POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

On-site process monitoring and autonomous control of a CNC milling machine using CNN and Bayesian Optimization

Laurea Magistrale in Mechanical Engineering - Ingegneria Meccanica

Author: Filippo Danilo Michelacci

Advisor: Professor Annoni Massimiliano Pietro Giovanni

Co-advisor:

Academic year: 2021-2022

1. Introduction

CNC milling machines are widely used in nowadays manufacturing industries due to their capability of cutting different materials with high accuracy. However, unintended undulation with high surface roughness can often be found in the machined surfaces which requires aggressive finishing and therefore deteriorates the dimensional accuracy. To maintain certain process quality, the machining parameters should be appropriately chosen at priori, yet current approaches mostly rely on a trial and error strategy with many attempts in the neighbour of the tool's machining tables. In this work, we implemented an on-site monitoring system to a 5-axis CNC machine including built-in optical microscopes and convolution neural network (CNN) architectures to directly inspect the surface quality. Multi-objective Bayesian optimization was adopted to find the proper feed and speed for a trochoidal slotting operation at fixed depths of cut and with few attempts. We can also change the machining parameters after a single pass when the machining results were not as expected; testing the model's performance as an online optimiser directly connected to the monitoring system. The on-machine vision system, if further improved, can be extended to industrial-scale CNC machines toward realization of autonomous and automated systems even in the small to medium companies' reality.

2. Surface Roughness prediction

In this work the *Arithmetical mean height of the assessed surface*, S_a was chosen as the metric to evaluate the final quality of the machined surfaces. This expresses more information than R_a because it considered along two dimensions instead of a single one [1].

$$S_a = \frac{1}{A}\int\int_A |z(x,y)|dxdy \qquad [\mu m] \quad (1)$$

Such metric can be easily measured using optical machines that based their estimations on the reflection of a scattered light projected on the surface of the sample. However, this method require time and the use of a machine outside from the CNC environment forcing the operators to move the machined pieces from one station to another increasing the overall production time. Moreover, this measurement can only be performed once the operation is over and there is no possibility of controlling the machining process

while it is cutting. This affects the production yield because corrections are hard to make on already cut pieces.

2.1. Training of the models

To predict the surface roughness, various Convolutional Neural Network (CNN) architectures were trained to find the most suitable one. The *ResNet50* and *Xception* models were selected, as they had been used in other related studies [2]. To train the models, a dataset of 4886 samples was created by machining billets of aluminium 6061 (Figure 1) with different combinations of machining parameters and tool diameters to achieve a wide range of surface roughnesses.

ϕ	Parameter	Min	Max
1.5 mm	a_p [mm]	0.1	0.5
	a_e [mm]	0.2	1.2
	n [rpm]	20000	25000
	V_f $[\frac{mm}{min}]$	50	550
3 mm	a_p [mm]	0.1	0.6
	a_e [mm]	0.2	1.4
	n [rpm]	20000	25000
	V_f $[\frac{mm}{min}]$	50	550
4 mm	a_p [mm]	0.1	2
	a_e [mm]	0.2	0.8
	n [rpm]	16000	16000
	V_f $[\frac{mm}{min}]$	50	550

Table 1: Machining parameter ranges divided by tool diameter and relative to the aluminium billets. Please notice that the choice of one parameter influences the range of the others due to the limited power of the machine.

Before selecting the parameters, an estimation of the power was made to prevent damage to the machine. The Computer-Aided Manufacturing (CAM) software Autodesk Fusion 360 automatically generated four different machining strategies based on the depth and position of each sample. This increased the variability of the dataset and the adaptability of the trained models. For each machined billet, pictures of each sample were taken and linked to their corresponding S_a values. Each picture was then augmented to increase the dataset size six-fold. Three different loss functions were used to train the models, and the dataset was split so that 80% was used for training, 10% for validation, and 10% for testing. The training process was stopped after 100 epochs, and the learning rate was reduced as the epoch number increased to improve performance and reduce overfitting.

Figure 1: Two machined aluminium billets. The four different adopted machining strategies can be noted on the surface. Each machined rectangle represents a sample.

2.2. Testing of the models

The selection of the best model was conducted with great care, considering both the performance on the training and test datasets in order to avoid the selection of an overfitted model. Validation and testing errors were meticulously analyzed to locate the potential start of overfitting and to evaluate the training process (Figure 2).

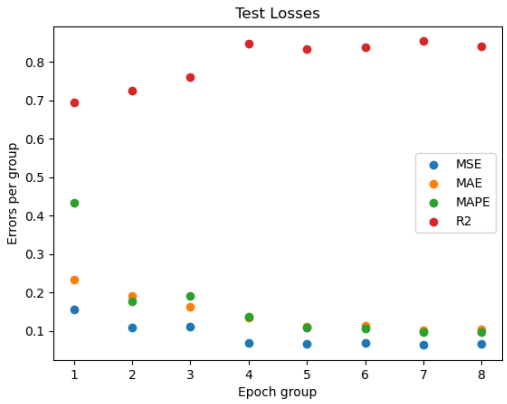

Figure 2: Evolution of test errors (this was plotted for each trained model). Each epoch group considers the model after 8 epochs of training.

Various metrics were employed to assess the aforementioned processes, and ultimately, the correlation between the predicted values and the ground truth was plotted to evaluate the accuracy evolution in each epoch. The final out-

Model		*CompleteDataset*				*TestDataset*			
Architecture	Loss	MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2
Xception	MSE	0.015	0.038	0.036	0.97	0.062	0.102	0.097	0.854
Xception	MAPE	0.049	0.057	0.038	0.912	0.138	0.143	0.114	0.717
Xception	Huber	0.016	0.042	0.037	0.969	0.063	0.102	0.096	0.854
ResNet50	MSE	0.022	0.044	0.038	0.956	0.072	0.109	0.1	0.827
ResNet50	MAPE	0.069	0.12	0.089	0.872	0.107	0.164	0.145	0.77

Table 2: Best models errors and metric calculated using the complete dataset and the test one.

comes of the training and testing processes are presented in Table 2.
Following the identification of the best model, its accuracy was tested on a new dataset consisting of 52 previously unseen samples to validate the model's architecture and to verify its adaptability to novel surfaces. Two conditions were examined to test the accuracy: a natural environment, employing only ambient light, and a controlled environment in which the LED light of the camera was turned on and regulated to full brightness, resulting in the clearest images. To improve the accuracy and stability of the predictions, the final estimation was computed as the mean of all those made by the model on the base image and all its enhancements. This approach helped to reduce the impact of noise or variations that may exist in a single prediction and provide more reliable results. The use of multiple predictions and their averaging is a common practice in machine learning and can enhance the robustness of the models, particularly in scenarios where there are multiple sources of variability or uncertainty. It can be observed from Figure 3 that the average error of the model was acceptable for a normal milling process without any stringent requirements. However, the existence of at least one significant outlier greater than 1μm may pose a challenge for implementing such a monitoring system in a real machining environment.

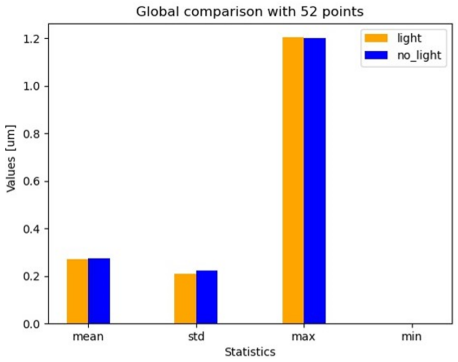

Figure 3: Analysis of the *AE* between the estimations performed over the new 52 machine samples.

3. Machining parameters optimization

The choice of the best combination of machining parameters represents a challenge for any type of optimizer as the objective functions conflict with each another. In this case, two were chosen: maximizing the *Material Removal Rate* and minimizing the S_a . There cannot be found a unique solution to such problems but only a family of feasible ones that form the so-called *Pareto Front*. Only the *feed rate* and the *rpm* were chosen to be optimised while the depths of cut were fixed as changing them leads to the need for the generation of new machining paths and G-Codes. The input parameter space is represented in Table 3.

Parameter	**Minimum**	**Maximum**
a_p [mm]	0.6	0.6
a_e [mm]	0.6	0.6
n [rpm]	11500	13500
V_f $[\frac{mm}{min}]$	500	950

Table 3: Corners of the hyper-space of the inputs of the process.

3.1. Setting the optimizer

In order to build a Gaussian Regression Process (GPR) a *kernel* must be chosen. This represents a crucial element as it holds the relationship between different input points and determines the behaviour of the regression function that is fitted on the input data points. In this application, the *Matern 5/2* kernel was chosen

as it was used in other related studies and is more sensitive to a change in the input data compared to the simpler and basic *Exponential* kernel which also means that possesses more hyper-parameters. Such GPR must be fitted and built on the available data points chosen as input. In this work, seven points were used including the corners of the hyper-rectangle (Figure 4). The hyper-parameters of the model were found using the maximum likelihood estimate approach which aims to maximize the posterior likelihood probability of such parameters to belong to the model with the chosen inputs and outputs. Regarding the latter, these were generated by combining the two objective functions in a weighted sum that allowed to transition from a Multi-Objective Optimization problem to a Single-Objective one.

$$f_j(\boldsymbol{x}_i) = w_j \cdot y_{1,i} + (1 - w_j) \cdot y_{2,i}, \qquad (2)$$

$$where \quad y_{1,i} = -\frac{MRR_i}{60}, \quad y_{2,i} = Sa_i \qquad (3)$$

The weights w_j used in Equation 2 were chosen to be 0, 0.05, 0.1, and 0.2. The MRR was divided by 60 (minutes to seconds) and made negative as the aim of the optimizer was to minimize the objective function. By changing the values of the weights the optimizer can be focused on finding parameters that prioritise productivity over quality, as the value of the weight is increased, or vice-versa, on guaranteeing a better surface finishing to the detriment of the MRR. This allows the optimiser to move its choice close to the theoretical *Pareto Front* of this Multi-Optimization problem.

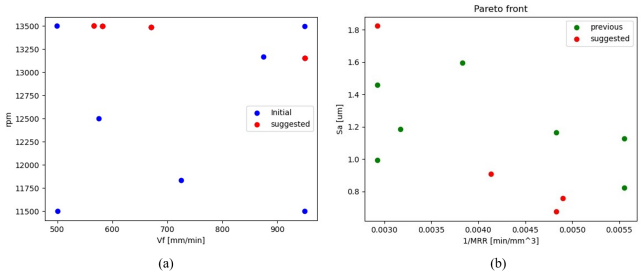

Figure 4: Picture (a) represents the suggested points by the optimiser after the first iteration is completed. Picture (b) represents the outputs associated with each input data point and the newly machined ones.

3.2. Optimization procedure

In Bayesian optimization (BO), the initial step is to generate objective functions. In this study, four objective functions were created, each corresponding to a different weight. Afterwards, Gaussian processes (GPs) were fitted to each of these objective functions, using the input data points to generate the models. Finally, the acquisition function was run over the model to suggest the best combination of parameters to minimize the objective function. This acquisition function makes its decision based on a trade-off between exploitation and exploration. Exploitation involves searching for better combinations of parameters in the vicinity of known ones, while exploration aims to search for unexplored regions of the hyper-space that may offer a better solution. The "Expected Improvement" function was used, as it strikes a good balance between these two goals and is commonly used in the literature on BO (see [3], [4]). Once a new combination of parameters was suggested a new sample was created and a new data point was added to the input space and the optimization process was run again to obtain a new suggestion. In this specific case, each iteration suggested four new points because, as mentioned before, four different objection functions were optimised, one at a time. The optimization loop was stopped after the optimizer could not provide any more useful points as, after a single iteration, it was suggesting some already known points.

3.3. Optimization results

The optimizer was capable of finding four different unused combinations of parameters that can be noticed in Figure 4 marked by the red dots. In Figure 5 one of the four optimization results is represented. The 'Posterior Mean' plot shows how the mean of the GPR is distributed in the parameter space and the red dots represent the input data points. The same can be said for the 'Posterior sd.' while the 'Acquisition function' represents how the acquisition function is looking for the point to suggest. As it can be noticed its focus was around the area with high sd but not exactly centred on the maximum point as its exploitation part is focusing it on a more known area.

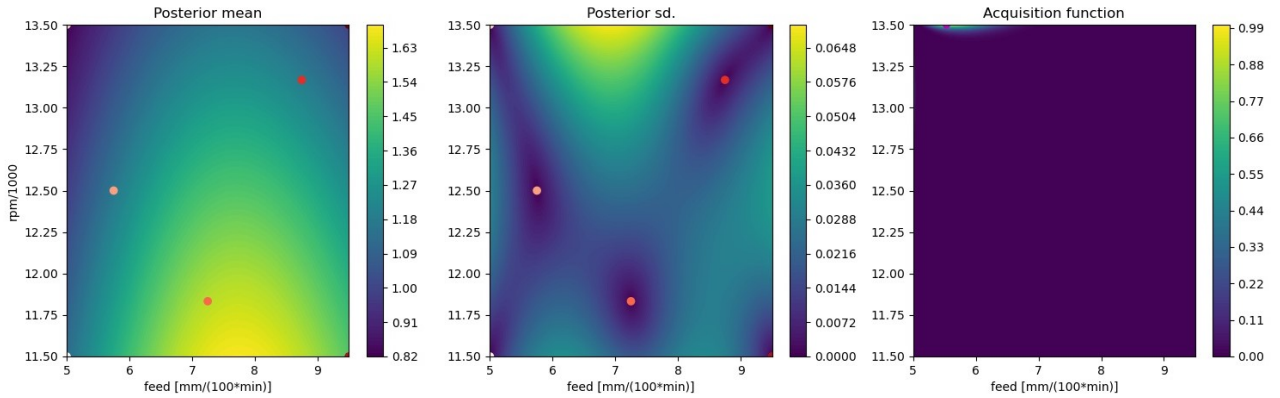

Figure 5: $w_{MRR} = 0$ and $w_{Sa} = 1$. Posterior mean, Posterior standard deviation and Acquisition function of the different models created using different weights. They are named "Posterior" because computed after the GPR model was fitted on the available data points and the BO process was concluded. The red points present on the first two heat maps represent the data points used to t the model. The one on the last picture to the right instead, represents the newly suggested one from the BO process.

4. Automated system

Finally, the monitoring system and the previously found optimised combinations of parameters were joined to improve the 5-axis CNC machine (Pocket NC V2-50 by PentaMachine) to nearly an autonomous one. The machine was capable of machining a slot, estimating its bottom S_a using the USB Camera (Bysameyee USB Digital Microscope) mounted next to the spindle and, if the estimated roughness was below the imposed threshold, changing the feed rate and rpm of the G-Code of the next machining operation in order to achieve the desired surface finish. Such a choice was based on the previously found parameters using the BO process. The process can be broken down into these main points:

1. Initial G-Code modification
2. G-Code uploading on the interface
3. Machining operation
4. Machined surface's picture acquisition
5. Optimisation
6. G-Code modification
7. G-Code uploading on the interface
8. Second machining operation

The G-Code of each machining operation was modified such that the machine could be paused during the surface roughness estimation and home the tool once the prediction was done. The latter was performed at the end of the machining operation before the tool was positioned at the reference (home) position. This operation required another modification of G-Code to position the camera in the centre of the machined area. All the modifications of the G-Codes were performed by a *Python* script that was able to read the ISO Code, identify the machined area based on paths' coordinates and insert the line of codes to position the camera and stop the machine. Another script was developed to modify the next G-Code with the newly chosen parameters and to control the machine interface which was a web page. In detail, the code had to be able to home the machine once the estimation of the S_a was completed, change the new G-Code, upload it and make the machine start with the new machining operation. To communicate with the web page the *Selenium* package was used as it offered various useful functions.

5. Bibliography

- Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, *O'Reilly Media*, SECOND EDITION

6. Conclusions

In this work the feasibility of creating an effective but simple monitoring system was investigated along with the possibility of exploring the benefits of Bayesian Optimization applied in a machining environment. The final assembled system was successful in autonomously modify-

ing its cutting conditions once the surface finish of the previously machined area was not as desired. Convolutional Neural Networks proved to be effective tools to perform feature-free estimations with a fast computational time and low memory use. Even if easy to implement because of the vast amount of open-source libraries developed by skilled researchers, their biggest disadvantage relies upon the need for a dataset to be properly trained. The creation of such resources consumed nearly half of the time required to create this work and also needed an extensive amount of tools and materials. Such challenges might be overcome by implementing shared repositories in a near future. The best-trained network tested over 52 new samples reported an *Mean Absolute Error* lower than 0.4 μm with a *Standard Deviation* lover than 0.35 μm. Further improvements to the monitoring system will aim to reduce the number of outliers and increase the adaptability of the models. Bayesian Optimization proved to be an effective tool when the amount of data available is scarce and when the input-to-output relationship is not known and must be considered as a black box. Only seven starting points were needed to find four new combinations of machining parameters and, even if the tool wear was not taken into consideration, two among the four newly suggested points resulted in producing a lower S_a possessing a higher *MRR* compared to points with lower feed rates. In future developments, this optimizer could be used even with different materials as it only requires a low amount of samples to operate and can easily adapt to different machining conditions and operations. In addition, tool wear should also be considered even if this upgrade would change the optimizer from a static one to a time dependent one, increasing the complexity of the system and the monitoring system which will have to be able to identify also the tool's worn area.

7. Acknowledgements

I would to express my deepest thanks to my Korean advisor, Professor Sahan Kim, who helped me graduating at KAIST and followed me during the making of this work showing the beauties and importance of the scientific research universe, and to my Italian, Professor Annoni Massimiliano Pietro Giovanni who supported me in graduating here at Politecnico of Milano, revising my work and suggesting me interesting new research topics still related to this work. Last but not least I would love to thank my family, parents and friends for always staying close to me during this amazing journey.

References

[1] Geometrical Product Specifications (GPS) — Surface texture: Profile method — Terms, definitions and surface texture parameters, ISO 4287:1997.

[2] Yonglun Chen, Huaian Yi, Chen Liao, Peng Huang, Qiuchang Chen. Visual measurement of milling surface roughness based on Xception model with convolutional neural network, *Measurement*, Vol. 186, 2021.

[3] Peter I. Frazier. A tutorial on Bayesian Optimisation, *arXiv:1807.02811v1*, 2018.

[4] Alejandro Morales-Hernández, Sebastian Rojas Gonzalez, Inneke Van Nieuwenhuyse, Jereon Jordens, Maarten Witters, Bart Van Doninck. Constrained multi-objective optimisation of process design parameters in settings with scarce data: an application to adhesive bonding, *arXiv:2112.08760v1*, 2021.