Supporting robotic welding of aluminium with a laser line scanner-based trigger definition method

1st Jaime Marco-Rider*, 2nd Tuan Anh Tran*, 3rd Eirik B. Njaastad†, 4th Olav Egeland*, 5th Andrei Lobov*

*Dept. of Mechanical and Industrial Eng. Norwegian University of Science and Technology NTNU, Trondheim, Norway

†SINTEF Manufacturing AS, Trondheim, Norway

Email: *{jaime.marco, tuan.a.tran, olav.egeland, andrei.lobov}@ntnu.no, †eirik.njaastad@sintef.no

Abstract—Automation and the use of robots for welding operations is an important research topic. Being able to automate and, thus, save time for setting up and using robotic welding for complex, large-scale structures made of reflective materials, such as aluminium, will provide clear economic and competitive advantages. However, challenges coming from the ability to accurately detect and calibrate the robot for a given physical workpiece in addition to noises, such as the reflections, make it hard to develop and demonstrate a feasible automation solution. This paper proposes combining laser line scanning technology with CAD-based analysis of a workpiece geometry to support the identification of relevant elements of the workpiece in the physical world and thus support welding operations. An extendable trigger definition method is proposed to identify features of interest in a workpiece. The method can potentially support the execution of welding sequences, which in our case can be represented as a sequence of triggers that have to be observed and followed at the robot runtime to weld the workpiece together.

Index Terms—Robotic welding, Aluminium welding, Laser line scanner, Computer-aided design

I. INTRODUCTION

Building complex products out of aluminium may require hundreds of kilometres of welding paths. Engineering such products, definition and preparation for manual welding may require weeks and months of work. The welding of aluminium demands special protective equipment and good ventilation for welders as the process involves serious HSE risks. Developing a reliable robot-based solution would bring economic advantages that, besides saving time and costs, will let humans avoid dangerous jobs.

Developing a robotized solution requires bringing together software-intensive and cyber-physical system (CPS) approaches to allow robot welding work cells to cope with uncertainties adjusting the process dynamically for the currently welded workpiece. Using Computer-Aided Design (CAD) models describing expected workpieces being currently welded together, the reliable vision system can provide the right combination for building a dynamically adjustable system.

Some gaps and limitations exist between CAD and real-world data sensing-driven approaches. CAD-based offline programming approaches [1] promise automation of program generation for robotic welding based on simulated environments. However, real-world imperfections and deviations are not considered. Thus, to transition into practical welding and achieve higher automation, sensors have to be adapted. Simultaneously, purely vision-based systems depend on image processing quality and may face difficulties against complex geometries. That is especially important when considering the sensitivity of aluminium welding to deviation.

Previous research works on different robotic welding triggers in combination with different sensor technologies are available in the literature. In [2], and [3], a hardware-based trigger generator is used to acquire the laser scanner data at fixed time intervals. A stop trigger defined by volumes outside which no seam tracking is allowed was presented in [4]. A trigger of fault detection and diagnosis that identifies when the feature points deviate from the requirements was presented in [5]. A literature review on the topic of robotic welding triggers has revealed that several challenges remain to be resolved. Hence, in this paper, we propose to support the robotic welding of aluminium with a laser line scanner-based trigger definition method that takes into account the problems inherent to aluminium welding. Such as the problematic reflectivity of the material during the laser line scanning and the high accuracy of the weld seam detection needed. In addition to taking into account the CAD-based approach needed for automating as much as possible the process.

The contributions of this paper can be summarized as follows. 1) A robotic welding trigger definition method composed of trigger creation and evaluation algorithms. 2) Trigger threshold values that can be used for correct pose-reached validation based on results from sensitivity characterization tests (performed on aluminium parts) and analyzed data from proposed evaluation metrics. 3) A procedure for creating and evaluating robotic welding triggers that can be used to verify welding paths generated from offline programming environments.

The rest of the paper is organized as follows. Section II describes key works in the state of the art. Section III presents the approach for triggers definition to assist robotic welding operations. Section IV describes an implementation, followed by Results in Section V, and Discussion in Section VI. Finally, the conclusions, and future work are outlined in Section VII.

II. STATE OF THE ART

The nominal welding paths are commonly taught to the robot in a CAD environment via a manual and offline programming process (OLP). However, the actual location and orientation of the weld joints of the workpiece will differ from the ideal CAD environment due to tolerances and imprecisions in the sim-to-real calibration. It is thus necessary to introduce sensors for aligning the nominal workpiece with the observed geometry.

Sensors for robotic welding are primarily used to detect and measure process parameters and features such as the weld joint geometry and the location and shape of the weld pool, inspecting welds for defects and assessing their quality, in addition to online control and correction of the robotic welding process [6]. Sensors for geometry measurements are utilized in seam-tracking [7], [8] and weld joint or component search, allowing the nominal robot path to be altered in response to geometrical deviations. The objective of the weld joint search is to ensure that the weld bead is accurately deposited in the weld joint. The weld joints or seams are then typically located by performing one or more measurements around the nominal starting point of each weld joint before welding. Alternatively, several weld joint positions and orientations can be found simultaneously, but less accurately, by estimating the full 3D pose of the workpiece [9].

The research on locating the weld seams using optical sensing methods can be divided into two categories: passive vision and active vision. In active vision systems, an image sensor is typically combined with an external light source, optical filters, and lenses [10]–[12]. In active measurement systems for welding, laser beams are preferred to other light sources since bright, well-focused, and narrow beams can be generated with lightweight sources. In addition, single-frequency sources allow for better filtering of undesired frequencies and do not disperse as much as full-spectrum light sources.

Several researchers have contributed to reducing the gap between simulations and observations using sensor data in robotic welding. Liu et al. [13] proposed a feature mapping algorithm for offline correction in case of workpiece or fixture redesign or change of the robot workstation layout. Chen et al. [14] detected welding start positions using a proposed two-step curve fitting method on seam lines in images from a passive stereo camera system. Based on assumptions of how the laser stripe will be distorted by the welding groove, Ding et al. [15] suggested a vision system with a shape-matching algorithm to achieve initial weld point guiding.

III. APPROACH

A. Proposed system

The proposed system architecture shown in Fig. 1 was implemented through two different subsystems running on different machines, which we call CAD and SEN, respectively. The CAD subsystem machine acquires the welding paths from the CAD model, validating that the trajectory is correct (i.e. with no collision) and sending this welding path
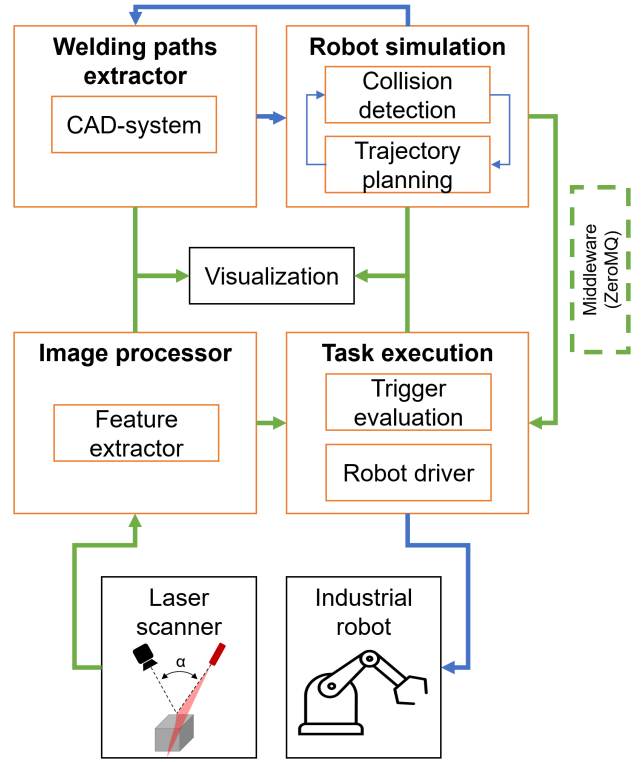

Fig. 1: Proposed system overview.

to its software-based robot controller to move the welding robot to the correct position. To achieve this, Siemens NX along with the NXOpen API was used, shown in Fig 4 with the defined weld. Geometry and weld paths were then transferred automatically using the NXOpen API [16] to the Python Robotics Toolbox [17] offline programming environment (shown in Fig. 5) for trajectory generation. Meanwhile, the SEN subsystem machine acquires images of the laser profile with the polarization image sensor [18] implemented on an industrial camera, extracting the position of the laser line in image coordinates with subpixel precision, and computing the proposed trigger. The CAD and SEN subsystem machines were connected to a small robot communication network using middleware. The middleware chosen was the ZeroMQ [19] messaging library and adopted a Publish/Subscribe pattern that provides asynchronous communication.

B. Proposed trigger definition method

Our approach for defining the trigger to start or stop the robotic welding using a laser line scanner is composed of two parts: the trigger creation and the trigger evaluation. We also consider the two cases where the trigger is either image-based (i.e. the plain image captured from the laser line scanner is used without further processing) or vector-based (i.e. the laser line coordinates extracted from the image). This difference between the image representation (Figures 2a and 2b), and vector representation (Figures 2d and 2c) is depicted in Fig. 2. In this paper, I_r, and I_w are the images of the laser used either as a reference or during the welding stage, respectively.
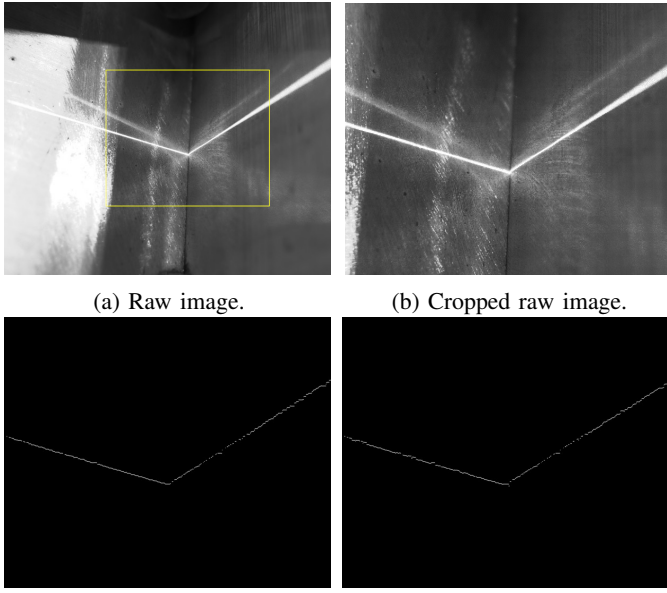
(a) Raw image. (b) Cropped raw image.

(c) FP laser extraction method. (d) COG laser extraction method.

Fig. 2: Laser line image, and vector representations.

Algorithm 1 Trigger creation

Input: Message to create trigger M_{TrigPush}
and Similarity mode Sim_{mode}
Output: None

1: **procedure** TRIGGERCREATE(M_{TrigPush}, Sim_{mode})
2:     $I_r \leftarrow I_{\text{raw}}$ ▷ Raw laser image is used as reference.
3:     $I_r \leftarrow \text{Crop}(I_r)$ ▷ Crop laser reference image.
4:     **if** Sim_{mode} is Vector-based **then**
5:         $L_r \leftarrow \text{ExtractLaser}(I_r)$ ▷ Extract laser coords.
6:         Trigger $\leftarrow L_r$ ▷ Trigger is a vector.
7:     **else if** Sim_{mode} is Image-based **then**
8:         Trigger $\leftarrow I_r$ ▷ Trigger is an image.
9:     **end if**
10:     $\text{Buffer}_{\text{Trigger}} \leftarrow \text{Push(Trigger)}$ ▷ Push trigger to buffer.
11: **end procedure**

Whereas the L_r, and L_w vectors represent the position of the laser line in image coordinates with sub-pixel precision used either as a reference or during the welding stage, respectively. Furthermore, n represents the number of columns of the laser line images.

The trigger creation procedure starts with a message containing the order to create the trigger M_{TrigPush}, plus the similarity mode chosen Sim_{mode}. Then, the raw image acquired from the laser line scanner I_{raw} is cropped and used as the reference image I_r. If the Sim_{mode} chosen is image-based, I_r is used as a trigger, whereas if the Sim_{mode} chosen is vector-based, a laser line extraction is performed on the image, and the resulting laser line coordinates vector L_r is used as a trigger. Before finishing the procedure, the trigger is pushed to the trigger buffer $\text{Buffer}_{\text{Trigger}}$. The whole procedure to create the trigger is summarized in Algorithm 1. The

Algorithm 2 Trigger evaluation

Input: Message to start evaluating trigger M_{TrigPop}
and Similarity mode Sim_{mode}
Output: Publish trigger found message M_{TrigTrue}

1: **procedure** TRIGGEREVALUATE(M_{Pop}, Sim_{mode})
2:     Trigger $\leftarrow \text{Pop}(\text{Buffer}_{\text{Trigger}})$ ▷ Pop from buffer.
3:     $I_w \leftarrow I_{\text{raw}}$ ▷ Raw laser image is used as weld.
4:     $I_w \leftarrow \text{Crop}(I_w)$ ▷ Crop laser weld image.
5:     **if** Sim_{mode} is Vector-based **then**
6:         $L_r \leftarrow$ Trigger   ▷ Use laser line coords.
7:         $L_w \leftarrow \text{ExtractLaser}(I_w)$ ▷ Extract laser coords.
8:         Similarity $\leftarrow \text{Vect}_{\text{Sim}}(L_r, L_w)$ ▷ Vector.
9:     **else if** Sim_{mode} is Image-based **then**
10:         $I_r \leftarrow$ Trigger   ▷ Use a plain image.
11:         Similarity $\leftarrow \text{Image}_{\text{Sim}}(I_r, I_w)$ ▷ Image-based.
12:     **end if**
13:     **if** Similarity $<=$ Threshold **then**
14:         Start **procedure** again
15:     **else if** Similarity $>$ Threshold **then**
16:         **return** $\text{Publish}(M_{\text{TrigTrue}})$ ▷ Publish Message.
17:     **end if**
18: **end procedure**

trigger evaluation procedure starts with a message containing the order to evaluate the trigger M_{TrigPop}, plus the Sim_{mode} chosen. Then, a previously created trigger is popped from the $\text{Buffer}_{\text{Trigger}}$, and a new raw laser image I_w is acquired from the laser line scanner and cropped. If the Sim_{mode} chosen is image-based, an image-based similarity of I_w against I_r is computed. Meanwhile, if the Sim_{mode} chosen is image-based, a laser line extraction is performed on I_w to obtain its laser coordinates L_w, and a vector-based similarity of L_w against L_r is computed. If the resulting similarity is lower or equal to a threshold, the procedure is started again, whereas it is higher than the threshold, a $M_{\text{TrigPTrue}}$ message stating that the trigger has been found is published. The whole procedure to evaluate a trigger is summarized in Algorithm 2.

C. Trigger definition method metrics

There are different metrics to compute the similarity, and/or errors, of either images [20], or features extracted from an image available in the literature. In this work, three different metrics for measuring the similarity of the laser lines extracted from an image were used: two based on computing the similarity or error between two vectors (Cosine Similarity, and Mean Squared Error), and another metric based on computing the similarity between two images (Multiscale Structural Similarity).

a) Cosine similarity: Cosine similarity is a commonly used metric [21] that is based on finding the cosine of the angle between two multidimensional vectors. The cosine similarity

of a 2-dimensional vector can be computed as

$$\text{COSSIM}(L_r, L_w) = \frac{L_r \cdot L_w}{\|L_r\| \|L_w\|} = \frac{\sum\limits_{i=0}^{n} L_{r_i} L_{w_i}}{\sqrt{\sum\limits_{i=0}^{n} L_{r_i}^2} \sqrt{\sum\limits_{i=0}^{n} L_{w_i}^2}} \quad (1)$$

where the result is a value between 0 and 1. Being 1 the ideal case where both vectors are the same.

b) Mean Squared Error: The Mean Squared Error (MSE) was used to assess the quality of the laser line coordinates vector during the welding stage L_w with respect to the laser line coordinates vector used as a reference L_r. The expression can be written as

$$\text{MSE}(L_r, L_w) = \frac{1}{n} \sum_{i=1}^{n} (|L_{r_i} - L_{w_i}|)^2 \quad (2)$$

c) Multiscale Structural Similarity: The Structural Similarity (SSIM) [22] is a similarity metric that is based on a perceptual model, and it is used to measure the similarity between two images, not only based on the intensity of the image, but based on the interdependence of its structure also. The SSIM can be computed as

$$\text{SSIM}(I_r, I_w) = \frac{(2\mu_r\mu_w + C_1)(2\sigma_{rw} + C_2)}{(\mu_r^2 + \mu_w^2 + C_1)(\sigma_r^2 + \sigma_w^2 + C_2)} \quad (3)$$

where μ_r, μ_w, and σ_r^2, σ_w^2, are the means, and variances of I_r, and I_r respectively. Meanwhile, σ_{rw} is the covariance between I_r and I_w. Furthermore $C_1 = (K_1 L)^2$ and $C_1 = (K_2 L)^2$ are small constants, where L is the dynamic range of the pixel values ($L = 255$ for 8 bits precision), and $K_1 \ll 1$ and $K_2 \ll 1$ are two scalar constants. In this work, the SSIM is computed at different scales, and then the results combined, as it was presented in [23], to conform a Multiscale SSIM (MS-SSIM).

d) Laser line extraction methods: The use of different methods to extract the position of the L_r, and L_w laser line coordinates vectors from the I_r, and I_w laser images may lead to a different result. As it is not the main focus of this paper, it will be just mentioned that the two laser line extraction methods considered in this work are based either on the Center Of Gravity (COG) [24], or the FIR-Peak (FP) [25] methods.

IV. EXPERIMENTAL IMPLEMENTATION

The experimental setup, as shown in Fig 3, was composed of an industrial welding robot, its welding equipment, and a laser line scanner attached next to the end-effector of the industrial welding robot employing a bracket. The industrial welding robot used was a Yaskawa Motoman GP25-12 with position control over Ethernet, and the welding equipment was a Fronius WF60I Robacta Drive CMT. Meanwhile, the laser line scanner was composed of a laser line projector model Z-LASER Z25M18S3-F-640-LP45 operating in the red wavelength and a polarization image sensor implemented in the industrial camera model MATRIX VISION mvBlueFOX3-2051pC. For more details about the hardware and software architecture of the laser line scanner used, the reader is referred to [18].
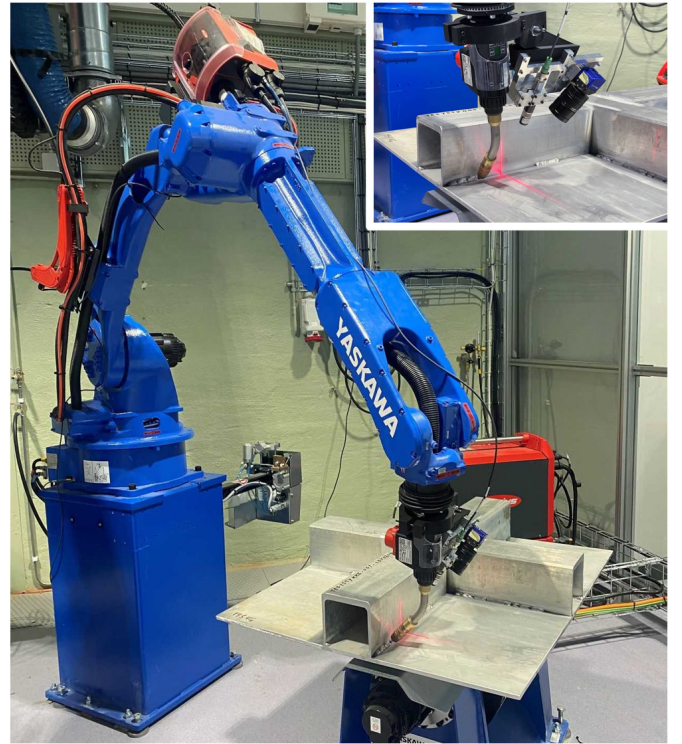

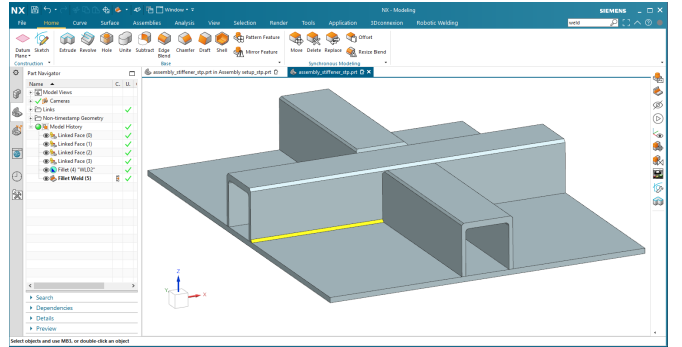Fig. 3: The setup used in the experiments. Welding robot with a laser profile scanner.

Fig. 4: CAD model of the test part with weld defined (shown in yellow).

A. Experimental software architecture

The software architecture used in the experiments is composed of two well-differentiated execution modes, each of them with different states and tasks, as can be seen in Fig. 6. When the system initializes, it enters the first mode, called Teach mode in Fig. 6a. During the Teach mode, the CAD machine is the publisher of the robot communication network, whereas the SEN machine is the subscriber and awaits the orders from the CAD machine. Once the Teach mode has started, the CAD1 task is executed, moving the welding robot to the CAD model welding start position state. As the position of where the welding should start in the CAD model is some millimetres away from the real-world coordinates, a
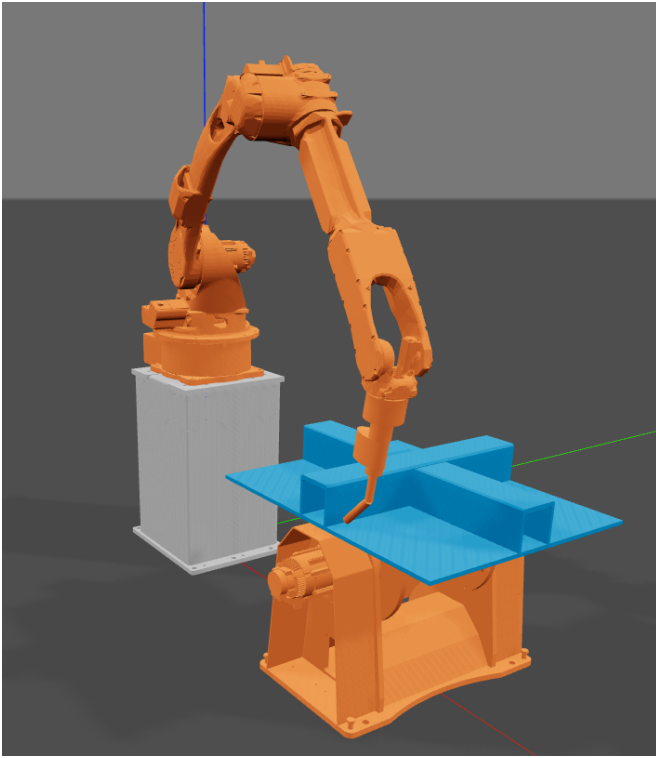
Fig. 5: Robotics toolbox used as offline programming environment with part and weld path loaded (visualized with Swift).

fine adjustment is done manually using the welding robot teach pendant during the HUMAN1 task to correct this small deviation. After this fine manual adjustment, the system is considered to be in the real-world position for starting to weld. The CAD machine then updates its welding-start position (task CAD2) and publishes a message to the network, in which the SEN machine is the subscriber, to create the welding-start trigger (task SEN2). In this state, the welding-start trigger is considered taught. Then, a similar process is followed again during the CAD3, HUMAN2, CAD4, and SEN2 task sequence to create the welding-end trigger. After the Teach mode is finished, and the welding-start and welding-end triggers are taught, the system enters the Weld mode.

During the Weld mode in Fig. 6b, human intervention is only needed to start the process, once it is started, the rest of the tasks are fully automated. This is in contrast with the Teach mode, in which human intervention was needed in two of the tasks to adjust the positions. In the Weld mode, and contrary to the Teach mode, the SEN machine becomes the Publisher of the network, and the CAD machine, the subscriber. The reason for this exchange of roles is that the CAD machine has to wait for the message coming from the SEN system with the sensor feedback of when it should continue with the next task. When the Weld mode is started, the SEN machine starts evaluating the trigger for the welding-start (task SEN3) and publishes a message to tell the CAD machine to start moving the robot to its CAD-model welding-start position (task CAD5). Once the welding-start trigger evaluation is finished, and the

welding-start similarity is found, the SEN machine stops the trigger evaluation (task SEN4), and publishes a message to tell the CAD machine to stop the robot, and start the welding torch (task CAD6). Then, the SEN machine starts evaluating the trigger for the welding-end (task SEN5) and publishes a message to tell the robot to start moving the robot to the CAD-model welding-end position (task CAD7). Once the welding-end trigger evaluation is finished, and the welding-start similarity is found, the SEN machine stops the trigger evaluation (task SEN6), and publishes a message to tell the CAD machine to stop the robot, and stop the welding torch (task CAD8). In this state, the welding is considered finished, and the system can go back to the Teach mode.

This process of welding-start, and welding-end trigger creation, and evaluation strategy can be performed again on every desired weld seam in the aluminium part, although for the presented experiments it was done for one weld seam only.

B. Sensitivity characterization of the trigger method

To characterize the sensitivity of the trigger metrics, a series of discretized weld gun positions were examined based on common torch angles and translation shifts experienced during welding. Table I summarizes the parameters along with the respective boundaries and step sizes used. The work angle R_w, and stick angle R_s can be seen depicted in Fig. 7a and 7b respectively. The tool direction represents translational movement along the weld gun vector. The transverse movements are decomposed into two perpendicular directions, X, and Z, relative to the weld direction. Each parameter was explored in a single degree of freedom, totalling 47 positions.

TABLE I: Testing parameters

Parameter	Minimum	Maximum	Discretization Step
Work angle, R_w	$-5°$	$5°$	$0.5°, 1°(R_w > \pm 2)$
Stick angle, R_s	$-5°$	$5°$	$0.5°, 1°(R_s > \pm 2)$
Tool direction, T_z	-5 mm	11 mm	2 mm
Transverse direction, T_x	1 mm	9 mm	2 mm

V. RESULTS

The proposed trigger metrics elaborated in Section III-C, Cosine Similarity (COSSIM), Multiscale Structural Similarity (MS-SSIM), and Mean Square Error (MSE) with laser line extraction methods FIR-Peak (FP), and Center Of Gravity (COG) are graphed in Fig. 8. The data points were collected based on the weld gun configurations derived from the parameters described in Table I.

VI. DISCUSSION

The COSSIM, MS-SSIM and MSE trigger metrics tests were carried out on the setup depicted in Fig. 3 based on initial positions generated from CAD following the procedure depicted in Fig. 6, and finally graphed in Fig. 8. Based on the weld gun displacements explored, all the proposed metrics seem capable of determining whether a trigger position is reached given that a viable threshold value is set. On
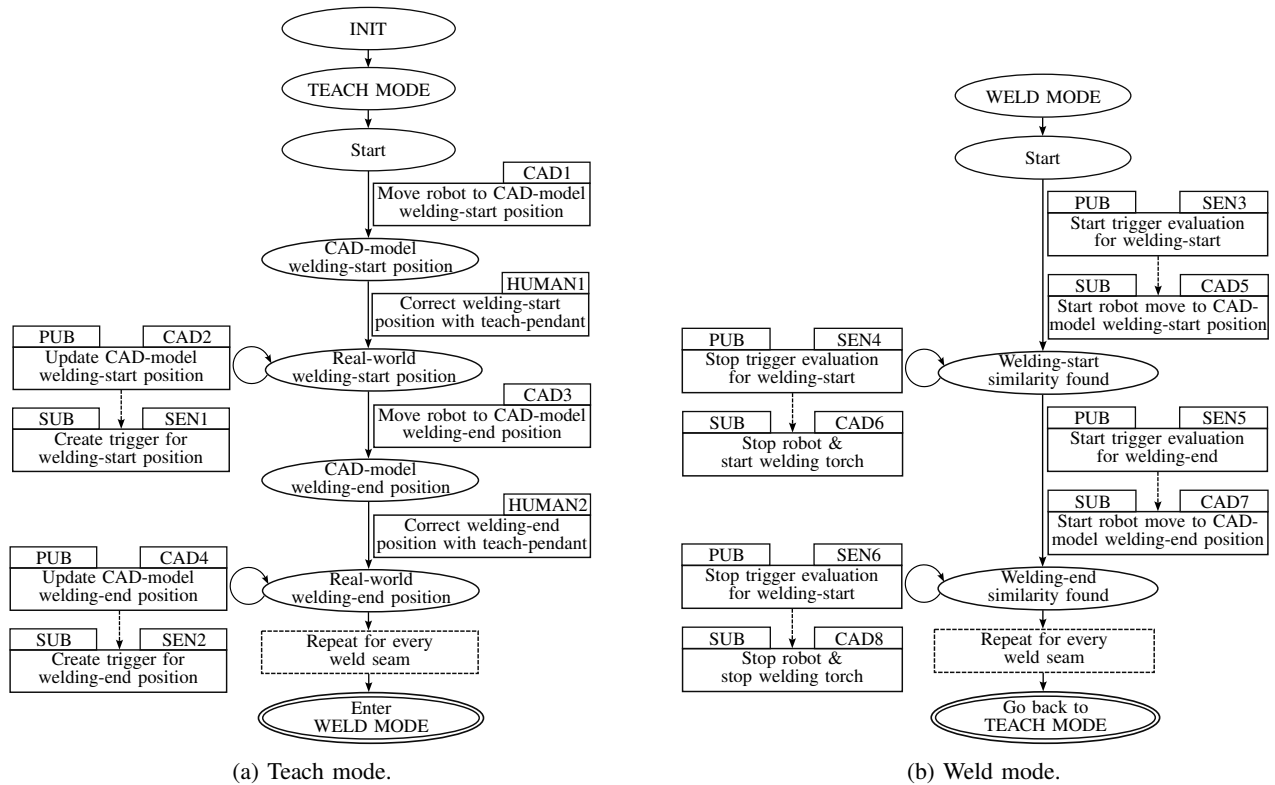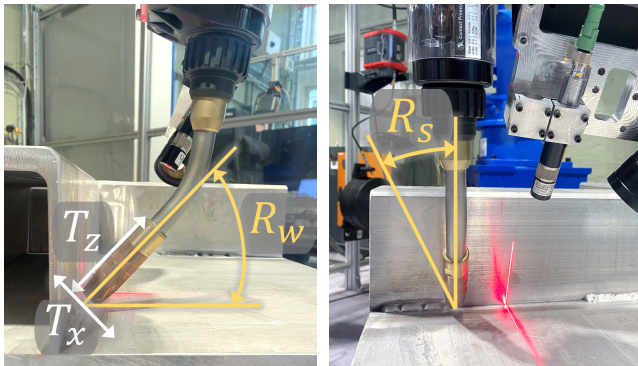
(a) Teach mode.

(b) Weld mode.

Fig. 6: System states and tasks for each mode of the software architecture used in the experiments.

(a) Work angle R_w, tool direction T_z and transverse direction T_x.

(b) Stick/drag angle R_s.

Fig. 7: Welding torch angles, and directions.

initial inspection, MS-SSIM seems to perform better than its counterparts, with the output value decreasing corresponding to displacements of the weld gun. However, it is important to note that as the underlying mechanism of the MS-MSSIM is based on image pixel analysis, exposure to arc welding light and welding sparks would create random noise in the image, disrupting the output value of the MS-SSIM. So, it can be considered a valid metric to trigger the welding-start, but the COSSIM and MSE metrics (which use the laser line coordinates, instead of the unprocessed image) should be

prioritized to trigger the welding-end, as they are more robust to random changes in the image.

Regarding the threshold values that could potentially be used in a robotic welding system, and based on the experiments in Fig. 8. We consider that a viable trigger threshold for the vector-based similarity metric COSSIM could be set to 0.95 for both laser line extraction methods. Whereas, a viable trigger threshold for the image-based similarity metric MS-SSIM, could be set to 0.90, as the experiments show this is a more sensitive metric. Finally, for the MSE, which is based on measuring the error, we consider a viable threshold to be 2×10^4 approximately.

In its current state, the manual steps, HUMAN1, and HUMAN2, described in Fig. 6 were intended primarily for testing purposes. However, to achieve a fully automatic system, simulations of projected laser lines from CAD could in theory be used to bypass these manual steps, if combined with other calibration methods. Works such as Abu et al. [26], and Mohammadikaji et al. [27], have previously incorporated such techniques successfully for optical measurement purposes. Examination of this is left as subject to further work. The robot trajectories were pre-calculated prior to movement (CAD1) and the manually corrected (HUMAN1) to characterize the trigger sensitivities. Hence, real-time motion was not implemented for the given scope. For a full system, an extended solution with real-time seam-tracking should be implemented as a movement policy after weld positions have been identified. Furthermore, to overcome the computational

(a) Work angle R_w deviation (deg) Similarities.

(b) Work angle R_w deviation (deg) Mean Squared Errors.

(c) Stick/drag angle R_s deviation (deg) Similarities.

(d) Stick/drag angle R_s deviation (deg) Mean Squared Errors.

(e) Stick translation T_x deviation (mm) Similarities.

(f) Stick translation T_x deviation (mm) Mean Squared Errors.

(g) Tool translation T_z deviation (mm) Similarities.

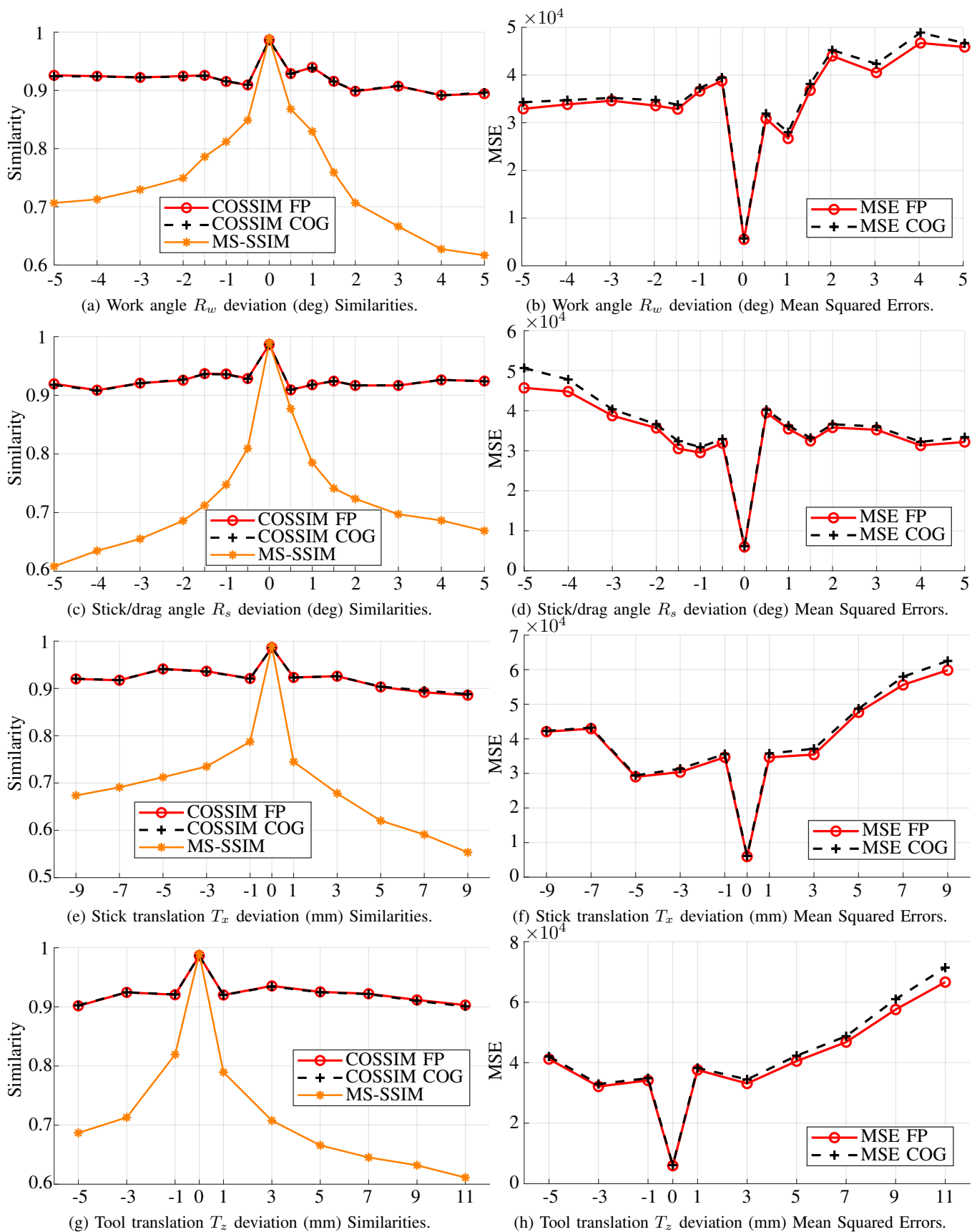(h) Tool translation T_z deviation (mm) Mean Squared Errors.

Fig. 8: Similarities and errors of the two angles, and two translations tested.

costs, the performance of CAD-to-Robot can be improved by implementing GPU for certain operations [28].

VII. CONCLUSION

A trigger method has been proposed to create and evaluate robotic welding triggers (i.e. for starting or stopping the robotic welding) with laser line scanners to verify real-world welding positions to those generated in CAD. The architecture of a laser line scanner makes it possible to calculate the similarity between two laser lines using either the raw image without further processing or performing a laser line extraction on the images and then calculating the similarity between the resulting vectors. The decision to compute either an image-based or vector-based similarity makes choosing the right metric for each case a necessity to trigger the robotic welding. The COSSIM, MS-SSIM and MSE metrics were used to identify viable methods for weld gun location and pose triggering for laser line projection setups. The considered metrics were found viable, meaning trigger threshold values could be defined to determine whether a correct weld gun pose was achieved. While MS-SSIM was shown in general to have more consistent output values with the displacement of the weld gun, its underlying mechanism is subject to potential noise for reasons elaborated in Section VI.

Potential trigger threshold values have been proposed by the authors for all of the metrics. Values of 0.95, 0.90, and 2×10^4 were found to be viable within the frame of the experiments for COSSIM, MS-SSIM, and MSE, respectively. Finally, a procedure for creating and evaluating robotic welding triggers was described in Section IV-A and depicted in Fig. 6.

REFERENCES

[1] C. Zheng, Y. An, Z. Wang, H. Wu, X. Qin, B. Eynard, and Y. Zhang, "Hybrid offline programming method for robotic welding systems," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102238, 2022.

[2] R. Manorathna, P. Phairatt, P. Ogun, T. Widjanarko, M. Chamberlain, L. Justham, S. Marimuthu, and M. R. Jackson, "Feature extraction and tracking of a weld joint for adaptive robotic welding," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014, pp. 1368–1372.

[3] M. Miller, B. Mi, A. Kita, and I. C. Ume, "Development of automated real-time data acquisition system for robotic weld quality monitoring," *Mechatronics*, vol. 12, no. 9-10, pp. 1259–1269, 2002.

[4] M. Fridenfalk and G. Bolmsjö, "Design and validation of a universal 6d seam-tracking system in robotic welding using arc sensing," *Advanced Robotics*, vol. 18, no. 1, pp. 1–21, 2004.

[5] Y. He, Z. Yu, J. Li, G. Ma, and Y. Xu, "Fault correction of algorithm implementation for intelligentized robotic multipass welding process based on finite state machines," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 28–35, 2019.

[6] J. N. Pires, A. Loureiro, and G. Bölmsjo, *Welding robots: technology, system issues and application*. Springer Science & Business Media, 2006.

[7] W. Gu, Z. Xiong, and W. Wan, "Autonomous seam acquisition and tracking system for multi-pass welding based on vision sensor," *The international journal of advanced manufacturing technology*, vol. 69, no. 1, pp. 451–460, 2013.

[8] P. Xu, G. Xu, X. Tang, and S. Yao, "A visual seam tracking system for robotic arc welding," *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 1, pp. 70–75, 2008.

[9] E. B. Njaastad and O. Egeland, "Automatic touch-up of welding paths using 3d vision," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 73–78, 2016.

[10] J. Muhammad, H. Altun, and E. Abo-Serie, "Welding seam profiling techniques based on active vision sensing for intelligent robotic welding," *The International Journal of Advanced Manufacturing Technology*, vol. 88, no. 1, pp. 127–145, 2017.

[11] X. Lü, D. Gu, Y. Wang, Y. Qu, C. Qin, and F. Huang, "Feature extraction of welding seam image based on laser vision," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4715–4724, 2018.

[12] N. Wang, K. Zhong, X. Shi, and X. Zhang, "A robust weld seam recognition method under heavy noise based on structured-light vision," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101821, 2020.

[13] Z. Liu, W. Bu, and J. Tan, "Motion navigation for arc welding robots based on feature mapping in a simulation environment," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 137–144, 2010.

[14] X. Chen and S.-B. Chen, "The autonomous detection and guiding of start welding position for arc welding robot," *Industrial Robot: An International Journal*, 2010.

[15] Y. Ding, W. Huang, and R. Kovacevic, "An on-line shape-matching weld seam tracking system," *Robotics and Computer-Integrated Manufacturing*, vol. 42, pp. 103–112, 2016.

[16] A. Lobov and T. A. Tran, "Object-oriented approach to product design using extended NX Open API," *Procedia Manufacturing*, vol. 51, pp. 1014–1020, 2020.

[17] P. Corke and J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 357–11 363.

[18] J. Marco-Rider, L. Tingelstad, and O. Egeland, "Polarization image sensor-based laser scanner for reflective metals: architecture and implementation," in *2021 IEEE Sensors*. IEEE, 2021, pp. 1–4.

[19] P. Hintjens, *ZeroMQ: messaging for many applications*. "O'Reilly Media, Inc.", 2013.

[20] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through fsim, ssim, mse and psnr—a comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.

[21] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[22] W. Zhou, "Image quality assessment: from error measurement to structural similarity," *IEEE Trans Image Process*, vol. 13, pp. 600–613, 2004.

[23] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[24] Z. Gan and Q. Tang, *Visual sensing and its applications: integration of laser sensors to industrial robots*. Springer, 2011.

[25] J. Forest, J. Salvi, E. Cabruja, and C. Pous, "Laser stripe peak detector for 3d scanners. a fir filter approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3. IEEE, 2004, pp. 646–649.

[26] B. A. Abu-Nabah, A. O. ElSoussi, and A. E. K. Al Alami, "Virtual laser vision sensor environment assessment for surface profiling applications," *Measurement*, vol. 113, pp. 148–160, 2018.

[27] M. Mohammadikaji, *Simulation-based planning of machine vision inspection systems with an application to laser triangulation*. KIT Scientific Publishing, 2020, vol. 17.

[28] T. A. Tran and A. Lobov, "Enhancing CAD-integrated automatic feature recognition of weld joints with GPU-accelerated multi-directional slicing (accepted)," in *2022 23nd IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2022.