
Article:
Zotov, E., Tiwari, A. and Kadirkamanathan, V. orcid.org/0000-0002-4243-2501 (2021) Conditional StyleGAN modelling and analysis for a machining digital twin. Integrated Computer-Aided Engineering, 28 (4). pp. 399-415. ISSN 1069-2509

https://doi.org/10.3233/ica-210662

Undefined 0 (0) 1
IOS Press

Conditional StyleGAN modelling and analysis for a machining digital twin

Evgeny Zotov [a], Ashutosh Tiwari [a], Visakan Kadirkamanathan [a,*]
[a] *Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, S1 3JD, Sheffield, UK*
{ezotov1, a.tiwari, visakan}@sheffield.ac.uk

Abstract. Manufacturing digitalisation is a critical part of the transition towards Industry 4.0. Digital twin plays a significant role as the instrument that enables digital access to precise real-time information about physical objects and supports the optimisation of the related processes through conversion of the big data associated with them into actionable information. A number of frameworks and conceptual models has been proposed in the research literature that addresses the requirements and benefits of digital twins, yet their applications are explored to a lesser extent.

A time-domain machining vibration model based on a generative adversarial network (GAN) is proposed as a digital twin component in this paper. The developed conditional StyleGAN architecture enables (1) the extraction of knowledge from existing models and (2) a data-driven simulation applicable for production process optimisation. A novel solution to the challenges in GAN analysis is then developed, where the comparison of maps of generative accuracy and sensitivity reveals patterns of similarity between these metrics. The sensitivity analysis is also extended to the mid-layer network level, identifying the sources of abnormal generative behaviour. This provides a sensitivity-based simulation uncertainty estimate, which is important for validation of the optimal process conditions derived from the proposed model.

Keywords: Generative adversarial networks, Digital twin, Machining, Simulation, Time-domain signals

1. Introduction

1.1. Digitalisation in Industry 4.0

The 4th industrial revolution, i.e. the strategic vision of transition to Industry 4.0, draws a path to a totally customisable production with viable single-item batch production, just-in-time execution and high resource-efficiency. Advances along this path are believed to be feasible as a result of pervasive digitalisation throughout the industry, spanning from the shop-floor to the whole supply chain and to the users of the end-products [29].

Total factory digitalisation is being made possible by the technologies emerging from the research fields of big data, cyber-physical systems (CPS) and industrial internet of things (IIoT). The implementation of these technologies is becoming increasingly accessible as the big data technology stack has matured over the last couple of decades, offering many open-source and proprietary software tools supporting its deployment on commodity hardware [2]. As a result, IIoT implementation became feasible [46] and one anticipates seeing the industry shift rapidly towards widespread adoption of IIoT technologies as companies compete to disrupt the market or to maintain their position on it by reaping the performance and efficiency benefits. This would be a significant step in the evolution towards complete CPS integrated end-to-end through the value chain. Despite notable progress in development of the strategic vision of Industry 4.0 and increasing affordability of hardware and software solutions, actual implementation of smart factory technologies on manufacturing shop-floors remains relatively low, es-

*Corresponding author. Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, S1 3JD, Sheffield, UK. Tel.: +44 114 222 5680; Fax: +44 114 222 5661; E-mail: visakan@sheffield.ac.uk

pecially among small and medium enterprises [69]. Researchers and industry experts attribute this fact to the high complexity of execution, long return on investment periods and high investment costs [69,32].

Precise representation of physical objects or processes in the digital realm is researched within the digital twin domain. Definition of the digital twin has varied throughout its existence: from the aircraft-oriented definition originally introduced in the aerospace domain research [61] to the modern concept of digital twin as a complete digital recreation of whole ecosystems [7]. The main aspect common to all the proposed scales is the absolute information equivalence between the perfect digital and the physical twins, implying that any interaction with one is mirrored in another [24]. Development of digital twins is an important step of the digitalisation process, as the unification of digital and physical data within a single virtual object enables significant efficiency improvements across multiple stages of the object's life cycle [68].

A holistic digital twin requires integration of multiple interconnected models and metrology tools that capture the different aspects of the complete system. The progress towards this goal would thus be followed in stages with gradual development of digital twin components, including infrastructure, monitoring and predictive systems. Only after significant research advancement in these directions would the digital twin become usable as a decision-making tool capable of delivering the robust efficiency improvements promised by technology visionaries [24,47].

1.2. Simulation and data-driven modelling

Simulation modelling is a widely used technique employed in the verification of engineering designs and evaluation of their functionality and performance. By utilising the known information about the product geometry, its material characteristics and operating conditions, the virtual recreation of its operation environment enables analysis of the product's behaviour. A product can be similarly studied during a manufacturing process, and its feature characteristics both after and during processing can be predicted [53].

The enhanced efficiency is going to be increasingly attractive to the manufacturers of high-value products, as simulations enable a shifting of the physical experimentation costs to virtual analysis. And the increased flexibility shall enable agile product design, which is a prerequisite for agile manufacturing. In both cases a significant reduction of waste, cost and lead times is to be expected [17].

The computational constraints limiting the usefulness of simulation modelling in previous decades is significantly compensated for by the modern hardware and software advances. Thus, simulations have become a critical tool utilised in the analysis of the various aspects related to the machining processes, such as process stability [4], surface finish [9], cutting forces [1] or tool stress and temperature [51] and in the validation of physics-based models [70,62,5] over the last several decades [65]. Their further refinement with hybrid data-driven and knowledge-based approaches is an ongoing research topic [23].

The variety of error sources and their dynamic nature significantly distort the model predictions and present a considerable challenge that has been and still is thoroughly studied by manufacturing researchers [79,45,15,42,71]. These sources include the material uncertainties (such as its workability, shear stress or deviations from material specifications) and machining uncertainties (for example, tool wear and run-out, machine geometry and thermal errors). These production process-related uncertainties are propagated and magnified throughout the potential measurement errors arising from the measurement strategy choice, fixturing and environment variability, as well as from measurement tool and software errors [19,53].

Physics-based analytics models tend to achieve high accuracy rates, but have several drawbacks that can become blocking factors for implementation of a digital twin. On one hand, in an interconnected CPS environment interactions between the components introduce very high complexity of the modelled phenomena. On the other hand, the incremental character of module development and the fluid module composition cause an almost constant stream of changes in the system [49]. This and the utilisation of empirically estimated parameters that approximate some of the unobserved factors in physics-based models implies that a given model has to be manually adapted to every new environment or scenario it is deployed in. Data-driven modelling addresses this issue by making use of the big data produced by the various manufacturer's CPSs and automating the modelling process, thus aligning the digital twin state with the evolutionary changes in the modelled systems. Therefore, despite the wide success of physics-based simulation models for prediction of abnormal conditions during milling and turning processes, data-driven simulation methods offer greater

flexibility at adapting to a broader range of conditions, including dynamically changing ones [20].

1.3. Artificial neural networks in manufacturing

Development of efficient and flexible data-driven simulation models of physical manufacturing processes is an important step towards CPS digitalisation in general, and particularly to wide adoption of digital twins throughout the industry. Artificial neural network (ANN) is a machine learning model inspired by the information flow structure of a brain's neurons. Scaled versions of these networks labelled as deep learning models [41] have shown increasingly impressive state-of-art results on many data-driven problems (e.g., [64,34,80,58,73]) and have attracted a lot of attention from the manufacturing research community for their wide applicability to problems from different domains and of different scopes [55]. ANNs have been utilised in manufacturing applications ranging from conventional machine health monitoring [33,83,42] to product quality monitoring in additive manufacturing [63]. The proposed applications of ANNs to machining domain problems include chatter prediction [12,50], fault diagnosis [78], surface defect detection [39] and several others [38].

Generative adversarial network (GAN) is a type of ANN architecture based on a competitive minimax game between two ANNs: the generator that learns to produce artificial data samples and the discriminator that learns to identify fake data samples [22]. The data distribution produced by the generator network thus approaches a representation of the true data which can be directly utilised for simulation of the process underlying the data. The approach was extended in multiple directions, including but not limited to research on various neural network architectures for the generator and the discriminator (e.g., Deep Convolutional GAN [57], BigGAN [8], StackGAN [82], WaveGAN [14], SeqGAN [81], Bayesian GAN [59]), reviews of the GAN training approaches and the networks' loss functions (notably, the application of Earth Mover distance as the loss metric in Wasserstein GAN [6], its extension with gradient penalty in WGAN-GP [25] and progressive GAN growing [36]) and experiments with the conditioning of the GAN by additional inputs or outputs, as proposed in Conditional GAN [43], InfoGAN [10] and ss-InfoGAN [66].

Recently GANs have advanced the state of the art results in various domains, including generation of music [16], speech [14,21,35], text [81,48] and, most notable, in simulation of realistic high-resolution images of human faces [37]. With most of GAN studies focussed on image-generation, the generation of time-domain signals with GANs remains a very narrow field of research. Some examples can be found within the publications on healthcare [72,18], energy [11], music [44] and manufacturing [77].

Data augmentation aimed at the support of a primary classification model [26,76,75] currently dominates the research agenda on GANs in manufacturing. This is also evidenced by the review papers touching on ANN and GAN applications in manufacturing, where the only identified use case for GAN models is data augmentation [56,33,74,40]. Adoption of a GAN's generator as a primary instrument for manufacturing problems are discussed within image-generation GAN research considering generative material design [67] and sample super-resolution [3]. Additionally, a very recent anomaly detection approach was proposed in [13] via inversion of the generator, i.e. the authors considered the optimisation of the generator input with the rest of the network fixed for a given data sample. The generative accuracy of the sample produced by the GAN this way is suggested as a measure of the abnormality of the given data sample.

At the current stage of industrial digitalisation it is likely that a data-driven method would lack the range of empirical data variability necessary for capturing the underlying process behaviour. It is thus likely that a training regime based on hybrid dataset comprised of the experimental data and the data obtained from an existing model would be employed at first. The reuse of large pre-trained models and their adaptation to the real world conditions is discussed within the field of transfer learning [52].

1.4. Paper novelty and outline

An unexplored area of research of GAN applications in the manufacturing field is the potential use of controllable generative features of a GAN for analysis of the manufacturing processes. Research in this direction can potentially uncover ANN-based data-driven simulation techniques that could significantly augment the decision-making process pipelines in a manufacturing enterprise. This work focusses on simulation via data-driven generation as a component of a future digital twin. GAN is a suitable candidate for digital twin development due to its efficiency at inference time and the generative nature of the model, in addition to the flexibility benefits of a data-driven method that reduce

the expected cost of implementation of the model for highly variable processes.

This paper proposes the first StyleGAN-based digital twin machining simulation component. A conditional StyleGAN architecture is developed that captures the conditional distribution of a vibration signal. The process signal generation is controllable via manipulation of the input machining process parameters. The model may thus act as a vibration simulation tool that maps process parameter inputs to vibration signal outputs. This makes the proposed GAN usable as a process optimisation instrument. An optimisation process loop would search for the best process parameters by interrogating the model to obtain parameters-signal pairs and determining the process quality based on the obtained signals. This paper also introduces a method of uncertainty analysis that is applicable to the optimised process state. To this end, a novel generation sensitivity analysis technique is proposed that aims to estimate the conditions under which the simulation yields reliable results.

The following paper describes the data and the neural network model in section 2, followed by the methodology and the experimental analysis in sections 3 and 4 and concluded with a discussion of the implications of the analysis results for potential applications and future research in section 5.

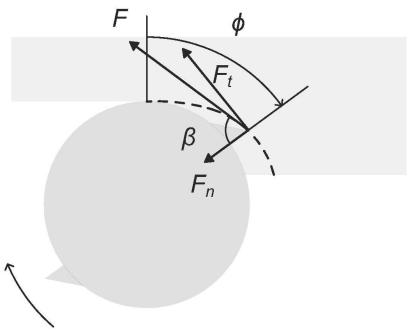


Fig. 1. Geometrical representation of the forces simulated by the physics-based model. F is the cutting force and β is the force angle. F_t and F_n denote the tangential and the normal cutting forces and ϕ is the cutting tooth angle.

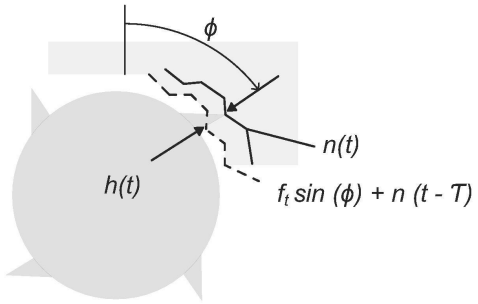


Fig. 2. Workpiece geometry produced by the simulation model. $h(t)$ is the instantaneous depth of cut at time t that is the distance between the current normal direction vibration level at angle ϕ and the cut surface at angle ϕ produced at time $t-T$, where T is the time period of cutting tool revolution between two neighbouring cutting teeth.

2. GAN model as a machining digital twin

2.1. Dataset: Machining Tool Vibration

Manufacturing process data is a scarce resource at the current moment due to its acquisition cost. Commercial confidentiality of such data adds an additional impediment to its use in public research. It can be expected that actual implementations of data-driven digital twins would be initially trained on the existing and proven physics-based models, due to the experimental data scarcity mentioned above, and fine-tuned using a mixture of simulated and empirical data. The simulation that produced the dataset used in this work represents a surrogate of a real operational data-generating process. On one hand, this allows for a rigorous analysis of the digital twin component performance due to the full control over data generation. On the other hand, the proposed approach approximates a real world scenario of transition from a pure physics-based modelling to a scenario with mixed physics-based and experimental data.

The GAN model described in this work is trained on a synthetic dataset produced by a physics-based time-domain simulation model adopted from [60]. The simulation iteratively calculates the forces produced by the interaction between the cutting teeth of a non-rigid machining tool and a rigid workpiece (Figure 1). These forces are used in the derivation of the acceleration, velocity and displacement, i.e. the vibration, of the cutting tool. Vibration is selected as the analysed signal type based on the low expected cost of its acquisition and potential usefulness in the analysis of the machining process. The simulation tracks the position of each cutting tooth and the workpiece geometry produced by material removal (Figure 2) to identify which cutting teeth are performing the cut at each time step. The operation considered in this paper is a linear non-slotting

Table 1
Milling time-domain simulation parameters

Parameter type	Parameter	Value
Machining parameters	chip width b	0.004 to 0.005
	spindle speed ω	3000 to 4000
	feed rate f	10.2
Process-dependent parameters	number of cutting teeth N_t	3
	start angle of cut ϕ_s	126.9
	exit angle of cut ϕ_e	180
	process dependent coefficient K_s	2250e6
	force angle β	75
	x direction dynamics parameter k_x	9e6
	x direction dynamics parameter ζ_x	0.02
	y direction dynamics parameter k_y	1e7
	y direction dynamics parameter ζ_y	0.01
Simulation parameters	steps per revolution	256

milling cut performed with a straight-teeth cutting tool on a metal workpiece.

The physics-based model accepts several variables that control the deterministic simulation, including the machining parameters controllable during the configuration of the metal cutting process and the parameters dependent on the characteristics of the workpiece material, the machining tool and the manufactured product. These variables are detailed in Table 1 followed by the values used for generation of the training data. The parameter values are constant throughout each cutting operation, and the parameters varied across the samples in the produced dataset are chip width and spindle speed in ranges from 0.004 to 0.005 mm and 3000 to 4000 rpm respectively. The generated signals represent the displacement of the cutting tool along the x-direction during the third revolution of the cutting tool, sampled at a rate proportional to the spindle speed.

Algorithm 1 Physics-based simulation algorithm

1: **for** t in range(simulation time steps) **do**
2: $h(t) \leftarrow$ Instantaneous chip thickness
3: $F(h) \leftarrow$ Cutting force
4: $Vibration(F) \leftarrow$ Displacement values
5: $\phi \leftarrow \phi + d\phi$ (increment tooth angle)
6: **end for**

A signal sample is obtained for each combination of 200 linearly spaced chip width and 200 spindle speed parameter values in the specified ranges, resulting in 40 000 signal samples within the dataset. The only pre-processing applied to this data is the mean and standard deviation normalisation that is applied to each of the process parameters separately and to the time-domain signals. The validation dataset containing 40 000 samples is produced using the same approach, but with the process parameters values shifted a half of the step, i.e. chip width from 0.004025 to 0.005025 and spindle speed 3002.5 to 4002.5. Both training and validation datasets are publicly available at `https://gitlab.com/ezotov1/conditional-stylegan-digital-twin`.

2.2. Model Architecture

The digital twin component architecture discussed in this paper is inspired by StyleGAN [37], an image generation model based on two-dimensional deep convolutional networks with the enhancement of the generator by style-injection adopted from style transfer research works. Elements of StyleGAN are repurposed for the 1D case of a time-domain signal [84]. The noise inputs and the mixing regularisation (regularisation applied during training that randomly mixes the disentangled latent with another one to produce a sample from the generator G) are excluded from the model, as the variation of outputs of the target distribution is deterministic with respect to the input process parameters, i.e. the training data contains not more than a single sample for each unique label set. The architecture is enhanced with the substitution of the random input latent vector for continuous labels C: the machining process parameters, chip width and spindle speed. The process parameters are used as inputs to the generator and as outputs of the discriminator, i.e. the discriminator learns to not only identify synthetic data sam-

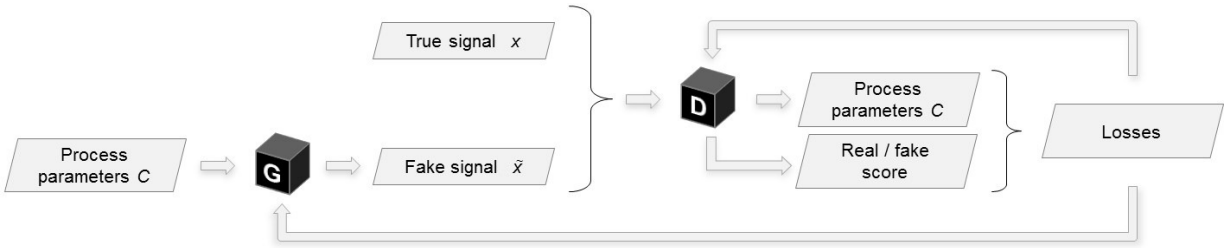


Fig. 3. Conditional GAN architecture. G denotes the generator network, D - the discriminator.

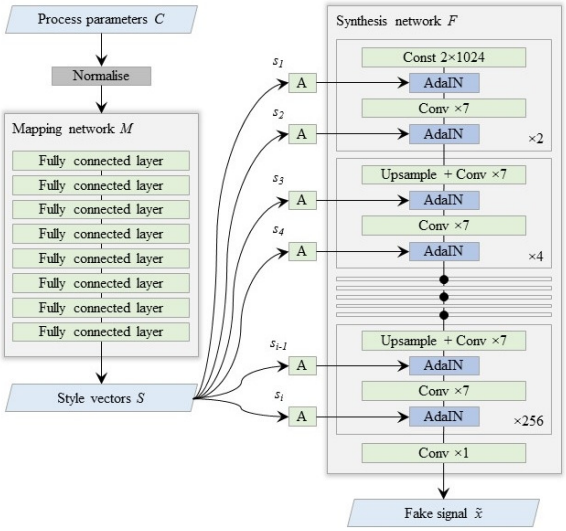


Fig. 4. Architecture of the GAN generator network. "A" denotes learned affine transformations of style components s_i; "AdaIN" - adaptive instance normalisation [31], outputs of which are modulated by the transformed style components.

ples, but also to estimate the labels associated with a given time-series, as visualised on Figure 3. The architecture includes a non-linear mapping network M that projects process parameter inputs into a disentangled latent space. The styles $S = M(C)$ produced from the input labels C by the mapping network subsequently control the modulation of outputs of the convolutional layers within the synthesis network F of the generator (see Figure 4).

The mapping network M is implemented as a multilayer perceptron and consists of 8 layers with 32 neurons each with leaky ReLU activation functions. M maps the input process parameters C into style vectors S of length 256. The first input to the synthesis network is a learned constant vector of dimension 2x1024, where the first dimension is the signal length and the second dimension is the number of filters. This learned constant is sequentially processed by multiple blocks each containing two convolutional layers with convolutional kernel of size 7. Except for the first block where the constant vector takes place of the first convolutional layer's output, the first convolutional layer in each block upscales the signal length by a factor of 2 and reduces the number of filters by a factor of 2 until the number of filters reaches 64. After each convolutional layer the signals are passed through a leaky ReLU activation function and then normalised and mixed with the respective style component vector within the AdaIN operation [31] defined as

$$\text{AdaIN}(x_f, s_i) = s_i^s \frac{x_f - \mu(x_f)}{\sigma(x_f)} + s_i^b, \tag{1}$$

where x_f is a filter response that is each normalised separately and s_i^s and s_i^b are the scaling and the bias components of the style vector at level i.

The last layer of the synthesis network F applies a convolution operation with kernel 1 to aggregate the 64 filter outputs produced by the final block into a single signal of length 256.

The GAN loss function is based on Wasserstein GAN with gradient penalty (WGAN-GP) [25]. WGAN-GP losses for the generator and the discriminator are

$$\begin{aligned} L_G^{wgan\text{-}gp} &= -\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})], \\ L_D^{wgan\text{-}gp} &= \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda_{gp} L^{gp} \end{aligned} \tag{2}$$

respectively, where

$$L^{gp} = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)^2]4 \tag{3}$$

is the gradient penalty, λ_{gp} is its scaling hyperparameter, D is the discriminator network, x and $\tilde{x}$ denote the real and fake signals respectively, $\mathbb{P}_r$ and $\mathbb{P}_g$ are the real and the generator signal distributions. $\mathbb{P}_{\tilde{x}}$ is a distribution sampled uniformly from straight lines be-

tween pairs of points from $\mathbb{P}_r$ and $\mathbb{P}_g$ [25]. The loss functions are adjusted to accommodate the inclusion of machining process parameters in the networks architecture by addition of terms that penalise inaccurate label predictions. This is similar to the approach followed by the authors of InfoGAN [10], with the following difference. The accuracy of label predictions for training data L_D^{info} impacts only the discriminator, while the accuracy of the predictions for fake data samples L_G^{info} is taken into account only by the generator. The loss terms are

$$\begin{aligned} L_G^{info} &= \sqrt{\frac{1}{n}\sum_{j=1}^{n}(c_{t,j} - \tilde{c}_{f,j})^2}, \\ L_D^{info} &= \sqrt{\frac{1}{n}\sum_{j=1}^{n}(c_{t,j} - \tilde{c}_{r,j})^2}, \end{aligned} \tag{4}$$

where $\tilde{c}_{f,j}$ is a value of parameter j predicted by the discriminator based on a fake signal, $\tilde{c}_{r,j}$ is a value predicted from a real signal, and $c_{t,j}$ are the true parameter values. On one hand, the generator is thus incentivised to encode the label information in an identifiable way within the synthesised samples. On the other hand, the discriminator learns the relationship between labels and samples only on the real data, thus preserving the non-cooperative nature of the minimax game between the generator and the discriminator. With λ_{info} denoting the scaling factor for the label prediction accuracy error loss, the total loss functions for the generator L_G and the discriminator L_D are therefore:

$$\begin{aligned} L_G &= L_G^{wgan\text{-}gp} + \lambda_{info} L_G^{info}, \\ L_D &= L_D^{wgan\text{-}gp} + \lambda_{info} L_D^{info}. \end{aligned} \tag{5}$$

The generator network learns the conditional distribution of the time-domain vibration signal with respect to the machining process parameters that control the basic milling conditions. The use of style-based architecture of the generator enables a reduction of a trained model's complexity via inspection of its style-level components. This and the degree of control provided by the conditional component of the generator increases the model's potential interpretability and enables a reductionist approach to the analysis of its black-box inner mechanisms.

The models are trained using the Adam optimiser for both the generator and the discriminator with learning rates of 0.00001 and 0.0001 respectively. The network losses are parametrised with $\lambda_{info} = 10$ and $\lambda_{gp} = 10$. The GAN described in this paper is trained until convergence, and the models used in correlation analysis described in Section 4.1 were trained for the same number of epochs as the main network, and thus have not necessarily converged. Training convergence was assessed based on the rate of improvement of the root mean square error (RMSE) metric measured and averaged across the validation dataset.

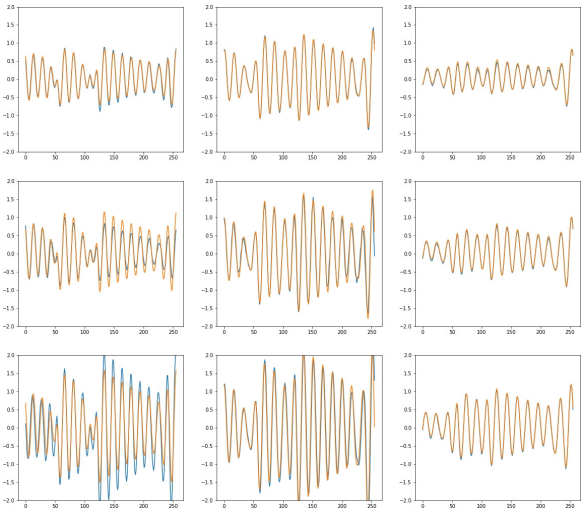


Fig. 5. Comparison of generated time-series samples and validation data samples. X-axis represents time steps, Y-axis - displacement in log scale. Real signal is represented by the yellow curves and the generated signal by blue.

3. Digital twin performance analysis

The paper shows that the generator successfully learns to capture the relationship between the process parameters and the time-domain signal and performs well both on training and validation data. Figure 5 depicts several samples of generated time-series against the signals from training data produced using the same process parameters.

The metrics used in the analysis of the generator network performance are aimed at capturing the accuracy of the model conditional on the input parameters. The potential mode collapse (i.e. the inability of the generator to produce parts of the target distribution) inevitably affects the accuracy of the generator due to the deterministic nature of the experimental data coupled with the conditional generation. Therefore, the discussed experimental setting permits less attention

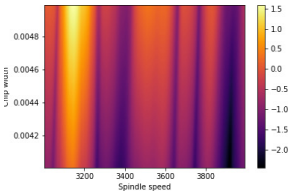


Fig. 6. Standard deviation of training data samples for each set of process parameters values (log scale).

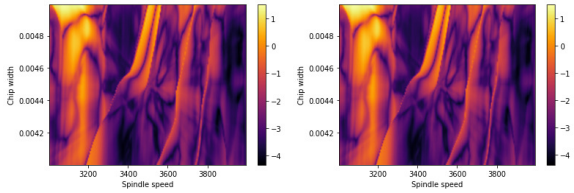


Fig. 7. Error $\mathcal{E}$ values in log scale for generated time-series across various process parameter values for training data (left) and validation data (right). A single point on the map represents the generator $G(C)$ error value $\mathcal{E}$ measured at process parameter values of chip width and spindle speed C corresponding to the Y-axis and the X-axis values respectively. The error pattern similarity on both maps indicates low over-fitting.

on the variety of the generated samples, and this paper focusses on the analysis of the generator accuracy. The true accuracy value for some labels C is measured by the root mean square error (RMSE) $\mathcal{E}(C)$ between the signal $x(C)$ from the training data and the signal $\tilde{x}(C) = G(C)$ produced by the generator:

$$\mathcal{E}(C) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i(C) - \tilde{x}_i(C))^2}, \tag{6}$$

where n is the signal length.

The generative performance of the neural network is investigated via analysis of metrics mapped across machining process parameter values, chip width and spindle speed. This is visualised by calculating the inspected metric for a range of the process parameter pairs and plotting the values on a two-dimensional figure with spindle speed varied across the horizontal axis and chip width across the vertical axis. The training data exhibits high variability, especially across the spindle speed parameter values range, visible on the plot of standard deviations of the training data signals (Figure 6). Figure 7 depicts the error $\mathcal{E}(C)$ of the generated time-series on training and on validation data. Error distributions are nearly identical, i.e. the generator performs equivalently during both training and validation. The validation dataset is of the same size as the training dataset, which is feasible due to the control over the source data synthesis.

In contrast to the high accuracy of the generator in the areas of the parameter space characterised by low dispersion in the training data, the generative performance is suboptimal in some regions of high training data variance. Closer inspection of a region between 3450 and 3550 spindle speed reveals that the generator experiences a local mode collapse at high chip widths (for an example refer to Figure 8). The trough visible on the error map in this region represent a parameter space where the generator successfully learnt the mode of the target signal, while the peaks to the sides along the X-axis from this narrow band of high accuracy are indicative of the dropped modes. An inspection of the dynamics of change of the training data signals compared to the change of the synthesised signals reveals that whereas the training data signals change shape linearly with the change in spindle speed, the generated signals remain constant for most of the inspected parameter space and sharply switch to the next mode near the peak on the error map.

4. Digital twin sensitivity analysis

4.1. Generator sensitivity

The metric discussed in the following sections is the sensitivity of the generator output to the input parameters δ:

$$\delta(C) = \frac{1}{m}\sum_{j=1}^{m}\left(\left|\frac{\partial\tilde{x}(C)}{\partial c_j}\right|\right), \tag{7}$$

where $c_j \in C$ and m is the number of input parameters in a label set C.

The generator sensitivity $\delta(C)$ provides an insight into the origin of the parameter space regions where the GAN produces samples with relatively high inaccuracy. Empirical observation of the sensitivity maps obtained from several trained GANs reveals a complex but stably recurring relationship between the distributions of the generator sensitivity $\delta(C)$ and the generator error $\mathcal{E}(C)$ across the parameter space. Comparison of the correlation between the point-wise sensitivity and error metrics for different generator networks implies a moderately strong connection between these metrics that is nevertheless consistent across the generators, with average correlation of 0.45 for 52 different generator networks. Examples of this comparison are

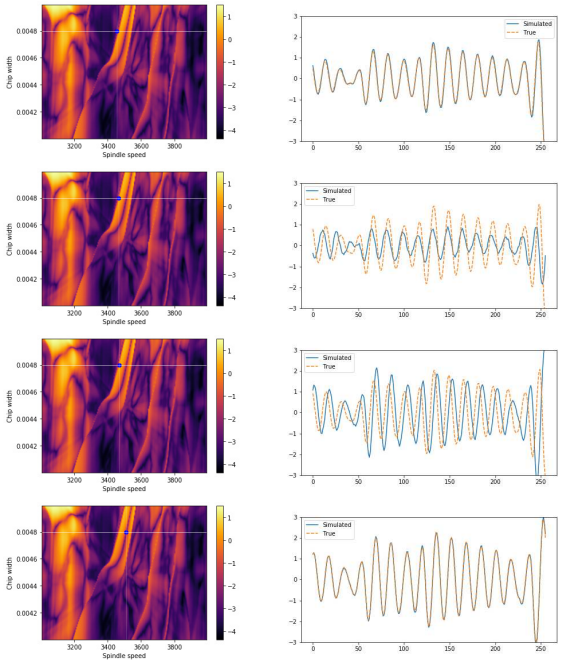


Fig. 8. Example of a path in labels space where the transition between the signal modes is smooth in the training data, but abrupt in the signals produced by the generator. The blue dot on the error maps (left) indicates the parameter values used to compare the two signals (right), real signal in yellow and generated signal in blue. The differences between the fake signals along the labels transition paths shown on the two top figures and the two bottom figures are much lower than for the real signal. The opposite is true for the transition captured by the two middle figures.

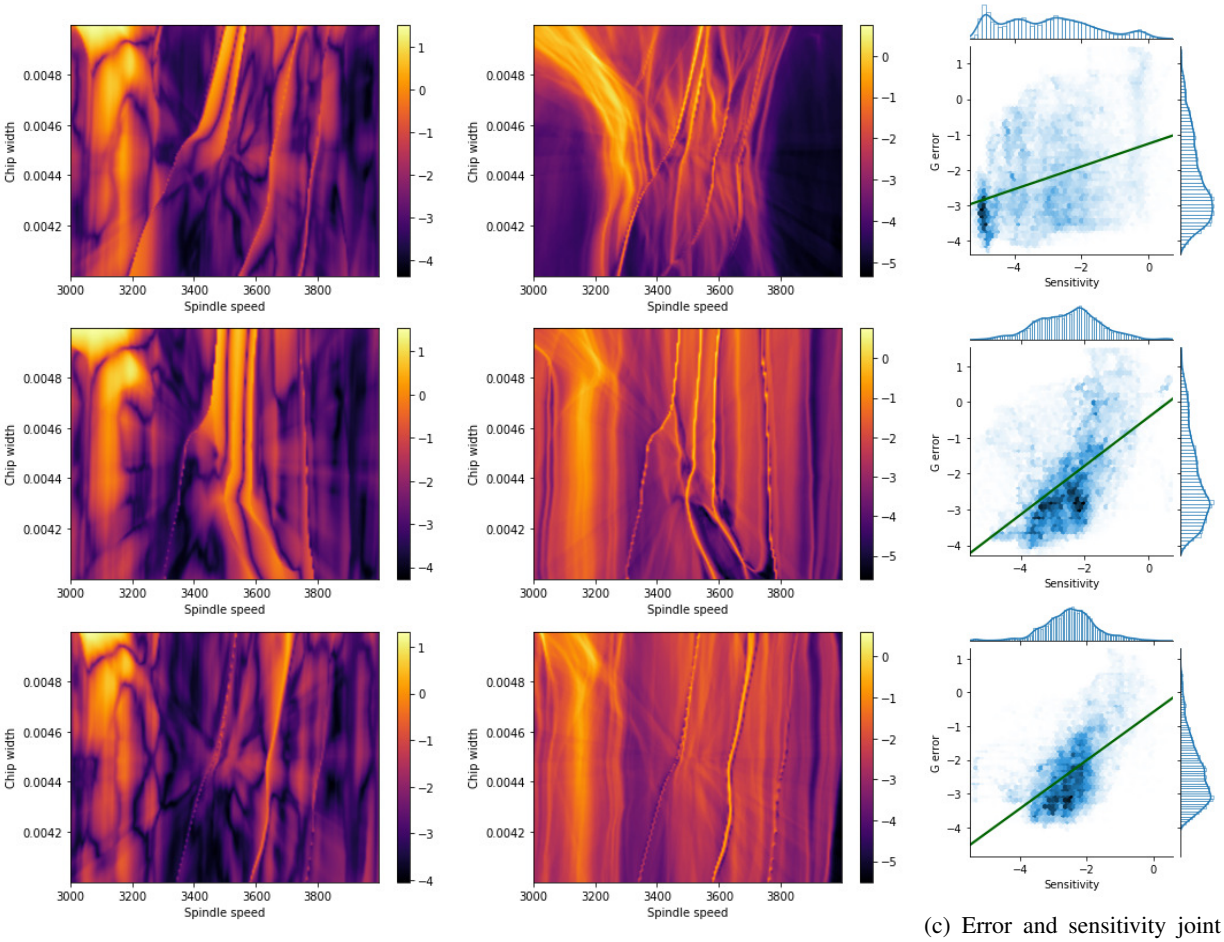
presented on Figure 9. Both the empirical observations and the correlation analysis imply that a complex, potentially non-linear, relationship between the two metrics exists. If mapped onto the generator accuracy, the generator sensitivity $\delta(C)$ can be used as an accuracy estimator. Thus the sensitivity can represent the generation uncertainty at some input parameter values using only the information obtained from the generator model itself. Calculation of $\delta(C)$ requires generation of an additional sample from the generator per parameter in C for each evaluated signal, which yields a negligible additional computational cost for GANs implemented in most modern neural network libraries that allow batch evaluation of the models. Generator sensitivity can thus be efficiently used at inference time to enhance the produced data samples with an accuracy uncertainty metric without requiring any external data and additional measurements.

The discussed relationship between the generative accuracy and sensitivity to input labels implies that the cause of the modes of failure of the generator originates from the parameters inputs and consequently from their style representation. This link is studied in the following section.

4.2. Interpolation Analysis

The style-based neural network architecture enables a reduction of the model analysis complexity via the inspection of the influence of the disentangled input parameter vectors s_i at the different layers i of the synthesis network $F(S)$ within the GAN, where $S = \{s_i\} = M(C)$ is a set of disentangled style vectors produced by the mapping network M of the generator from the input label vectors C. This is performed via the analysis of the changes in the generated output signals arising from alteration of the disentangled inputs. Two style sets are obtained from the mapping network using different input parameters, $S^o = M(C^o)$ and $S^t = M(C^t)$ for $C^o \neq C^t$, and two signals are generated using these parameter sets, $\tilde{x}^o = F(S^o)$ and $\tilde{x}^t = F(S^t)$. An interpolated signal $\tilde{x}^c = F(S^c)$ is calculated by feeding an affine combination of style sets S^o and S^t into the synthesis network, $S^c = \{s_i^c\} = \{s_i^o * w_i + s_i^t * (1 - w_i)\}$, where i is the index denoting the style level increasing towards the output layer of the model. Thus, at each style level i the style component of S^c is the weighted sum of the components of S^o and S^t at this level, with the weight $w_i = 0$ indicating that only the component of the first style set S^o is used and $w_i = 1$ implying that only the second style S^t is applied at this layer.

The variation of the generated signals produced as a result of gradual changes to one or several style components reveals the features that the style components at the respective levels control. By performing a linear interpolation between s_i^o and s_i^t for each $i = k$ individually, i.e. while keeping $s_i^c = s_i^o$ for each $i \neq k$, we observe that the generator model style layers can be classified into two groups: high-level styles S_H and low-level styles S_L, where $S_H = \{s_{1..5}\}$ and $S_L = \{s_{6..16}\}$. The high-level styles significantly affect the low-frequency features of the output signal like its phase and general envelope shape. The low-level styles impact the high-frequency detail of the generated output. Figure 10 visualises the end points of this interpolation: an initial signal $\tilde{x}^o$, its interpolation towards $\tilde{x}^t$ in high-level styles only (i.e. $\tilde{x}^c = M(S_H^t \cup S_L^o)$) on the top plot and in low-level styles only (i.e. $\tilde{x}^c = M(S_H^o \cup S_L^t)$) on the bottom one.



(a) Error $\mathcal{E}$ (b) Sensitivity δ (c) Error and sensitivity joint distribution

Fig. 9. Error $\mathcal{E}$ and sensitivity δ maps for three trained models, a model's plots on each row. The right column contains plots of the joint distributions of these two measures, with the shaded regions representing the frequency of samples for the corresponding $\mathcal{E}$ and δ values and the solid lines are the fitted linear regressions.

An analogous inspection of the style-level interpolation effects is performed at the layer level of the generator, i.e. the variation of the activations produced by the intermediary convolutional layers of the model is observed as a result of changes in the style parameters. Empirical analysis reveals that the sensitivity of the layer activations to the changes in the high-level style components exhibits the irregular non-smoothness similar to the sharp gradients notable in $\delta(C)$. The search for the origin of the error-producing high sensitivity discussed in section 4.1 can thus be narrowed to the layers of the GAN where the high-level styles are injected. The impact of the style modulation measured at a single-layer level seems insignificant, as evidenced by the minor difference between the distributions of the input and output activations of the style injection layers. Example of this is depicted on Figure 11. Nevertheless, these minor changes are propagated through the generator network and magnified by the complex black box processing within the downstream convolutional layers, resulting in major variability of the network outputs.

4.3. Activation node sensitivity

One of the possible ways of further reduction of the complexity of the analysed chaotic system is an additional refinement of the analysed object's scale. Thus, the sensitivity of individual activation nodes within the generator layers relative to the sensitivities of other nodes in the same layer is of interest. This sensitivity δ_{node}^{layer} is measured as the sensitivity of the activation

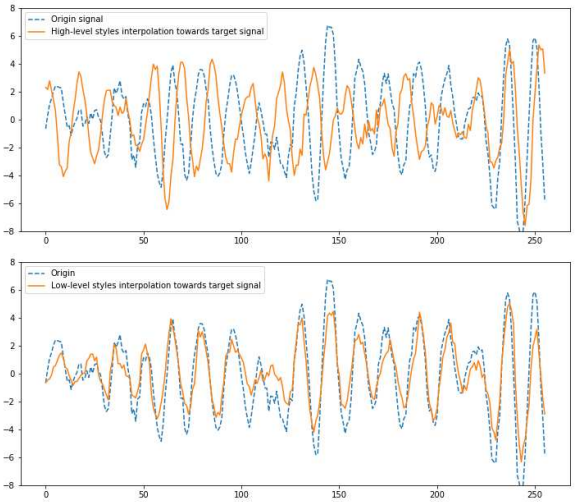


Fig. 10. Comparison of the interpolations over high- and low-level styles on the top and the bottom figures respectively. The dashed line represents the initial signal $\tilde{x}^o = F(S^o)$ on both plots. Top solid line, the signal generated using interpolated styles $S^c = \{s^t_{1..5}\} \cup \{s^o_{6..16}\}$, is significantly different from $\tilde{x}^o$ in its phase and modes. Bottom solid line, the signal produced from $S^c = \{s^o_{1..5}\} \cup \{s^t_{6..16}\}$, differs from the source in its local high--frequency features.

outputs to the changes in input process parameters averaged across a set of process parameter values $\mathcal{C}$.

$$\delta^{layer}_{node} = \left| \frac{1}{\sigma k} \sum_{p=1}^{k} (\delta'(C_p)) \right|, \text{ where}$$
$$\delta'(C_p) = \frac{1}{m} \sum_{j=1}^{m} \left(\frac{\partial z^{layer}_{node}(C_p)}{\partial c_j} \right), \tag{8}$$

$z^{layer}_{node}(C_p)$ is the activation output of a single node within a layer, δ' is the sensitivity of this activation to change in the process parameter input measured at process parameter values C_p and σ is the standard deviation of δ' across the process parameter values.

Therefore, δ^{layer}_{node} is high for the activation nodes that are highly sensitive to the process parameters C robustly across the parameter space. Comparison of δ^{layer}_{node} calculated using the complete process parameter set from training data with δ^{layer}_{node} based on a limited parameter set containing only the parameter values that produce the top 5% of generator sensitivity $\delta(C)$ is presented on Figure 12. Most of the nodes producing high sensitivity in regions where the whole generator network is highly sensitive to input are also producing relatively high sensitivity in the process parameter regions not characterised by high generator sensitivity. This metric thus makes possible the identification of the most sensitive activation nodes within the network and consequently the convolutional kernels that produce this sensitivity, which can potentially be used for remedying the anomalously high sensitivity $\delta(C)$ of the generator and reducing the errors in the respective regions.

5. Discussion

The conditional neural network architecture described in the paper allows a significant degree of control over the generator via controllable input process parameters, thus enhancing the flexibility of the model and creating the potential for exploratory analysis of the modelled process. Such analysis could serve the purpose of CNC machining process planning and optimisation. Researchers have shown how signal data can be utilised to predict product conformity to quality standards [28,54], therefore enabling the prediction of manufacturing errors and product quality prior to manufacturing when coupled with signal simulation proposed in this paper. Additionally, the machining process stability estimation can be enhanced with a generative model substitution for some of the physical measurements, which is a future research direction also proposed by machining stability experts [23].

In a real-world scenario a physics-based model would be the same for a fleet of machines, but each machine would be operated in different conditions and have slightly varying characteristics. The calibration of such physics-based models is time-consuming and difficult to perform on a large scale. Therefore, while a physics-based model might act as a digital twin simulation component, it would inevitably have made simplifications and idealisations about the process, likely omitting individual variations of environmental and dynamic factors that influence the manufacturing process due to their modelling complexity or computational cost. These complex phenomena are nevertheless reflected in the real process data and can thus be captured via data-driven training of a GAN model. The physics-based model used in the presented paper acts to prepare a surrogate for the real-world data that a digital twin is likely to have access to, as well as a mechanism to incorporate the knowledge contained within the physics-based model into the data-driven modelling process.

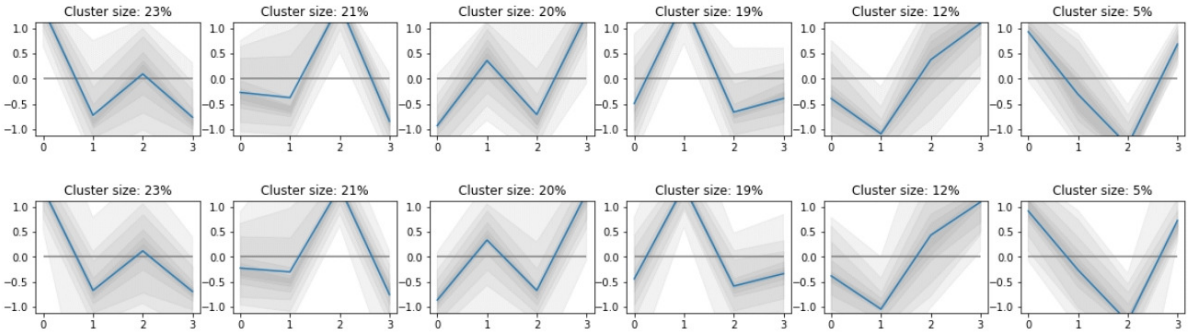


Fig. 11. Distributions of activation clusters that are the input (top plot) and the output (bottom plot) of a high-level style modulation layer of the generator network. Solid line is the cluster activations' mean and the shaded areas show the cluster activations' distribution covering the whole range of activation values in a cluster. Style injection at this layer changes neither the cluster sizes nor the activation distribution significantly.

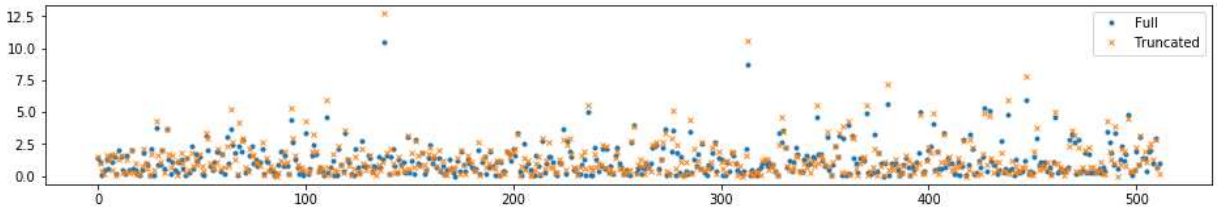


Fig. 12. Distribution of the activation node sensitivity metric δ_{node}^{layer} for one of the generator layers with values of δ_{node}^{layer} along the Y-axis and node indices along the X-axis. Crosses denote the sensitivity values measured based on the process parameters values from the regions where the whole generator network is highly sensitive to input. The δ_{node}^{layer} calculated on the full process parameter set are marked by the filled dots.

The validation of the proposed GAN model using an almost raw vibration signal data, pre-processed only by domain-agnostic mean and standard deviation normalisation, implies that the findings discussed in this paper are generalisable and relevant for other time-domain signal generation applications. The neural network architecture at the base of the proposed model is computationally cheap at inference time. This and the generative nature of GAN enable the development of a machining digital twin component that simulates the underlying physical process in real-time, which is an important step towards implementation of data-driven simulation models in the development of digital twins for Industry 4.0. The proposed analysis methodology utilising the style-based input disentanglement enables a reductionist approach to the model's performance analysis, and the generative uncertainty metrics based on the generator sensitivity presented in this paper increase the model's transparency and interpretability. Both are important barriers to the widespread adoption of complex models in the industrial context [32].

The conditional component of the proposed GAN is naturally extensible to simulation of longer signals via auto-regressive model based generation, i.e. the inclusion of signal history as a conditioning input to the generator. The use of recurrent neural network layers might also prove beneficial for this extension of the generator network. Conditioning can additionally be applied for adoption of the proposed model to a signal transformation problem, e.g. for prediction of cutting forces from the vibration signals, similar to the image style transfer idea [31].

A real world implementation would likely be limited in terms of available data due to the relatively high cost of acquisition of non-production experimental data. With the lack of data for creation of a complete map across the various operating conditions, the approach would have to be implemented using a limited subset of these conditions. An attentive consideration of the sampling efficiency of true data would thus be important in the optimisation of the model's ability to learn with less data. This could also be remedied by a hybrid data generation approach that assembles the training and validation datasets using any existing expert- or physics-based models complemented with the real world data.

6. Conclusion

This paper presents the first application of a style-based GAN for machining process simulation. The work uses both the style and conditional components of the GAN architecture for attainment of deeper insights into the generative function of the GAN. A novel sensitivity analysis approach for conditional GANs is presented and applied to the proposed model, establishing a link between the true accuracy and the label sensitivity of the generator network. The style component of the GAN architecture is utilised to further advance this analysis to the level of intermediate neural network layers, resulting in the identification of the activation nodes that produce abnormal generative behaviour.

Being one of the first few papers in the manufacturing domain that focus on the generator of GAN, as opposed to the purely support role of GAN considered in most relevant works in the field, this work aims to promote the use of GANs in manufacturing past simple data augmentation for imbalanced datasets towards more sophisticated simulation applicable to a wider range of use cases.

The current work considers a conditional GAN with only continuous inputs, but an adaptation of the described sensitivity analysis method to the conditional GAN models with categorical inputs would make the proposed technique universally applicable in GAN performance analysis. Altering the neural network architecture could also make further insights available by further shifting the complexity of the multi-layer convolutional interactions to other operations, such as via skip connections [30] or residual connections [27].

Future research directions include comparative evaluation of different neural network architectures and validation of the proposed model on real manufacturing data, as well as broadening of the scope of the digital twin simulation with inclusion of multiple data sources and simultaneously modelled processes.

7. Acknowledgements

Professor Kadirkamanathan acknowledges the support from EPSRC funded Advanced Metrology Hub project (Reference EP\P006930\1).

Professor Tiwari acknowledges the support of the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme (RCSRF1718\5\41).

References

[1] S. M. Afazov, S. M. Ratchev, and J. Segal. Modelling and simulation of micro-milling cutting forces. *Journal of Materials Processing Technology*, 210(15):2154–2162, 2010. ISSN 09240136. doi: 10.1016/j.jmatprotec.2010.07.033.

[2] Divyakant Agrawal, Sudipto Das, and Amr El Abbadi. Big data and cloud computing. In *Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ICDT '11*, page 530, New York, USA, 2011. ACM Press. ISBN 9781450305280. doi: 10.1145/1951365.1951432.

[3] Mohamed Baker Alawieh, Yibo Lin, Zaiwei Zhang, Meng Li, Qixing Huang, and David Z. Pan. GAN-SRAF: Sub-resolution assist feature generation using conditional generative adversarial networks. *Proceedings - Design Automation Conference*, (i):1–6, 2019. ISSN 0738100X.

[4] Y. Altintas and M. Weck. Chatter stability of metal cutting and grinding. *CIRP Annals - Manufacturing Technology*, 53(2):619–642, 2004. ISSN 00078506. doi: 10.1016/S0007-8506(07)60032-8.

[5] Y. Altintas, P. Kersting, D. Biermann, E. Budak, B. Denkena, and I. Lazoglu. Virtual process systems for part machining operations. *CIRP Annals - Manufacturing Technology*, 63(2): 585–605, 2014. ISSN 17260604. doi: 10.1016/j.cirp.2014.05.007.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN, 2017. ISSN 1701.07875. URL `http://arxiv.org/abs/1701.07875`.

[7] Manas Bajaj, Dirk Zwemer, and Bjorn Cole. Architecture to geometry - Integrating system models with mechanical design. *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, (September):1–19, 2016. doi: 10.2514/6.2016-5470.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. sep 2018. URL `http://arxiv.org/abs/1809.11096`.

[9] Marc L. Campomanes and Yusuf Altintas. An improved time domain simulation for dynamic milling at small radial immersions. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 125(3):416–422, 2003. ISSN 10871357. doi: 10.1115/1.1580852.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, jun 2016. ISSN 978-3-319-16807-4. URL `http://arxiv.org/abs/1606.03657`.

[11] Yize Chen, Yishen Wang, Daniel Kirschen, and Baosen Zhang. Model-Free Renewable Scenario Generation Using Generative Adversarial Networks. *IEEE Transactions on Power Systems*, 33(3):3265–3275, may 2018. ISSN 0885-8950.

[12] Harish Cherukuri, Elena Perez-Bernabeu, Miguel Selles, and Tony Schmitz. Machining chatter prediction using a data learning model. *Journal of Manufacturing and Materials Processing*, 3(2), 2019. ISSN 25044494. doi: 10.3390/jmmp3020045.

[13] Clayton Cooper, Jianjing Zhang, Robert X. Gao, Peng Wang, and Ihab Ragai. Anomaly detection in milling tools using acoustic signals and generative adversarial networks. *Procedia Manufacturing*, 48(2019):372–378, 2020. ISSN 23519789. doi: 10.1016/j.promfg.2020.05.059.

[14] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial Audio Synthesis. 2018. ISSN 14341948. URL

http://arxiv.org/abs/1802.04208.

[15] Waguih Elmaraghy, Hoda Elmaraghy, Tetsuo Tomiyama, and Laszlo Monostori. Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology*, 61(2): 793–814, 2012. ISSN 00078506. doi: 10.1016/j.cirp.2012.05.001.

[16] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–17, 2019.

[17] EPSRC Future Metrology Hub. UK Metrology Research Roadmap. Technical report, 2020.

[18] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs, jun 2017. URL http://arxiv.org/abs/1706.02633.

[19] A. B. Forbes. Uncertainty associated with form assessment in coordinate metrology. *International Journal of Metrology and Quality Engineering*, 4(1):17–22, 2013. ISSN 21076839. doi: 10.1051/ijmqe/2012032.

[20] Jens Friedrich, Jonas Torzewski, and Alexander Verl. Online Learning of Stability Lobe Diagrams in Milling. *Procedia CIRP*, 67:278–283, 2018. ISSN 22128271. doi: 10.1016/j.procir.2017.12.213.

[21] Yang Gao, Rita Singh, and Bhiksha Raj. Voice Impersonation Using Generative Adversarial Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:2506–2510, 2018. ISSN 15206149.

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, 2014. ISSN 10495258. URL http://arxiv.org/abs/1406.2661.

[23] Noel P Greis, Monica L Nogueira, Sambit Bhattacharya, and Tony Schmitz. Physics-Guided Machine Learning for Self-Aware Machining. In *2020 AAAI Spring Symposium on AI and Manufacturing*, 2020.

[24] Michael Grieves and John Vickers. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, pages 85–113. Springer International Publishing, 2017. ISBN 978-3-319-38756-7.

[25] Ishaan; Gulrajani, Faruk; Ahmed, Martin; Arjovsky, Vincent; Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs, mar 2017. URL http://arxiv.org/abs/1704.00028.

[26] Te Han, Chao Liu, Wenguang Yang, and Dongxiang Jiang. A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. *Knowledge-Based Systems*, 165:474–487, 2019. ISSN 09507051.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Multimedia Tools and Applications*, pages 1–17, dec 2015. ISSN 15737721. URL http://arxiv.org/abs/1512.03385.

[28] Q. Peter He and Jin Wang. Statistical process monitoring as a big data analytics tool for smart manufacturing. *Journal of Process Control*, 67:35–43, 2018. ISSN 09591524. doi: 10.1016/j.jprocont.2017.06.012.

[29] Kagermann Henning, Wahlster Wolfgang, and Helbig Johannes. Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Technical Report April, 2013.

[30] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:2261–2269, 2017. ISSN 0022-4790.

[31] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. mar 2017. URL http://arxiv.org/abs/1703.06868.

[32] Ahmad Issa, Dominik Lucke, and Thomas Bauernhansl. Mobilizing SMEs Towards Industrie 4.0-enabled Smart Products. *Procedia CIRP*, 63:670–674, 2017. ISSN 22128271. doi: 10.1016/j.procir.2017.03.346.

[33] Jinyang Jiao, Ming Zhao, Jing Lin, and Kaixuan Liang. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing*, 417:36–63, 2020. ISSN 18728286. doi: 10.1016/j.neucom.2020.07.088.

[34] M I Jordan and T M Mitchell. Machine learning: Trends, perspectives, and prospects. 349(6245), 2015.

[35] Lauri Juvela, Bajibabu Bollepalli, Xin Wang, Hirokazu Kameoka, Manu Airaksinen, Junichi Yamagishi, and Paavo Alku. Speech Waveform Synthesis from MFCC Sequences with Generative Adversarial Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:5679–5683, 2018. ISSN 15206149.

[36] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017. ISSN 10450823. URL http://arxiv.org/abs/1710.10196.

[37] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. dec 2018. URL http://arxiv.org/abs/1812.04948.

[38] Dong Hyeon Kim, Thomas J.Y. Kim, Xinlin Wang, Mincheol Kim, Ying Jun Quan, Jin Woo Oh, Soo Hong Min, Hyungjung Kim, Binayak Bhandari, Insoon Yang, and Sung Hoon Ahn. Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 5(4):555–568, 2018. ISSN 21980810. doi: 10.1007/s40684-018-0057-y.

[39] Thomas Konrad, Lutz Lohmann, and Dirk Abell. Surface defect detection for automated inspection systems using convolutional neural networks. *27th Mediterranean Conference on Control and Automation, MED 2019 - Proceedings*, pages 75–80, 2019.

[40] Andrew Kusiak. Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, 0(0):1–11, 2019. ISSN 1366588X. URL https://doi.org/00207543.2019.1662133.

[41] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 14764687.

[42] Yingguang Li, Changqing Liu, James X. Gao, and Weiming Shen. An integrated feature-based dynamic control system for on-line machining, inspection and monitoring. *Integrated Computer-Aided Engineering*, 22(2):187–200, 2015. ISSN 18758835. doi: 10.3233/ICA-150483.

[43] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets, nov 2014. URL http://arxiv.org/abs/1411.1784.

[44] Olof Mogren. C-RNN-GAN: Continuous recurrent neural networks with adversarial training, 2016. URL http://arxiv.org/abs/1611.09904.

[45] László Monostori. AI and machine learning techniques for managing complexity, changes and uncertainties in manufacturing. *Engineering Applications of Artificial Intelligence*, 16(4):277–291, 2003. ISSN 09521976. doi: 10.1016/S0952-1976(03)00078-2.

[46] Egon Mueller, Xiao Li Chen, and Ralph Riedel. Challenges and Requirements for the Application of Industry 4.0: A Special Insight with the Usage of Cyber-Physical System. *Chinese Journal of Mechanical Engineering (English Edition)*, 30(5):1050–1057, 2017. ISSN 21928258. doi: 10.1007/s10033-017-0164-7.

[47] Elisa Negri, Luca Fumagalli, and Marco Macchi. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*, 11(June):939–948, 2017. ISSN 23519789. doi: 10.1016/j.promfg.2017.07.198.

[48] Weili Nie, Nina Narodytska, and Ankit Patel. RelGAN: Relational Generative Adversarial Networks for Text Generation. In *ICLR*, pages 1–20, 2019.

[49] Oliver Niggemann, Gautam Biswas, John S. Kinnebrew, Hamed Khorasgani, Sören Volgmann, and Andreas Bunte. Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control. *CEUR Workshop Proceedings*, 1507:185–192, 2015. ISSN 16130073.

[50] Ibone Oleaga, Carlos Pardo, Juan J. Zulaika, and Andres Bustillo. A machine-learning based solution for chatter prediction in heavy-duty milling machines. *Measurement: Journal of the International Measurement Confederation*, 128(May):34–44, 2018. ISSN 02632241. doi: 10.1016/j.measurement.2018.06.028.

[51] Tuğrul Özel and Taylan Altan. Process simulation using finite element method — prediction of cutting forces, tool stresses and temperatures in high-speed flat end milling. *International Journal of Machine Tools and Manufacture*, 40(5):713–738, apr 2000. ISSN 08906955. doi: 10.1016/S0890-6955(99)00080-2.

[52] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. ISSN 10414347. doi: 10.1109/TKDE.2009.191.

[53] Moschos Papananias, Thomas E. McLeay, Mahdi Mahfouf, and Visakan Kadirkamanathan. A Bayesian framework to estimate part quality and associated uncertainties in multistage manufacturing. *Computers in Industry*, 105(February):35–47, 2019. ISSN 01663615. doi: 10.1016/j.compind.2018.10.008.

[54] Moschos Papananias, Thomas E. McLeay, Mahdi Mahfouf, and Visakan Kadirkamanathan. An intelligent metrology informatics system based on neural networks for multistage manufacturing processes. *Procedia CIRP*, 82(June):444–449, 2019. ISSN 22128271. doi: 10.1016/j.procir.2019.04.148.

[55] D. T. Pham and A. A. Afify. Machine-learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(5):395–412, 2005. ISSN 09544054. doi: 10.1243/095440505X32274.

[56] Xinbo Qi, Guofeng Chen, Yong Li, Xuan Cheng, and Changpeng Li. Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives. *Engineering*, 5(4):721–729, 2019. ISSN 20958099. doi: 10.1016/j.eng.2019.04.012. URL https://doi.org/10.1016/j.eng.2019.04.012.

[57] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015. ISSN 0004-6361. URL http://arxiv.org/abs/1511.06434.

[58] Francisco J. Rodriguez Lera, Francisco Martín Rico, and Vicente Matellán Olivera. Neural networks for recognizing human activities in home-like environments. *Integrated Computer-Aided Engineering*, 26(1):37–47, 2018. ISSN 18758835. doi: 10.3233/ICA-180587.

[59] Yunus Saatchi and Andrew Gordon Wilson. Bayesian GAN, 2017. ISSN 10495258. URL http://arxiv.org/abs/1705.09558.

[60] Tony L. Schmitz and K. Scott Smith. *Machining Dynamics*. Springer International Publishing, Cham, 2019. ISBN 978-3-319-93706-9.

[61] Mike Shafto, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne, and Lui Wang. DRAFT Modeling, Simulation, information Technology & Processing Roadmap - Technology Area 11. *National Aeronautics and Space Administration*, page 27, 2010.

[62] Nagaraja Shetty, S. M. Shahabaz, S. S. Sharma, and S. Divakara Shetty. A review on finite element method for machining of composite materials. *Composite Structures*, 176:790–802, 2017. ISSN 02638223. doi: 10.1016/j.compstruct.2017.06.012.

[63] S. A. Shevchik, C. Kenel, C. Leinenbach, and K. Wasmer. Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks. *Additive Manufacturing*, 21:598–604, 2018. ISSN 22148604.

[64] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, oct 2017. ISSN 0028-0836.

[65] S. Smith and J. Tlusty. Overview of modeling and simulation of the milling process. *Journal of engineering for industry*, 113 (2):169–175, 1991. ISSN 00220817. doi: 10.1115/1.2899674.

[66] Adrian Spurr, Emre Aksan, and Otmar Hilliges. Guiding InfoGAN with Semi-supervision. In Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 119–134. Springer International Publishing, Cham, jul 2017. ISBN 978-3-319-71248-2.

[67] Ren Kai Tan, Nevin L. Zhang, and Wenjing Ye. A deep learning–based method for the design of microstructural materials. *Structural and Multidisciplinary Optimization*, pages 1–22, 2019. ISSN 1615-147X.

[68] Fei Tao, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. Digital twin-driven product design, manufacturing and service with big data. *International Journal of Advanced Manufacturing Technology*, 94(9-12):3563–3576, 2018. ISSN 14333015.

[69] Alfred Theorin, Kristofer Bengtsson, Julien Provost, Michael Lieder, Charlotta Johnsson, Thomas Lundholm, and Bengt Lennartson. An event-driven manufacturing information sys-

tem architecture for Industry 4.0. *International Journal of Production Research*, 55(5):1297–1311, 2017. ISSN 1366588X. doi: 10.1080/00207543.2016.1201604.

[70] Thanongsak Thepsonthi and Tuğrul Özel. 3-D finite element process simulation of micro-end milling Ti-6Al-4V titanium alloy: Experimental validations on chip flow and tool wear. *Journal of Materials Processing Technology*, 221:128–145, 2015. ISSN 09240136. doi: 10.1016/j.jmatprotec.2015.02.019.

[71] Khaoula Tidriri, Nizar Chatti, Sylvain Verron, and Teodor Tiplica. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, 42: 63–81, 2016. ISSN 13675788. doi: 10.1016/j.arcontrol.2016.09.008.

[72] Huan-Hsin Tseng, Yi Luo, Sunan Cui, Jen-Tzung Chien, Randall K. Ten Haken, and Issam El Naqa. Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical Physics*, 44(12):6690–6705, dec 2017. ISSN 00942405.

[73] F. J. Vera-Olmos, E. Pardo, H. Melero, and N. Malpica. DeepEye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering*, 26(1): 85–95, 2018. ISSN 18758835. doi: 10.3233/ICA-180584.

[74] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X. Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48: 144–156, 2018. ISSN 02786125.

[75] Junliang Wang, Zhengliang Yang, Jie Zhang, Qihua Zhang, and Wei Ting Kary Chien. AdaBalGAN: An Improved Generative Adversarial Network with Imbalanced Learning for Wafer Defective Pattern Recognition. *IEEE Transactions on Semiconductor Manufacturing*, 32(3):310–319, 2019. ISSN 15582345.

[76] Yanxia Wang, Kang Li, Shaojun Gan, Che Cameron, and Min Zheng. Data augmentation for intelligent manufacturing with generative adversarial framework. *1st International Conference on Industrial Artificial Intelligence, IAI 2019*, pages 1–6, 2019.

[77] Zirui Wang, Jun Wang, and Youren Wang. An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition. *Neurocomputing*, 310:213–222, 2018. ISSN 18728286.

[78] Long Wen, Xinyu Li, Liang Gao, and Yuyan Zhang. A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method. *IEEE Transactions on Industrial Electronics*, 65 (7):5990–5998, 2018. ISSN 02780046.

[79] R. G. Wilhelm, R. Hocken, and H. Schwenke. Task specific uncertainty in coordinate measurement. *CIRP Annals - Manufacturing Technology*, 50(2):553–563, 2001. ISSN 00078506. doi: 10.1016/S0007-8506(07)62995-3.

[80] Tao Yang, Cindy Cappelle, Yassine Ruichek, and Mohammed El Bagdouri. Multi-object tracking with discriminant correlation filter based deep learning tracker. *Integrated Computer-Aided Engineering*, 26(3):273–284, 2019. ISSN 18758835. doi: 10.3233/ICA-180596.

[81] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, sep 2016. ISSN 2168-6106.

[82] Han Zhang, Tao Xu, and Hongsheng Li. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, volume 2017-Octob, pages 5908–5916. IEEE, oct 2017. ISBN 978-1-5386-1032-9.

[83] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019. ISSN 10961216.

[84] Evgeny Zotov, Ashutosh Tiwari, and Visakan Kadirkamanathan. Towards a digital twin with generative adversarial network modelling of machining vibration. In Lazaros Iliadis, Plamen Parvanov Angelov, Chrisina Jayne, and Elias Pimenidis, editors, *Proceedings of the 21st EANN (Engineering Applications of Neural Networks) 2020 Conference*, pages 190–201, Cham, 2020. Springer International Publishing. ISBN 978-3-030-48791-1.