CHALMERS
UNIVERSITY OF TECHNOLOGY

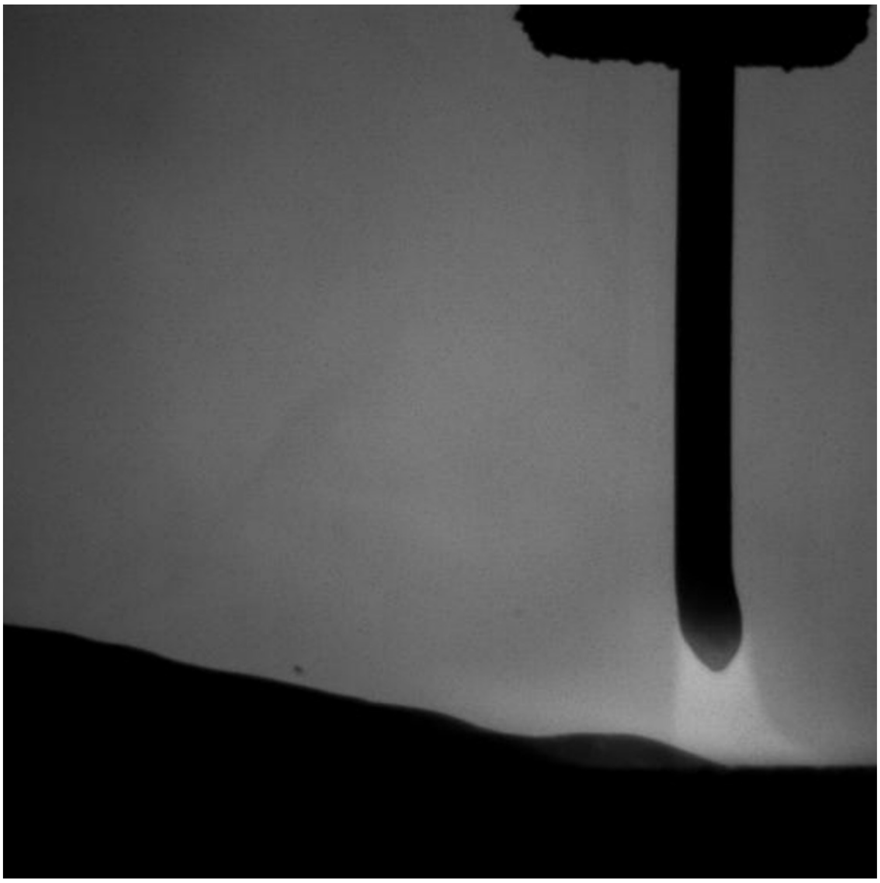


Image-Based Analysis and Estimation of Welding Features for Adaptive Control

Master's thesis in Systems, Control and Mechatronics

Samuel Ollila, Dharunkumar Senthilkumar

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Image-Based Analysis and Estimation of Welding Features for Adaptive Control

Samuel Ollila, Dharunkumar Senthilkumar

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Image-Based Analysis and Estimation of Welding Features for Adaptive Control
Samuel Ollila, Dharunkumar Senthilkumar

Supervisor: Willem de Wilde, ESAB
Examiner: Petter Falkman, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Frame in slow-motion video of back-lit welding process with visible wire, weld pool, and plasma channel.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Image-Based Analysis and Estimation of Welding Features for Adaptive Control
Samuel Ollila, Dharunkumar Senthilkumar
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Real-time monitoring and control of Pulsed Gas Metal Arc Welding (GMAW-P) is important for maintaining process stability and weld quality. This thesis investigates whether image-derived welding features can be used to support data-driven estimation of the physical state of the welding process. High-speed imaging was used to quantify geometric process features, which were then synchronized with electrical measurements to create a dataset for training and evaluating soft-sensing models.

Features related to arc behavior, droplet motion, weld pool position, and wire geometry were extracted from high-speed video. Two imaging configurations were evaluated: laser-lit and back-lit imaging. Laser illumination improved the visibility of fine structures but introduced strong reflections and artifacts that reduced feature extraction reliability. In contrast, back-lit imaging produced high-contrast silhouettes with cleaner object boundaries, making it more suitable for robust geometric feature extraction.

A computer vision pipeline was used to extract geometric features, including wire edges, contact tip, wire tip, and weld pool position. Additionally, droplets and spatters were detected and tracked using blob detection and simple motion rules. These image-based features were then synchronized with electrical signals and process parameters to build a time-series dataset.

Several statistical, machine learning, and deep learning models were evaluated for predicting image-derived features from electrical inputs, primarily voltage, current, and wire feed speed, along with additional engineered features.

Overall, the results show that electrical signals contain information related to key geometric welding features, but the estimation accuracy remains limited. The models generally capture the cyclic patterns and overall trends in the signals, but struggle with amplitude variations between sequences. The results support the feasibility of soft sensing for GMAW-P monitoring, but not yet the accuracy and robustness necessary for reliable adaptive control.

Keywords: Computer Vision, State Estimation, Soft-Sensing, Welding, Time-series.

Acknowledgements

We would like to thank ESAB for providing access to equipment, facilities, and experimental data that made this thesis work possible. The opportunity to work in an industrial research environment provided valuable practical insight throughout the project.

We would also like to express our gratitude to our supervisor, Willem de Wilde at ESAB, for his technical support, guidance, and continuous feedback throughout the thesis.

Furthermore, we would like to thank our examiner and supervisor Petter Falkman at Chalmers University of Technology, for taking on this thesis and for his academic guidance during the project.

Samuel Ollila & Dharunkumar Senthilkumar, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AC	Alternating Current
ARX	AutoRegressive with eXogenous input
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
CTWD	Contact Tip to Workpiece Distance
DC	Direct Current
EMD	Empirical Mode Decomposition
FFT	Fast Fourier Transform
FPS	Frames Per Second
GMAW	Gas Metal Arc Welding
GMAW-P	Pulsed Gas Metal Arc Welding
HDF5	Hierarchical Data Format
IIR	Infinite Impulse Response
LED	Light Emitting Diode
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
MIMO	Multiple-Input Multiple-Output
MLP	Multilayer Perceptron
MM	Milli-meter
MSE	Mean Squared Error
MSP	Multi-Stage Prediction
MTL	Multi-Task Learning
NaN	Not a Number
NARX	Nonlinear AutoRegressive with eXogenous input
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
ROI	Region Of Interest
RRSE	Root Relative Squared Error
SVM	Support Vector Machine
SVR	Support Vector Regression
WFS	Wire Feed Speed
XGBoost	eXtreme Gradient Boosting

Contents

1

Introduction

1.1 Background

Welding is a manufacturing process in which two or more components are permanently joined using heat, pressure, or both, often with filler material to enhance fusion and joint strength. Welding methods differ in heat source, heat input, and material transfer mechanisms, which strongly influence process dynamics and weld quality [1].

In arc welding, an electric arc is established between an electrode and the base material to create localized melting. In most arc welding processes, the filler material is used as the electrode, which melts in the process and becomes a part of the welded metal. Shielding, typically provided by a gas mixture or flux, is used to protect the molten metal and optimize heat and metal transfer characteristics [2].

A widely used form of arc welding is Gas Metal Arc Welding (GMAW). In GMAW a consumable filler wire is continuously fed through a welding torch and used as the electrode, while shielding gas is supplied through the nozzle. The electric arc melts both the electrode and the base material, forming a weld pool of molten metal. The shielding gas protects the molten metal from contamination, thereby improving weld quality [3]. A schematic of a GMAW process, specifically the Metal Inert Gas (MIG) and Metal Active Gas (MAG), is shown in Figure 1.1.

Pulsed Gas Metal Arc Welding (GMAW-P) is a form of GMAW where the arc is controlled using current pulses, which allows for improved control of the metal transfer and weld pool. It is widely used in safety-critical industrial sectors, including automotive manufacturing, construction, and aerospace, where consistent weld quality is essential, as defects may compromise structural integrity [4].

Despite the widespread use of GMAW-P, monitoring and control of the process remain challenging due to many interacting parameters and coupled thermal, electrical, and electromagnetic phenomena. Many process variables related to weld quality, such as droplet formation, arc geometry, and weld pool behavior, cannot be measured directly during operation. Instead, process monitoring primarily relies on indirect measurements such as voltage and current signals, which only partially reflect the underlying physical process states.

This limited observability motivates the use of data-driven approaches for process monitoring and state estimation in welding processes. High-speed imaging enables direct observation of arc behavior, droplet transfer, and geometric relationships between key welding features such as the wire tip, solid-molten transition point, weld pool, and plasma attachment. By extracting these geometric welding features from synchronized high-speed image sequences, datasets can be constructed linking image-derived geometric measurements with electrical process signals. By combining synchronized high-speed imaging, computer vision, and learning-based methods, this thesis investigates whether key geometric welding features can be estimated from electrical signals without relying on explicit physical process models.

In the long term, the industrial objective is to enable automatic real-time control of physical geometric process states, such as arc length, rather than relying primarily on indirect electrical indicators. In current practice, electrical process signals are commonly used to infer information about the underlying welding process, but their relation to physical quantities such as arc length depends on factors such as welding geometry and contact-tip-to-workpiece distance (CTWD). As a result, different process conditions may correspond to different electrical signal levels, meaning that monitoring and control strategies must balance responsiveness with robustness during the different phases of the pulse. This thesis contributes to this long-term objective by investigating whether image-derived geometric welding features can be estimated from electrical process signals, thereby providing a potential basis for more direct monitoring and future feedback control of the GMAW-P process.

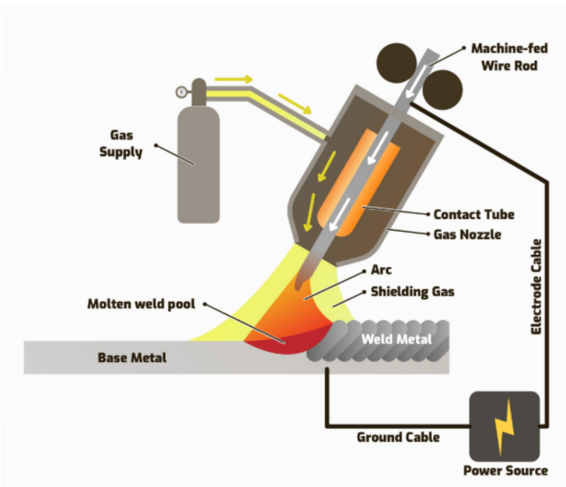


Figure 1.1: Schematic of the MIG/MAG welding process. Adapted from [5].

1.2 Purpose

The purpose of this master's thesis is to investigate whether geometric welding features in GMAW-P can be estimated from measured electrical process signals and static process context. The work aims to evaluate the feasibility of using such estimates as a basis for improved process monitoring and future feedback control of physical welding process states.

1.3 Objectives

The main objectives of this project are:

1. To develop computer vision methods for extracting selected image-derived geometric welding features from high-speed GMAW-P video data, and to evaluate their suitability as reference measurements for process-state estimation.
2. To construct a synchronized dataset by aligning selected image-derived geometric features with voltage, current, and relevant process parameters.
3. To develop and evaluate supervised learning models that estimate selected image-derived welding features from electrical process signals and static process context, and to compare their performance across different model classes.

In addition to the main estimation pipeline, the thesis also investigates droplet and spatter detection and tracking as a supplementary image-analysis task.

1.4 Limitations

This thesis is limited to the GMAW-P process for a specific base material, 304L stainless steel, shielding gas composition, M12 (Ar + 2% CO_2), and filler wire, 308LS, selected by ESAB. The scope was further constrained by the equipment and time available for data acquisition. As a result, the dataset only includes selected process configurations, captured using two welding machines and two high-speed imaging configurations: laser-lit and back-lit imaging.

The estimation models are limited to data from the back-lit imaging configuration, since this setup provided more reliable image-based feature extraction. Consequently, the model evaluation is limited to the investigated material, shielding gas, machine settings, and imaging conditions, and the results should not be interpreted as generally valid for all GMAW-P processes.

The work is further limited to offline evaluation of the proposed estimation approach. Real-time implementation, computational constraints, closed-loop control performance, and integration with an industrial controller were outside the scope of the thesis. The estimated targets are also based on image-derived reference measurements, meaning that uncertainties in the computer vision pipeline may affect both model training and evaluation.

2

Theory

2.1 Welding Process and Arc Physics

A metal transfer mode in GMAW describes how molten metal is transferred from the wire electrode to the weld pool, and is primarily influenced by the welding current. At low currents, metal transfer commonly occurs through repeated short-circuiting between the wire and the weld pool. As the current increases, larger droplets may detach more irregularly, while spray transfer occurs above a critical current level, where electromagnetic forces become the dominant detachment mechanism. These transfer modes differ mainly in heat input, spatter formation, droplet size, and transfer stability [6].

GMAW-P is a GMAW process where droplet detachment is controlled by adjusting the current waveform, which alternates between high current peaks that form and detach a droplet and a lower base current reducing the average current. The peak current and its duration define the size of the electromagnetic pinch force which detaches the droplet and accelerates it to the weld pool. If the pinch forces are excessive it may cause spatter due to the droplet impacting the weld pool with such force causing a splash, or the detachment could become violent causing the droplet to fragment into smaller particles [7]. An excessively long peak duration may also cause secondary droplets to detach following the primary droplet, leading to unstable transfer, excessive heat input, and increased spatter. Conversely, a too short peak duration may result in pulses where no droplet is detached, allowing the droplet to grow larger while attached to the wire, which can disrupt transfer stability [8]. Proper control of the current waveform and amplitude enable stable transfer and precise control of the droplet detachment. This additional control of the process results in improved arc stability, reduced spatter, and improved weld quality compared to conventional GMAW [9].

The welding arc in GMAW-P is governed by interactions between electrical, thermal, and electromagnetic phenomena that influence arc geometry, metal transfer behavior, and weld pool dynamics. Process parameters such as current, voltage and shielding gas composition affect observable arc characteristics, including arc length, arc width, droplet size, and spatter formation, which are closely related to weld quality and process stability.

Arc stability refers to the ability of the welding arc to maintain a consistent geometry and energy distribution during the welding process. A stable arc promotes uniform metal transfer, reduced spatter, consistent droplet formation and repeatable weld bead geometry. In contrast, arc instability can lead to irregular droplet detachment, excessive spatter and inconsistent penetration and bead geometry [10].

The arc geometry describes the shape and spatial distribution of the plasma column between the electrode and the workpiece. In GMAW-P, the arc typically exhibits a constricted plasma channel near the wire tip that expands into a bell-shaped profile toward the weld surface. The geometry results from the balance between electromagnetic pinch forces, plasma pressure, and thermal expansion of the ionized shielding gas. A constricted arc column produces higher current density and a concentrated heat source, which generally leads to deeper penetration and more stable droplet transfer. In contrast, a wider and more diffuse arc distributes heat over a larger area, reducing penetration and may contribute to less controlled droplet detachment leading to increased spatter [11].

Several process parameters influence the arc geometry and stability in GMAW-P. The current level and current waveform directly control arc energy, electromagnetic forces, and metal transfer behavior. Higher currents increase electromagnetic constriction and arc stiffness, often producing a smaller arc column leading to deeper penetration. Lower currents generate a cooler and less forceful arc, which may reduce penetration and make droplet transfer less stable. The pulsed current waveform is especially important in GMAW-P, since the peak current promotes droplet detachment while the background current maintains the arc between pulses.

Voltage is closely related to arc length and heat distribution. A higher voltage increases arc length, allowing the plasma column to expand and spread the heat over a larger area. This results in a broader arc profile and reduced heat concentration. A lower voltage shortens the arc and concentrates the heat input, but if the arc becomes too short, it may disturb the metal transfer and increase the risk of instability. Therefore, maintaining a suitable and consistent arc length is essential. A short, well-controlled arc promotes stable metal transfer, while excessive arc length can cause arc wandering and unstable droplet detachment due to reduced electromagnetic confinement.

Shielding gas composition also affects arc shape and stability. Argon-rich shielding gases generally promote a narrower and more stable plasma column, while changes in active gas content influence ionization behavior, heat transfer, and weld pool fluidity. Wire feed speed and travel speed further affect arc behavior and bead geometry. Increasing the wire feed speed raises the deposition rate and usually requires a corresponding increase in current to maintain stable transfer. Travel speed determines the heat input per unit length; a lower travel speed increases heat input and can produce wider and deeper beads, whereas a higher travel speed reduces heat input and may lead to insufficient penetration or unstable bead formation [12].

Contact Tip-to-Work Distance (CTWD) is another important process parameter that influences arc behavior and metal transfer characteristics. CTWD represents the distance between the contact tip and the workpiece surface and determines the electrical extension of the wire electrode before it enters the arc region. Increasing CTWD increases the electrode extension length, resulting in greater resistive heating of the wire before melting. This can reduce the current delivered to the arc for a constant wire feed speed, leading to changes in arc length, penetration depth, and droplet transfer behavior. Conversely, a shorter CTWD generally increases current density and heat concentration, promoting deeper penetration and a more concentrated arc. Variations in CTWD can therefore affect arc stability, weld pool behavior, and the consistency of metal transfer. Maintaining a stable CTWD is important for achieving repeatable process conditions and consistent weld quality [13].

The power input, which is closely related to the average current and arc voltage, directly affects the heat input delivered to the workpiece. Heat input plays a critical role in determining the weld quality since insufficient heat input may lead to incomplete fusion and lack of penetration, whereas excessive heat input can result in reduced joint strength [14]. In GMAW-P, the heat input is not only determined by the electrical parameters, but also by process dynamics such as travel speed, arc efficiency, metal transfer behavior and material properties. However, welding current and arc voltage are considered the parameters with the greatest influence on heat input. [15].

Shielding gas composition also affects the arc by influencing arc conductivity, ionization potential, plasma temperature and heat distribution. Both inert and active gases can be used in GMAW-P, either individually or as mixtures, and the choice of gas type and gas flow significantly influences arc stability, metal transfer behavior and heat input [3, 16]. Argon-rich mixtures generally promote a stable and constricted arc due to its low ionization potential, which supports smooth droplet transfer and reduced spatter, Additions of active gases, such as carbon dioxide or oxygen, can improve arc stiffness, weld pool wetting and penetration. However, excessive active gas content may lead to a more turbulent arc and increase spatter, reducing overall process stability.

Other contributing factors include wire feed stability and shielding gas flow. Variations in wire feeding can disturb the current transfer and droplet formation. Improper shielding gas flow can also disturb the plasma and reduce arc stability. By coordinating these parameters, the shape, intensity, and stability of the arc can be controlled to achieve consistent droplet transfer, reduced spatter, suitable penetration, and improved weld quality [17].

2.2 Welding Monitoring

To address adaptive control of GMAW-P processes, prior research has focused on time-series data analysis, image-based monitoring, and the integration of electrical signals and imaging. Voltage and current signals are particularly important, as their fluctuations are closely correlated with process stability and weld quality. The primary reason for this focus is that voltage and current are among the most readily available process signals that can be measured continuously during welding without directly interfering with the process. Techniques such as Empirical Mode Decomposition (EMD) and its variants have been widely applied to analyze non-stationary welding signals, demonstrating strong correlation to metal transfer modes and penetration depth [18]. Additionally, changes in arc voltage during the peak current period can be used to estimate weld pool surface characteristics and penetration depth [19]. High-speed data acquisition systems record these signals in real time, providing insights into arc stability and metal transfer modes [20].

2.3 Imaging Techniques

Image-based analysis complements electrical signal monitoring by providing direct observations of welding process characteristics such as droplet transfer, arc behavior, plasma geometry, and weld pool dynamics. High-speed imaging techniques are commonly used for this purpose, although direct observation is challenging due to the intense radiation emitted by the welding arc. Various illumination and optical filtering techniques are therefore employed to suppress arc brightness and enhance visibility of specific process regions [21].

Back-lit imaging uses an external light source, typically a high-intensity LED, positioned behind the welding region relative to the camera. The illumination creates a high-contrast silhouette of the electrode wire, droplets, and weld pool boundaries, enabling accurate extraction of geometric features such as wire contour, droplet size, and metal transfer behavior. Since the imaging relies on object silhouettes rather than emitted light intensity, back-lit imaging is particularly effective for detecting object boundaries and tracking dynamic features. A visualization of back-lit imaging is shown in Figure 2.1.

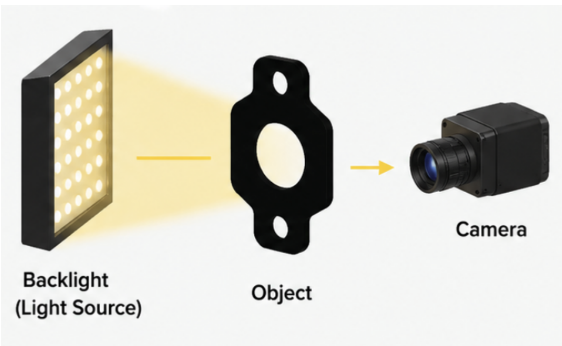


Figure 2.1: Overall visualization of the back-lit imaging process. (AI Generated)

Laser-lit imaging uses coherent laser illumination together with narrow-band optical filters centered around the laser wavelength. The narrow spectral bandwidth allows most of the broadband arc radiation to be rejected while preserving reflected laser light. This approach enables visualization of process regions that are otherwise obscured by arc intensity and can provide improved visibility of plasma structures, arc contours, and weld pool behavior [22]. A visualization of laser-lit imaging is shown in Figure 2.2.

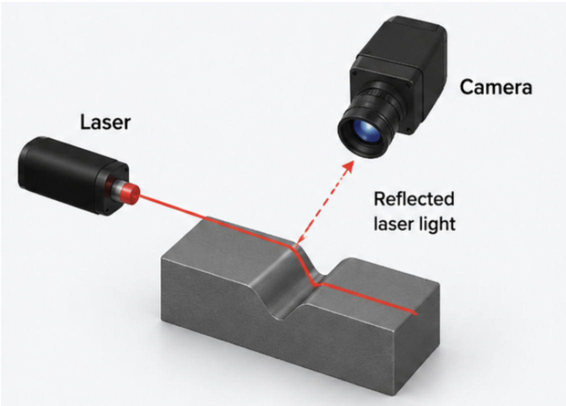


Figure 2.2: Overall visualization of the laser-lit imaging process. (AI Generated)

Temporal filtering methods, including pulsed illumination and synchronized camera exposure, can further suppress arc radiation and improve image quality. Depending on the welding process and observed phenomena, high-speed cameras typically operate at frame rates ranging from several hundred to more than 200,000 frames per second.

2.4 Feature Extraction from Welding Images

To extract geometric features from images, classical image processing techniques such as contour detection, connected component analysis, Sobel operators, K-means clustering, and Otsu thresholding have been applied [23]. Median filtering is commonly applied to reduce noise or suppress spatter. Advanced methods, like the improved Canny algorithm and the fast marching method, enhance edge detection and arc shape extraction [24, 25].

Detecting extremely thin linear structures, such as welding wires, can be achieved using edge detection combined with the Hough Transform [26]. Detecting thin welding wires begins with edge extraction using the Canny operator. Image gradients are computed as

$$M(x, y) = \sqrt{G_x^2 + G_y^2}$$

where G_x and G_y represent horizontal and vertical intensity gradients. Pixels with large gradient magnitude are retained as edge pixels.
The detected edge points are subsequently transformed into Hough parameter space using

$$\rho = x \cos \theta + y \sin \theta$$

Multiple edge points belonging to the same wire accumulate votes at similar (ρ, θ) values, allowing the thin welding wire to be extracted as a dominant line feature. Edge pixels mapped into parameter space allow reliable extraction of linear features even in noisy environments. Similar approaches enable robust detection of welding wires as dominant line structures.

2.4.1 Arc Characteristics

The detection of plasma arcs typically relies on exploiting their high-intensity and structured appearance [27]. Gradient-based methods highlight regions where intensity changes sharply, indicating edges or boundaries. High-gradient regions often correspond to plasma boundaries. Logarithmic intensity transformation

$$I_{\log} = \log(I + 1)$$

compresses high intensities while expanding low intensities, enhancing subtle structures in darker regions. Connected regions or contours can then be extracted to identify candidate arcs. Connected component analysis operates on a binarized image $B(x, y)$ obtained via thresholding:

$$B(x, y) = \begin{cases} 1, & I(x, y) > T \\ 0, & \text{otherwise} \end{cases}$$

Spatially connected foreground pixels are grouped into labeled components using 4- or 8-neighborhood connectivity. Each component C_i represents a spatially coherent region corresponding to a arc or plasma candidate.

For each component, basic descriptors are computed as:

$$A_i = \sum_{(x,y) \in C_i} 1, \qquad (x_i, y_i) = \left(\frac{1}{A_i} \sum x, \frac{1}{A_i} \sum y \right)$$

Shape and positional constraints, such as convexity, aspect ratio, or proximity to regions of interest, help filter spurious detections. For arcs with linear or smooth curvatures, ridge detection, polyline fitting, or Hough transforms can model their path and extract geometric properties such as length, curvature, or centroid.

Temporal median filtering [28] smooths out transient noise in time-series image sequences, preserving edges while mitigating outliers:

$$y_i = \text{median}(x_{i-k}, \ldots, x_i, \ldots, x_{i+k})$$

Similar challenges arise in astrophysical imaging, where faint plasma structures are embedded in images with large intensity variations. Gradient-based edge detection and logarithmic intensity scaling reveal subtle plasma boundaries, while ridge detection and contour-based techniques trace filamentary plasma structures. These methods inspire robust approaches for characterizing welding arcs in noisy, high dynamic range images.

2.5 Droplet and Spatter Detection and Tracking

Connected component analysis and blob detection [29] form a common basis for droplet and spatter analysis in welding imagery. These methods detect spatially coherent pixel groups based on intensity and size constraints. Each detected blob represents a candidate physical event such as droplet transfer or spatter ejection [30, 31]. Blob detection refines this by having shape consistency. A common measure is circularity:

$$\mathcal{C} = \frac{4\pi A}{P^2}$$

where A is the region area and P its perimeter. This enables separation of compact droplet-like blobs from irregular spatter regions.

Geometric constraints are used to reduce false detections. Area thresholds remove noise and plasma artifacts. Circularity constraints separate stable droplets from irregular arc emissions. ROI masking isolates the weld pool region and separates central transfer events from peripheral spatter. These constraints improve robustness under varying illumination and arc conditions
Extensions of these methods include fuzzy clustering and fuzzy C-means segmentation. These approaches handle ambiguity in pixel assignment and improve separation of overlapping structures. Hybrid vision pipelines combine segmentation, edge detection, and clustering to improve feature extraction quality for spatter and droplet statistics [23].

Temporal tracking extends spatial detection into motion analysis. Detected blobs are linked across frames to estimate trajectories, velocity, and transfer frequency. Kalman filtering [32] remains a standard approach due to computational efficiency and robustness under measurement noise.

Temporal tracking is formulated as a recursive state estimation problem under uncertainty. The object state is modeled as a linear dynamical system:

$$\mathbf{x}^{(t)} = \mathbf{F}\mathbf{x}^{(t-1)} + \mathbf{w}^{(t)}$$

$$\mathbf{z}^{(t)} = \mathbf{H}\mathbf{x}^{(t)} + \mathbf{v}^{(t)}$$

where $\mathbf{x}^{(t)}$ is the hidden state vector, $\mathbf{z}^{(t)}$ is the measurement, and $\mathbf{w}^{(t)}$, $\mathbf{v}^{(t)}$ represent process and measurement noise, assumed to be zero-mean Gaussian with covariances $\mathbf{Q}$ and $\mathbf{R}$.
For droplet and spatter tracking, the state typically encodes position and velocity:

$$\mathbf{x}^{(t)} = \begin{bmatrix} x^{(t)} & y^{(t)} & v_x^{(t)} & v_y^{(t)} \end{bmatrix}^T$$

The Kalman filter recursively performs two steps. The prediction step:

$$\hat{\mathbf{x}}^{(t|t-1)} = \mathbf{F}\hat{\mathbf{x}}^{(t-1|t-1)}$$

$$\mathbf{P}^{(t|t-1)} = \mathbf{F}\mathbf{P}^{(t-1|t-1)}\mathbf{F}^T + \mathbf{Q}$$

and the update step:

$$\mathbf{K}^{(t)} = \mathbf{P}^{(t|t-1)}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}^{(t|t-1)}\mathbf{H}^T + \mathbf{R}\right)^{-1}$$

$$\hat{\mathbf{x}}^{(t|t)} = \hat{\mathbf{x}}^{(t|t-1)} + \mathbf{K}^{(t)} \left(\mathbf{z}^{(t)} - \mathbf{H}\hat{\mathbf{x}}^{(t|t-1)}\right)$$

$$\mathbf{P}^{(t|t)} = \left(\mathbf{I} - \mathbf{K}^{(t)}\mathbf{H}\right) \mathbf{P}^{(t|t-1)}$$

Under Gaussian assumptions, the optimal recursive estimator is given by the Kalman filter [32], which minimizes the posterior estimation error covariance.

Data association methods such as nearest-neighbor matching and gating stabilize identity tracking under partial occlusion. Data association between detections and tracks is formulated as an assignment problem. A common cost metric is Euclidean distance:

$$d_{ij} = \|\mathbf{z}_i^{(t)} - \hat{\mathbf{z}}_j^{(t)}\|_2$$

Assignments are constrained using a gating function:

$$d_{ij} \leq d_{\text{gate}}$$

to reject physically inconsistent matches under motion constraints and occlusion.

Modern multi-object tracking frameworks extend this formulation by jointly optimizing detection confidence and temporal consistency. This can be expressed as:

$$\arg\max_{\mathcal{A}} \sum_{(i,j)\in\mathcal{A}} S_{ij}$$

where S_{ij} combines spatial proximity, motion consistency, and appearance similarity. These frameworks combine learned detectors with motion models to improve performance under fast dynamics and object overlap. These systems support stable estimation of droplet paths and spatter motion in high-speed welding sequences [33, 34]. Deep learning based tracking frameworks such as FairMOT and CenterTrack improve identity consistency in dense and dynamic scenes by coupling detection confidence with motion continuity [35, 36]. These approaches improve estimation of droplet transfer rates and spatter distribution under unstable arc conditions.

2.6 Analysis of Welding Signals

The electrical signals collected during gas metal arc welding (GMAW), particularly in pulsed direct-current and alternating-current variants, primarily consist of synchronized voltage and current time series. These signals capture the dynamic behavior of the arc, metal transfer modes (e.g., globular, spray, or short-circuit), and process instabilities such as droplet detachment, arc re-ignition, and plasma fluctuations. Typical characteristics include periodic pulsing in pulsed GMAW (with peak/base current phases), transient short-circuit events, and stochastic variations due to shielding gas interactions or workpiece irregularities. High-speed imaging synchronized with electrical data acquisition enables direct correlation of arc geometry and arc voltage as indicated in the paper [37, 38].

Synchronization helps match features from electrical signals with features from images. These combined features can be used to predict or classify welding outcomes, such as arc stability, transfer mode changes, and weld geometry

2.6.1 Data Analysis Methods

Literature on welding signal analysis consistently frames the problem as a study of temporal dependence, nonlinear coupling, and regime transitions in noisy, nonstationary time-series. A first step in most studies is the assessment of signal relationships using correlation-based measures. Pearson correlation is commonly used to capture linear coupling between voltage and current, while Spearman correlation extends this analysis to monotonic nonlinear relationships. These measures are often paired with scatter-based visual analysis to reveal operating regimes and arc resistance behavior [39].

Temporal structure is typically analyzed through delay and memory estimation. Cross-correlation is widely used to identify dominant lags between input and output signals, particularly between current, voltage, and derived weld pool responses. Autocorrelation is used to quantify intrinsic memory effects in the process, while mutual information is introduced in nonlinear settings to capture dependencies missed by linear measures. In many studies, lag selection is further refined through data-driven window ablation, where predictive models are evaluated across different temporal contexts to identify an effective memory horizon [40, 41].

Beyond temporal alignment, welding signals are frequently analyzed in the frequency domain to characterize periodicity and oscillatory behavior. Fast Fourier transform (FFT) [42] is used to extract dominant frequencies associated with pulsed transfer modes, while spectral analysis provides a global description of energy distribution across frequency bands. Event-driven phenomena such as short-circuiting and droplet transfer are commonly studied using peak detection [43] and threshold-based event extraction, which allows mapping of discrete physical events to signal patterns.

To understand interactions between multiple signals, studies frequently rely on mutual information and correlation structures to quantify coupling strength and nonlinear dependence. In addition, Principal Component Analysis (PCA) is often used to project high-dimensional measurements into lower-dimensional representations, enabling visualization of dominant process regimes and variability across welding conditions [44].

2.6.2 Signal Processing

Signal processing prepares raw welding data for analysis by mitigating noise and extracting transient features. Techniques include moving-average smoothing and low-/band-pass filters for denoising, spectral analysis (FFT) for frequency content, and wavelet-based decomposition for non-stationary transients. Event-driven processing applies thresholding and peak detection to isolate short-circuit phases or droplet events, with windowing to segment pulsed cycles. EMD and its variants, ensemble EMD, and complete ensemble EMD with adaptive noise, have been successfully applied to GMAW current signals, decomposing them into intrinsic mode functions that reveal amplitude-frequency distributions linked to welding stability and quality diagnosis [45, 46, 47].

In high-frequency welding applications such as GMAW-P, measured voltage and current signals may contain switching ripple and high-frequency fluctuations that are not directly relevant for the slower process dynamics of interest. Filtering can therefore be used to smooth the signals before analysis. In this thesis, a phase-corrected sixth-order Chebyshev low-pass filter with a cut-off frequency of 15 kHz was applied during offline preprocessing. However, because phase-corrected filtering is non-causal, this exact filtering approach would not be directly applicable in a real-time control implementation.

2.6.3 Feature Engineering

Feature engineering transforms processed signals into predictive inputs. Time-domain features include statistical moments such as mean, standard deviation, slope (time derivatives), min and max amplitudes, and event indicators, e.g., short-circuit duration. Frequency-domain features derived from FFT, e.g., dominant harmonics and power spectra, or wavelet transforms as time-frequency energy distributions. Physics-inspired features incorporate power ($P = U \cdot I$), heat input, or resistance estimates. Lagged features explicitly encode time dependencies (e.g., voltage at $t-k$), while normalization/scaling as z-score or min-max per window handles drifts, and event indicators flag thresholds for arc presence or droplet detachment. These engineered features, often combined with PCA for reduction, feed downstream models while preserving physical interpretability [46].

2.7 State Estimation

In related research fields dealing with complex and non-stationary time-series data, a wide range of statistical, machine learning, and deep learning methods have been employed for state estimation.

2.7.1 Prediction Strategy

In data-driven modeling of complex industrial processes such as GMAW-P, the selection of an appropriate prediction strategy is critical due to the presence of nonlinear dynamics, partial observability, and strong coupling between process variables. As described in Section 2.1, the welding process is governed by interacting electrical, thermal, and electromagnetic phenomena, where measurable signals such as current and voltage influence multiple underlying physical states, including wire, weld pool, and arc geometry and droplet transfer behavior.

A key challenge in this context is that not all target variables are equally observable from the available measurements. While certain arc characteristics, such as arc length, are closely related to electrical signals (e.g., arc voltage), other features, such as droplet size or spatter formation, depend on more complex interactions and may only be indirectly observable. This motivates the need for structured prediction strategies that account for varying degrees of observability among target variables [48].

A straightforward approach to predict the features is to try to predict them using only the measured signals, i.e.,

$$input\ signals \rightarrow target\ feature$$

Direct prediction or single-task learning is simple to implement and does not require assumptions about dependencies between target variables. However, the approach might be suboptimal in scenarios where some targeted features only have a weak correlation to input signals but strongly correlate with other target features. In that case, Multi-Stage Prediction (MSP) might be a better approach [49]. MSP is a strategy in which target features that are closely correlated with the input signals are first estimated as intermediate variables, and then later used as additional inputs to predict more complex or indirectly observable features, i.e.,

$$input\ signals \rightarrow intermediate\ states \rightarrow complex\ features$$

This approach enables the prediction of complex features even when the available input data is limited by leveraging intermediate representations that capture relevant process dynamics. However, it requires assumptions regarding the structure of the process, particularly in selecting appropriate intermediate variables. Such assumptions could be based on prior knowledge, for example, known physical relationships between input signals and target features.

By explicitly modeling these dependencies, multi-stage prediction can improve the estimation accuracy of complex features compared to direct prediction. However, the approach is susceptible to error propagation, as inaccuracies in the intermediate predictions may accumulate and negatively affect subsequent estimates [50].

Multi-Task Learning (MTL) is an approach where multiple features are predicted simultaneously, i.e.,

$$input\ signals \rightarrow [feature_1\ feature_2\ ...\ feature_N]$$

A key advantage of MTL is that it allows the model to learn a shared representation of the input signals that captures common factors influencing multiple outputs. This can act as an inductive bias, improving generalization and robustness. Furthermore, jointly learning related tasks can reduce the risk of overfitting compared to training separate models for each target feature. However, MTL does not explicitly enforce a structured dependency between target variables, as in hierarchical or multi-stage approaches. While correlations between targets may be implicitly captured through shared representations, the model does not guarantee physically consistent relationships between outputs. Additionally, conflicts between tasks may arise if different targets require competing representations, potentially degrading performance for some variables [51].

2.7.2 Statistical Analysis

Time series regression analysis aims at describing the development of data through time by leveraging their dependency on previous observations and, if possible, exogenous factors. These techniques use stochastic modeling tools in order to create parameterized models that describe dynamics and dependencies through time. They have been used extensively due to their explanatory power, effectiveness, and robustness in situations with little data or close-to-linear dynamics.

Classical statistical approaches include AutoRegressive (AR) and AutoRegressive with eXogenous input (ARX), which explicitly model temporal dependencies and have been widely used in signal forecasting and process monitoring. These models are often combined with frequency-domain or time–frequency feature extraction techniques, such as Fourier analysis or wavelet transforms, to improve performance on signals with varying periodicity and transient behavior [52].

An ARX model structure can be defined as:

$$y(k) = \sum_{i=1}^{n_a} a_i y(k-i) + \sum_{j=1}^{n_u} \sum_{i=0}^{n_b-1} b_{j,i} \mathbf{u}_j (k - n_k - i) \tag{2.1}$$

where $y(k)$ is the target variable at time k, u_j denotes the jth input signal, with $j = 1, \ldots, n_u$, n_u is the number of input signals, n_a is the number of previous outputs, n_b is the number of previous inputs, and n_k is the input lag used in the model.

A Nonlinear ARX model (NARX) can be used to represent a wide range of nonlinear systems. The general NARX model structure is described by:

$$
\begin{aligned}
y(k) = F[&y(k-1), y(k-2), ..., y(k-n_y), \\
&\mathbf{u}(k-d), \mathbf{u}(k-d-1), ..., \mathbf{u}(k-d-n_u+1)] \\
&+ e(k) \\
&= F(z(k)) + e(k)
\end{aligned} \tag{2.2}
$$

where y and u are the outputs and inputs, and F is a polynomial function of degree n defined as:

$$F(\mathbf{z}(k)) = \beta_0 + \sum_i \beta_i z_i(k) + \sum_{i \leq j} \beta_{ij} z_i(k) z_j(k) + \cdots + \sum_{i \leq \cdots \leq i_n} \beta_{i \cdots i_n} \prod_{m=1}^{n} z_{i_m}(k)$$

Beyond ARX and NARX formulations, several alternative frameworks are commonly used to capture more complex dynamics and uncertainty in time-series systems. State-space models represent the system using latent variables, enabling recursive estimation methods such as Kalman filtering for linear cases and nonlinear extensions [53] for more general dynamics. Probabilistic approaches such as Gaussian Process regression [54] provide a nonparametric alternative, where system behavior is inferred through kernel-based covariance structures with inherent uncertainty quantification. In addition, Volterra series [55] offer a functional expansion for weakly nonlinear systems by incorporating higher-order input interactions, while modern neural state-space [56] generalizes these classical structures by learning nonlinear state transitions directly from data, effectively unifying and extending traditional statistical modeling approaches.

2.7.3 Machine Learning Approaches

Beyond purely statistical methods, traditional machine learning models have been extensively applied to time-series prediction using engineered features. These include Support Vector Machines/Regression, K-Nearest Neighbors, Ensemble methods such as Random Forests and Gradient Boosting, which are valued for their robustness, interpretability, and effectiveness. Such models are commonly used in the monitoring and diagnosis of industrial conditions, where signals are noisy and labeled data are scarce, making them directly relevant for welding applications [57].

Ensemble methods
Ensemble methods have for a long time been used in a variety of domains, including machine learning for regression. The main idea behind ensemble methods is to gather several models to obtain an aggregated model that outperforms all of its sub-models [58]. Within ensemble methods, there are several approaches, including random forests, extra trees, and boosting methods. The large difference between the methods is if the trees are gathered in parallel or sequentially.

Random forests and extra trees are parallel ensemble methods, meaning that the sub-models are trained on a subset independently from each other, and the aggregated model then averages the outputs or utilizes voting depending on whether it is a regression or classification problem. Boosting methods train their sub-models sequentially, where the next sub-model uses the output from the previous model together with the input to try and correct the previous model's errors [59]. Each method has its advantages and drawbacks, e.g., parallel learning is more resistant against overfitting while sequential learning reduces bias.

Within parallel ensemble learning, split selection in decision tree models varies by method. In terms of regression, let the input-output mapping be defined as $f : \mathbb{R}^p \to \mathbb{R}^d$, where d denotes the number of output variables in a MIMO setting. The ensemble prediction is given by

$$\hat{\mathbf{y}}(x) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{T}_m(x)$$

where $\mathbf{T}_m(x)$ denotes the prediction of the m-th regression tree.
In Random Forest regression, each tree is trained on a bootstrap sample of the dataset, and splits are selected by maximizing variance reduction. For a candidate split s, the reduction in impurity is defined as

$$\Delta\mathcal{V}(s) = \mathcal{V}(\mathcal{S}) - \frac{|\mathcal{S}_L|}{|\mathcal{S}|}\mathcal{V}(\mathcal{S}_L) - \frac{|\mathcal{S}_R|}{|\mathcal{S}|}\mathcal{V}(\mathcal{S}_R)$$

where the node impurity is given by the within-node variance

$$\mathcal{V}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \|\mathbf{y}_i - \bar{\mathbf{y}}_\mathcal{S}\|_2^2 .$$

In Extra Trees regression, the ensemble structure is identical, but split selection is randomized instead of optimized. For a randomly selected feature j, the split threshold is drawn from a uniform distribution

$$t \sim \mathcal{U}(x_j^{\min}, x_j^{\max})$$

and the resulting partition is defined as

$$x_j \leq t \to \mathcal{S}_L, \quad x_j > t \to \mathcal{S}_R.$$

Random Forests choose splits by optimizing regression impurity criterion, such as variance or mean squared error reduction, while Extremely Randomized Trees (Extra Trees) select split thresholds randomly to improve efficiency. Both methods often achieve similar performance, though Random Forests may offer slightly higher accuracy at a higher computational cost. Extra Trees typically reduce variance due to increased randomness, while Random Forests reduce overfitting relative to single decision trees through aggregation and feature randomness.

Gradient boosting builds an ensemble of weak learners, typically decision trees, in a sequential manner, in contrast to parallel ensemble methods. The approach can be interpreted as performing gradient descent in function space [60]. The model is constructed additively as

$$\hat{y}(x) = \sum_{m=1}^{M} f_m(x)$$

where each $f_m(x)$ represents a weak learner. Training proceeds iteratively by updating the model as

$$f_m(x) = f_{m-1}(x) + \eta \cdot h_m(x)$$

where $h_m(x)$ approximates the negative gradient and η is the learning rate.

Several optimization techniques for gradient boosting improve both computational efficiency and predictive performance. One widely used method is eXtreme Gradient Boosting (XGBoost), which extends standard gradient boosting with explicit regularization, second-order optimization, and efficient tree construction [61].
XGBoost optimizes a regularized objective function [62] defined as:

$$\mathcal{L} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where l denotes the loss function, K denotes number of trees, and Ω penalizes model complexity. The regularization term is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where T denotes number of leaves, w_j denotes leaf weight, γ penalizes tree complexity, and λ controls weight magnitude.
Training proceeds in an additive manner. At iteration t, the model is updated as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Each new tree f_t is fitted using a second-order Taylor approximation of the loss function:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t)$$

where $g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$ represent first and second order gradients.
This formulation enables direct use of curvature information from the loss surface, improving split decisions and convergence stability. Split selection maximizes gain defined as:

$$\text{Gain} = \frac{1}{2} \left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda} \right) - \gamma$$

where G and H are aggregated gradients and Hessians for parent and child nodes.
This objective structure provides more control over model complexity while maintaining fast convergence and stable performance in high-dimensional and sparse

feature spaces.

Deep learning based methods
Deep learning methods enable direct modeling of temporal structure without explicit feature engineering and have become increasingly popular in time-series analysis. Recurrent neural networks (RNNs), and in particular long short-term memory (LSTM) [63] based architectures, are widely used to capture long-range temporal dependencies and dynamic process evolution. These models have been successfully applied in domains such as speech recognition, power system monitoring, and manufacturing process control.

LSTM networks [64] is a form of multi-task prediction model architecture 2.7.1 which extends recurrent neural networks through gated memory cells that enable stable learning of long-term temporal dependencies. At each time step t, the network receives the current input x_t, previous hidden state h_{t-1}, and previous cell state c_{t-1}. The gate operations and state updates are defined as:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}$$

Variable	Meaning
f_t	Forget gate
i_t	Input gate
$\tilde{c}_t$	Candidate state
o_t	Output gate
c_t	Cell state
h_t	Hidden state

where $\sigma(\cdot)$ denotes the sigmoid activation function, $\tanh(\cdot)$ denotes the hyperbolic tangent activation function, and $\odot$ denotes element-wise multiplication.
This gated structure enables selective retention and suppression of temporal information, allowing the network to learn both short-term transitions and long-range dependencies in sequential process signals.

Nonlinear extensions to classical time-series models may take place through the application of machine learning, replacing traditional linear functions by nonlinear function approximation methods. In particular, the NARX model follows such a methodology by retaining lagged dependency relations while using the function ($f_\theta(\cdot)$) to learn the mapping from inputs to outputs.

The Neural NARX (NN-NARX) model represents the predicted output as a nonlinear function of past input and output values. It can be written as

$$\hat{y}_t = f_\theta(\mathbf{x}_t) = f_\theta\left(\mathbf{u}_t, \mathbf{u}_{t-1}, \ldots, \mathbf{u}_{t-n_u}, y_{t-1}, y_{t-2}, \ldots, y_{t-n_y}\right), \tag{2.3}$$

where n_u denotes the number of input lags, n_y denotes the number of output lags, and θ represents the trainable network parameters.

In this work, the nonlinear mapping $f_\theta(\cdot)$ is modeled using a multilayer perceptron (MLP). For an MLP with L hidden layers, the input to the network is defined as

$$\mathbf{h}_t^{(0)} = \mathbf{x}_t.$$

The hidden-layer activations are then computed recursively according to

$$\mathbf{h}_t^{(\ell)} = \phi\left(\mathbf{W}^{(\ell)}\mathbf{h}_t^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \qquad \ell = 1, 2, \ldots, L,$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the weight matrix and bias vector of layer ℓ, respectively, and $\phi(\cdot)$ is a nonlinear activation function. Finally, the network output is given by

$$\hat{y}_t = \mathbf{W}^{(L+1)}\mathbf{h}_t^{(L)} + \mathbf{b}^{(L+1)}. \tag{2.4}$$

More recent approaches explore hybrid and multi-modal architectures, where time-series models are combined with image-based models. In such frameworks, Convolutional neural networks (CNNs) are used to extract spatial features from images, while LSTM [65] networks process synchronized voltage and current signals. The learned representations are then fused at the feature or decision level to enable joint prediction. Similar fusion strategies are common in video understanding, autonomous driving, and biomedical monitoring, where heterogeneous sensor data must be jointly interpreted [66].

The transfer of these modeling strategies to GMAW-P is well motivated, as welding current and voltage signals share fundamental characteristics with time-series data encountered in other complex industrial systems, including nonlinearity, noise, strong coupling between process variables, and multi-scale temporal behavior.

2.7.4 Evaluation Metrics

Regression performance in time-series learning is commonly evaluated using error magnitude and signal agreement metrics, which quantify the deviation between predicted values and corresponding ground-truth observations in time-indexed samples [67]. Different metrics capture different aspects of prediction quality and are therefore often used together to provide a more comprehensive assessment of model performance.

One of the most widely used measures is Root Mean Squared Error (RMSE), which expresses the prediction error in the original units of the target variable:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

RMSE represents the typical magnitude of prediction error while giving greater emphasis to larger deviations due to the squaring operation. Consequently, it is commonly employed when the interpretability of error magnitude in physical units is important.

Another commonly used metric is Mean Absolute Error (MAE), which measures the average absolute deviation between predicted and true values:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Unlike RMSE, MAE treats all errors equally and does not disproportionately penalize large deviations. Therefore, it is often preferred when a more uniform representation of prediction error is desired.

While RMSE and MAE quantify the magnitude of prediction errors, they do not evaluate the similarity in temporal behavior between predicted and measured signals. To assess the degree of linear association between two signals, the Pearson correlation coefficient is commonly employed [39]. This metric evaluates the strength of linear correlation independent of signal amplitude and is defined as:

$$r_{x,y} = \frac{cov(x, y)}{\sigma_x \sigma_y}$$

where *cov* is the covariance and σ is the standard deviation. For a sequence containing n samples, the Pearson correlation coefficient can be expressed as:

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

where x_i and y_i are the values at time i, and $\bar{x}$ and $\bar{y}$ are the mean across all samples. Higher correlation values indicate stronger agreement in signal trends, even when differences in amplitude exist.

In addition to measuring prediction error and signal correlation, it is also useful to compare model performance against a baseline predictor. Root Relative Squared Error (RRSE) is frequently used for this purpose [68]. RRSE evaluates model performance relative to a predictor based on the mean of the target values and is defined as:

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n} (true - pred)^2}{\sum_{i=1}^{n} (true - mean)^2}}$$

where an $RRSE < 1$ is better than predicting the mean, $RRSE = 1$ has the same performance as predicting the mean and $RRSE > 1$ is worse than predicting the mean.

Although these performance metrics provide insight into the overall predictive capability of a model, they do not explain how individual input variables contribute to model behavior. Understanding such relationships is particularly important in multivariate systems, where multiple input variables influence the predicted output. Feature importance methods are therefore employed to quantify the influence of input variables on model predictions.

A widely used approach to estimate the contribution of features in machine learning models is split-based feature importance. This method is commonly applied in tree-based ensemble methods and assigns importance according to the contribution of each feature to reducing the model loss function. The importance of feature j is calculated by aggregating the contribution across all tree splits:

$$I_j = \sum_{t=1}^{T} \sum_{s \in S_{t,j}} \Delta L_{t,s}$$

where T denotes number of trees, $S_{t,j}$ denotes splits in tree t using feature j, and $\Delta L_{t,s}$ denotes loss reduction at split s. Features with larger importance values are repeatedly selected in high-gain splits, indicating a greater contribution to predictive performance [69].

2.8 Research Gap

The uniqueness of this research is the use of the imaging method to characterize the welding process with respect to the arc, droplets, molten pool, and wire. The focus here is to identify some important attributes, such as the shape of the arc, intensity of the arc, and process state, which are related to easily identifiable process parameters like current, voltage, and wire feeding rate. These relations are then used to construct prediction models based solely on electrical measurements.

3
Method

3.1 Data acquisition

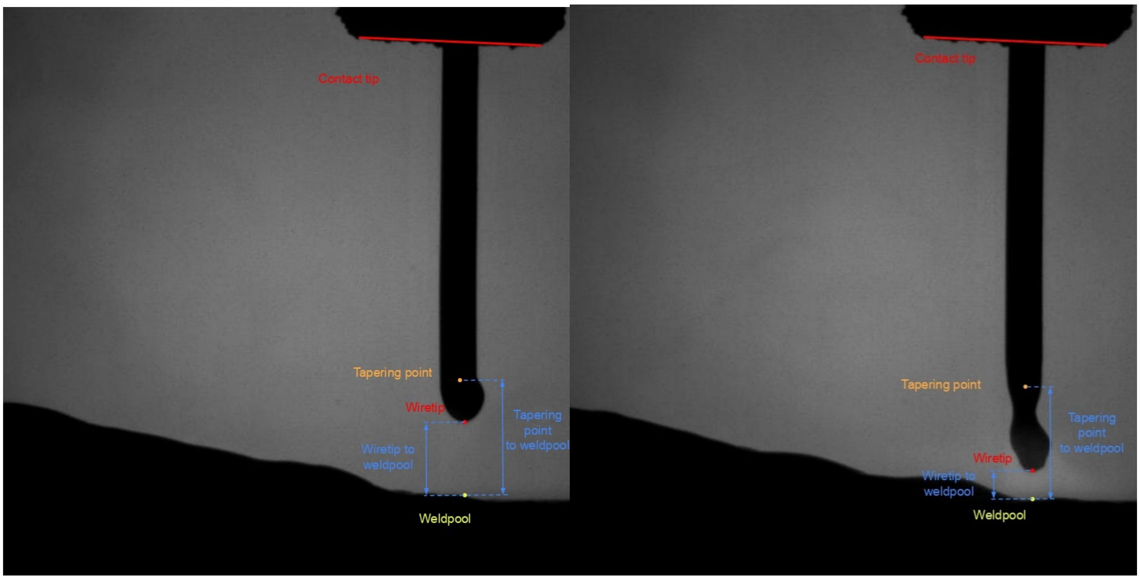
The data used in this thesis was captured in a laboratory setting at ESAB's office in Gothenburg. The footage was recorded using a high-speed camera at 10,000 Frames Per Second (FPS), equipped with optical filters selected based on the lighting configuration, laser-lit or back-lit. Two welding machines were used in the experiment to increase variation in the dataset. The two machines were set at different settings varying the CTWD, Wire Feed Speed (WFS), and weld dynamic settings such as arc length oscillation and pulse shape, which alter the arc dynamics and droplet transfer behavior.

The laser-lit footage was captured using a 500W laser at 810 nm that illuminates the wire and weld pool against a background. The laser's intensity and the illumination duration can be adjusted to allow for a visible arc while still capturing the contour of the wire and weld pool. The back-lit footage was captured using 70W LED backlighting at 630nm, which clearly reveals the contours of the objects in the image but provides limited depth information. The LED backlight's intensity can also be adjusted to allow for a visible arc while capturing the contours of the objects in the footage.

During the weld sequence, internal measurements from the welding machine are recorded at 200kHz and synchronized with the video frame trigger times. The recording is then saved in an HDF5 file, which contains all measurement history and rolling variables calculated by the machine during the weld sequence, and the time each frame is captured.

3.2 Computer Vision-Based Feature Extraction

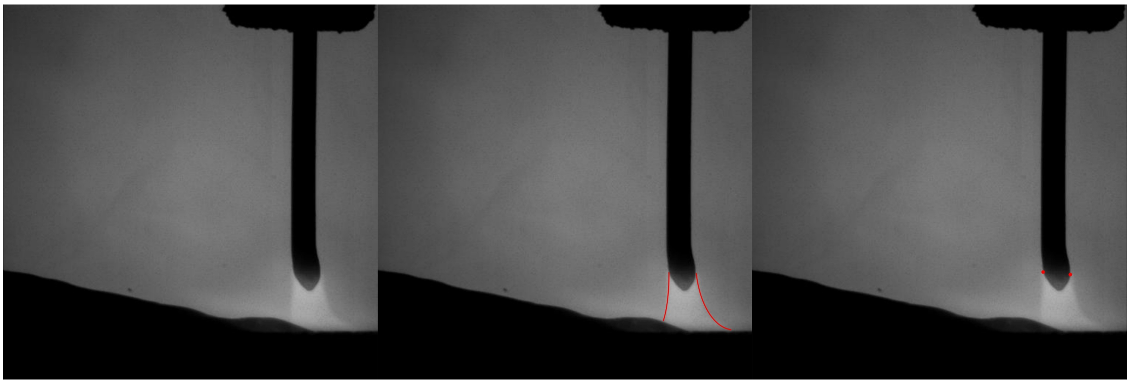
This section describes the full computer vision pipeline, from extracting the frames from the videos, preprocessing the individual frames, and feature extraction. A visualization of the desired features, wire tip, tapering point, weld pool, and the distances between them, and the contact tip can be viewed in Figure 3.1.



(a) (b)

Figure 3.1: Desired features from a sample footage collected at the lab

A visualization of the desired arc features plasma channel edges and plasma attachment points can be viewed in Figure 3.2.



(a) Original (b) Plasma channel edges (c) Plasma attachment

Figure 3.2: Desired arc features from a sample footage collected at the lab

3.2.1 Preprocessing

Each welding sequence provided has a video recording in .avi format. The first step is to extract the individual frames from the video to prepare it for further analysis. This was done using the OpenCV's `VideoCapture` and the frames were temporarily saved and sorted into a folder per video.

Since the footage used were captured in both a back-lit and laser-lit setting, separate preprocessing steps for each method were needed to enhance desired features. The first step was the same in all videos and it was to make sure it was in a grayscale format using `cvtColor`, and normalize the image in the range (0,255).

Since the wire in the laser-lit footage is subject to direct illumination, glare on the wire is produced. Through analysis, this glare was found to interfere with the detection of the desired features and an algorithm for glare removal was developed. The glare removal process starts with applying Gaussian blur with a kernel size of 5×5 and sigma of 1.0 to reduce image noise. A top-hat morphological operation is then performed using a 7×7 elliptical kernel, highlighting bright regions such as spots or streaks smaller than the kernel. The highlighted regions are then extracted as a binary mask using Otsu thresholding, where pixels identified as glare are assigned 255 and other pixels 0. This mask is then dilated once using a 7×7 structuring element, expanding the detected glare regions to include neighboring pixels improving coverage. Finally, the masked area is inpainted using the algorithm `INPAINT_TELEA`, which propagates the surrounding pixels in a radius of 9 pixels inward. Glare removal was only applied to the laser-lit footage, since the back-lit footage was not affected by the same direct reflections.

The last step of the preprocessing was to perform Canny edge detection on the preprocessed image. The Canny edge detector computes the image intensity gradients using Sobel kernels in the x and y directions, where the aperture size was set to 3×3. Subsequently, a hysteresis operation is performed using two empirically chosen Canny thresholds, an upper and a lower threshold. The pixels are then classified based on their gradient magnitude and the Canny thresholds. Pixels are classified as non-edges, weak edges, or strong edges if their gradient magnitudes are below the lower threshold, between the thresholds, or above the upper threshold, respectively. The weak edges are kept only if they are connected to a strong edge, and all strong edges are kept in the final result. The thresholds used for each image type are defined in Appendix A.

For arc detection a separate preprocessing pipeline was used, keeping the processed image more similar to the raw footage by only normalizing the footage in the range (0,255) and applying Gaussian blur with a kernel size of 5×5 and sigma of 1. This was needed to keep the subtle edges of the plasma channel.

Preliminary analysis showed that the laser-lit recordings produced strong specular reflections and unstable contour extraction, resulting in unreliable feature segmentation and increased false detections. Although glare-removal preprocessing was investigated, the extracted features remained less consistent than in the back-lit recordings. Consequently, the subsequent feature extraction and regression analysis were restricted to the back-lit footage.

3.2.2 Feature extraction

This section describes the extraction of required features such as points or lines detected in the image that are later used to calculate the target physical features.

Wire identification
To identify the wire in the image, the straight shape of the solid wire and the parallelism of the edges were utilized. First, all straight edge segments were detected in the image using probabilistic Hough transform through the OpenCV function `HoughLinesP` on the Canny edge map. All detected lines were then filtered to retain only near-vertical lines within the expected length range for the wire edges. Remaining lines were then compared with each other and grouped together if they were determined to be parts of the same edge. These groups were then merged together by fitting a straight line segment to the group. The remaining lines were then compared with each other in terms of relative orientation, where two parallel line segments, whose distance fell within a predetermined range, were selected as the final wire edges. The parameter settings used for the probabilistic Hough transform, line filtering, grouping, and final edge-pair selection are provided in Appendix A.

Definition of Search space
To minimize the risk of error due to background noise, irrelevant portions of the images were excluded by defining a rectangular search space based on the wire location. The search space included all relevant regions of the image containing the wire and arc features of interest.

The upper boundary of the search space was determined by locating the contact tip, the part from which the wire extends. This was achieved by searching upward along two lines parallel to the wire edges offset from the wire centerline, until an edge was detected in the Canny edge map. In images where the contact tip was not included or could not be located, the top of the image was used as the upper boundary.

The lower boundary was set as the base metal's surface y-value in the image. The base metal's surface y-value was fixed in the laser-lit footage as the base metal was at the same y-level throughout the footage. In the back-lit videos, the base metal y-value was found using a vertical search upward from the bottom in the Canny edge map along both sides of the image. The base metal y-value was set when an edge with a width over a certain threshold was detected on either side to avoid assuming a fixed weld pool location. The threshold values used for base metal detection are defined in Appendix A.

Wire tip detection
Canny edge detection was performed again, but now on the cropped preprocessed image with only the search space. An elliptical morphological closing operation was then applied to connect broken edges of the wire due to arc flash or noise. The resulting edges were then used to identify contours using OpenCV's `findContours`. The detected contours were then filtered to retain only those of the wire based on expected contour properties such as angle, length and width, which are defined in Appendix A. If no remaining contour exceeded the length threshold, the wire contour was assumed to be split into multiple parts. Through visual analysis, this was found to often consist of two parts, so instead of selecting the single longest con-

tour, the two longest contours were selected. Since all footage was captured upright, the wire tip was identified as the lowest point of the wire contour. In the case of a split contour, the wire tip was selected as the lowest point among the two contours.

Weld pool tip detection

The tip of the weld pool was determined by approximating the wire centerline using the two wire edges, extending it downward through the image. Starting one pixel below the wire tip height, a stepwise search was performed downward along the line in the Canny edge map. At each step, a profile perpendicular to the centerline was sampled over a 5-pixel radius until an edge was detected. The location of the edge pixel closest to the centerline was defined as the weld pool tip.

In parts of the footage, detached droplets intersect this line which can lead to an incorrect estimation of the weld pool tip position. To address this issue, a filtering method was implemented to remove droplets by identifying contours with approximately circular shapes. This was done by calculating each contours circularity. Each contour with a circularity value higher than 0.4 was then removed prior to the search, reducing the risk of incorrect weld pool tip detection.

Solid-molten transition point detection

The solid-molten transition point refers to the location along the wire where the material changes from a solid state to a molten state due to heating. Since the transition occurs gradually, there is no single exact point at which the change takes place. Instead, the transition can be defined using different observable features. In this work, two definitions are used, the point where the texture of the wire changes and when the wire begins to taper.

In the laser-lit footage, a change in the surface texture can be observed along the wire as a result of melting. This texture change is not observable in the back-lit footage due to the lack of direct illumination. To identify the texture change, a binary glare mask was extracted using morphological top-hat filtering with a 7×7 elliptic kernel and Otsu thresholding. The mask components were then dilated using a 3×3 elliptical kernel to connect nearby mask components. To isolate the glare mask along the wire, all components below the wire tip were discarded and the largest remaining component was selected. The solid-molten transition point was then defined as the lowest point of the selected glare mask component.

The point at which the wire starts to taper was also determined. To detect this, a line was fitted to each of the wire edges, extending over the full Canny edge map. Starting 20% of the segment length above the lower end of each edge segment, a search was performed along the fitted lines downwards. At each step, the distance between the fitted line and the wire edge pixels was measured. The search was stopped when no valid edge pixels were detected for a specified number of consecutive rows, when the closest detected edge pixel deviated from the fitted line by more than a predefined vicinity threshold, or when the projected line left the image frame. The threshold and the number of consecutive steps needed to end the search are defined in Appendix A. The final valid edge pixel for each wire edge was defined as that edge's stop point. If stop points were found for both wire edges, the tapering point was calculated as the midpoint between them.

Arc characterization
The arc was characterized by quantifying features of the plasma channel, such as width and attachment height on the wire.

Since the arc is only visible during a specific part of each pulse, an initial step in the image analysis was to detect if it was present or not. This was done by analyzing the intensity in a band of pixels between the located wire tip and weld pool. The mean intensity of the band was compared with a threshold calculated from the background intensity, which was extracted from a predetermined area to the left of the wire, where the threshold was defined as:

$$\text{Threshold} = \mu_{bg} + 3 \cdot \sigma_{bg}$$

where μ_{bg} and σ_{bg} are the mean and standard deviation of the intensity in background area, respectively. If the mean intensity in the band was higher than the threshold, the arc was considered present.

If an arc was present, the preprocessed arc image was passed to a function that finds the plasma channel edges. The image was cropped and only the arc ROI, defined as the area between the wire tip and weld pool, was used in the analysis. The ROI was further preprocessed using Contrast Limited Adaptive Histogram Equalization (CLAHE) with a clip limit of 2.0 and grid size of 8×8 to enhance local contrast in the image. A Gaussian blur with kernel size of 5×5 and sigma of 1.0 was also applied to reduce image noise.

The ROI was divided into bands, where the first band was one pixel high and the following bands are of a predetermined height. The assumptions about the plasma channel that are used to identify the plasma channel edges are that the plasma is brighter than the surrounding background and that the edges are on either side of the wire tip in the image. Starting with the initial band, the gradient was calculated in each band where the strongest positive gradient was selected as the left edge and the strongest negative gradient was selected as the right edge.

After the initial edge was detected, the search in the subsequent band was limited to the vicinity, defined by a tracking window radius, of the detected edge x-coordinate, due to the plasma edge being continuous. The search was stopped when all bands have been analyzed or if the largest gradient in the previous edge vicinity was below a threshold, due to the edge being too faint. The plasma channel width was then defined as the average distance between the detected edges.

The plasma attachment point on each side of the wire was estimated within a rectangular ROI positioned above the wire tip. The ROI was of a predetermined height and width, and was centered horizontally on the wire tip. Gaussian blur was applied with a kernel size of 5×5 and sigma of 1.0 to reduce image noise, after which an intensity threshold was calculated as:

$$\text{Threshold} = \mu_{ROI} + k \cdot \sigma_{ROI}$$

where μ_{ROI} and σ_{ROI} are the mean and standard deviation of the pixel values withing the ROI, respectively, and k is a predetermined factor.

The ROI was then scanned row by row from top to bottom where the left and right half of each row was analyzed separately. When one or more pixels above the threshold was detected in a row, the one closest to the wire centerline was selected. The first detected point on each side was defined as the plasma attachment point. The plasma attachment height was calculated by averaging the y-coordinates of both points.

The parameters used for the arc characterization are included in Appendix A.

3.2.3 Droplets and Spatters

This section describes formulation methods used for droplet and spatter detection and tracking. The pipeline consists of feature detection, state estimation, and track management.

Feature Detection
Each frame $I(x, y)$ is converted to grayscale and processed using a blob detection stage based on OpenCV `SimpleBlobDetector`. Detection is performed under fixed parameter settings: minimum area $A_{\text{min}} = 10$ pixels, maximum area $A_{\text{max}} = 3000$ pixels, and minimum circularity $C_{\text{min}} = 0.5$. Inertia, convexity, and color filters are disabled.
Each detected blob is represented by a keypoint:

$$\mathbf{k}_i = (x_i, y_i, r_i) \tag{3.1}$$

where x_i, y_i are pixel coordinates and r_i is the blob radius.
Area and equivalent diameter are computed as:

$$A_i = \pi r_i^2, \qquad d_i = 2\sqrt{\frac{A_i}{\pi}} \tag{3.2}$$

Detections are retained if:

$$10 \leq A_i \leq 3000 \tag{3.3}$$

Circularity is defined as:

$$C_i = \frac{4\pi A_i}{P_i^2} \tag{3.4}$$

and only detections with $C_i \geq 0.5$ are kept. We evaluated circularity with different values between 0-1 and 0.5 resulted in best detection.

Spatial Segmentation
A search region Ω is defined per frame using a polygonal mask. Its bounding box is:

$$x^{\Omega}_{\text{min}} = \min(\Omega_x), \quad x^{\Omega}_{\text{max}} = \max(\Omega_x) \tag{3.5}$$
$$y^{\Omega}_{\text{min}} = \min(\Omega_y), \quad y^{\Omega}_{\text{max}} = \max(\Omega_y) \tag{3.6}$$

The horizontal center is:

$$x_c = \frac{x^{\Omega}_{\text{min}} + x^{\Omega}_{\text{max}}}{2} \tag{3.7}$$

The center region of interest is defined as:

$$\mathcal{R}_c = \{(x, y) \mid |x - x_c| \leq 30(pixels),\ y \leq y_{\text{pool}}\} \tag{3.8}$$

Note that the distance is from the vertical center line. Left and right regions are:

$$\mathcal{R}_l = \{x < x^{\Omega}_{\text{min}}\} \tag{3.9}$$
$$\mathcal{R}_r = \{x > x^{\Omega}_{\text{max}}\} \tag{3.10}$$

Classification
Each detection is assigned a class based on area and spatial location.
Droplets:

$$A_i \geq 500\ \wedge\ (x_i, y_i) \in \mathcal{R}_c \tag{3.11}$$

Spatters:

$$A_i < 500\ \vee\ (x_i, y_i) \notin \mathcal{R}_c \tag{3.12}$$

State Estimation
Tracking is performed using a constant-velocity Kalman filter.
State vector:

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & v_{x,t} & v_{y,t} \end{bmatrix}^T \tag{3.13}$$

Measurement vector:

$$\mathbf{z}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{3.14}$$

State transition:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{w}_t \tag{3.15}$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Noise covariances:

$$\mathbf{Q} = 0.03\,\mathbf{I}_4, \qquad \mathbf{R} = 0.5\,\mathbf{I}_2 \tag{3.17}$$

Data Association

Association is performed using Euclidean distance between predicted track position (x_t^p, y_t^p) and detection (x_i, y_i):

$$d_{i,t} = \sqrt{(x_i - x_t^p)^2 + (y_i - y_t^p)^2} \tag{3.18}$$

Matching thresholds:

$$d_{i,t} \leq 20 \quad \text{droplets} \tag{3.19}$$
$$d_{i,t} \leq 18 \quad \text{spatters} \tag{3.20}$$

Greedy assignment is applied with one-to-one constraints.

Track Management
Each track is represented as:

$$\mathcal{T} = \{(x_t, y_t, A_t)\}_{t=1}^{N} \tag{3.21}$$

A track is confirmed if:

$$N_{\text{hits}} \geq 2 \tag{3.22}$$
$$\sqrt{(x_N - x_1)^2 + (y_N - y_1)^2} \geq 6 \tag{3.23}$$
$$y_N - y_1 \geq 4 \quad \text{(droplets)} \tag{3.24}$$

Track termination is applied after:

$$3 \text{ missed frames} -> \quad \text{droplets} \tag{3.25}$$
$$2 \text{ missed frames} -> \quad \text{spatters} \tag{3.26}$$

Velocity Estimation
Velocity is computed as:

$$v_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{3.27}$$

With unit frame spacing, this corresponds to velocity in pixels per frame.

Trajectory Output
Each trajectory is represented as:

$$\mathcal{T} = \{(x_t, y_t, A_t, d_t)\}_{t=1}^{N} \tag{3.28}$$

The output consists of per-frame measurements including class label, track ID, centroid position, area, and equivalent diameter.

3.2.4 Qualitative Evaluation

The accuracy of the extracted features was evaluated iteratively during the development of the extraction methods. The main focus was to identify false, missing, or inaccurate detections in order to improve robustness through parameter tuning, algorithmic adjustments and additional geometric constraints.

The evaluation was primarily performed through visual inspection of representative frames with feature overlays. This allowed failure cases to be identified, such as missing edges, false contour detections, incorrect weld pool localization, or unstable arc segmentation. These observations were then used to refine the feature extraction pipeline.

The iterative qualitative evaluation also informed the development of the anomaly detection stage described in Section 3.3.1. By analyzing recurring failure cases in the extracted features, criteria for anomaly identification and correction could be established.

3.3 Data Preparation

This section describes how the data was analyzed and altered to facilitate for model training.

3.3.1 Anomaly Detection

Anomaly detection was performed on the time-series outputs to identify temporally inconsistent or physically implausible measurement sequences. Each temporal sequence containing the results from the CV-pipeline was processed independently using a rule-based scoring framework. Even though the rules developed are applicable to all the features extracted from images, only the following features are considered for the study - $wiretip(x, y)$, $tapering(x, y)$, mm_per_px as these are the features used for creating the ideal outputs for the model to be trained on in later stages. A frame was flagged as anomalous when its aggregated score exceeded a threshold τ, defined in Appendix A.

Each frame received an anomaly score defined as

$$S_t = \sum_k w_k \cdot \mathbb{1}_k(t)$$

where w_k is the weight of anomaly rule k, and $\mathbb{1}_k(t)$ indicates whether rule k is triggered at time t. A frame is classified as anomalous when

$$S_t \geq \tau$$

Three types of rules were used. The first was missing value detection, defined as missing values in critical process variables:

$$\mathbb{1}_{missing}(t) = \mathbb{1}(x_t = \text{NaN})$$

The second was sudden change detection identified abrupt deviations using a rolling baseline:

$$\mu_t = \frac{1}{w} \sum_{i=t-w}^{t-1} x_i$$

$$\mathbb{1}_{jump}(t) = \mathbb{1}\left(\frac{|x_t - \mu_t|}{|\mu_t| + \epsilon} > \delta\right)$$

where w is the window size and δ is the relative change threshold, which are defined in Appendix A.

The third rule was transient event detection. Identified short-lived isolated activations were detected and a binary presence signal p_t was defined for each event type. A transient anomaly was flagged when

$$\mathbb{1}_{transient}(t) = \mathbb{1}(p_t \text{ is isolated within window } w \text{ and run length } \leq l_{max})$$

Final anomaly decision followed:

$$\text{anomaly}(t) = \begin{cases} 1 & \text{if } S_t \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

In the implementation used in this thesis the weight from each anomaly rule was equal and was alone enough to exceed the threshold, meaning that one rule breach was enough to mark the frame as anomalous. Detected anomaly frames had its values removed in downstream modeling to avoid propagating corrupted measurements.

3.3.2 Data imputation

To minimize data loss, missing values in the features that were missing or removed due to anomalies, were reconstructed through linear interpolation when they stretched over less than 10 frames. This was done using the closest previous and subsequent valid measurement. When detected anomalies stretches for 10 frames or longer, the measurements was not interpolated and the time-series are left untouched.

3.3.3 Feature Calculation and Data Synchronization

To prepare the data for model training the detections from the CV-pipeline needed to be merged and synchronized with the measurements from the process. The merged data will then also have to be split into different parts to allow for proper model training and validation.

Table 3.1: Target features

Target feature	Definition
Wire tip to weld pool	$\lvert \text{wire tip}_y - \text{weld pool}_y \rvert$
Tapering point to weld pool	$\lvert \text{tapering\_point}_y - \text{weld pool}_y \rvert$
Plasma attachment to weld pool	$\lvert \text{plasma\_attachment\_height} - \text{weld pool}_y \rvert$
Contact tip to weld pool	$\lvert \text{contact\_tip}_y - \text{weld pool}_y \rvert$
Plasma channel width	Average distance between detected edges
Arc detected	Arc detected between wire tip and weld pool
Droplet detected	Droplet detected in the image
Droplet diameter	Droplet diameter
Spatter detected	Spatter detected in the image

The resulting targets was saved as a Comma-Separated Values (CSV) file to accommodate for subsequent steps.

Each video sequence has a corresponding HDF5 file that contains the analog inputs; voltage, current and Wire Feed Speed (WFS), together with mathematical data, filtered voltage, current and WFS, calculated power, etc., and the frames captured, all in relation to time.

Due to the difference in the sampling frequency between the measurement data and the video frame rate the data was merged into two datasets: one where the detections from the videos were up-sampled through linear interpolation to match the measurement frequency, and one where the measurements were down-sampled through averaging to match the video frame rate.

The down-sampled dataset were constructed by averaging all the measurements connected to each frame. To find which measurements that is connected to each frame, upper and lower time bounds is calculated for each frame. The first and last frame had upper and lower bounds of $-\infty$ and ∞, respectively. The upper and lower bounds of a frame, $frame_t$ was calculated accordingly:

$$
\begin{aligned}
lower\_bound &= frame_t - \frac{frame_t - frame_{t-1}}{2} \\
upper\_bound &= frame_t + \frac{frame_{t+1} - frame_t}{2}
\end{aligned}
$$

The up-sampled dataset was created to not lose any information within the measurement time series. The actual output from the computer vision analysis is placed on the row that has the closest measurement time to when the frame was captured. Then each frame value is linearly interpolated up until the next true output from the subsequent frame.

In the merging process, the pixel measurements from the computer vision analysis are converted to mm and used to calculate the relevant distances, such as wire tip, tapering point, contact tip, and plasma attachment height to the weld pool, as well as arc width.

3.4 Estimation of Target Features

Different estimation and prediction models were developed and benchmarked to investigate their performance in predicting target features in the welding process. The models span three domains of increasing complexity: statistical methods, machine learning, and deep learning.

3.4.1 Data Curation

To accommodate optimal training, each full welding sequence was chronologically split into subsequences of equal length. The subsequences were then randomly divided into a training, validation, and testing parts of 60%, 20%, and 20%, respectively. To reduce direct temporal leakage between the subsequences, no subsequences had any overlap, and a predefined gap was set between the end of a sequence and the start of a subsequent one. If the last chunk in a sequence was shorter than a predefined threshold, it was discarded.

The split algorithm also extracted machine type, CTWD, and WFS from the file names and included these variables as static context in the subsequence manifest. These variables were used to balance the training, validation, and test splits with respect to machine type, CTWD, WFS, and number of samples. CTWD was only used for split balancing and was not included as an input feature in the estimation models. All parameters used for the data split are listed in Appendix B.

3.4.2 Feature selection

A predefined feature configuration was established through iterative model training and evaluation. This design enforced a consistent input structure across all models and enabled direct comparison. The input space was organized into three feature groups:

- **Measured and calculated signals:** voltage (V), current (I), voltage change (ΔV), current change (ΔI), power (P)
- **Rolling window statistics:** mean, standard deviation, slope, minimum, maximum
- **Metadata:** WFS and machine type

Feature construction was performed on two levels. At the first level, time series data were converted into fixed-length sliding windows of size w. Window sizes varied across experiments to evaluate sensitivity to temporal context length. For a window size w, the inputs for one of the measured or calculated signals at time t were defined as

$$\mathbf{a}_{t,w} = [a_t, \ldots, a_{t-w+1}]$$

where $\mathbf{a}_{t,w}$ represents the sliding window vector for signal a at timestep t. At the second level, statistical descriptors were computed over a specified window w_s:

$$\bar{a}_{t,w_s} = \frac{1}{w_s}\sum_{i=0}^{w_s-1} a_{t-i} \qquad \sigma_{a,t,w_s} = \sqrt{\frac{1}{w_s}\sum_{i=0}^{w_s-1}\left(a_{t-i} - \bar{a}_{t,w_s}\right)^2} \qquad m_{a,t,w_s} = \frac{a_t - a_{t-w_s+1}}{w_s - 1}$$

where $\bar{a}_{t,w_s}$, σ_{a,t,w_s} and m_{a,t,w_s} are the mean, standard deviation and slope of a at time t over the window w_s, respectively. The input vector was then set as:

$$\mathbf{x}_t = \begin{bmatrix}\mathbf{x}_1 & \cdots & \mathbf{x}_{16}\end{bmatrix}^{\mathsf{T}} \tag{3.29}$$

with

$$\begin{aligned}
&\mathbf{x}_1 = \mathbf{I}_{t,w}, && \mathbf{x}_2 = \mathbf{V}_{t,w}, && \mathbf{x}_3 = \Delta\mathbf{I}_{t,w}, && \mathbf{x}_4 = \Delta\mathbf{V}_{t,w},\\
&\mathbf{x}_5 = \mathbf{P}_{t,w}, && x_6 = \bar{I}_{t,w_s}, && x_7 = \bar{V}_{t,w_s}, && x_8 = \sigma_{I,t,w_s},\\
&x_9 = \sigma_{V,t,w_s}, && x_{10} = m_{I,t,w_s}, && x_{11} = m_{V,t,w_s}, && x_{12} = \min(I_{t,w_s}),\\
&x_{13} = \min(V_{t,w_s}), && x_{14} = \max(I_{t,w_s}), && x_{15} = \max(V_{t,w_s}), && x_{16} = \text{WFS}
\end{aligned} \tag{3.30}$$

The prediction target corresponded to the target value at the final timestep of that sequence:

$$\hat{y}_t = f(\mathbf{x}_t)$$

3.4.3 ARX

First a baseline ARX was implemented, as described in Equation 2.1 with the input defined in Equation 3.29 where $n_b = w = w_s$. The model was fitted to predict the wire tip-to-weld pool distance using both a one-step-ahead prediction, where the true output history was used, and in a freerun simulation, where the model used its own previous predictions as the output history. To find the model with the best results a grid search was performed over

$$n_a, n_b \in [0, 1, 3, 5, 10, 25, 50], \qquad n_k \in [0, 5]$$

For each candidate model order, the regressor matrix was constructed from the training subsequences and the ARX coefficients were estimated using ordinary least squares. The final model order was selected by minimizing the RMSE of the validation freerun closed-loop simulation.

3.4.4 NARX

A NARX model as described in Equation 2.2, was implemented and benchmarked together with an adaptation of the NARX model which predicted the difference $\Delta y = F(z(k)) + e(k)$ instead of the actual target value, making the prediction model:

$$\hat{y}(k) = y(k-1) + F(z(k)) + e(k)$$

where the initial value for $y(k-1)$ is set as the true previous output in the sequence.

The model can be adjusted by modifying the number of previous inputs and outputs the model uses, defined by n_y, n_u, and also by adjusting the input delay d. When fitting the model, additional parameters determine the model complexity and regularization strength. The degree n of the polynomial function F determines the model order, and a variable α defines the regularization strength when estimating the coefficients β. To find the best setup of parameters, a parameter sweep was performed, trying every combination of $n_a, n_b \in [0, 1, 3, 5, 10, 25, 50]$, $d \in [0, 3]$, and $n = 2$.

3.4.5 XGBOOST

XGBoost regression was used for multivariate time-series prediction. The implementation used `XGBRegressor` from the `xgboost` library together with `MultiOutputRegressor` from `scikit-learn`. One independent regression model was trained for each target variable.

Each temporal window was flattened into a one-dimensional vector before training. The model therefore learned from lagged feature representations rather than recurrent sequence states.
XGBoost constructs an additive ensemble of regression trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F}$$

where f_k represents an individual regression tree and K denotes the total number of boosting iterations. Training minimized the squared error objective

$$\mathcal{L} = \sum_i (y_i - \hat{y}_i)^2$$

The model uses 600 estimators, a maximum tree depth of 6, learning rate of 0.05, row sub-sampling ratio of 0.8, and feature sub-sampling ratio of 0.8.

3.4.6 Parallel ensemble

Parallel ensemble methods were also evaluated for the estimation of the wire tip and tapering to the weld pool distance. Both Random Forest and Extra Trees regressors were investigated due to their ability to model nonlinear relationships while remaining relatively robust to noisy and high-dimensional input features. The model inputs consisted of measurements, intermediate temporal features derived from these signals, and setpoints for machine type and WFS.

A grid search was performed over a selection of window sizes in past inputs and past calculated features, $[0, 5, 10, 25, 50]$, and the rolling window summaries, $[0, 25, 50, 75, 100]$. All evaluated configurations used 160 trees.

The maximum tree depth was fixed at 24 and the minimum leaf size at 8. These parameters were selected to regularize the ensemble and reduce overfitting by limiting excessively deep decision paths and preventing predictions based on very small subsets of the training data.

3.4.7 Neural NARX

A neural nonlinear autoregressive model with exogenous inputs (NN-NARX), as described in Equation 2.3, was evaluated for estimation of the wire tip to weld pool and tapering point to weld pool distances. Instead of directly predicting the absolute distance, the network was trained to predict the change in distance between consecutive samples, from which the absolute position was reconstructed recursively. This formulation was used to reduce the influence of slow signal drift and encourage the model to learn local temporal dynamics.

The model inputs consisted of measured current and voltage signals together with engineered features derived from these measurements. Rolling means, standard deviations, and slopes were calculated using a temporal window of 25 samples. Static context variables representing the wire feed speed (WFS) and machine type were also included.

The model used the present value and the preceding 10 values of each dynamic input feature, together with the preceding three values of the target distance. The network architecture consisted of two fully connected hidden layers with 64 and 32 neurons, respectively, using Rectified Linear Unit (ReLU) activation functions and a dropout rate of 0.15. A relatively shallow architecture was selected to limit overfitting, given the limited number of welding sequences available for training.
Training was performed for a maximum of 200 epochs using the AdamW optimizer. Early stopping based on validation loss was applied with a patience of eight epochs to reduce overfitting and prevent unnecessary training after convergence.

3.4.8 LSTM

An LSTM network was used for multivariate time-series prediction. The model was implemented in `PyTorch` using a two-layer LSTM architecture followed by fully connected regression layers. Unlike the XGBoost approach, the LSTM processed sequential data directly without flattening the temporal structure.

The LSTM sequentially updated an internal hidden state across timesteps according to

$$h_t = f(x_t, h_{t-1})$$

where h_t denotes the hidden state at timestep t. The hidden representation from the final timestep was passed through fully connected layers to generate the output prediction.

The network used two stacked LSTM layers with hidden size 64 and dropout rate 0.2. A fully connected regression head consisting of linear layers and ReLU activation was applied after the recurrent layers. Model optimization used the Adam optimizer with learning rate 1×10^{-3}, weight decay 1×10^{-4}, and mean squared error loss:

$$\mathcal{L} = \sum_i (y_i - \hat{y}_i)^2$$

Input features and target variables were standardized using `StandardScaler` prior to training. Training was performed using mini-batches of size 256 for a maximum of 25 epochs. Early stopping with patience of 5 epochs was applied using validation RRSE to reduce overfitting.

3.4.9 Evaluation

Model performance was evaluated on training, validation, and test splits using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Relative Root Squared Error (RRSE), and Pearson correlation coefficient.

A sliding window formulation was used for both models. For window size w, input sequences were defined as

$$X_t = [x_{t-w+1}, \ldots, x_t]$$

with target prediction at the final timestep:

$$\hat{y}_t = f(X_t)$$

Window sizes were evaluated over

$$w \in \{5, 10, 15, 20, 25, 30, 40, 50, 75, 100\}$$

and average RMSE across targets was used to select optimal window length.

For XGBoost, each model consists of an additive ensemble of regression trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

trained by minimizing squared error loss:

$$\mathcal{L} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

For LSTM, a two-layer recurrent network was used. The hidden state evolves as

$$h_t = \text{LSTM}(x_t, h_{t-1})$$

The final hidden state is mapped to outputs through fully connected layers and mean squared error loss was used.

Feature importance

Feature importance is evaluated after training for all models and is estimated through sensitivity-based analysis, where input perturbations are linked to prediction changes. Results are aggregated across output targets and across all window sizes to produce a unified feature ranking. The average of the feature importance across all models are calculated and presented as chart in later stages.
Feature importance is computed from total loss reduction across tree splits:

$$I_j = \sum_{k=1}^{K} \sum_{s \in S_{j,k}} \Delta \mathcal{L}_s$$

Predictions were further analyzed using line plots and scatter plots over a subset of 500 samples per target.

4

Results

4.1 Data acquisition

Table 4.1: Summary of recorded sequences and their process configurations

Machine	CTWD (mm)	WFS (m/min)
Machine 1	20	5
Machine 1	17	10
Machine 1	12	10
Machine 1	20	10
Machine 1	17	10
Machine 1	18	15
Machine 1	15	15
Machine 2	15	5
Machine 2	20	5
Machine 2	10	5
Machine 2	10	6
Machine 2	15	6
Machine 2	13.5	10
Machine 2	20	10
Machine 2	12	10
Machine 2	12	7
Machine 2	20	13
Machine 2	20	16
Machine 2	20	16

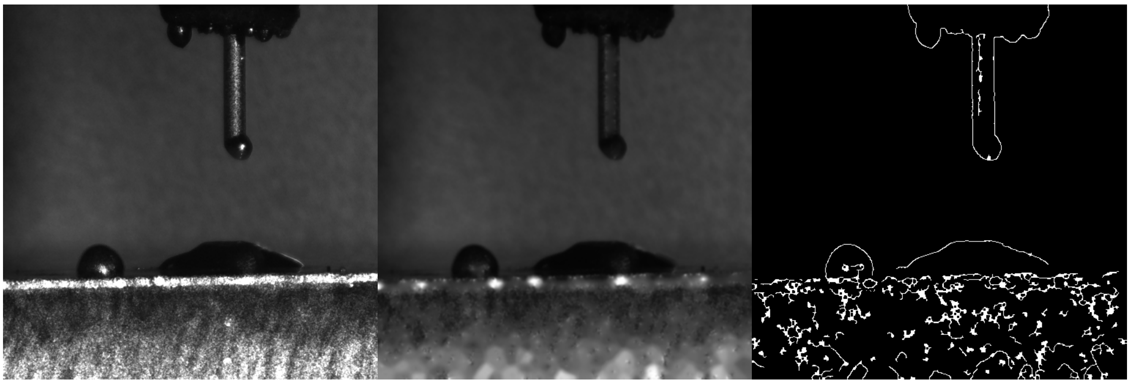
The different parameter configurations used in the sequences recorded in the data acquisition process are summarized in Table 4.1. Machine 1 is from ESAB and Machine 2 is from competitor.

4.2 Computer Vision-Based Feature Extraction

Below are the results from the computer vision analysis.

4.2.1 Preprocessing

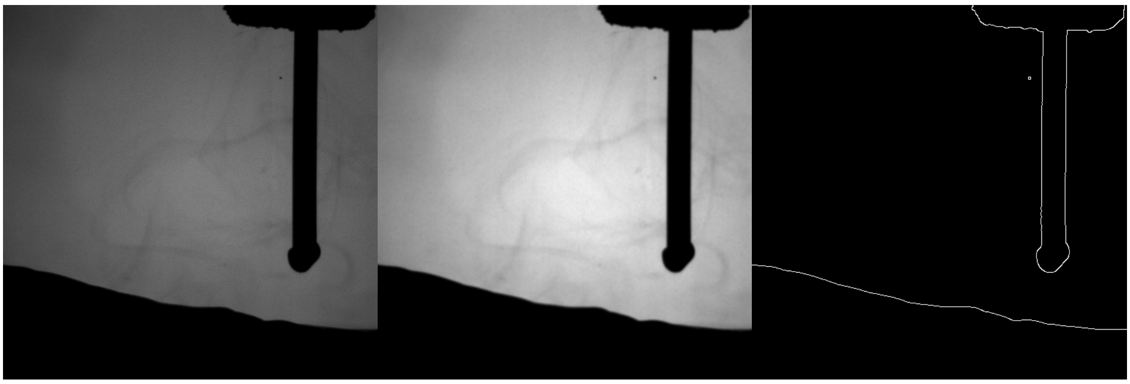
The preprocessing results of a laser-lit frame including the original image, the preprocessed image and the Canny edge map can be viewed in Figure 4.1.



(a) Original **(b)** Preprocessed **(c)** Canny edges

Figure 4.1: Preprocessing of laser-lit image

The preprocessing results of a backlit frame including the original image, the preprocessed image and the Canny edge map can be viewed in Figure 4.2.



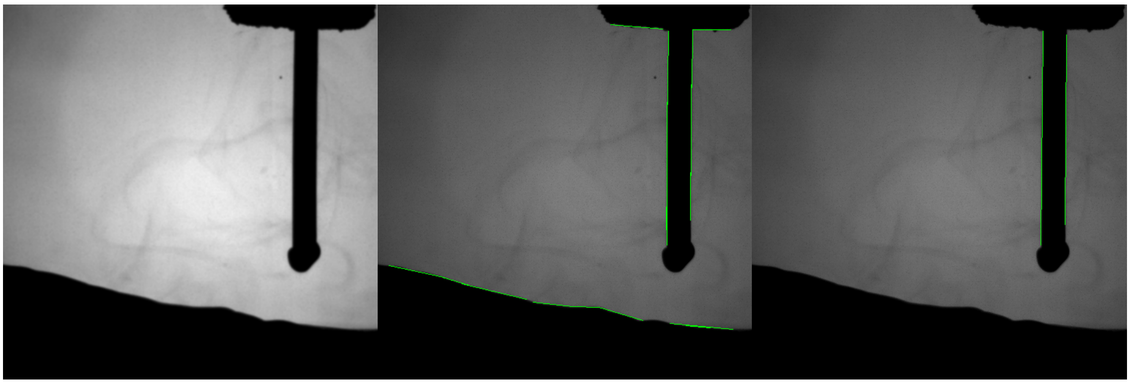
(a) Original **(b)** Preprocessed **(c)** Canny edges

Figure 4.2: Preprocessing of backlit image

4.2.2 Feature Extraction

Wire identification

When the `HoughlinesP` operation is performed on the full preprocessed image, Figure 4.3a the result is shown in Figure 4.3b.



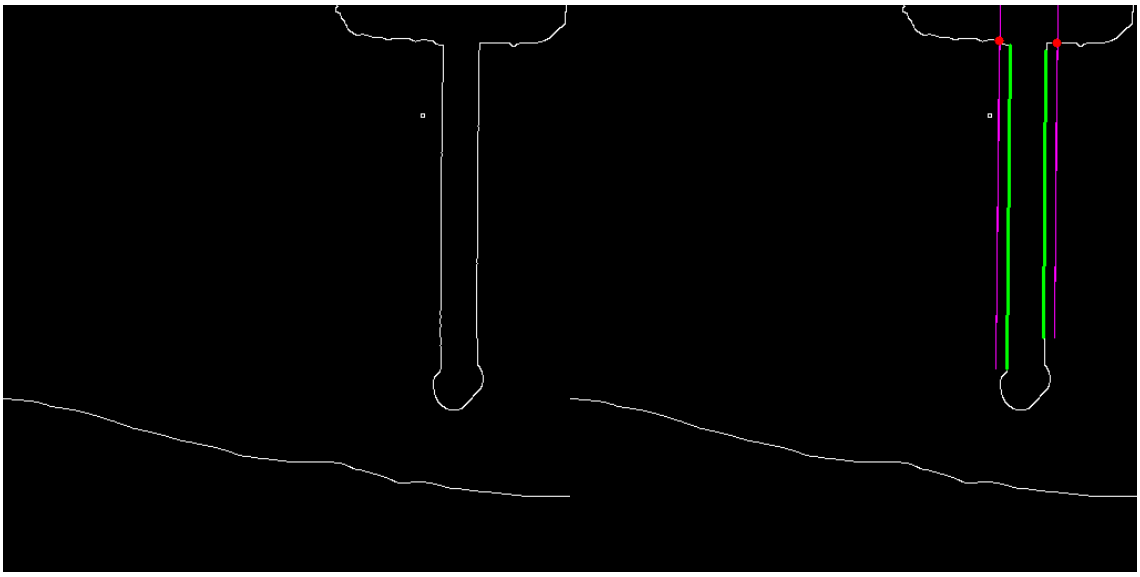
(a) Original **(b)** All detected Houghlines **(c)** Chosen wire edges

Figure 4.3: Houghline detection for finding the wire edges

All detected lines are then passed through a function that scores pairs of lines where parallelism is favored. The result from this operation performed on Figure 4.3b can be viewed in Figure 4.3c.

Search-Space Definition

The wire edges in Figure 4.4a is then used to find the contact tip by moving the lines outward from the wire center, extending them upward and performing a search in the Canny edge map to find the contact tip edge on both sides of the wire, as shown in Figure 4.4b.



(a) Canny edges **(b)** Contact tip detection

Figure 4.4: Contact tip detection

The bottom of the search-space is then chosen as the base metal surface. The orientation of the search-space is chosen based on the angle of the wire, defined as the angle of the longest wire edge. The resulting search-space box can be viewed in Figure 4.5.

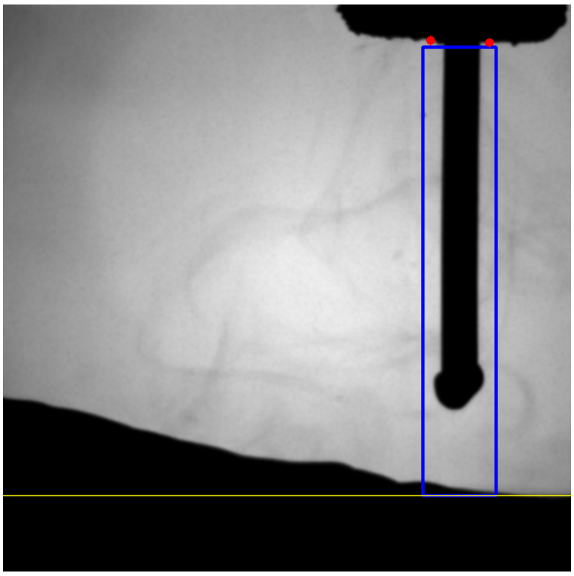


Figure 4.5: Searchspace visualization: searchspace box (blue), contact tip points (red), basemetal level (yellow)

Wire tip

The location of the wire tip is chosen as the lowest point of the best fitting wire contour. The best fitting contour with the lowest point marked as the wire tip can be viewed in Figure 4.6.

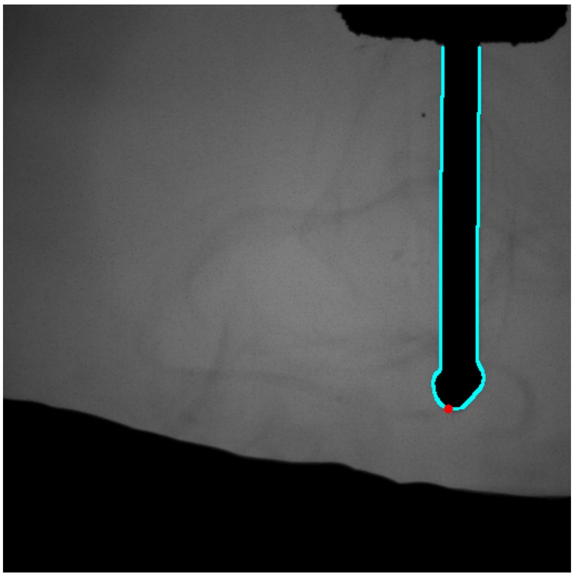
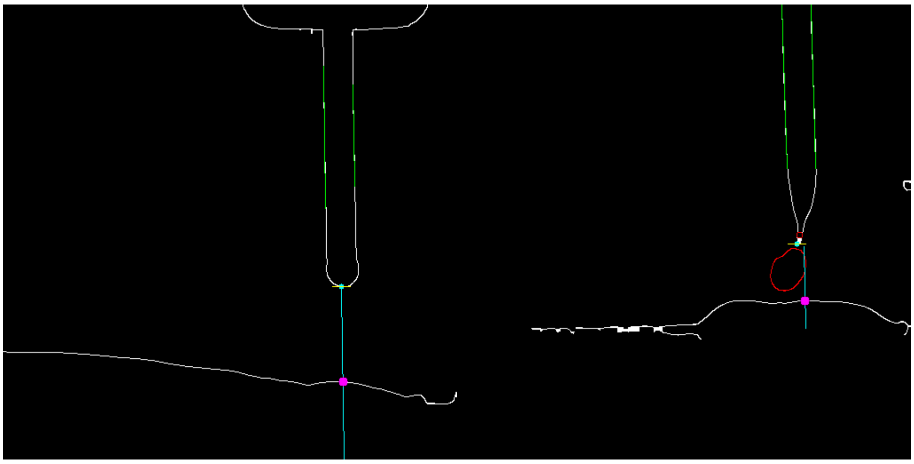


Figure 4.6: Contour with the wire tip

Weld pool

The weld pool found using the Canny edge map of the search-space and a wire centerline, concluded from the wire edges, as shown in Figure 4.7a. An image with droplets in between the wire tip and weld pool, in which the detection algorithm removes the droplet edges and correctly identifies the weld pool can be viewed in Figure 4.7b.

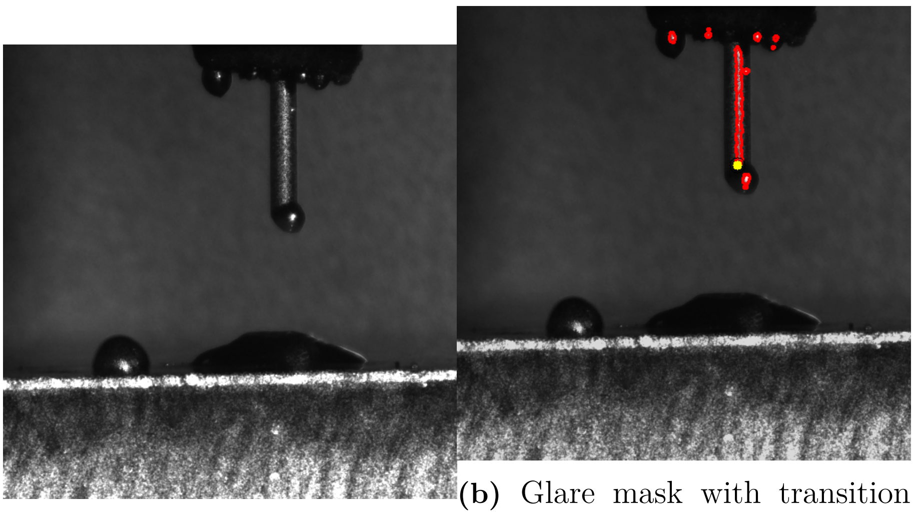


(a) Without droplets **(b)** With droplets

Figure 4.7: Weld pool detection

Solid-molten transition

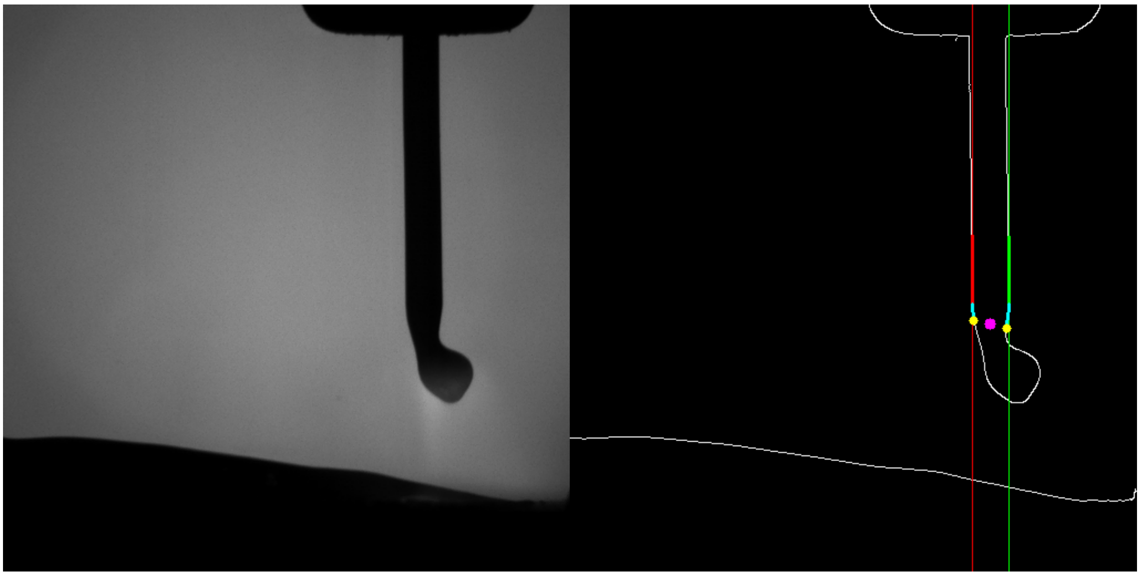
To find the solid-molten transition using the texture change of the wire surface a glare mask is applied to Figure 4.8a. The glare mask found on the laser-lit image can look like as shown in Figure 4.8b.



(a) Original image **(b)** Glare mask with transition point

Figure 4.8: Solid molten transition point based on wire texture

The tapering point is found by identifying where the wire edge deviates outside a threshold over multiple consecutive steps along the extended wire edges, as shown in Figure 4.9b.



(a) Original image **(b)** Tapering point detection

Figure 4.9: Solid molten transition point based on wire tapering

Arc characteristics

The preprocessing of the arc is different than for the other feature extractions, and the image was kept closer to the raw footage. Then the arc ROI is extracted from the image, as shown in Figure 4.10.

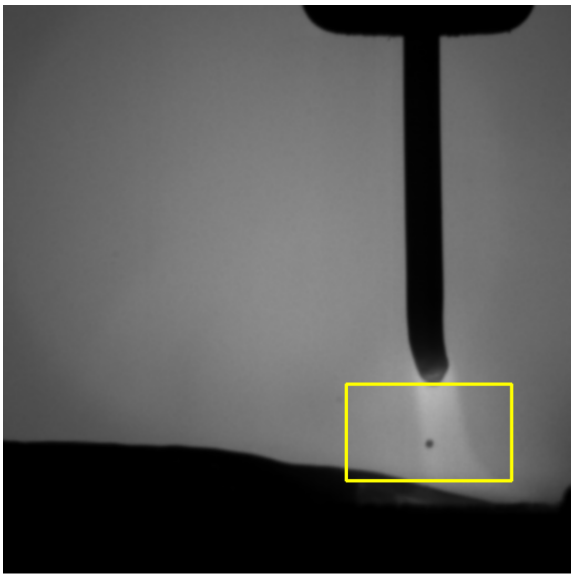
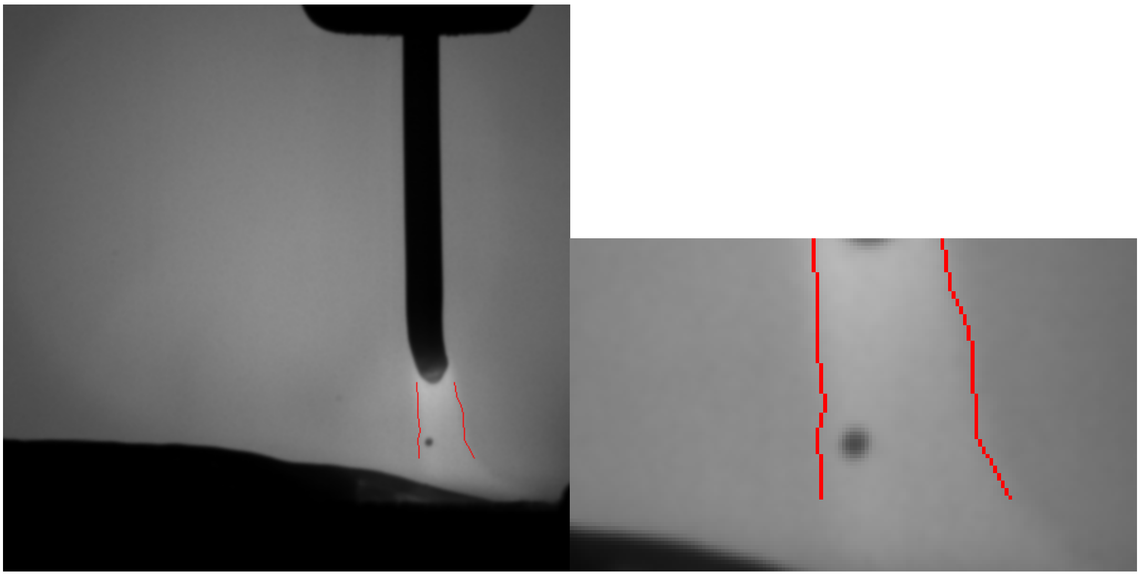


Figure 4.10: ROI of arc marked in original image

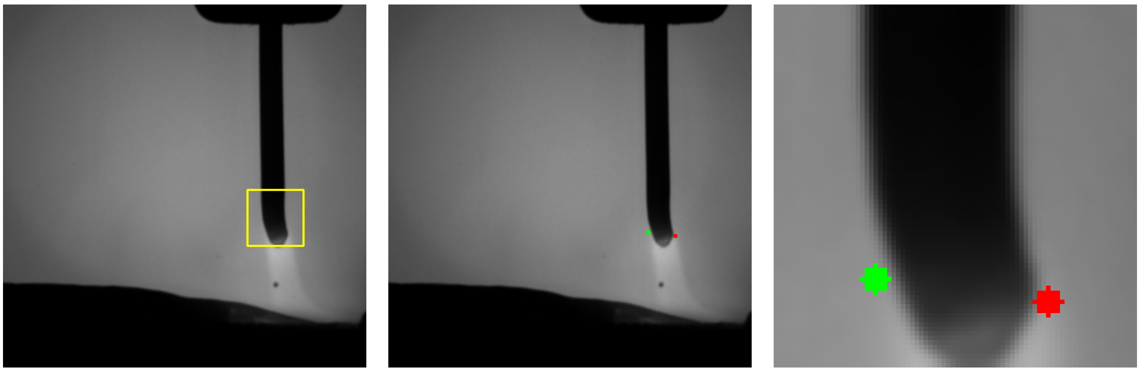
The results from the plasma channel detection can then be viewed in Figure 4.11, where (a) is the edge overlay on the full image and (b) on the ROI.



(a) Full image **(b)** ROI

Figure 4.11: Plasma channel edges, typically called as arc wall

The attachment ROI is also extracted from the image and the results after the analysis is as shown in Figure **??**.



(a) Full image with ROI search box **(b)** Plasma attachment points **(c)** Zoomed in view of plasma attachment points

Figure 4.12: Plasma attachment points detection flow

4.2.3 Droplets and Spatters

This section presents the tracking performance of droplets and spatters across the analyzed video sequence. The results are characterized by a combination of spatial trajectory data and instantaneous velocity profiles derived from high-speed backlit imaging.

Qualitative Tracking Visualization

Figure 4.13 displays a representative frame from the tracking sequence. The backlit configuration allows for precise localization of the wire tip, the tapering point, and the resulting metal transfer. The tracking algorithm distinguishes between primary droplets, which follow a predictable path toward the weld pool, and smaller spatters, which often exhibit more erratic or high-velocity motion. The visualization confirms the alignment of the droplet **D7** along the central axis of the electrode.

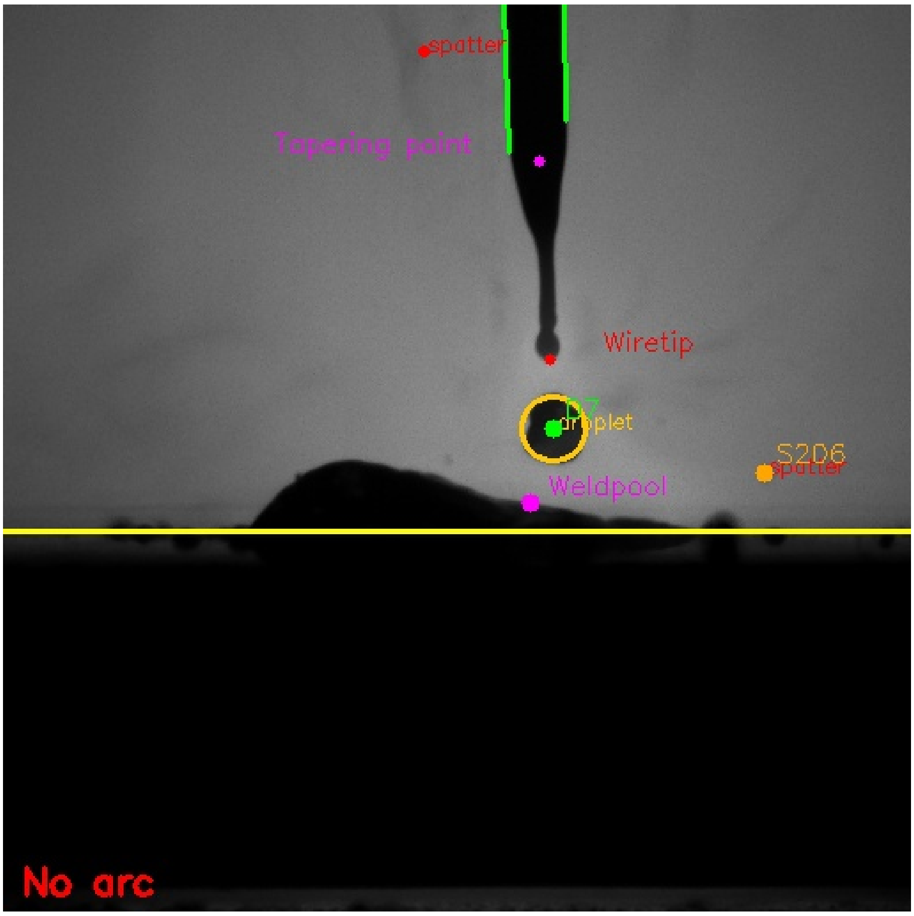


Figure 4.13: Backlit tracking visualization showing the detection of the wire tip, tapering point, spatters, and droplet D7 moving toward the weld pool.

Trajectory Analysis

The spatial evolution of the tracked features is plotted in Figure 4.14. The x-position plot reveals that most droplets (D0–D9) maintain a very stable lateral position, centered roughly between 375 and 385 pixels, indicating a highly directional spray transfer.

As the y-axis of the image coordinates are inverted, the position of the objects may seem going upwards, but instead it is actually going downward. The y-position plot shows distinct linear slopes for each droplet, representing their downward flight. Notably, spatter track S0 (indicated by the dashed line) shows a significantly steeper and more prolonged vertical displacement compared to the droplets, suggesting a different ejection mechanism or higher initial momentum.

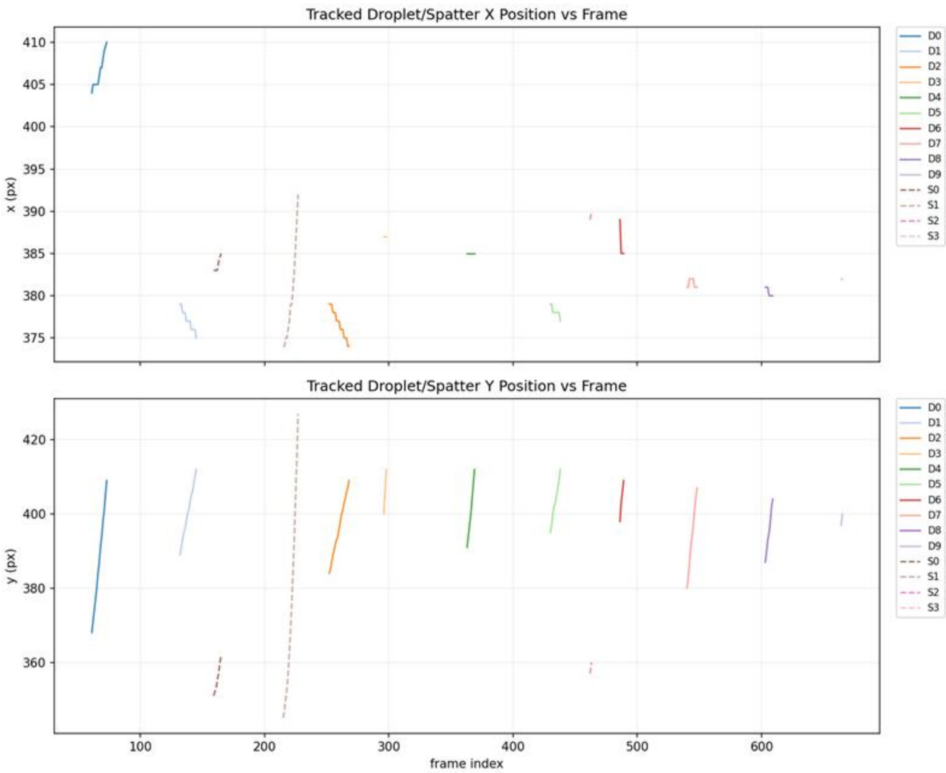


Figure 4.14: Temporal evolution of x and y coordinates of droplets and spatters. Droplets show consistent lateral stability, while vertical slopes indicate steady downward acceleration.

Velocity Analysis

Instantaneous velocity profiles are calculated from frame-to-frame displacement, as shown in Figure 4.15. The velocity data reveals a clear distinction between object types:

- **Droplets (D0–D9):** Exhibit relatively low and stable velocities, generally ranging between 1 and 4 px/frame.
- **Spatters (S0):** Track S0 shows a dramatic velocity spike, accelerating from 2 to over 12 px/frame in a very short duration (frames 200–230).

The high-frequency oscillations observed in the droplet velocity curves (e.g., D2 and D3) are likely indicative of sub-pixel tracking noise or slight shape deformations (oscillations) of the molten metal during flight.

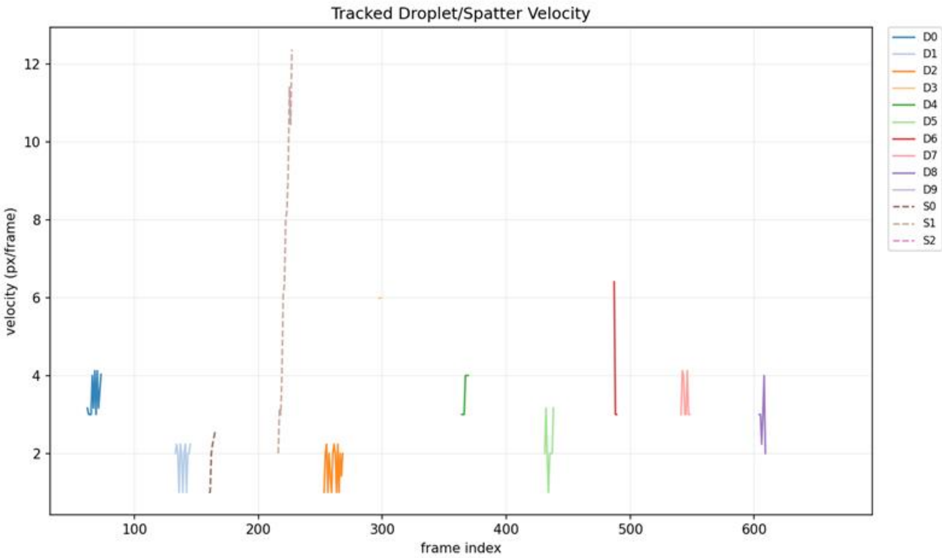


Figure 4.15: Velocity profiles of droplets and spatters, showing the contrast between steady droplet motion and the high-velocity excursion of spatter S0.

4.3 Data Preparation

The anomaly detection results can be viewed in Table 4.2.

Table 4.2: Anomaly detection results

Machine	CTWD (mm)	WFS (m/min)	Total frames	Anomaly frames	Share (%)
2	12	10	19 974	199	1.00
2	13	10	42 284	921	2.18
2	20	10	41 204	393	0.95
2	20	13	41 857	2 235	5.34
2	20	16	42 014	3 836	9.13
2	20	16	18 473	1 945	10.53
2	10	5	19 923	76	0.38
2	15	5	20 225	288	1.42
2	20	5	20 234	653	3.23
2	20	5	20 081	129	0.64
2	10	6	19 944	1 972	9.89
2	15	6	39 669	1 024	2.58
2	15	6	38 190	495	1.30
2	12	7	40 772	1 590	3.90
1	12	10	41 897	1 784	4.26
1	17	10	41 609	569	1.37
1	17	10	41 462	866	2.09
1	20	10	41 703	796	1.91
1	15	15	41 657	5 101	12.24
1	15	15	41 695	2 532	6.07
1	20	5	41 948	468	1.12
Both	All	All	716 815	27 872	3.88

All anomaly reasons in descending order can be viewed in Table 4.3, multiple anomalies can exist within the same frame. The variable mm_to_px corresponds to missing wire edges, which makes the pipeline abort the full analysis of the specific frame.

Table 4.3: Anomaly reasons

Anomaly reason	Count	Share of anomaly reason
Jump: weldpool_y	13 308	44.57%
Missing: mm_to_px	11 582	38.80%
Missing: weldpool_y	4 966	16.63%
Jump: mm_per_px	1	0.0%

After the interpolation, when fewer than 10 consecutive frames were flagged due to missing or abnormal values, the portion of missing data was reduced to 1.49%, see Table 4.4.

Table 4.4: Anomaly interpolation results

Total frames	Total anomaly frames	Anomaly share
716 815	10 694	1.49%

4.4 State estimation

Below are the state estimation results for each of the evaluated models.

4.4.1 ARX

The selected model for both the wire tip to weld pool distance and the tapering point to weld pool distance excluded previous outputs from the input space, and used 50 previous inputs without delay. A line plot of the predictions and true values in a sample can be viewed in Figure 4.16. A box plot of the residuals for a selection of buckets can be viewed in Figure 4.17. At last, a scatter plot of the predictive performance of both models on all sets is available in Figure 4.18.

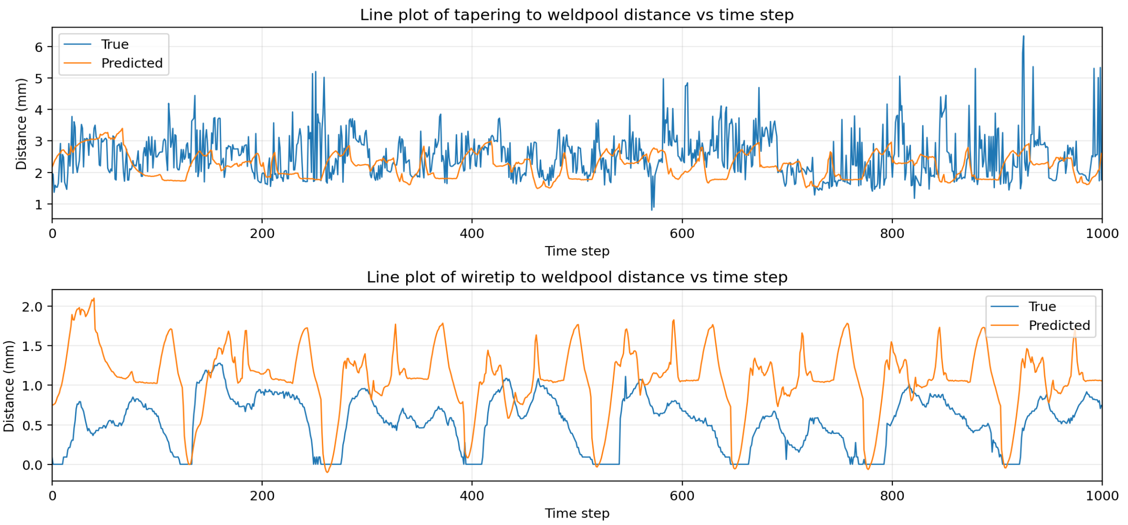


Figure 4.16: Line plot of ARX model outputs across a test subsample

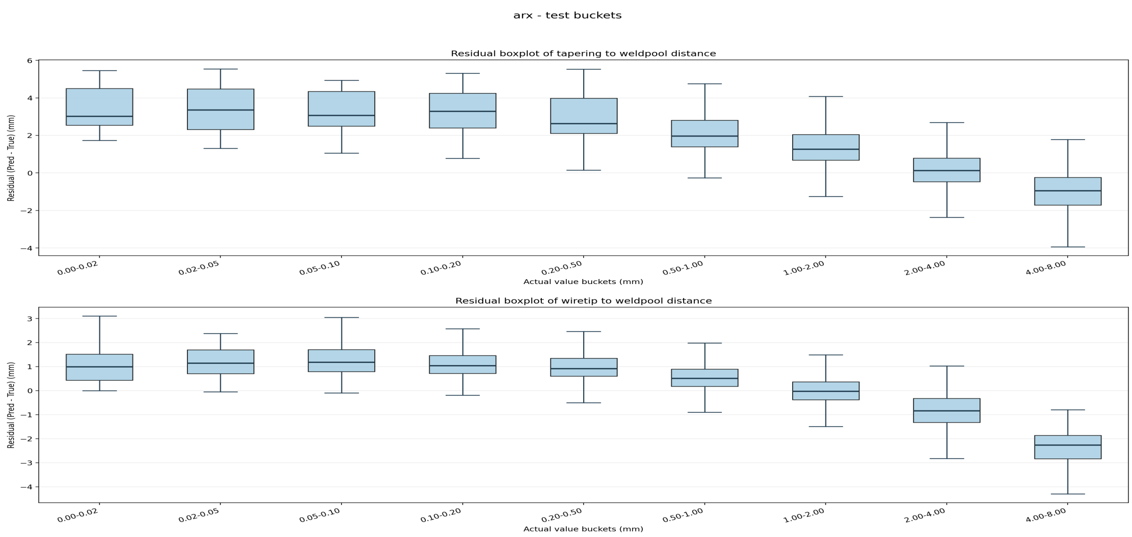
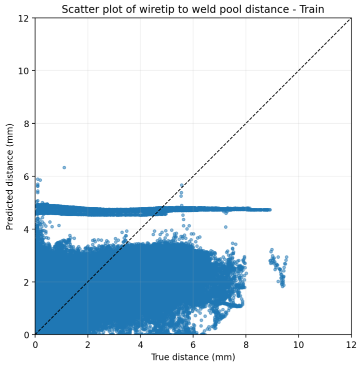
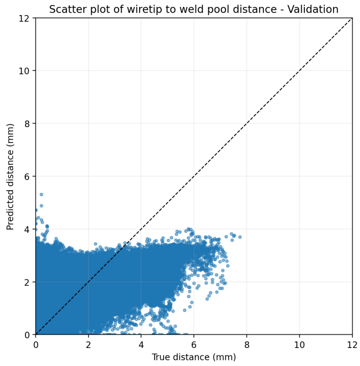


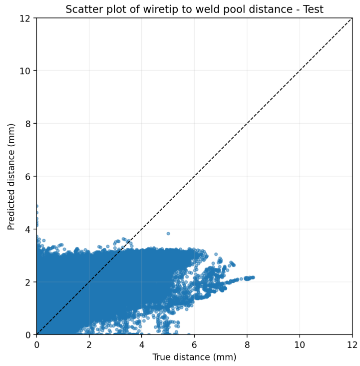
Figure 4.17: Box plot of the ARX model performance on the test data



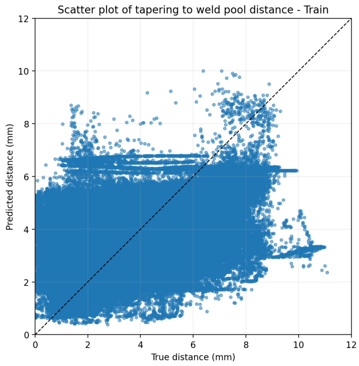
(a) Train



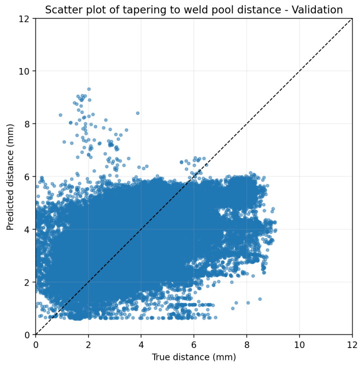
(b) Validation



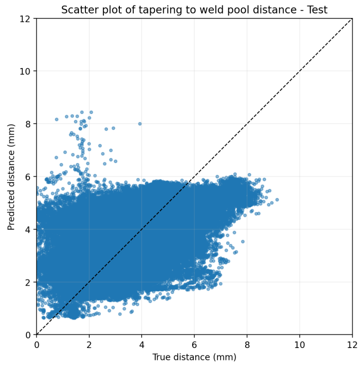
(c) Test



(d) Train



(e) Validation



(f) Test

Figure 4.18: Scatter plots of predicted versus measured distances by ARX. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

4.4.2 NARX

The selected model for both the wire tip to weld pool distance and the tapering point to weld pool distance excluded previous outputs from the input space, was of the second degree, and used 50 previous outputs without input delay. A line plot of the predictions and true values in a sample can be viewed in Figure 4.19. A box plot of the residuals for a selection of buckets can be viewed in Figure 4.20. At last, a scatter plot of the predictive performance of both models on all sets is available in Figure 4.21.

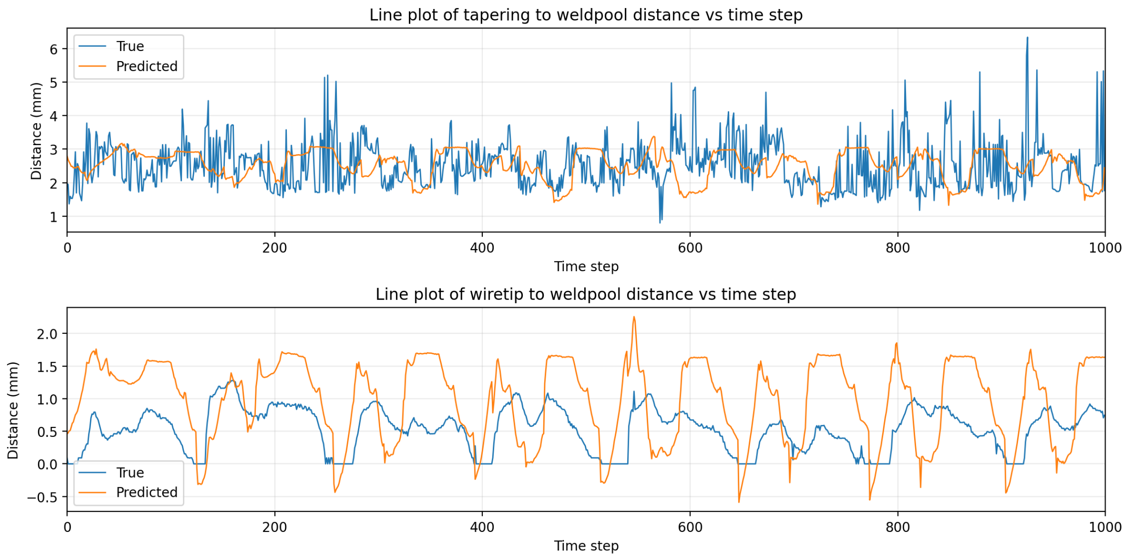


Figure 4.19: Line plot of NARX model outputs across a test subsample

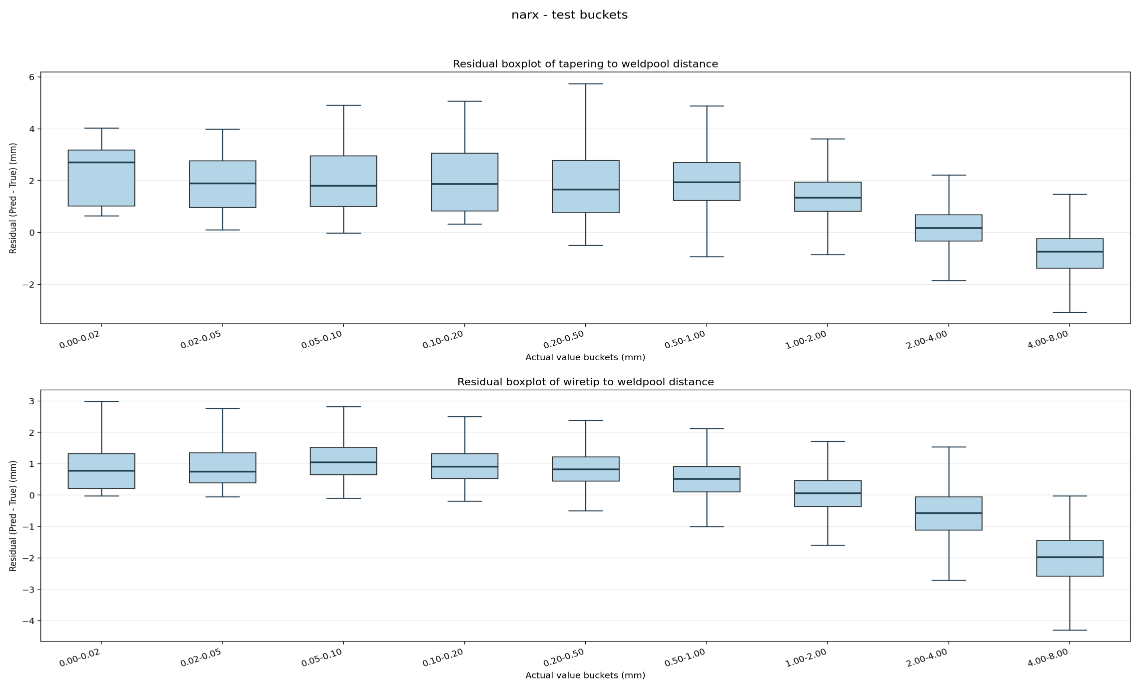
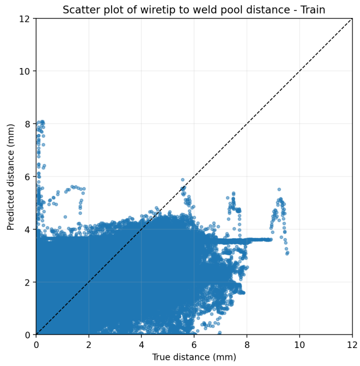
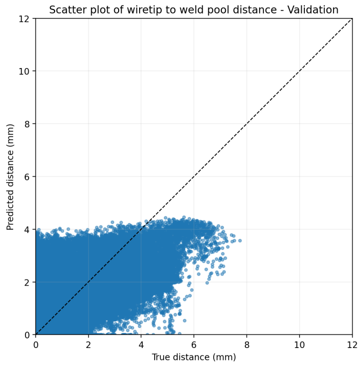


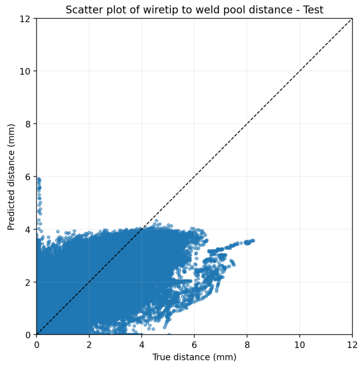
Figure 4.20: Box plot of the NARX model performance on the test data



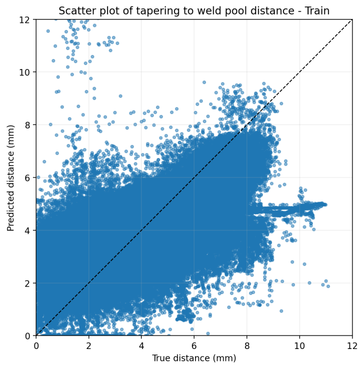
(a) Train



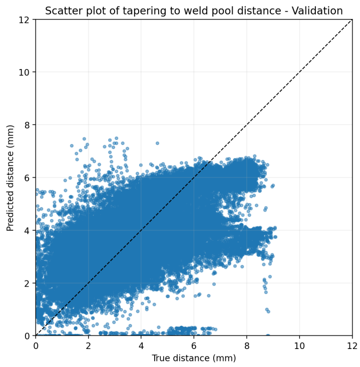
(b) Validation



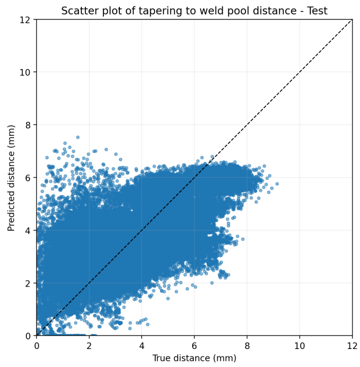
(c) Test



(d) Train



(e) Validation



(f) Test

Figure 4.21: Scatter plots of predicted versus measured distances by NARX. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

4.4.3 XGBoost

This section includes scatter plots across train, test, and validation splits used across all the models, along with line plot over a sample of test data and a box plot for all test results for the output features.

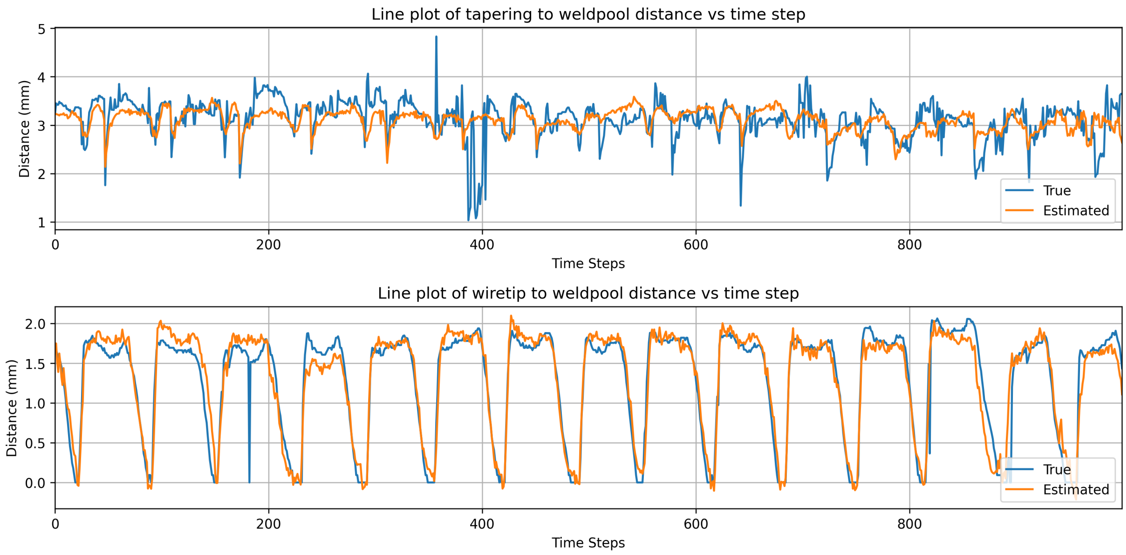


Figure 4.22: Line plot of XGBOOST model outputs across a test subsample

The line plot of both outputs in Figure 4.22 shows that the mean value and trend are captured well, but the model struggles with sudden, abrupt changes.

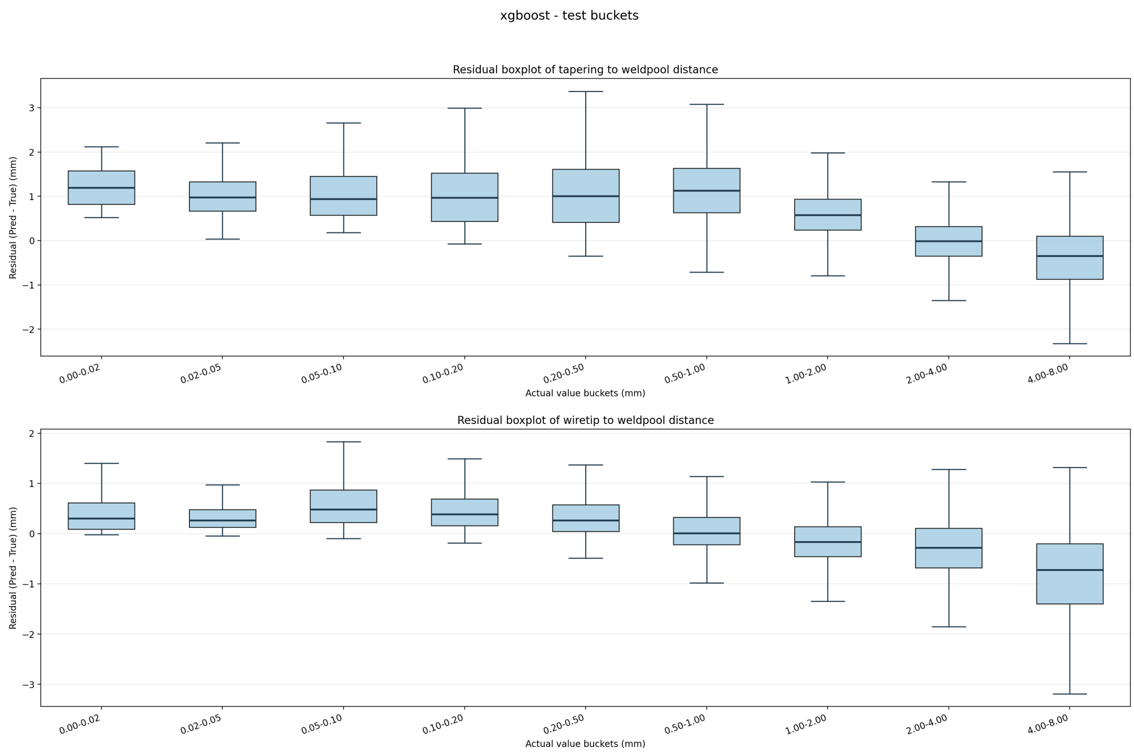
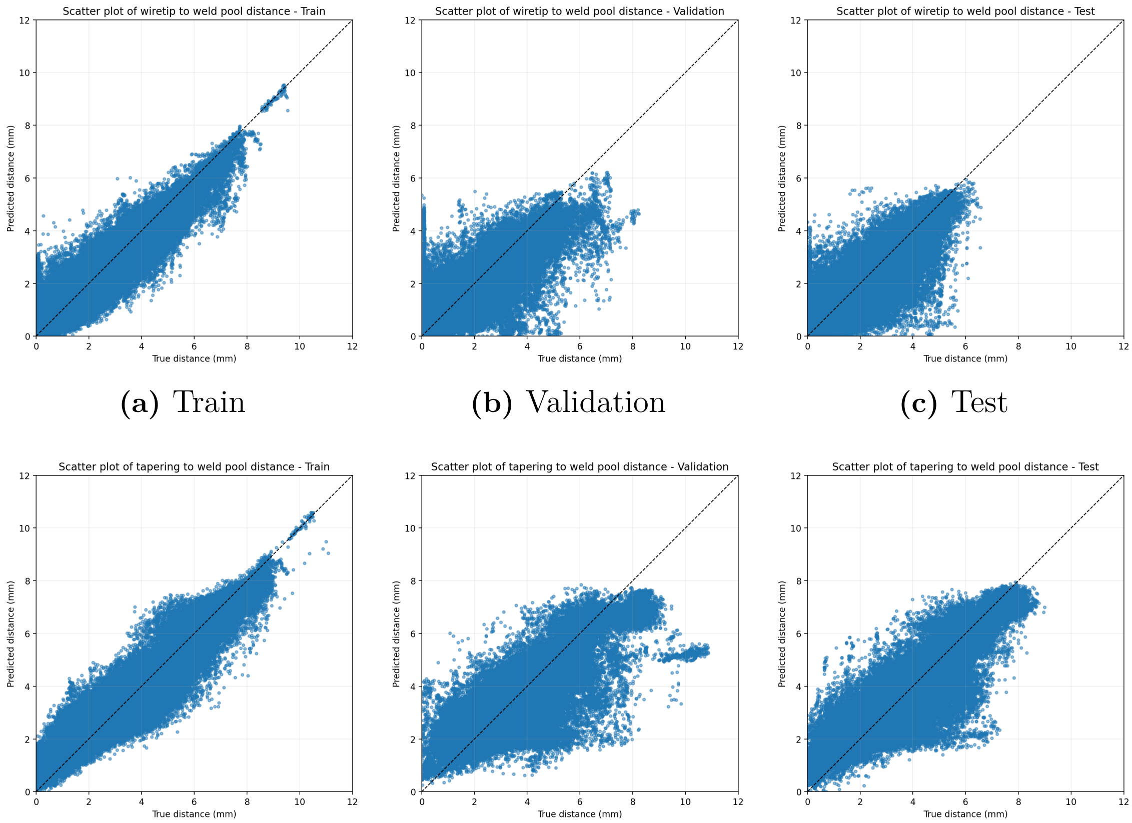


Figure 4.23: Box plot of the XGBOOST model performance on the test data

The box plot across both outputs is shown in Figure 4.23. First plot shows that the residual is a lot higher in the range of 0.2 - 0.1 while the lower ends are comparatively less, meanwhile the residual for the range more than 2 is a lot higher compared to the lower ends. It shows a pattern of having the model perform better in very low values, which are less than 1, rather than higher values of more than 2.



(a) Train **(b)** Validation **(c)** Test

(d) Train **(e)** Validation **(f)** Test

Figure 4.24: Scatter plots of predicted versus measured distances by XGBOOST. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

The scatter plots across train, test, and validation splits for both outputs seem to converge to a straight line with minor deviations in validation and test data for larger distances.

4.4.4 Parallel ensemble

The results from the parallel ensemble experiments showed that the extra trees had the best performance on estimating the wire tip to weld pool and the tapering point to weld pool distance. The final model selected the largest window size and rolling window sizes of 50 and 100, respectively. A sample of the test performance on each target variable is available in Figure 4.25.

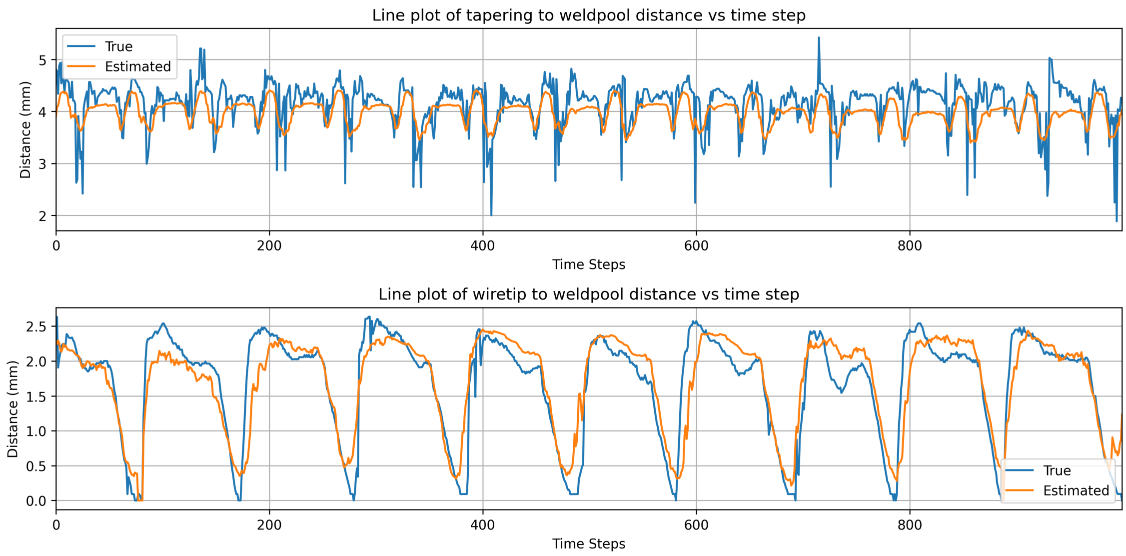


Figure 4.25: Line plot of the Extra Trees model outputs across a test subsample

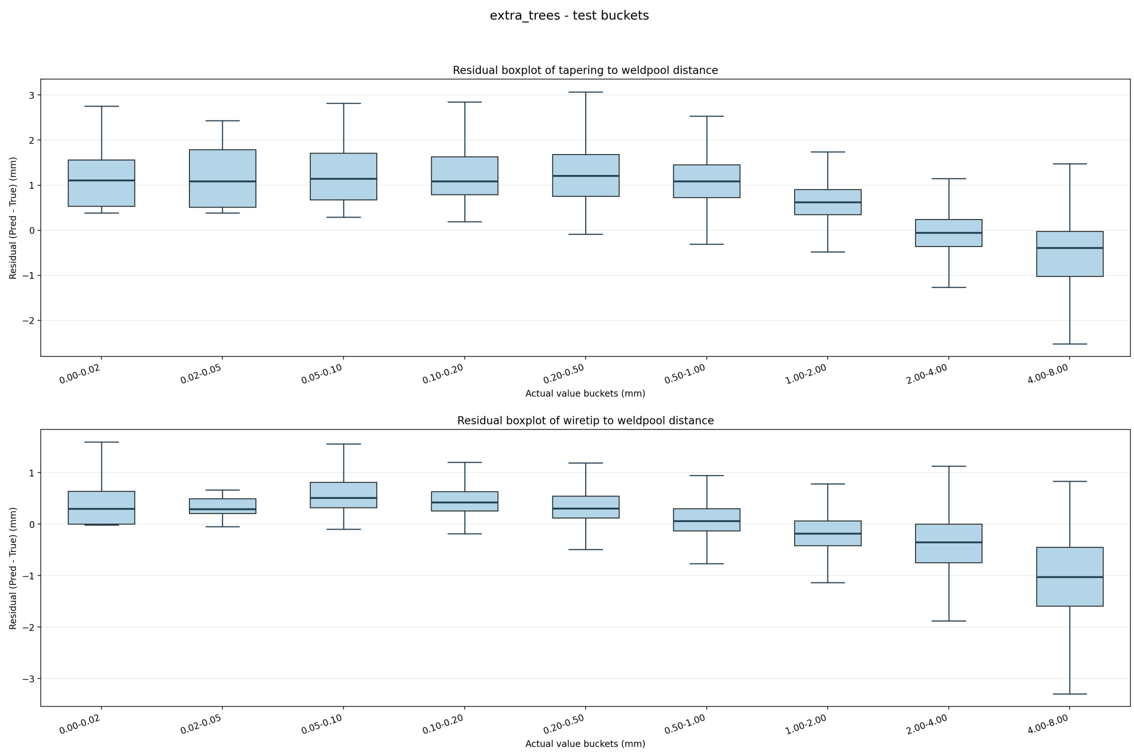
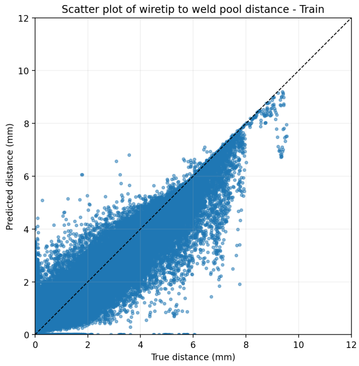
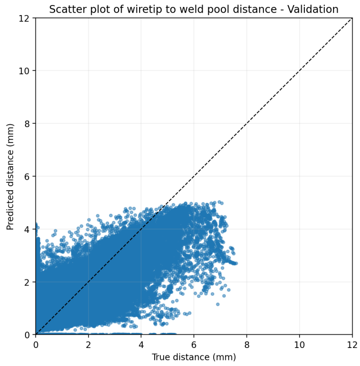


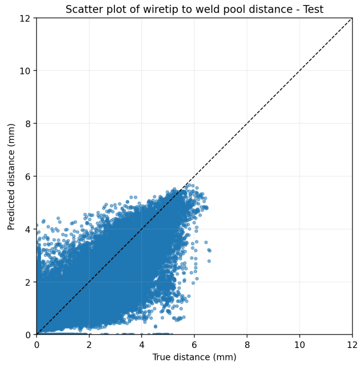
Figure 4.26: Box plot of the Extra Trees model performance on the test data



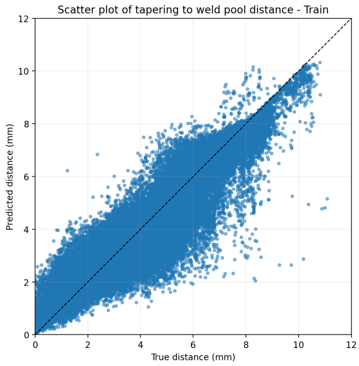
(a) Train



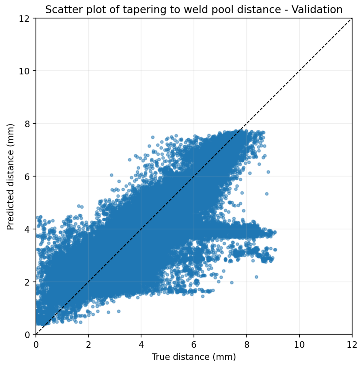
(b) Validation



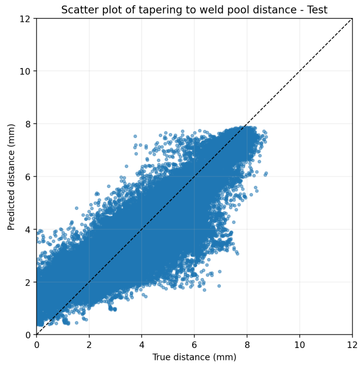
(c) Test



(d) Train



(e) Validation



(f) Test

Figure 4.27: Scatter plots of predicted versus measured distances by extra trees. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

4.4.5 Neural NARX

This section includes scatter plots across train, test, and validation splits used across all the models, along with a line plot across the subsample test data and a box plot across the same subsample test data for output features.

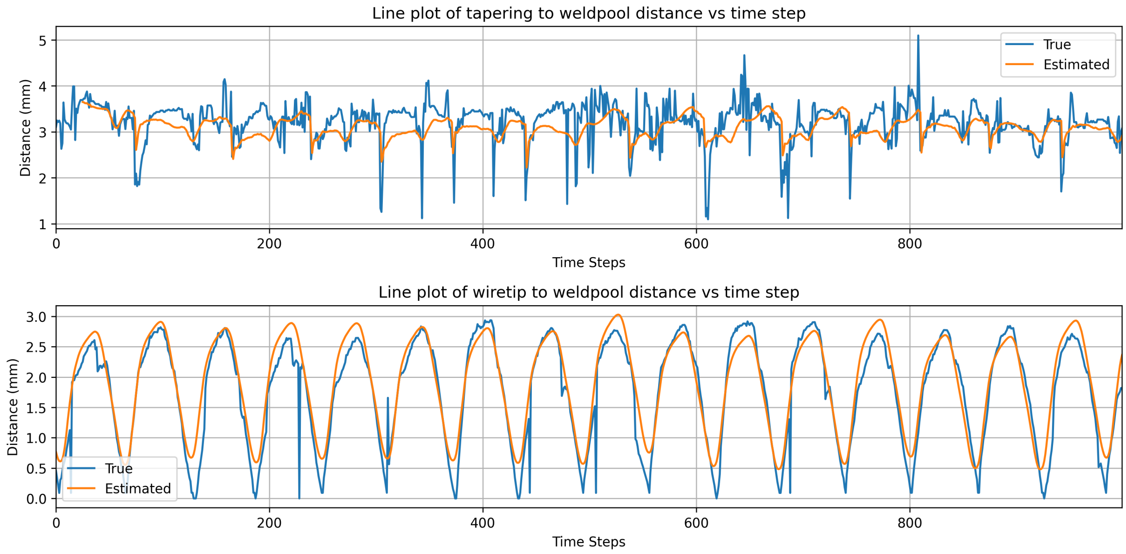


Figure 4.28: Line plot of the NN-NARX model outputs across a test subsample

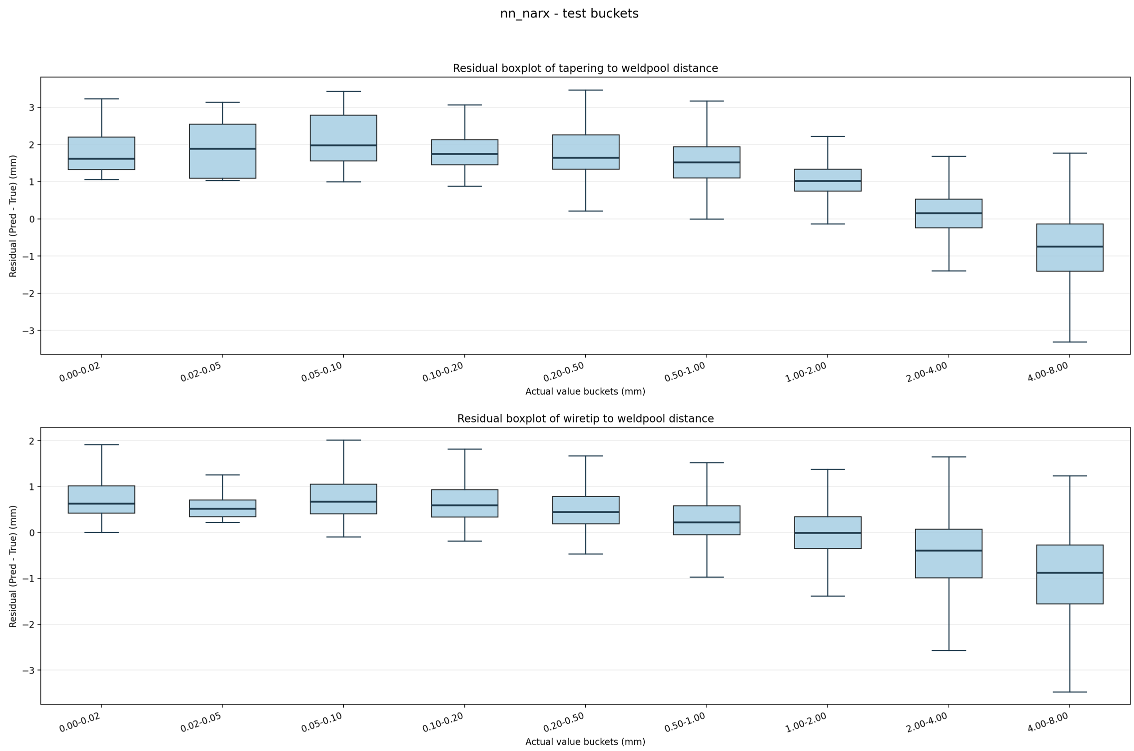
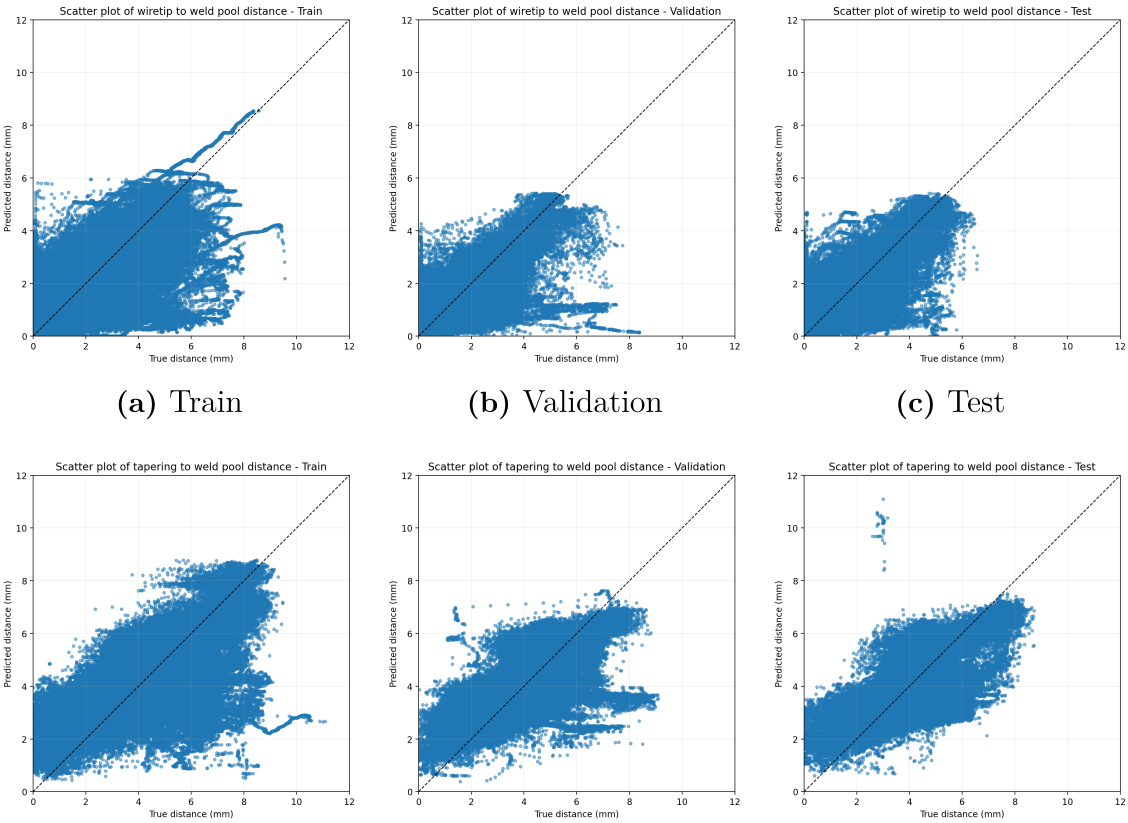


Figure 4.29: Bucket plot of the NN-NARX model performance on the test data



(a) Train **(b)** Validation **(c)** Test

(d) Train **(e)** Validation **(f)** Test

Figure 4.30: Scatter plots of predicted versus measured distances by Neural NARX. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

4.4.6 LSTM

This section includes scatter plots across train, test, and validation splits used across all the models, along with a line plot across the subsample test data and a box plot across the same subsample test data for output features.

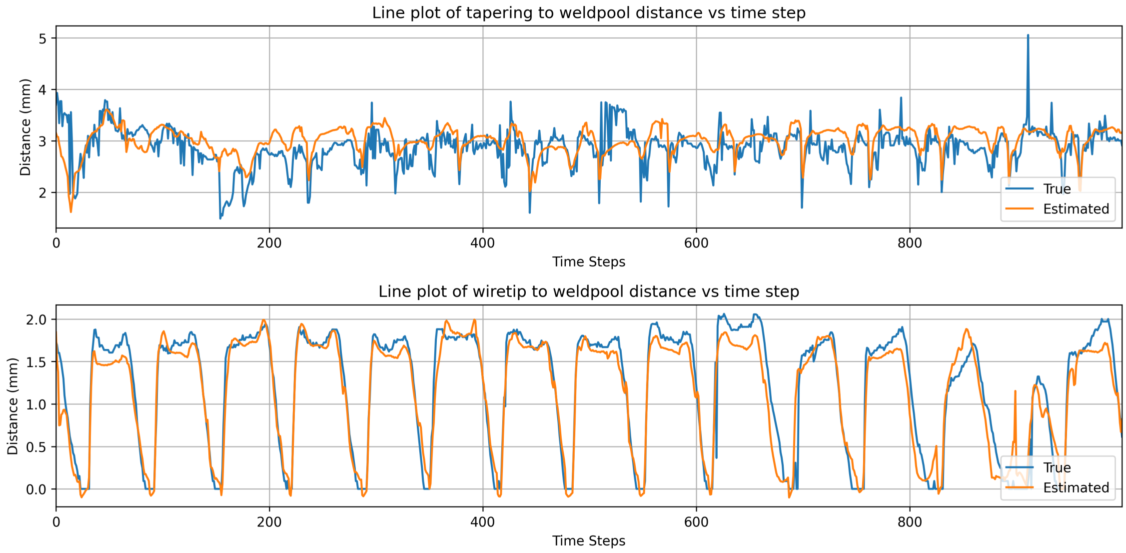


Figure 4.31: Line plot of LSTM model outputs across a test subsample

From the line plot in Figure 4.31, it can be understood that LSTM shows similar performance to XGBOOST in terms of capturing mean value and overall trend, but it fails to respond to rapid changes in the wire tip to weld pool data.

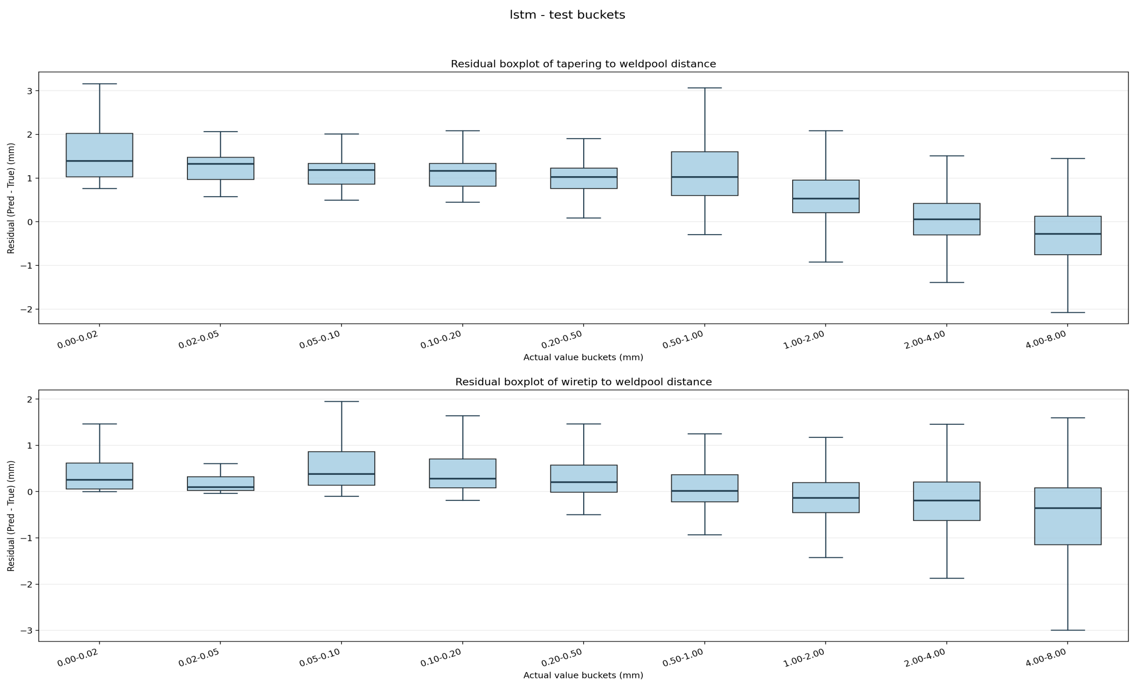
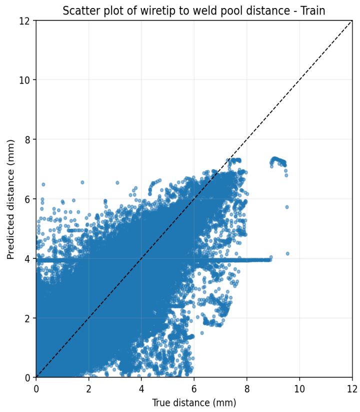
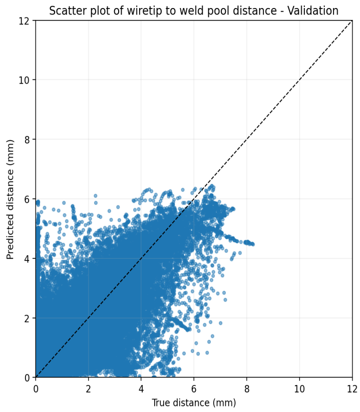


Figure 4.32: Box plot of LSTM model outputs across a test subsample

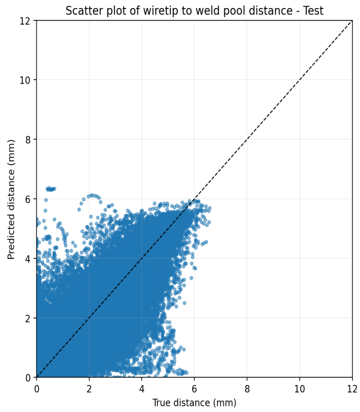
The results of the box plot for both outputs of the LSTM model in Figure 4.32 differ from XGBoost in that the variation of residual is larger across the values and doesn't show a clear trend.



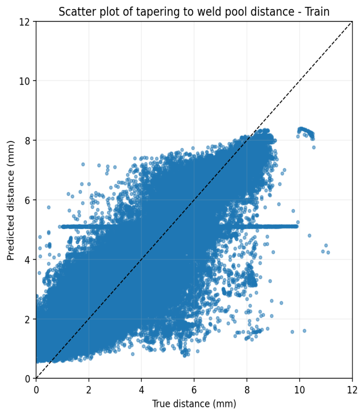
(a) Train



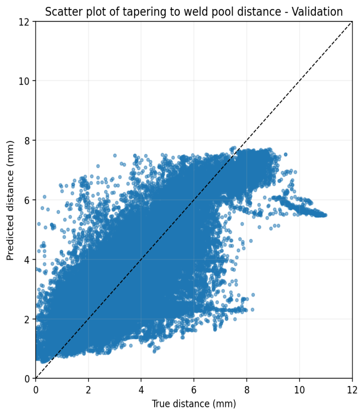
(b) Validation



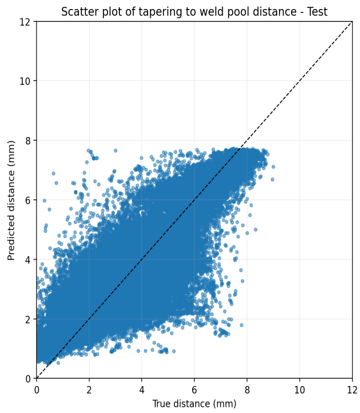
(c) Test



(d) Train



(e) Validation



(f) Test

Figure 4.33: Scatter plots of predicted versus measured distances by LSTM. The first row shows the wire tip-to-weld pool distance model, while the second row shows the tapering point-to-weld pool distance model. Columns correspond to the training, validation, and test datasets.

The scatter grid in Figure 4.33 shows that the model tries to capture the mean value with higher variations for larger values and very small values.

4.4.7 Evaluation

This section includes metrics used for evaluating different model architectures explored on the test data for output feature, which includes RMSE, MAE, RRSE, R2, Pearson coefficient, along with a plot of features used versus average feature importance of all models for both outputs.

Tables 4.5 and 4.6 show that the tree-based models achieved the strongest overall performance. For the wire tip-to-weld pool distance, Extra Trees achieved the best performance across all reported metrics. For the tapering point-to-weld pool distance, XGBoost achieved the lowest RMSE and MAE, while Extra Trees achieved the best RRSE, R2, and Pearson correlation.

Table 4.5: Best performance achieved by each model on test data, estimating wire tip to weld pool distance.

Model	RMSE (mm)	MAE (mm)	RRSE	R2	Pearson
ARX	1.115	0.877	0.878	0.214	0.480
NARX	1.052	0.806	0.807	0.349	0.595
Extra trees	0.681	0.478	0.540	0.708	0.845
XGBoost	0.688	0.491	0.545	0.702	0.838
Neural NARX	0.860	0.633	0.681	0.536	0.735
LSTM	0.731	0.504	0.579	0.664	0.820

Table 4.6: Best performance achieved by each model on test data estimating tapering point to weld pool distance.

Model	RMSE (mm)	MAE (mm)	RRSE	R2	Pearson
ARX	1.306	0.979	0.867	0.191	0.522
NARX	1.154	0.894	0.768	0.410	0.644
Extra trees	0.779	0.566	0.512	0.738	0.864
XGBoost	0.773	0.562	0.537	0.710	0.844
Neural NARX	1.026	0.799	0.679	0.539	0.734
LSTM	0.792	0.572	0.550	0.696	0.837

The feature importance in Figure 4.34 shows that current has the most influence on the outputs on average across the output features and the model architectures, along with power, voltage, and the delta of voltage and current. WFS has the least influence on the outputs.

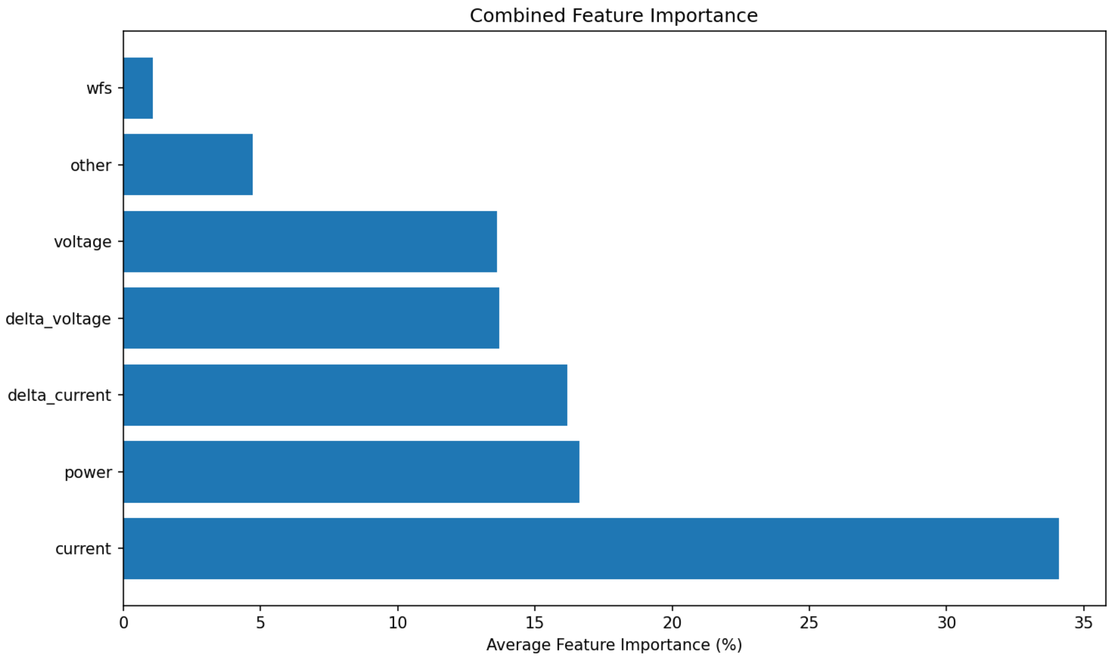


Figure 4.34: Average Feature importance analyzed for all model architectures

The average RMSE of both output and rolling input window sizes from 0 to 100 frames indicates that larger window sizes are better for performance than shorter window sizes. A window size of 75-100 increases performance and yields the best results in terms of RMSE.

5

Discussion

5.1 Data acquisition

The two imaging setups exhibited considerably different characteristics in terms of feature visibility and reliability. Laser-lit recordings contained strong reflections that introduced artifacts and interfered with edge-based segmentation and contour extraction. These artifacts produced unstable detections, making the results less reliable for robust feature extraction. In contrast, the back-lit setup produced high-contrast silhouettes with more clearly defined object boundaries, improving the robustness of edge and contour detection as well as downstream feature extraction. Consequently, the final regression modelling was restricted to back-lit footage, while laser-lit footage was mainly evaluated as part of the image-analysis comparison.

5.2 Computer Vision-Based Detection of Welding Features

The most robust features detected in the computer vision analysis were the wire edges, contact tip and wire tip. Among these, the wire edge detection showed the most consistent performance. However, this should partly be interpreted as a consequence of its central role in the analysis. Since wire edge detection was a prerequisite for downstream feature extraction steps, it was prioritized during development and refined iteratively. The missing `mm_to_px` values identified during anomaly detection were due to frames in which the wire was not visible, rather than incorrect edge detection in usable frames. In addition, the relatively consistent experimental setup made it possible to apply narrow spatial and geometric constraints, likely contributing to the high detection rate.

The contact tip was also detected reliably when visible in the image, mainly due to its well-defined edges and its distance from the arc region. Wire tip detection was generally robust, although the presence of the arc occasionally distorted the lower part of the wire, causing the detected edge to become fragmented or temporarily undetectable. In such cases, the detected wire tip often shifted to the nearest detectable edge close to the true wire tip position. This introduced a local positional error, but the deviation was considered small (a few tenths of mm) relative to the scale of the target distances since the detections remained close to the true position and the distortion occurred only during a limited part of each sequence.

The weld pool was affected by similar detection problems as the wire tip due to its proximity to the arc region. During periods of high arc intensity, the emitted light sometimes distorted the weld pool edges to the extent that reliable detection was not possible, resulting in missing or inaccurate detections. In some cases, the weld pool was detected below its true position, while irregularly shaped droplets occasionally caused false detections above the actual weld pool. To improve robustness, geometric constraints and anomaly detection were implemented to identify physically implausible positions or temporal inconsistencies. In addition, missing or anomalous values were reconstructed using linear interpolation. Since the strongest arc distortions typically happened over short time intervals, the interpolation method helped maintain the temporal consistency of the weld pool position.

The solid-molten transition point extracted from laser-lit footage was less reliable in frames when the glare was inconsistent across the wire, making the gaps too large for the algorithm to bridge. This occasionally caused incorrect localization of the transition point. This was identified quickly, but a fix was de-prioritized since the final analysis focused on back-lit footage, where the texture-based transition point was not visible.

The tapering point detection was generally accurate based on qualitative inspection of random samples. Although the exact tapering point is hard to define manually, the algorithm typically placed the tapering point in the correct region. Initial inaccuracies were mainly caused by fragmented or slightly misaligned wire edge detections, which affected the extrapolated edge lines used to identify the tapering point. These inaccuracies disappeared when the wire edge detection was improved through merging the shorter detected edge segments, increasing wire edge alignment.

The most challenging tasks in the computer vision analysis were developing robust arc detection and plasma characterization. The main difficulty was the large variation in illumination strength and arc intensity across the recorded videos, which made a fixed threshold unsuitable for reliable arc and plasma channel detection. Automatic thresholding and gradient-based edge identification improved the results, but arc-related features still showed the lowest robustness among the evaluated features. Missed detections mainly occurred when the arc was too faint to be separated from the background, or when the wire tip and weld pool were in contact, making the arc region visually ambiguous.

Droplets and spatters are assumed to be approximately circular, so a Simple Blob Detector was used. This assumption fails when droplets merge during short-circuit transfer, producing irregular shapes that are not consistently classified. Reducing *minCircularity* below 0.5 increases sensitivity to such cases but introduces false detections from partial structures, especially wire tip overlaps. In laser-lit images, specular reflections further create false circular blobs, while back-lit imaging reduces this issue due to cleaner silhouettes.

Classification is based on simple area and position rules, which are effective and interpretable but depend on manually tuned thresholds that vary with image scale and setup. Even though the area is limited within the weld pool, it is to be noted that spatters can also be outside of the weld pool area, typically stuck to the plate are classified as bad spatters. Tracking is performed using a Kalman filter with nearest-neighbor matching due to noisy measurements and weak visual consistency across frames. More complex appearance-based or global optimization methods were avoided because droplet appearance is not stable over time. The constant-velocity model works for smooth motion but fails during merges, splits, or sudden acceleration events. Greedy matching is used for efficiency but can lead to incorrect associations when multiple droplets are close together.

To reduce false detections, simple motion and duration constraints are applied, which improves precision but removes short-lived or near-stationary events, especially during short-circuit phases. Velocity analysis shows that spatters have higher acceleration and shorter lifetimes, while droplets are slower and more stable, although tracking gaps occur due to occasional missed or incorrect detections. Overall, the approach remains fast and simple, but performance degrades in dense scenes where droplets interact or overlap visually.

5.3 Data Preparation

The anomaly detection approach uses simple temporal and geometric rules instead of a learned model. The main motivation for this choice is interpretability, since each anomaly can be directly linked to a measurable condition such as missing values, sudden positional jumps, or short-lived events. This makes the method straightforward, easy to debug, and avoids the need for training data or model tuning. An additional advantage is that the logic is deterministic, meaning that the output does not vary between runs and is not affected by dataset bias or overfitting.

This is particularly useful in welding data, where noise, reflections, and detection errors are common and can easily mislead a learned anomaly detection model. However, the rule-based approach also has clear limitations. Since the rules are based on manually chosen thresholds, performance depends on parameter tuning. If the thresholds are too strict, normal variations are flagged as anomalies, while thresholds that are too loose may fail to flag incorrect detections. The thresholds were tuned for the provided dataset and manually verified, but may need to be adjusted when applying the method to other imaging configurations or process conditions.

Analysis of the detected anomalies showed that missing weld pool values were the most common cause of anomaly detection. The missing conversion rate between mm and pixels is also a common reason. These two cases are closely related, since the conversion rate can only be calculated when the wire edges are successfully detected. The entire analysis depends on successful wire edge detection, if it fails, the analysis of that particular frame is aborted. The most common reason they are not found is that the wire isn't visible within the frame at all, while a less frequent reason is that the arc brightness distorted the image enough to prevent wire edge detection.

Another limitation is the lack of longer temporal patterns, where each frame is evaluated mostly independently or using short windows. Therefore, longer temporal patterns such as gradual drift and slow degradation is not captured as efficiently. The method also treats all features equally, even though some signals are more physically meaningful than others. In addition to this, the system cannot adapt to new data or changing process conditions, unlike learning-based approaches. Any new failure mode must be manually encoded as another rule, which makes the approach less scalable.

Overall, the method is robust and explainable, but rigid and limited in its ability to capture complex or evolving system behavior. Further work could investigate autoencoders or other machine-learning-based anomaly detection methods, as well as additional rule-based criteria derived from welding process physics.

Linear interpolation was applied to minimize data loss after anomaly removal. The limit of 10 consecutive frames was selected as a compromise between preserving as much data as possible and avoiding the introduction of synthetic erroneous data. During a 10-frame interval, approximately 1 ms passes, corresponding to roughly one fifth of a pulse cycle. Over such short durations, the wire tip and weld pool positions typically move continuously, which motivates the use of linear interpolation for short missing intervals.

During the initial statistical model training, it was observed that the upsampled data resulted in substantially longer processing times, since it contained 20 times more data points than the down sampled data. However, the voltage and current measurements used as model inputs were IIR-filtered with 15kHz cut-off, which removed much of the high-frequency content present in the analog signals. As a result, the additional samples introduced through upsampling mainly increased the temporal resolution of already smooth signals rather than providing substantially more additional process information. Therefore, the upsampled dataset did not appear to improve the predictive capability of the evaluated models sufficiently to justify the increased computational cost. It should be noted, however, that signal processing on the order of microseconds may still be required for detecting fast transient events, such as short-circuit breakage, in a real-time control system. Subsequent model training was therefore performed using the down sampled dataset, which provided a more practical tradeoff between computational cost and predictive performance.

5.4 Estimation of Welding Features

The statistical and machine learning models showed clear differences in how they handled the temporal and nonlinear characteristics of the welding process. An observation made when fitting autoregressive models was that the one-step-ahead performance was substantially better than the performance in freerun closed-loop simulations. In autoregressive models, previous outputs are used as inputs for subsequent predictions. During initial training, the implemented procedure used the true previous output values as input variables. Since consecutive target values were strongly correlated, with $d_{t+1} \approx d_t$, the models could achieve low prediction errors simply by propagating recent output values forward. However, when the models were evaluated in closed-loop simulation using their own previous predictions as inputs, small local prediction errors accumulated over time and propagated into future inputs. This caused the estimates to gradually drift, resulting in poor long-term prediction performance. This showed that one-step-ahead accuracy alone was not a reliable indicator of model suitability for real-time state estimation. One possible improvement would be to include detected short-circuit events as an additional event feature or reset condition, since these events are clearly visible as voltage drops.

To address this issue, alternative training strategies were investigated, including using only past input signals as model inputs, mixing true outputs with model predictions during training, and excluding previous true outputs entirely. These experiments highlighted an important limitation of purely autoregressive models for this application, their sensitivity to error propagation in recursive operation. Consequently, model selection for autoregressive models was based on free-run performance on unseen test data rather than only one-step-ahead prediction accuracy. In the freerun setting, the model cannot rely on true previous target values during prediction. This better reflects the conditions of a real-time implementation, where future ground-truth image-derived features are not available.

Another important challenge during model development was overfitting to the training data. The more advanced models consistently achieved better performance on the training data than on the validation and test data. A likely reason was the relatively small number of welding sequences combined with the large variation of process settings between sequences. Variations in welding machines, waveform characteristics, WFS, and process parameters created significantly different operating conditions and dynamics across the dataset. When entire sequences were assigned exclusively to training, validation, and test partitions, the resulting data distributions differed considerably.

To reduce this distribution mismatch, the longer sequences were split into shorter subsequences before partitioning. This allowed all parameter settings to be represented across the training, validation, and test datasets. However, this approach also reduced the statistical independence between the partitions, since subsequences from the same welding sequence could still share similar temporal patterns. The evaluation should therefore be interpreted as measuring the generalization across operating conditions within the available dataset, rather than complete generalization to entirely unseen welding processes.

5.4.1 Evaluation of model performance

The evaluated autoregressive models showed different behavior depending on how strongly they relied on previous outputs. The linear ARX models were generally unable to capture the nonlinear dynamics present in the welding process. The nonlinear ARX models improved the representation of nonlinear relationships by expanding the regressor space through nonlinear combinations of lagged variables. However, this also increased the dimensionality of the input space, increasing computational cost and overfitting.

The tree-based ensemble methods generally handled the nonlinear relationships more effectively than the autoregressive models. XGBoost captured complex local dependencies between electrical signals and geometric features. Still, the estimates showed a clear regression-to-the-mean effect, with small target values overestimated and large target values underestimated. This was probably caused by the imbalance in the target distributions, together with the limited number of samples representing extreme operating conditions. Since tree-based models cannot extrapolate outside the value range observed during training, prediction errors increased near the boundaries of the target distributions. Since each time step is treated independently, the model lacks representation of temporal continuity. As a result, the estimations react strongly to instantaneous changes in the electrical signals, producing jitter. Although this enabled the model to capture sharp local changes, it struggles to maintain smooth and consistent behavior over time.

The Extra Trees model showed more stable generalization behavior than the other evaluated ensemble methods. A likely explanation is the increased randomness introduced during tree construction, where both split variables and split thresholds are randomized. This reduces the tendency to fit sequence-specific correlations or noise patterns in individual lag variables. Compared with XGBoost, which sequentially focuses on correcting previous residuals, Extra Trees produced smoother and more stable predictions across different operating conditions. This suggests that, for the available dataset, robustness to noise and dataset variability was more important than optimizing local prediction accuracy.

The LSTM was evaluated to investigate whether incorporating temporal dependencies could improve prediction performance. Compared with tree-based methods, the LSTM retains information from previous states and is therefore more suitable for modeling the cyclic and time-dependent characteristics of the welding process. The results showed that the LSTM better preserved the overall waveform structure and phase behavior of the target signals, particularly for the wire tip to weld pool distance. This indicates that temporal dependencies in the electrical signals contain information related to the underlying geometric welding features. The recurrent structure allowed the model to learn broader process dynamics rather than relying only on instantaneous relationships between lagged variables and outputs.

However, the recurrent state representation also introduced a smoothing effect, causing rapid local variations to be underestimated. While the estimations appeared more temporally consistent than those from XGBoost, the model struggled to reproduce sharp drops and high-frequency fluctuations. Similar to the other evaluated models, the LSTM also showed reduced accuracy near the boundaries of the target features, indicating that temporal modeling alone was insufficient to fully capture rare events.

Although the tree-based models do not explicitly model temporal dependencies, the inclusion of lagged measurements and engineered temporal features introduced a finite temporal context into the input representation. Features such as rolling means, rolling standard deviations, and slopes within the window size allowed the models to capture short-term process dynamics and local temporal trends without maintaining an internal recurrent state. This likely contributed to the comparatively strong performance of the ensemble methods despite their lack of explicit sequence modeling. However, unlike the LSTM, the temporal dependencies are limited to constant window size and manually engineered summaries, which may restrict the ability to capture longer-range or evolving process dynamics.

The differences in estimation performance between target variables also suggest that some welding features are more observable from electrical signals than others. The wire tip to weld pool distance was predicted more accurately than the tapering point to weld pool distance across most evaluated models. A likely explanation is that the wire tip distance exhibited a clearer cyclic structure and stronger coupling to the pulsed welding dynamics visible in the electrical measurements. In contrast, the tapering point distance varied within a narrower range, exhibited noisier behavior, and was more sensitive to uncertainty in the image-based feature extraction. This made the underlying input-output relationship more difficult for the models to learn consistently.

A recurring pattern across nearly all evaluated models was substantially better performance on the training data than on the validation and test data, indicating limited generalization to unseen welding conditions. A likely explanation is the relatively small number of welding sequences combined with the large variation in process settings between sequences. Differences in welding machine, waveform characteristics, wire feed speed, CTWD, and process settings created substantially different operating regimes across the dataset. This increases the complexity of the learning problem, since similar electrical signal patterns may correspond to different physical process states depending on the operating conditions. As a result, the models may partially learn sequence-specific relationships rather than general process dynamics.

The performance of the evaluated model architectures is not yet sufficient for adaptive control implementation. However, increasing dataset size and diversity, reducing target imbalance, and exploring hybrid model architectures that combine the strengths of the evaluated approaches could improve the feasibility of the proposed soft-sensing methodology.

6
Conclusion

In this thesis, we developed and evaluated an image-based soft-sensing framework for estimating geometric features in a GMAW-P welding process using electrical measurements and static process context. The results show that the imaging setup strongly influences the reliability of classical vision methods. Laser illumination improves visibility of wire and arc-related structures but introduces specular reflections that generate false features. Back-lit imaging provides cleaner silhouettes with more stable boundaries, providing more reliable extraction of wire tip, contact tip, and weld pool positions.

Classical edge and contour-based methods perform well for clearly defined geometric structures but are unreliable for diffuse or illumination-dependent features. Arc and plasma regions are particularly unstable due to intensity fluctuations and weak boundary definition. While adaptive thresholding and gradient-based techniques improve consistency, robustness across full sequences remains limited.

The combined dataset, formed by synchronizing electrical signals (current, voltage), process parameters (WFS), and vision-based geometric features, enables time-series modeling of welding behavior. A range of models was evaluated, including autoregressive models, tree-based ensembles, NN-NARX, and LSTM networks, to capture both temporal structure and nonlinear relationships.

Across models, predictions consistently capture the overall cyclic trends and general directional behavior of the target variables, reflected in correlation with the image-derived reference measurements. Performance differences are primarily driven by how each model handles noise, temporal structure, and distribution imbalance, with more flexible models better capturing nonlinear dynamics but also showing increased sensitivity to sparse regions and outliers.

For further development toward closed-loop control, increasing dataset size and diversity is critical to reduce overfitting to narrow operating regimes. Incorporating additional process descriptors such as CTWD and other contextual variables would further constrain the learning problem and improve generalization in industrial conditions. To address merged droplets, manually annotated data could be used to train and evaluate more advanced deep-learning-based detection or segmentation methods. Also, as different model architectures have their own advantages, combining model outputs through ensemble or hybrid approaches could help improve the performance. In addition, evaluating model performance across different frequency ranges could provide further insight into how well the models capture both slow process trends and fast transient behavior.

Overall, the thesis supports the feasibility of using electrical process signals for soft sensing of selected geometric welding features in GMAW-P, but further work is required before the approach can be considered sufficiently accurate and robust for adaptive control.

Bibliography

[1] Shravan C, N. Radhika, N. H. Deepak Kumar, and B. Sivasailam. A review on welding techniques: properties, characterisations and engineering applications. *Advances in Materials and Processing Technologies*, 10(2):1126–1181, 2024.

[2] Anthony B. Murphy and John J. Lowke. *Heat Transfer in Arc Welding*, pages 2657–2727. Springer International Publishing, Cham, 2018.

[3] P. Kah and J. Martikainen. Influence of shielding gases in the welding of metals. *International Journal of Advanced Manufacturing Technology*, 64(9-12):1411 – 1421, 2013.

[4] Pingsha Dong. Quantitative weld quality acceptance criteria: An enabler for structural lightweighting and additive manufacturing. *Welding Journal*, 99(2):39S – 51S, 2020.

[5] J. Grill, "What is GMAW (MIG Welding) & How Does it Work?," *WeldGuru*. Accessed: Jun. 11, 2026. [Online]. Available: https://weldguru.com/wp-content/uploads/mig-welding-process.jpg.

[6] E.F. da Silva, J.R. Macedo, A. Scotti, and J.C. de Oliveira. Power quality analysis of gas metal arc welding process operating under different drop transfer modes. *Renewable Energy and Power Quality Journal*, 1(9):362 – 368, 2011.

[7] Zhendan Zheng, Shaojie Wu, Limin Fan, Hao Wu, and Fangjie Cheng. The formation mechanism and suppression strategies of spatter in pulsed gas metal arc welding for titanium alloy. *Welding Journal*, 103(8):225s – 235s, 2024.

[8] J. Hu and H.L. Tsai. Effects of current on droplet generation and arc plasma in gas metal arc welding. *Journal of Applied Physics*, 100(5), 2006.

[9] P.K. Palani and N. Murugan. Selection of parameters of pulsed current gas metal arc welding. *Journal of Materials Processing Technology*, 172(1):1–10, 2006.

[10] Marjan Suban and Janez Tušek. Methods for the determination of arc stability. *Journal of Materials Processing Technology*, 143-144:430–437, 2003.

[11] Yuze Liu, Honglei Zhao, Yiwen Li, Siyu Zhang, Junyan Miao, Senyao Xing, Hao Dong, and Yunlong Chang. Arc characteristics and welding joints in pulsed tig welding with different current waveforms. *International Journal of Precision Engineering and Manufacturing*, 26:1643–1653, 2025.

[12] Insa Henze and Thorsten Mattulat. Metal vapor-induced arc instability in stationary tig-welding. *The International Journal of Advanced Manufacturing Technology*, 133:3891–3900, 2024.

[13] M. Rohe, M. Knester, J. Hildebrand, and J.P. Bergmann. Development of an indirect measurement method for the contact tube to workpiece distance (ctdw)

in the direct energy deposition – arc (ded-arc) process for different arc types. *Journal of Advanced Joining Processes*, 9:100228, 2024.

[14] Wei Guo, Xincheng Zhao, Yonglin Zhao, Yuxiang Liu, Han Li, and Bingyuan Han. Research on optimal heat input parameter for tig welding of thin plate 5083 aluminum alloy. *Scientific Reports*, 15(1):15593, 2025.

[15] A.S. Shahiand and S. Pandey. Welding current prediction in gmaw and ugmaw processes using response surface methodology. *Science and Technology of Welding and Joining*, 11(3):341 – 346, 2006.

[16] F.H. Ley, S.W. Campbell, A.M. Galloway, and N.A. McPherson. Effect of shielding gas parameters on weld metal thermal properties in gas metal arc welding. *International Journal of Advanced Manufacturing Technology*, 80(5-8):1213 – 1221, 2015.

[17] Z. Yang, K. Wang, Y. He, Y. Rao, X. Yang, P. Zhao, C. Fang, and Y. Chen. Effects of shielding gas composition on process stability and arc behavior of laser-cable-type welding wire arc hybrid welding. *Metals*, 16:20, 2026.

[18] Vikas Kumar, Manoj K. Parida, and Shaju K. Albert. Analysis of gas metal arc welding process using improved complete ensemble empirical mode decomposition with adaptive noise. *Transactions of the Indian Institute of Metals*, 77(11):3279 – 3291, 2024.

[19] Zhijiang Wang, Yuming Zhang, and Lin Wu. Adaptive interval model control of weld pool surface in pulsed gas metal arc welding. *Automatica*, 48(1):233 – 238, 2012.

[20] Zhixiang Chen, Jun Zhang, Yonglun Song, and Yongzhong Ding. Synchronized record and analysis of high speed images and process parameters for welding process. *Hanjie Xuebao/Transactions of the China Welding Institution*, 31(3):69–72+76, 2010.

[21] Peizhuo Zhai, Songbai Xue, Tao Chen, Jianhao Wang, and Yu Tao. An image-processing method for extracting kinematic characteristics of droplets during pulsed gmaw. *Applied Sciences (Switzerland)*, 9(24), 2019.

[22] Lijun Xu, Mengyu Zhang, Fangzhou Zhao, Pengbo Zhang, Wenzhi Chen, and Zezhou Wang. Temperature distribution of arc plasma based on image processing technology. In *Proceedings of SPIE - The International Society for Optical Engineering, Vol. 11427*, 2020.

[23] Ziquan Jiao, Haijian Qin, Xingyu Gao, Zhiqiang Feng, Yuanzhao Xu, Shanben Chen, Nanhui Shi, and Wenjing Liu. Image processing and feature extraction for hull structure gmaw based on weld pool visual sensing. *Journal of Sensors*, 2023(1), Jan 2023.

[24] Bo Guo, Yonghua Shi, and Yaoyong Yi. Metal transfer modes identification based on arc shape. *Hanjie Xuebao/Transactions of the China Welding Institution*, 38(11):27 – 31, 2017.

[25] Wenhang Li, Yunhong Ji, Jing Wu, and Jiayou Wang. A modified welding image feature extraction algorithm for rotating arc narrow gap mag welding. *Industrial Robot*, 42(3):222 – 227, 2015.

[26] Jisoo Hong, Youngjin Jung, Jihwan Bae, Seungho Song, and Sung-Woo Kang. Robust detection of extremely thin lines using 0.2mm piano wire, 2025.

[27] Jun Lu, Aodong Yang, Xiaoyu Chen, Xingwang Xu, Ri Lv, and Zhuang Zhao. A seam tracking method based on an image segmentation deep convolutional neural network. *Metals*, 12(8):1365, 2022.

[28] J. Zhang et al. A robust weld seam recognition method under heavy noise based on structured-light vision. *Robotics and Computer-Integrated Manufacturing*, 55:87–100, 2019.

[29] Mingjie Zhu, Zilong Shan, Wei Ning, and Xuanye Wu. Image segmentation and quantification of droplet dpcr based on thermal bubble printing technology. *Sensors*, 22:7222, 09 2022.

[30] Deyong You, Xiangdong Gao, and Seiji Katayama. Monitoring of high-power laser welding using high-speed photographing and image processing. *Mechanical Systems and Signal Processing*, 49(1):39–52, 2014.

[31] Mingxuan Yang, Hao Zheng, and Ling Li. Arc shape characteristics with ultra-high-frequency pulsed arc welding. *Applied Sciences*, 7(1), 2017.

[32] Prithvi Raj Singh, Raju Gottumukkala, and Anthony Maida. An analysis of kalman filter based object tracking methods for fast-moving tiny objects, 2025.

[33] X. Li et al. Multi-object droplet tracking for high-frame-rate video analysis in welding processes. *Expert Systems with Applications*, 213:119047, 2023.

[34] Mihir Durve et al. Droptrack: Automatic droplet tracking using deep learning for microfluidic applications. *arXiv preprint arXiv:2205.02568*, 2022.

[35] Yifu Zhang, Chunyu Wang, Xinggang Wang, et al. Fairmot: On the fairness of detection and re-identification in multiple object tracking. In *International Journal of Computer Vision*, 2021.

[36] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision (ECCV)*, 2020.

[37] E. B. F. Dos Santos, R. Pistor, and A. P. Gerlich. On the visualization of gas metal arc welding plasma and the relationship between arc geometry and arc voltage. *Applied Sciences*, 7(5):503, 2017.

[38] J. Huang et al. Study on the effect of pulse waveform parameters on droplet transfer and weld formation in pulsed gmaw. *Metals*, 13(2):199, 2023.

[39] Alexei Stepanov. On correlation coefficients, 2024.

[40] X. Liang et al. Causal inference of multivariate time series in complex industrial systems. *Advanced Engineering Informatics*, 2024.

[41] A. Wismüller et al. Large-scale nonlinear granger causality for inferring directed dependence from short multivariate time-series data. *Scientific Reports*, 11:87316, 2021.

[42] Francisco Galan-Sales, Pablo Reina-Jiménez, Manuel Carranza-García, and José María Luna-Romera. *An Approach to Enhance Time Series Forecasting by Fast Fourier Transform*, pages 259–268. 08 2023.

[43] Girish Keshav Palshikar. Simple algorithms for peak detection in time-series. 2009.

[44] Yubo Ye. A comprehensive review of principal component analysis. *Academic Journal of Science and Technology*, 17:224–227, 11 2025.

[45] Y. Huang et al. A quality diagnosis method of gmaw based on improved empirical mode decomposition and extreme learning machine. *Journal of Manufacturing Processes*, 54:1–12, 2020.

[46] Y. Huang et al. Feature extraction for gas metal arc welding based on emd and time–frequency entropy. *The International Journal of Advanced Manufacturing Technology*, 92:2789–2797, 2017.

[47] P. Yu et al. The on-line identification and location of welding short-circuit transition process based on ceemd. *Metals*, 14(4):396, 2024.

[48] Y. Li et al. Physics-informed feature learning for welding process monitoring using intermediate variables. *Materials*, 16(3):987, 2023.

[49] H. Chen et al. Multi-task deep learning for welding defect detection and process parameter estimation. *International Journal of Machine Tools and Manufacture*, 175:103–114, 2022.

[50] Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. Multistep-ahead time series prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3918 LNAI:765 – 774, 2006.

[51] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[52] Timothy Hall and Khaled Rasheed. A survey of machine learning methods for time series prediction. *Applied Sciences*, 15(11), 2025.

[53] Gerasimos G. Rigatos. A derivative-free kalman filtering approach to state estimation-based control of nonlinear systems. *IEEE Transactions on Industrial Electronics*, 59(10):3987 – 3997, 2012.

[54] Chenyi Lyu, Xingchi Liu, and Lyudmila Mihaylova. Review of recent advances in gaussian process regression methods. In George Panoutsos, Mahdi Mahfouf, and Lyudmila S. Mihaylova, editors, *Advances in Computational Intelligence Systems*, pages 226–237, Cham, 2024. Springer Nature Switzerland.

[55] C.M. Cheng, Z.K. Peng, W.M. Zhang, and G. Meng. Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing*, 87:340–364, 2017.

[56] Reese Eischens, Tao Li, Gregory W. Vogl, Yi Cai, and Yongzhi Qu. State space neural network with nonlinear physics for mechanical system modeling. *Reliability Engineering System Safety*, 259:110946, 2025.

[57] Vaia I. Kontopoulou, Athanasios D. Panagopoulos, Ioannis Kakkos, and George K. Matsopoulos. A review of arima vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*, 15(8), 2023.

[58] Ye Ren, Le Zhang, and P.N. Suganthan. Ensemble classification and regression-recent developments, applications and future directions [review article]. *IEEE Computational Intelligence Magazine*, 11(1):41 – 53, 2016.

[59] Jiangbin Zheng, Farkhanda Afzal, Hui Lin, Saleem Riaz, and Atif Mehmood. Supervised ensemble learning methods towards automatically filtering urdu fake news within social media. *PeerJ Computer Science*, 7, 03 2021.

[60] Richard Johansson. An intuitive explanation of gradient boosting, 2019.

[61] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[62] Soukaina Hakkal and Ayoub Ait Lahcen. Xgboost to enhance learner performance prediction. *Computers and Education: Artificial Intelligence*, 7:100254, 2024.

[63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[64] Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, Amgad Muneer, Ebrahim Hamid Sumiea, Alawi Alqushaibi, and Mohammed Gamal Ragab. Rnn-lstm: From applications to modeling techniques and beyond—systematic review. *Journal of King Saud University - Computer and Information Sciences*, 36(5):102068, 2024.

[65] Yi Qiao, Kaiwen Xu, and Kai Zhou. Research on time series based on improved lstm. In *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 951–958, 2023.

[66] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE TPAMI*, 2018.

[67] Timothy Hodson. Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not. *Geoscientific Model Development*, 15:5481–5487, 07 2022.

[68] Hoda Khoshvaght, Ratish Ramyad Permala, Amir Razmjou, and Mehdi Khiadani. A critical review on selecting performance evaluation metrics for supervised machine learning models in wastewater quality prediction. *Journal of Environmental Chemical Engineering*, 13(6):119675, 2025.

[69] Zhengze Zhou and Giles Hooker. Unbiased measurement of feature importance in tree-based methods, 2020.

A
Appendix 1

```
CONFIG = {
# General parameters
"wire_width_mm": 1.0,

"wire_tool_transition": {
    "offset_from_center": 10,
    "start_offset": 50,
    "extend_upward": 1000,
},

"wire_searchspace": {
    "top_shift_px": 5,
    "length_scale": 1.2,
    "width_scale": 3,
},

"solid_molten_transition": {
    "width_threshold_px": 10.0,
    "min_run": 10,
},

"weldpool": {
    "min_edge_width_px": 1,
    "scan_half_width_px": 10,
    "remove_circular_droplets": True,
    "droplet_min_area_px": 1,
    "droplet_circularity_threshold": 0.4,
    "show_verification": False,
},

"wire_contour": {
    "min_length": 100,
    "min_width": 2,
    "max_width": 1000,
    "max_angle": 30.0,
    "min_h_to_w": 1,
    "split_threshold": 400,
```

```
    },

    "anomaly_detection": {
        "min_anomaly_score": 1.0,
        "jump_mean_pct_threshold": 0.20,
        "window_size": 30,
        "short_life_max_len": 2,
        "jump_weight": 1.0,
        "missing_weight": 1.0,
        "transient_weight": 2.0,
    },




    # Laserlit parameters
    "laserlit": {
        "laserlit_edge": {
            "canny1": 25,
            "canny2": 90,
            "morph": True,
        },

        "laserlit_basemetal": {
            "min_edge_pixels": 2,
            "min_edge_width": 10,
        },

        "laserlit_wire_edges": {
            "hough_threshold": 10,
            "min_line_length": 30,
            "max_line_gap": 15,
            "rho": 1,
            "theta": np.pi/360,
            "min_length": 30,
            "angle_tolerance_deg": 3.0,
            "vertical_tolerance_deg": 10.0,
            "min_x_distance": 25,
            "max_x_distance": 50,
        },

        "laserlit_contours": {
            "canny1": 25,
            "canny2": 75,
            "morph": True,
        },
```

```
    "laserlit_tapering_point": {
        "search_window_radius": 8,
        "vicinity_radius": 2,
        "max_consecutive_misses": 2,
    },

    "arc_detection": {
        "plasma_channel_edge": {
            "box_width": 150,
            "search_band_height": 5,
            "track_window": 5,
            "min_gradient_score": 3.0,
            "gradient_std_factor": 0.05,
        },
        "plasma_channel_attachment": {
            "box_width": 80,
            "box_height": 80,
            "blur_kernel": 5,
            "std_threshold_factor": 0.75,
            "percentile_threshold": 85,
        },
    },

},

# Backlit parameters
"backlit": {
    "backlit_edge": {
        "canny1": 75,
        "canny2": 125,
        "morph": False,
    },
    "backlit_basemetal": {
        "min_edge_pixels": 2,
        "min_edge_width": 10,
    },

    "backlit_wire_edges": {
        "hough_threshold": 15,
        "min_line_length": 30,
        "max_line_gap": 5,
        "rho": 1,
        "theta": 1,
        "min_length": 30,
        "angle_tolerance_deg": 2.0,
```

```
            "vertical_tolerance_deg": 15.0,
            "min_x_distance": 30,
            "max_x_distance": 100,
        },

        "backlit_contours": {
            "canny1": 75,
            "canny2": 125,
            "morph": False,
        },

        "backlit_tapering_point": {
            "search_window_radius": 8,
            "vicinity_radius": 1,
            "max_consecutive_misses": 2,
        },

        "arc_detection": {
            "plasma_channel_edge": {
                "box_width": 150,
                "search_band_height":8,
                "track_window": 5,
                "min_gradient_score": 1.3,
                "gradient_std_factor": 0.10,
            },
            "plasma_channel_attachment": {
                "box_width": 80,
                "box_height": 80,
                "blur_kernel": 5,
                "std_threshold_factor": 0.85,
                "percentile_threshold": 95,
            },
        },
    },
}
```

B

Appendix 2

```
CONFIG = {
    "preprocessing": {
        "input_path": str(PROJECT_ROOT / "output" / "measurements"),
        "detect_path": str(PROJECT_ROOT / "output" / "detect" / "csv"),
        "anomaly_path": str(PROJECT_ROOT / "output" / "anomaly" / "csv"),
        "h5_input_path": str(PROJECT_ROOT / "data" / "backlit" / "to_be_processed")
        "h5_output_path": str(PROJECT_ROOT / "output" / "measurements"),
        "output_path_org": str(PREDICTIVE_ROOT / "data" / "merged_org"),
        "output_path_clipped": str(PREDICTIVE_ROOT / "data" / "merged_clipped"),
        "output_path_clipped_norm": str(PREDICTIVE_ROOT / "data" /
        "merged_clipped_norm"),
        "interpolation": {
            "interpolation_columns": [
                "plasma_attachment_height",
                "plasma_channel_avg_width",
                "tapering_y",
                "weldpool_y",
                "wiretip_y",
                "tooltip_x1",
                "tooltip_x2",
                "tooltip_y1",
                "tooltip_y2",
                "mm_per_px",
            ],
        },
        "max_anomaly_length": 10,
        "upsample": False,
        "split": {
            "target_col": "wiretip_to_weldpool_dist",
            "history_cols": ["voltage", "current", "delta_voltage",
            "delta_current"],
            "context_cols": ["wfs"],
            "offline_metadata_cols": ["wfs", "ctdw"],
            "frame_order_col": "meas_time",
            "subsequence_length": 4_000,
            "subsequence_split_gap_rows": 50,
            "min_subsequence_rows": 1_000,
```

```
            "train_fraction": 0.60,
            "validation_fraction": 0.20,
            "split_shuffle_trials": 2_000,
            "random_state": 42,
        },
    },
}
```

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

AVANCEZ
1829

CHALMERS
UNIVERSITY OF TECHNOLOGY