Beyond parabolic weld bead models: AI-based 3D reconstruction of weld beads under transient conditions in wire-arc additive manufacturing

Contents lists available at ScienceDirect

Journal of Materials Processing Tech.

journal homepage: www.elsevier.com/locate/jmatprotec

Beyond parabolic weld bead models: AI-based 3D reconstruction of weld beads under transient conditions in wire-arc additive manufacturing

Jan Petrik [a,*], Benjamin Sydow [b], Markus Bambach [a]

[a] Advanced Manufacturing Laboratory, ETH Zürich, Switzerland
[b] Chair of Mechanical Design and Manufacturing, Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany

ARTICLE INFO

Associate Editor: Robert Gao

Keywords:
Wire arc additive manufacturing
Automated process planning
Weld bead geometry prediction
Artificial intelligence
Artificial neural network
Gated recurrent unit
3D reconstruction

ABSTRACT

The ability to predict the geometry of the weld bead plays a key role in accurate path planning and determination of welding parameters in wire arc additive manufacturing. However, little attention has been paid to the weld bead geometry and its prediction when the deposition path is not straight. Thus, this work focuses on the 3D reconstruction of the weld bead based on the deposition path. One of the main findings of this paper is that the weld bead shape changes from a symmetrical cross-section in straight portions of the path to an asymmetrical shape in non-straight regions. To predict the 3D geometry of the weld bead, an AI-based architecture called AIBead was developed. A suitable parametrization of the deposition path is proposed that is a key to train the AIBead properly and to outperform currently used parabolic models.

1. Introduction

Wire arc additive manufacturing (WAAM) is directed energy deposition (DED) process that utilizes arc welding to manufacture metal parts. The welding torch is guided along a deposition path using either a CNC machine or an industrial robot, so that 3D shapes can be manufactured. Compared to other additive manufacturing technologies, the WAAM technology has a bigger build envelope, higher production rates and higher material utilization (Karunakaran et al., 2010). Furthermore, the metal parts fabricated using WAAM have favorable mechanical properties such as high density and strength comparable to those manufactured with traditional manufacturing methods (Mughal et al., 2006). Finally, WAAM is considered to have lower process chain costs and lower environmental impact than other additive manufacturing technologies (Cunningham et al., 2017; Priarone et al., 2020).

As visualized in Fig. 1, predicting the weld bead geometry plays a crucial role in path planning as well as in the determination of welding parameters such as wire-feed rate, travel speed, and stick-out length. Thus, error-free welding is conditioned by the exact model of the weld bead shape including the prediction of the geometry within radii. The importance of addressing the forecast of the weld bead geometry in the radius is demonstrated in Fig. 2. Here Fig. 2(c) and (d) visualizes the cross-sections of the radii R0.001 mm and R16 mm (see Fig. 2(a) and (b)). It can be seen that radius R0.001 mm, unlike R16 mm, exhibits a wavy surface, which is caused by the asymmetry of the weld bead geometry inside the radius. This geometric asymmetry affects the calculation and value of ideal weld bead overlaps, which is not taken into account by current methods assuming a parabolic shape. In addition, the changing orientation of the individual weld beads after each layer shown in Fig. 2(c) is caused by a change in the welding direction.

Scientific works dealing with weld bead geometry prediction can be divided into two categories according to the used method: classical, using mainly regression analysis, and artificial intelligence approaches utilizing MLP (multilayer perceptron).

Suryakumar et al. (2011) uses second-order regression analysis to determine the weld bead height as a function of wire feed rate and torch speed. The weld bead width is obtained analytically using the assumption that the shape of the weld bead corresponds to a parabola. The regression model is fitted and tested on 42 cross-sections and the resulting accuracy is determined as a comparison between the real and calculated area of the parabola. The mean error and its standard deviation between the ground truth and predicted area is 0.5 ± 5.9%. Another work published by Xiong et al. (2014) utilizes also second-order regression analysis, where the authors use wire feed rate, welding speed, arc voltage, and nozzle-to-plate distance as model input parameters. The output of their model is the width and height of the weld bead. The

[*] Corresponding author.
E-mail address: jpetrik@ethz.ch (J. Petrik).

https://doi.org/10.1016/j.jmatprotec.2021.117457
Received 15 October 2021; Received in revised form 26 November 2021; Accepted 2 December 2021
Available online 9 December 2021

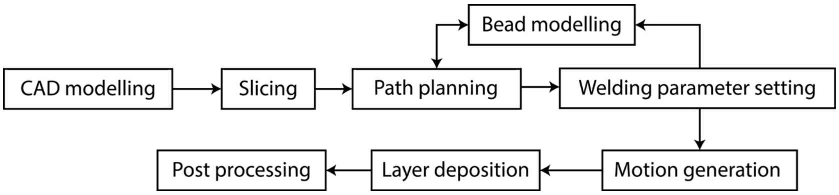
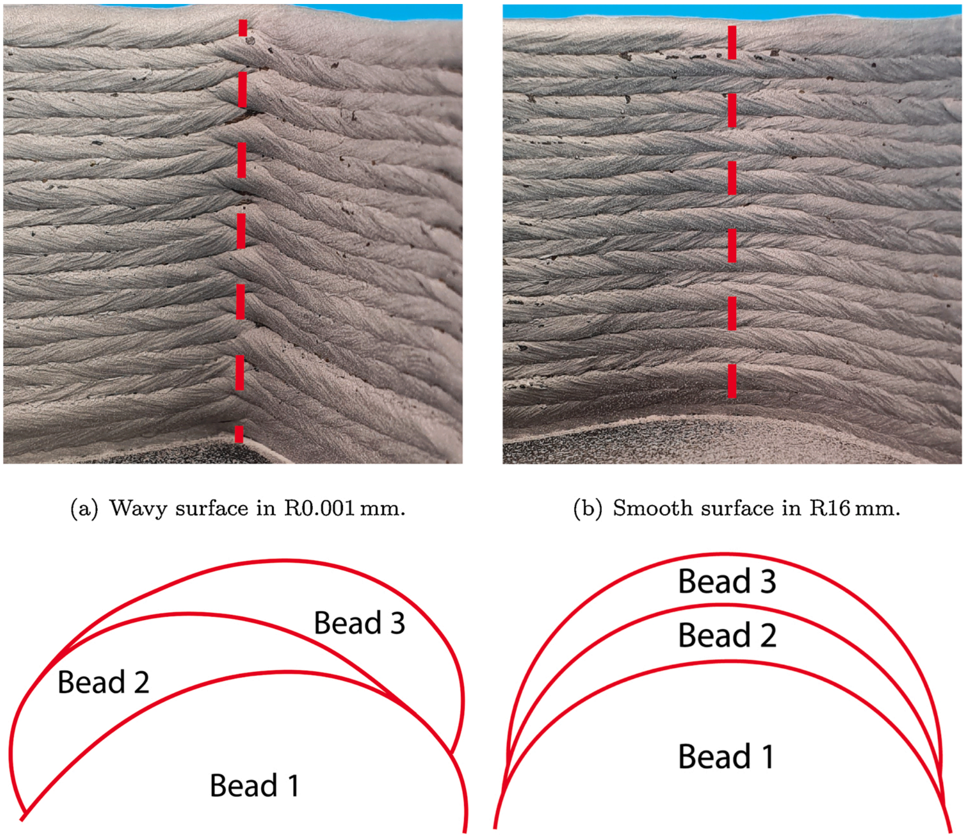


Fig. 1. WAAM process chain (Karmuhilan et al., 2018).



(a) Wavy surface in R0.001 mm.

(b) Smooth surface in R16 mm.

(c) Illustration of the cross section of R0.001 mm.

(d) Illustration of the cross section of R16 mm.

Fig. 2. Difference in surface waviness for radius R0.001 mm and R16 mm.

model is trained via 31 cross-sections and tested on a separate test set consisting of 12 samples. The average error between the real and predicted bead width, height is 2.6 ± 1.8% and 2.3 ± 1.8%, respectively. In addition, Mollayi and Eidi (2018) design a support vector regression (SVR) algorithm that, based on process parameters such as wire feed rate, welding speed, arc voltage, and nozzle-to-plate distance, computes the weld bead height and width. The authors use the same training and testing data like those presented by Xiong et al. (2014) and obtain a testing error of 5.52% for both, weld bead height and width.

One of the first models predicting weld bead geometry using artificial intelligence is described by Nagesh and Datta (2002). The goal of this work is to predict the height, width, depth of penetration, and area of penetration of the weld bead produced via shielded metal arc welding (SMAW). For this purpose, a neural network is trained, which uses electrode feed rate, arc power, arc voltage, arc current, arc length, and arc travel rate as input. The network is trained via 15 samples and tested with 3 others. Furthermore, the mean output errors for all the predicted variables are 1 ± 8.3%, 3 ± 5.2% and 5.8 ± 4.1%. Xiong et al. (2014) focus on predicting the width and height of the weld bead in gas metal arc welding (GMAW) using a neural network trained with wire feed rate, welding speed, arc voltage, and nozzle-to-plate distance. The data used in this publication are already described in the preceding paragraph. The network reaches a mean bead width and height error of 1.9 ± 1.2%, 2.1 ± 1.5%. The same goal and welding process have also been modeled using a neural network described by Ding et al. (2016), with the difference that the input data is given by the wire feed rate, travel speed, and stick-out length. Moreover, the model is trained and tested via 16 and 7 samples, respectively. There are no quantitative results listed. Kumar et al. (2017) examine the weld bead shape prediction in cold metal transfer welding. To predict the width, height, and depth of penetration, a neural network is trained, which uses welding current, speed, voltage, and shielding gas flow rate as input parameters. The numbers of the cross-sections used for training and testing are 27 and 6, respectively. Afterwards, the resulting mean errors are 2.1 ± 1% for the width, 1 ± 0.3% for the height, and 8.1 ± 1.2% for the depth of penetration. Weld bead height and width manufactured via wire arc additive manufacturing (WAAM) process were predicted using a neural network outlined by Karmuhilan et al. (2018), which was trained on voltage,

Table 1
Review of the literature on weld bead geometry prediction. All abbreviations are explained in the table legend.

	Input[a]	Output[b]	Model	Error [%]
Suryakumar et al. (2011)	WFR, WS	BH	2nd Regres.	0.5 ± 5.9
Xiong et al. (2014)	WFR, WS V, NTPD	BH, BW	2nd Regres.	2.6 ± 1.8, 2.3 ± 1.8
Mollayi and Eidi (2018)	WFR, WS V, NTPD	BH, BW	SVR	5.52
Nagesh and Datta (2002)	WFR, V, P I, L, WS	BH, BW DOP	MLP	1 ± 8.3, 3 ± 5.2, 5.8 ± 4.1
Xiong et al. (2014)	WFR, WS V, NTPD	BH, BW	MLP	1.9 ± 1.2, 2.1 ± 1.5
Ding et al. (2016)	WFR, WS SL	BH, BW	MLP	–
Kumar et al. (2017)	I, WS V, SGFR	BH, BW DOP	MLP	2.1 ± 1, 1 ± 0.3 8.1 ± 1.2
Karmuhilan et al. (2018)	V, WS WFR	BH, BW	MLP	–
Ahmed et al. (2018)	A, L, WS, ED, WG	BH, BW DOP	RBFN	10%

[a] WFR = wire feed rate [m/min], WS = welding speed [m/min], V = arc voltage [V], NTPD = nozzle-to-plate distance [mm], I = arc current [A], L = arc length [mm], SL = stick-out length [mm], SGFR = shielding gas flow rate [lit/min], ED = electrode diameter [mm], WG = welding gap [mm], P = arc power [W].

[b] BH = bead height [mm], BW = bead width [mm], DOP = depth of penetration [mm].

Table 2
Welding parameters of GEFERTEC arc605 metal printer with Fronius TPS 400i welding source.

Welding wire	Wire diameter	Shielding gas	Stick-out length
SG2 1.5125	1.2 mm	18% CO_2, 72% Ar	10–15 mm
Welding speed	Wire feed speed	Substrate	Intermediate layer temperature
Variable (see Table 3)	5 m/min	S235JR, 250 mm × 125 mm × 5 mm	max 120 °C

Table 3
Overview of performed welding experiments with respect to welded radii, number of beads and layers. Finally, the torch speeds with which the experiments were carried out are given.

	Radii [mm]	Number of multibeads	Number of layers	Torch speed [mm/min]
Setup n.1	R0.001, R05, R1, R2, R4, R8, R12, R16	1	10	500
Setup n.2	R0.001, R1, R4, R12	1	10	250, 333
Setup n.3	R0.001, R1, R4, R8, R12, R16	1	10	250, 333, 500

wire feed rate, and welding speed. The authors use 14 samples for training and the other 4 for testing purposes. The resulting percentage errors for the test data are not listed. Ahmed et al. (2018) predict SMAW bead geometry, more precisely the width, height and depth of penetration, with the help of a radial basis function network (RBFN). This architecture was trained using the SMAW parameters as welding current, arc length, welding speed, electrode diameter, and welding gap. The created dataset consists of 33 samples where 80% were used for training and the rest for testing. The mean error is 10% for all predicted variables.

All the above-mentioned publications are summarized in Table 1.

There are three main shortcomings of the above-cited literature. The first one is that all publications focus on the reconstruction of the weld bead geometry along straight deposition paths. Thus, there is no systematic analysis of the shape of weld beads deposited along curved paths such as radii. The second drawback is that no attempt was made to reconstruct the weld bead cross-section shape in a non-straight section of the deposition path. The third is the absence of a ready-to-go solution where the developed algorithm would reconstruct a 3D model of the weld bead based on the deposition path.

Therefore, the aim of this work is to find a solution to the above mentioned knowledge gaps by answering the following research questions:

- How is the weld bead geometry affected by curvature changes in the deposition path?
- What is the course of geometric characteristics of the weld bead geometry along the curvature, i.e., are there steady-state and transient regions?
- Are classical regression models sufficient to predict the weld bead geometry in the case of curvature changes or are more sophisticated models based on artificial intelligence necessary?
- What is a suitable deposition path parameterization to train a geometry prediction model?
- If transient behavior occurs when the curvature of the deposition path changes, is an MLP sufficient to reconstruct the weld bead geometry or is a more complex AI architecture (e.g., GRU, see Section 2.2) required?

Moreover, the solution to these gaps should allow for a 3D reconstruction model that takes a predefined deposition path (e.g., a G-code path) as input and outputs a 3D weld bead geometry.

Section 2 presents the developed AIBead. It consists of subsections clarifying the experimental data, GRU theory, AIBead input preparation, AIBead architecture, and finally, its training settings. The used dataset is available on demand. Afterwards, the developed machine learning model is tested and the results are presented in Section 3. Furthermore, the results of the chosen approach, its advantages in comparison with the current literature and its shortcomings are discussed in Section 4. Finally, the publication is concluded in Section 5 including hints on possible future work.

2. Machine learning model

2.1. Experimental data

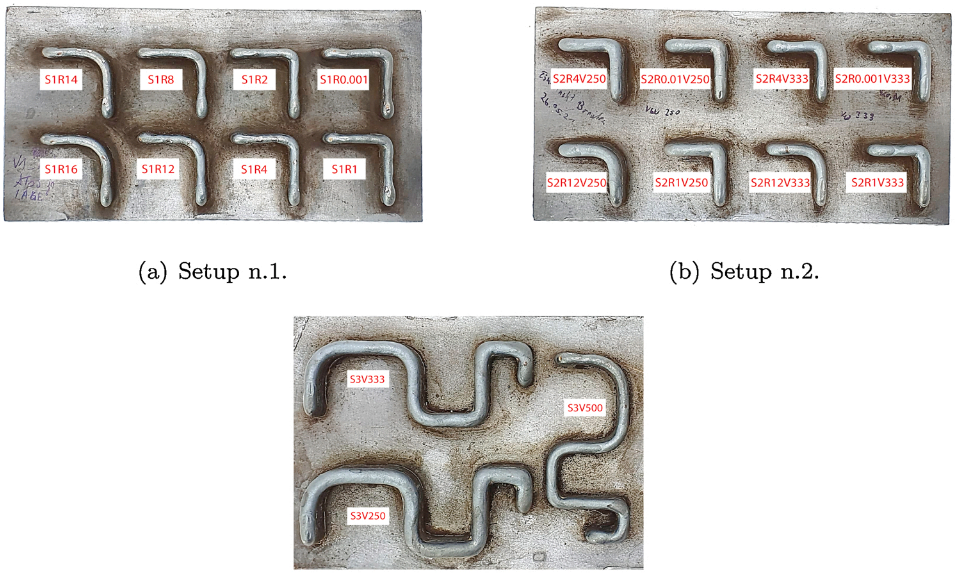
The first objective of this work was to create a dataset that would allow for a comparison of different techniques in the field of weld bead reconstruction. Therefore, weld beads were produced using a GEFERTEC 3D metal printer (see Table 2) and then scanned with an ATOS 3D scanner to obtain 3D models.

Although this work deals only with the reconstruction of single weld beads the general dataset containing multi-layer and multi-bead welds was created so that it can be used in the development of future models aiming at reconstruction that takes into account multiple weld beads. An overview of the dataset is detailed in Table 3.

From Table 3 it can be seen that the smallest radius is 0.001 mm and the largest is 16 mm. The value R0.001 mm was chosen because of the later parameterization of the deposition path, where it is necessary to assign a value greater than zero to the radius. Moreover, the difference between R0.001 mm and R0 mm is negligible and the radius R0.001 mm can be considered as a sharp corner.

Furthermore, examples of deposited and scanned weld beads are presented in Fig. 3.

In order to perform the weld bead shape analysis and prepare the data for AIBead training, the weld beads were cut along their center lines in different positions and radii. The center line is defined as the midline between the lateral contours of the weld bead, which is calculated as the



(a) Setup n.1.

(b) Setup n.2.

(c) Setup n.3.

Fig. 3. Visualization of the welded setups in different layers, which are closely described in Table 3.

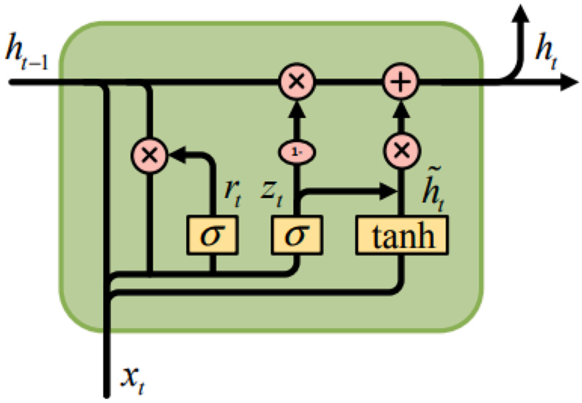


Fig. 4. Basic GRU cell architecture (Pan et al., 2018).

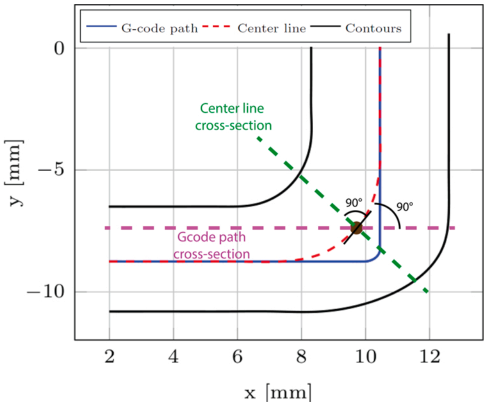


Fig. 5. Comparison of cross-sections of a weld bead in brown point with respect to center line (green dashed line) and G-code path (pink dashed line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

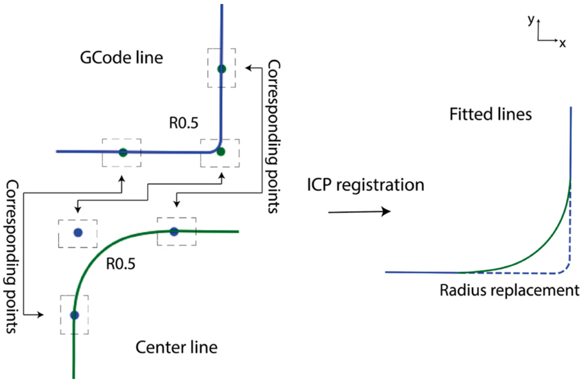


Fig. 6. A fitting algorithm that transfers the G-code path to the center line. It works on the basis of Iterative Closest Point Algorithm, which finds the transformation between the 3 correspondence points of the G-code path and the central line. Using these, the two curves are fitted to each other and the G-code path is replaced by the central line between the two extreme points.

average of their coordinates. These cross-sections are represented by 16 2D points and they are used as ground truth training and testing data.

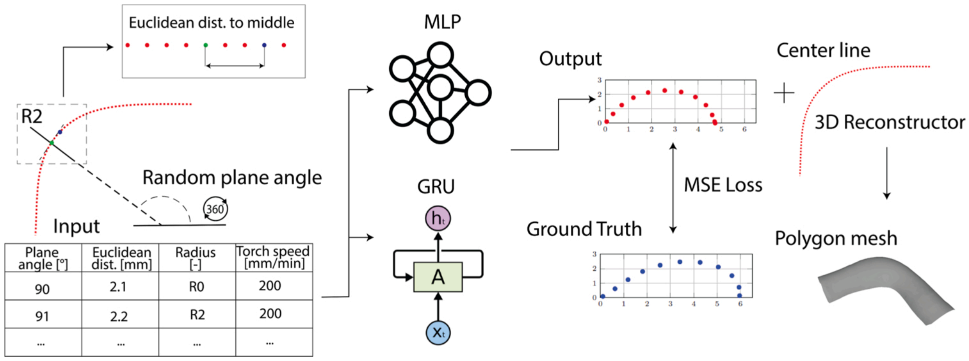
2.2. Gated recurrent unit

Gated recurrent unit (GRU) is a type of recurrent neural network introduced by Cho et al. (2014). The GRU unit is depicted in Fig. 4.

It consists of four parts. The first two are vectors called update gate z_t and reset gate r_t. The update gate decides how much information from the previous step should be passed on to the future. Conversely, the reset gate determines how much information from the previous step is to be forgotten. The third part is called candidate hidden state $\widetilde{h}_t$, which stores the relevant information determined by the reset gate. Finally, the last part is a vector named hidden state h_t, current GRU unit information holder, which is passed to the next unit.

The whole GRU workflow can be explained using the following equations. Firstly, the update gate z_t is computed as

$$z_t = \sigma\left(W^{(\mathrm{xz})} x_t + W^{(\mathrm{hz})} h_{t-1} + b^{(z)}\right) \tag{1}$$



Plane angle [°]	Euclidean dist. [mm]	Radius [-]	Torch speed [mm/min]
90	2.1	R0	200
91	2.2	R2	200
...	...	...	...

Fig. 7. AIBead architecture consisting of either MLP or GRU. Each point of the central line of the weld bead is parameterized by 4 values: plane angle, Euler distance, radius and welding speed. These values are then used as input to GRU or MLP, which predict the 2D point cloud of the weld bead cross-section. This is compared with its ground truth using the MSE loss function, and the value is then backpropagated. In addition, the individual 2D pointclouds can be positioned back on the weld bead's center line to obtain its 3D model.

Table 4
Optimization parameters and their boundaries of a genetic algorithm utilized to optimize both MLP and GRU architectures.

	Number of layers	Number of neurons per layer	Sequence length
MLP	[3, 8]	[16, 32, …, 512]	–
GRU	[1, 4]	[20, 25, …, 70]	[2, 4, 8]

Table 5
MLP parameters, where BS signifies the used batch size. In addition, the first layer input is [BS, 4].

Layer	Output	Parameters
LeakyReLU(Linear)	[BS, 32]	128
LeakyReLU(Linear)	[BS, 256]	8192
LeakyReLU(Linear)	[BS, 256]	65,536
LeakyReLU(Linear)	[BS, 32]	8192
Linear	[BS, 32]	1024

Table 6
Parameters of the GRU unit utilized in the AIBead.

	Input size	Hidden size	Number of layers	Dropout
GRU	4	50	2	0.2
	Bidirectional	Bias	Sequence length	Output size
GRU	True	True	4	32

where the input vector x_t is multiplied by its trained weight matrix $W^{(xz)}$ and is summed with the bias vector $b^{(z)}$ and the product of the output of the previous GRU cell h_{t-1} with its trained weight matrix $W^{(hz)}$. Finally, the sigmoid activation function is applied to force the output to lie between 0 and 1.

Afterwards, the reset gate r_t is computed as follows:

$$r_t = \sigma\left(W^{(xr)}x_t + W^{(hr)}h_{t-1} + b^{(r)}\right) \tag{2}$$

Eq. (2) is almost the same as the one used for the update gate. The only difference is the usage of its own trained weight matrices.

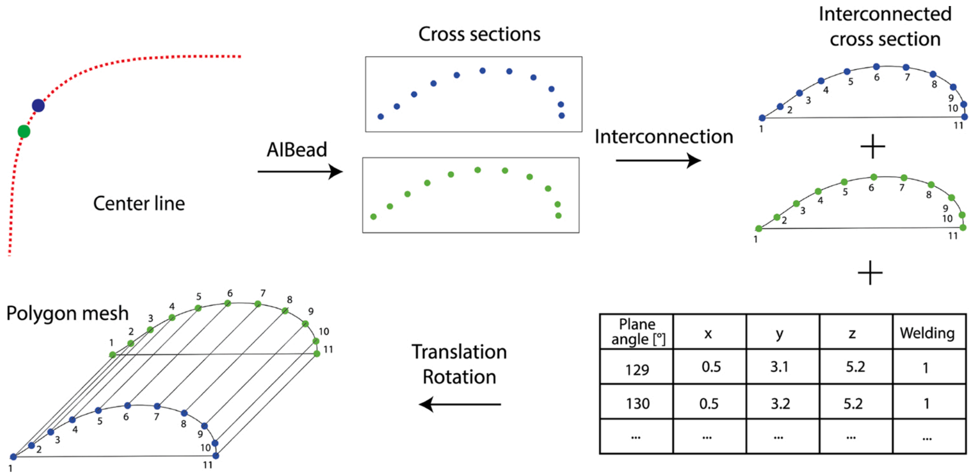
The calculation of the candidate hidden state $\widetilde{h}_t$ is given as

Table 7
Training parameters of the AIBead.

	Optimizer	Learning rate	Beta 1	Beta 2	Batch size	Epochs
GRU	Adam	0.01	0.9	0.99	32	250
MLP	Adam	0.01	0.9	0.99	32	250

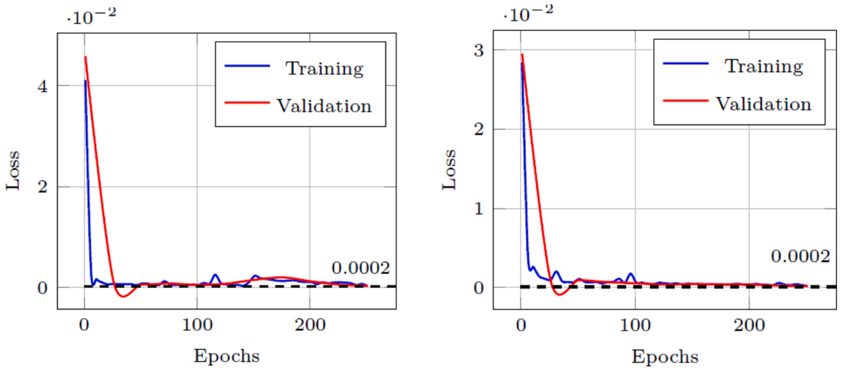
Table 8
Number of cross-sections used to train and test AIBead from each experimental setup presented in Fig. 3.

	Train	Test
Setup n.1	1002	–
Setup n.2	1054	–
Setup n.3	–	540



Plane angle [°]	x	y	z	Welding
129	0.5	3.1	5.2	1
130	0.5	3.2	5.2	1
...	...	...	...	...

Fig. 8. Polygon mesh reconstruction based on center line and AIBead architecture. This procedure is illustrated by two points on the central line, where the AIBead first predicts their 2D point clouds. Then the corresponding points between the two point clouds are marked and these are then rotated and translated to the central line using x, y, z coordinates and rotation angles. Finally they are connected to form a polygonal mesh.



(a) MLP.

(b) GRU.

Fig. 9. Training and evaluation curves for both MLP and GRU architectures.

Table 9
Geometric characteristics of the 10 cross-sections in the middle of each radius for $v = 500$ mm/min.

	Width [mm]	Height [mm]	Area [mm^2]	Circumference [mm]	Inner toe angle [°]	Outer toe angle [°]
R0.001	6.1	2.6	10.6	8.7	45	84
R1	5.7	2.5	10.5	8.5	52	86
R2	5.5	2.3	9.3	8.1	50	89
R4	4.9	2.4	8.7	7.7	58	85
R8	4.7	2.3	7.9	7.2	62	75
R16	4.7	2.2	7.8	7.2	68	66
Straight	4.5	2.3	7.6	7.1	73	66

$$\widetilde{h}_t = \tanh\left(W^{(\mathrm{xh})}x_t + (r_t \odot h_{t-1})W^{(\mathrm{hh})} + b^{(h)}\right) \tag{3}$$

It consists of three steps. First, the weight matrix $W^{(hh)}$ is multiplied with the input x_t. Secondly, the Hadamard (element-wise) product between the reset gate r_t and $W^{(hh)}h_{t-1}$ is used to determine what has to be removed from the previous time steps. Lastly, these two results are added together and the hyperbolic tangent function (tanh) activation function is applied.

The last part of the architecture is the calculation of the final hidden state h_t, which decides about which information should be collected from the current memory content $\widetilde{h}_t$ and which from the previous step h_{t-1} (see Eq. (4)):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{4}$$

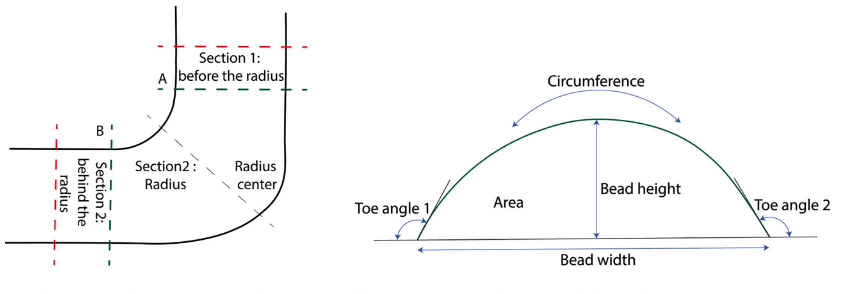
There are a couple of important parameters, which have to be defined to be able to train the GRU architecture. Besides the common ones such as input size, output size, bias utilization, or dropout rate are these given as follows:

- Hidden size, which describes the number of neurons in the hidden state h, which is depicted in Fig. 4.
- The number of layers determining the number of stacked GRUs, e.g., 3 layers means that 3 stacked GRUs were used, where the second one takes as input the output from the first one and so on.
- Bidirectionality. If a bidirectional GRU (BiGRU) is used, each GRU is composed of two GRU units, where one takes the input sequence in a forward direction and the second in a backward direction. Afterwards, both directions are considered for the output calculation.
- Sequence length, that expresses the size of the input sequence.

2.3. Models

The aim of the developed machine learning model called AIBead (see Fig. 7) is to predict the 3D geometry of the weld bead based on the G-code path. A possible AI concept to achieve this goal is either a multilayer perceptron (MLP, see (Murphy, 2021)) or GRU (see Section 2.2). These types of architectures were chosen because of the possibility to learn non-linear relationships in the data so that the non-symmetrical weld bead shape can be plausibly modeled. In addition, the GRU can potentially benefit from being able to take into account information from the surrounding of the currently computed cross-section. Finally, GRU got priority over LSTM because it seems to be computationally more efficient and converges faster (Chung et al., 2014).

AIBead is also compared with classical approaches such as second-order linear regression and Support Vector Regression (SVR, see (Awad and Khanna, 2015)). Both of these techniques are trained to



(a) Individual sections of the radius including its boundaries A and B.

(b) Examined characteristics of a weld bead cross-section.

Fig. 10. Radius sections as well as the examined geometric characteristics.

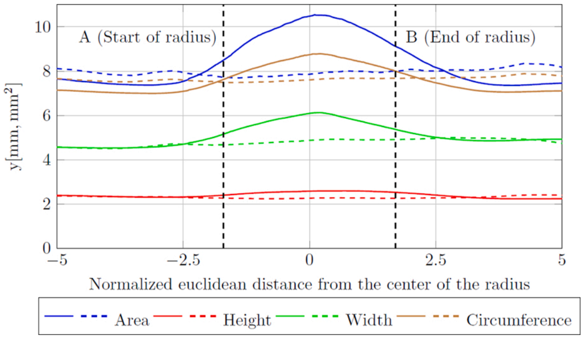


Fig. 11. Course of geometric characteristics of cross-sections cut along middle line containing R0.001 (smooth line), R8 mm (dashed line) and v = 500 m/min.

predict the coefficients a and b that affect the parabolic shape as described in Eq. (5):

$$y = \mathrm{ax}^2 + b \tag{5}$$

2.4. AIBead input preprocessing

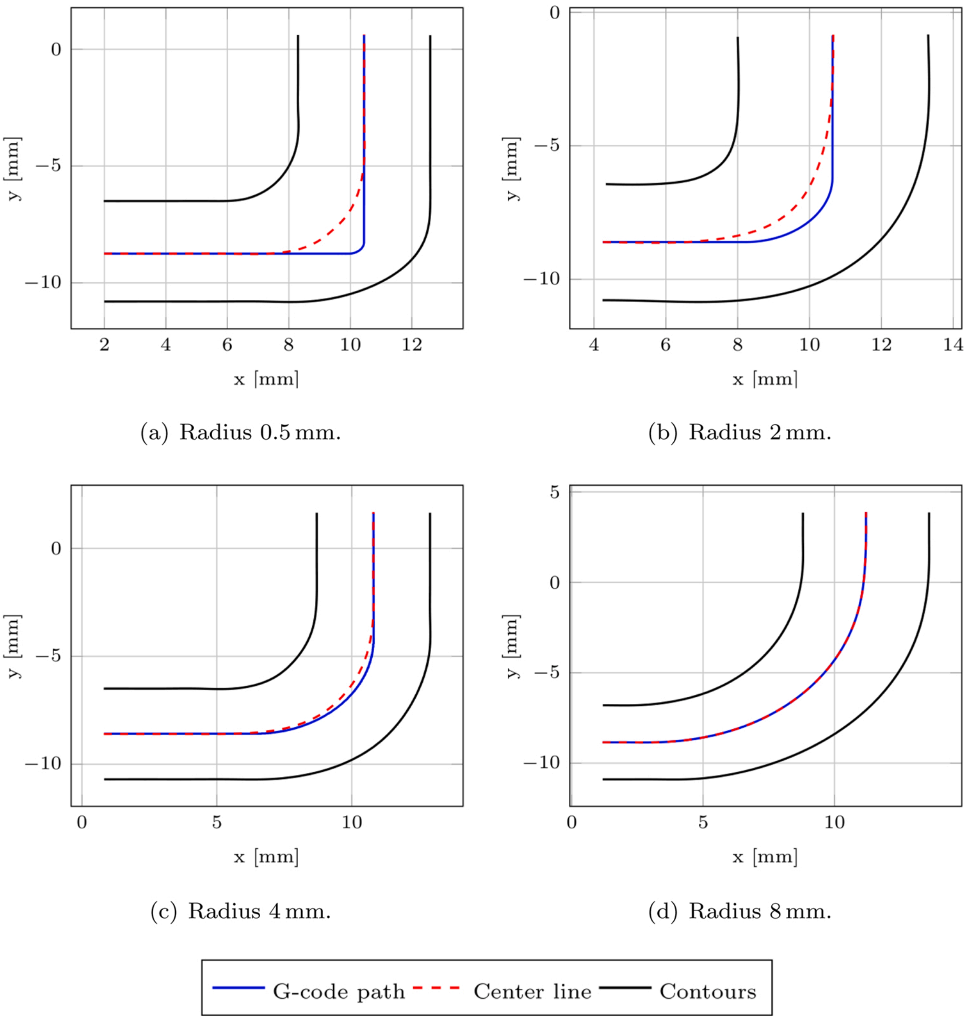
In order to train AIBead, the input G-code path needs to be replaced by a center line, which is defined as the midline between the lateral contours of the weld bead calculated as the average of their coordinates. As shown in Fig. 5, these two paths are not identical for small radii.

Furthermore, the specific reasons for this step are as follows:

- The cross-sections, which serve as training data, have to be cut along the center line in order to obtain real shape of the weld beads. This is illustrated in Fig. 5, where the cross-sections in the brown point perpendicular to the G-code path and the central line are shown as an example. Here it can be seen that only the green cross-section, which is perpendicular to the tangent in the brown point, captures the actual weld bead shape.
- AIBead is trained via cross-sections, which have been obtained as perpendicular to the tangent of the central line at each point. Thus, in order to be able to assemble a 3D model from the individual predicted cross-sections, it is necessary to place them on the central line and not on the G-code path.

Fig. 6 outlines how the G-code path is replaced by a central line using fitting method called Iterative Closest Point algorithm (ICP, see (Besl and McKay, 1992)).

For the most accurate fit, only three points are taken into account: two points that bound the radius and the center of the radius. Then the radius of the G-code path is replaced by the radius of the center line, as shown on the right side of Fig. 6.



(a) Radius 0.5 mm.

(b) Radius 2 mm.

(c) Radius 4 mm.

(d) Radius 8 mm.

Fig. 12. Difference in the center line and the G-code planning line of the weld bead. Approximately the same difference can be observed for all the examined welding velocities.

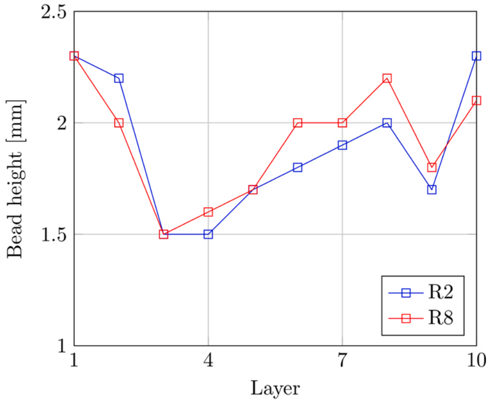


Fig. 13. Dependency of bead height on the layer number for R2 and R8 mm.

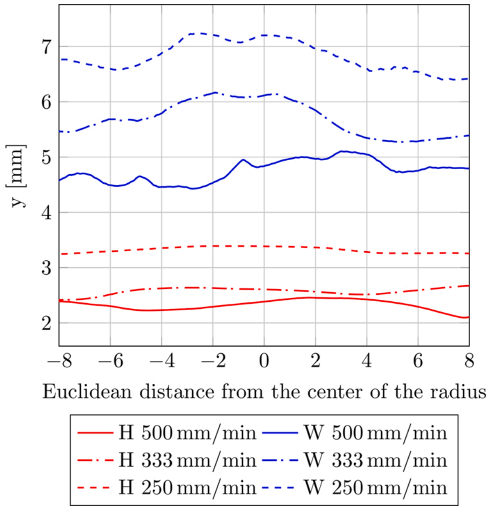


Fig. 14. Dependency of bead height and width on the velocity for R4 mm.

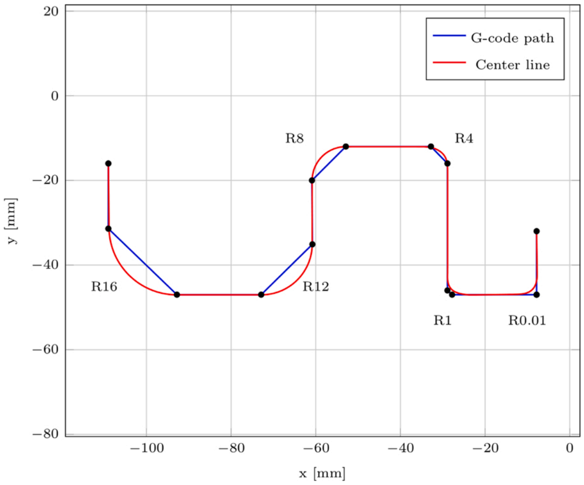


Fig. 15. Result of fitting algorithm that transfers a G-code path to weld bead center line.

2.5. Architecture

The overview of the AIBead architecture is presented in Fig. 7.

The deposition path shown in Fig. 7 is parametrized using the following variables:

- Plane angle: the angle between the tangent at the point at which the shape of the weld bead is calculated and a randomly oriented plane. Thus, this input allows to distinguish between a straight region and other radii, and also where exactly a point on a given radius lies. Furthermore, the random rotation during the training ensures, that the AIBead architecture does not overfit on specific plane rotation and therefore, is rotation invariant.
- Euler distance: the distance between the inspected cross-section and the middle of its closest radius. This input provides the architecture with information about where the examined cross-section is located within the radius. For example, whether the cross-section is in the part before or after the center of the radius. This information allows to model the delayed return of the cross-section to the shape before the radius.
- Radius: the curvature of the deposition path on which the point is located. This makes it possible to distinguish different radii, whose resulting geometry is different.
- Torch speed: the input allowing the capture of geometry changes depending on the welding speed.

These parameters serve as the input of the AIBead and form the input feature vector **x**, where

$$\mathrm{x} \in R^4 \tag{6}$$

Moreover, the output of the used AI unit, i.e., either MLP or GRU, is 16 points of the weld bead cross-section. Thus, the representation of each weld bead cross-section is a 2D point cloud.

Both MLP and GRU architectures were optimized using a genetic algorithm (see Sivanandam and Deepa, 2008), where the settings described in Table 4.

Furthermore, the resulting architectures were chosen so that the difference in the number of parameters between them does not exceed 30%. This enables their objective comparison. Specifically, MLP contains 83,680 parameters, while GRU has 65,632 parameters.

The final MLP architecture is described in Table 5.

In addition, the GRU unit settings are outlined in Table 6. Sequence length in this context means that in order to calculate the weld bead geometry at point x, it is necessary to use as input four 4D feature vectors at position $x-2$, $x-1$ and $x+1$, $x+2$.

2.6. 3D Reconstruction

To transform the individual 2D points to a polygon mesh, a 3D Reconstructor was developed (see Fig. 8).

It accepts the following inputs:

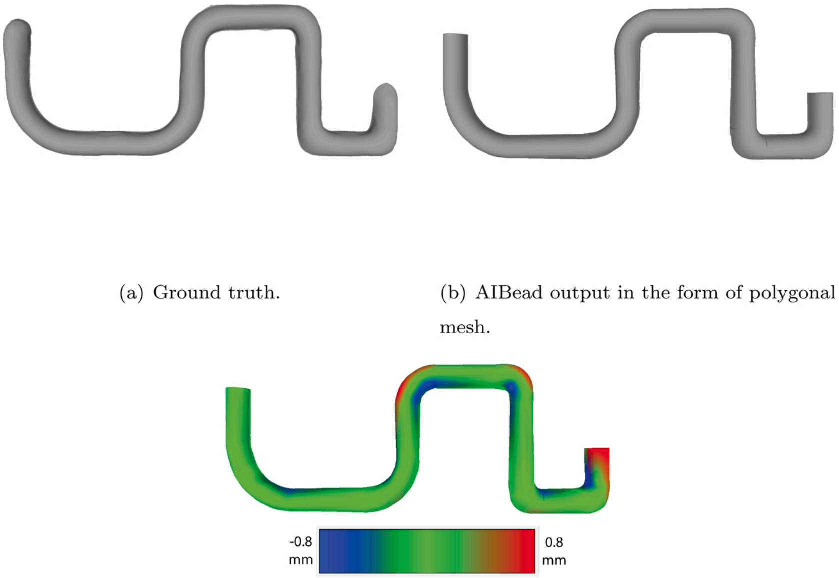
- *XYZ* coordinates of the weld bead center line.
- Binary variable expressing whether the machine was welding on a particular point of the center line.
- Tangent angle of each point creating the center line to a predefined plane.

Afterwards, it is iterated through the points of the center line and all the cross-sections are positioned to their appropriate position as well as rotated using the calculated points of the central line and their corresponding angles. Finally, the close-lying points are interconnected to create a polygon mesh.

2.7. Training

The AIBead architecture was trained with the parameters shown in Table 7.

The used loss function is defined as:



(a) Ground truth.

(b) AIBead output in the form of polygonal mesh.

(c) Hausdorff distance comparison of the ground truth mesh and the mesh produced via AIBead (Serra, 1998).

Fig. 16. Qualitative result and ground truth comparison of a weld bead S3V250 from test setup n.3 shown in Fig. 3.

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{7}$$

where y_i is the ground truth vector with X, Y coordinates and $\hat{y}_i$ is the corresponding prediction.

In addition, the data presented in Table 3 were split into training and testing sets to be able to examine the network performance (see Table 8). Finally, the train set is internally split into a final training set and a validation set in the ratio of 9:1.

The training and evaluation curves for MLP and GRU are shown in Fig. 9.

3. Results

3.1. Data analysis

First, the results of the data analysis of weld beads with various radii are presented. Table 9 reveals the mean geometric characteristics of 10 cross-sections in the middle of each radius.

Based on Table 9, a general rule can be observed: the smaller the radius, the greater the width, height, area and circumference of its cross-section. These geometrical properties, visualized in Fig. 10(b), are plotted along the center line for R0.001 and R8 mm in Fig. 11. Moreover, Fig. 10(a) shows the sections into which the radius is divided and where the start and end of the radius is.

Thanks to Fig. 11, two phenomena can be noticed. First, the change of geometric characteristics based on the radius change is observable only in R0.001 mm. Second, the weld bead does not return to its original shape immediately after completion of the radius, but there is a noticeable transient period.

Another of the discovered changes while welding a radius is the offset between the center line of the weld bead and the G-code path (see Fig. 12).

The trend is the same as in Fig. 11, namely that the difference between the two lines increases with smaller radius. Moreover, the difference is no longer noticeable for R8 mm. To take this behavior into account, a fitting algorithm was programmed as described in Section 2.4.

The weld bead behavior in respect to other beads is examined as well. Fig. 13 shows the dependency of weld bead height on the welded layer.

It reveals a non-linear growth in height along the build direction, where the weld bead height does not remain constant, but firstly drops and afterwards, grows progressively.

Finally, as shown in Fig. 14, the velocity also has a significant influence on the final shape of the weld bead. Generally speaking, the higher the velocity, the smaller the height and the width of a weld bead.

3.2. AIBead

The AIBead architecture is tested with setup n.3 presented in Fig. 3 (c). As shown in Fig. 15, the developed fitting algorithm is first executed on the G-code path to obtain the center line of the weld bead, which serves as the AIBead input.

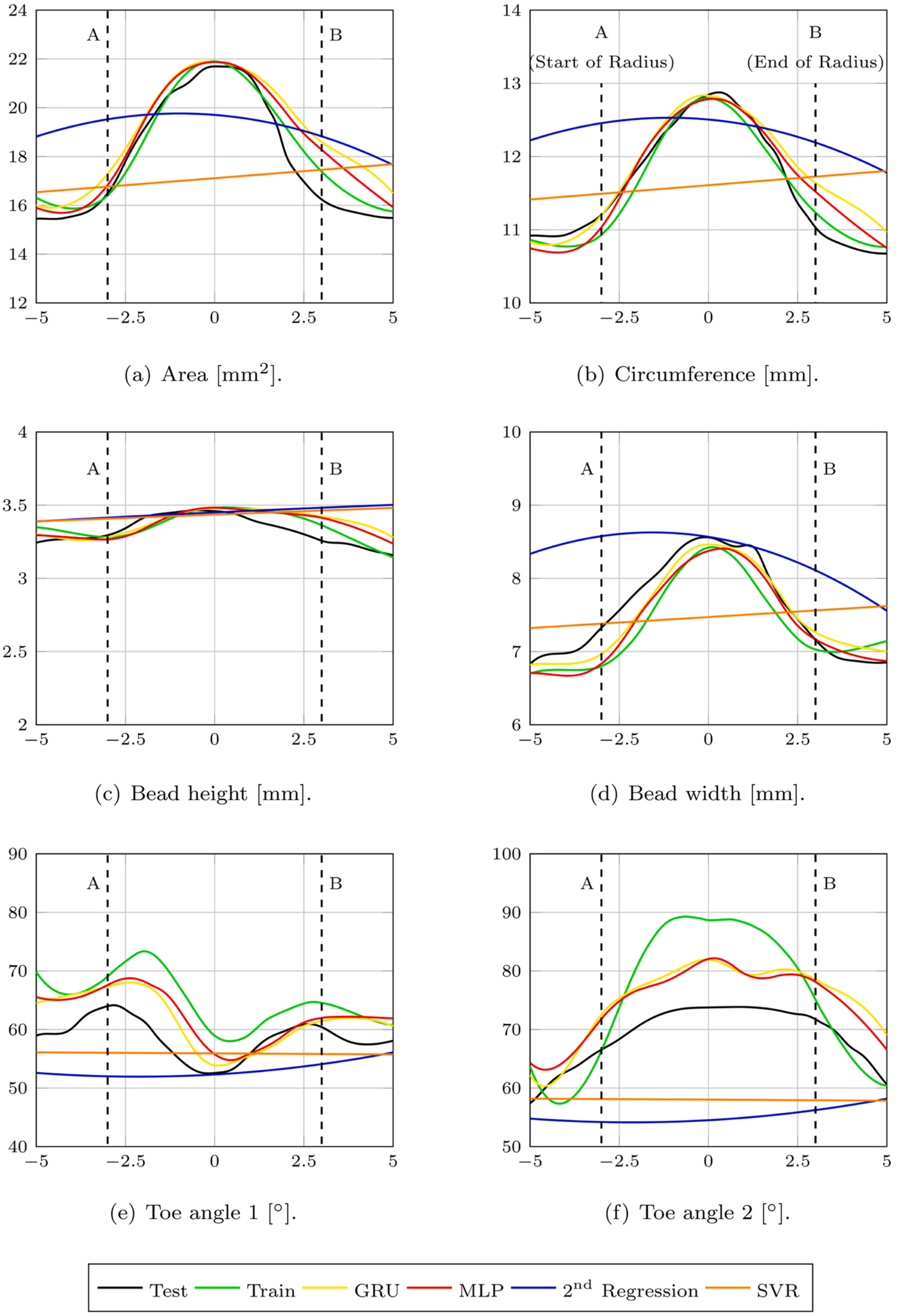
Second, the AIBead architecture reconstructs a 3D model based on the center line. The results for a speed of 250 mm/min and the ground truth are shown in Fig. 16.

Figs. 17–20 show the course of the geometric characteristics along the deposition path for 4 different radii. Furthermore, the geometric characteristics plotted on these figures are explained in Fig. 10(b).

The geometric characteristics course visualized via the last 4 figures can be summarized using Fig. 21. The general rule that applies here is that the smaller the radius, the smaller the central stable section, which for example at R0.001 mm almost disappears. On the other hand, with R8 mm it can be seen that the transient sections have vanished.

Fig. 22 shows the cross-sections for the velocity of 250 mm/min in the middle of four different radii where the weld geometry is most deformed and therefore most difficult to predict (see Fig. 10(a)). In addition, the results for the cross-section before and after the radius are shown in Figs. 23 and 24 . The ground truth cross-sections are compared here with MLP, GRU, and also with second-order regression and SVR.

The quantitative results based on MSE are listed in Table 10. The results are obtained for three different regions of all radii that form the test setup. The first two regions are located before the beginning and after the end of the radius, where the shape is affected by a radius, and



(a) Area [mm^2].

(b) Circumference [mm].

(c) Bead height [mm].

(d) Bead width [mm].

(e) Toe angle 1 [°].

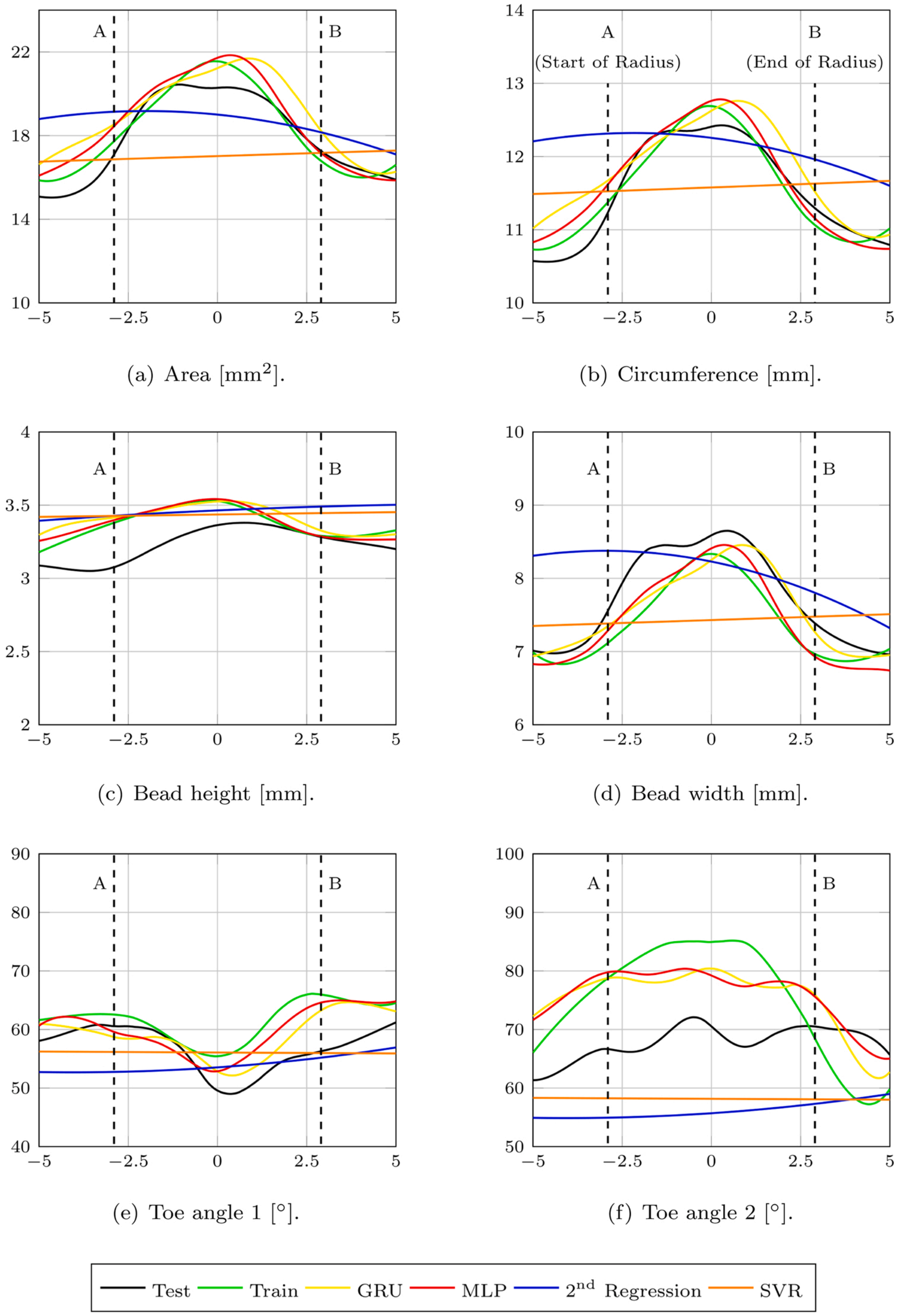
(f) Toe angle 2 [°].

Fig. 17. The course of geometric characteristics of the weld cross-sections at a radius of 0.001 mm. *X*-axis expresses the normalized Euclidean distance from the center of the radius.

the middle of a radius, where the weld bead shape experiences the maximum deformation (see Figs. 10(a) and 22). In addition, each of these regions consists of 10 cross sections, and the results are given as their mean and standard deviation. Finally, to compare the AIBead with the state-of-the-art, the same cross sections were also predicted using second-order regression and SVR.

4. Discussion

The results of this publication first show the steady state and transient shapes of the weld bead when the deposition path changes its curvature. The differences between straight and curved sections is smaller the larger the radius and is no longer noticeable for R8 mm and for the material and welding settings used in this work (see Fig. 11). In addition, this publication justified the choice of data-driven method in predicting the shape of the weld bead in the radius. The reason is that the process of welding and deposition of metal via a liquid droplet on the surface is a very complex multi-physical process, which contains various phenomena such as the Lorenz force, the pressure of the electric arc, gravity, the movement of the melted pool, all of which affect the geometry of the weld bead. The combined effect of these phenomena causes, e.g., the waviness of the surface shown in Fig. 2(a) and the



(a) Area [mm^2].

(b) Circumference [mm].

(c) Bead height [mm].

(d) Bead width [mm].

(e) Toe angle 1 [°].

(f) Toe angle 2 [°].

Fig. 18. The course of geometric characteristics of the weld cross-sections at a radius of 1 mm. *X*-axis expresses the normalized Euclidean distance from the center of the radius.

deviation from the shape of the parabola at the small radii shown, for example, in Figs. 2(c) and 22. However, it is not yet possible to predict such complex behavior accurately enough via physically based simulation. Finally, it is demonstrated that this shape is more accurately predicted using the developed AIBead than the state-of-the-art methods utilizing a parabola for weld bead shape approximation.

The current literature uses either classical methods such as second-order regression, SVR presented in Suryakumar et al. (2011), Xiong et al. (2014) and Mollayi and Eidi (2018) or a machine learning regressor summarized by Barrionuevo et al. (2021) to predict the parabola parameters that approximate the weld bead. However, all these procedures use only process parameters such as current, voltage, torch speed, or wire feed rate as input and specify the parameters of the parabola as output. Thus, they do not take into account the deposition path and are not directly usable for the planning algorithm. In contrast, AIBead solves this shortcoming by parameterizing the deposition path, which serves as input together with the welding speed. Therefore, this solution is able to accurately model the geometry of the weld bead cross-section depending on its position. Finally, this approach allows the output not only to predict the shape of the individual cross-sections but also to stack them together and create the final 3D model (see Fig. 16). Thus, AIBead can be used as a building block for developing an

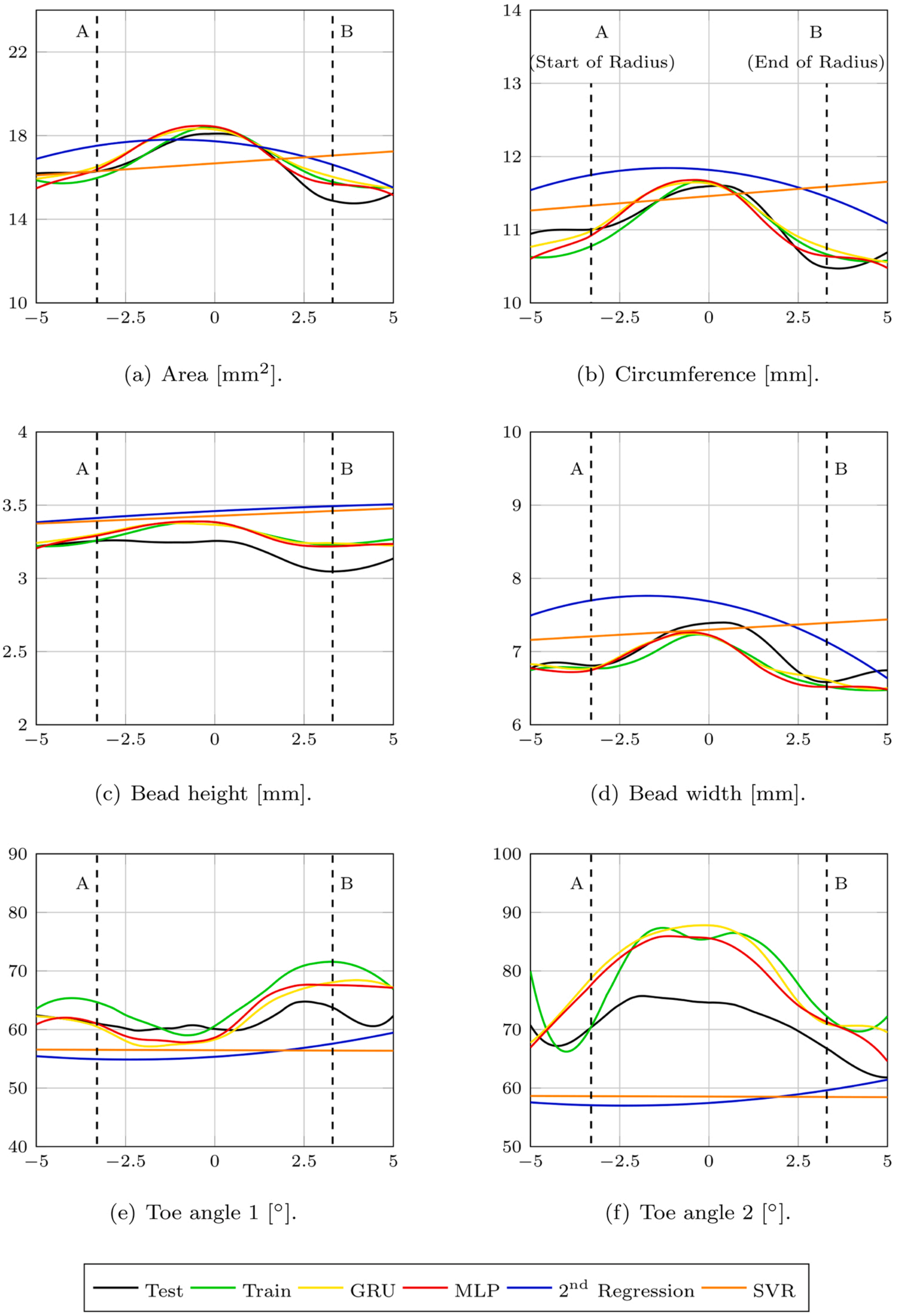
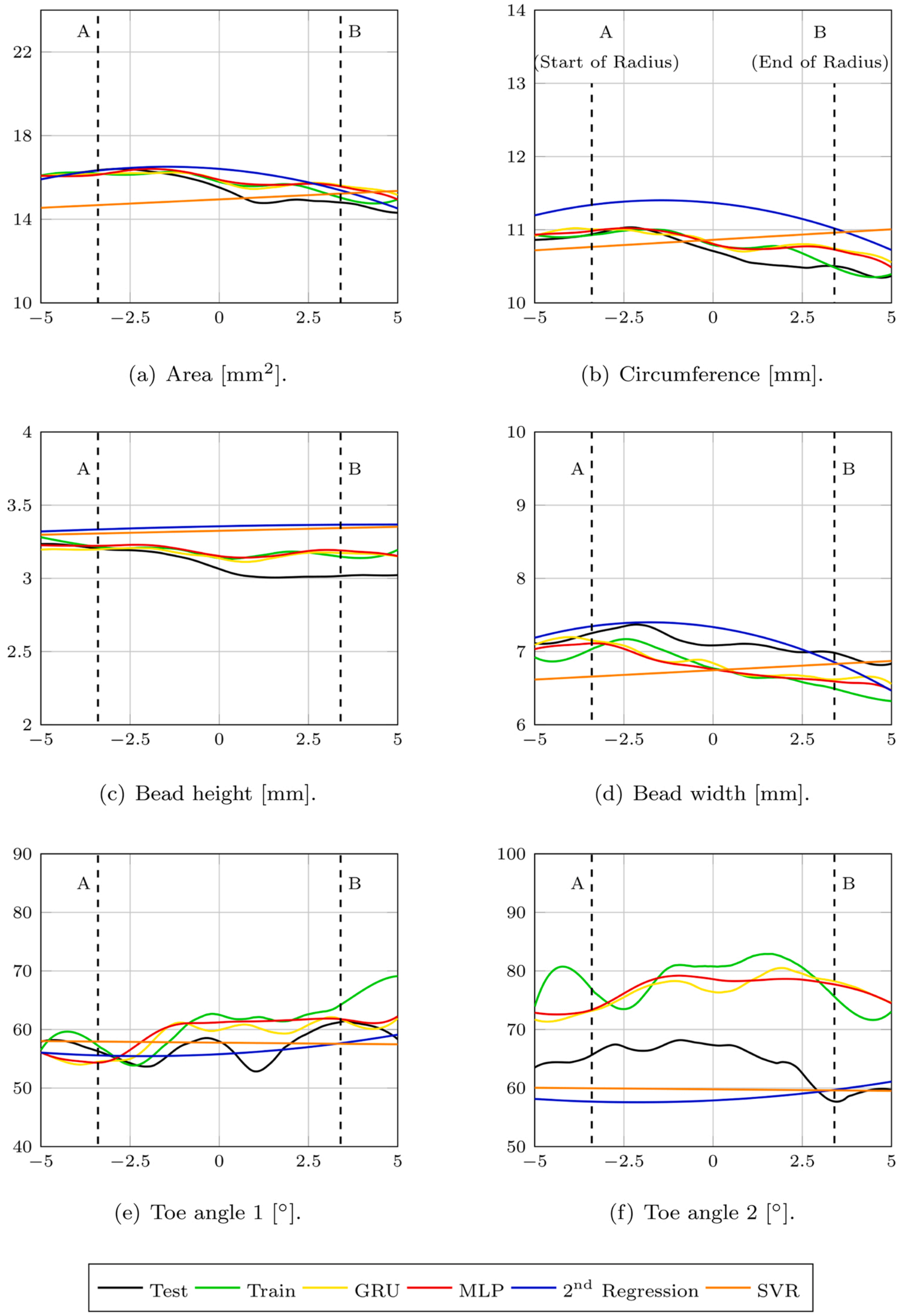


Fig. 19. The course of geometric characteristics of the weld cross-sections at a radius of 4 mm. *X*-axis expresses the normalized Euclidean distance from the center of the radius.

algorithm for planning the deposition path using, for example, volume optimization.

In Figs. 17–20, which show the course of geometric characteristics of the cross-sections along the central line of the weld bead, it is possible to notice three facts. First, these results correspond to the data shown in Table 10, namely that AIBead beats the previously used methods in the accuracy of weld bead reconstruction. The second observation is the difference between the training and testing data. It can be seen here, as already mentioned, that welding, as a nonlinear multi-physics process, does not lead to the same weld bead shape with the same process parameters and deposition path, but there is an observable deviation. Therefore, it is essential to train AIBead on more data so that the developed model does not suffer from overfitting to a certain type of deviation, but rather is able to reconstruct a model that will correspond to the average of these geometric deviations. The third finding is the detection of the transient behavior of the geometry within the radius. As can be seen in Fig. 21, before the beginning, after the end, and in the middle of the radius the change of the weld bead geometry is stable, so there is no or only a very slight change in the geometry of the weld geometry. However, between these parts, there are sections where the weld bead geometry changes transiently. The size of these sections is larger the smaller the radius. This is well demonstrated by two different



(a) Area [mm^2].

(b) Circumference [mm].

(c) Bead height [mm].

(d) Bead width [mm].

(e) Toe angle 1 [°].

(f) Toe angle 2 [°].

Fig. 20. The course of geometric characteristics of the weld cross-sections at a radius of 8 mm. *X*-axis expresses the normalized Euclidean distance from the center of the radius.

radii, namely R0.001 mm and R8 mm. While in R0.001 mm the steady state middle part almost disappears and the whole change of the weld bead geometry within the radius is transient, in R8 mm, on the contrary, there are no observable transient sections and thus the geometry of the weld bead does not change or only very slightly within the common variations.

This work also allowed the comparison of two techniques in the field of artificial intelligence, namely the classical MLP approach and the more complex GRU approach. At the beginning, it was assumed that GRU would achieve better results because this architecture is able to take into account the surroundings of the cross-section and not only the individual cross-sections separately. This fact was considered to play an important role because of the observed transient behavior within the radius. Nonetheless, the results of both approaches turned out to be almost identical. This statement can be supported for example by the results in Table 10, Figs. 17–20 and 22 –24 . This can be also assumed on the basis of the training and validation curve shown in Fig. 9. Here it can be seen that both architectures converged to the same value of 0.002 in approximately 250 epochs. Moreover, considering the course of the training and the convergence to such a small mean squared error value, it can also be said that the training of both networks was successful, there is only negligible room for improvement and the training does not

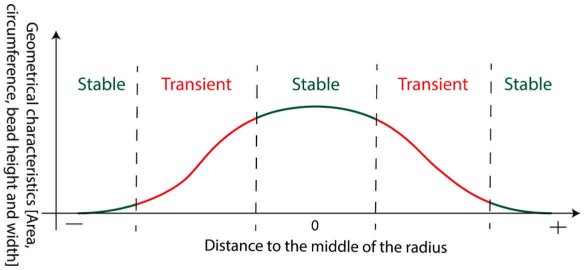
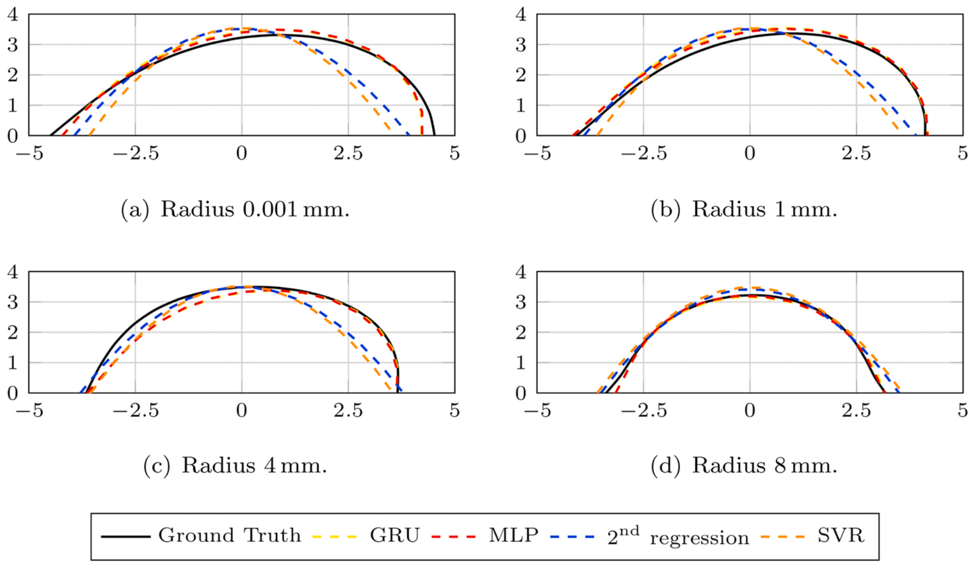

Geometrical characteristics [Area, circumference, bead height and width]
Stable
Transient
Stable
Transient
Stable
−
0
+
Distance to the middle of the radius

Fig. 21. Transient and stable sections of geometric characteristics of weld bead cross-sections in radii smaller than R8 mm.

suffer from overfitting. Finally, the answer why MLP is able to achieve the same results as GRU lies in the parameterization of the deposition path, i.e., the 4D input feature vector. This 4D vector itself contains information about the position of the cross-section within the deposition path and therefore it is redundant to use information from its surroundings as input to the AIBead.

Another discovery worth mentioning is the importance of the central line for the final 3D reconstruction. As shown in Table 10 and Fig. 22, the cross-sections are accurately predicted, but this is only a first step. Then these cross-sections are placed relative to each other on the center line in 3D space (see Fig. 16). It can be observed that there is a noticeable difference in the 3D geometry of the ground truth and reconstructed weld beads at radii R8 mm and R4 mm, even though the 2D shapes themselves are accurately predicted. This shows that the central line of these two radii is slightly different for the training and testing datasets.

The chosen approach has a couple of limitations. First, the AIBead is trained with only one weld bead per radius, which may not be sufficient to produce a sufficiently robust model. Second, it would be useful to include other process parameters as model input, such as wire feed rate or weld bead surface temperature, all of which affect the resulting weld geometry. In addition, another limitation is that the fitting algorithm



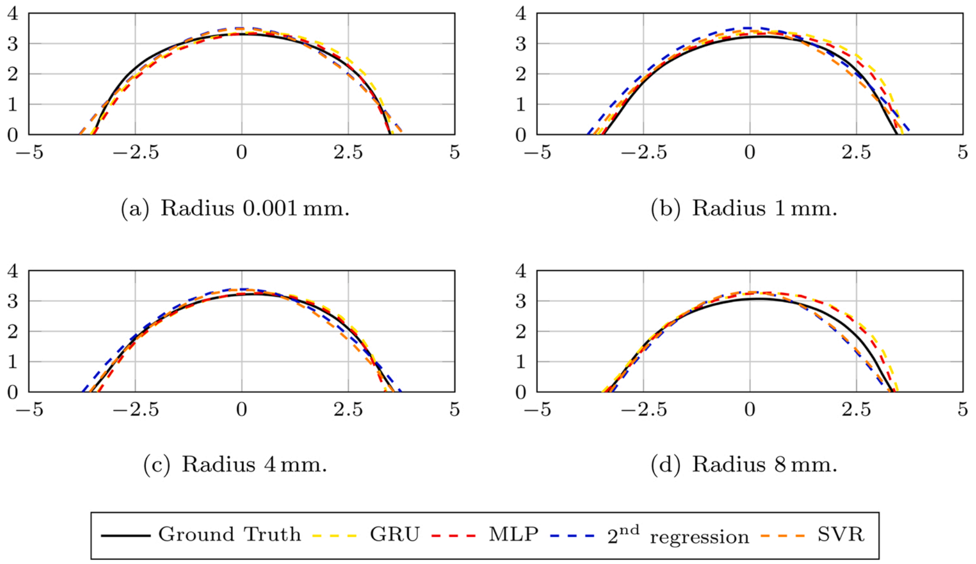
(a) Radius 0.001 mm.

(b) Radius 1 mm.

(c) Radius 4 mm.

(d) Radius 8 mm.

Fig. 22. Cross-section shapes in the middle of various radii obtained from AIBead, second-order regression and SVR for the welding velocity of 250 mm/min. In addition, both axes are in millimeters.



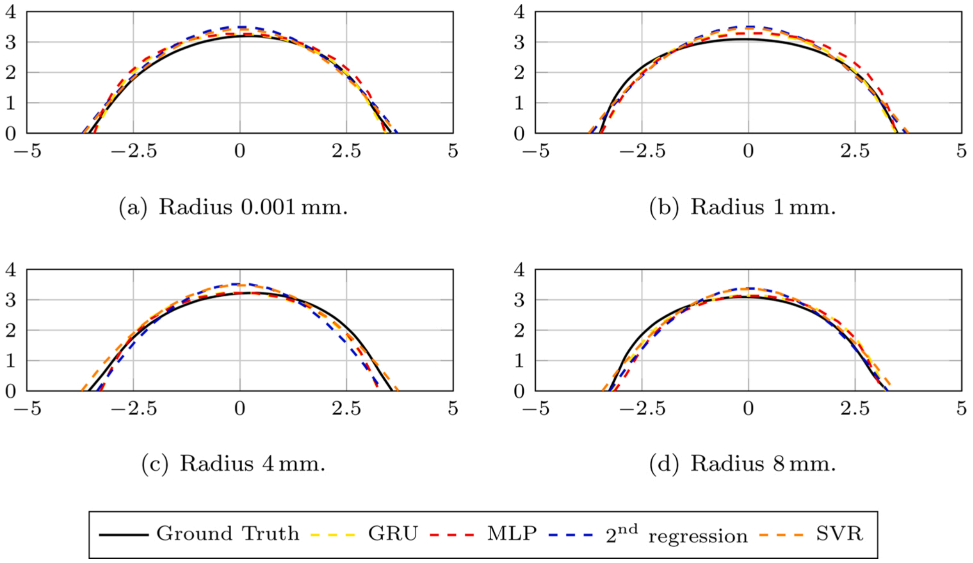
(a) Radius 0.001 mm.

(b) Radius 1 mm.

(c) Radius 4 mm.

(d) Radius 8 mm.

Fig. 23. Cross-section shapes before the beginning of various radii obtained from AIBead, second-order regression and SVR for the welding velocity of 250 mm/min. In addition, both axes are in millimeters.



(a) Radius 0.001 mm.

(b) Radius 1 mm.

(c) Radius 4 mm.

(d) Radius 8 mm.

Fig. 24. Cross-section shapes after the end of various radii obtained from AIBead, second-order regression and SVR for the welding velocity of 250 mm/min. In addition, both axes are in millimeters.

Table 10
Results in the form of MSE and their standard deviations for the part before the beginning, in the center and after the end of each radius in the test setup n.3. Moreover, the results are averaged for all three velocities and given in millimeters. Finally, the minimum MSE value for each examined part is given in bold.

		GRU	MLP	SVR	2nd Regres.
R0.001	Before	0.2 ± 0.07	**0.2 ± 0.02**	1.2 ± 0.03	1.5 ± 0.08
	Middle	**0.1 ± 0.06**	0.2 ± 0.06	4.1 ± 0.3	2.2 ± 0.2
	After	**0.6 ± 0.1**	**0.6 ± 0.1**	1.9 ± 0.2	2 ± 0.4
R1	Before	**0.2 ± 0.04**	0.2 ± 0.07	1.4 ± 0.05	1.8 ± 0.02
	Middle	**0.1 ± 0.01**	**0.1 ± 0.01**	3.3 ± 0.08	2.3 ± 0.02
	After	0.2 ± 0.02	**0.2 ± 0.01**	1.1 ± 0.06	1.2 ± 0.02
R4	Before	**0.4 ± 0.02**	**0.4 ± 0.02**	0.8 ± 0.1	0.6 ± 0.01
	Middle	**0.2 ± 0.01**	0.3 ± 0.01	1.4 ± 0.03	1 ± 0.02
	After	**0.3 ± 0.01**	0.4 ± 0.01	0.8 ± 0.02	0.6 ± 0.01
R8	Before	0.3 ± 0.08	**0.2 ± 0.06**	0.8 ± 0.05	0.9 ± 0.09
	Middle	**0.3 ± 0.02**	**0.3 ± 0.02**	1.1 ± 0.02	0.7 ± 0.02
	After	**0.4 ± 0.02**	0.4 ± 0.04	1.1 ± 0.03	0.8 ± 0.07
R12	Before	**0.08 ± 0.03**	**0.08 ± 0.03**	1 ± 0.06	1.2 ± 0.04
	Middle	**0.05 ± 0.01**	**0.05 ± 0.01**	1.3 ± 0.02	1.2 ± 0.03
	After	0.1 ± 0.01	**0.09 ± 0.01**	1.3 ± 0.04	1.5 ± 0.01
R16	Before	**0.1 ± 0.01**	**0.1 ± 0.01**	1 ± 0.04	1 ± 0.04
	Middle	0.07 ± 0.01	**0.06 ± 0.01**	1 ± 0.02	1 ± 0.02
	After	**0.1 ± 0.01**	**0.1 ± 0.01**	1.2 ± 0.03	1.3 ± 0.1

only works with radii that are available in the training dataset. The solution would be to develop a model that would be able to interpolate between the center lines of the above-mentioned radii. Moreover, it is necessary to extract the multiple center lines within each radius and use the averaged one to get more robust results. Finally, the AIBead focuses only on the single weld bead, which is not sufficient to simulate real word objects. In other words, the interaction behavior between weld beads that are either on top of each other or next to each other has yet to be developed.

5. Summary and conclusions

First of all, it was demonstrated that the geometry of the weld bead produced in the experiments is subject to deformation at radii smaller than 8 mm, which reaches its maximum in the center of the radius. Moreover, these radii are characterized by transient phases, where the geometry of the weld bead changes the most. The smaller the radius, the more these transient sections increase towards the center of the radius. On the other hand, these transient sections disappear for large radii and therefore their geometry does not change along the deposition path. For these reasons, it is not possible to approximate the geometry of the weld bead at small radii accurately enough using the parabola, which is utilized by traditional regression models. As a solution, the data-driven architecture AIBead was developed, which shows better results than the regression models. The parameterization of the deposition path, where the developed 4D feature vector was used, proved to be the key for the successful training of this architecture. First, it consists a plane angle, defining the angle between the tangent at the point for which the weld shape is calculated and a randomly rotated plane. Furthermore, it contains the Euler distance between the calculated cross-section and the center of its nearest radius, and the radius at which the point is located. Finally, it incorporates the welding speed. In the end, it turned out that thanks to the above-mentioned parameterization it was not necessary to use more complicated GRU, but MLP was enough to get the same results.

To summarize, a data-driven model called AIBead has been developed that is able, based on G-code path, to calculate a 3D model of a weld bead radius that undergoes transient behavior.

In future work, the authors plan to focus on the following points:

- Train AIBead with more data. This means with more radii, welding velocities and also add weld beads with various wire feed rates.
- Develop an interpolation model for the center lines of different radii.
- Incorporate the surface temperature of the weld bead as an input to the model.
- Extend the existing model with the possibility of multi-layer and multi-bead reconstruction.

Authors’ contribution

Jan Petrik: conceptualization, methodology, writing – original draft, writing – review & editing, software, validation. Benjamin Sydow: data curation. Prof. Dr. Markus Bambach: supervision.

Declaration of Competing Interest

The authors report no declarations of interest.

References

Ahmed, A.N., Noor, C.M., Allawi, M.F., El-Shafie, A., 2018. Rbf-nn-based model for prediction of weld bead geometry in shielded metal arc welding (smaw). Neural Comput. Appl. 29, 889–899.

Awad, M., Khanna, R., 2015. Support vector regression. Efficient Learning Machines 67–80.

Barrionuevo, G.O., Ríos, S., Williams, S.W., Ramos-Grez, J.A., 2021. Comparative evaluation of machine learning regressors for the layer geometry prediction in wire arc additive manufacturing. 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT) 186–190.

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-d shapes. Sensor Fusion IV: Control Paradigms and Data Structures 586–606.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555.

Cunningham, C., Wikshåland, S., Xu, F., Kemakolam, N., Shokrani, A., Dhokia, V., Newman, S., 2017. Cost modelling and sensitivity analysis of wire and arc additive manufacturing. Proc. Manuf. 11, 650–657.

Ding, D., Shen, C., Pan, Z., Cuiuri, D., Li, H., Larkin, N., van Duin, S., 2016. Towards an automated robotic arc-welding-based additive manufacturing system from cad to finished part. Comput.-Aided Des. 73, 66–75.

Karmuhilan, M., et al., 2018. Intelligent process model for bead geometry prediction in waam. Mater. Today: Proc. 5, 24005–24013.

Karunakaran, K., Suryakumar, S., Pushpa, V., Akula, S., 2010. Low cost integration of additive and subtractive processes for hybrid layered manufacturing. Robot. Comput.-Integr. Manuf. 26, 490–499.

Kumar, N.P., Devarajan, P.K., Vendan, S.A., Shanmugam, N., 2017. Prediction of bead geometry in cold metal transfer welding using back propagation neural network. Int. J. Adv. Manuf. Technol. 93, 385–392.

Mollayi, N., Eidi, M.J., 2018. Application of multiple kernel support vector regression for weld bead geometry prediction in robotic gmaw process. Int. J. Electr. Comput. Eng. 8, 2310.

Mughal, M., Fawad, H., Mufti, R., 2006. Three-dimensional finite-element modelling of deformation in weld-based rapid prototyping. Proc. Inst. Mech. Engrs. Part C J. Mech. Eng. Sci. 220, 875–888.

Murphy, K.P., 2021. Probabilistic Machine Learning: An introduction. MIT Press. probml.ai.

Nagesh, D., Datta, G., 2002. Prediction of weld bead geometry and penetration in shielded metal-arc welding using artificial neural networks. J. Mater. Process. Technol. 123, 303–312.

Pan, F., Li, J., Tan, B., Zeng, C., Jiang, X., Liu, L., Yang, J., 2018. Stacked-gru based power system transient stability assessment method. Algorithms 11, 121.

Priarone, P.C., Pagone, E., Martina, F., Catalano, A.R., Settineri, L., 2020. Multi-criteria environmental and economic impact assessment of wire arc additive manufacturing. CIRP Ann. 69, 37–40.

Serra, J., 1998. Hausdorff distances and interpolations. Comput. Imaging Vision 12, 107–114.

Sivanandam, S., Deepa, S., 2008. Genetic algorithms. Introduction to Genetic Algorithms 15–37.

Suryakumar, S., Karunakaran, K., Bernard, A., Chandrasekhar, U., Raghavender, N., Sharma, D., 2011. Weld bead modeling and process optimization in hybrid layered manufacturing. Comput.-Aided Des. 43, 331–344.

Xiong, J., Zhang, G., Hu, J., Wu, L., 2014. Bead geometry prediction for robotic gmaw-based rapid manufacturing through a neural network and a second-order regression analysis. J. Intell. Manuf. 25, 157–163.