Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/22735

To cite this version :

Anthony SURLERAUX, Romain LEPERT, Pierre KERFRIDEN, Samuel BIGOT, Jean-Philippe PERNOT - Machine Learning-Based Reverse Modeling Approach for Rapid Tool Shape Optimization in Die-Sinking Micro Electro Discharge Machining - Journal of Computing and Information Science in Engineering - Vol. 20, n°3, p.11p. - 2020

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Arts et Métiers Institute of Technology

Machine Learning-Based Reverse Modeling Approach for Rapid Tool Shape Optimization in Die-Sinking Micro Electro Discharge Machining

Anthony Surleraux
Cardiff School of Engineering,
Cardiff University,
Cardiff CF24 3AA, UK
e-mail: anthony.surleraux@gadz.org

Romain Lepert
Arts et Métiers,
LISPEN EA 7515, HeSam,
Aix-en-Provence 13617, France
e-mail: romain.lepert@gadz.org

Jean-Philippe Pernot
Arts et Métiers,
LISPEN EA 7515, HeSam,
Aix-en-Provence 13100, France
e-mail: jean-philippe.pernot@ensam.eu

Pierre Kerfriden
Cardiff School of Engineering,
Cardiff University,
Cardiff CF24 3AA, UK;
Centre des Matériaux, Mines ParisTech,
PSL University,
Évry cedex BP87 F-91003, France
e-mail: pierre.kerfriden@mines-paristech.fr

Samuel Bigot[1]
Cardiff School of Engineering,
Cardiff University,
Cardiff CF24 3AA, UK
e-mail: BigotS@cardiff.ac.uk

This paper focuses on efficient computational optimization algorithms for the generation of micro electro discharge machining (μEDM) tool shapes. In a previous paper, the authors presented a reliable reverse modeling approach to perform such tasks based on a crater-by-crater simulation model and an outer optimization loop. Two-dimensional results were obtained but 3D tool shapes proved difficult to generate due to the high numerical cost of the simulation strategy. In this paper, a new reduced modeling optimization framework is proposed, whereby the computational optimizer is replaced by an inexpensive surrogate that is trained by examples. More precisely, an artificial neural network (ANN) is trained using a small number of full reverse simulations and subsequently used to directly generate optimal tool shapes, given the geometry of the desired workpiece cavity. In order to train the ANN efficiently, a method of data augmentation is developed, whereby multiple features from fully simulated EDM cavities are used as separate instances. The performances of two ANN are evaluated, one trained without modification of process parameters (gap size and crater shape) and the second trained with a range of process parameter instances. It is shown that in both cases, the ANN can produce unseen tool shape geometries with less than 6% deviation compared to the full computational optimization process and at virtually no cost. Our results demonstrate that optimized tool shapes can be generated almost instantaneously, opening the door to the rapid virtual design and manufacturability assessment of μEDM die-sinking operations. [DOI: 10.1115/1.4045956]

Keywords: die-sinking micro-EDM, tool shape optimization, process simulation, reduced modeling, artificial neural network, machine learning, artificial intelligence, computational geometry, computer-aided manufacturing, engineering informatics, machine learning for engineering applications, multiscale modeling and simulation

1 Introduction

1.1 Background. Micro and nano manufacturing technologies are widely recognized as promising sources of innovation for the decades to come [1]. However, manufacturing outputs at small scales become increasingly sensitive to variations in manufacturing strategies and machine parameters. Due to multiple interactions between phenomena such as large tool/part deformations, inconsistent material removal rates (MRRs), and spurious process instabilities at smaller scales, the development of micro and nano technologies is largely impaired by today's lack of predictive and control capabilities for the quality of manufacturing outputs. The solution is, of course, to develop predictive computer models for these technologies [2]. Such models would enable engineers to design micro/nano manufacturing virtually, thereby facilitating manufacturability studies and reducing the need for physical trial-and-error calibration procedures. Additionally, simulators could be used to automatically optimize manufacturing strategies and help control processes on-the-fly, using virtual twinning approaches.

The study presented in this paper is relevant to this category of developments. The digital optimization of micro/nano thermal ablation processes such as laser milling or micro electro discharge machining (μEDM) is of particular interest. Modeling strategies in this area fall in several complementary categories. Most studies in this area attempt to represent individual discharges of energy using detailed mathematical models of the physics, enabling the prediction of MRRs as a function of machining parameters [3–5]. Alternatively, such physics-based models may be replaced by purely data-driven statistical regressors [6], which require more data and risk a certain lack of generalization but circumvent the need for deriving a physical model and devising an appropriate calibration strategy. In turn, MRR models may be used as inputs for the automatic design of manufacturing strategies using layer-by-layer toolpath generation algorithms, the material removal rate being required to characterize the parameters of an "equivalent mechanical mill." Alternatively, one may wish to make use of MRR models to perform crater-by-crater simulations [7–9]. Although the resulting transient simulators are numerically expensive, they are expected to predict the effect of sharp, localized features more accurately than models employing a spatially homogeneous removal rate. Let us also notice that very detailed multiphysics modeling strategies are currently being employed to *better understand* the physics involved in precision thermal manufacturing [10]. However, the inherent complexity of such models (larger number of parameters) makes them less suitable to blind predictions in today's operational environment.

Goal: The present study follows a previous contribution from the authors on crater-by-crater simulations of μEDM die-sinking processes, which will be fully detailed further below. As mentioned

[1]Corresponding author.

Manuscript received July 1, 2019; final manuscript received November 27, 2019; published online January 11, 2020. Assoc. Editor: Bin He.

in the previous paragraph, such time-dependent simulations are extremely time-consuming, making them unusable in practical computer-aided manufacturing. Following modern approaches in model order reduction [11], meta-modeling [12–14], and using appropriate elements of statistical learning, the aim is to create an efficient and goal-oriented model compression algorithm that will be capable, based on a reasonably low amount of full-scale μEDM simulations, to produce new simulation results at a fraction of the cost.

1.2 Wear Challenge in μEDM Die Sinking. Electrical discharge machining (EDM) is a manufacturing process where material removal is due to successive electrical discharges. Its main interest lies in the possibility of machining any conductive material regardless of its hardness. Without discussing the details of specific variants of EDM, the principle of this technology is as follows. The tool and the workpiece (or electrodes) are immerged in a dielectric fluid and submitted to an electrical current. The gap between them is reduced until the dielectric reaches its breakdown voltage. The current is then free to flow from one electrode to the other, creating a plasma channel. In the process, part of the tool and workpiece evaporate in the region where the electric discharge happens, leading to the formation of craters.

μEDM is the application of EDM to manufacturing operations involving micrometric dimensions. In this context, tool wear is of importance: as the process goes on, material is removed from the tool, which modified its shape. While tool wear of manageable consequences in classic EDM, its influence is most notable in its micro counterpart as shown in μEDM drilling [15] (Fig. 1) and results in unavoidable changes in electrode shape and dimensional errors.

Indeed, while tried-and-tests methods exist for μEDM milling to compensate for the tool wear, this is not easily achieved in die-sinking μEDM. Compensation method have been considered, such as the use of a self-repair method [16] which can reduce the machining error caused by the bottom and corner wear of the electrode through a carefully controlled tool motion that attempt to influence tool wear location. But such approach is limited to simple shape electrode rather that true 3D shapes. Thus, generally die-sinking μEDM would often require the use of multiple tools to obtain the desired geometries within specific tolerances, for instance to achieve sharp corners at micro scale.

1.3 μEDM Simulation and Tool Optimization. The ability to predict the location and intensity of wear in die-sinking μEDM would enable the design of optimal tools able to compensate for the upcoming wear and reduce the number of electrodes required for a successful machining.

The optimization of tool electrode to compensate for tool wear is used regularly in drilling and in μEDM milling. For instance, when producing blind holes using μEDM drilling without wear compensation, because of tool wear, the real depth of the hole will be significantly smaller [17]. In this case, a method to achieve a specific depth is to compensate for wear by constant electrode feeding in the z-axis [18], which is equivalent to extending the working length of the tool using the expected wear ratio.

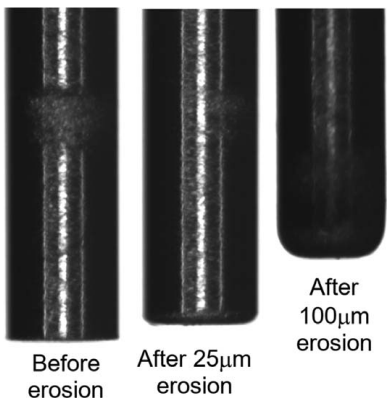

Fig. 1 Electrode shape changes on a Ø 150 μm electrode after two erosion depths

In μEDM milling, the use of simple shaped electrode (generally cylindrical) makes the wear prediction relatively straightforward [19]. It uses specific layer-by-layer machining strategies, such as the uniform wear method [20], which ensures that the machining occurs only on the bottom surface of the electrode and results in a relatively linear wear on the z-axis. Therefore, only feeding the electrode in z-axis can do the wear compensation, which is again equivalent to extending the working length of the tool.

In the case of die-sinking μEDM, where more complex shapes are used as electrodes, the compensation is not as straightforward because the way electrode geometries are affected by the wear is complex.

This was addressed by developing a crater-by-crater simulator in Ref. [7] where the final shape of tool electrode was predicted in one step using a "reverse simulation" technique. In this simulation, the removal of volume on both workpiece and electrode is achieved using voxels' representations and appears to be performed per specific time periods on large areas rather than discharge by discharge. In addition, the machining gap is not taken into account and it is not clear if this method can be applied to micro-EDM, in particular due to resolution that can be applied to the voxels' representation used.

This was achieved by the authors with the production of an efficient simulation method [21] using voxels in an octree data structure, which allows for high resolution μEDM simulation. Later, this simulation method was used as a building block for the development of a tool electrode shape optimization framework [22] which uses an iterative loop method to create an optimum tool shape from a given target workpiece cavity. This method is applicable to both 2D and 3D simulation and was successfully validated for 2D application but proved highly computationally expensive for the optimization of 3D shapes, making it not directly scalable to industrial problems where an end-user would aim at efficiently designing optimal tools in 3D as they would require access to high-performance computing. In this paper, a new optimization framework is proposed based on the use of an artificial neural network (ANN), trained to generate directly an optimal tool shape from a given targeted workpiece cavity shape and thus replacing the costly iterative optimization method.

The paper is organized as follows: Sec. 2 introduces the voxel-based die-sinking μEDM simulation framework and the tool shape optimization framework. Both are used to generate the instances used to define and tune an ANN presented in Sec. 3. Section 4 presents more results while considering the machining parameters as additional attributes. It also compares the classical tool shape optimization and the advanced machine learning-based tool shape optimization proposed in this paper. Finally, the last section ends this paper and provides conclusion and future works.

2 Die-Sinking Micro Electro Discharge Machining Simulation Framework and Tool Shape Reverse Modeling

This section introduces the voxel-based die-sinking μEDM simulation framework (Sec. 2.1) and the tool shape optimization framework (Sec. 2.2). The optimization algorithm makes use of the simulation tool to iteratively reverse model the tool shape in order to anticipate and compensate for the wear phenomenon. Advantages and limitations are stressed and the opportunity of using a machine learning approach to speed-up the tool shape reverse modeling is first discussed in Sec. 2.3.

2.1 Voxel-Based Die-Sinking μEDM Simulation Framework. As it plays a central role within the optimization framework, the basics of the previously developed die-sinking

μEDM simulator are briefly recalled in this section [23]. This algorithm simulates and keeps track of the progress of wear on the tool and workpiece electrodes using a microscale geometrical modeling approach.

Given the initial geometric models of the tool and workpiece at the start of the process, the die-sinking is simulated by altering the geometry of the electrodes during every one of a series of simulated discharges. This is done by introducing, for each discharge, conjugate pairs of craters, modeled geometrically by spheres, at the location where the distance between the electrodes is minimum, which is where the appearance of an electric spark is physically the most likely.

An overview of the overall μEDM simulation algorithm is given in Fig. 2. An individual discharge is simulated as follows. First, the minimum distance d between the tool and the workpiece is computed, and then:

- If this distance is larger than the machining gap (i.e., the minimum between the workpiece and the tool required for electric discharges to take place, which is a parameter of the die-sinking manufacturing process), the tool electrode is moved downwards by a small increment.
- If this distance is smaller than the machining gap, pairs of conjugate craters are crated on both the tool and the workpiece, at the location of the minimum distance. Two additional parameters of the model are required for this operation, namely, the workpiece crater volume and the tool crater volume, which are, respectively, the volume of "material" removed from the workpiece and the tool during each discharge.

The algorithm is stopped once the targeted depth of the tool electrode is reached. Throughout the simulations, all geometric data are represented by means of voxels embedded in a voxel octree. The octree is a tree structure in which each node can have up to eight children. A node without any children is called a leaf node, while nodes are called roots. In this way, the modeling is purely volumetric and the computation times are significantly reduced, while octree structures allow to control the amount of memory required to represent the electrodes at the scale of individual craters.

The minimum distance search exploits the hierarchical data structure of the octree tree. The algorithm starts with the root nodes of each electrode and computes the minimum distance between the possible couples of their children:

- If both children are leaves, the minimum distance between them can be immediately found.
- If not, the minimum distance can be bounded between a maximum $d_{\max}$ and a minimum value $d_{\min}$.

All the couples of nodes for which $d_{\min} \leq d_{\text{smallest Max}}$ (where $d_{\text{smallest Max}}$ is the smallest of all the $d_{\max}$) are kept. All the remaining couples' children are then compared in an identical manner until only couples of leaves remain. A fast exist condition is used in the case where a couple of nodes has an upper bound $d_{\max}$ that is smaller than the machining gap.

The crater insertion algorithm also uses the hierarchical data structure of the octree structure. At the relevant location, the intersections between a sphere and the root node's children are computed and the following rules are applied:

- if a child node does not intersect, nothing happens,
- if a node is completely inside the sphere, it is deleted,
- nodes on the boundary of the sphere are kept for the next iterations.

The algorithm is then reapplied on the new candidate nodes until a certain resolution is met. Figure 3 shows a cross-sectional view of a crater and the result of a full simulation on a workpiece can be seen in Fig. 4.

2.2 Optimization Framework for Tool Shape Reverse Modeling. This section introduces the basics of the previously developed optimization framework that uses the μEDM simulator to predict the tool shape that would be required for a die-sinking process to produce a given target workpiece [21]. The algorithm proposed to achieve this goal is so far, purely computer based. It could be applied to a particular *in situ* die-sinking operation once the parameters of the simulator (i.e., crater volumes and machining gap) are calibrated from data, which is not covered in this paper.

It should be mentioned that although the simulation tool presented in the previous section deals with three-dimensional simulations to demonstrate and validate the optimization concept more efficiently and rapidly, it was decided to perform optimizations using the simulation with two-dimensional profiles. However, since all the algorithms have been developed taking this fact into account, the method can easily be adapted for use with three-dimensional shapes. This is further discussed in Sec. 2.3.

The simulator described in Sec. 2.1 can be seen as an implicit, complex function that, for given initial geometries of the tool and workpiece electrodes and for fixed parameters of the μEDM wear simulator, associates the resulting geometry on the workpiece, together with, incidentally, the resulting geometry of the worn tool. When attempting to achieve a specific target cavity on the

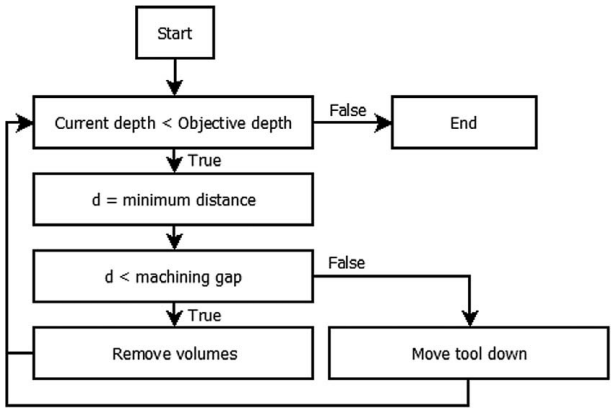

Fig. 2 Overall μEDM simulation framework

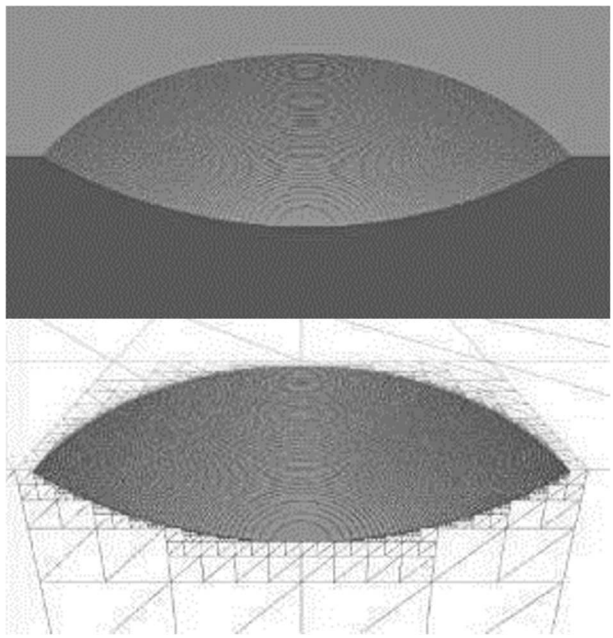

Fig. 3 Cross-sectional view of a crater at a resolution of one voxel per 125 nm

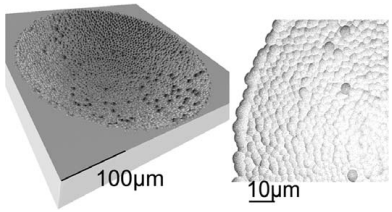

Fig. 4 Spherical feature obtained by simulation (left) and detail of the craters (right)

workpiece and therefore looking for the initial geometry of the tool that would produce this cavity, it is required to formally invert this complex function.

Such inversions are ill-posed in general, a first step of this new approach is to recast it as finding the solution of an optimization problem, whereby the shape of the workpiece delivered by the simulator is required to be as close as possible to its nominal target. Then, the solution of this optimization problem is found using a specifically developed predictor/corrector algorithm. The predictor step is a direct simulation of the die-sinking µEDM process, for a given iterate of the tool geometry, while the corrector step changes the shape of the tool, in an attempt to reduce the distance between the nominal target geometry of the cavity and its simulated counterpart. Intuitively, an excess of material removal on the workpiece indicates that material should be removed from the tool during the correction step, while insufficient material removal on the workpiece would require adding material to the tool. One iteration of the optimization algorithm is as follows (Fig. 5):

- *Error computation and convergence estimation*: The difference ε between the target workpiece W_{target} and the workpiece profile W_{result} that was delivered by the simulator during the previous iteration is computed. For the first iteration, W_{result} is the initial workpiece without any feature. Hence, the first difference ε represents the actual overall target area (or volume in 3D) to be removed.
  The accuracy of the optimization process, $A\%$, is measured using the following metric:

$$A\% = \frac{A_{W_{\text{target}}} - \varepsilon}{A_{W_{\text{target}}}} \times 100 \tag{1}$$

  where $A_{W_{\text{target}}}$ is the area (in μm^2) of the target workpiece feature and ε is the mismatch area (in μm^2) previously defined. In all examples shown in this paper, the optimization algorithm is stopped once $A\% \leq 0.5\%$.
- *Corrector*: Using the mismatch ε between simulated and targeted workpiece profiles (Fig. 6), a corrected tool profile T_c is computed. This is done by updating an intermediate object, the "virtual workpiece" profile W_{virtual}, which corresponds to the profile of the workpiece that would be generated if no tool wear occurred and if the effect of micro-craterization could be neglected (i.e., negative of the tool, taking into account the machining gap). Once the modification of W_{virtual} has been performed, a corrected iterate of the tool profile T_c is computed by applying the machining gap to the "virtual workpiece."
- *Predictor*: An EDM simulation using the current tool profile T is performed and generates the workpiece W_{result}; the accuracy of which will be evaluated at the beginning of the next iteration of the algorithm.

An illustrative example of the capabilities of the tool shape optimization process is given in Fig. 7. Elements (a) and (b) represent, respectively, the targeted workpiece profile W_{target} to be produced on the workpiece and the initial tool shape. The initial tool shape is chosen as the negative of the target profile, accounting for the expected machining gap. Element (c) is the optimized tool shape, while element (d) is the resulting workpiece. This result has been obtained after five iterations of the optimization algorithm, and it corresponds to a target error of $A\% = 98.2\%$ (deviation between (a) and (d)).

2.3 Machine Learning Opportunity. Section 2.2 has briefly recalled the basics of the tool shape optimization framework based on an iterative loop method. In addition to converging quite rapidly, the algorithm was found to be quite efficient achieving high accuracy values (in the range of 95%). However, this method was developed with the idea that it would be implemented in three dimensions. Thus, considering the length of a single 3D simulation, around 2 h for a 100 μm deep machining, a single optimization process that could take 4, 5, or more iterations could be potentially lengthy.

From this emerged the idea to train a neural network using the optimized profiles from the iterative optimization method in order to limit the number of loops to be done and instead use a predictive neural network model that would be able to output directly an optimal tool shape.

3 Machine Learning-Based Tool Shape Optimization

This section details the proposed machine learning-based tool shape optimization framework, which makes use of the previously introduced simulation and optimization tools to generate a large number of instances for training purpose. In this paper, to demonstrate the feasibility of this approach and to enable an efficient evaluation of its performance, these instances will be generated using 2D profiles. Once validated, the same approach will be tested in 3D as part of future development.

3.1 Introduction. The idea behind using ANNs to propose optimal tool shapes is to be able to train it once and with a specific set of optimized shapes/profiles and to make use of the resulting acquired model to estimate the optimal tool shape for a given unseen target workpiece. Figure 8 provides with the overview of the proposed framework.

The algorithm will try to maximize its performance P while accomplishing a task T, through learning from experience E. The features of the algorithm are as follows:

- Task (T): Predict the optimal virtual workpiece for a target profile.

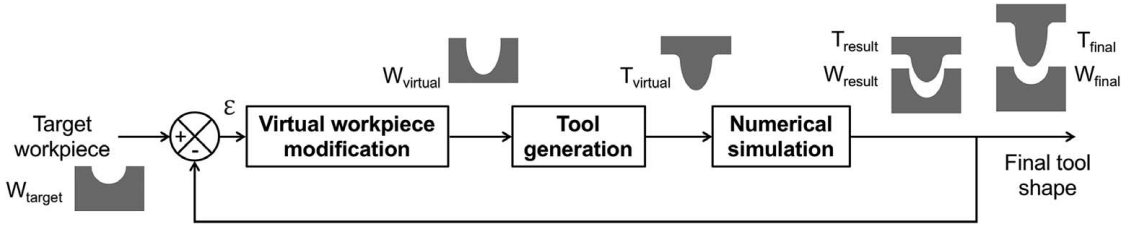

Fig. 5 Tool shape optimization process with iterative calls to the µEDM simulator

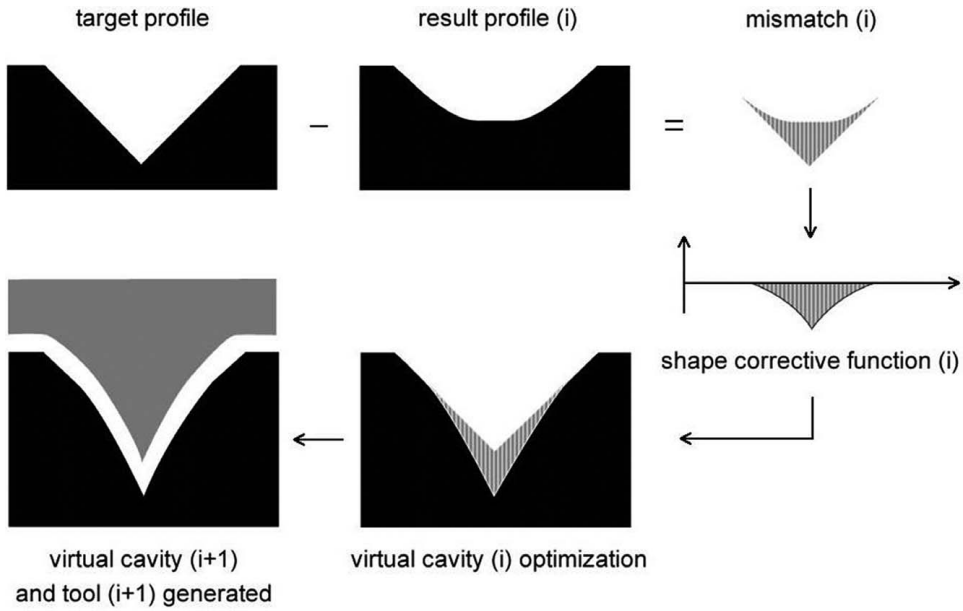

Fig. 6 A single iteration of the correction process

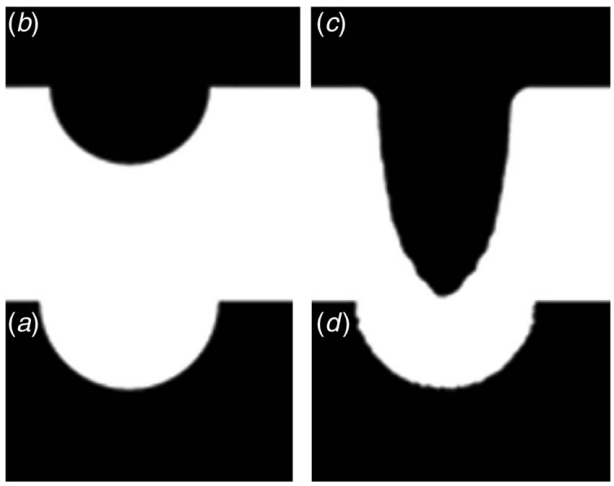

Fig. 7 Tool shape optimization capabilities. Left (top and bottom): the non-optimized tool and target profile. Right (top and bottom): the optimized tool and the cavity it produces.

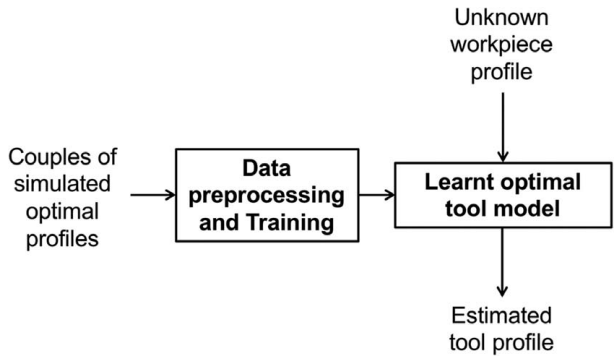

Fig. 8 Overview of the proposed framework

Table 1 The machining and simulation parameters considered as constant

Parameter	Value
Machining gap (μm)	10.0
Workpiece crater radius (μm)	3.00
Workpiece crater depth (μm)	2.25
Tool crater radius (μm)	2.25
Tool crater depth (μm)	1.50
Resolution (μm/voxel)	0.5

- Experience (E): A corpus of target profiles with their respective optimal virtual workpieces.
- Performance (P): Prediction accuracy, i.e., the relative difference (expressed as a percentage) between the predicted virtual workpiece and the actual virtual workpiece obtained through the iterative optimization.

The first approach is to only consider a specific set of machining parameters and focus on the prediction of optimal tool shapes in the case of a different target workpiece rather than different machining parameters. The reasoning behind it is to first verify that a machine learning method can be used in this simple test case before being extended to a more general environment that would include the machining gap and crater dimensions as input variables for the training of the model. The machining parameters are therefore constant, during the generation of the data for the initial machine learning model training, and their values are tabulated in Table 1. In Sec. 4, those parameters are not anymore considered as constant.

3.2 Multiplying the Training Data. Given the little training set instances available and the requirements when considering the use of a neural network, it is important to look at the problem differently and to redefine the instances the algorithm will train with.

Throughout the optimization process, modifications regarding virtual workpiece profiles are applied strictly vertically via the shape optimization function that is based on the mismatch of each iteration. Furthermore, the first mismatch considered is the area of the target profile. This means that the virtual workpiece profile is indirectly a modified version of the target profile. Each point of the target profile can be considered to be individually moved vertically to form the virtual workpiece profile. Given the point-to-point

correlation between the target profile and the virtual workpiece profile, it is possible to redefine the problem as follows: for a given point on the target profile, predict the vertical position of the respective point on the virtual workpiece profile. Therefore, the predicted optimal virtual workpiece profile is the combination of the predictions of all the points of the optimal workpiece profile from all the points of the target profile. Thus, the features of the estimation algorithm can be redefined as follows:

- Task (T): Predict the position on the optimal virtual workpiece for a point of a target profile.
- Experience (E): A corpus of points from target profiles with their respective counterpart on the optimal virtual workpiece.
- Performance (P): Prediction accuracy, i.e., the relative difference (expressed as a percentage) between the predicted virtual workpiece position and the actual virtual workpiece position.

An overview of the propose machine learning model is depicted in Fig. 9. It clearly shows the inputs and the output to be correlated. During the training phase, for each point of the target profile, the algorithm tries to correlate the underlying attributes to the vertical position of the respective point on the virtual workpiece profile. The adopted attributes are further detailed in Sec. 3.3. For sake of clarity, Fig. 9 does not detail all the attributes used as inputs of the learning model, but they can be found in Sec. 3.3.

Given the maximum sizes of the profiles and the resolution, each test profile will provide up to 600 points as data to work with. Thus, using 30 different profiles (Fig. 10) results in approximately 18,000 instances to distribute in the following sets:

- The training set (60%): 10,800 instances
- The validation set (20%): 3600 instances
- The test set (20%): 3600 instances

More precisely, the data points extracted from each of the fully simulated test cases are shuffled randomly and distributed to the training, validation, and test sets according to the proportions reported earlier.

3.3 Feature Selection. Considering that each point is now considered separately from the rest of the profile, every attribute will have as a purpose to give valuable information regarding the environment the given point is in, i.e., information regarding the whole target profile. Thus, the attributes can be grouped into three input categories and one output category as follows (see Fig. 9 where in some of those attributes are illustrated):

- Attributes providing information on the local context (i.e., the point and its surroundings):
  - yValue: Y-value of the point on the target workpiece profile (in μm).
  - parametricDescription: Three of the four parameters describing the third degree Lagrange interpolating polynomial [24] that approximates the neighborhood of the given point with a window width equal to two machining gaps (Fig. 9, in blue). The constant parameter is excluded

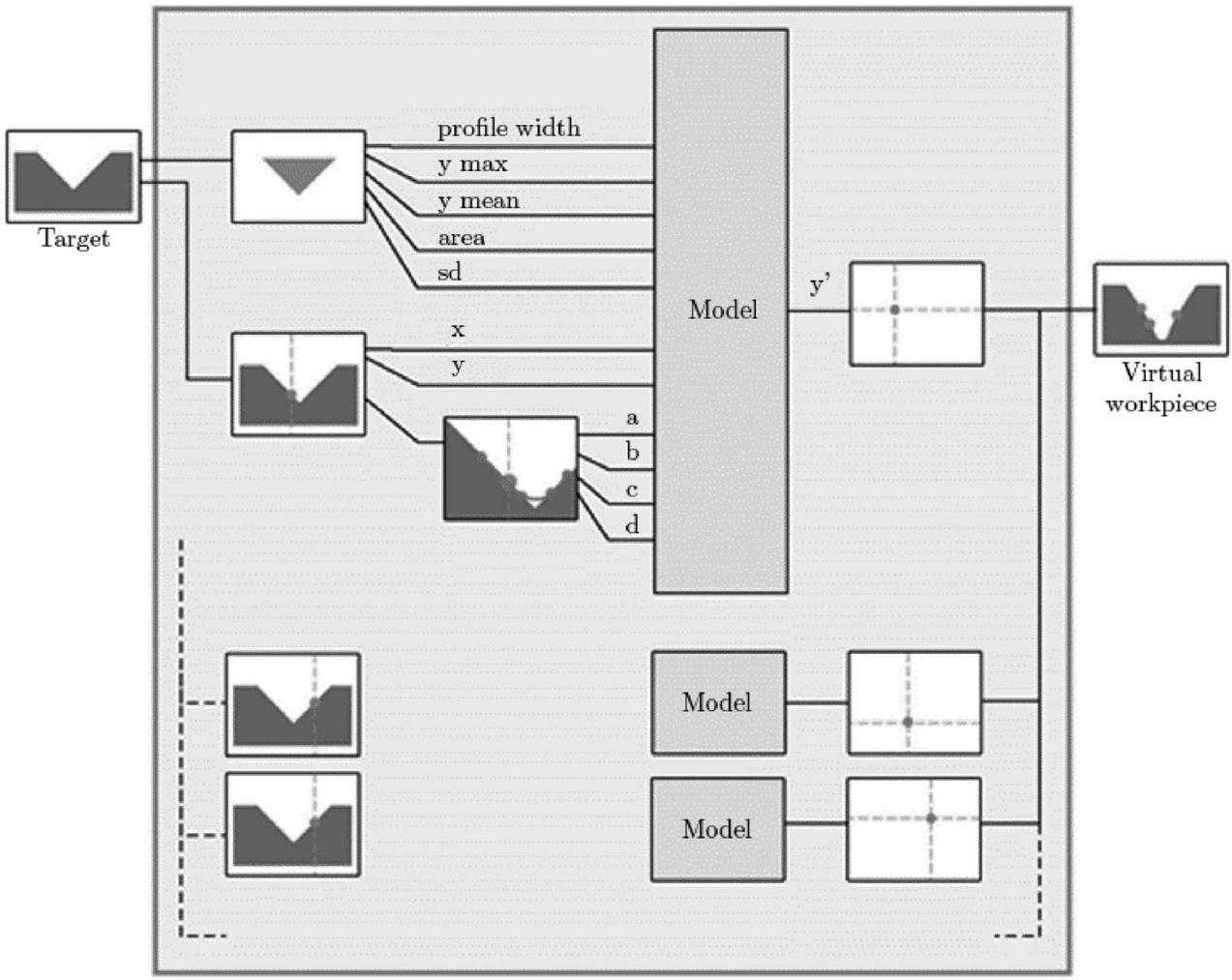

Fig. 9 A detailed overview of the machine learning model together with examples of its inputs and its output

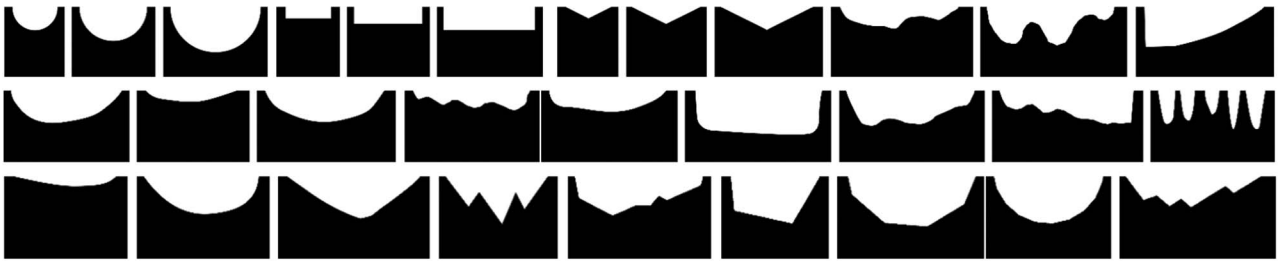

Fig. 10 The 30 shapes used to train the neural network to estimate the optimal tool shape

since the interpolation is centered on the given point, making the constant always equal to 0.

- Attributes providing information on the global context (i.e., the target workpiece):
  - profileWidth: Total width of the target workpiece (in μm).
  - profileHeight: Total height of the target workpiece (in μm).
  - mean: Mean *Y*-value of the target workpiece points (in μm).
  - area: Total area (or volume in 3D) of the target workpiece (in μm^2).
  - sd: Standard deviation of the *Y*-values of the target cavity points (in μm).
- Attributes providing information on the point relative to the rest of the target cavity:
  - xValueRel: *X* position of the point relative to the rest of the profile (0 being the center and 1 corresponding to an extremity of the profile).
  - yValueRel: *Y* position of the point relative to the rest of the profile (0 being the base and 1 corresponding to the maximum height of the profile).
  - dirToLocalMax: Slope toward the closest local maximum.
  - distToLocalMax: Distance to the closest local maximum (in μm).
  - equivalents: Number of times the target workpiece profile goes through the point's altitude.
- Output variable:
  - yValuePattern: *Y*-value of the respective point on the pattern workpiece profile.

This results in 14 input attributes and 1 output variable, all of which are continuous real values without any missing data.

3.4 Machine Learning Model

3.4.1 Model Selection. Machine learning has a large variety of models that come with their respective advantages and disadvantages. When it comes to supervised learning for regression models like in our case, there are two types of model that usually perform best: ANNs and support vector machines (SVMs).

It has been decided to use ANNs because they can discover the most complex relationships between inputs and outputs and thus have the potential to give excellent results. However, they are to be used carefully. Indeed, the model complexity of ANNs rises quickly with the number of features, making it slow to train and prone to suffer from multiple local minima. As a consequence, SVMs are usually favored when dealing with high dimension input vectors. Fortunately, this is not the case presently since only 14 input attributes are to be considered.

Furthermore, the simulation tool has been developed in C#. In order to make use of the tool prediction process, it is advantageous that the model can be implemented inside the application itself. However, this language only has a few machine learning frameworks compared to other more computationally oriented languages like PYTHON, R, or MATLAB. None of these available machine learning frameworks have a good support for SVMs, whereas the C# framework Encog [25] supports a large variety of neural network algorithms making it possible to optimize the performances of the model with already available features.

As a consequence, ANN has been deemed the most adequate candidate model in this context.

3.4.2 Model Configuration. One of the crucial steps when using ANNs is determining the model's configuration, meaning choosing how many hidden layers and hidden neurons the networks should have. This task is very complex and has a huge impact on the performance of the network. If the hidden structure is too complex, the model will take long to train, will be prone to overfitting, and will have a hard time dealing with local minima. If the hidden structure is too simple, the model will not learn the problem (underfitting).

The incremental pruning technique [26] was implemented to determine the hidden structure to use. The objective of incremental pruning is to figure out the most promising hidden layer configuration out of a variety of potential configurations.

First, some basic rules of thumb were used to determine the reasonable ranges for the number of hidden layers and their ranges of neurons. Then, every network configuration was tested three times for 300 iterations, then the five networks giving the best scores were saved. Finally, only the network with the simplest configuration of all is selected for training efficiency as well as overfitting prevention.

The most promising model in this case turned out to be a network with a single hidden layer composed of 14 neurons. Now that the model configuration is set, the network is ready to be trained.

3.4.3 Training Criteria. The method used for training is the resilient backpropagation algorithm (RPROP) for feedforward ANN. RPROP is one of the best general-purpose training methods for neural network and have the advantage of having no parameters to tune [27].

In order to reach the minimum error possible, the training method makes use of the cross-validation set. While iterating through the training set and adjusting the weights of the network accordingly to the RPROP method, the algorithm computes the error of the network on the validation set. As long as the validation error keeps improving by a certain amount over multiple iterations, the training goes on. This method ensures that the training continues as long as it is effective even on unseen data while preventing overfitting. Indeed, overfitting occurs when the model starts fitting the training data too well and does not generalize well anymore on unseen data (the cross-validation set).

3.4.4 Model Performances. Once it is trained, the model performances can be evaluated with various metrics on the test set. Here are the two commonly adopted metrics considered in this work:

- Mean absolute error (MAE) gives the mean absolute error of the predicted values.
- Coefficient of determination (R^2) indicates how well the model predicts the values.

Table 2 Evaluation metrics for point altitude prediction

Metric	Value
R^2	0.957
MAE (μm)	3.88

Table 3 Evaluation metrics for the predicted tools

Metric	Value
Training set accuracy MAE (%)	2.06
Test set accuracy MAE (%)	3.06

The values present in Table 2 mean that on average each predicted point of the pattern cavity is 3.88 μm away from its actual position (given by the iterative optimization process).

Although these error measures provide insight on how well the model can predict the pattern cavity, the end goal is the performance of the resulting tool. In order to evaluate the performance of the predicted tool, the accuracy metric introduced in Sec. 2.2 is used. By simulating the µEDM machining of the predicted tool and the actual optimized tool, it is possible to compare their accuracy and measure the mean absolute error between them.

The figures in Table 3 indicate that, on average, the predicted tool for a new target workpiece would be 3.06% less accurate than the actual optimized tool.

Also, it is good to know that, on average, the predicted tool is as accurate as an optimized tool that would be halfway through the iterative optimization process. In other words, the prediction does half the work of the optimization process. Thus, one way to make use of the prediction model is to start the tool optimization process with the predicted tool right from the beginning. Doing so will save a considerable amount of time, especially if 3D shapes were to be involved.

4 Experimentations and Discussion

Section 3 has introduced the details of the proposed machine learning-based tool shape optimization without considering the machining parameters as potential attributes to be taken into by the machine learning model. Such a possibility is addressed in this section thus extending the capabilities of the proposed approach.

4.1 Data Generation. The previous section focused on applying a machine learning method for a single set of machining parameters while focusing on varying the target workpiece profiles. This section explores the possibility of varying other parameters while studying the change on the method's accuracy.

In order to get a sufficient amount of data to obtain robust results, multiple tool shape optimizations were performed using a large variety of target profiles, machining gaps, and crater dimensions:

- Target profiles: 30 shapes (shown in Fig. 10).
- Machining gaps: 4 values (5 μm, 10 μm, 15 μm, 20 μm).
- Crater dimensions: 45 combinations (see Table 4).

The crater dimensions have been generated starting with five values for the workpiece crater radii: 2, 2.5, 3, 3.5, and 4 μm. Each of those is associated with three crater depths generated from three values called crater ratios γ: 1, 1.5, and 2 such as:

$$\gamma = \frac{R_e}{D_e} \tag{2}$$

Table 4 Crater dimension combinations used for the generation of the training data

	Workpiece crater		Tool crater			
No.	R_w (μm)	D_w (μm)	R_t (μm)	D_t (μm)	γ	β
1	2	2.00	2.44	2.44	1	1.5
2	2	2.00	2.00	2.00	1	1
3	2	2.00	1.41	1.41	1	0.5
4	2	1.33	2.44	1.63	1.5	1.5
5	2	1.33	2.00	1.33	1.5	1
6	2	1.33	1.41	0.943	1.5	0.5
7	2	1.00	2.44	1.22	2	1.5
8	2	1.00	2.00	1.00	2	1
9	2	1.00	1.41	0.707	2	0.5
10	2.5	2.50	3.06	3.06	1	1.5
11	2.5	2.50	2.50	2.50	1	1
12	2.5	2.50	1.76	1.76	1	0.5
13	2.5	1.66	3.06	2.04	1.5	1.5
14	2.5	1.66	2.50	1.66	1.5	1
15	2.5	1.66	1.76	1.17	1.5	0.5
16	2.5	1.25	3.06	1.53	2	1.5
17	2.5	1.25	2.50	1.25	2	1
18	2.5	1.25	1.76	0.884	2	0.5
19	3	3.00	3.67	3.67	1	1.5
20	3	3.00	3.00	3.00	1	1
21	3	3.00	2.12	2.12	1	0.5
22	3	2.00	3.67	2.44	1.5	1.5
23	3	2.00	3.00	2.00	1.5	1
24	3	2.00	2.12	1.41	1.5	0.5
25	3	1.50	3.67	1.83	2	1.5
26	3	1.50	3.00	1.50	2	1
27	3	1.50	2.12	1.06	2	0.5
28	3.5	3.50	4.28	4.28	1	1.5
29	3.5	3.500	3.500	3.500	1	1
30	3.5	3.500	2.475	2.475	1	0.5
31	3.5	2.333	4.287	2.858	1.5	1.5
32	3.5	2.333	3.500	2.333	1.5	1
33	3.5	2.333	2.475	1.650	1.5	0.5
34	3.5	1.750	4.287	2.143	2	1.5
35	3.5	1.750	3.500	1.750	2	1
36	3.5	1.750	2.475	1.237	2	0.5
37	4	4.000	4.899	4.899	1	1.5
38	4	4.000	4.000	4.000	1	1
39	4	4.000	2.828	2.828	1	0.5
40	4	2.667	4.899	3.266	1.5	1.5
41	4	2.667	4.000	2.667	1.5	1
42	4	2.667	2.828	1.886	1.5	0.5
43	4	2.000	4.899	2.449	2	1.5
44	4	2.000	4.000	2.000	2	1
45	4	2.000	2.828	1.414	2	0.5

where R_e and D_e are, respectively, the radius and depth of the craters for the tool ($e = t$) and the workpiece ($e = w$).

The dimensions for the tool crater radii and depth are found using the following system of equations:

$$\begin{cases} \gamma = \dfrac{R_w}{D_w} = \dfrac{R_t}{D_t} \\ \beta = \dfrac{V_t}{V_w} \end{cases} \tag{3}$$

where β is the tool wear ratio which is here the ratio between the volume of the crater of the tool and the area (or volume in the 3D case) of a crater of the workpiece. It yields

$$\begin{cases} R_t = \sqrt{\beta} R_w \\ D_t = \dfrac{\sqrt{\beta} R_w}{\gamma} \end{cases} \tag{4}$$

For information, the three-dimensional equivalent of this system is

$$\begin{cases} R_t = \sqrt[3]{\beta} R_w \\ D_t = \dfrac{\sqrt[3]{\beta} R_w}{\gamma} \end{cases} \tag{5}$$

Those 45 combinations associated with the four machining gap values and 30 different shapes lead to a total number of optimizations of 5800. Those were accomplished in under 12 days using two different computers running the optimizations concurrently.

4.2 Training the Model. Similarly to what has been described in Sec. 4.1, the model has been generated and trained. The available data were divided into

- A training set (60%): 3480 optimizations (approximately 1 million instances).
- A cross-validation set (20%): 1160 optimizations (approximately 350,000 instances).
- Test set (20%): 1160 optimizations (approximately 350,000 instances).

Once again, an ANN was trained using the resilient backpropagation algorithm described previously. The main difference here is the addition of a few input variables: the machining gap and the crater dimensions. As a result, the input parameters have a count of 19 for one output parameter.

4.3 Computational Performances. This new approach models the output of the optimization process directly. Therefore, the neural network prediction is used as a surrogate for the model-based iterative optimization process, each of the iteration consisting of a detailed time-dependent simulation of the EDM process. Therefore, this meta-modeling approach is expected to yield extremely large computational savings.

For the 2D study described previously, the optimization of a typical test cavity is completed using four iterates of the optimization algorithm. Each of the forward evaluation of the EDM model takes about 15 s to run, leading to a total of 60 s.

On the same computer, the evaluation of the neural network for a particular local feature takes about 10 ms (the neural network is shallow). For a tool width described using 600 pixels, the total prediction of the tool shape using the meta-model is roughly equal to 100 ms, taking into account that the 600 forward evaluations of the neural network are performed in batch to accelerate the process. Therefore, this leads to a total speed-up of more than 5 orders of magnitude (i.e., 60 divided by 0.0001).

In the context of 3D tool shape predictions, the costly computation power required to create the initial training set for the neural network would still require the use of high-performance computing, but once created, the resulting neural network will be significantly less computationally costly when predicting optimum 3D tool shapes. It is expected that the speed-up would be at least larger than the one mentioned above for the 2D study.

4.4 The Test Set. Once the training of the model is achieved, it is submitted to the test set. A first result yields the following metrics (Table 5). As it could be expected, those error values are greater than in the case of the limited test set (R^2 of 0.931 against 0.957 and a mean absolute error of 4.02 μm against 3.88 μm).

Table 5 Evaluation metrics for point altitude prediction for the extended dataset

Metric	Value
R^2	0.931
MAE (μm)	4.02

Table 6 Test set optimizations used for direct comparison of the achieved workpiece results

No.	Shape no.	Machining gap (μm)	β	γ	R_w (μm)
1	9	20	0.5	1	2
2	6	15	0.5	2	3.5
3	4	10	1.5	1.5	3.5
4	29	15	0.5	1.5	2
5	24	5	1	1.5	4
6	2	10	1.5	1.5	2.5
7	19	5	0.5	1	3
8	14	10	0.5	2	4
9	18	15	0.5	1	3.5
10	12	10	0.5	2	2.5
11	10	5	0.5	2	3.5
12	1	5	1.5	2	2.5
13	16	20	1.5	1	2.5
14	17	10	1.5	2	4
15	13	20	1	1.5	3.5
16	15	5	0.5	2	3
17	2	15	0.5	1	4
18	20	20	1	1.5	2
19	22	5	0.5	2	2.5
20	27	15	1.5	2	4

Once again what is of interest is to compare the performances of a tool generated through the iterative process against one generated by the ANN model. Due to the huge size of the test set, not all of the tools generated by the ANN could be tested individually against their iterative counterparts. A selection of 20 of them was made across a broad range of parameters. Those are tabulated in Table 6.

The various accuracies of the two methods (iterative and machine learning) are given in Table 7 and the mean, variance, and standard deviation of the differences are given in Table 8.

Once again, the accuracy of the method based on machine learning is worse than of the iterative one. However, considering the much larger range in terms of number of parameters, these results are quite good. In order to provide with a visual representation of those errors, an example of optimization using the iterative method and its equivalent with the machine learning method are depicted in Fig. 11.

Table 7 Comparison of the accuracies of the machine learning method and the iterative method

No.	Accuracy machine learning (%)	Accuracy iterative (%)	Difference (%)
1	45.528	54.159	8.630394
2	34.032	40.350	6.317104
3	16.918	23.730	6.811837
4	37.907	44.414	6.507592
5	57.505	60.315	2.810144
6	82.353	85.432	3.078615
7	66.468	73.562	7.09389
8	47.791	52.209	4.417671
9	62.920	67.627	4.707012
10	46.224	52.387	6.163142
11	48.312	57.806	9.493671
12	74.272	81.956	7.683353
13	17.105	24.934	7.828947
14	32.836	34.453	1.616915
15	63.314	69.930	6.61646
16	35.729	43.340	7.610994
17	65.506	67.247	1.740506
18	27.028	32.968	5.939885
19	21.608	26.131	4.522613
20	59.149	64.771	5.622359

Table 8 Statistical properties of the accuracy differences between the machine learning and iterative methods for the 20 optimizations chosen

Measure	Value
Mean	5.760
Variance	4.597
Standard deviation	2.144

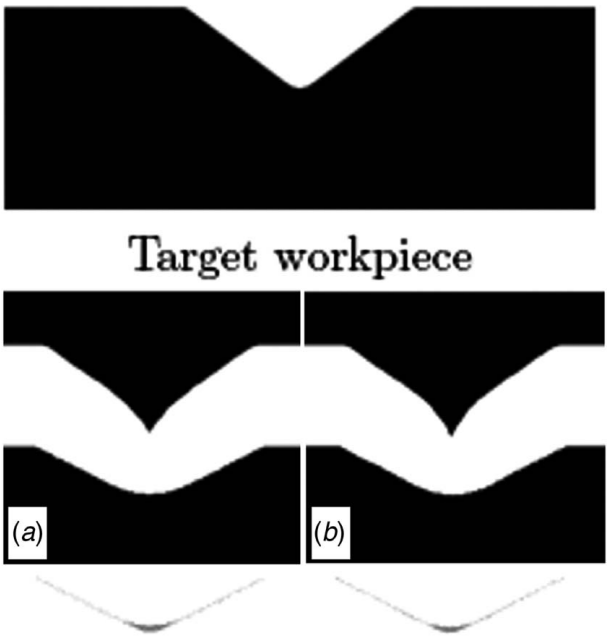

Fig. 11 Visual side-by-side comparison of the machine learning method (*a*) and the iterative method (*b*) for test optimization number 9

5 Conclusion and Future Work

The data-driven model reduction approach proposed in this paper demonstrated that an ANN could be trained and used as a replacement for costly numerical EDM tool shape optimization tasks. It was shown that tool shapes corresponding to a wide range of targeted and unseen geometrical workpiece cavities could be accurately generated by the ANN.

With fixed μEDM process parameters (gap size and crater shape), the ANN predicted tool shapes for unseen targeted workpiece cavities with an accuracy only 3.06% lower than the full iterative tool shape optimization and with a similar accuracy to half-completed iterative optimization processes. Thus, the method could also be used to significantly accelerate the iterative optimization process if higher accuracy is needed. As part of the proposed study, a new ANN training framework was designed to increase the number of training instances, from 30 optimal tool shapes to 18,000 localized tool surface optimizations, thus enabling an accurate training with a limited number of EDM simulations.

Using a similar approach, another ANN was produced, using various machining gaps and crater shapes, thus increasing significantly the search domain. The ANN was trained using 3480 optimized tool shapes, which was equivalent to around one million feature instances. For this second ANN, the predicted tool for a new target workpiece was 5.76% less accurate than the fully optimized tool.

Overall, the results demonstrated that accurate μEDM die-sinking tool shape could be generated almost instantaneously using the proposed data-driven modeling approach. It is expected that this work will help product designers virtually design, optimize, and assess the feasibility of μEDM operations.

Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council [EP/J004901/1].

Nomenclature

$d_{\max}$ = minimum distance upper bound between a tool node and a workpiece node
$d_{\min}$ = minimum distance lower bound between a tool node and a workpiece node
$d_{\text{smallest Max}}$ = smallest recorded $d_{\max}$
$A_{W_{\text{target}}}$ = area (in μm^2) of the target workpiece feature
D_e = crater depth
D_t = tool crater depth
D_w = workpiece crater depth
R_e = crater radius
R_t = tool crater radius
R_w = workpiece crater radius
T_c = corrected tool profile
V_t = tool crater volume
V_w = workpiece crater volume
W_{result} = resulting workpiece profile
W_{target} = targeted workpiece profile
W_{virtual} = "virtual workpiece" profile
R^2 = coefficient of determination, indicates how well the model predicts the values
dirToLocalMax = slope toward the closest local maximum
distToLocalMax = distance to the closest local maximum (in μm)
sd = standard deviation of the *Y*-values of the target cavity points (in μm)
xValueRel = *X* position of the point relative to the rest of the profile
yValue = *Y*-value of the point on the target workpiece profile (in μm)
yValueRel = *Y* position of the point relative to the rest of the profile
yValuePattern = *Y*-value of the respective point on the pattern workpiece profile
$A\%$ = accuracy of the optimization process
β = tool wear ratio
γ = crater ratio
ε = area mismatch (in μm^2)

References

[1] Islam, N., and Miyazaki, K., 2009, "Nanotechnology Innovation System: Understanding Hidden Dynamics of Nanoscience Fusion Trajectories," Technol. Forecast. Soc. Change, **76**(1), pp. 128–140.
[2] Maropoulos, P. G., 2003, "Digital Enterprise Technology-Defining Perspectives and Research Priorities," Int. J. Comput. Integr. Manuf., **16**(7–8), pp. 467–478.
[3] Shao, B., and Rajurkar, K. P., 2015, "Modelling of the Crater Formation in Micro-EDM," Procedia CIRP, **33**, pp. 376–381.
[4] Kalajahi, M. H., Ahmadi, S. R., and Oliaei, S. N. B., 2013, "Experimental and Finite Element Analysis of EDM Process and Investigation of Material Removal Rate by Response Surface Methodology," Int. J. Adv. Manuf. Technol., **69**(1–4), pp. 687–704.
[5] Tan, P. C., and Yeo, S. H., 2008, "Modelling of Overlapping Craters in Micro-Electrical Discharge Machining," J. Phys. D: Appl. Phys., **41**(20), pp. 1–12.
[6] Santos, P., Teixidor, D., Maudes, J., and Ciurana, J., 2014, "Modelling Laser Milling of Microcavities for the Manufacturing of DES With Ensembles," J. Appl. Math., **2014**, pp. 1–15.
[7] Kunieda, M., Kanekob, Y., and Natsub, W., 2012, "Reverse Simulation of Sinking EDM Applicable to Large Curvatures," Precis. Eng., **36**(2), pp. 238–243.
[8] Gilbert, D., Stoesslein, M., Axinte, D., Butler-Smith, P., and Kell, J., 2014, "A Time Based Method for Predicting the Workpiece Surface Micro-Topography Under Pulsed Laser Ablation," J. Mater. Process. Technol., **214**(12), pp. 3077–3088.

[9] Claus, S., Bigot, S., and Kerfriden, P., 2018, "CutFEM Method for Stefan–Signorini Problems With Application in Pulsed Laser Ablation," SIAM J. Sci. Comput., **40**(5), pp. B1444–B1469.
[10] Otto, A., and Schmidt, M., 2010, "Towards a Universal Numerical Simulation Model for Laser Material Processing," Phys. Procedia, **5**(Part A), pp. 35–46.
[11] Kerfriden, P., Passieux, J. C., and Bordas, S. P. A., 2011, "Local/Global Model Order Reduction Strategy for the Simulation of Quasi-Brittle Fracture," Int. J. Numer. Methods Eng., **89**(2), pp. 154–179.
[12] Ghanem, R., and Spanos, P., 1991, *Stochastic Finite Elements: A Spectral Approach*, Springer Verlag, New York.
[13] Rasmussen, C. E., and Williams, C. K. I., 2006, *Gaussian Processes for Machine Learning*, The MIT Press, Boston, MA.
[14] Chinesta, F., Leygue, A., Bordeu, F., Aguado, J. V., Cueto, E., Gonzalez, D., Alfaro, I., Ammar, A., and Huerta, A., 2013, "PGD-Based Computational Vademecum for Efficient Design," Arch. Comput. Meth. Eng., **20**(1), pp. 31–59.
[15] Pham, D., Ivanov, A., Bigot, S., Popov K., and Dimov, S., 2007, "A Study of Micro-Electro Discharge Machining Electrode Wear," Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci., **221**(5), pp. 605–612.
[16] Liang, W., Tong, H., Li, Y., and Li, B., 2019, "Tool Electrode Wear Compensation in Block Divided EDM Process for Improving Accuracy of Diffuser Shaped Film Cooling Holes," Int. J. Adv. Manuf. Technol., **103**(5–8), pp. 1759–1767.
[17] Pham, D. T., Dimov, S. S., Bigot, S., Ivanov, A., and Popov, K., 2004, "MicroEDM—Recent Developments and Research Issues," J. Mater. Process. Technol., **149**(1–3), pp. 50–57.
[18] Bleys, P., Kruth, J. P., Lauwers, B., Zryd, A., Delpretti, R., and Tricarico, C., 2002, "Real Time Tool Wear Compensation in Milling EDM," CIRP Ann., **51**(1), pp. 157–160.
[19] Bissacco, G., Hansen, H. N., Tristo, G., and Valentinčič, J., 2011, "Feasibility of Wear Compensation in Micro EDM Milling Based on Discharge Counting and Discharge Population Characterization," CIRP Ann., **60**(1), pp. 231–234.
[20] Yu, Z., Masuzawa, T., and Fujino, M., 1998, "3D Micro-EDM With Simple Shape Electrode," Int. J. Electr. Mach., **3**, pp. 7–12.
[21] Surleraux, A., Pernot, J., and Bigot, S., 2016, "A Comparative Study Between NURBS Surfaces and Voxels to Simulate the Wear Phenomenon in Micro-EDM," Comput. Aided Des. Appl., **13**(6), pp. 792–798.
[22] Surleraux, A., Lepert, R., Pernot, J.-P., and Bigot, S., 2015, "Computer-Aided Micro-EDM Die-Sinking Tool Design Optimization," J. Innov. Impact, **8**(2), pp. 552–571.
[23] Bigot, S., Pernot, J., Surleraux, A., and Elkaseer, A., 2013, "Micro-EDM Numerical Simulation and Experimental Validation," Proceedings of the 10th International Conference on Multi-Material Micro Manufacture, San Sebastian, Spain, Oct. 8–10, pp. 55–58.
[24] Kudryavtsev, L. D., and Samarin, M. K., 2011, "Lagrange Interpolation Formula," http://www.encyclopediaofmath.org/index.php?title=Lagrange_interpolation_formula&oldid=17497, Accessed 6 April, 2019.
[25] Heaton, J., 2013, *Artificial Intelligence for Humans, Volume 1: Fundamental Algorithms*, CreateSpace Independent Publishing Platform.
[26] Cassandra, A., Littman, M. L., and Zhang, N. L., 1997, "Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes," Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), Providence, RI, Aug. 1–3, pp. 54–61.
[27] Heaton, J., 2011, *Introduction to the Math of Neural Networks*, Heaton Research Inc.