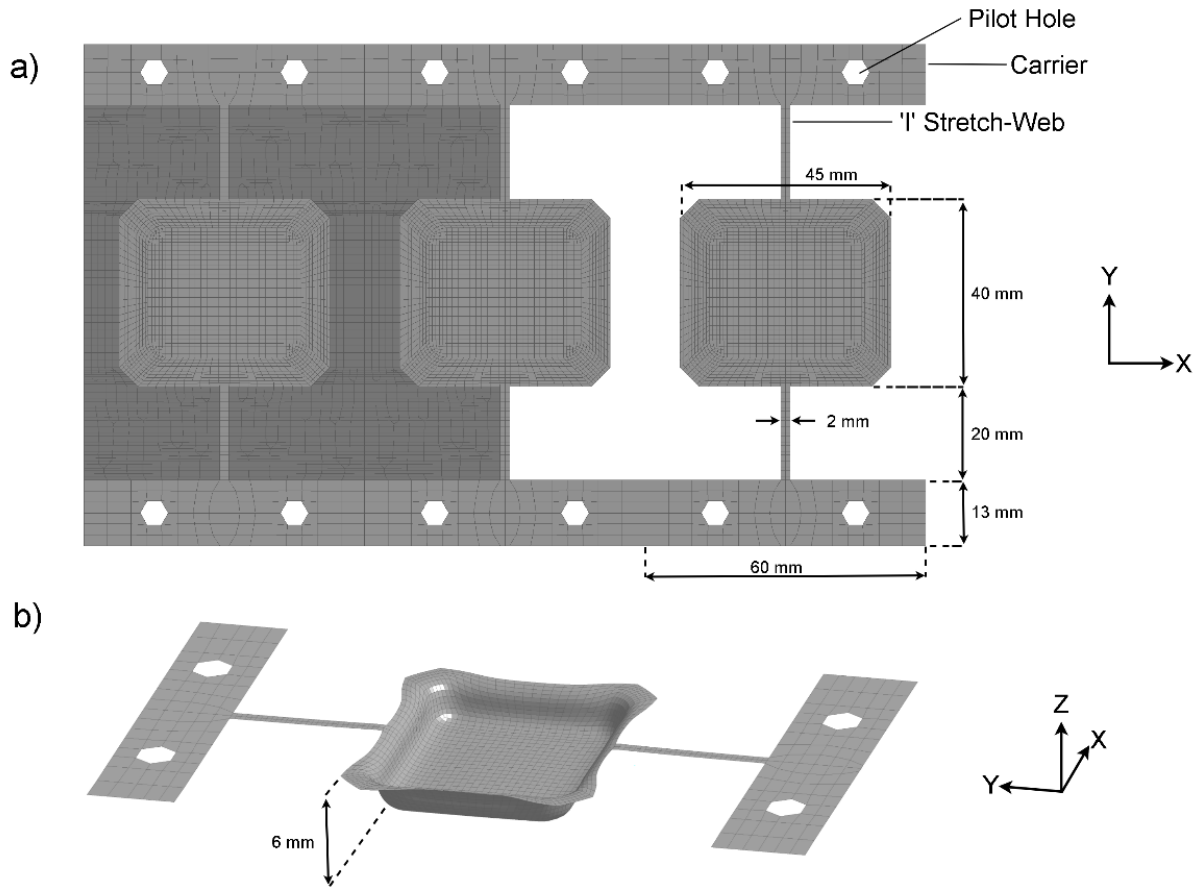

Figure 26: Strip layout for the 'Detailed’ model which features 'I' stretch-webs and two rows.

4.2.2 Boundary Conditions

The ‘Detailed’ model adopts boundary conditions which reflect a progressive die operation seen in industry, utilizing passive spring-operated strip lifters and a spring-actuated binder (Figure 27). Later iterations of the FE model use variable trajectories for the lifter and feeder instead, such as in the ‘Final’ model introduced in Section 4.6.

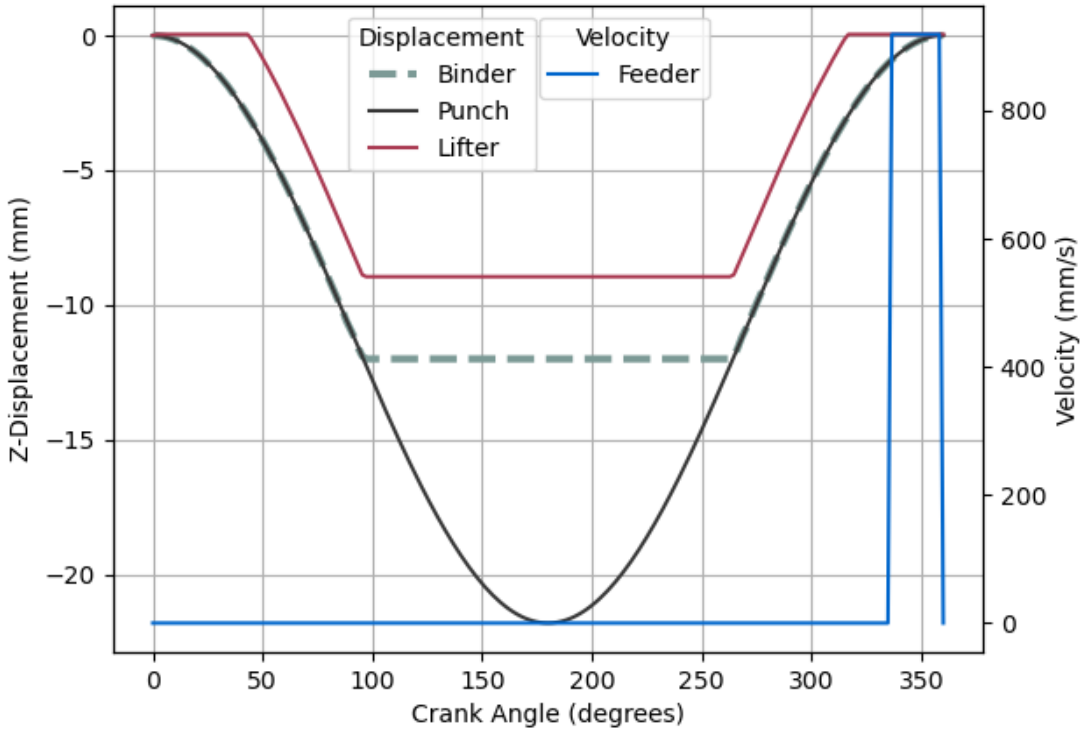

Figure 27: Tooling and strip lifter/feeder motion prescribed in the 'Detailed' model.

4.2.2.1 Punch, Locating Pins, & Binder

The 'Detailed' model includes a cropping and forming operation. Each of the punches follow an identical sinusoidal displacement by utilizing the '*BOUNDARY_PRESCRIBED_MOTION_RIGID' keyword. The travelling distance of the punch varied between FE models and is mentioned later when relevant to specific models.

The locating pins (Figure 25) followed the same motion and use the same boundary condition as the punches as they are also fixed to the upper tooling. The only difference is that the locating pins have a vertical offset from the punches which allow for the locating pins to locate the strip prior to the binder and punch contacting the strip.

The binder (Figure 28) follows a similar trajectory as the locating pins and punch, but, utilizes a spring to apply pressure to the strip during the forming process in a manner which reflects the physical process. The binder follows the same tooling path as the punch until it clamps the strip against the die-

face which restricts the binder from lowering further. As the upper tooling continues to lower, the binder clamps the strip and the binder spring compresses, applying increasing pressure to the strip. The applied pressure is maintained until the upper tooling re-opens, allowing the binder spring to relax and retract the binder from the strip. To mimic this physical process in a FE model, a linear elastic spring is directly attached to the binder using the '*ELEMENT_DISCRETE' and '*MAT_SPRING_ELASTIC' keywords. The binder spring uses a stiffness of 100 N/mm and is pre-loaded to 6800 N to apply 2.5 N/mm^2 of pressure on contact with the strip. To move the binder and spring, the upper node of the spring (Figure 28) has the same prescribed motion as the punch and locating pins which allows for the spring to compress once the binder clamps the strip against the die. To prevent vertical (Z-axis in Figure 28) oscillation of the binder while it is not in contact with the strip, a vertical constraint is applied using the '*CONSTRAINED_RIGID_BODY_STOPPERS' keyword. The vertical constraint uses the trajectory of the punch and locating pins as a lower bound for the minimum displacement of the binder. Finally, both nodes of the binder spring are restricted to a single degree of freedom to permit motion exclusively along the vertical axis.

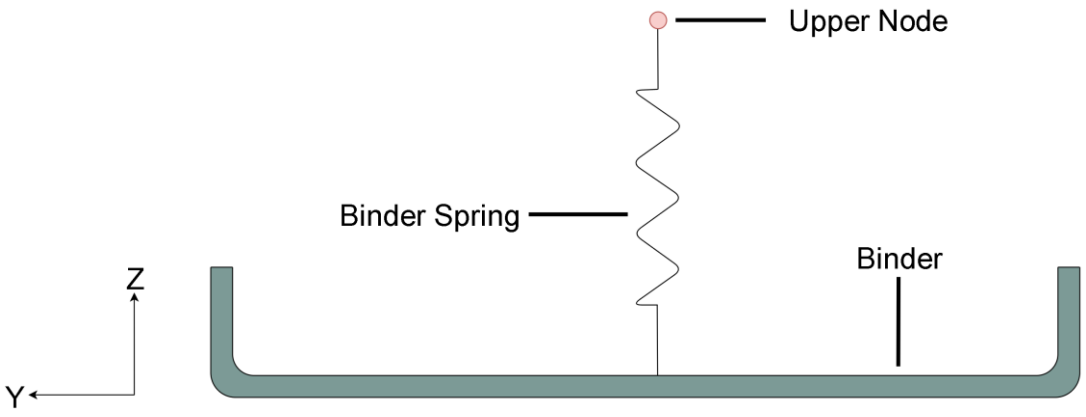

Figure 28: Standard implementation of binder using a passive spring.

4.2.2.2 Strip Lifters

While a goal of this research is to identify inputs to an active servo-driven strip lifter system which can assist in maintaining high stroke rates, in the 'Detailed' model (Figure 29), the strip lifters are passively activated by springs resembling lifters used in industrial applications. The implementation of the passive strip lifters is similar to the binder with the exception that the strip lifters do not have any motion directly applied in the form of a boundary condition. The strip lifters rest on a spring, again using the '*ELEMENT_DISCRETE' and '*MAT_SPRING_ELASTIC' keywords. Each of the fourteen lifter

springs have a stiffness of 1 N/mm and are pre-loaded to 54 N to commutatively support twice the weight of the strip (766 g). The upper node of the spring is attached to the strip lifters with motion exclusively along the vertical axis (*Z*-axis) and the lower node is fixed in place with zero degrees of freedom. Boundary conditions are applied to the lifters which restrict displacement between the die face and the feed height (Figure 29) using the '*CONSTRAINED_RIGID_BODY_STOPPERS' keyword. The lifters and strip are pressed downward by the binder while the press is closing (Figure 30).

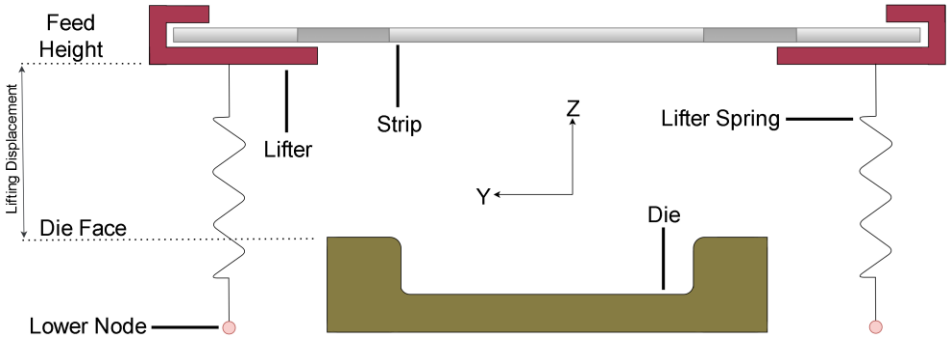

Figure 29: Passive spring-actuated lifters.

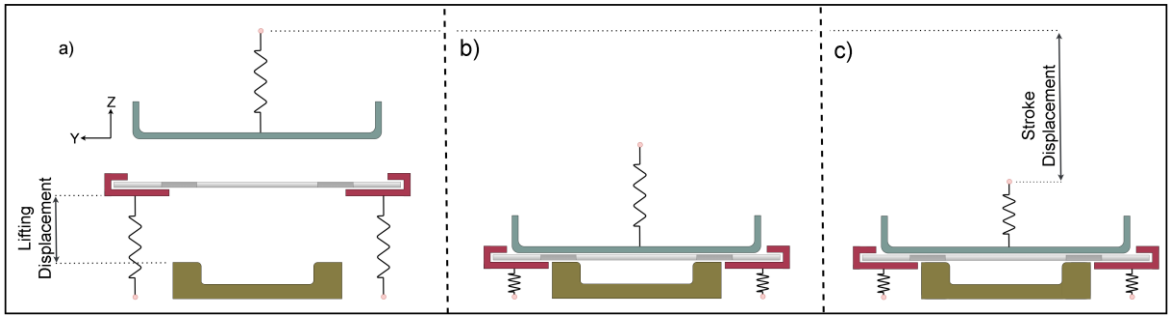

Figure 30: Illustration of the lifter and binder at different stages of the stroke where (a) the press is fully open, (b) the lifters reach their lower position, and (c) the press is fully closed.

4.2.2.3 Strip Feeder

All FE models used the same implementation for the strip feeder. The '*BOUNDARY_PRESCRIBED_MOTION_SET_BOX' keyword was used to apply a translational trajectory along the feeding axis (X-axis) to a set of nodes which are contained within a defined box. (Figure 31) Once a given node is outside the box, the boundary condition no longer applies to it. Nodes

contained inside the box have motion restricted in all degrees of freedom with the exception of the feeding axis during strip progression.

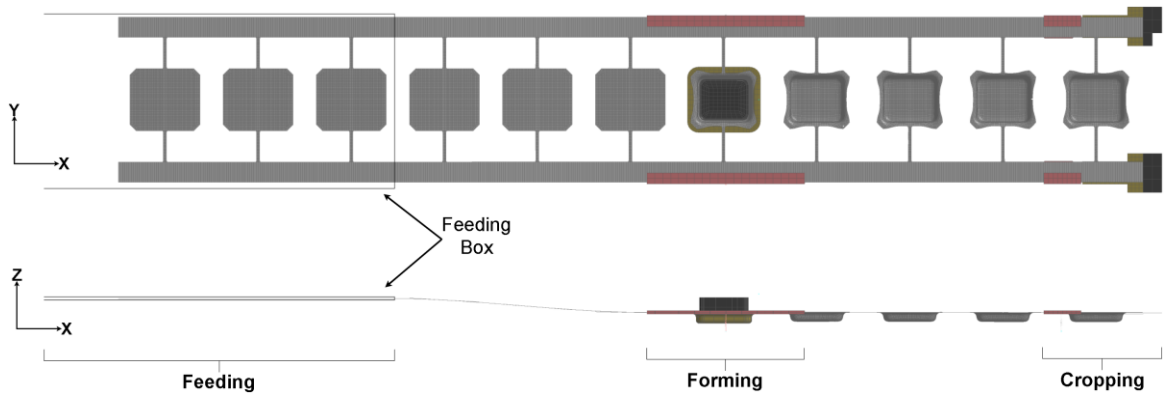

Figure 31: Top and side view of FE model with a feeding box on at beginning of tooling (left side).

The trajectory for the strip feeder was generated by the '*DEFINE_CURVE_SMOOTH' keyword (Figure 32) in which the area under the curve is the distance between tooling operations. The strip feeding begins once there is a 1 mm clearance between the strip and binder and ends when the tooling reaches the top-dead-center position. To reduce the jerk during strip feeding, 5 % of the total feeding time is used for the both the acceleration and deceleration of the strip, denoted as T_{rise} in Figure 32.

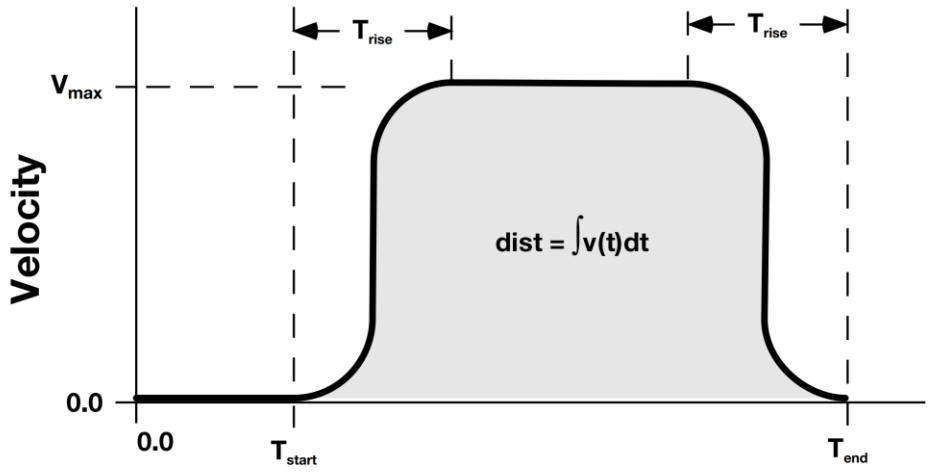

Figure 32: Variables used to definite velocity profile for the strip feeder [125].

4.2.3 Blanking & Cropping

To simplify the blanking and cropping process, the blanking and cropping of the sheet metal strip was not directly modelled as a shearing process. Instead, the trimmed regions of the blank are simply removed from the simulation when the punches contact the strip, thereby mimicking the blanking and cropping process using the '*DEFINE_CONSTRUCTION_STAGES' keyword. Contact between the blanking and cropping punches was not considered and their incorporation within the model are largely for visualization purposes in model post-processing.

4.2.4 Material Properties

To model the sheet metal strip a simple von Mises plasticity model with isotropic hardening *(*MAT_PIECEWISE_LINEAR_PLASTICITY*) was adopted as a computationally efficient material model. Material data for a 1 mm thick sheet of AA5182 was used for the material model. While both steel and aluminum were of interest in the research project, aluminum was used throughout the FE models to account for the smaller time-step, a result of the Courant Criterion, when developing a computationally efficient FE model [126]. The material properties for the AA5182 include a yield strength of 121.4 MPa (Figure 33), a UTS of 364 MPa, Young's modulus of 67.9 GPA, Poisson ratio of 0.34 and density of 2.4 g/cm^3.

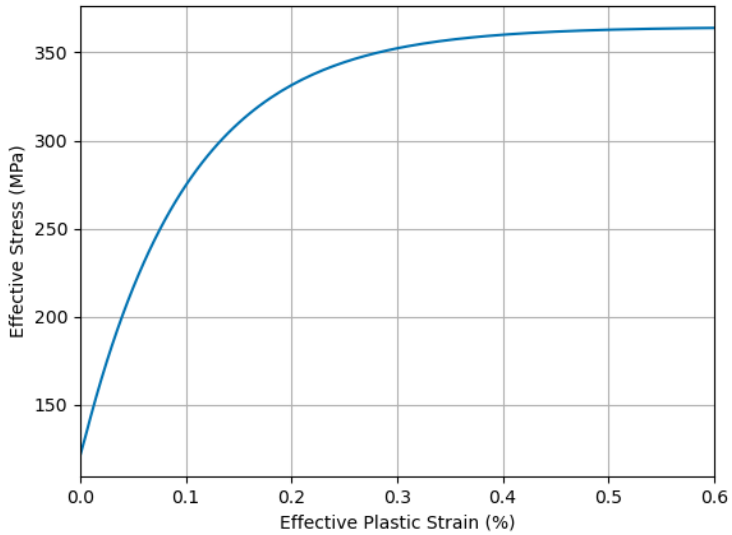

Figure 33: Stress-strain curve for 1 mm thick AA5182.

The tooling components (*i.e.* strip lifters, die, binder, and punches) were modeled as rigid bodies with steel material properties using the '*MAT_RIGID' keyword. The material parameters include a Young's modulus of 207 GPa, Poisson ratio of 0.3 and density of 7.85 g/cm^3. Components with vertical translation have constraints in all degrees of freedom with the exception of the vertical axis (Z-axis). The die is fixed throughout the FE simulations and therefore is constrained in all degrees of freedom.

4.2.5 Component Modelling

All components in the 'Detailed' model used 1 mm thick, quadrilateral shell elements. Fully-integrated (*Type-16*) shell elements for the sheet metal strip to avoid hour-glassing. Belytschko-Tsay shell elements (*Type-2*) were considered in some iterations of the FE simulation model, but, were quickly eliminated as a potential candidate due to observed excessive hour glassing behavior. Seven through-thickness integration points were used through-out the sheet metal strip. Rigid shell elements were used for the tooling components.

Contact was defined for each interface between the strip and the tooling, for a total of five contact interfaces, which employed a penalty function-based contact algorithm. A thickness offset was applied to the strip to account for the sheet thickness. A constant friction coefficient of 0.05 was prescribed with the viscous damping coefficient set to 20% of critical damping.

4.2.6 Outcome

The 'Detailed' model included just under 240,000 elements, 230,000 of which are deformable elements used for the strip. The 'Detailed' model was never run for any period longer than a few minutes due to the massive run-times needed to complete the simulation. Even while using 10x mass scaling, the estimated run-time for the simulation to complete is about 650 hours for 10 strokes at 60 strokes per minute (SPM), or, 65 hours per stroke using LS-Dyna SMP with 8 cores. To put this into perspective, to create a dataset with 100 samples in which each sample is generated from a single stroke of a simulation, it would take about 9 months of continuous computation.

Fundamentally, the 'Detailed' model was judged too computationally expensive to be used to generate data that could be used for training a machine learning model. Several factors contributed to the long run-times. First, the number of deformable elements was one of the largest contributors and was a result of the strip layout using two rows of work-pieces, as well as the size of the tooling. The length of the tooling combined with the number of strokes attempted to be modelled resulted in a large

number of individual work-pieces. If fewer strokes were being modelled, fewer work-pieces would be required, but, at minimum 21 work-pieces are required to simulate a single stroke. In addition, there is a large number of relatively small elements, the size of which is relatively small in order to maintain simulation accuracy which limits the degree to which the mesh can be coarsened. Since the simulation is using explicit time-integration, the size of the elements limit the maximum time-step which further increases the run-time of the simulation. While the 'Detailed' model was never run for even a single stroke, it provided a valuable reference for the decisions required to refine the FE model in subsequent iterations. The subsequent iterations of the of FE model, introduced in Sections 4.3, 4.5 and 4.6, present a series of attempts to reduce the computational cost of the simulation, including the 'Final' model adopted to generate the data for the machine learning models.

4.3 Early Model Optimization Attempts

Following the 'Detailed' model, a concerted effort was made to further reduce the computational expense of the FE simulations. While neither of the two early modelling attempts, termed the 'Implicit' and 'Beam Element' models, were capable of producing useable data to study the dynamic response, they proved useful in the development of the computationally-effective models introduced in Section 4.5. The 'Implicit' model sought to take advantage of the unconditional stability of implicit time-integration, in which larger time-steps could potentially be used to reduce the computational expense of the simulation. The 'Beam Element' model sought to reduce the high computational cost of using deformable shell elements by replacing them with beam elements along the strip carrier and stretch-webs. Nether modelling approach proved numerically tractable, so both were abandoned. The details of both early model optimization attempts can be found in Appendix A

4.4 Measuring Process Outcomes

An important consideration when developing the FE models is the required output(s) from the FE model to be used for training of the ML model. As outlined in Section 4.4, the goal of the ML model is to use the process inputs (*i.e.* SPM, feeder trajectory and lifter trajectory) to predict the outcome of the process. This section defines which predictions from the FE models were adopted to characterize the process outcome of a progressive die process.

A desirable outcome of all progressive die processes is to produce parts consistently within tolerance. The outcome can be further improved if the process operates at a high production rate without affecting

the part quality. In other words, when optimizing a progressive die process one seeks the highest stroke rate at which parts are produced consistently within tolerance and without process failures such as collisions.

The forming and shearing operations are the primary influences controlling the tolerance of a produced part. As discussed in Section 2.3, recent research has applied closed-loop control to adjust binder pressure and punch displacement to account for varying forming parameters such as material properties or lubrication. However, research to-date typically assumes correct placement of the work-piece in the forming operation which does not necessarily reflect industrial implementation. Rapid transfer of the strip can induce excessive oscillations which affect the placement of the work-piece on the forming or shearing die. Regardless of the adaptability of the forming operation, a produced part will not achieve nominal geometry if its placement in the tooling is skewed. Thus, excessive dynamic oscillations that lead to poor part placement on the forming die will result in poor part quality.

In industry practice, variations in the forming process will certainly occur, while in a virtual world, such as FEA, the variations are typically neglected. Variations in the material properties or friction can be introduced, but, those particular complexities were not considered in the current work. Furthermore, the punch velocity can influence the strain rate of the material, but, rate hardening effects were not considered in the implementation of the material properties. Therefore, in the current FE simulations, the primary variable that influences the geometric accuracy of the formed part is the placement of the work-piece.

As described in Section 3.1, the forming operations in the demonstrator tool were selected to be relatively simple and ‘easy to form’. The intent in selecting such a simple geometry is to not limit the process by the forming operation and, instead, put focus on the limitations of the strip transfer. While there is a limit to the speed at which the cups considered in this work can be formed, this limitation was not considered in order to focus on the limitations of the strip transfer.

Another consideration is the potential for collisions during strip transfer. Excessive oscillation of the strip can lead to collisions between the strip and tooling which results in an immediate failure of the process. Even if the blank manages to be positioned correctly after a collision with the tooling, the system is potentially a small perturbation away from a larger collision. Furthermore, a strip may become misaligned after strip transfer which prevents the locating pins from positioning the pilot holes on the

strip. A strip misalignment may also lead to undesirable tooling collision, damaging the tooling and/or the strip.

Other predictions are of potential interest in this work, but do not describe the outcome of the process. In particular, directly measuring the motion of each work-piece may elucidate how the process inputs affect the response of the strip. However, in the current work, when considering the process outcome, the strip dynamics only matter in the context of the placement of the work-piece and whether there is a collision or strip misalignment. Large oscillations naturally lead to an increased chance of a poor outcome of the process, but defining what is a permissible magnitude or frequency of oscillations can only be objectively defined in the context of the process outcome.

Ultimately, under identical forming conditions, the tolerance of the formed part is a function of its placement on the die. Furthermore, the placement of the work-piece is contingent on the avoidance of collisions and strip misalignments. Therefore in this work, the process outcome is measured by the placement of the work-piece on the die at the forming operation contingent on whether there was a collision or strip misalignment.

The last consideration when measuring the process outcome is whether the system is transient or is approximately steady-state. To achieve consistent part quality, it is necessary for the process to achieve an approximate steady-state behavior. In general, steady-state behavior refers to the behavior of a system being constant with respect to time. The 'work-piece placement' may be observed to reach steady-state, but, is ultimately dependent on the dynamic response of the strip. A more direct means of evaluating the steady-state behavior of a progressive die operation is by comparing the time-history of multiple work-pieces as they move through a section of the tooling. Once a progressive die operation reaches an approximate steady state, it would be expected that the motion of any given work-piece would follow nearly identical motion as it progresses through a section of tooling. The difference in trajectories of work-pieces can be measured by the mean-squared error between a reference work-piece and one that is assumed to have reached steady-state as it passes through the sampled zone (Figure 34). The MSE can be calculated by

$$MSE^i = \frac{1}{n-1}\sum_{t=1}^{n}\left(x_t^{ij} - \bar{x}_t^{j}\right)^2 \tag{14}$$

where $\bar{x}_t^j$ is the steady-state response of the reference work-piece at time, t, measured in the degree of freedom, j, and x_t^{ij} is the measured response of the i^{th} work-pieces to pass through the sampled zone at time, t, measured in the degree of freedom, j. When the MSE of work piece, i, and $i+1$ does not change, then the system has reached an approximate steady-state since the transient behavior is relatively constant between each consecutive work-piece. This steady-state characterization of the predicted response will be considered in the 'Deformable-Rigid Switching' model (Section 4.5.2) to identify how many strokes it takes to reach steady-state.

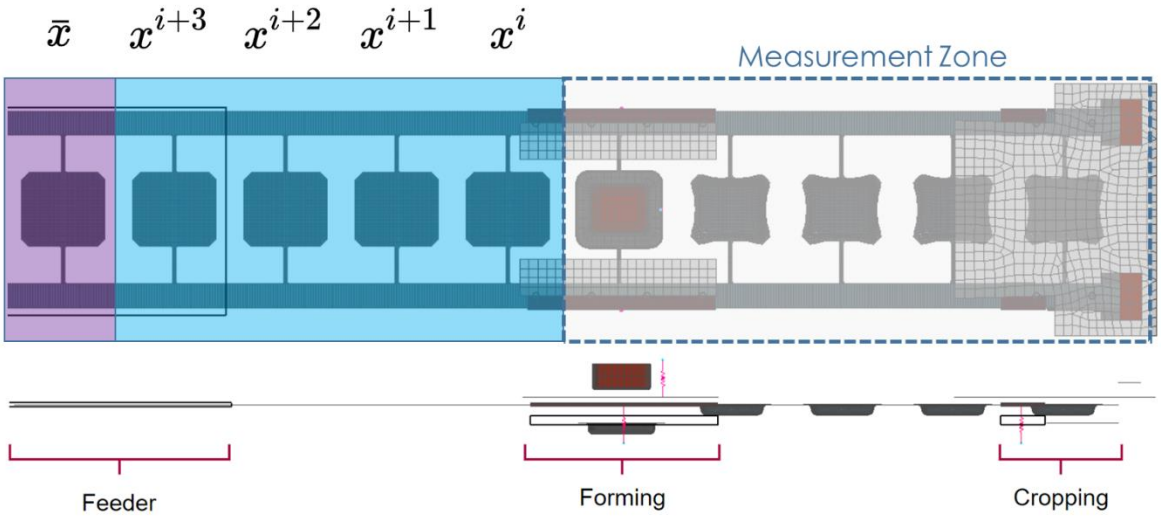

Figure 34: Example of how the MSE is calculated where the last work-piece (purple) is taken as the steady-state reference and compared against the response of the prior work-pieces (blue).

Once the system reaches an approximate steady-state, the outputs of the process will create a distribution of outcomes since the remaining transient response due to perturbations in the system will prevent identical outcomes from every stroke. Therefore, when analyzing input permutations to the progressive die process, the distribution of 'work-piece placement' from the approximate steady-state process should be compared. However, measuring the distribution of placements at steady-state would require multiple strokes to be simulated which is computationally expensive and inauspicious for developing a dataset for training a machine learning model. The practicality of measuring the steady-state behavior of the progressive die process will be explored in the 'Deformable-Rigid Switching' model (Section 4.5.2).

4.5 Computationally-Effective Models

The following two models, described in Sections 4.5.1 and 4.5.2 take an alternative approach to the 'Detailed' model by reducing the complexity of the progressive die process being modelled. Instead of modelling the entire progressive die operation, the process was first reduced to focus on the forming and cropping operations. The goal of this approach was to first develop a computationally-effective simplified model, after which complexity was reintroduced through a series of model iterations. The computationally-effective models presented in this section use the same numerical implementation as the 'Detailed' model.

4.5.1 Single Work-Piece Model

4.5.1.1 Overview

In the case of the 'Single Work-Piece' model (Figure 35), the strip includes only a single work-piece that is initially positioned two stages prior to the forming operation which progresses through the forming and cropping operations. This model utilizes the tooling geometry taken directly from a single row of the 'Detailed' model. Three different strip layouts were considered using either an 'I' or 'S' stretch-web and one or two strip carriers (Figure 36). The geometry of the strip layout is a single row of the strip in the 'Detailed' model. By reducing the strip to a single work-piece, significantly fewer deformable elements are present and therefore a large reduction in the computational expense can be realized.

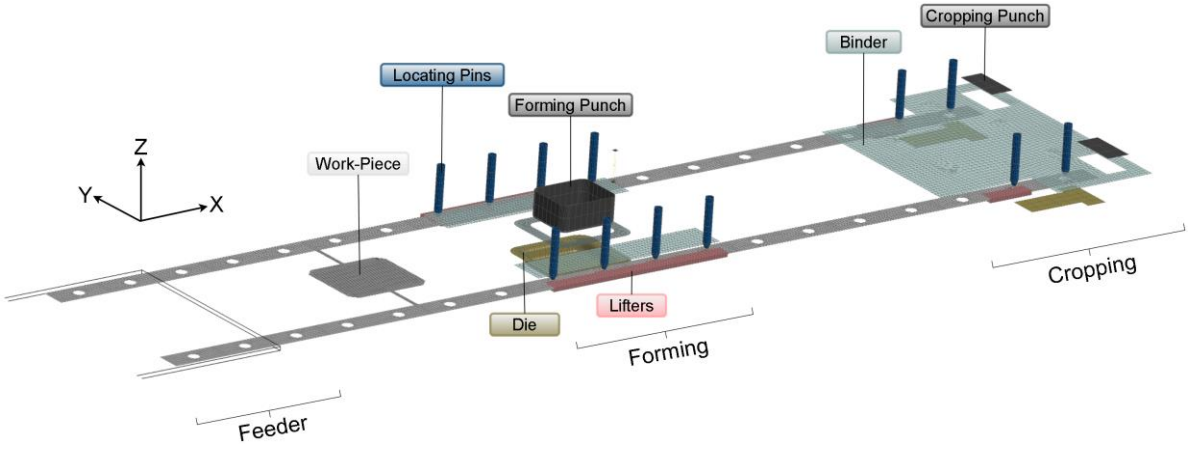

Figure 35: Overview of 'Single Work-Piece' model.

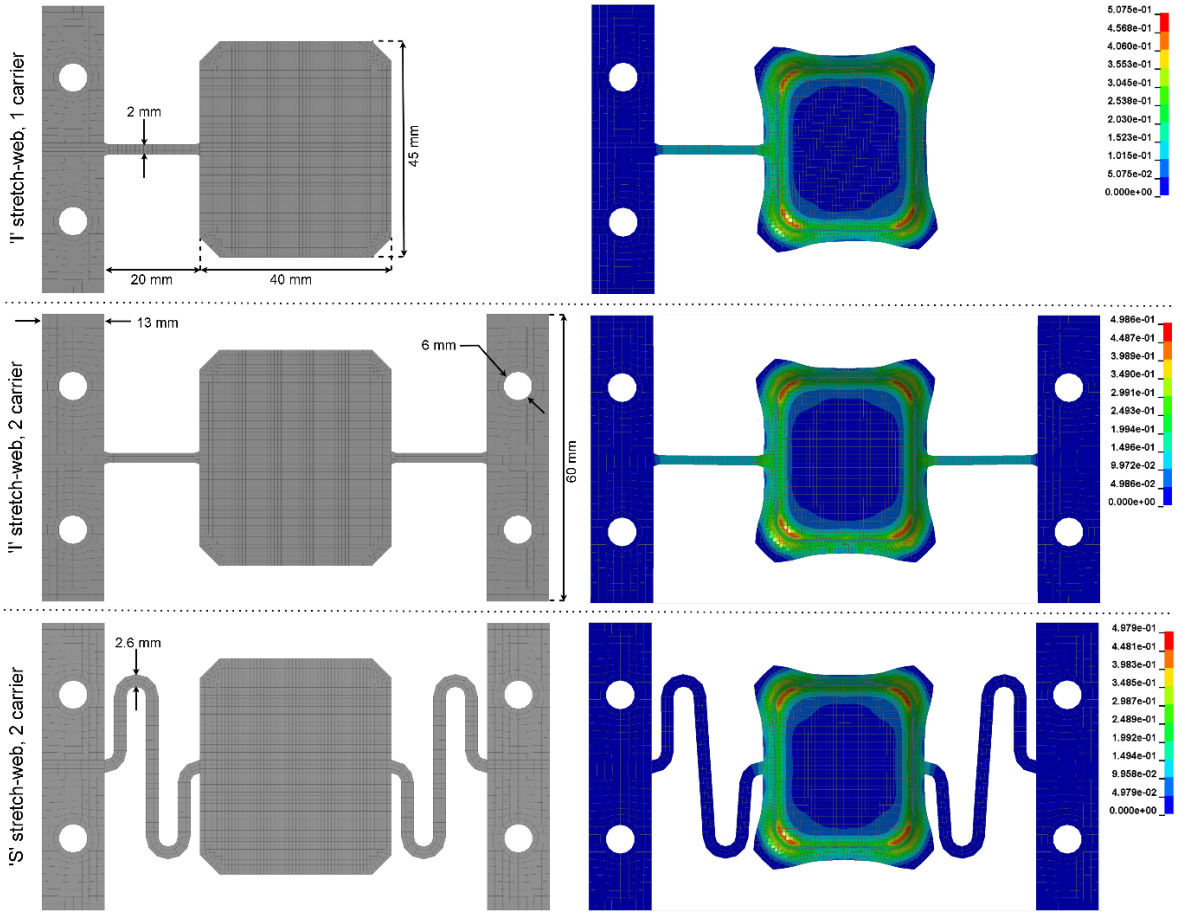

Figure 36: Strip layouts used in 'Single Work-Piece' model with (left) blanked work-pieces and (right) effective plastic strain of formed parts with a draw depth of 6 mm.

Reducing the number of work-pieces presented the opportunity to enhance the discretization of the strip mesh while still maintaining a net decrease in deformable elements. The pilot holes, stretch-web and blank all were re-meshed using a finer mesh relative to the 'Detailed' model. Increasing the density of elements on the work-piece introduced several smaller elements on the work-piece which reduced the critical time-step. A mild degree of mass scaling was applied to increase the critical time-step of the smallest elements by directly setting the limiting time step to 1.25×10^{-7} seconds. The mass of the work-piece is 4.45 grams and was increased by 1 % pre-forming and 2.9 % post-forming. A consequence of removing work-pieces from the strip is the increased propensity for the strip carrier to derail off of the strip lifters during the strip progression. To mitigate this potential issue, boundary

conditions were applied to the nodes along the strip carrier where the missing stretch-webs would be located. The '*BOUNDARY_SPC_SET' keyword was used to restrict the motion of the nodes of the missing stretch-webs in the lateral direction (Y-axis). With exception of these additional nodal constraints, the remainder of the FE set-up is identical to that described in Section 4.24.1.

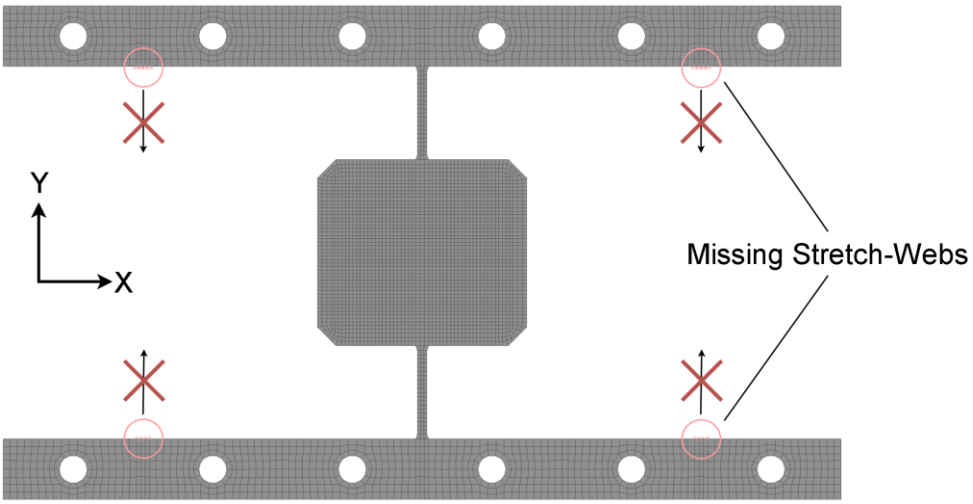

Figure 37: Work-piece in the 'Single Work-Piece' model with missing stretch-web boundary conditions.

4.5.1.2 Results

A total of six simulations were completed by running the three strip layouts at 60 SPM and 120 SPM until the work-piece was cropped at the end of tooling for a total of seven strokes. To compare the dynamic response of each simulation, the Z-displacement and Y-rotation of the work-piece (Figure 38) are predicted and plotted (Figure 39). Referring to Figure 38, the Z-displacement is predicted based on the nodal displacement of the center node of the work-piece. Since the work-piece is deformed out-of-plane, the Z-displacement is adjusted during the forming operation to account for the depth of the formed cup. The Y-rotation of the work-piece is predicted according to the following equation

$$\theta = \sin^{-1}\left(\frac{n_c^Z - n_e^Z}{d_{ce}}\right) \tag{15}$$

where n_c^Z and n_e^Z are the Z-displacement of center node and edge node respectively, and d_{ce} is the direct distance between the two nodes and is assumed to be a constant 22.5 mm.

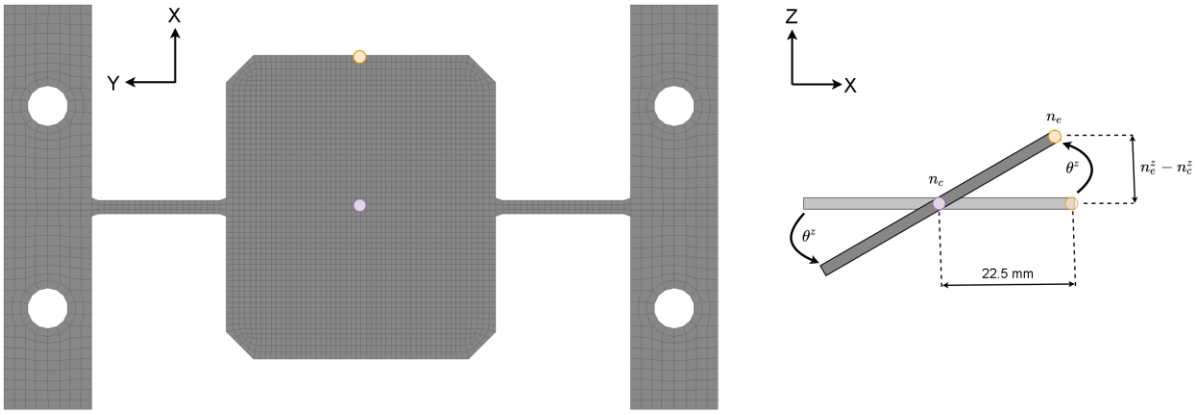

Figure 38: Nodes for predicting the displacement and rotation of the work-piece.

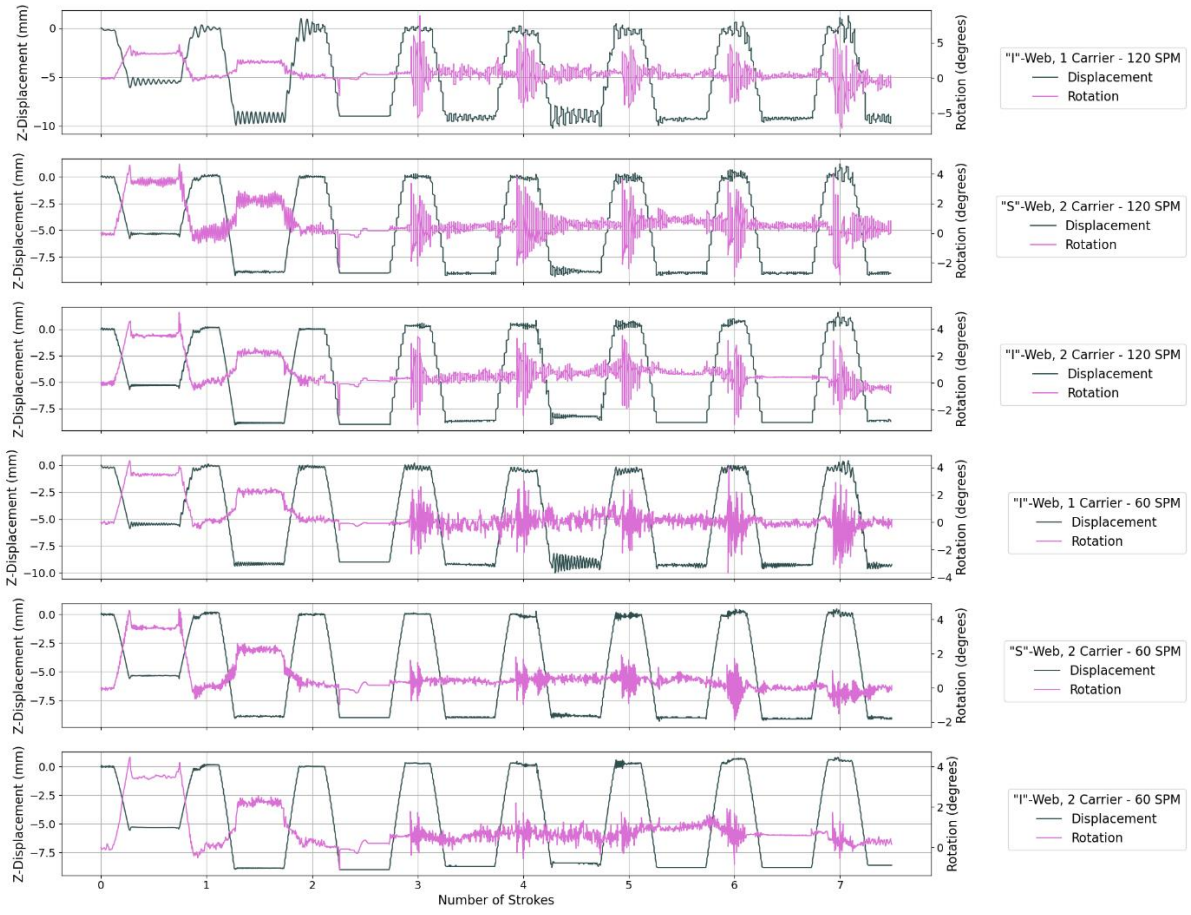

Figure 39: Displacement and rotation of work-pieces for the 'Single Work-Piece' model.

The displacements in Figure 39 are illustrated by the dark green lines which form a “square-wave” pattern. The Z-displacement of the work-piece follows the lifter trajectory quite closely, but, once the work-piece is lifted to the feeding height, the work-piece will oscillate up and down due to the sudden deceleration. The oscillations are noticed to a lesser degree when the lifters are in their lower position since the binder is clamping portions of the strip. It should be noted that for the first stroke, the Z-displacement has a higher valley since the work-piece is near the feeder which is fixed in height. After multiple strokes, the work-piece is further from the feeder and therefore has lower valleys. With respect to the Y-rotation of the work-piece around the stretch-web, similar behavior is observed between each simulation. After the third stroke, the work-piece is formed and the center of mass is moved out-of-plane, increasing its moment of inertia about the stretch-webs. This increased moment of inertia results in the larger oscillations after the third stroke of each simulation. Furthermore, the largest rotations are observed during the middle of the strip feeding, when the Z-displacement is highest. The acceleration for the strip progression is the primary driver of the rotation about the stretch-web since the center of mass is out-of-plane with respect to the stretch-webs.

Comparing the result of each simulation also illustrates how the stroke rate and strip layout influence the dynamic response. The simulations operated at higher stroke rates produce larger oscillations and rotations than the same strip layout operated at lower stroke rates. Furthermore, larger oscillations and rotations are observed when the strip layout is more compliant. For example, the ‘I’ stretch-web is more stiff then the ‘S’ web and therefore experiences a smaller dynamic response for both stroke-rates. As well, the single carrier configuration is more compliant than a two carrier strip layout and therefore experiences a larger dynamic response as compared to the two carrier strip layouts.

4.5.1.3 Outcome

While the ‘Single Work-Piece’ model was useful as a prototyping model to generate preliminary data, the simulations required considerable run times. Referring to Table 2, the simulations took between 22 to 63 hours to complete five strokes using LS-Dyna MPP with 32 cores. To properly model the progressive die process, additional work-pieces need to be incorporated in the model which will increase the computational cost of the model. Therefore, future iterations of the model focused on further optimizing the computational requirements.

Table 2: Completion time for the ‘Single Work-Piece’ model.

Model	SPM	Number of Carriers	Stretch-Web	Number of strokes	Simulation Time (seconds)	Number of Cores	Solver	Completion time (hours)
Single Work-Piece	60	1	I	5	5	28	MPP	45
		2	I	5	5	32	MPP	N/A
		2	S	5	5	32	MPP	63
	120	1	I	5	2.5	32	MPP	22
		2	I	5	2.5	32	MPP	N/A
		2	S	5	2.5	32	MPP	31

4.5.2 Deformable-Rigid Switching Model

4.5.2.1 Overview

The ‘Deformable-Rigid Switching’ model (Figure 40) combines the modelling techniques used in the ‘Single Work-Piece’ model with the addition of “rigid body switching” to enable simulation of the response of the strip for multiple strokes with multiple work-pieces. The reduced progressive die operation first introduced in the ‘Single Work-Piece’ was used and employs strip layouts with a single or double carrier configuration using ‘I’ stretch-webs, operating at 60, 120, or 180 SPM for a total of six simulations.

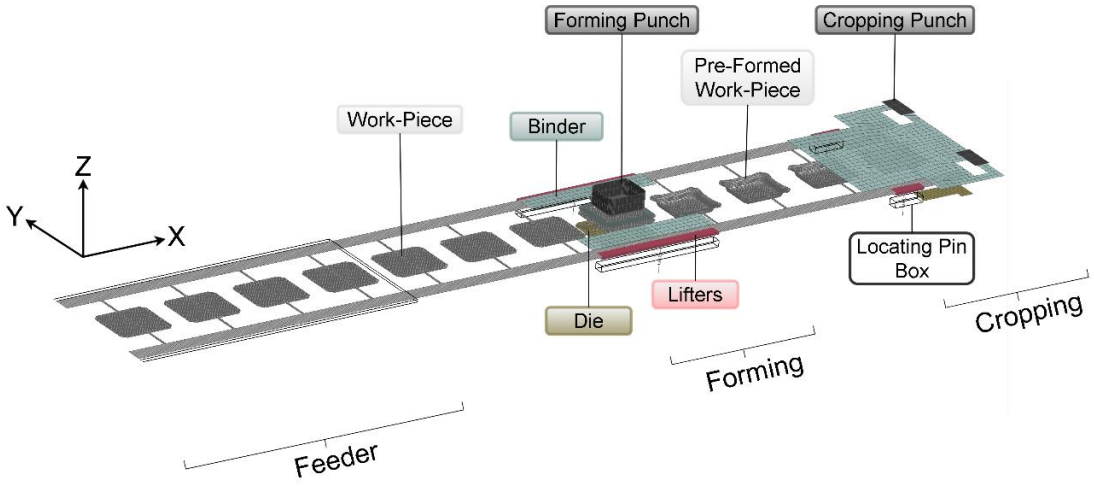

Figure 40: Overview of the ‘Deformable-Rigid Switching’ model.

The strip was initialized with work-pieces at every tooling station, including pre-formed work-pieces after the forming operation in order to help reduce start-up transients. The mesh for the strip layouts is very similar to the ‘Single Work-Piece’ model with the exception of the strip carrier (Figure 41). To reduce the computational cost as compared to the previous modelling attempts, the mesh of the strip

carrier was coarsened such that the carrier was a single element in width. The coarsened mesh of the strip carrier is inspired by the 'Beam Element' model (Appendix A) and the use of beam elements to model the strip carrier which accurately captured the bending of the strip carrier along its length. Deformable fully-integrated shell elements are used with 7 integration points for the entire strip.

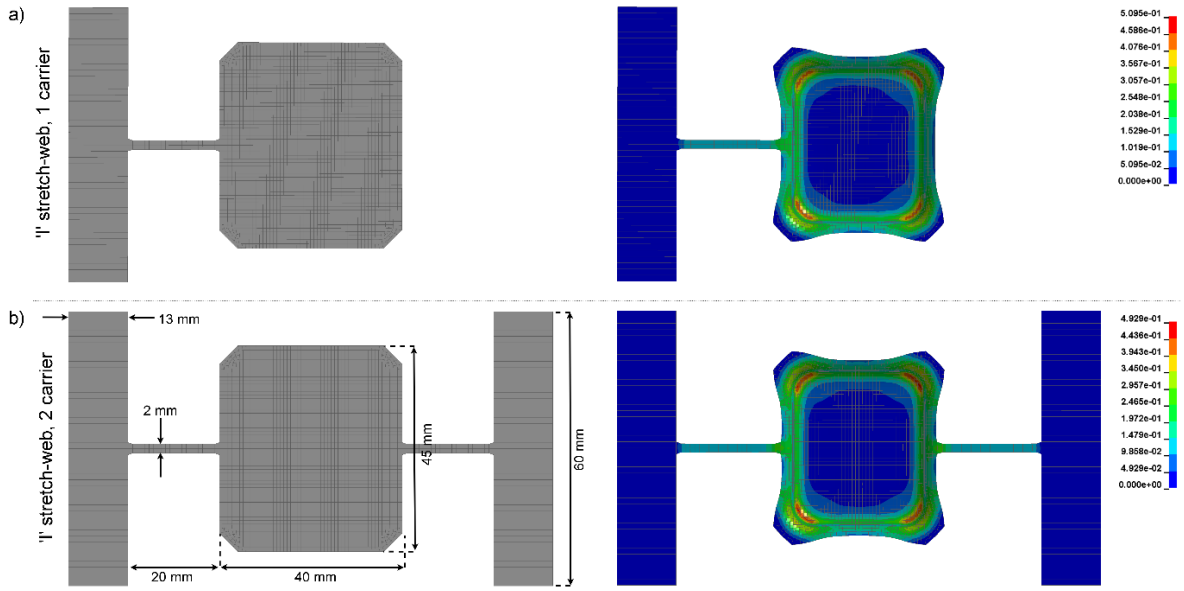

Figure 41: Comparing effective plastic strain of formed parts for a (a) single carrier and (b) dual carrier strip layout.

As a result of the coarsened mesh along the strip carrier, pilot holes for the locating pins were no longer included. By not including pilot holes, a contact definition between the locating pins and strip was removed, further reducing the computational cost of the model. However, a new boundary condition was required to locate the strip after strip progression and mimic the restraining effect of the pins (Figure 42). To locate the strip, the '*PRESCRIBED_MOTION_SET_BOX' keyword is used to restrict the lateral motion (Y-axis) of the nodes along the outer edge strip carrier. The boundary condition is applied only to the nodes which are positioned within a user-defined box at the forming and cropping operations and only active during the period in which the pins would be in contact with the strip

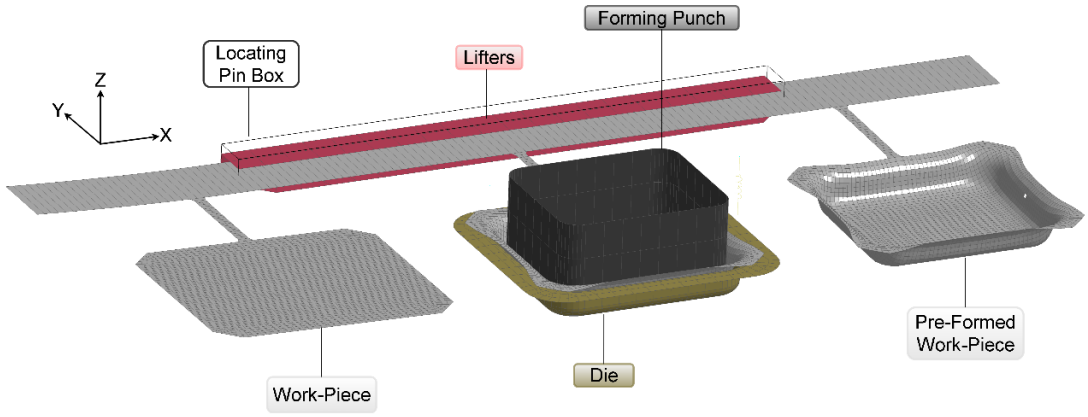

Figure 42: Boundary condition applied to nodes along outer edge of carrier to replace the locating pins.

4.5.2.2 Rigid Body Switching

To further reduce the computational expense of the FE simulation and enable consideration of multiple work-pieces, rigid body switching was used on the work-pieces to switch between rigid elements and deformable elements. In this approach, all of the work-pieces are normally treated as rigid bodies, except during the actual forming process at which point they are numerically switched to become deformable bodies. This approach greatly reduces the computational time and was considered justified since, in the 'Single Work-Piece' model, the deformable work-piece did not exhibit active deformation during strip transfer. Using this approach, all work-pieces begin the simulation as rigid bodies. Once a given work-piece has been transferred to the forming station, prior to lowering onto the die face, the elements of the work-piece are switched to deformable elements using the '*DEFORMABLE_TO_RIGID_AUTOMATIC' keyword. This switch occurs at 1 degree of crank angle prior to the binder clamping the strip. The elements are switched back to rigid elements at 1 degree of the crank angle after the binder releases the strip and the work-piece has been formed.

A key benefit in terms of computational expense lies in the smallest elements in the model being located in the work-pieces. These small elements control (reduce) the time step during the deformable phase, however, the time step can be increased significantly once the work-piece is switched back to rigid. As an additional measure, mild mass scaling during forming was applied by manually setting the limiting time-step to 1.463×10^{-7} seconds (this resulted in an increase of the part mass by 2.9%). Note that this mass scaling only affects the work-piece during the forming operation while it is supported by

the forming die and clamped by the binder, preventing significant dynamic effects. Mass scaling is disabled once the work-piece is switched back to rigid so that dynamic effects during transfer of the strip are not affected.

4.5.2.3 Damping

Early results from the 'Deformable-Rigid Switching' model successfully formed multiple work-pieces, but, non-physical behavior was observed along the stretch-webs which would spontaneously 'sag' while not in contact with any tooling components (Figure 43). The non-physical behavior was mitigated by introducing material damping to the strip.

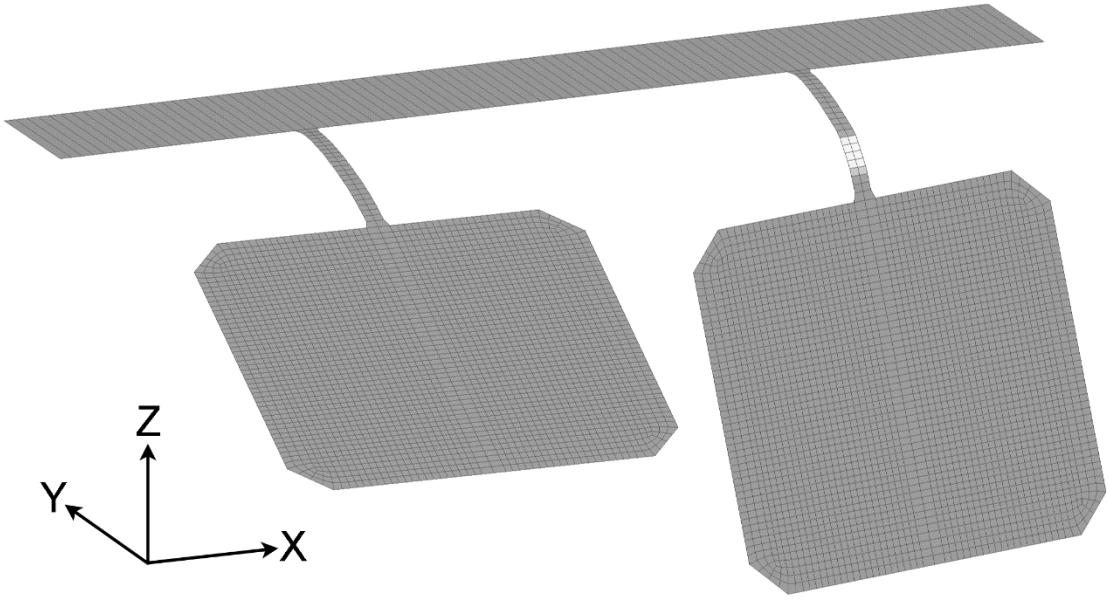

Figure 43: Observed 'sagging' of the work-piece while not in contact with tooling.

Early results from an impulse test conducted at TUM [127] on a half-symmetry of a work-piece stretch-web structure was used to characterize the damping of work-piece. The center of the work-piece was excited with an impulse and the velocity response was recorded with a laser vibrometer (Figure 44). A FE model of the experiment was created to calibrate the numerical material damping. The FE model used a half symmetry of the work-piece and applied a unit impulse to the center of the work-piece while restricting motion in the strip carrier. Multiple damping parameters were considered and the closest results were obtain using frequency independent damping with 1 % of critical damping applied to the entire strip over the range of 1 Hz to 1500 Hz which includes the first the first five natural

frequencies. Characterizing the damping behavior of the strip was initially thought to be out of the scope of this thesis, but, was found to be necessary to produce stable numerical results. Future work should take a more systematic approach to characterizing the damping behavior of the strip.

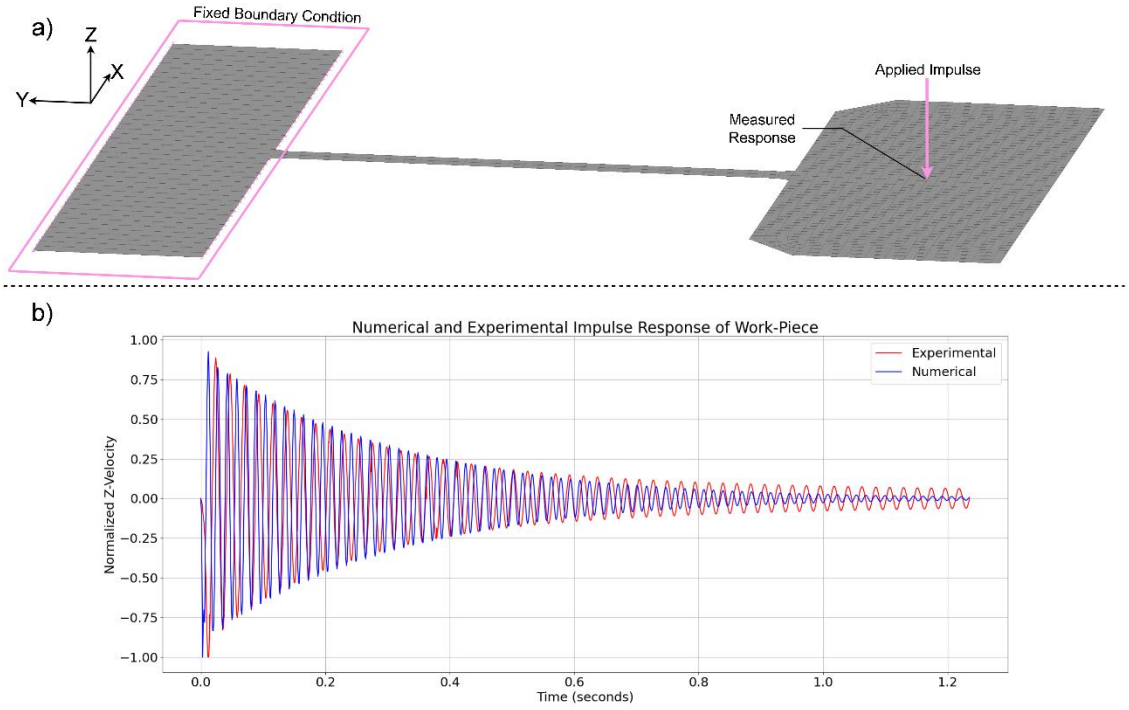

Figure 44: Overview of damping calibration where (a) the FE used to replicate the experimental impulse test, and, (b) the experimental and numerical results of the impulse test.

4.5.2.4 Results

Both strip layout configurations were tested at 60, 120, and, 180 SPM for five strokes creating a total of six simulations. Using LS-Dyna MPP with 20 cores, the simulations took between 63 and 105 hours to complete five strokes (Table 3). Similar to the 'Single Work-Piece' model, the rotations and displacements of the work-piece positioned immediately before the forming operation were predicted for the duration of the simulation. Instead of using the nodal displacements to calculate rotation and displacement, the rigid body outputs from the work-piece are used which directly correspond to the displacement and rotation of each work-piece. Three orientations are considered which capture the degrees of freedom which exhibit the largest response of the work-pieces.

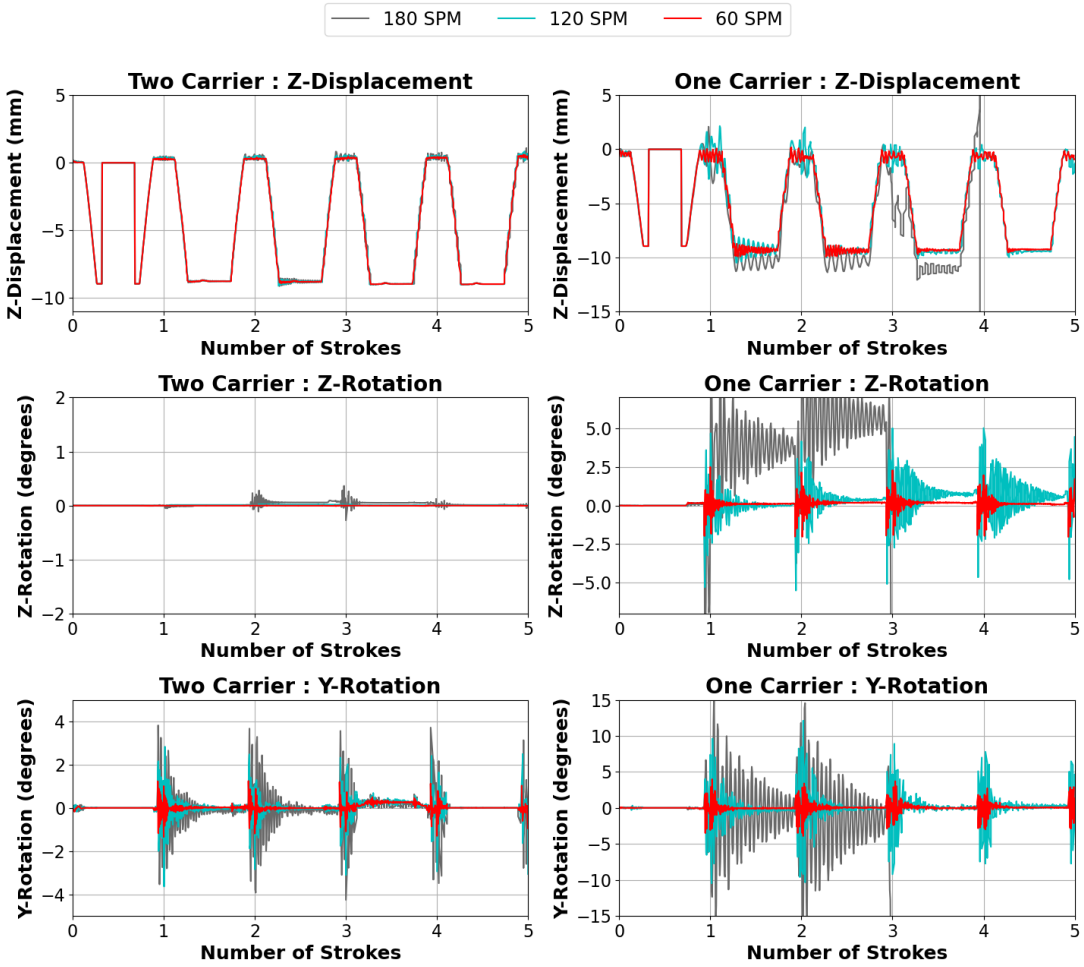

Figure 45: Rigid body predictions of a single work-piece positioned immediately before the forming operation for each simulation.

The predicted dynamic for each strip layout and stroke rate can be seen in Figure 45. In the case of the Z-displacement, the lifting and lowering of the strip is the primary influence for the dynamic response. Since the single carrier configuration is unsupported on one side and creates a cantilevering structure, it is more susceptible to exhibiting larger deflections than the two carrier configuration. With respect to the Z-rotation, the strip progression directly induces rotation about the Z-axis. The two carrier configuration has limited predicted rotation about the Z-axis since its motion is restricted by the stretch-web on either side of the work-piece. However, in the case of the single carrier configuration, a larger dynamic response is observed that increases with stroke rate. Last, the Y-rotation is the predicted rotation about the stretch-webs and is exhibited by both strip layouts. Rotation about the Y-axis is

induced by the center of mass of the work-piece being out-of-plane with respect to the stretch-webs which creates a moment during strip progression. Both strip layouts show the largest predicted response about the Y-axis with the single carrier configuration exhibiting a larger response than the two carrier configuration at any given stroke rate. In the case of the single carrier configuration at 180 SPM, the response became so volatile that collision with the binder during strip progression was predicted, producing disproportionate predictions which go out of the range of the plots in Figure 45. Similar to the results from the 'Singe Work-Piece' model (Figure 39), the predictions in Figure 45 show that that the stroke rate and strip layout effect the dynamic response. In all cases, the two carrier configuration exhibits a stiffer response with less motion as compared to the one carrier configuration. Furthermore, as the stroke rate increases, all strip layouts exhibit a larger predicted response for each degree of freedom observed.

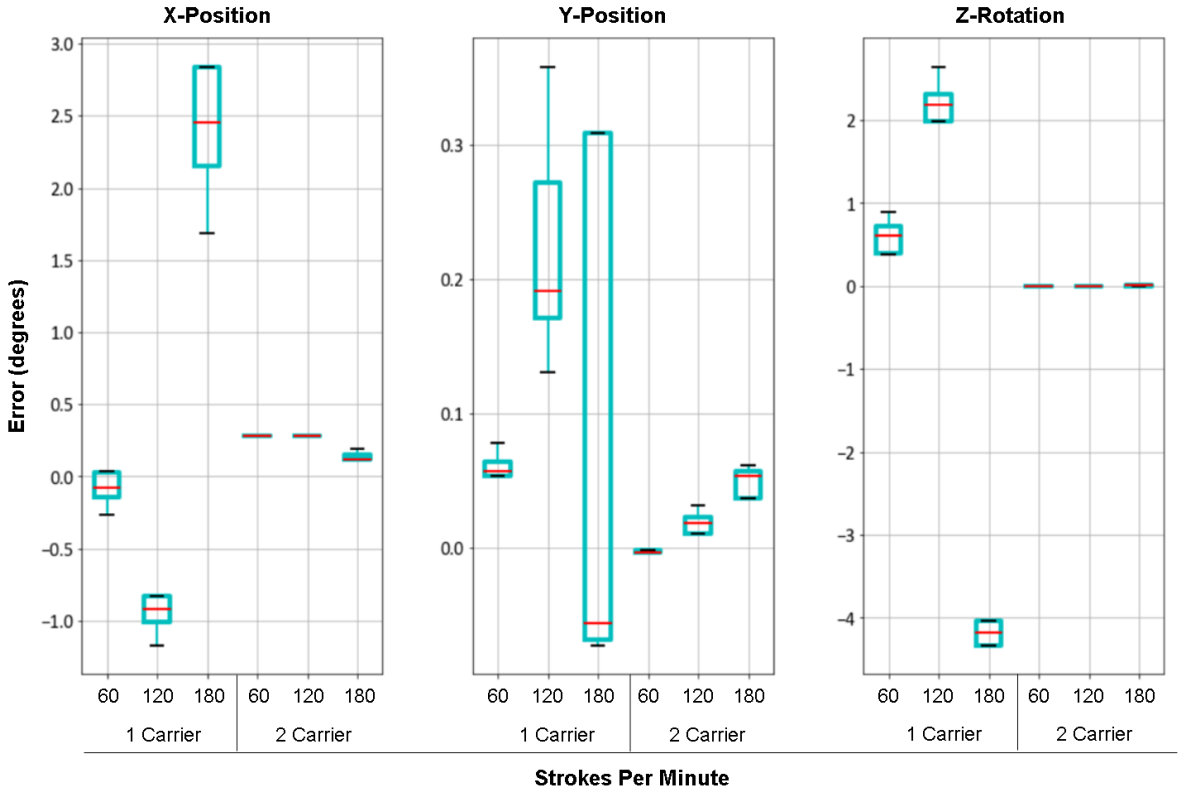

Figure 46: Boxplots of the work-piece placement positional error for each of the six simulations predicted for each degree of freedom for the blank placement.

While direct assessment of the predicted dynamic response (Figure 45) serves to illustrate how the feeder and lifter speed impact the strip stability, it is also possible to predict the blank placement error as a means to illustrate how the strip layout and stroke rate influence in the predicted forming behavior (Figure 46). The blank placement has three degrees of freedom which are the X- and Y-position and the Z-rotation. The blank placement error is taken as the difference between the center of the work-piece (predicted once the binder clamps the strip) and the center of the forming die. The results for the positional error are consistent with the dynamic response for each strip configuration. The placement for the two carrier configurations experiences minimal variance and a mean closer to zero which corresponds to the limited predicted dynamic response as compared to the single carrier configuration. However, since locating pins are not included in the model, the position of the strip carrier shifts slightly in the X-direction prior to the binder clamping the strip. On the other hand, the single carrier configuration experiences larger variances and mean errors that increase with the stroke rate since the blank in the single carrier configuration is free to rotate about the Z-axis during the strip progression. The single carrier configuration is at risk of producing part quality issues at speed of 120 SPM. At 180 SPM, the single carrier the produced parts diverge significantly from the desired geometry and a collision leads to the strip carrier sliding off the lifters (Figure 47).

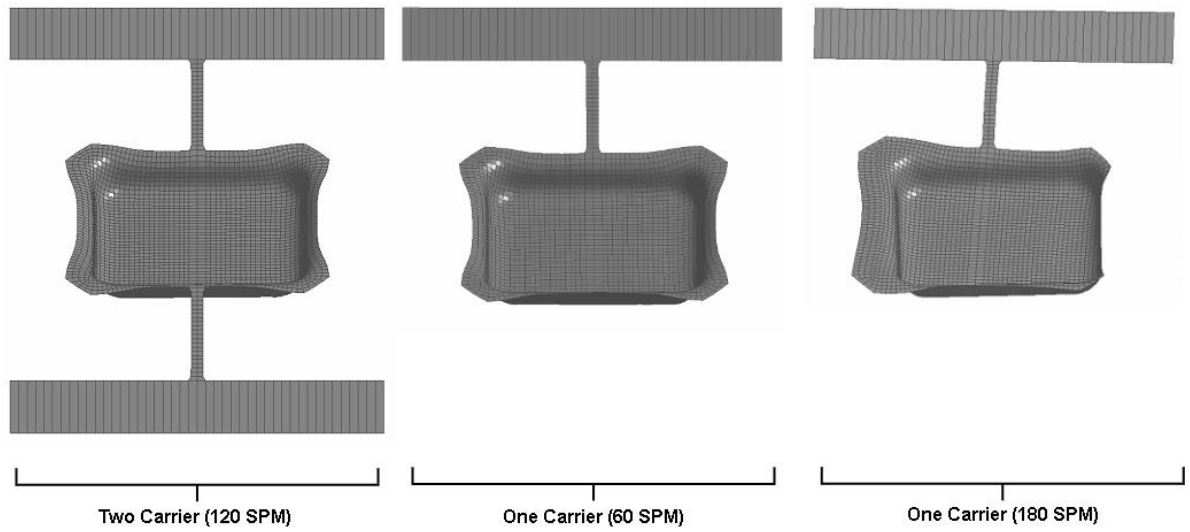

Figure 47: Formed cups for both strip layouts at different strokes. The two carrier configuration produces consistent geometry at all stroke rates. The single carrier configuration has minimal errors at 60 SPM and significant defects at 180 SPM.

4.5.2.5 Steady-State Analysis

An important consideration is whether the results obtained from the progressive die operation were sampled at steady-state or reflect a transient behavior. As described in Section 4.4 the steady-state behavior can be identified if the MSE is small and constant between the dynamic responses of multiple work-pieces sampled through the same section of tooling. To observe whether the progressive die operation is predicted to achieve steady-state behavior, the 'Deformable-Rigid Switching' model was used to simulate a strip layout with 'I' stretch-webs, a single carrier and a stroke rate of 120 SPM for 25 strokes. To evaluate whether steady-state behavior is reached, the mean variance of the predicted response is calculated between a reference part and each work-piece as it passes through a region of the tooling. More specifically, the last work-piece is taken as the steady-state reference response since it will be closest to resembling the steady-state response and is compared to prior work-pieces as they pass between the forming and cropping stations. The rigid body outputs of the work-pieces are used as the predictions for the dynamic response of the work-piece.

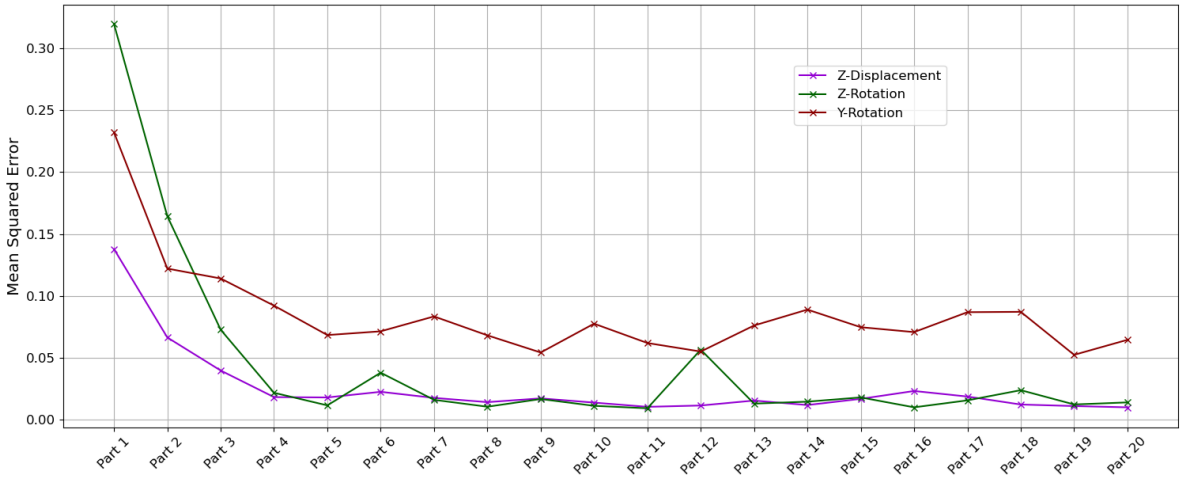

Figure 48: Predicted MSE between work-pieces for multiple degrees of freedom.

Figure 48 shows the predicted MSE for multiple work-pieces as they pass through the forming and cropping operations. After four to six work-pieces have been produced, the MSE of the predicted response plateaus, indicating that the differences between predicted responses are relatively consistent and minimal. In other words, the predicted displacement and rotation of each work-piece has limited variability between each consecutive work-piece and the process is approximately steady-state.

Further evidence that an approximately steady-state behavior is achieved can be observed by overlaying the predicted transient displacement and rotation for each work-piece as it passes through the forming and cropping operations. Figure 49 shows how the predicted Z-transient displacement histories of the work-pieces begin to converge to reach a steady-state behavior as the process continues. The last work-pieces to pass through the predicted section of the tooling (yellow lines) have nearly identical responses, indicating an approximate steady-state response.

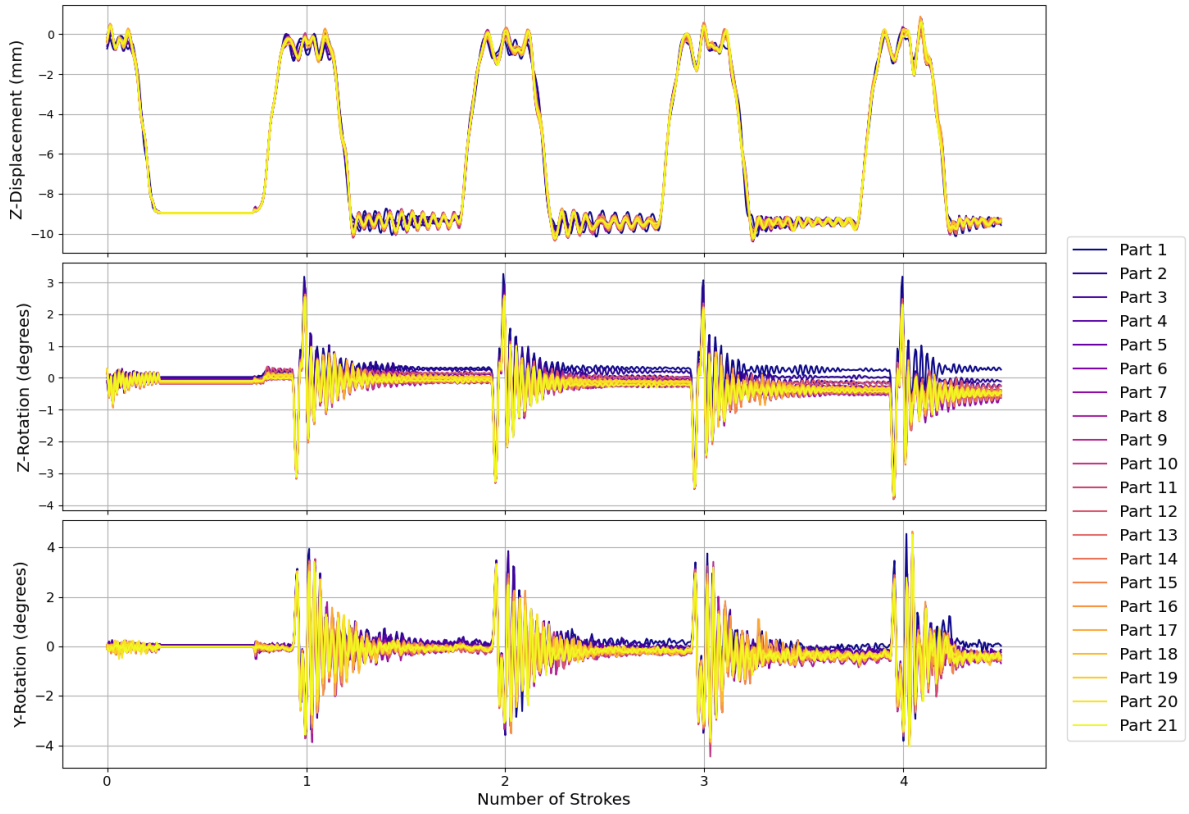

Figure 49: Predicted response of 21 work-pieces as they move through the forming and cropping stations. Darker colors indicate earlier work-pieces while lighter colors indicate work-pieces that passed through the predicted section last.

A consequence of implementing the material damping to the entire sheet is the change in response when additional work-pieces are added to the simulation. Introducing additional work-pieces to the model affects the critical damping ratio. However, the additional work-pieces are positioned in the feeder which restricts their motion until they are progressed into the tooling. Therefore, the portion of the strip which is influenced by the damping remains the same while the effective damping is changed.

This consequence can be observed by the smaller predicted transient displacement and rotation in Figure 49 when compared to the response of the same model simulated with fewer work-pieces in Figure 45.

4.5.2.6 Outcome

While evaluating the steady-state behavior is of inherent interest to studying the dynamic response of the strip, the computational cost cannot be overstated. Referring to Table 3, the 25 stroke simulation took 437 hours to complete using LS-Dyna MPP with 32 cores. While the prediction of the distribution of steady-state responses would likely not require all 25 strokes, the simulation would still require at least 5 strokes before the predicted displacements and rotations are within a steady-state distribution.

Overall the results of the 'Deformable-Rigid Switching' model were promising, with individual models requiring a much lower amount of time to produce results for multiple strokes (Table 3) compared to the 'Detailed' model in Section 4.2. However, to generate an adequately sized training set with several hundred data points, the time for a simulation still needs further reduction. Even using a single stroke as a data point, a dataset with 500 samples would require almost half a year to produce. Therefore, attempting to predict the distribution of steady-state outcomes, as discussed in in this section, was deemed impractical and was dropped from the scope of this research project due to the immense computational cost that would be required. As an alternative approach to demonstrate the ML training procedure for this forming operation, it was decided to generate the ML training dataset based on the predicted response for a single press stroke.

Table 3: Completion time for the 'Deformable-Rigid Switching' model.

Model	SPM	Number of Carriers	Stretch-Web	Number of strokes	Simulation Time (seconds)	Number of Cores	Solver	Completion time (hours)
Deformable Rigid Switching	60	1	I	5	5	20	MPP	86
		2	I	5	5	20	MPP	105
	120	1	I	5	2.5	20	MPP	66
		2	I	5	2.5	20	MPP	63
		1	I	25	12.5	32	MPP	437
	180	1	I	5	1.66	24	MPP	64
		2	I	5	1.66	24	MPP	64

4.6 Final Model

The 'Final' model used to develop a ML training set considered a similar model set-up as the 'Deformable-Rigid Switching' model described in Section 4.5.2. This model included the final

geometry of the demonstrator tool (Figure 50), as well as varying inputs to the lifter and feeder. To make the generation of a dataset suitable for machine learning training tractable, further reductions in the computational cost were required, the most significant simplification in support of this being to consider only the first press stroke, as discussed in the previous section. In addition, the largest computational cost persisting in the 'Deformable-Rigid Switching' model is associated with the calculation of the mechanical behavior of the remaining shell elements.

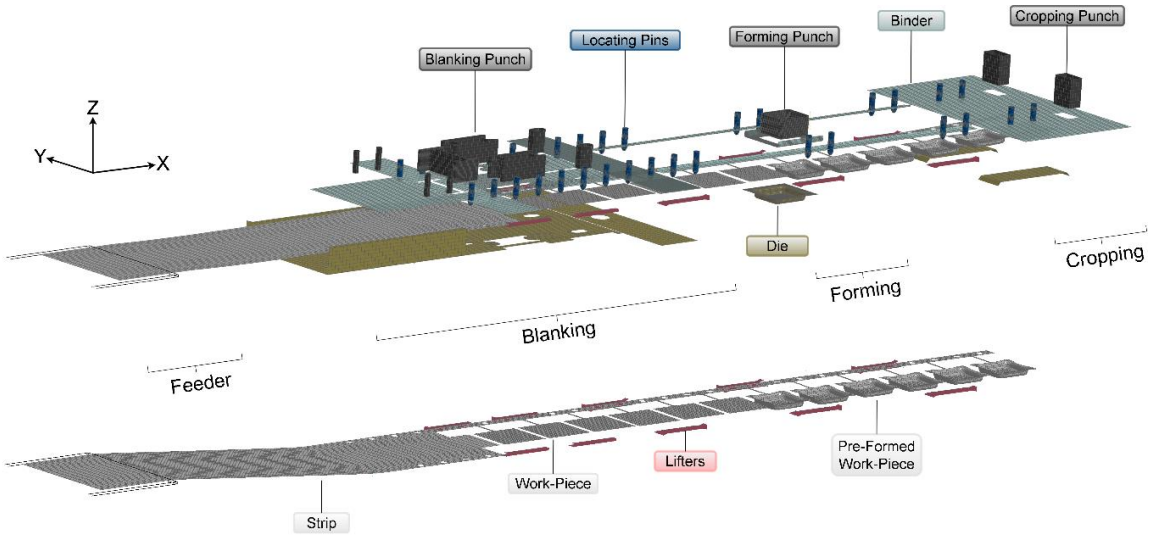

Figure 50: Overview of the 'Final' model.

The ideal approach to testing of each input permutation would consist of each simulation achieving an approximate steady-state behavior and predicting the 'work-piece placement' for multiple strokes to sample the distribution of placements. The computational cost of this idealized testing was very high and deemed intractable for this research. Therefore a more pragmatic approach was adopted that uses the outcome from a single press stroke. Referring back to the discussion on process outcomes in Section 4.4, this research is ultimately focused on predicting the placement of the work-piece after strip progression provided there are no collisions or strip misalignment. Therefore, in the case of a single stroke model, the simulation is only required to simulate the strip transfer such that the simulation can be completed once the binder clamps the work-pieces.

By focusing the model on the strip transfer, the forming operation can be removed from the model entirely. Not only is the total simulation time reduced by removing the forming process, but, the work-pieces may be modelled as rigid bodies throughout the entirety of the simulation. While such an

approach, in which only the transfer process is modelled, neglects the forming process itself, it is far more practical in terms of generating a data set for training a machine learning model, particularly if the goal of the simulation is to predict the accuracy of blank positioning and whether tooling collisions occur during transfer. Furthermore, the single stroke approach provides a heuristic for the performance a given process input by demonstrating which process inputs are not worth considering past a single stroke.

Figure 51 shows the general tool paths in the 'Final' model, and the portion of the press stroke which is considered. The simulation begins with the binder and lifters rising from the die face and is completed once the strip is transferred and is clamped by binder at the subsequent tooling operation. A total of 256 degrees of the crank angle are included in each simulation. The 'Final' model uses the restart file from a prior simulation to give the initial state of the operation. The initial simulation completes the forming operation and ends immediately before the lifters begin to rise allowing for the 'Final' model to be initialized at the beginning of the transfer process.

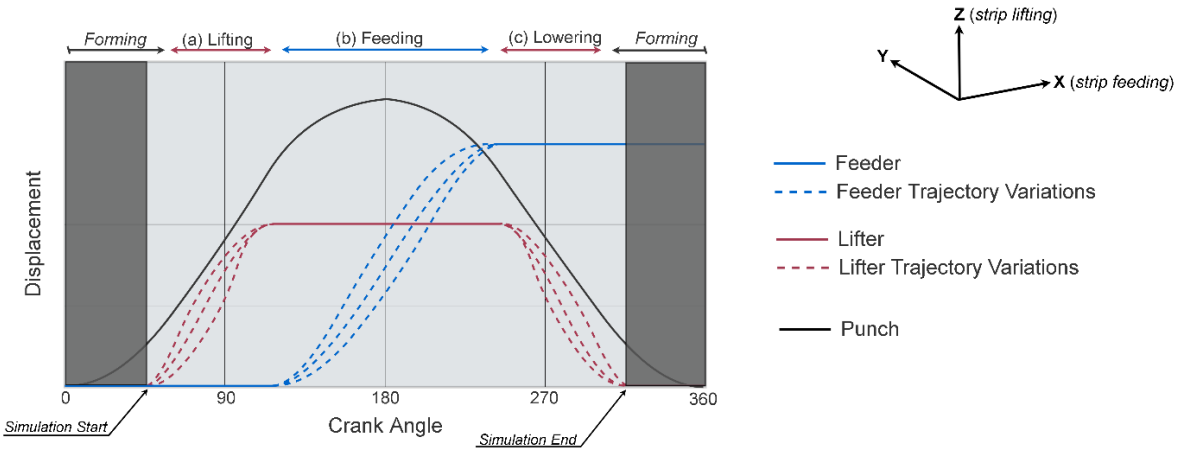

Figure 51: Illustration showing the start and end points of the 'Final' model which focuses on simulating the strip transfer.

To further reduce the computational cost of the model, the number of through-thickness integration points of the deformable shell elements was reduced. The integration points for the stretch-webs are reduced from seven to five. For the reminder of the strip, the number of integration points are reduced to three. The reduction in integration points was deemed to be justified since the model does not include a forming operation and the strip does not experience intentional yielding (yielding only occurred in the event of a collision with the tooling).

Pilot holes were reintroduced into the strip of the ‘Final’ model to properly model their role in locating the strip after the transfer operation. While coarsening the mesh along the strip carrier reduced the number of shell elements in the ‘Deformable-Rigid Switching’ model, the necessity of the locating pins was reconsidered and they were needed to more accurately capture the ‘work-piece placement’. In the ‘Deformable-Rigid Switching’ model, the strip would effectively be held in place once the locating pin boundary condition was applied whereas in practice the locating pins actually may pull the strip into position if the strip misalignment is marginal. The ‘Final’ model also introduces the un-blanked strip coil into the simulation. All prior simulations used a pre-blanked strip, which reduced the number of deformable shell elements but deviates from the reality of progressive dies. The inclusion of the additional material of the strip coil is necessary since it affects the strip dynamics, but was too computationally expensive for prior simulations. Since the ‘Final’ model only simulates a single stroke, meshing of the strip coil was simplified and utilizes a coarser discretization than other portions of the strip (Figure 52). Two strip layouts were included in the ‘Final’ model, an ‘I’ stretch-web and an ‘S’ stretch-web, both with a single strip carrier. While the demonstrator tool developed at TUM is not capable of producing an ‘S’ stretch-web, it was of interest to test a stretch-web that was more compliant than the ‘I’ and ‘O’ stretch-webs available in the current tooling set at TUM

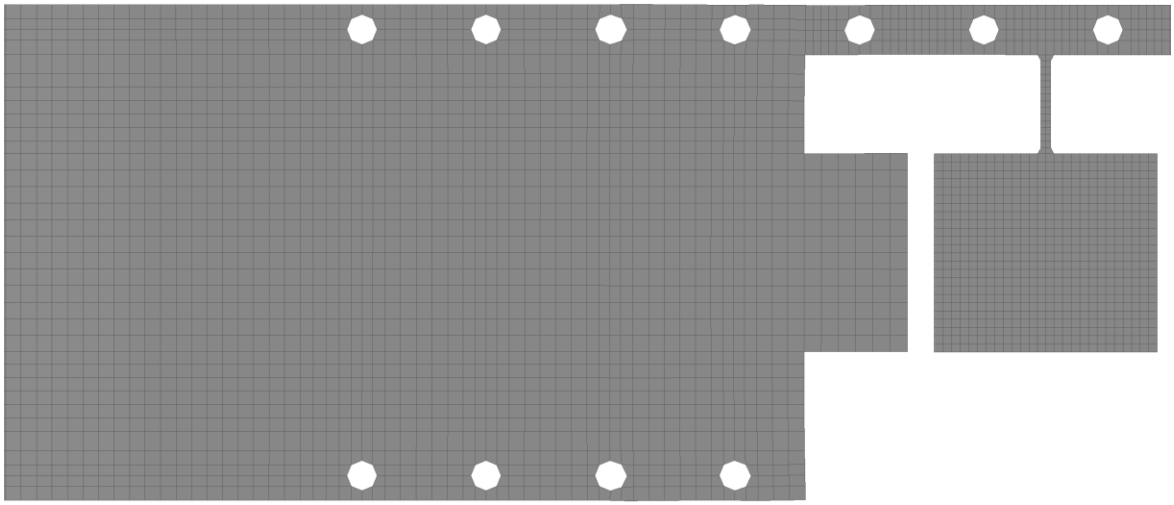

Figure 52: Section of the strip at the blanking operation where the strip coil is cut into individual work-pieces.

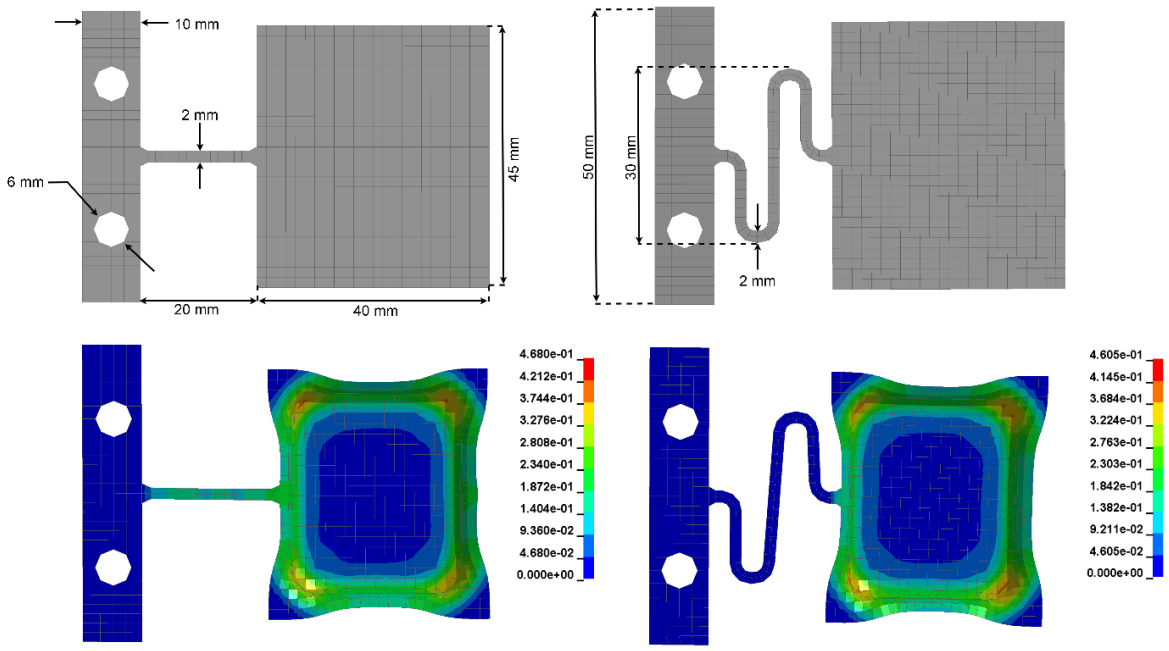

Figure 53: Strip layouts used in the 'Final' model with (left) 'I' stretch-web and (right) 'S' stretch-web. The nominal dimensions are shown on the top and the effective plastic strain after forming is shown on the bottom.

The remainder of the 'Final' model uses the same modelling techniques as described in the 'Deformable-Rigid Switching' model with exception of the inputs to the feeder and lifters. Instead of using passive lifters, active strip lifters are implemented to permit the use of various trajectories. The '*PRESCRIBE_MOTION_RIGID' keyword is used to prescribe a given trajectory to the strip lifters. The feeder uses the same implementation described previously, but, uses a variety of trajectories. The selection of input permutations, including the stroke rate, feeder trajectory, and lifter trajectory, will be described in Chapter 5 which documents the ML training approach. Furthermore, the results of the simulations for the 'Final' model will also be presented in Chapter 5.

Between the reduction in simulation time and number of integration points for the shell elements, the overall run-time of the 'Final' model is substantially reduced compared to the prior models. The runtime varied between under two hours to up to twelve hours (Table 4), depending on the stroke rate of the given permutation, making this model appropriate for generating a dataset, comprised of 1000 simulations, to train a machine learning model.

Table 4: Overview of completion times for all models.

Model	SPM	Number of Carriers	Stretch-Web	Number of strokes	Simulation Time (seconds)	Number of Cores	Solver	Completion time (hours)
Detailed model	60	4	I	10	10	8	SMP	647
Single Work-Piece	60	1	I	5	5	28	MPP	45
		2	I	5	5	32	MPP	N/A
		2	S	5	5	32	MPP	63
	120	1	I	5	2.5	32	MPP	22
		2	I	5	2.5	32	MPP	N/A
		2	S	5	2.5	32	MPP	31
Implicit	120	2	I	4	2	8	SMP	5
Beam Element	120	2	I	4	2	24	MPP	38
Deformable Rigid Switching	60	1	I	5	5	20	MPP	86
		2	I	5	5	20	MPP	105
	120	1	I	5	2.5	20	MPP	66
		2	I	5	2.5	20	MPP	63
		1	I	25	12.5	32	MPP	437
	180	1	I	5	1.66	24	MPP	64
		2	I	5	1.66	24	MPP	64
Final	60	1	I	1	1	24	MPP	5.8
		1	S	1	1	24	MPP	12.15
	360	1	I	1	0.1667	24	MPP	1.8
	330	1	S	1	0.1667	24	MPP	1.5

Chapter 5
Smart Trajectory Generator

5.1 Data Generation

In a conventional mechanical press, the timing of the strip lifters and feeder is controlled by the crank angle of the press. The adoption of controllable strip lifters and feeder presents an interesting opportunity for fine tuning of the strip transfer. Not only is the trajectory adjustable for both the strip lifters and feeder, but, the intervals in which the lifting and feeding are also adjustable. The lifters can be actuated over a longer period than would be permitted when actuated passively by springs. Furthermore, the actuation of the strip feeder may be permitted to overlap with the strip lifting and lowering rather than having each phase of the progression occur sequentially and independently. However, considering both the interval and the trajectory is a very large permutation space to consider when testing various input permutations with a large computational cost associated with each permutation. To respect the general goals of this thesis, the interval in which the strip lifters and feeder are actuated remains fixed. Last, the trajectory of the lifter for raising and lowering the strip is symmetrical. In the context of the 'Final' model, the goals of the machine learning models can now be specified more clearly. Given the initial state of the 'Final' model, $X(t_0)$, find the set of input permutations, {SPM, f_{feed}, f_{lift}} such that the error in the placement of work-piece {Δx, Δy, $\Delta\theta_z$} is within a specified tolerance provided there is no collision or strip misalignment. The set of identified permutations which satisfies the specified tolerance constitutes a look-up table which can be used for selecting suitable process control inputs. To take advantage of the adjustable control intervals, the strip lifters actuate over a period of 66 degrees of the crank angle for both the lifting and feeding phases, instead of 44 degrees which would be used in a passive configuration. The feeder is actuated over 120 degrees of the crank angle with the lifting and feeding phases separated by 2 degrees of the crank angle. The goal was to produce 500 simulations for both the 'I' and 'S' stretch-webs for a total of 1000 simulations using the 'Final' model with a single carrier strip layout. The single carrier layout was selected since it represented a greater challenge to maintain positional accuracy and avoid collisions due to the higher compliance of this configuration relative to a double carrier layout.

5.1.1 Trajectory Selection

Revisiting the purpose of the machine learning model, the goal is to use the inputs to the process, {SPM, f_{feed}, f_{lift}}, to predict the process outcome, {Δx, Δy, $\Delta \theta_z$}. An important aspect of this goal is that the trajectories must be used as inputs to the machine learning model. To avoid a large feature input size, the class of function used to produce the trajectory must be parametrically defined with variables that describe the shape of the curve. Splines, and in particular, cubic splines, provide a convenient continuous function to parametrically describe trajectories using a simple set of parameters. Splines are piecewise polynomial functions defined by a vector of points called 'knots'. In the case of a cubic spline, the piecewise polynomials are third-order. In the case of a 2D curve, each knot is specified by an X-Y coordinate which can be used to easily adjust the shape of the curve. Therefore, for each knot used in the splines, two additional inputs to the ML model are required.

Since there is not any indication as to whether the strip progression or strip lifting is more critical for reducing the dynamic response or the shape of curve which is optimal, the spline creation for both use the same process. The splines are used to describe the displacement trajectory as a function of the crank angle. The knots at the beginning and endpoints for the splines are defined by the displacement of the lifter and feeder respectively which remain fixed between each input permutation. Furthermore, the velocity at both endpoints are set to zero. The splines are defined by the intermediate knots located between the endpoints. A total of two intermediate knots are used to define each spline because a wide variety of trajectories can be produced while still maintaining a relatively low number of input features. With two knots used to define each spline, four inputs are required for each trajectory (X and Y position of each knot).

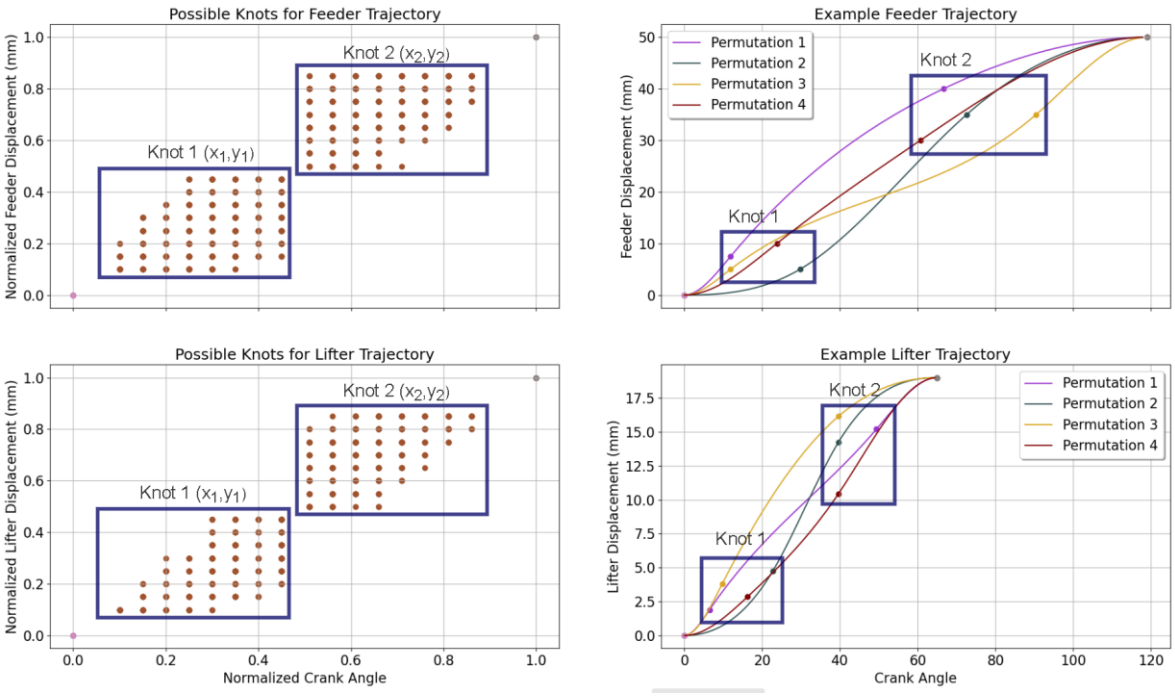

Figure 54: Possible knots for feeder and lifter trajectories (left) and example of feeder and lifter trajectories (right).

Additional restrictions are applied to the trajectories to limit the complexity of the simulation solution-space and provide a better opportunity for the ML models to learn the behavior of the system. All splines are required to be monotonically increasing to restrict the feeder and lifters from reversing their motion during their trajectory. Furthermore, the splines were restricted to avoid saddle points and intermittent periods of near zero velocity. More specifically, the normalized velocity was bounded above 0.25 between the intermediate knots. Restricting the velocity between the intermediate knots prevented the selection of splines which slow down the feeder or lifter displacement in the middle of the trajectory which would require larger accelerations and decelerations at the beginning and end of trajectories to compensate for the intermediate slow period. To further refine the selection of trajectories, the intermediate knots were restricted to a sampling area where the normalized X and Y coordinates of 'Knot 1' are bounded between [0.1, 0.5] and 'Knot 2' is bounded by [0.5, 0.9]. Finally, the binder limits the displacement of the lifters and therefore the splines considered for lifter trajectory are upper bounded by the binder trajectory.

To generate the input permutation space of possible trajectories, the intermediate knots are sampled from their respective bounded coordinates using a grid with spacing of 0.05 between each sample,

forming an 8x8 grid for a total of 64 possible positions for each knot and a total of 4225 combinations. Any of the 4225 trajectories that do not satisfy the restrictions previously mentioned are removed from the set, leaving a total of 606 feeder trajectories and 363 lifter trajectories.

5.1.2 Permutation Generation

The input permutations were generated independently for each stretch-web. For each permutation, the lifter and feeder inputs were randomly sampled from the possible 606 feeder trajectories and 363 lifter trajectories. A stroke rate was also randomly sampled from a specified range. In the case of the 'I' stretch-web, the stroke rate was randomly sampled in the range of 60 to 360 SPM. After reviewing the results from the first 500 simulations, the sampling range was changed to 60 to 330 SPM for the 'S' stretch-web. To generate all 1000 simulations took roughly six weeks of continuous computation with two simulations running in parallel at a time using LS-Dyna MPP with 16 cores on an Intel Xeon 8160 cluster.

5.1.3 Data Labelling

Using the description of process outcomes in Section 4.4, the predicted outcome of each simulation can be labeled. The process outcome is quantified by the placement of the work-piece provided that no collisions or strip misalignment occurs. Therefore, first the simulations are labeled whether they are able to successfully place the work-piece or if there is a collision or strip misalignment. Four possible outcomes are identified: (1) 'collision', (2) 'strip misalignment'', (3) 'early failures', and (4) 'successful'. (1) A collision is defined if there is a non-zero contact force between the strip and tooling during strip feeding. Contact during the strip lifting is not considered since there can often be chatter between the strip and binder during the lifting and lowering of the strip. Furthermore, the contact between the strip lifters and strip are not considered a collision. (2) A strip misalignment is defined by the improper location of the locating pins in the pilot holes. Referring to Figure 55, the X and Y position of nodes around the locating holes are predicted and compared against the position of the locating pins. A simulation is classified as 'strip misalignment' if the position of the predicted nodes is outside of the permitted tolerance, which is taken to be 0.1 mm. (3) 'early failures' are simulations which demonstrated erratic behavior as a result of errors in the set-up of the FE simulation. These errors arose from missing or incorrect boundary conditions and therefore permutations which result in an early failure are removed from the dataset. (4) 'Successful' outcomes are any simulation which were able to place the work-piece after strip progression without a 'collision', 'strip misalignment' or 'early failure'.

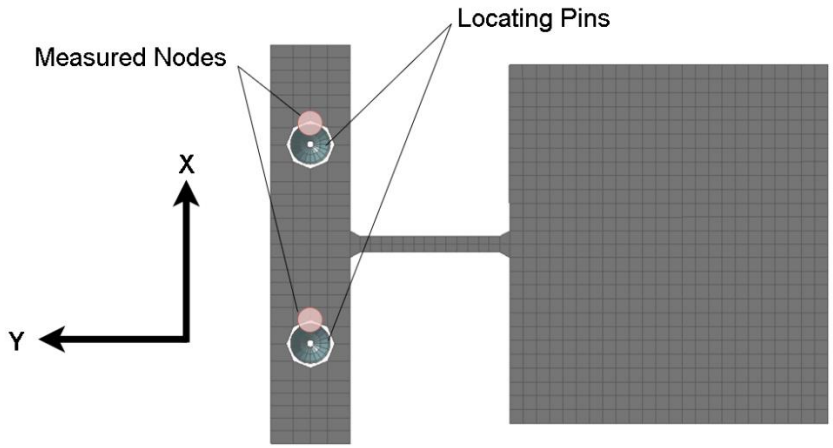

Figure 55: Example of nodes used to predict the strip alignment.

The outcome of the simulations for each stretch-web type can be seen in Figure 56. For both stretch-webs, a majority of simulations were ‘successful’ and able to place the work-piece. In general, when a ‘strip misalignment’ is observed a ‘collision’ is also observed. However, a considerable number of permutations produced a ‘collision’ without incurring a ‘strip misalignment’. ‘Early failures’ were observed in a minority of permutations. Permutations with ‘early failures’ were removed from the samples leaving a dataset size of 461 and 466 for the ‘I’ and ‘S’ stretch-webs respectively.

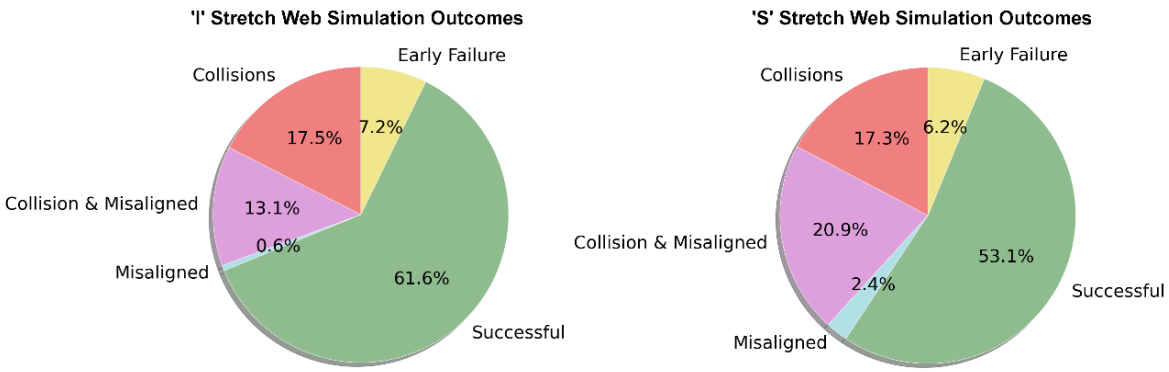

Figure 56: Simulation outcome for 'I' (left) and 'S' (right) stretch-web.

5.1.4 Data Exploration

Similar to Chapter 4, the dynamic response of the strip can be predicted throughout the duration of each simulation to illustrate the effects of the control inputs. Figure 57 shows the response of work-pieces that were successfully transferred onto forming die for both stretch-web types. The response of the

work-piece is predicted at the center-of-mass for all ‘successful’ permutations. All degrees of freedom are predicted with exception of the transfer displacement (X-axis). Light colours (yellow) indicate higher stroke rates and darker colours (dark blue) indicate lower stroke rates. The influence of the stroke rate is observed most clearly in the displacement along the Z-axis (lifting). As the stroke rate increases, the overshoot of the work-piece above the nominal lifting height (19 mm) increases which makes a collision between the work-piece and binder more probable. While the work-piece overshoots the nominal lifting height at higher stroke rates, the strip carrier remains in the strip lifters. The difference in height between the strip carrier and work-piece as a result of overshooting causes the work-piece to rotate about the X-axis (feeding direction). While the rotation about the X-axis and displacement along the Z-axis are explained by higher lifting speeds, rotation about the Z-axis is influenced primarily by the strip feeding. Similar to the strip lifting, a higher stroke rate requires a higher acceleration of the strip feeding which causes larger rotations of the work-piece about the Z-axis, particularly in the case of the ‘S’ stretch-web. In general, the same trend applies to the remaining degrees of freedom where a larger response is observed at higher stroke rates.

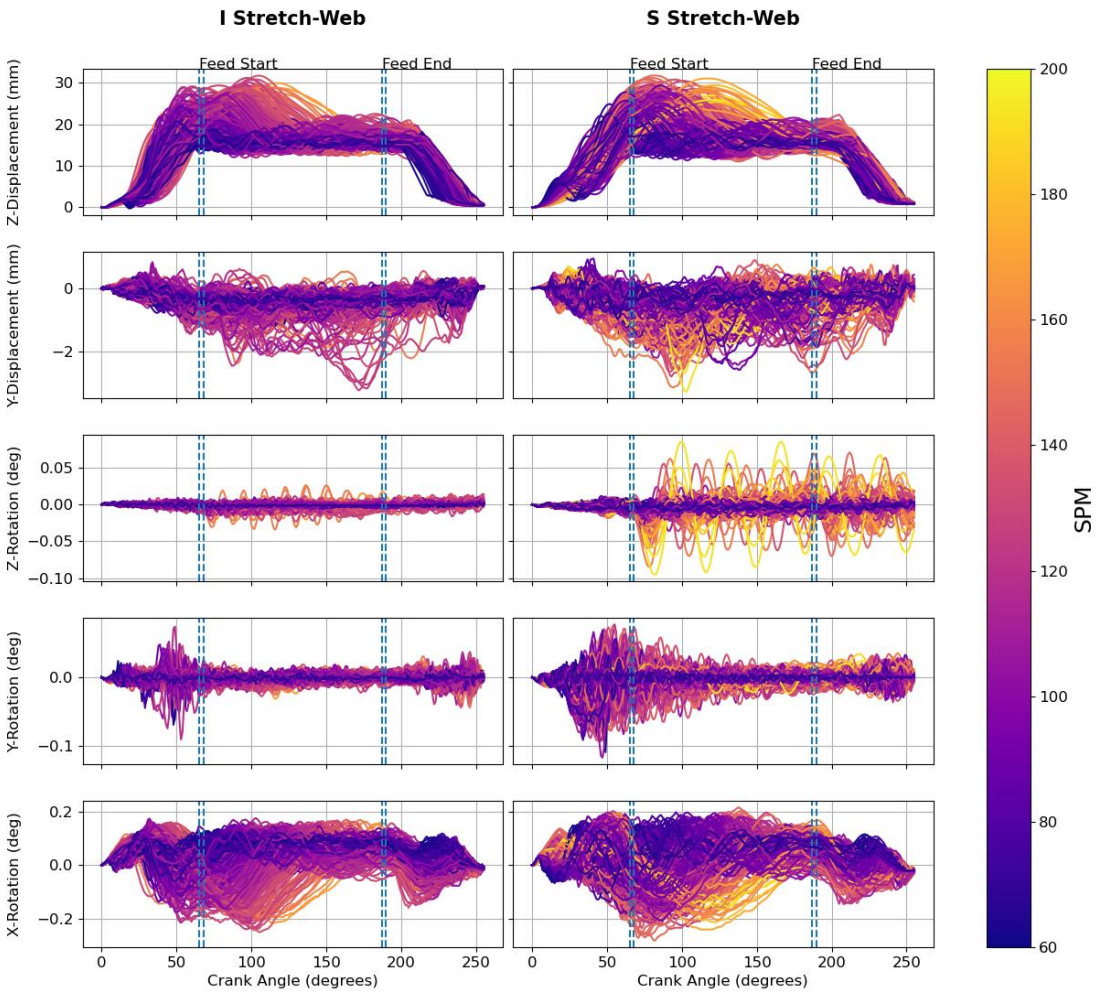

Figure 57: Dynamic response of 'successful' permutations for the 'I' (left) and 'S' (right) stretch-webs predicted in various degrees of freedom, coloured by stroke rate.

The influence of the stretch-web selection on the response of the work-piece can be observed by overlaying the distributions of the responses. Figure 58 shows the mean response for each stretch-web (solid lines) and two standard deviations from the mean (shaded areas) for the same degrees of freedom as in Figure 57. The 'S' stretch-web exhibits a larger variance in the predicted response for each degree of freedom as a result of its more compliant structure. The increased variance is most notable when predicting rotation about the Y-axis (stretch-web axis) and rotation about the Z-axis (lifting direction). The increased variance in the rotation about the Z-axis is particularly important since any rotation will

influence the placement of the work-piece on the forming die (X-Y plane) and ultimately affect the final formed geometry of the work-piece. Since the ‘S’ stretch-web produces a larger variance in the predicted dynamic response as compared to an ‘I’ stretch-web, it is recommended that the ‘S’ stretch-web should only be used in a strip layout when additional compliance is required during the forming operation.

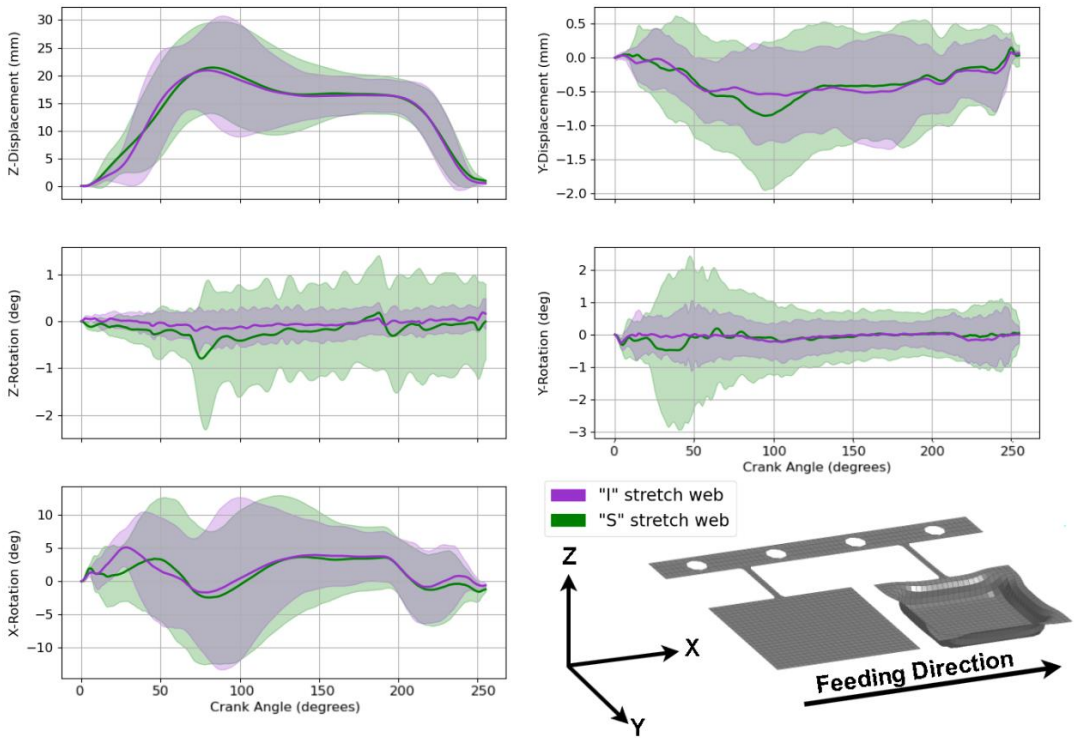

Figure 58: Distribution of dynamic responses of ‘successful’ permutations for the ‘I’ (purple) and ‘S’ (green) stretch-webs predicted in various degrees of freedom.

To further see the influence of the stretch-web selection, the ‘work-piece placement’ on the forming die can be predicted. Figure 59 shows the placement of the work-piece for ‘successful’ permutations for both stretch-webs. The influence of the additional compliance of the ‘S’ stretch-web can again be observed by the wider distribution of ‘work-piece placement’ as a result of the higher variance in the dynamic response as shown in Figure 58. Furthermore, the ‘I’ stretch-web has limited error along the Y-axis as compared to the ‘S’ stretch-web. The ‘S’ stretch-web’s larger rotation about the Z-axis at the end of the predicted response contributes to the larger distribution in blank-placement.

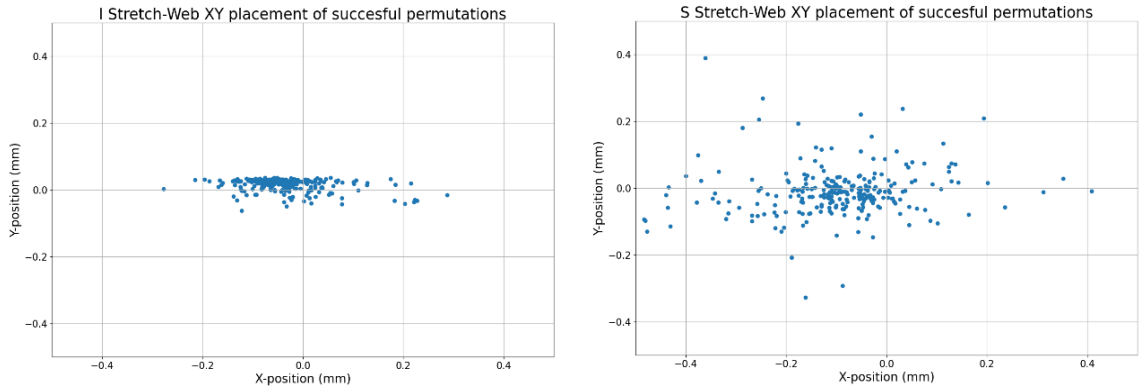

Figure 59: 'Work-piece placement' of 'successful' permutation for 'I' (left) and 'S' (right) stretch-webs.

To further illustrate how overshooting during the lifting phase of strip transfer can increase the probability of collisions, Figure 60 shows the minimum clearance between the binder and the nearest work-piece during the strip feeding versus the stroke rate for both stretch-web types. For every permutation generated by the FE model, the 'minimum feed clearance' is plotted against the stroke rate (Figure 61). The colour of each permutation indicates whether the simulation was a 'failure' (*i.e.* 'strip misalignment' or 'collision') or was a 'success' with all 'early-failure' permutations removed.

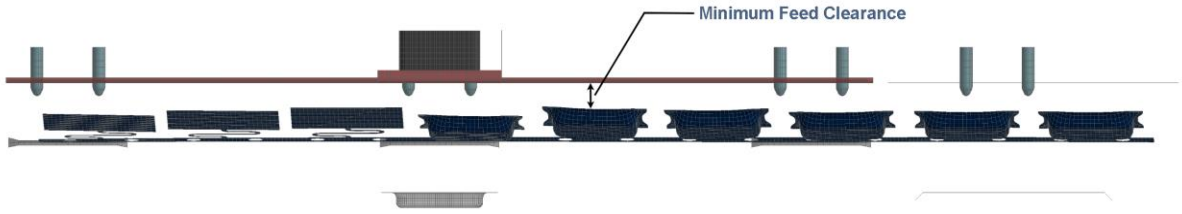

Figure 60: Illustration of the 'minimum feed clearance' which is the closest predicted distance between all work-pieces and the binder during strip progression.

A linear trend emerges in Figure 61 showing that the stroke rate is highly correlated to the overshooting of the work-pieces during the lifting phase. As the 'minimum feed clearance' decreases to zero, the proportion of failed permutations increases, with all permutations below zero resulting in a 'failure'. Since the predictions for the minimum clearance are taken between the binder and center-of-mass of each work-piece, several permutations fail prior to zero clearance since there are areas of the work-piece which are higher than the center-of-mass.

While the stroke rate has a significant impact on the outcome of each permutation, the influence of the feeder and lifter inputs is observed when a given permutation deviates from the linear trend and can be understood as using ‘high’ and ‘low’ performing control inputs for the feeder and lifter. It should be noted that this loose definition of performance only relates to the susceptibility of a given permutation to result in a collision or strip misalignment and does not relate to the placement of the work-piece on the forming die. At a given stroke rate, permutations which achieve a larger minimum clearance than the general trend can be considered ‘high’ performing while a minimum clearance below the trend at the same stroke rate would indicate a ‘low’ performing permutation.

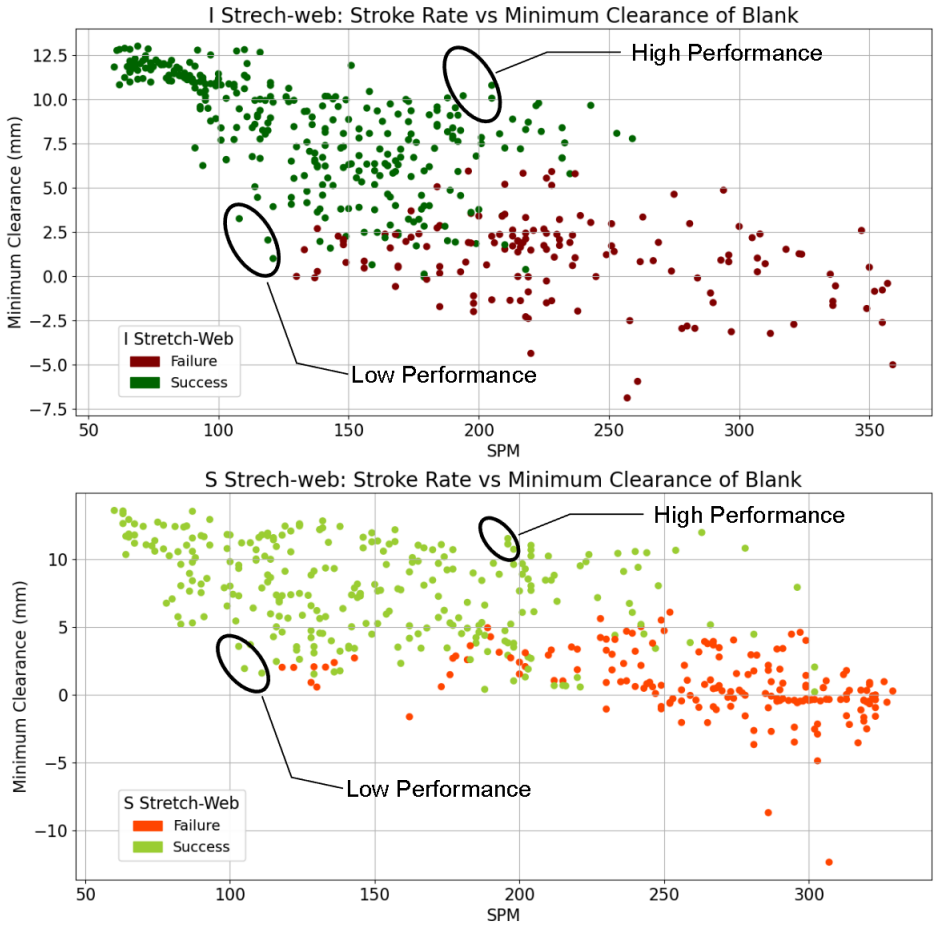

Figure 61: Minimum clearance of work-piece from binder during strip feeding versus SPM for all permutations except ‘early failures’ for the (top) ‘I’ and (bottom) ‘S’ stretch-webs.

The six ‘high’ and ‘low’ performance permutations circled in Figure 61 are plotted in Figure 62 and serve to illustrate how the ‘minimum feed clearance’ and the overshooting of the work-piece is

correlated to the shape of the lifter trajectory, with little influence from the feeder trajectory. For the lifter trajectory, the selected ‘low’ performance permutations followed an ‘S’-shaped trajectory whereas the ‘high’ performance permutations followed a more linear trajectory. With respect to the selected permutations of the feeder input, the shape of the ‘high’ and ‘low’ performance trajectories cannot be as easily categorized or distinguished. This indicates limited correlation between the shape of the feeder trajectory and whether the permutation was labeled as ‘high’ or ‘low’ performance. However, limited correlation to the feeder trajectory should be expected since the categorization of the ‘high’ and ‘low’ performance permutations were based on the ‘minimum feed clearance’ which is largely influenced by the stroke rate and the lifters causing the work-piece to overshoot.

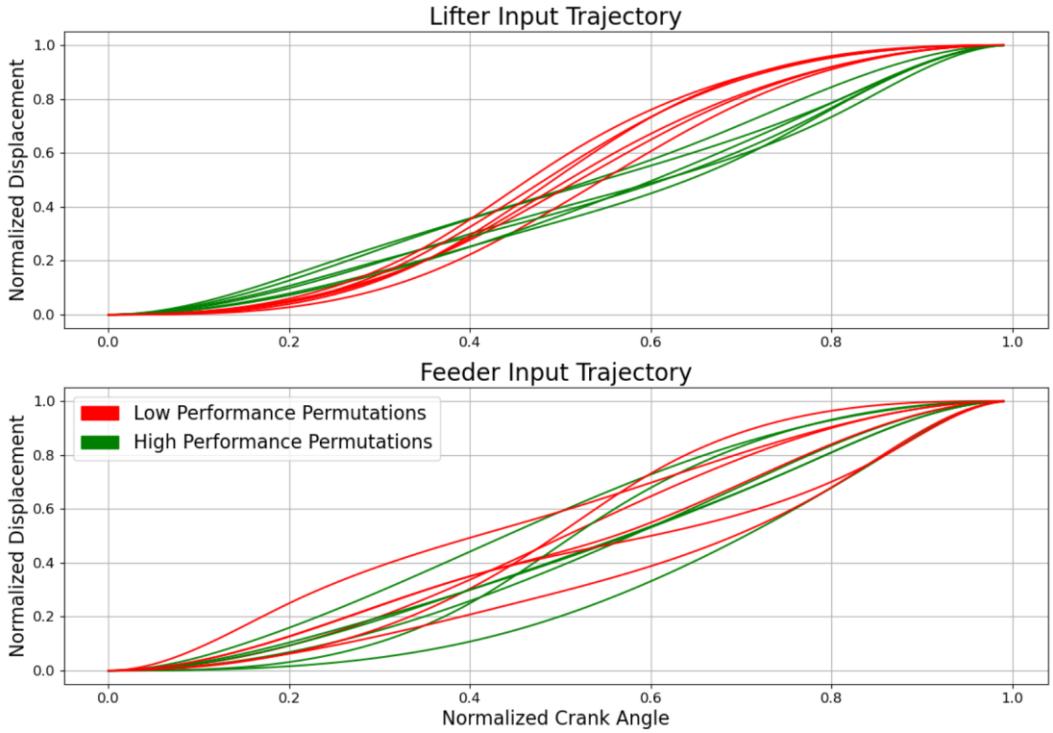

Figure 62: Examples of 'high' and 'low' performance permutations as indicated in Figure 61.

5.2 Machine Learning

By training a machine learning model on data generated from FE simulations using various control inputs, all possible permutations of control inputs can be interpolated. The process of interpolating all possible permutations generates a look-up table which allows for identifying the maximum stroke rate for any given combinations of lifter and feeder trajectories which can be achieved with a desirable

process outcome. The best performing control inputs which allow for the maximum overall stroke rate for the process may than be identified and used in additional simulations for observing the progressive die operation over multiple strokes of the process. All machine learning models were trained on an Intel Core i9-9900K CPU, however, future work should consider using a GPU to reduce the training time through parallelization and permit a wider hyperparameter search.

5.2.1 Inputs and Outputs

Each of the FE simulations used to generate the ML training set considered separate control inputs (*i.e.* lifter trajectories, feeder trajectories, and stroke rate), with the remainder of the simulation set-up being identical. Therefore, the variation in predicted process outcomes is a result of the variations in the control inputs. Consequently, the inputs to the machine learning models are limited to the control inputs of the FE model.

The stroke rate is a simple discrete value in the range of 60 to 360 SPM which is suitable for most machine learning architectures. The lifter and feeder trajectories are continuous and can be used in a variety of ways. In this work, the lifter and feeder trajectories are defined using the two adjustable knots. Therefore, the 'X' and 'Y' position for each of the two adjustable knots for both trajectories can be used as feature inputs. This would result in four features for each trajectory. However, a small difference in the position of a knot can lead to a large difference in the shape of the final trajectory. Early results from the machine learning models indicated that the models were unable to associate 'families' or 'groups' of curve shapes with a given process outcome. Using the knot positions alone does not inform the learning algorithms about the influence of the trajectory shape. In other words, the early machine learning models were unable to capture any spatial information.

CNNs provide an excellent solution for learning spatial information [128]. By applying a 1-D convolutional filter across the lifter and feeder trajectories, a CNN could extract spatial information which is predictive of the process outcome. However, the use of a CNN increases the number of learned weights in the machine learning model. With limited data, a CNN approach was susceptible to overfitting the training data and performing poorly on test data, making this architecture untenable for the dataset size.

Inspired by the use of CNNs, an alternative approach to identify spatial information was created. By introducing a metric which compares the similarity of trajectories, the machine learning models are provided the opportunity to predict process outcomes based on the similarity of a given trajectory to

the shape of other trajectories. A simple metric for measuring the similarity of curves is the absolute area between curves. A set of $t \times n$ and $t \times m$ matrices can be constructed by finding the absolute area between each pair of trajectories for the lifter and feeder in the training dataset of size t and all the considered trajectories for the lifter and feeder of size n and m respectively. However, these additional inputs create a total of $n + m$ additional features to the machine learning model. A high dimensionality feature input space introduces the 'curse of dimensionality' in which the space between data points grows with additional features requiring exponentially more data to generate a reliable estimator for the space [128]. Therefore, to reduce the dimensionality of the additional features constructed from the trajectory similarity while retaining the data's variation, principal component analysis (PCA) is used. The number of principal components is considered a hyperparameter which is identified as described in Section 5.2.3.

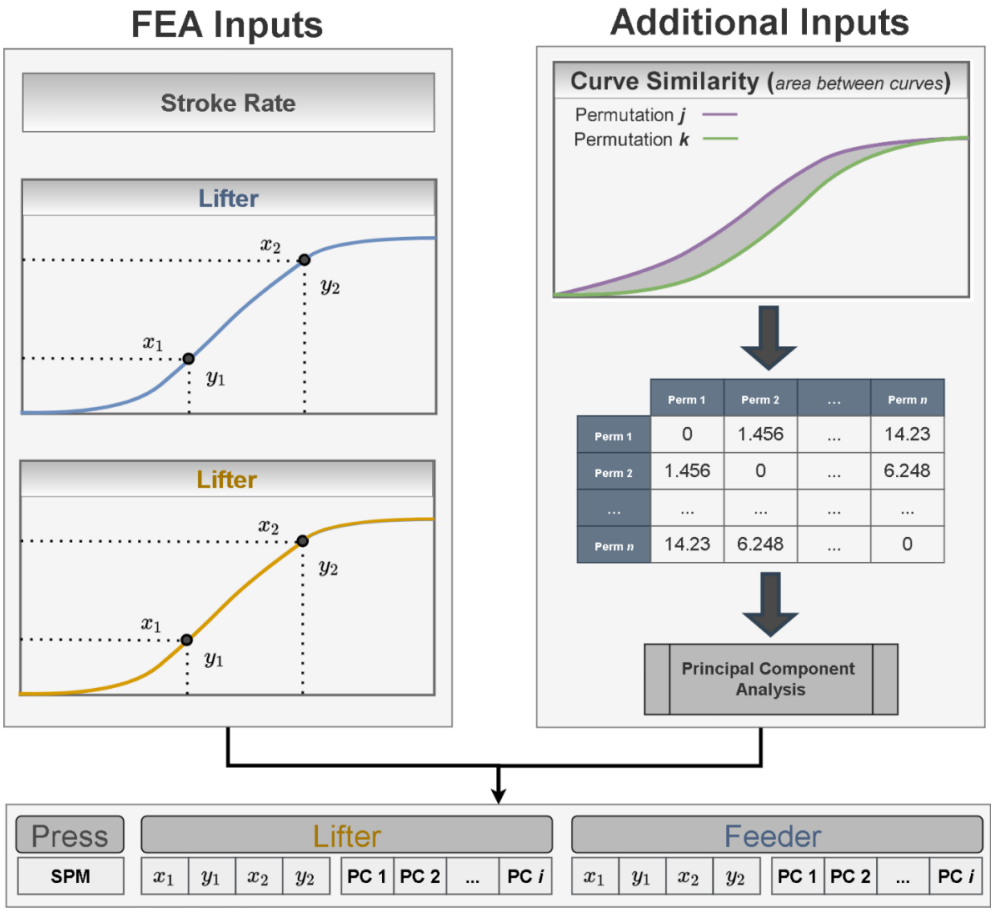

Figure 63: Overview of feature inputs used for machine learning models.

As discussed in Section 4.4, when studying the dynamic response of the strip in progressive die sheet metal forming, the most critical outcome which should be predicted to assess whether the dynamic response is acceptable is the placement of the work-piece. However, the placement of the work-piece is contingent on whether a collision or strip misalignment occurred during strip progression. Therefore, two machine learning models are developed. The first model is used to predict whether a collision or strip-misalignment occurred, and the second to predict the placement of the work-piece. These two machine learning models are used sequentially to first predict whether a given permutation is at risk of a collision or strip-misalignment and then predict the placement of the work-piece for permutations which are not at risk.

Referring to Section 5.1.4, the vast majority of strip-misalignments occurred in unison with a collision, with only 0.6% and 2.4% of strip-misalignments occurring without a collision for the 'I' and 'S' stretch-webs, respectively (Figure 56). Therefore, in the context of a machine learning model used to predict whether a collision or strip misalignment occurred, it makes sense to focus on predicting whether a collision occurred. However, simply predicting whether a collision presents a problem when interpolating all possible control inputs to identify the maximum stroke rate at which the tooling can operate at. There is a threshold where at a given stroke rate, the machine learning model would predict there is no collision, but, with a small increase in the stroke rate, a collision may be predicted. Operating at the dynamic limit of the process is unsuitable for high volume production components. Therefore, it is prudent to add a risk tolerance to machine learning predictions. The cause of collisions in the simulated process is a result of the work-pieces overshooting the nominal lifting height and colliding with the binder during strip progression. Rather than predicting whether that collision occurs, predicting how close the work-pieces gets to the binder can provide a safety factor in the prediction. The 'minimum feed clearance' (Figure 60) can be predicted for each simulated permutation and then labeled as to whether it was above or below a specified tolerance. Therefore, the first machine learning model is a classifier which predicts whether a given permutation is above or below the specified 'minimum feed clearance' used to label the data.

The second machine learning model focuses on predicting the placement of the work-piece after strip progression. Since the work-piece geometry considered in this work is not based directly on a production component, there does not exist specific tolerance requirements for the final formed geometry. Without specific tolerances for the formed geometry, it is not possible to directly identify tolerances for the placement of the work-piece on the forming die. Instead, a variety of tolerances are

considered in this work. There are three degrees of freedom for the placement of the work-piece on the forming die; X-position, Y-position, and Z-Rotation. Since the work-piece is connected to the stretch-web, any error in the Z-rotation of the placed work-piece is highly correlated to the X and Y position. Therefore, the placement of the work-piece may be characterized in terms of the error in the X and Y position. Furthermore, focusing on the X and Y position allows the tolerance to be reduced to a single tolerance which quantifies the magnitude of the error vector (Equation 16).

$$Placement\ Error = \sqrt{{y_{error}}^2 + {x_{error}}^2} \tag{16}$$

Similar to the 'minimum feed clearance' classifier, a permissible placement tolerance can be specified for labeling the data. If the position of a given permutation is within the specified tolerance, then it is labeled as successful, if it is outside the tolerance than it is labeled as a failure. Therefore, the second machine learning model is a classifier which predicts whether a given permutation is placed within a specified placement tolerance.

By using both classifiers described above sequentially, successful process control inputs can be identified (Figure 64). The full datasets for both the 'I' and 'S' stretch-webs are used to train the 'minimum feed clearance' classifiers using a specified tolerance for the feed clearance to label the data. Permutations which are predicted to be above the permissible tolerance are discarded and labeled as 'failed' permutations. Permutations which are predicted to be below the specified tolerance are used to construct a new dataset. The 'work-piece placement' classifier is used to predict whether a given permutation in the new dataset is within the placement tolerance. By training each classifier with the generated data from the FE simulations, all possible combinations of process control inputs can be evaluated to predict suitable process control inputs.

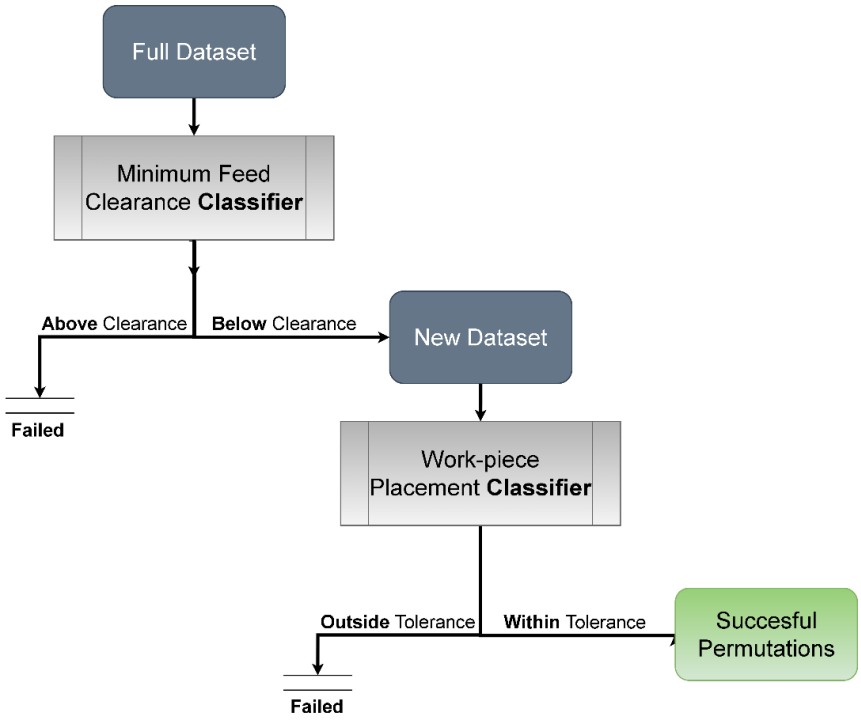

Figure 64: Flow diagram of machine learning models to predict successful process control inputs.

5.2.2 Architecture Selection

As described in Section 2.4, a multitude of machine learning models exist for classification with the most widely adopted approach in the literature for tabular data being the FFNN [89] which was introduced in Section 2.4.2. Even though FFNNs have been shown to perform very well on a wide range of datasets, it cannot be assumed that they would produce the lowest prediction error as compared to other machine learning architectures. Therefore, it is good practice to consider at least one other architecture for comparison when attempting to build an estimator for a dataset.

A commonly used algorithm which can achieve performance similar to FFNNs on many datasets are decision trees which were discussed in Section 2.4.3. Decision trees, in their standard implementation, offer a high degree of transparency in terms of the rationale behind the prediction process, but can be augmented with techniques such as ensembles and gradient boosting to improve their performance at the cost of this transparency [97]. One of the most popular augmentations is the XGBoost algorithm [98]. XGBoost utilizes regularized gradient boosting to sequentially build an ensemble of ‘weak’ learners which each focus on data samples the previous learner found difficult to predict.

This research considers both the FFNN and XGBoost architectures for predicting the process outcomes of the progressive die operation. Both architectures are implemented with a variety of hyperparameters during the model selection procedure.

5.2.3 Model Selection Procedure

There are multiple approaches to training and testing a machine learning model with each having their drawbacks. For all approaches to training a machine learning model, it is important to estimate the prediction error to evaluate the generalizability of the model to unseen data. One of the most widely used approaches is cross-validation which is often referred to as K-fold cross-validation [86]. This technique splits the dataset into *K* roughly equal-sized parts. The machine learning model is trained on *K*-1 folds and is then used to predict the remaining fold to evaluate the prediction error. This procedure is repeated *K*-times until each fold of the dataset has been used for evaluating the prediction error. While the selection of *K* is often either five, or ten, it may be as low as two and up to size *n*, where *n*, is the size of the dataset. When *K*=*n*, the machine learning model is approximately unbiased when estimating the prediction error, but, will suffer from high variance since each of the training sets are very similar. Another consideration is the high computational cost of training *n* models, especially when hyperparameter tuning is required. Inversely, a low value of *K*, may result in a higher bias, but less variance in the estimated prediction error.

A drawback of K-fold cross-validation is the issue of 'leaked' data when hyperparameter tuning. Since the same data is used for both training and testing at different stages, the hyperparameter selection procedure yields an overly optimistic estimate of the prediction score [129]. To overcome this challenge, nested cross-fold validation can be used which utilizes an inner and outer cross validation loop (Figure 65). The original dataset is split into *K* folds. The first set of *K*-1 folds is then taken and used to perform the inner-loop of the *K*-fold cross-validation, where it is split further into *K* folds. A multitude of models are trained and tested on the inner cross-validation loop with varying hyperparameters. The model with the highest average accuracy is then selected, trained on the *K-1* folds of the outer loop and evaluated on the remaining fold. Since different models may be selected for each inner loop, this procedure is effectively a model selection procedure for identifying suitable hyperparameters for each architecture considered.

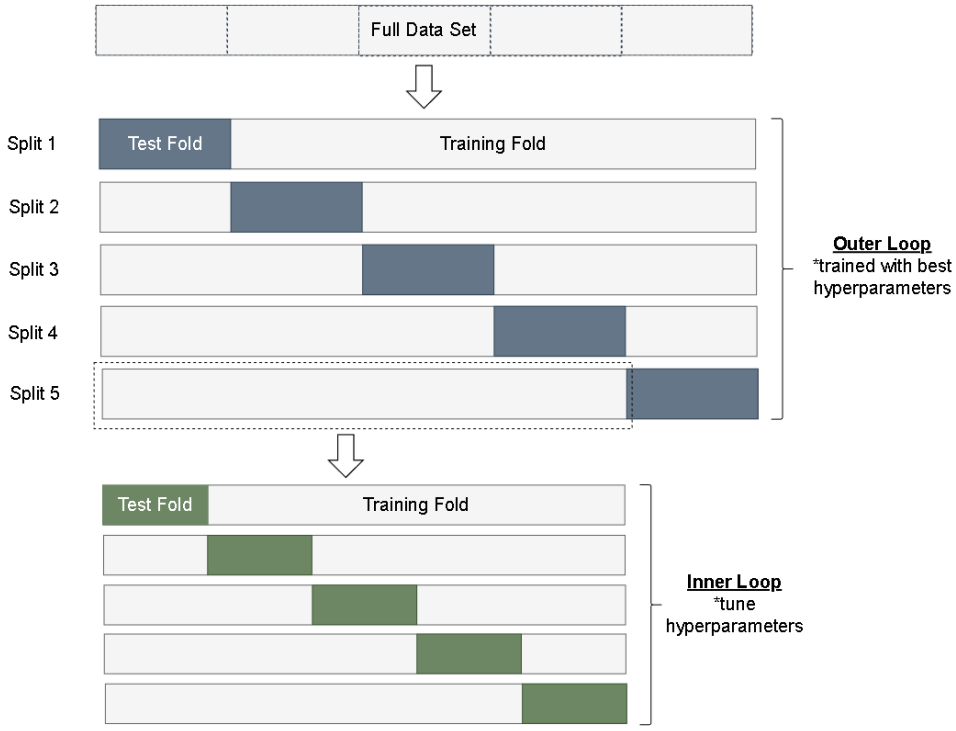

Figure 65: Example of nested cross-fold validation with five folds for each loop.

Nested cross-fold validation was adopted in this thesis with five folds for both the inner and outer loop to balance the computational cost of training models with various hyperparameters and the bias of the prediction error. As stated in Section 5.2.1, two classifiers are constructed for both the 'I' and 'S' stretch-web datasets separately using both a FFNN and XGBoost architectures. For each of the classifiers constructed, the permissible tolerance discussed in Section 5.2.1 is varied to relabel the data. For the 'minimum feed clearance' classifier, the tolerance is varied from 0 mm of clearance up to 10 mm of clearance (Figure 66). The tolerance for the 'work-piece placement' classifier is varied from 0.05 mm up to 0.25 mm (Figure 67).

The hyperparameters considered for the FFNN and XGBoost machine learning models are presented below in Table 5 and Table 6. A grid was constructed for all possible combinations of hyperparameters to be used with their respective architectures. The inner loop of the nested cross-fold validation trains a model for each combination of hyperparameters in the grid and selects the hyperparameters which achieve the highest average accuracy on the inner cross-validation. Both architectures consider 5, 10,

or 15 principal components for conducting the PCA on the lifter and feeder similarity matrices described in Section 5.2.1. The FFNN can use one of many activation functions for the neurons in the various layers including the Rectified Linear Unit ('ReLU') and hyperbolic tangent ('tanh') activation functions which are common choices for the neurons in the hidden layers since they are considered to generally perform better than a sigmoid activation for hidden neurons [130]. The formulas for the ReLU, tanh, and sigmoid activation functions can be seen in Equations 17, 18, and 19 respectively.

$$f(x) = \max(0, x) \tag{17}$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{18}$$

$$f(x) = \frac{1}{1 + e^{-x}} \tag{19}$$

The output layer consists of a single neuron with a sigmoid activation function to produce an output in the range of 0 to 1. The output is rounded using a threshold of 0.5 to produce the binary classification output of 0 or 1. Relatively shallow and narrow FFNN structures are considered since the datasets are limited in size. Two to four hidden layers are considered with either 4, 6, or 8 neurons in each hidden layer. With respect to the training, a wide range of learning rates from 5×10^{-4} up to 1×10^{-2} are included in the hyperparameter search to account for the varying number of hidden layers and number of neurons. The maximum number of epochs for the training process was 500, with early stopping occurring if the loss function does not reduce by at least 1×10^{-4} over 10 epochs. Batch sizes of 64 and 128 were found to work well for training the range of hidden layers and neurons considered. A stochastic gradient-based optimizer ('adam') [131] was used as the solver for each FFNN with a binary cross-entropy loss function (Equation 4). Early attempts also used a quasi-Newton solver ('lbfgs') [132], but, it failed to convergence satisfactorily during the training process.

Table 5: Hyperparameters considered for FFNN models.

Solver	Maximum Epochs	Learning rate	Activation Function	Batch Size	Number of Neurons	Hidden Layers	Principal components
adam	500	0.0005	tanh	64	4	2	5
		0.0007	ReLU	128	6	3	10
		0.001			8	4	15

The hyperparameters considered for the XGBoost model attempt to balance the creation of complex trees in the boosted model to learn the behavior of the high and low performing permutations while concurrently including regularization parameters to prevent overfitting. This is done by considering

both smaller trees in the boosted model with less regularization, and larger trees with increased regularization. The maximum tree depths considered were between 6 and 15 and measured as the number of nodes along the longest path in each tree. Each tree added to the XGBoost model considers between 80% and 100% of the input features for creating node splits (*i.e.* 'Column Samples per Tree'). To control the growth of each tree and prevent excessive node splitting, the minimum child weight is set between 0.8 and 2. To allow for further control of the node splits, gamma, which controls the minimum reduction in the loss required to make a node split, is set to either 0 or 1. To prevent overfitting, the regularization term, lambda, takes a value of either 0, 1, or 2. Finally, to accommodate the set of boosted tree models which may be constructed with aforementioned hyperparameters, the learning rates considered range from 0.1 to 0.3.

Table 6: Hyperparameters considered for XGBoost models.

Learning rate	Minimum Child Weight	Gamma	Lambda	Maximum Depth	Column Samples per Tree	Principal Components
0.1	0.8	0	0	6	0.8	5
0.2	1	1	1	10	0.9	10
0.3	2		2	15	1	15

The dataset used for the 'minimum feed clearance' classifier includes the entire dataset for both 'I' and 'S' stretch-webs with the 'early failure' permutations labeled in Section 5.1.3 being removed. This results in dataset sizes of 461 and 466 for the 'I' and 'S' stretch-web respectively. The 'work-piece placement' classifier is used after the 'minimum feed clearance' classifier to predict the placement of the work-piece for permutations which were predicted to be within the 'minimum feed clearance'. However, a change to the tolerance of the 'minimum feed clearance' classifier would change the dataset used for the 'work-piece placement' classifier. This would then require every 'work-piece placement' classifier for each placement tolerance to be trained using the data for each dataset used for the 'minimum feed clearance' classifier. To avoid the multitude of models that would be required, the 'work-piece placement' classifiers are simply trained on the entire dataset for each stretch-web type using only the 'successful' permutations labeled in Section 5.1.3. The dataset sizes for the 'work-piece placement' classifier are 305 and 264 for the 'I' and 'S' stretch-web respectively.

In the following section (5.2.4), the results from nested cross-fold validation used to estimate the model performance are explored. Section 5.2.5 then identifies the best performing hyperparameters and trains several models to demonstrate the predictive capability. The final machine learning models are

trained using the best performing hyperparameters and are used to construct a look-up table to identify the best performing lifter and feeder trajectories (5.3).

5.2.4 Nested Cross-fold Validation

The results from the nested cross-fold validation for the ‘minimum feed clearance’ classifier are shown in Figure 66. In general the test accuracy varied between 80% and 95% depending on the tolerance used and the stretch-web type. The XGBoost model generally performed better than the FFNN on the ‘S’ stretch-web data, whereas both architectures performed similarly on the ‘I’ stretch-web data. All permutations with less than 0 mm of feed clearance result in collisions and all models are fairly capable at predicting permutations using this tolerance. However, referring back to Figure 61 which illustrates the ‘minimum feed clearance’ versus the stroke rate for each permutation, collisions are recorded up to a ‘minimum feed clearance’ of 5 mm. Furthermore, a ‘minimum feed clearance’ of 5 mm was predicted in the FE model over a wide range of stroke rates, making permutations with a ‘minimum feed clearance’ close to 5 mm more challenging to predict accurately.

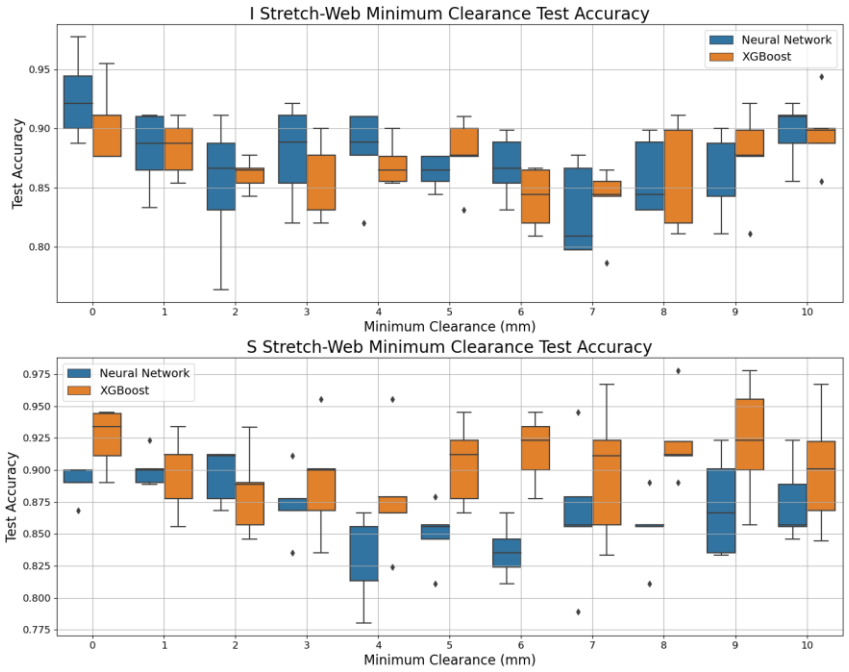

Figure 66: Test accuracy for the ‘minimum feed clearance’ classifiers on the outer loop of the nested cross-fold validation.

The results from the nested cross-fold validation for the ‘work-piece placement’ classifier are shown in Figure 67. The test accuracy for all models is fairly reasonable for both the lower and higher tolerances considered with accuracies ranging from 80% to upwards of 90%. However there is a slump in the accuracy for the intermediate ‘work-piece placement’ tolerances with accuracies below 65%. This wide range of test accuracy can be attributed to the class balance for the dataset associated with each tolerance. When the placement tolerance is close to 0.05 mm or 0.25 mm, nearly all samples are either within the tolerance or outside of it, allowing for the model to simply predict the majority class for every sample and achieve a high accuracy. In general, the classifiers trained on the ‘S’ stretch-web data produce a wider variance in the test accuracy as compared to the ‘I’ stretch-web classifiers. The higher variance in the test accuracy can be attributed to wider variance in the ‘work-piece placement’ of the ‘S’ stretch-web simulations, making the dataset more challenging to predict.

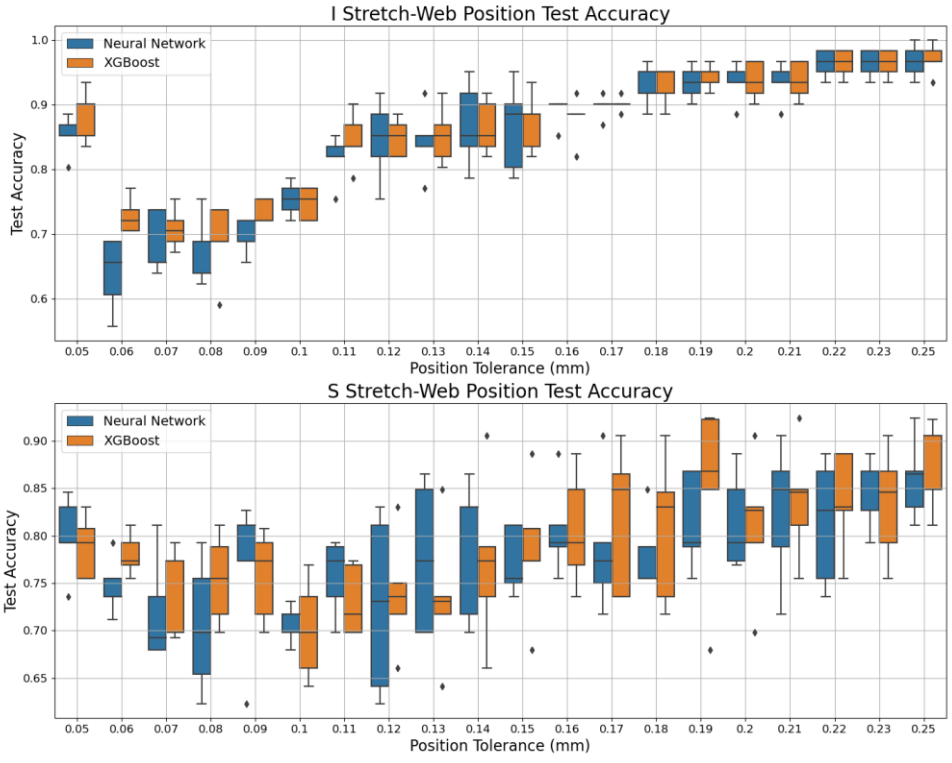

Figure 67: Test accuracy for the ‘work-piece placement’ classifiers on the outer loop of the nested cross-fold validation.

Comparing the results for the ‘minimum feed clearance’ and ‘work-piece placement’ classifiers, it is clear that predicting the ‘minimum feed clearance’ is less challenging than predicting the blank placement. This can in part be explained by the prediction of the ‘minimum feed clearance’ occurring earlier in the dynamic process. Since the dynamic response is highly non-linear, the effects of the non-linearity throughout the duration of the process accumulate, causing small changes in the process inputs to deviate by increasing amounts as the process continues. The challenge of predicting the ‘work-piece placement’ is further exacerbated by the smaller dataset size. The ‘work-piece placement’ classifier achieves a lower test accuracy for intermediate tolerances.

5.2.5 Final Model Selection

To identify suitable hyperparameters for the final machine learning models, 5-fold cross-validation is done on the entire datasets used for each classifier for the nested cross-fold validation done in Section 5.2.3 using the same hyperparameters previously considered. The hyperparameters which achieve the highest average accuracy over all 5-folds is then selected as the hyperparameters for the final machine learning model and are used to construct the look-up table in Section 5.3.

The tolerances considered for labeling the data were 10 mm for the ‘minimum feed clearance’ and 0.11 mm for the ‘work-piece placement’. These tolerances were selected to ensure ample clearance during the feeding phase while ensuring minimal placement error. Only the XGBoost architecture was considered for the final models since the accuracy was comparable to the FFNN and training times were shorter on a CPU. The final hyperparameters selected from the 5-fold cross-validation can be seen in Table 7.

Table 7: Selected Hyperparameters for the XGBoost model used to construct the look-up tables in Section 5.3.

								Principal Components	
Model	Web Type	Learning rate	Minimum Child Weight	Gamma	Lambda	Maximum Depth	Column Samples per Tree	Lifter	Feeder
Minimum Feed Clearance	I	0.3	0.8	0	0	10	0.9	5	5
	S	0.2	2	1	0	6	1	5	10
Work-Piece Placement	I	0.2	0.5	0	1	10	0.8	5	10
	S	0.2	2	0.5	0.5	10	1	15	5

To demonstrate the predictive capability of the hyperparameters selected for the ‘minimum feed clearance’ and ‘work-piece placement’ classifiers, XGBoost models are constructed using the hyperparameters shown in Table 7 for both stretch-webs. A total of four classifiers are constructed,

using a 60% of the data for training and 40% for testing. The ‘minimum feed clearance’ classifiers use the entire datasets for the ‘I’ and ‘S’ stretch-webs while the ‘work-piece placement’ classifiers consider only the permutations labeled as ‘Successful’ in Section 5.1.3 (*i.e.* there was no collision or strip misalignment).

Figure 68 shows the ‘minimum feed clearance’ vs stroke rate for each permutation in the considered dataset and is coloured by prediction of the respective classifier. A ‘True Negative’ (green) is a permutation which was correctly predicted to be within the considered tolerance whereas ‘True Positive’ (red) is a permutation which is correctly predicted to be outside the considered tolerance. A ‘False Negative’ (brown) is a permutation incorrectly predicted to be within tolerance, while a ‘False Positive’ (purple) is a permutation which is incorrectly predicted to be outside the tolerance. The classifiers for the ‘I’ and ‘S’ stretch-webs achieve a test accuracy of 89% and 93% respectively with the confusion matrices being shown in Figure 69. Both classifiers were able to generally predict permutations which follow the linear relationship between the ‘minimum feed clearance’ and the stroke rate. In the case of the classifier for the ‘S’ stretch-web, the higher test accuracy of can be attributed to the model’s capacity to correctly predict outlier permutations which were able to achieve the ‘minimum feed clearance’ tolerance of 10 mm at strokes rates above 200 SPM. For both classifiers, the permutations which resulted in ‘False Negatives’ had a smaller ‘minimum feed clearance’ at lower stroke rates as compare to the general trend, while ‘False Positives’ had a larger ‘minimum feed clearance’ at higher stroke rates. All ‘False Positive’ and ‘False Negative’ predictions for both classifiers tend to be located near the specified tolerance boundary of 10 mm, indicating both classifiers are fairly adept at classifying each permutation, but, struggle to create an accurate classification which aligns with the specified tolerance.

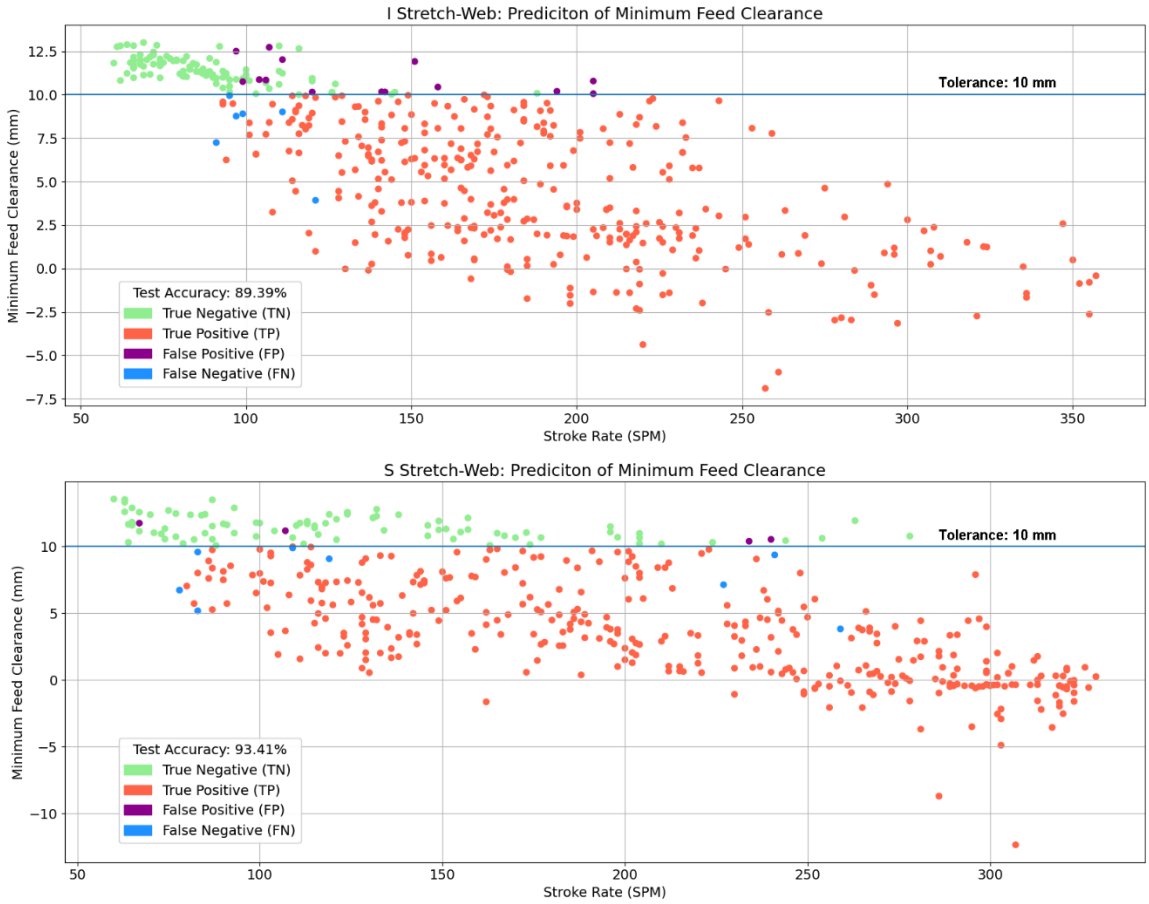

Figure 68: Plots of stroke rate *vs.* 'minimum feed clearance' for 'I' and 'S' stretch-webs with each permutation coloured by the prediction of the respective XGBoost classifier.

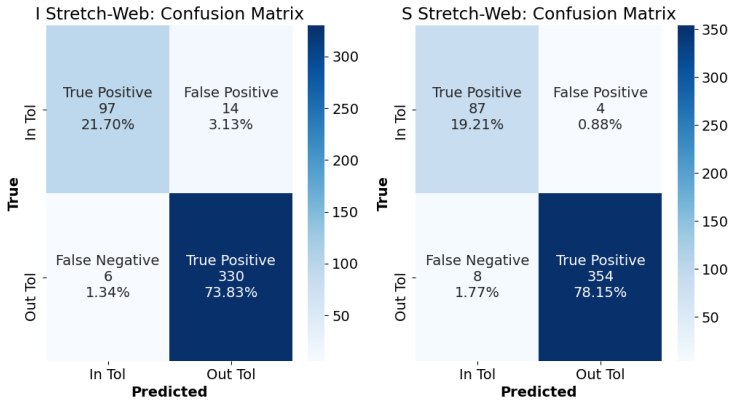

Figure 69: Confusion matrices for the 'minimum feed clearance' classifiers.

Figure 70 shows the ‘work-piece placement’ for each permutation in the considered datasets and are coloured by the prediction of the ‘work-piece placement’ classifier for each stretch-web. The ‘work-piece placement’ classifiers for the ‘I’ and ‘S’ stretch-web achieve a test accuracy of 83% and 75%, respectively with the confusion matrices being shown in Figure 71. The lower test accuracy of the ‘S’ stretch-web classifier can be attributed to the large variance in ‘work-piece placement’ which makes it more challenging to predict whether a permutation was within the specified placement tolerance of 0.11 mm. Unlike in Figure 68, the ‘False Positives’ and ‘False Negatives’ do not tend to be located near the tolerance boundary of 0.11 mm. This suggests that the misclassified permutations of the ‘work-piece placement’ classifier were not a result of being near the specified tolerance boundary, but, rather from a lack of predictive capability of the ‘work-piece placement’ classifier.

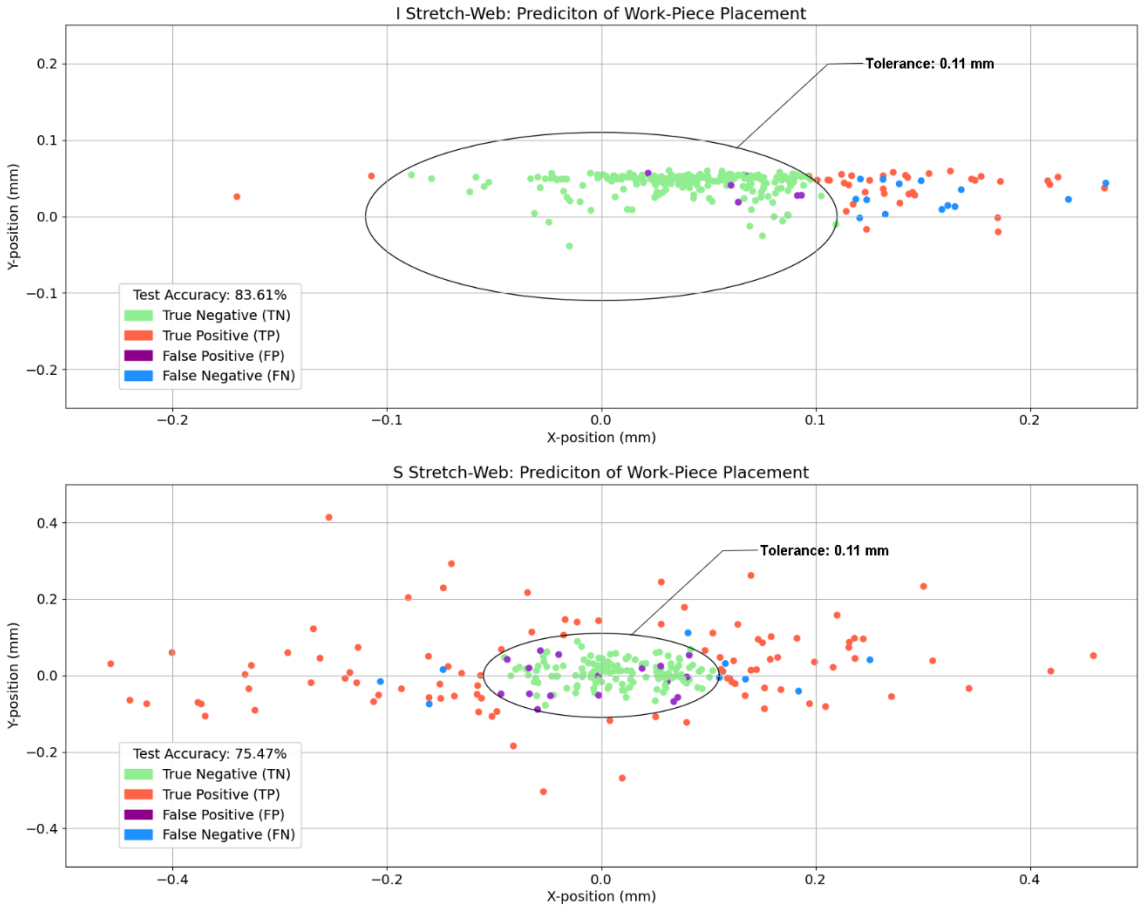

Figure 70: Plots of the ‘work-piece placement’ for ‘I’ and ‘S’ stretch-webs with each permutation coloured by the prediction of the respective XGBoost classifier.

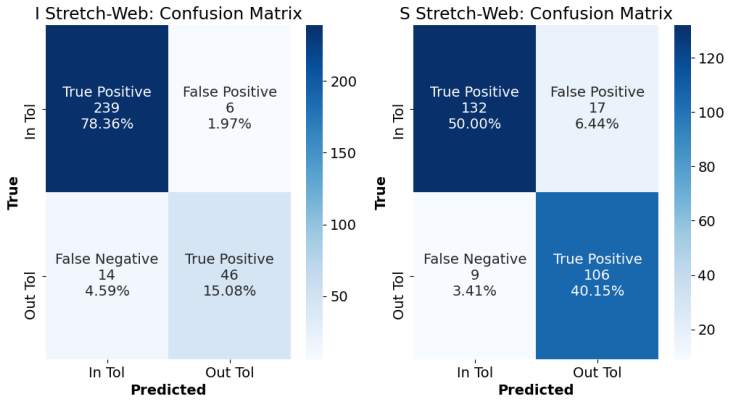

Figure 71: Confusion matrices for the 'work-piece placement' classifiers.

5.3 Look-up Table

The final machine learning models are created by training the 'minimum feed clearance' and 'work-piece placement' classifiers on all of the available data using the selected hyperparameters from Section 5.2.5. A look-up table is constructed by predicting the 'minimum feed clearance' and 'work-piece placement' for all possible pairs of trajectories at various stroke rates. The look-up table is then used to identify the maximum predicted stroke rate that a given pair of lifter and feeder trajectories can achieve while satisfying the tolerances.

The combination of all 363 lifter trajectories and 606 feeder trajectories generates a total of 22×10^5 trajectory pairs. Considering each of the trajectory pairs at stroke rates between 60 and 360 SPM generates a total of 66×10^6 permutations which are tested on the final machine learning models using the procedure illustrated in Figure 64. For each pair of the 22×10^5 trajectory pairs, the maximum predicted stroke rate that is achievable while satisfying the tolerance for the 'minimum feed clearance' and 'work-piece placement' is identified. If the stroke rate increases above the maximum predicted stroke rate, then the trajectory pair is predicted to not satisfy the tolerance requirements and be considered a 'failed' permutation as per Figure 64. Figure 72 shows histograms, for both stretch-web types, of the maximum predicted stroke rate of every possible trajectory pair. For both stretch-webs, there are several spikes in the histograms with no clear distribution of the maximum predicted stroke rate achieved amongst the possible trajectory pairs. The best performing trajectory pairs can be considered as those which achieve the highest stroke while satisfying the 'minimum feed clearance' and 'work-piece placement' tolerances (right side of Figure 64) while the worst performing trajectory

pairs are those which must operate at the lowest strokes to satisfy the tolerances (left side of Figure 64). The best performing trajectory pairs have a maximum predicted stroke rate of 205 SPM and 221 SPM for ‘I’ and ‘S’ stretch-webs respectively while the worst performing trajectory pairs have a maximum predicted stroke rate of 60 SPM for both stretch-webs.

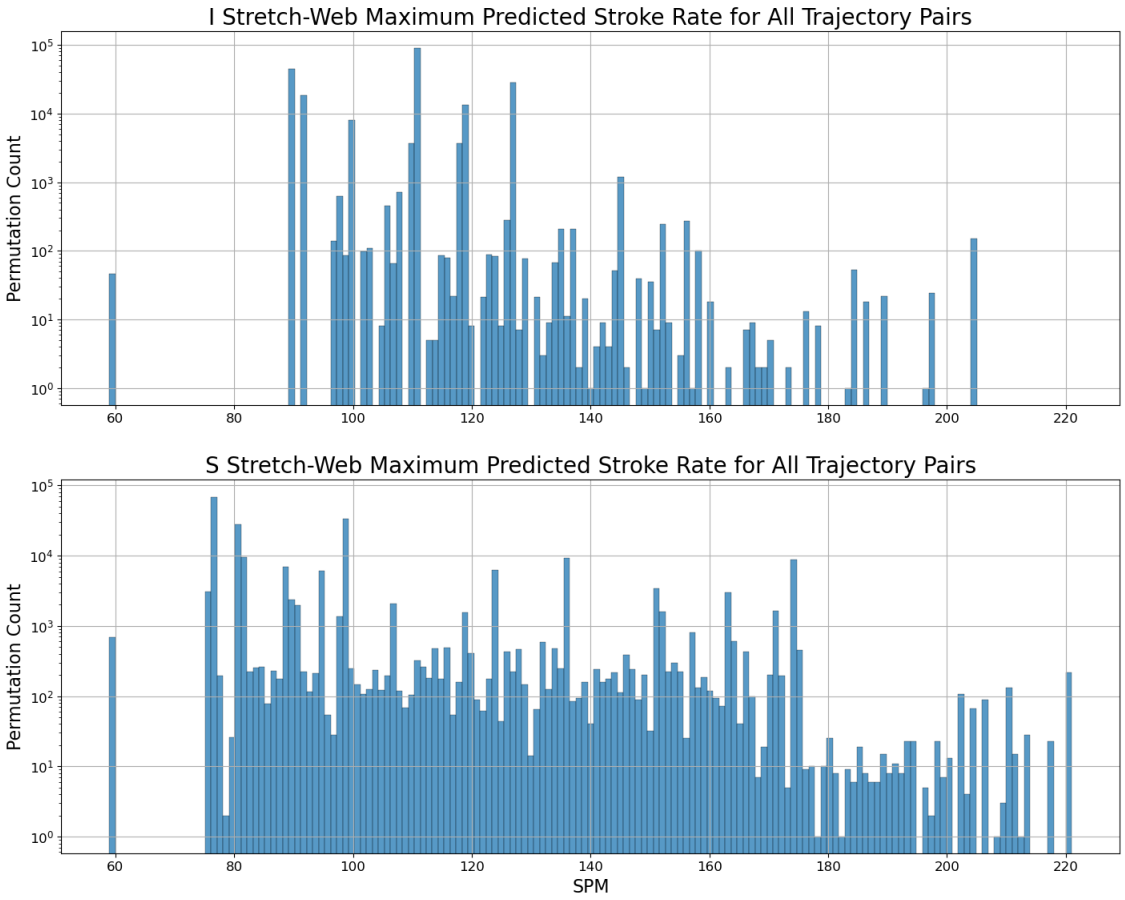

Figure 72: Histogram of maximum stroke rate for all possible trajectory pairs.

Figure 73 and Figure 74 plot the worst and best performing trajectories for the ‘I’ and ‘S’ stretch-webs respectively. For the ‘I’ stretch-web, the best performing lifter trajectories tended to follow a linear path with minimal acceleration and deceleration along with several trajectories which utilized a longer deceleration. The worst performing lifter trajectories for the ‘I’ stretch-web tended to have a rapid deceleration. With respect to the feeder trajectories for the ‘I’ stretch-web, the best performing ones tended to balance the acceleration and deceleration while the worst performing ones had either

significant acceleration or deceleration. The clear trends for the best and worst performing trajectories indicate that the combination of the two trajectories are important for the selection of suitable control inputs for the ‘I’ stretch-web.

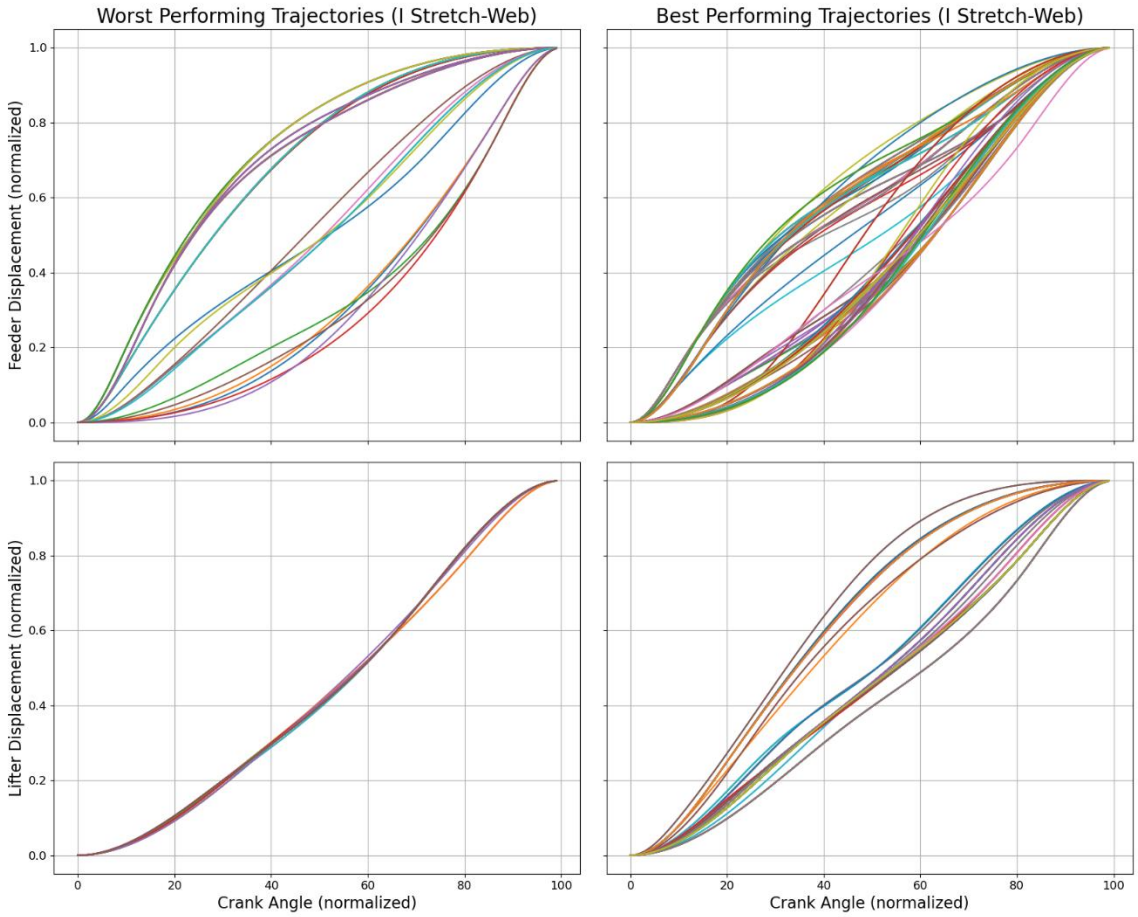

Figure 73: Best and worst performing feeder and lifter trajectories for the ‘I’ stretch-web.

Similar to the ‘I’ stretch-web, the best performing lifter trajectories for the ‘S’ stretch-web followed a linear path with minimal acceleration and deceleration whereas the worst performing trajectories followed an s-shaped curve. The best performing feeder trajectories for the ‘S’ stretch-web follow a sigmoid-shaped curve with a gradual acceleration and deceleration. In contrast, the worst performing feeder trajectories took a variety of shapes, indicating that the worst performing lifter trajectory would cause collisions between the strip and tooling, therefore making the selection of feeder trajectory arbitrary. In general, it appears that the selection of the lifter trajectory is of more importance for the

‘S’ stretch-web since fewer unique trajectories were grouped into the worst and best performing groups as compared to the feeder trajectories.

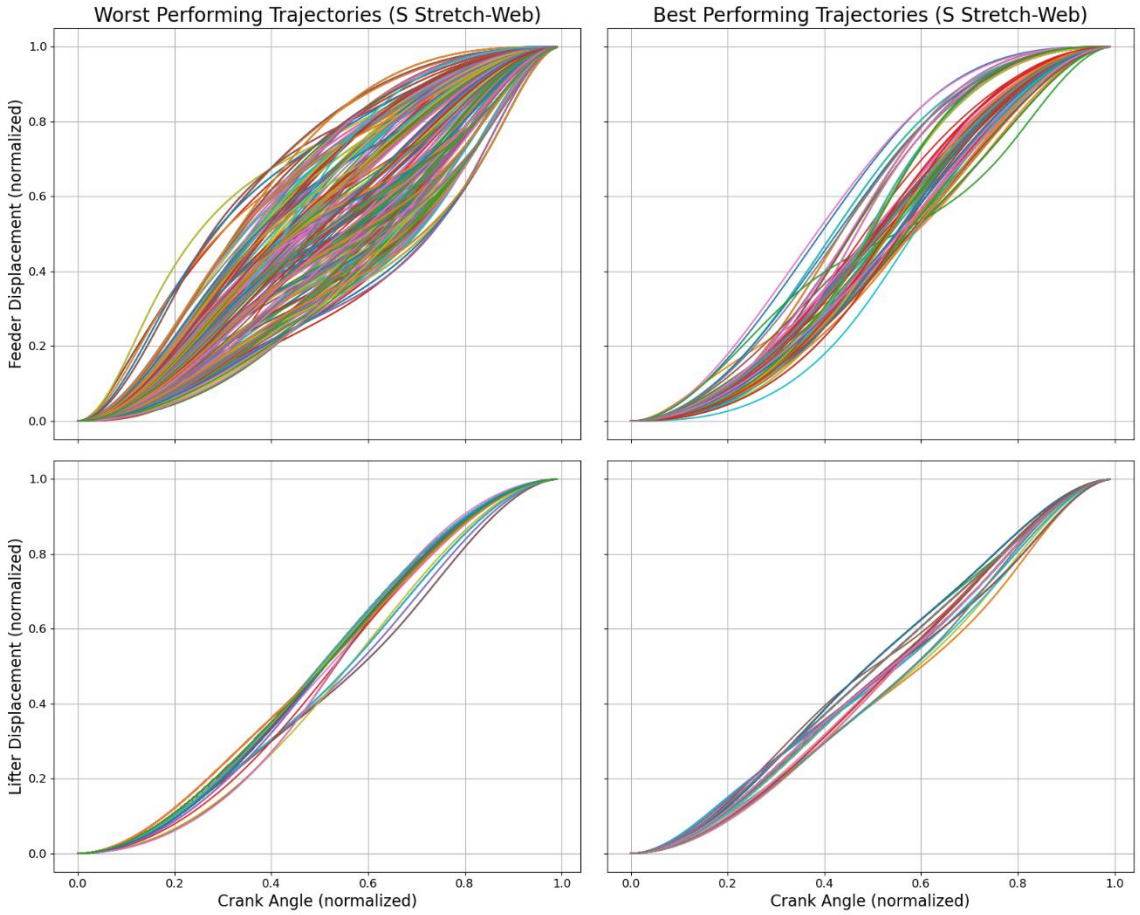

Figure 74: Best and worst performing feeder and lifter trajectories for the ‘S’ stretch-web.

From the results of the best and worst performing lifter trajectories for both stretch-webs, it appears that a linear trajectory with a slight acceleration and deceleration are the best performers. The higher performance of the linear trajectories may be rationalized by the fact that the work-pieces remain closer to the binder during the lifting phase than more s-shaped trajectories, preventing excessive oscillation while lifting and ultimately resulting in a large ‘minimum feed clearance’. While this linear trend is quite similar to the passive actuation used in conventional strip lifters, there are some aspects to note which may differentiate their performances. First, the best performing lifter trajectories include a slight

acceleration and deceleration which is not realized using passive strip lifters in current industrial practice. However, a damping component could be in introduced to passive strip lifters to introduce a more gradual acceleration and deceleration. Second, the active strip lifters are actuated over a longer duration of the crank angle as compared to the passive configuration, allowing for a lower peak velocity and additional clearance between the binder and strip at the nominal lifting height, reducing the risk of collision. The difference in the length of actuation also prevents a direct comparison between the trajectories considered in this work and a passive configuration. Finally, the best performing lifter trajectories identified may be specific to the strip layouts considered in this research. While the best performing strip lifter trajectories were similar for both stretch-webs, the best performing feeder trajectories were different, suggesting the feeder trajectory selection is dependent on stretch-webs used in the strip layout.

With all possible permutations predicted using the machine learning models, a look-up table has effectively been created which allows a user to identify the maximum stroke rate for any given trajectory pair. However, it would be illogical to randomly select control inputs for the feeder and lifter to use for the progressive die operation. Rather, the trajectories identified as the best performers in Figure 73 and Figure 74 should be used for any stroke rate up to their maximum as it can be reasoned that they would perform better than all other trajectories by providing additional clearance during the feeding phase while satisfying the requirements for the placement of the work-piece. To validate the predicted best performing trajectory pairs, an additional FE model are used to simulate the behavior of the progressive die operation for multiple strokes. Since there are multiple trajectories which are considered ‘best performing’, five were randomly sampled from the ‘I’ stretch-web results and simulated for five strokes. The sampled trajectories can be seen in Figure 75.

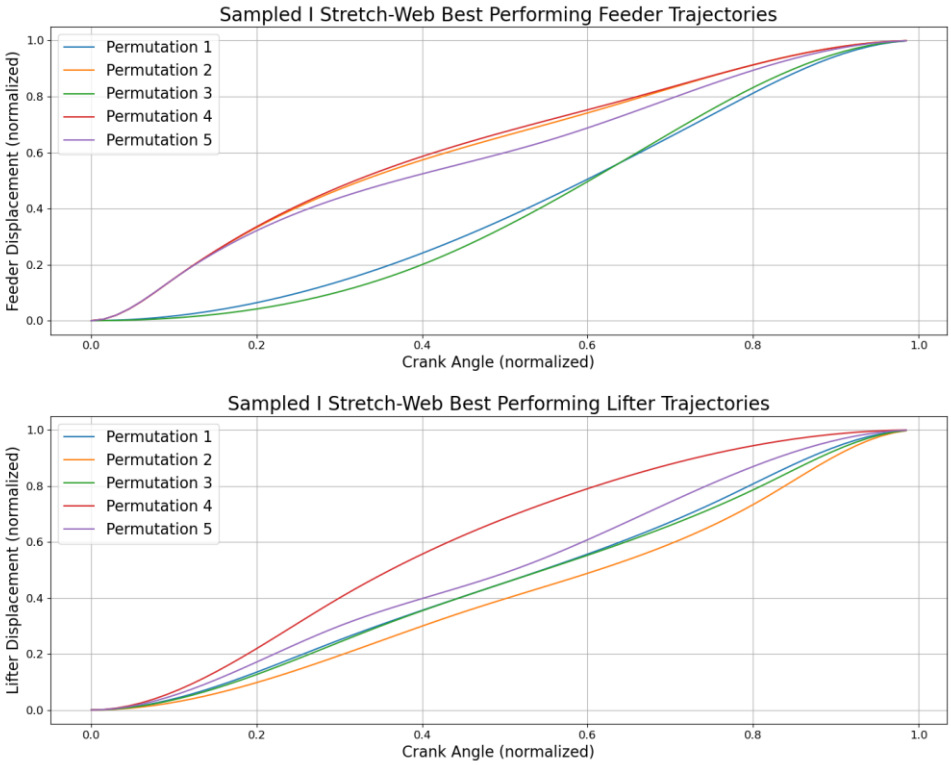

Figure 75: Sampled trajectories from the 'I' stretch-webs best performing permutations.

5.4 Extension to Multiple Strokes

While limiting the ‘Final’ model to a single press stroke reduced the computational cost of each simulation to allow for a dataset to be generated and train the machine learning models, it falls shorts of illustrating the behavior of the strip for multiple strokes. The dataset generated with the ‘Final’ model includes multiple permutations which operate at stroke rates above 200 SPM without collision or strip-misalignment; however, it is unclear whether these high stroke rate permutations can avoid a collision or a strip misalignment for multiple strokes. To evaluate the performance of the best performing trajectories taken from the look-up table in Section 5.3 (Figure 75), the ‘Final’ model is modified to consider multiple strokes using an ‘I’ stretch-web connector with a single strip carrier.

To simulate each selected permutation for five strokes, modifications to the strip mesh were made to allow for five work-pieces to be blanked and formed in the simulation. The difference in the mesh can be seen in Figure 76. With a new strip mesh, the extended simulations required a new simulation initialization which was created in the same manner as described in Section 4.6 by running the

simulation for a portion of a stroke at a low stroke rate (30 SPM) until the end of the forming operation. All five of the extended simulations start from this newly initialized simulation. The five selected permutations were simulated for five strokes at their maximum stroke rate (205 SPM) as predicted by the machine learning models.

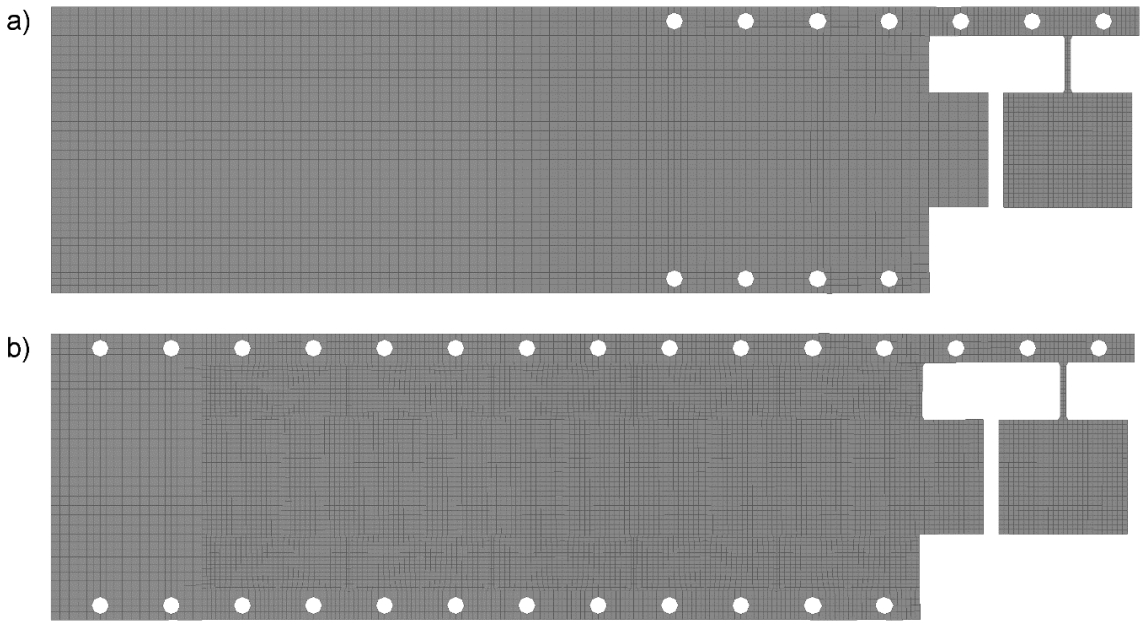

Figure 76: Comparison of strip mesh for the (a) 'Final' model and (b) extended version of the 'Final' model.

While none of the permutations resulted in a collision, strip misalignment occurred for all of them as early as the second stroke and as late as the fourth stroke with ‘Mis’ indicating the stroke at which strip misalignment occurred. For every extended permutation, the strip misalignment occurs in a nearly identical manner, with the four locating pins near the cropping operation at the end of the tooling unable to properly locate the pilot holes of the strip as seen in **Figure 77**. Unlike the misalignments observed in Section 5.1.3, which were largely a result of collisions, the strip misaligns as a result of the strip shifting away from the strip lifters during the strip progression. The occurrence of strip misalignments after multiple strokes highlights one of the challenges with generating the data from a single stroke of

the progressive die operation. While the single stroke approach allows for a larger dataset to be generated, it does not provide information on the long-term response of the strip for multiple strokes.

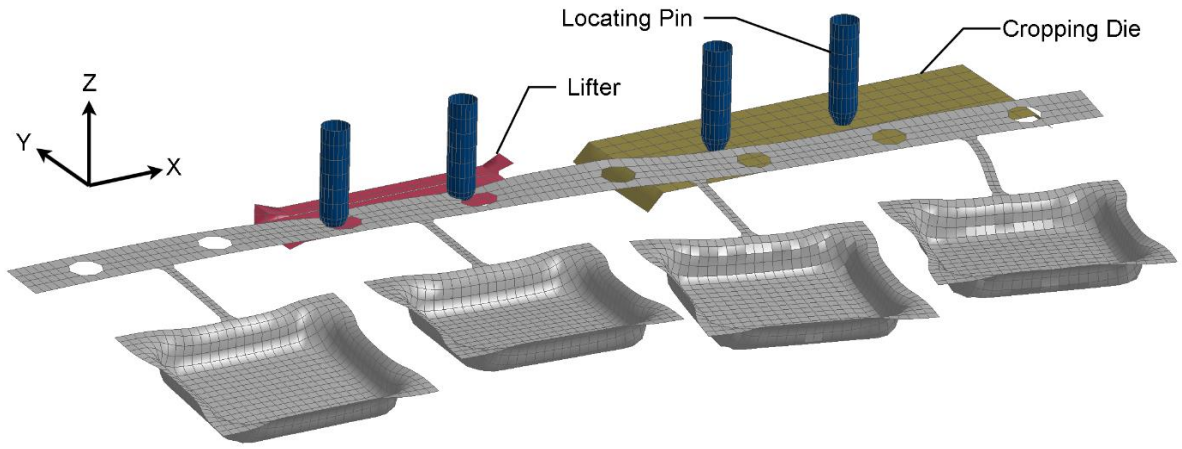

Figure 77: Example of strip misalignment in the extended FE simulations.

The ‘minimum feed clearance’ and ‘work-piece placement’ predictions from the extended (multi-stroke) FE models can be seen in Table 8 and Table 9 respectively. Nearly all permutations satisfied the 0.11 mm error for the ‘work-piece placement’ for the first stroke. While, none of the permutations met the requirement of a 10 mm feed clearance, most permutations had at least 8.5 mm of clearance between the strip and binder during the feeding phase of the first press stroke with the smallest ‘minimum feed clearance’ being 6.58 mm. Besides an incorrect prediction by the machine learning model, this difference in the predicted ‘minimum feed clearance’ by the FE models and the predicted clearance by the machine learning model may be attributed to the simulating the permutations at their maximum stroke rate at which they would satisfy the tolerances. Alternatively, the difference in the predicted outcomes of the simulations may be a side effect of altering the mesh of the strip for the extended simulations. By changing the mesh density, the stiffness and damping characteristics of the strip may have been altered which would in turn influence the dynamic response and inhibit the machine learning models ability to make accurate predictions. This variation in the predictions of the extended FE simulations and the predicted ‘minimum feed clearance’ and ‘work-piece placement’ by the machine learning models merits the use of a mesh sensitivity analysis in future work

Table 8: 'Minimum feed clearance' predicted by extended FE models five extended permutations simulated at 205 SPM using an 'I' stretch-web connector.

	Minimum Feed Clearance (mm)		
Permutation	Stroke 1	Stroke 2	Stroke 3
1	9.22	9.32	Mis
2	6.58	Mis	Mis
3	9.39	9.74	10.95
4	8.62	Mis	Mis
5	8.79	Mis	Mis

Table 9: 'Work-piece placement' error predicted by extended FE models five extended permutations simulated at 205 SPM using an 'I' stretch-web connector.

	Work-Piece Placement Error (mm)		
Permutation	Stroke 1	Stroke 2	Stroke 3
1	0.035	0.124	Mis
2	0.304	Mis	Mis
3	0.047	0.159	0.210
4	0.066	Mis	Mis
5	0.114	Mis	Mis

Since none of the selected permutations were able to complete five strokes and did not satisfy the tolerance requirements, additional simulations were completed using the same five trajectory pairs at lower stroke rates. The mean maximum stroke rate for all trajectory pairs of the 'I' stretch-web was predicted to be 108 SPM and was used for the additional simulations, with their results displayed in Table 10 and Table 11.

Table 10: 'Minimum feed clearance' predicted by extended FE models for five permutations simulated at 108 SPM using an 'I' stretch-web connector.

	Minimum Feed Clearance (mm)		
Permutation	Stroke 1	Stroke 2	Stroke 3
1	10.81	10.29	Mis
2	9.94	9.04	Mis
3	9.42	9.77	8.31
4	10.60	7.81	9.71
5	10.18	8.27	4.43

Table 11: 'Work-piece placement' error predicted by extended FE models for five permutations simulated at 205 SPM using an 'I' stretch-web connector.

	Work-Piece Placement Error (mm)		
Permutation	Stroke 1	Stroke 2	Stroke 3
1	0.106	0.075	Mis
2	0.099	0.04	Mis
3	0.015	0.059	0.281
4	0.067	0.082	0.121
5	0.127	0.054	0.251

Similar to the simulations completed at 205 SPM, all permutations resulted in a misalignment (**Figure 77**) as early as the third stroke and as late as the fourth stroke. The lower stroke rate ultimately improved the 'minimum feed clearance' and 'work-piece placement' error. For the first stroke, three of the permutations satisfied the 'minimum feed clearance' tolerance of 10 mm and the other two permutations were within 1 mm. Similarly, four of the permutations satisfied the 'work-piece placement' tolerance of 0.11 mm for the first stroke with the last permutation being close. All permutations were within the 'work-piece placement' tolerance for the second stroke. 'Permutation 1' satisfied both tolerances for two strokes, while 'Permutation 4' satisfied both tolerances for a single stroke.

Even after lowering stroke rate, the strip misaligned for every permutation implying the single carrier strip layout is inherently susceptible to misaligning. Since the tooling has large regions in which the work-piece is unsupported, the mass of the work-piece pulls the strip carrier away from the strip lifters during the feeding process when the pilot holes are not fixed by the locating pins. After several strokes,

the increasing slippage of the strip carrier off the strip lifters results in a misalignment. Anecdotally, at the time of writing this thesis, early experimental testing on the physical demonstrator tool at the Technische Universität München had similar issues with strip misalignment when using single carrier strip layouts. Simulating some of the best performing permutations for the ‘S’ stretch-web would likely result in a strip misalignment after several strokes as well since the strip layout has a larger variance in its dynamic response and generally poorer performance of the ‘work-piece placement’ as previously described in Section 5.1.4.

Chapter 6
Conclusions & Recommendations

6.1 Conclusions

This research has developed a methodology for identifying suitable stroke rates and control inputs to an active strip lifter and feeder for a progressive die operation. Several strip layouts and stretch-webs were considered in various sections of this work to study the influence of geometry on the dynamic response of the strip. Machine learning models were trained on data generated from FE models to predict the process outcome for a variety of different control inputs. Due to the high computational expense of the FE models, the machine learning data set was limited to the first forming stroke, although extension of the control inputs to multiple strokes was considered.

The conclusions that emerge from this research are as follows:

1. Numerous FE models were trialed for modeling progressive die operations, which utilize various techniques to reduce the computational cost of the simulations.
    a. A 'Detailed' model was created which incorporated a fine mesh over the entire strip, but, was constrained by the large run-times of the simulation.
    b. A 'Single Work-Piece' model was used to demonstrate the dynamic response of a single work-piece as it progresses through a progressive die operation for various strip layouts, but, did not account for the influence of multiple work-pieces.
    c. An 'Implicit' model used implicit time-integration to allow for larger time-steps in the FE model and showed promising results, but, was not pursued further due to the sensitivity of the implicit control settings and the uncertainty it presented.
    d. A 'Beam Element' model leveraged the computational efficiency of beam elements to achieve faster simulations times, but, exhibited non-physical behavior in the dynamic response of the strip.
    e. A 'Deformable-Rigid Switching' model was capable of simulating the entire strip for multiple press strokes by using the computational benefits of rigid body switching. However, the run-times of the model were not suitable for generating a dataset for training machine learning model.

2. Single and dual carrier strip layouts were considered in this work. The results from the various FE models indicate that a single carrier strip layout produces a larger dynamic response than a dual carrier strip layout.

3. By measuring the MSE of the dynamic response of the strip in multiple degrees of freedom, the convergence to a steady-state response can be observed in progressive die operations over multiple strokes.

4. The most computationally efficient FE model developed was the 'Final' model which has run times suitable for generating a dataset used for machine learning. The computational efficiencies of the 'Final' model were established by (i) limiting the simulation to considering only the first forming stroke; and (ii) only the strip lift and transfer portion of the process; and (iii) using the computational benefits of rigid body switching and lower through-thickness integration points for the shell elements of the strip.

5. To support the machine learning training, a total of 1000 permutations were simulated using the 'Final' FE model, 500 permutations each for both the 'I' and 'S' stretch-web strip layouts with a single carrier, by varying the stroke rate and lifter and feeder trajectories. The lifter and feeder trajectories were generated using cubic splines, with each permutation being sampled randomly from the space of considered trajectories. The stroke rate for each permutation was randomly sampled from the range of 60 to 360 SPM.

6. The results of the 1000 FE simulations indicate that an 'S' stretch-web produces a larger variance in the predicted dynamic response of the strip, including the 'work-piece placement' as compared to an 'I' stretch-web. The larger variance in the predicted dynamic response can be attributed to the additional compliance of the 'S' stretch-web Therefore, the 'S' stretch-web should only be used when the additional compliance is required during the forming operation. Furthermore, the lifter trajectory and stroke rate were shown to have a large influence on the 'minimum feed clearance' and probability of a collision incurring. All controls inputs influence the 'work-piece placement' and the predicted placement was highly variable.

7. The 'work-piece placement' on the forming die was established as a metric for assessing the acceptable limits of the dynamic response of the strip and evaluating the performance of a

progressive die operation. The ‘work-piece placement’ is conditional on whether a collision or strip misalignment occurred.

8. Two types of classifiers were constructed using FFNN and XGBoost architectures and trained using the 1000 simulations generated by the ‘Final’ FE model. The first type of classifier is used to predict the ‘minimum feed clearance’ for a given permutation to ensure there was not a collision. The second classifier predicts whether the ‘work-piece placement’ was within a specified tolerance. The two-stage prediction process assesses whether a given permutation is able to satisfy the ‘work-piece placement’ tolerances while providing a clearance during the feeing phase to avoid a collision. Using nested cross-fold validation, the ‘minimum feed clearance’ classifiers achieved accuracies ranging from 85% to 95% while the accuracy of the ‘work-piece placement’ classifiers varied from below 70% to upwards of 85% depending on the tolerance used to label the data. The better performance of the ‘minimum feed clearance’ classifier can be attributed to the prediction of the ‘minimum feed clearance’ occurring earlier in the highly non-linear process, making it easier to predict than the ‘work-piece placement’.

9. The XGBoost model was used to predict the ‘minimum feed clearance’ and ‘work-piece placement’ for all possible permutations of the control inputs to construct a look-up table for two strip layouts. ‘I’ and ‘S’ stretch-web connectors were considered using a single carrier strip layout. The pairs of lifter and feeder trajectories which could operate at the highest stroke rate while satisfying the ‘minimum feed clearance’ and ‘work-piece placement’ tolerances were observed to follow distinct paths. For both stretch-webs, the best performing lifter trajectories tended to follow a linear tool path with a slight acceleration and deceleration. For the ‘S’ stretch-web the best performing feeder trajectories followed and s-shaped tool path, while the best performing feeder trajectories for the ‘I’ stretch-web had either a gradual acceleration or deceleration.

10. Five of the best performing lifter and feeder trajectories for the ‘I’ stretch-web were simulated for up to five strokes at speeds of 205 SPM and 108 SPM. The XGBoost classifiers proved fairly capable at accurately predicting the process outcomes after a single stroke, but, failed due to the strip misalignments which occurred after several strokes in all simulations.

6.2 Recommendations

The following recommendations stem for the current research:

1. Experimental validation of the numerical finite element model should be conducted to ensure the machine learning models are learning patterns that are emblematic of the physical process. This includes monitoring the response of the strip in the experimental demonstrator tool using a high-speed camera and thorough characterization of the damping characteristics of the strip.
2. To ensure the predictions of the extended 'Final' FE model which simulates multiple press strokes produce results that agree with the single stroke 'Final' FE models, a mesh sensitivity analysis should be conducted which also considers the number of through thickness integration points. This study would ensure that the predictions from both FE models would agree for the first stroke.
3. To avoid the issue of strip misalignment, additional strip layouts should be considered which utilize two strip carriers, reducing the susceptibility of misalignment. Additional datasets should be generated using the 'Final' FE model with a dual carrier strip layout for retraining additional machine learning models.
4. Explore the use of variable actuation periods for the strip lifter and feeder. While this work considered a fixed crank angle period for the strip lifters and feeder with a variable trajectory, varying the actuation period while keeping the trajectory fixed could have a large influence on the dynamic response. Furthermore, some servo-driven strip feeders are capable of actuating over a variable crank angle with a fixed trajectory and therefore this approach may be more conducive to direct industrial application.
5. Investigate alternatives to cubic splines for generating the lifter and feeder trajectories. Cubic splines are sensitive to small changes to the knot location since the spline passes through each knot making the coordinates of each knot less reliable as a feature inputs for the machine learning models. Alternative spline formulations, such as B-splines, do not require the spline to pass through each knot and are therefore less sensitive to small changes in the knot position. Furthermore, by using uniformly spaced knots, the coordinates of each knot for a B-spline may be able to convey spatial information more effectively to the machine learning models.

6. Training of additional machine learning models should be conducted on a GPU to reduce the training time and permit a wider hyperparameter search.
7. The machine learning models could potentially see incremental improvement using the same data by training a single classifier to predict whether a given permutation was successfully within the 'minimum feed clearance' and placement tolerance rather than using two machine learning models in series.
8. To generate a large dataset to study the progressive die operation over multiple strokes, a computationally efficient alternative to FEA should be considered. Section 2.4 details some machine learning approaches which augment the use of FEA such as the use of CNNs [117]. Furthermore, work done by Sanchez-Gonzalez and Pfaff *et al*. [133], [134] has shown that graph neural networks may be a promising computationally efficient alternative to finite element modeling.
9. Alternative control strategies such as robust control may be considered which do not require the selection of pre-defined trajectories for the strip lifters and feeder and may be prototyped in a co-simulation, similar to Steinlehner *et al.* [135], prior to implementation in experimental tooling. This would approach would not require a dataset generate by the FE model or the use of machine learning.

References

[1] T. Trzepieciński, “Recent Developments and Trends in Sheet Metal Forming,” *Metals (Basel)*, vol. 10, no. 6, p. 779, Jun. 2020, doi: 10.3390/met10060779.

[2] V. Boljanovic, *Sheet Metal Forming Processes and Die Design*. New York: Industrial Press, 2004.

[3] Schuler GmbH, *Metal Forming Handbook*. Springer, 1998.

[4] T. J. Nye, “Stamping Strip Layout for Optimal Raw Material Utilization,” *J Manuf Syst*, vol. 19, no. 4, pp. 239–248, 200AD.

[5] D. Budnick *et al.*, “Simulation of Dynamic Effects in Progressive Die Operation and Control,” *IOP Conf Ser Mater Sci Eng*, vol. 1157, no. 1, p. 012085, Jun. 2021, doi: 10.1088/1757-899x/1157/1/012085.

[6] O. Böttcher, M. Gruber, M. Raupach, and W. Volk, “Anbindungsoptimierung von Folgeverbundbauteilen,” 2016.

[7] K.-J. Bathe, *Finite Element Procedures*, 2nd ed. Klaus-Jürgen Bathe, 2014.

[8] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[9] M. R. Jamli and N. M. Farid, “The sustainability of neural network applications within finite element analysis in sheet metal forming: A review,” *Measurement (Lond)*, vol. 138, pp. 446–460, May 2019, doi: 10.1016/j.measurement.2019.02.034.

[10] “LS-DYNA.” Livermore Software Technology, Livermore, California, 2019.

[11] R. Pearce, *4000 Years of Sheet Metal Forming*. ASTM International, 1982.

[12] P. Ulintz, “The History of Progressive Die Stamping,” *Bahrs Die & Stamping*, Dec. 29, 2015.

[13] M. Worswick and C. J. Butcher, “Proposal for the Development and Implementation of a Control Loop for Connections in the Progressive Die Process Development Grant,” *Natural Sciences and Engineering Research Council of Canada*. University of Waterloo, Waterloo, Jun. 22, 2018.

[14] T. Altan and A. E. Tekkaya, *Sheet Metal Forming Fundamentals*. ASM International, 2012.

[15] S. Kumar and R. Singh, “An Automated Design System for Progressive Die,” *Expert Syst Appl*, vol. 38, no. 4, pp. 4482–4489, Apr. 2011, doi: 10.1016/j.eswa.2010.09.121.

[16] M. Ghatrehnaby and B. Arezoo, “Automatic Strip Layout Design in Progressive Dies,” *J Intell Manuf*, vol. 23, no. 3, pp. 661–677, Jun. 2012, doi: 10.1007/s10845-010-0417-0.

[17] M. R. Duffey and Q. Sun, “Knowledge-Based Design of Progressive Stamping Dies,” *J Mater Process Technol*, vol. 28, pp. 221–227, 1991.

[18] Ivana. Suchy, *Handbook of Die Design*. McGraw-Hill, 2006.

[19] H. Xiang, X. Zhuang, and Z. Zhao, “Knowledge-based System for Strip Layout Design in Fineblanking,” in *Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2009, vol. 3, pp. 443–447. doi: 10.1109/ICICISYS.2009.5358146.

[20] S. Kumar and R. Singh, “Automation of Strip-layout Design for Sheet Metal Work on Progressive Die,” *J Mater Process Technol*, vol. 195, no. 1, pp. 94–100, 2008, doi: https://doi.org/10.1016/j.jmatprotec.2007.04.119.

[21] P. M. Horton, J. M. Allwood, and C. Cleaver, “Implementing Material Efficiency in Practice: A Case Study to Improve the Material Utilisation of Automotive Sheet Metal Components,” *Resour Conserv Recycl*, vol. 145, pp. 49–66, Jun. 2019, doi: 10.1016/j.resconrec.2019.02.012.

[22] P. M. Horton and J. M. Allwood, “Yield Improvement Opportunities for Manufacturing Automotive Sheet Metal Components,” *J Mater Process Technol*, vol. 249, pp. 78–88, Nov. 2017, doi: 10.1016/j.jmatprotec.2017.05.037.

[23] J. M. Allwood, J. M. Cullen, and R. L. Milford, “Options for Achieving a 50% Cut in Industrial Carbon Emissions by 2050,” *Environ Sci Technol*, vol. 44, no. 6, pp. 1888–1894, Mar. 2010, doi: 10.1021/es902909k.

[24] J. M. Cullen and J. M. Allwood, “Mapping the Global Flow of Aluminium: from Liquid Aluminium to End-use Goods,” *Environ Sci Technol*, vol. 47, no. 7, pp. 3057–3064, Jan. 2013.

[25] J. M. Cullen, J. M. Allwood, and M. D. Bambach, “Mapping the Global Flow of Steel: From Steelmaking to End-use Goods,” *Environ Sci Technol*, vol. 46, no. 24, pp. 13048–13055, Dec. 2012, doi: 10.1021/es302433p.

[26] N. Ramesha and A. Hiremath, “The Significant Variables that Affect Metal During Deep Drawing Process in Sheet Metal Work,” *International Journal of Engineering Research & Technology*, vol. 4, no. 5, pp. 1636–1639, May 2015.

[27] Y.-M. Huang and D.-K. Leur, “Effects of Process Variables on V-die Bending Process of Steel Sheet,” *Int J Mech Sci*, vol. 40, no. 7, pp. 631–650, 1998.

[28] M. S. Buang, S. A. Abdullah, and J. Saedon, “Effect of die and punch radius on springback of stainless steel sheet metal in the air V-die bending process,” *Journal of Mechanical Engineering and Sciences*, vol. 8, pp. 1322–1331, Jun. 2015, doi: 10.15282/jmes.8.2015.7.0129.

[29] A. Wifi, M. Younan, H. Gharib, A. S. Wifi, M. Younan, and A. Nassef, “Optimization of the Blank Holder Force in Cup Drawing,” *Journal of Achievements of Materials and Manufacturing Engineering*, vol. 18, no. 1, pp. 291–294, Sep. 2006.

[30] J. F. Siekirk, “Process Variable Effects on Sheet Metal Quality,” *Journal of Applied Metalworking*, vol. 4, no. 3, pp. 262–269, Jul. 1986.

[31] A. Wifi and A. Mosallam, “Some Aspects of Blank-holder Force Schemes in Deep Drawing Process,” *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 24, no. 1, pp. 315–323, Sep. 2007.

[32] W. C. Emmens, *Formability: A Review of Parameters and Processes that Control, Limit or Enhance the Formability of Sheet Metal*. Springer, 2011.

[33] D. Banabic *et al.*, *Sheet Metal Forming Processes: Constitutive Modelling and Numerical Simulation*. Springer Science & Business Media, 2010. doi: 10.1007/978-3-540-88113-1.

[34] L. Ma and Z. Wang, “The Effects of Through-thickness Shear Stress on the Formability of Sheet Metal–A review,” *J Manuf Process*, vol. 71, pp. 269–289, Sep. 2021, doi: 10.1016/j.jmapro.2021.09.019.

[35] C. v. Nielsen and P. A. F. Martins, *Metal Forming: Formability, Simulation, and Tool Design*. Academic Press, 2021.

[36] S. P. Keeler, “Circular Grid System - A Valuable Aid for Evaluating Sheet Metal Formability,” *SAE Transactions*, pp. 371–379, 1968.

[37] G. M. Goodwin, “Application of Strain Analysis to Sheet Metal Forming Problems in the Press Shop,” *SAE Transactions*, pp. 380–387, 1968.

[38] Z. Marciniak and K. Kuczyński, “Limit Strains in the Processes of Stretch-Forming Sheet Metal,” *Int J Mech Sci*, vol. 9, no. 9, pp. 609–620, Feb. 1967.

[39] Koe. Nakazima, T. Kikuma, and K. Hasuka, “Study on the Formability of Steel Sheets,” Technical Research Institute, Yawata Works, Yawata, Japan, Sep. 1968.

[40] “ISO 12004-2 Metallic materials - Sheet and Strip - Determination of Forming-Limit Curves - Part 2: Determination of Forming-Limit Curves in the Laboratory,” Oct. 2008.

[41] J. v Laukonis and A. K. Ghosh, “Effects of Strain Path Changes on the Formability of Sheet Metals,” *Metallurgical Transactions A*, vol. 9, no. 12, pp. 1849–1856, Dec. 1978.

[42] T. B. Stoughton and X. Zhu, “Review of theoretical models of the strain-based FLD and their relevance to the stress-based FLD,” *International Journal of Plasticity*, vol. 20, no. 8–9. pp. 1463–1486, Aug. 2004. doi: 10.1016/j.ijplas.2003.11.004.

[43] H. J. Kleemola and H. J. Pelkkikangas, “Effect of Predeformation and Strain Path on the Forming Limits of Steel, Copper and Brass,” *Sheet Metal Industries*, vol. 64, pp. 591–599, Jun. 1977.

[44] R. Arrieux, Boivin M, and F. le Maître, “Determination of the Forming Limit Stress Curve for Anisotropic Sheets,” *CIRP Annals*, vol. 36, no. 1, pp. 195–198, 1987.

[45] T. B. Stoughton, “A General Forming Limit Criterion for Sheet Metal Forming,” *Int J Mech Sci*, vol. 42, no. 1, pp. 1–27, 2000.

[46] R. Zhang, Z. Shao, and J. Lin, “A Review on Modelling Techniques for Formability Prediction of Sheet Metal Forming,” *International Journal of Lightweight Materials and Manufacture*, vol. 1, no. 3, pp. 115–125, Sep. 2018, doi: 10.1016/j.ijlmm.2018.06.003.

[47] H. W. Swift, “Plastic Instability Under Plane Stress,” *J Mech Phys Solids*, vol. 1, pp. 1–18, May 1952.

[48] R. Hill, “On Discontinuous Plastic States, with Special Reference to Localized Necking in Thin Sheets,” *J Mech Phys Solids*, vol. 1, pp. 19–30, May 1952.

[49] L. Ellard, “Stamtec: Mechanical vs. Servo vs. Hydraulic Presses,” Manchester, 2013.

[50] M. G. Lee, C. Kim, E. J. Pavlina, and F. Barlat, “Advances in Sheet Forming-materials Modeling, Numerical Simulation, and Press Technologies,” *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 133, no. 6, Dec. 2011, doi: 10.1115/1.4005117.

[51] A. del Prete and T. Primo, “Sheet Metal Forming Optimization Methodology for Servo Press Process Control Improvement,” *Metals (Basel)*, vol. 10, no. 2, Feb. 2020, doi: 10.3390/met10020271.

[52] B. A. Behrens, R. Krimm, and S. Teichrib, “Contactless Feeder for Electrically Conductive Sheet Metals,” *Production Engineering*, vol. 11, no. 1, pp. 1–8, Dec. 2017, doi: 10.1007/s11740-016-0688-1.

[53] M. Kolbe, “High-Performance Stamping Presses,” in *Stamping Practice*, Springer, 2022, pp. 77–96. doi: 10.1007/978-3-658-34758-1_8.

[54] B. A. Behrens, A. Bouguecha, R. Krimm, S. Teichrib, and T. Nitschke, “Energy-efficient Drive Concepts in Metal-forming Production,” in *Procedia CIRP*, 2016, vol. 50, pp. 707–712. doi: 10.1016/j.procir.2016.05.039.

[55] “Bruderer Feed Units,” Sep. 2014.

[56] J. A. Polyblank, J. M. Allwood, and S. R. Duncan, “Closed-loop Control of Product Properties in Metal Forming: A Review and Prospectus,” *J Mater Process Technol*, vol. 214, no. 11, pp. 2333–2348, 2014, doi: 10.1016/j.jmatprotec.2014.04.014.

[57] D. E. Hardt, “Modeling and Control of Manufacturing Processes: Getting More Involved,” *J Dyn Syst Meas Control*, vol. 115, pp. 291–300, 1993, [Online]. Available: http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/115/2B/291/5546408/291_1.pdf

[58] J. M. Allwood *et al.*, “Closed-loop Control of Product Properties in Metal Forming,” *CIRP Ann Manuf Technol*, vol. 65, no. 2, pp. 573–596, 2016, doi: 10.1016/j.cirp.2016.06.002.

[59] Y. Lim, R. Venugopal, and A. G. Ulsoy, “Advances in the Control of Sheet Metal Forming,” in *Proceedings of the 17th International Federation of Automatic Control*, Jun. 2008, vol. 17, no. 1 PART 1, pp. 1875–1883.

[60] Y. Lim, R. Venugopal, and A. G. Ulsoy, *Process Control for Sheet-Metal Stamping: Process Modeling, Controller Design and Shop-Floor Implementation*. Springer, 2013.

[61] B. Endelt, S. Tommerup, and J. Danckert, “A Novel Feedback Control System - Controlling the Material Flow in Deep Drawing using Distributed Blank-Holder Force,” *J Mater Process Technol*, vol. 213, no. 1, pp. 36–50, 2013, doi: 10.1016/j.jmatprotec.2012.08.003.

[62] P. Fischer, D. Harsch, J. Heingärtner, Y. Renkci, and P. Hora, “A Knowledge-based Control System for the Robust Manufacturing of Deep Drawn Parts,” in *International Conference on the Technology of Plasticity*, Sep. 2017, vol. 207, pp. 42–47. doi: 10.1016/j.proeng.2017.10.735.

[63] P. Fischer, J. Heingärtner, Y. Renkci, and P. Hora, “Experiences with Inline Feedback Control and Data Acquisition in Deep Drawing,” in *17th International Conference on Metal Forming*, Sep. 2018, vol. 15, pp. 949–954. doi: 10.1016/j.promfg.2018.07.401.

[64] K. Siegert, M. Ziegler, and S. Wagner, “Closed Loop Control of the Friction Force. Deep Drawing Process,” *J Mater Process Technol*, vol. 71, pp. 126–133, 1997.

[65] N. Mahayotsanun, S. Sah, J. Cao, M. Peshkin, R. X. Gao, and C. tao Wang, “Tooling-integrated sensing systems for stamping process monitoring,” *Int J Mach Tools Manuf*, vol. 49, no. 7–8, pp. 634–644, Jun. 2009, doi: 10.1016/j.ijmachtools.2009.01.009.

[66] R. Michael Neugebauer *et al.*, “Sensitivity Analysis of Eddy Current Sensors Using Computational Simulation,” in *Progress In Electromagnetics Research Symposium Proceedings*, Sep. 2011, pp. 440–445.

[67] T. Yagami, K. I. Manabe, M. Yang, and H. Koyama, “Intelligent Sheet Stamping Process using Segment Blankholder Modules,” *J Mater Process Technol*, vol. 155–156, no. 1–3, pp. 2099–2105, Nov. 2004, doi: 10.1016/j.jmatprotec.2004.04.144.

[68] T. Bäume, W. Zorn, W. G. Drossel, and G. Rupp, “Iterative Process Control and Sensor Evaluation for Deep Drawing Tools with Integrated Piezoelectric Actuators,” *Manuf Rev (Les Ulis)*, vol. 3, 2016, doi: 10.1051/mfreview/2016002.

[69] K. Siegert, M. H. Èussermann, D. Haller, S. Wagner, and M. Ziegler, “Tendencies in Presses and Dies for Sheet Metal Forming Processes,” *J Mater Process Technol*, vol. 98, no. 2, pp. 259–264, 2000.

[70] J. Cao, B. Kinsey, and S. A. Solla, “Consistent and Minimal Springback Using a Stepped Binder Force Trajectory and Neural Network Control,” *J Eng Mater Technol*, vol. 112, no. 1, pp. 113–118, Jan. 2000, [Online]. Available: http://asmedigitalcollection.asme.org/materialstechnology/article-pdf/122/1/113/5746173/113_1.pdf

[71] P. Sun, J. J. Grácio, and J. A. Ferreira, “Control system of a Mini Hydraulic Press for Evaluating Springback in Sheet Metal Forming,” *J Mater Process Technol*, vol. 176, no. 1–3, pp. 55–61, 2006, doi: 10.1016/j.jmatprotec.2006.02.009.

[72] V. Viswanathan, B. Kinsey, and J. Cao, “Experimental Implementation of Neural Network Springback Control for Sheet Metal Forming,” *J Eng Mater Technol*, vol. 125, no. 2, pp. 141–147, Apr. 2003, doi: 10.1115/1.1555652.

[73] M. L. Bohn, S. G. Xu, K. J. Weinmann, C. C. Chen, and A. Chandra, “Improving formability in Sheet Metal Stamping with Active Drawbead Technology,” *J Eng Mater Technol*, vol. 123, no. 4, pp. 504–510, Oct. 2001, doi: 10.1115/1.1395577.

[74] J. Cao and M. C. Boyce, “Draw Bead Penetration as a Control Element of Material Flow,” *SAE Technical Paper*, no. 930517, 1993.

[75] Y. Tamai, Y. Yamasaki, A. Yoshitake, and T. Imura, “Improvement of Formability in Stamping of Steel Sheets by Motion Control of Servo Press,” *Steel Res Int*, vol. 81, no. 9, pp. 686–689, 2010.

[76] T. Maeno, K. Osakada, and K. Mori, “Reduction of Friction in Compression of Plates by Load Pulsation,” *Int J Mach Tools Manuf*, vol. 51, no. 7–8, pp. 612–617, Jul. 2011, doi: 10.1016/j.ijmachtools.2011.03.007.

[77] T. Nakano, “Press Machine Trends and Servo Press Forming Examples,” *Steel Res Int*, vol. 81, no. 9, pp. 682–685, 2010.

[78] K. Osakada, “Application of Servo Presses to Metal Forming Processes,” *Steel Res Int*, vol. 81, no. 9, pp. 9–16, 2010.

[79] K. Junlapen, K. Mizuochi, and N. Koga, “Reduction in Blanking Noise using NC Servo Press Machine,” *Journal of the Japan Society for Technology of Plasticity*, vol. 51, no. 591, pp. 353–357, 2010.

[80] X. Liao and G. G. Wang, “Evolutionary Path Planning for Robot Assisted Part Handling in Sheet Metal Bending,” *Robot Comput Integr Manuf*, vol. 19, no. 5, pp. 425–430, Oct. 2003, doi: 10.1016/S0736-5845(03)00053-X.

[81] G. Han and B. Eng, “Transfer Die System Timing and Parameter Optimization According to an Obstacle Map by,” 2016.

[82] E. Glorieux, “Multi-Robot Motion Planning Optimization for Handling Sheet Metal Parts,” PhD Thesis, University West, Trollhättan, 2017.

[83] H. F. Li, D. Ceglarek, and J. Shi, “A Dexterous Part-Holding Model for Handling Compliant Sheet Metal Parts,” *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 124, no. 1, pp. 109–118, Feb. 2002, doi: 10.1115/1.1406953.

[84] A. L. Fradkov, “Early History of Machine Learning,” in *International Federation of Automatic Control*, 2020, vol. 53, no. 2, pp. 1385–1390. doi: 10.1016/j.ifacol.2020.12.1888.

[85] J. S. Gero and F. Sudweeks, *Artificial Intelligence in Design ’96*. Springer, 1996. doi: 10.1007/978-94-009-0279-4.

[86] T. Hastie, R. Tibshirani, and J. Friedman, *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[87] D. Silver *et al.*, “Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[88] M. Jordan, J. Kleinberg, and B. Schölkopf, *Pattern Recognition and Machine Learning*, 4th ed., vol. 4. Springer, 2006.

[89] O. I. Abiodun *et al.*, “Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition,” *IEEE Access*, vol. 7, pp. 158820–158846, Oct. 2019, doi: 10.1109/ACCESS.2019.2945545.

[90] W. S. McCulloch and W. H. Pitts, “A logical Calculus of the Ideas Immanent in Nervous Activity,” *The Bulletin of Mathematical Biophysics* , vol. 5, no. 4, pp. 115–133, 1943.

[91] B. Widrow and M. E. Hoff, “Adaptive Switching Circuits,” Jun. 1960.

[92] K. Hornik, “Approximation Capabilities of Multilayer Feedforward Networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, Oct. 1991, doi: 10.1016/0893-6080(91)90009-T.

[93] G. E. Hinton and S. Osindero, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Comput*, vol. 18, no. 7, pp. 1527–1554, 2006.

[94] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An Introductory Review of Deep Learning for Prediction Models With Big Data,” *Front Artif Intell*, vol. 3, no. 4, Feb. 2020, doi: 10.3389/frai.2020.00004.

[95] F. Rosenblatt, “The Perceptron, A Perceiving and Recognizing Automaton,” Buffalo, Jan. 1957.

[96] A. R. Webb and K. D. Copsey, *Statistical Pattern Recognition*, 3rd ed. Malvern: John Wiley & Sons, 2011. [Online]. Available: www.wiley.com/go/statistical_pattern_recognition

[97] L. Canete-Sifuentes, R. Monroy, and M. A. Medina-Perez, “A Review and Experimental Comparison of Multivariate Decision Trees,” *IEEE Access*, vol. 9, pp. 110451–110479, 2021, doi: 10.1109/ACCESS.2021.3102239.

[98] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[99] S. Kashid and S. Kumar, “Applications of Artificial Neural Network to Sheet Metal Work - A Review,” *American Journal of Intelligent Systems*, vol. 2, no. 7, pp. 168–176, Jan. 2012, doi: 10.5923/j.ajis.20120207.03.

[100] R. Srinivasan, D. Vasudevan, and P. Padmanabhan, “Prediction of Spring-back and Bend Force in Air Bending of Electro-galvanized Steel Sheets using Artificial Neural Networks,” *Australian Journal of Mechanical Engineering*, vol. 12, no. 1, pp. 25–37, Jan. 2014, doi: 10.7158/M12-073.2014.12.1.

[101] S. K. Panthi, S. Hora, and M. Ahmed, “Artificial Neural Network and Experimental Study of Effect of Velocity on Springback in Straight Flanging Process,” *Indian Journal of Engineering & Materials Sciences*, vol. 23, pp. 159–164, 2016.

[102] R. Teimouri, H. Baseri, B. Rahmani, and M. Bakhshi-Jooybari, “Modeling and Optimization of Spring-back in Bending Process using Multiple Regression Analysis and Neural Computation,” *International Journal of Material Forming*, vol. 7, no. 2, pp. 167–178, 2014, doi: 10.1007/s12289-012-1117-4.

[103] W. Liu, Q. Liu, F. Ruan, Z. Liang, and H. Qiu, “Springback prediction for Sheet Metal Forming based on GA-ANN Technology,” *J Mater Process Technol*, vol. 187, pp. 227–231, Jun. 2007, doi: 10.1016/j.jmatprotec.2006.11.087.

[104] M. Bozdemir and M. Gölcü, “Artificial Neural Network Analysis of Springback in V Bending,” *Journal of Applied Sciences*, vol. 8, no. 17, pp. 3038–3043, 2008, doi: 10.3923/jas.2008.3038.3043.

[105] S. Toros and F. Ozturk, “Flow Curve Prediction of Al-Mg Alloys under Warm Forming Conditions at Various Strain Rates by ANN,” *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1891–1898, Mar. 2011, doi: 10.1016/j.asoc.2010.06.004.

[106] A. Forcellese, F. Gabrielli, and M. Simoncini, “Prediction of Flow Curves and Forming Limit Curves of Mg Alloy Thin Sheets using ANN-based Models,” *Comput Mater Sci*, vol. 50, no. 11, pp. 3184–3197, Oct. 2011, doi: 10.1016/j.commatsci.2011.05.048.

[107] M. el Mehtedi, A. Forcellese, L. Greco, M. Pieralisi, and M. Simoncini, “Flow Curve Prediction of ZAM100 Magnesium Alloy Sheets using Artificial Neural Network-based Models,” in *12th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, 2019, vol. 79, pp. 661–666. doi: 10.1016/j.procir.2019.02.050.

[108] Y. C. Lin, J. Zhang, and J. Zhong, “Application of Neural Networks to Predict the Elevated Temperature Flow Behavior of a Low Alloy Steel,” *Comput Mater Sci*, vol. 43, no. 4, pp. 752–758, Oct. 2008, doi: 10.1016/j.commatsci.2008.01.039.

[109] M. Manoochehri and F. Kolahan, “Integration of Artificial Neural Network and Simulated Annealing Algorithm to Optimize Deep Drawing Process,” *International Journal of Advanced Manufacturing Technology*, vol. 73, no. 1–4, pp. 241–249, 2014, doi: 10.1007/s00170-014-5788-5.

[110] A. Chamekh, H. Bel Hadj Salah, and R. Hambli, “Inverse Technique Identification of Material Parameters using Finite Element and Neural Network Computation,” *International Journal of Advanced Manufacturing Technology*, vol. 44, no. 1, pp. 173–179, Sep. 2009, doi: 10.1007/s00170-008-1809-6.

[111] R. Hill, “A Theory of the Yielding and Plastic Flow of Anisotropic Metals,” in *Royal Society of London. Series A. Mathematical and Physical Sciences*, 1948, vol. 67, pp. 281–297. [Online]. Available: https://royalsocietypublishing.org/

[112] F. Abbassi, T. Belhadj, S. Mistou, and A. Zghal, “Parameter Identification of a Mechanical Ductile Damage using Artificial Neural Networks in Sheet Metal Forming,” *Mater Des*, vol. 45, pp. 605–615, Mar. 2013, doi: 10.1016/j.matdes.2012.09.032.

[113] H. Aguir, A. Chamekh, H. BelHadjSalah, A. Dogui, and R. Hambli, “Identification of Constitutive Parameters using Hybrid ANN Multi-objective Optimization Procedure,” *International Journal of Material Forming*, vol. 1, no. 1, pp. 1–4, Jul. 2008, doi: 10.1007/s12289-008-0008-1.

[114] A. Hosseini and M. Kadkhodayan, “A Hybrid NN-FE Approach to Adjust Blank Holder Gap over Punch Stroke in Deep Drawing Process,” *International Journal of Advanced Manufacturing Technology*, vol. 71, no. 1–4, pp. 337–355, Mar. 2014, doi: 10.1007/s00170-013-5479-7.

[115] C. Bonatti and D. Mohr, “One for all: Universal Material Model based on Minimal State-space Neural Networks,” *Sci Adv*, vol. 7, no. 6, Jun. 2021, [Online]. Available: https://www.science.org

[116] C. Bonatti and D. Mohr, “On the Importance of Self-consistency in Recurrent Neural Network Models Representing Elasto-plastic Solids,” *J Mech Phys Solids*, vol. 158, Jan. 2022, doi: 10.1016/j.jmps.2021.104697.

[117] C. P. Kohar, L. Greve, T. K. Eller, D. S. Connolly, and K. Inal, “A Machine Learning Framework for Accelerating the Design Process using CAE Simulations: An Application to Finite Element Analysis in Structural Crashworthiness,” *Comput Methods Appl Mech Eng*, vol. 385, Nov. 2021, doi: 10.1016/j.cma.2021.114008.

[118] “Digital Twin: Manufacturing Excellence through Virtual Factory Replication.” [Online]. Available: https://www.researchgate.net/publication/275211047

[119] “Transdisciplinary Perspectives on Complex Systems New Findings and Approaches.”

[120] W. Volk *et al.*, “Models and modelling for process limits in metal forming,” *CIRP Annals*, vol. 68, no. 2, pp. 775–798, Jan. 2019, doi: 10.1016/j.cirp.2019.05.007.

[121] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the Digital Twin: A Systematic Literature Review,” *CIRP J Manuf Sci Technol*, vol. 29, pp. 36–52, May 2020, doi: 10.1016/j.cirpj.2020.02.002.

[122] F. Tao, M. Zhang, J. Cheng, and Q. Qi, “Digital twin Workshop: A New Paradigm for Future Workshop,” *Computer Integrated Manufacturing Systems*, vol. 23, no. 1, pp. 1–9, Jan. 2017, doi: 10.13196/j.cims.2017.01.001.

[123] Florian Steinlehner, “Private Communication.” 2021.

[124] “Abaqus Mesh Convergence,” *Massachusetts Institute of Technology*, 2017. https://abaqus-docs.mit.edu/2017/English/SIMACAEGSARefMap/simagsa-c-ctmmeshconverg.htm

[125] Livermore Software Technology Corporation, “LS-DYNA ® Keyword user’s manual - Volume 1.” Livermore, Jul. 27, 2017. [Online]. Available: www.lstc.com

[126] R. Courant, K. Friedrichs, and H. Lewyt, “On the Partial Difference Equations of Mathematical Physics,” *IBM J Res Dev*, vol. 11, no. 2, pp. 215–234, 1967.

[127] F. Steinlehner, A. Weinschenk, S. Kolb, S. Laumann, and W. Volk, “Numerical Description of the Physical Properties of Stretch Web Connectors in Progressive Die Stamping,” in *NUMISHEET*, 2022, pp. 921–935.

[128] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, 1st ed. arXiv preprint, 2021. [Online]. Available: http://arxiv.org/abs/2104.13478

[129] G. C. Cawley and N. L. C. Talbot, “On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation,” *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.

[130] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *arXiv preprint arXiv:1811.03378*, Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.03378

[131] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[132] D. C. Liu and J. Nocedal, “On the Limited Memory BFGS Method for Large Scale Optimization,” vol. 45, pp. 503–528, 1989.

[133] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, “Learning to Simulate Complex Physics with Graph Networks,” *International Conference on Machine Learning*, pp. 8459–8468, Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.09405

[134] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, “Learning Mesh-based Simulation with Graph Networks,” 2021. [Online]. Available: https://sites.google.com/view/meshgraphnets

[135] F. Steinlehner *et al.*, “Development of inline closed-loop vibration control in progressive die stamping using finite element simulation,” in *IOP Conference Series: Materials Science and Engineering*, Nov. 2020, vol. 967, no. 1. doi: 10.1088/1757-899X/967/1/012035.

[136] Livermore Software Technology Corporation, “Guideline for Implicit Analyses using LS-DYNA,” Jan. 13, 2018.

Appendix A
Details of Early Modelling Attempts

A.1 Beam Element Model

The 'Beam Element' model (Figure A.1) introduces beam elements to replace the shell elements of the stretch-web and strip carrier (Figure A.2). Beam elements offer the advantage of being computational faster than shell elements while still retaining the ability to capture through thickness strains and bending moments and simulate the dynamic response of the strip carriers and stretch-webs. Hughes-Liu beams elements were used with 5 integration points to minimize the error predicting the bending moment. Both the strip carrier and stretch-web primarily experience bending along the element length with the stretch-web also experiencing uniaxial tension and torsion during the forming and strip lift/advance operations, which in principle makes beam elements a suitable element selection. The cross sections of the beam elements were 13 mm and 2 mm wide for the strip carrier and stretch-webs, respectively, with a 1 mm thickness.

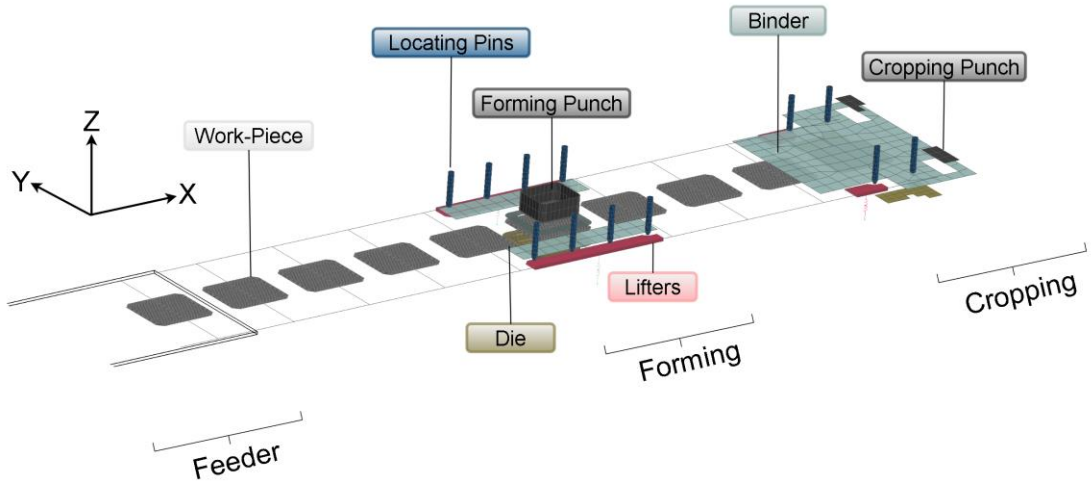

Figure A.1: Overview of 'Beam Element' model.

Beam elements have 1-dimensional interpolation, but, have a 3-dimensional bending response. Care was taken to ensure proper contact with the tooling. In particular, the elements of the strip carrier were placed along the outside edge of the strip carrier to ensure the elements contacted both the top and bottom of the lifters and were properly lifted and lowered. The elements of the beam carrier employed a cross-sectional offset that projected the cross-section towards the work-piece. As a result of the position of the strip carrier's elements, the stretch-web was extended to connect to the carrier, increasing

its length from 20 mm to 33 mm. The cross-section of the stretch-web was offset using the beam element as a center reference surface. A repercussion of using beam elements for the strip carrier is the inability for the locating pins to align the strip since pilot holes were unable to included. Instead, nodal constraints are applied to strip carrier to restrict lateral motion in the Y-direction using the '*Boundary_SPC_SET' keyword. This nodal constraint is similar to the ones employed by previous FE model iterations, but, is applied to the length of the strip carrier instead of a single node.

Similar to the 'Deformable-Rigid Switching' model, the '*DEFORMABLE_TO_RIGID' keyword was used to switch the work-pieces between rigid and deformable elements to efficiently incorporate multiple work-pieces into the model without drastically increasing the computational cost. All work-pieces begin as rigid bodies and are individually switched to deformable momentarily before the punch contacts the strip and switched back once the punch is released from the strip.

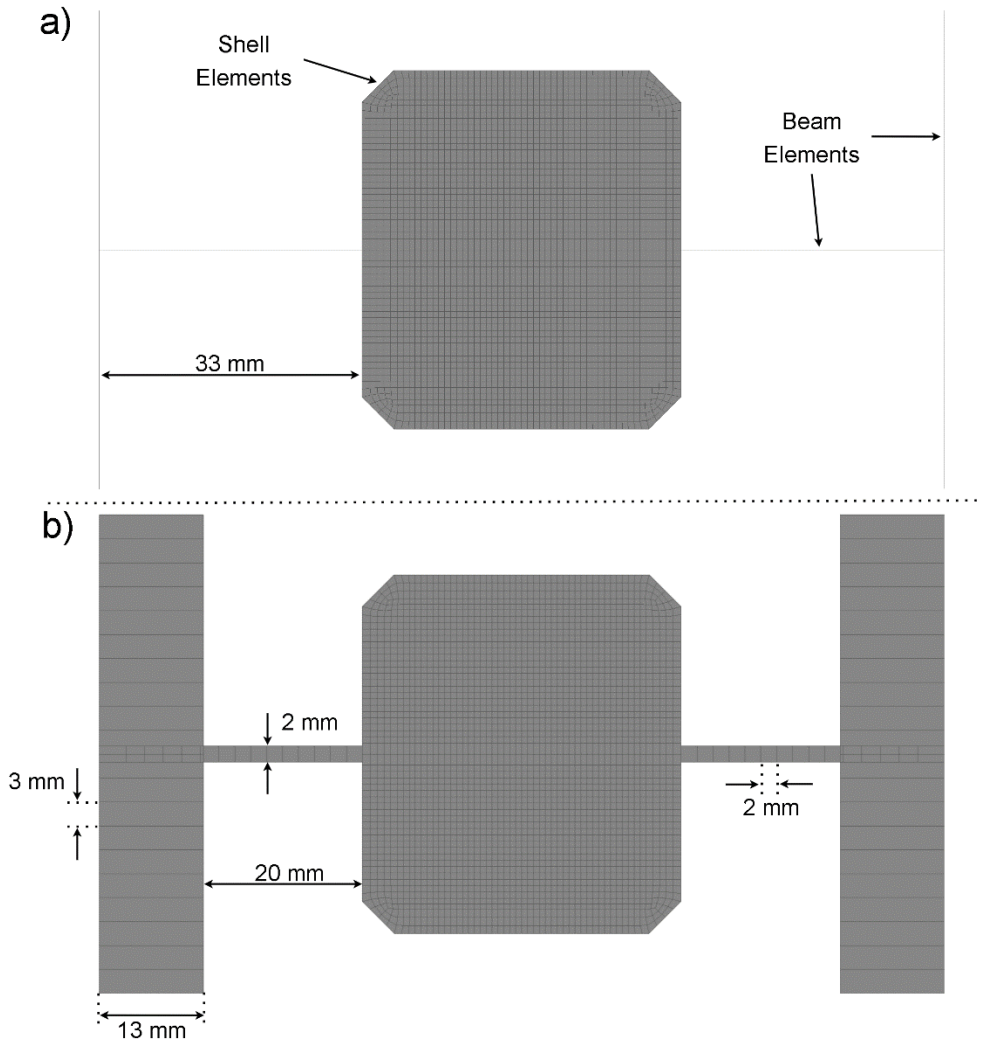

Figure A.2: Mesh of work-piece used in the 'Beam Element' model where (a) the beam cross-section is not visualized, and (b) the beam cross-section is visualized, showing the discretization of beam elements.

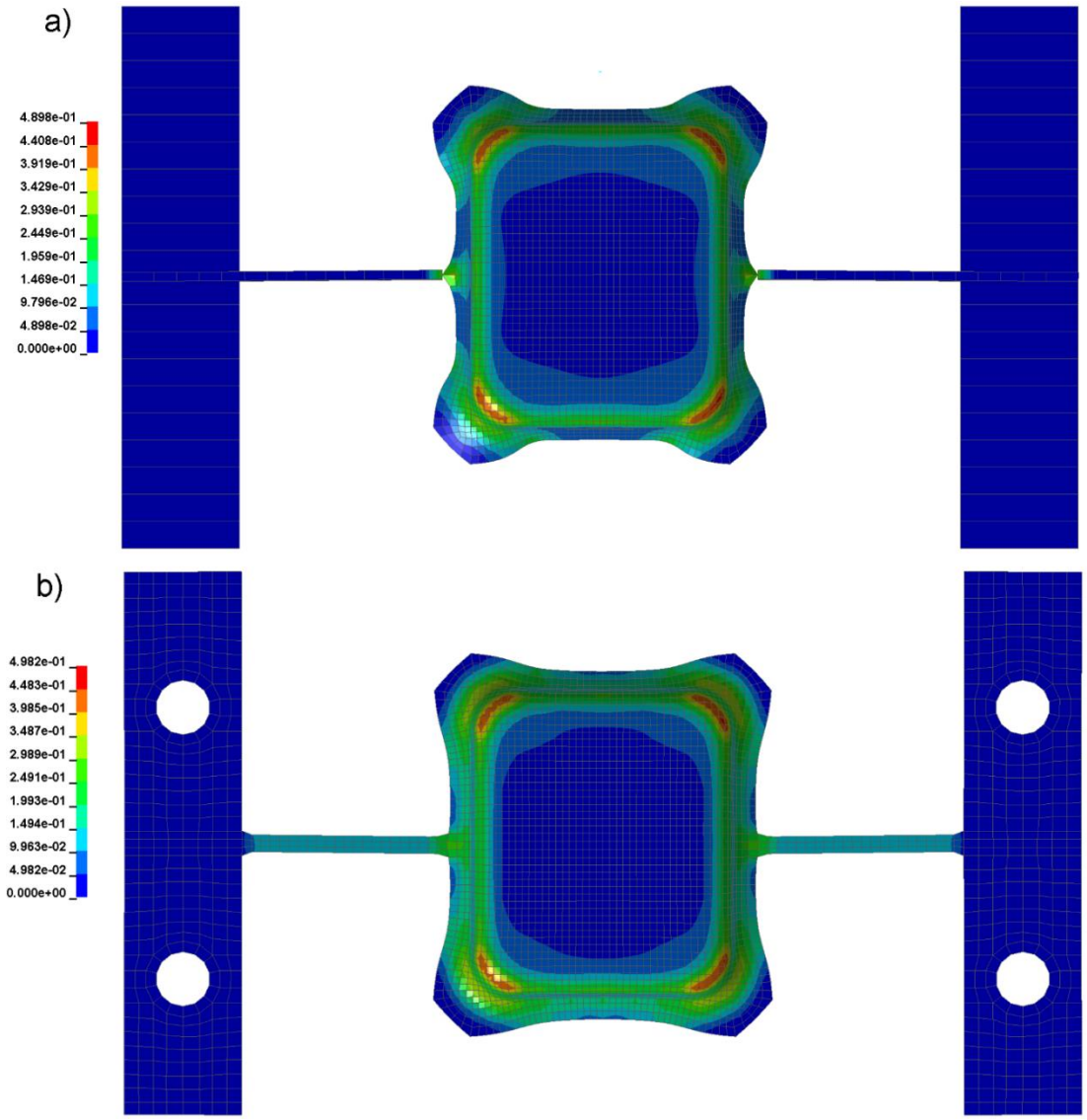

Figure A.3: Comparing effective plastic strain of formed parts for (a) 'Beam Element' model and (b) 'Single Work-Piece' model.

By using beam-elements and rigid-body switching, the number of shell elements is significantly reduced. Using LS-Dyna MPP with 24 cores, the completion time for four strokes at 120 SPM is 38 hours. While the ‘Beam Element’ model was able form multiple work-pieces fairly efficiently, the behavior of the forming and strip progression were undesirable. The shape and strain distribution of the formed work-piece deviate from the results of the ‘Single Work-Piece’ model (Figure A.3). Furthermore, the formed work-piece exhibited non-physical high-frequency oscillations after strip

progression which impedes the ability to accurately capture strip dynamics. Finally, the beam elements along the stretch-web and strip carrier erratically rotate throughout the simulation, further removing any confidence in the 'Beam Element' model as a candidate FE model.

A.2 Implicit Model

The 'Implicit' model uses the same mesh and set-up as the 'Single Work-Piece' model with exception of the implementation of the numerical method. Unlike explicit dynamic FEM, which can be implemented in a few simple steps, using an implicit method scheme requires several control inputs which define the integration scheme. Examples of control inputs include, but, are not limited to: the implicit solver method, frequency of stiffness reformation, time integration constants, and, displacement, energy and residual force convergence tolerances. Finding suitable implicit controls is often considered more of an art than a science since each variable influences the predictions of the FE model and requires special consideration that will differ between each problem. A considerable effort was made to identify suitable parameters for the implicit controls, using the guidelines published by Dynamore for LS-Dyna [136]. The most prominent issue encountered was lack of convergence of the time integration scheme. Early attempts of the 'Implicit' model failed to converge past the lowering of the strip during the down stroke of the press. The stiffness matrix would iteratively reform after reducing the size of the time-step until the minimum time-step was reached and convergence was still not achieved, causing model termination. Further iterations of the 'Implicit' model realized some success by reducing the convergence tolerance, decreasing the minimum time-step to 10^{-7} seconds, and implementing a "full Newton" solver (see Appendix A.3). The refined simulation was able to converge for the length of the simulation and form the work-piece with comparable strain distribution to the 'Single Work-Piece' model (Figure A.5). To simulate four strokes at 120 SPM, it took just over five hours using LS-Dyna SMP with 8 cores. However, the simulations were sensitive to non-linear behavior which required smaller time-steps, increasing the run-time. While the 'Implicit' model showed promise, an implicit method was ultimately not pursued due to concern over whether consistent results would be obtained when complexity would be reintroduced into the model. Through the multiple iterations of the 'Implicit' model, small changes to the implicit controls produced noticeable variations in the predicted response of the sheet metal strip as well as susceptibility to failed convergence. Furthermore, for the final iteration of the 'Implicit' model, erroneous behavior was observed along the strip carrier which is likely the result of numerical instability, supporting the decision to not proceed with an implicit formulation.

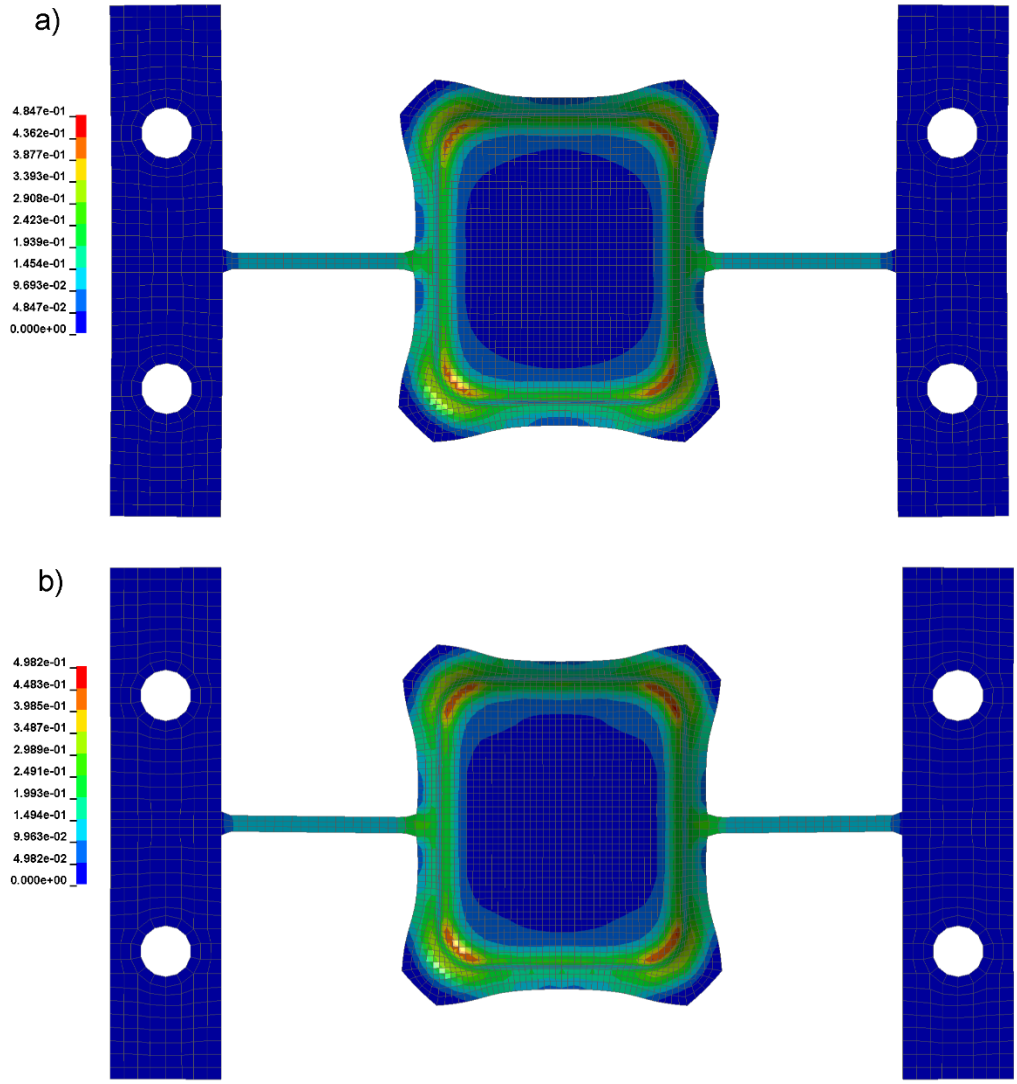

Figure A.5: Comparing effective plastic strain of formed parts for (a) 'Implicit' model and (b) 'Single Work-Piece' model.

A.3 Implicit Model Control Settings

The keyword cards for the implicit controls implemented in the FE model described in Appendix A.2 are presented below:

```
*CONTROL_IMPLICIT_AUTO
$#  iauto    iteopt    itewin          dtmin     dtmax     dtexp     kfail    kcycle
         2        11         5  1.00000E-7     0.001       0.0         2        25

*CONTROL_IMPLICIT_CONSISTENT_MASS
$#  iflag
         1

*CONTROL_IMPLICIT_DYNAMICS
$#  imass     gamma        beta     tdybir         tdydth         tdybur     irate
          1      0.55   0.27563        0.0   1.00000E28   1.00000E28         0

*CONTROL_IMPLICIT_GENERAL
$#  imflag            dt0   imform      nsbs       igs     cnstn      form    zero_v
         4   1.00000E-4          2         1         2         0         0         0

*CONTROL_IMPLICIT_SOLUTION
$#  nsolvr    ilimit    maxref     dctol     ectol     rctol     lstol         abstol
        12         1         30     0.001     0.001       0.0       0.9   1.0000E-20
$#   dnorm    diverg     istif   nlprint    nlnorm   d3itctl     cpchk
           1         1         1         3      -1.0       100         0
$#  arcctl    arcdir    arclen    arcmth    arcdmp    arcpsi    arcalf    arctim
         0         0       0.0         1         2         0         0         0
$#   lsmtd     lsdir      irad      srad      awgt      sred
         5         2       0.0       0.0       0.5       0.0

*CONTROL_IMPLICIT_SOLVER
$#  lsolvr    lprint     negev     order      drcm    drcprm   autospc   autotol
         5         2         2         0         4       0.0         1       0.0
$#  lcpack    mtxdmp
          2         0
```