Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 113 (2022) 17–22

www.elsevier.com/locate/procedia

ISEM XXI

Virtual Operators with Self and Transfer Learning Ability in EDM

D. Držajić[a], M. Wiessner[a], U. Maradia[a], D. Piga[b]

[a]GF Machining Solutions, Via dei Pioppi 2, Losone 6616, Switzerland
[b]Dalle Molle Institute for Artificial Intelligence (IDSIA) USI-SUPSI, Via la Santa 1, Lugano-Viganello 6962, Switzerland.

* Corresponding author. *e-mail address:* dalibor.drzajic@georgfischer.com

Abstract

Increasing the manufacturing resource efficiency requires pushing the process limits by customizing the machining parameters to the job at hand. Such optimizations performed usually based on empirical methods and experience due to the sheer complexity of physical modeling. Current research presents a system comprised of advanced machine learning methods, where model-based optimization employed to simultaneously create a digital twin and optimize the process as a self-learning virtual operator using a case study with the EDM processes.

Keywords: Iterative Machine Learning; Blackbox Optimization; Electrical Discharge Machining (EDM); Artificial Intelligence.

1. Introduction

Improving the resource efficiency for economic and environmental motives is an increasing demand across different industries, which is bolstered even further by innovations in materials, evolutions in part design and customization. These factors may adversely affect the manufacturing process efficiency if the process parameters are not set optimally for the application specific requirements, resulting in the machine tools used at a much lower efficiency then their intrinsic capacity. For example in Wire EDM, the wire annealing parameters used to prepare the wire for threading in the wire circuit are different for each wire material, diameter and annealing length. Thus tuning such parameters lead to significant improvements in the threading operation so guaranteeing the autonomy of the machine tool. Similarly, machining parameter settings such as pulse energy, frequency of sparks, servo loop reference could be adjusted based on the application in terms of used wire type, diameter, work piece material and its height to obtain optimal process and so maximal roughing speed outputs. Thus, machining parameter tuning is very common and performed by skilled operators. Additionally, skilled operators are capable to fine tune parameters faster in new similar (but not identical) situations in order to generalize their knowledge on that particular task. For this reason generalization of similar tasks are crucial for developing EDM technologies otherwise the lead-time may result easily infinite.

1.1 Manufacturing process optimization

In order to support operators and decrease the lead-time in development mechanical modeling and empirical modeling are widely used approaches to gain insights into the process and derive optimal results faster as described in [1]. In the example of Wire EDM, complex phenomena involved in the process [2] limit the use of physical modelling in an industrial context, where machine operators may not possess the necessary expertize. Furthermore, such models require fitting parameters due to their complexity, which may need to be determined using further experimentation. Such limitations

10.1016/j.procir.2022.09.113

have driven a number of researchers towards statistical and machine learning techniques [3] and optimization algorithms such as genetic algorithms or design of experiments methods for the purpose of process modeling and optimization. Complex manufacturing processes often require non-linear multi-objective optimization, where neural networks have been widely investigated. However, the robustness of such models are commonly compromised by noise in the measured outputs, originating from process repeatability, metrology, etc. Nevertheless, the required experiments to create a reliable model can be expensive. In fact, a cost-effective method robust against noise and requiring minimal experimentation is still lacking both in academia and in industry.

2. ViOLA: Virtual operator with self and transfer learning ability

The selection of machining data typically relies on the business objectives related to the machining costs and part quality. Here, mainly two extreme scenarios can be envisioned. In the first case, no prior knowledge is available to select the optimal parameters, typical for a new machining technology, material, etc. In the other scenario, sufficient data and expert knowledge are available to reach the machining

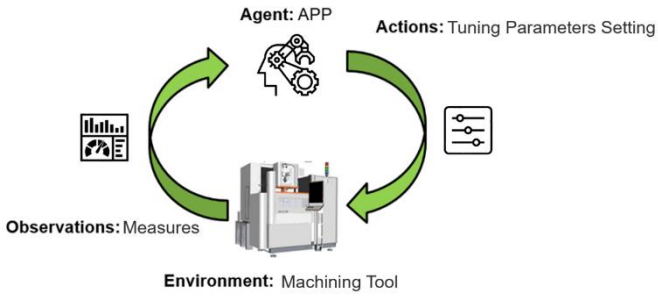

Fig. 1. Trial and error iterative optimization approach.

objective, however, it may not be accessible across production locations. In this work, model-based optimization (MLA) and transfer learning algorithm (TLA) are used to address these challenges. As shown in Fig. 1, by combining these algorithms in a system, a virtual operator is created, having the ability to optimize the process for unknown situations, but also utilize the existing data to meet the complex machining objectives, enabling autonomous extraction of the optimal process outputs from the machine tools.

3. MLA: Model-based optimization Learning Algorithm

Machining parameters optimization can be formulated into the problem of finding the optimum of an unknown function in a multidimensional space. In fact, a process model can be thought as a multiple input multiple output nonlinear *unknown* function

$$y = f(x) + \varepsilon \quad (1)$$

where y are the process outputs, $f(x)$ is an unknown function describing the process as the depicted in upper subplots in Fig. 2, $x \in X$ are the machining parameters in the feasible parameters setting space and ε is the noise which will be assumed as Gaussian to be addressed effectively with Gaussian Processes (GP) uncertainty confidence bounds. Here the goal is to find the best input x, that is the one that optimizes a scalar *known* function of the outputs $g(f(x))$ (e.g., machining speed, quality, wear, threading score, etc.). Since the function to be optimized $g(f(x))$ is unknown (because of unknown $f(x)$), optimization methods such as simplex method can be employed, in the case of small noise ε variance. On the other hand, machine learning methods with GP as described in detail in [4] can be used where noise ε variance is not trivial. GP Regression is used for reliable estimation of $f(x)$ for WEDM in [5], without a consideration for the cost of optimization of $g(f(x))$.
In the problems where experiments are expensive (in terms of time and/or cost) as in the case of manufacturing processes, Bayesian Optimization (BO) has been shown to outperform other state of the art global optimization algorithms on a number of challenging optimization benchmark functions [6]. What makes Bayesian Optimization different from other procedures is that it constructs a probabilistic model for $f(x)$ and then exploits this model to make decisions about where in X to next evaluate the function while integrating out uncertainty and with the goal of optimizing $g(f(x))$. This results in a procedure that can find the minimum of difficult non-convex functions with relatively few evaluations, at the cost of performing more computation to select the next point in X to try [7].

3.1. Objective function

In order to perform BO, a prior over functions that express the assumptions about the unknown function $f(x)$ must be selected. GP is a convenient and powerful prior distribution on functions. The support and properties of the resulting multivariate Gaussian distribution on functions are determined by a mean function and a positive definite covariance function (95% confidence region for the underlying function $f(x)$ as shown in Fig. 2), i.e. kernel function. By a suitable choice of the kernel, GPs encompass, as some particular cases, linear regression, spline interpolation, neural networks and their combinations. Multiple outputs can be taken into account in GPs assuming them uncorrelated one with each other. In current work, Matern 5/2 kernel is employed. Kernel related hyperparameters are optimized by maximizing the marginal likelihood function.

3.2. Acquisition function

The second main component of BO is the a*cquisition function* that is used to construct a utility function from the posterior of $f(x)$, which allows to determine the next input to evaluate x_{next}. There are several popular choices of acquisition functions and in this work it has been adopted the Expected Improvement (EI) one. If one aims to optimize $g(f(x))$, Furthermore a jitter $\xi > 0$ is introduced for trading-off exploration and exploitation by attempting to get a better cost by the amount of the selected jitter. EI could be formulated as written in the following key equation:

$$x_{next} = argmax_{x \in X}\, E\left[\max(0, g_{min} - g(f(x)) - \xi)\right] \quad (2)$$

where the expectation is with respect to the GP posterior distribution of $f(x)$. The g_{min} term stands for the minimum cost ever observed and is updated accordingly after each new evaluation is made. The jitter ξ allows the algorithm to behave optimistically by striving to beat g_{min} by setting it positive of the amount we are willing to outperform the current optimal cost and so restrict even more the non-zero EI region, in the attempt to exclude such a region which are not yielding the desired improvement ξ respect to g_{min}. When g is a nonlinear function, the expectations must be computed numerically (e.g. via Monte Carlo sampling), due to the fact that its posterior distribution is non-analytical. EI by nature provides a trade-off between exploration (evaluating at points with high uncertainty) and exploitation (evaluating at points with a low mean, the optimal point). As a consequence, BO finds the optimum of $g(f(x))$ while learning the function $f(x)$ as well as the variance of the noise ε. Hence, at the end of the optimization process, a robust process model (digital twin of the process) is obtained in relation to the inputs and outputs. Note that, GPs can deal with continuous, discrete or categorical variables via either variable transformations (e.g. encoding) or a suitable choice of the kernel function.

The procedure of BO is explained in a simplified manner using Fig. 2 and Fig. 3. In Fig. 2, two existing evaluations (experiments) are seen which are used to form a Gaussian prior with predicted variance and predicted mean, along with the actual process function $f(x)$. $EI(x)$ shows the maximum expected improvement in the value of y at x_{next} as depicted in the lower subplots Fig. 2. As the next evaluation is performed at x_{next} obtained from (2), the predicted mean and variance of $f(x)$ are updated based on the observed outputs y. This process is repeated until a good approximation of $f(x)$is obtained while also achieving optimized $g(f(x))$ (optimal process parameters), as illustrated in Fig. 2.

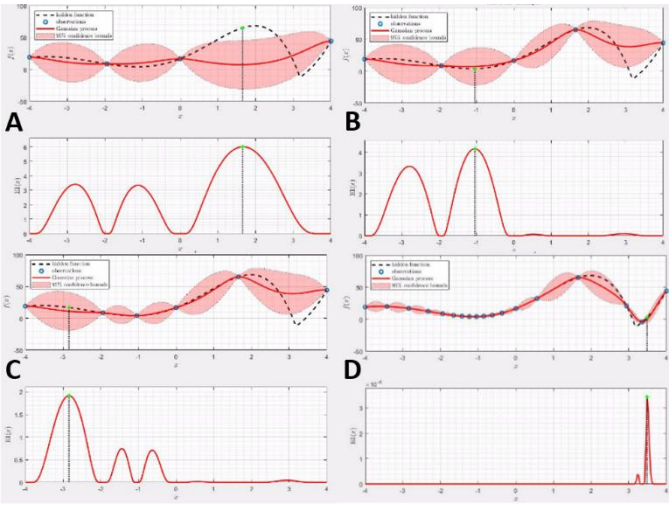

Fig. 2. Optimization session in different stages: A): after 4 experiments, B): after 5 experiments, C): after 6 experiments, D): after 18 experiments.

4. TLA: Transfer Learning Algorithm

We focus now on the implementation of a strategy that takes advantage of the prior knowledge of similar models to add so called transfer learning capabilities to the developed optimization algorithm described in Section 3.

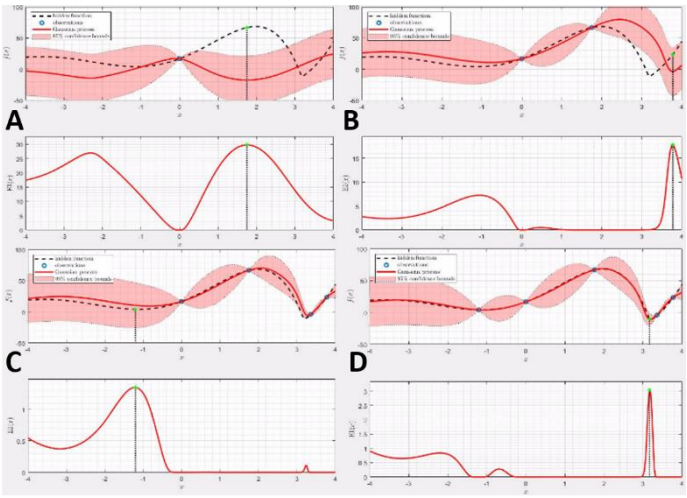

Fig. 3. Optimization session in different stages using transfer learning. A): after 1 experiment, B): after 2 experiments, C): after 4 experiments, D): after 5 experiments.

Different approaches has been assessed as a possible strategy that takes advantage of the prior knowledge acquired from previous experiments. To this aim, we enhanced the ordinary Bayesian Optimization with transfer learning capabilities in order to exploit all the available data from previously executed experiments. This knowledge is used to speed-up the process by reducing the number of experiments required to converge to an optimum of a new function to be minimized as depicted in Fig. 3. Prior knowledge is necessary for transfer learning feature. The following equation shows the general GPR [8]:

$$p(f_*|X_*, Y_*, X) = N(\mu_*, \Sigma_*)$$
$$\mu_* = m(X_*) + K_*^T[K + \sigma_\varepsilon^2 I]^{-1}(Y_* - m(X)) \quad (3)$$
$$\Sigma_* = K_{**} - K_*^T[K + \sigma_\varepsilon^2 I]^{-1}K_{**}$$

where $p(f_*|X_*, X, y)$ is the conditioned PDF of the unknown function, X_* the observed evaluation points and Y_* the observed outputs, X the points to be evaluated, $m(\cdot)$ is the prior of GP, $K(\cdot,\cdot)$ is the selected kernel function such that $K = K(X, X)$, $K_* = K(X_*, X)$, $K_{**} = K(X_*, X_*)$, σ_ε standard deviation of outputs noise.
The heuristic aims to use a non-zero prior function $m(\cdot)$ which can be parametric or non-parametric depending of the use-cases.

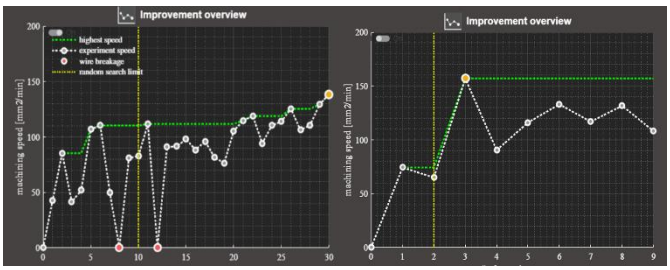

Fig. 4. Left: machining speed maximization without prior knowledge. Right: machining speed maximization with prior knowledge on a different material. In this case the optimum in the new situation is obtained after 3 iterations. However to make sure it is actually the new optimum, 9 iterations has been performed, yielding to 70% effort reduction compared to the 30 iteration in the source optimization session.

Conventionally was assumed that if μ_s is mean of GP of the source model and μ_t the mean of GP of the still unknown target model, we can plug $m(\cdot) = \mu_s(\cdot)$ and insert it in the equation (3).
With the model description in the source context, it has been tried to transfer this knowledge in a target context in which it is wanted to exploit similarities with the original one and hence get the optimal tuning much faster as depicted in Fig. 4. Among different use-cases in EDM, a reduction up to 80% of the iterations needed to achieve an optimal operational point has been recorded.

5. Case studies in EDM

5.1 Wire annealing and threading optimization

In WEDM more autonomous machining operations require highly reliable threading and rethreading capabilities in order

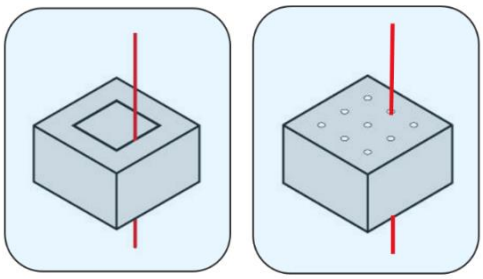

Fig. 5. Left: submerged threading in the workpiece slot without auxiliary jet. Right: submerged threading in the workpiece holes (Ø wire +0.1mm) without auxiliary jet.

resume machining in case of wire break and when a geometry is cut and we need to move to the next one and start from a small workpiece hole as shown in Figure. In that sense, one of the key feature to guarantee such a degree of autonomy in WEDM machines is the wire annealing system used to cut thermo-mechanically the wire in a such a way to confer it physical and geometrical properties for reliable threading. Typical properties can be easily resume as straight wire, diameter reduction, bullet-shaped tip as depicted in Fig. 6.

In general, fine-tuning such wire annealing parameters is extremely time consuming because generalizing the suitability of such parameters is not available for different workpieces

Fig. 6. Picture of a WEDM wire prepared for threading operation.

heights, wire materials and threading conditions like submerged workpiece without threading jet in a small hole. In that sense the optimality of parameters found in a determined condition would probably not be optimal even for small changes, meaning in a further fine-tuning of this parameters each time if demanding threading reliability is required.

A virtual operator was developed to fine-tune such annealing parameters as shown in Fig. 7. The main perks of this closed-loop optimization is that the session is carried out in full autonomy [9] as plug and play solution during the night or the weekend.

Fig. 7. GUI for automatic wire annealing parameters optimization to maximize the trade-off of threading reliability in terms of successful threading score in hard conditions and threading speed to reduce time spent in such operation during machining.

In order to optimize the threading performance in terms of scoring without failures and perform the latter as fast as possible to reduce machine downtime, 4 parameters namely heating power, stretching speed, cooling air delay and threading speed are accurately selected by the self-learning algorithm using previously described optimization techniques where no gradient is available, so called blackbox optimization. The goal is to find a suitable parameters setting $\theta \in \Theta$, where Θ is a hyper-rectangle whose vertices correspond to minimum, respectively maximum values of each parameter is allowed to take, such that θ maximizes the threading nonnegative scoring rate $THD_{score}(\theta)$ with a soft constraint on the threading nonnegative elapsed time $THD_{time}(\theta)$ which is used as a secondary objective which will play a crucial role on the Pareto optimal front trading-off score and elapsed time depending on the goal weight λ, as expressed in Equation 4. This constraint introduces a penalty term whose weight λ is a hyper-parameter selected by domain experts based on the importance degree it carries.

$$\min_{\theta \in \Theta} [-THD_{score}(\theta) + \lambda \cdot THD_{time}(\theta)] \quad (4)$$

The optimization session is performed iteratively where each experiment at stage k is evaluated using parameters setting θ_k measuring ${THD_{score}}^{(k)}(\theta_k)$ and ${THD_{time}}^{(k)}(\theta_k)$ which will be added to the list of all experiments done for stage 0…k as shown in Table 1, in order to obtain from the optimization engine a guess of the optimizer location θ_{k+1}. This procedure is repeated at user discretion until the desired target of threading performance is achieved as depicted in Fig. 7.

Table 1. List of experiments used to guess the next most promising θ to evaluate. The last row corresponds to the next evaluation point based on the available history of experiment in probabilistic terms of improvement.

Heating power [%]	Stretching speed [mm/s]	Cooling delay [ms]	Threading speed [mm/s]	THD Score	THD time [sec]
6.9	-1	0	100	23	53
13.4	-7	269	145	12	45
7.8	-3	434	159	5	39
⋮	⋮	⋮	⋮	⋮	⋮
9.5	-4	112	161	226	38
10.1	-6	350	178	573	34
10.4	-5	225	159	1'000	40
10.3	-5	215	141	1'134 ± 54	35 ± 3

5.2 WEDM roughing speed maximization

We introduce now the consecutive use-case of blackbox optimization respect to the first one described in Section 5.1,

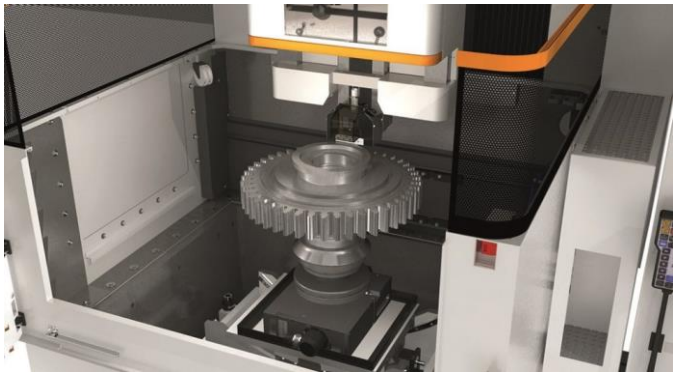

Fig. 8. Typical aerospace machining speed maximization for fir tree disk use-case where the iteration distance corresponds to each fir tree machined.

in the sense that this optimization can be concatenated to previous one and carried out as a combined optimization session to threading performance enhancement. The first step in WEDM technology development is to define a parameters

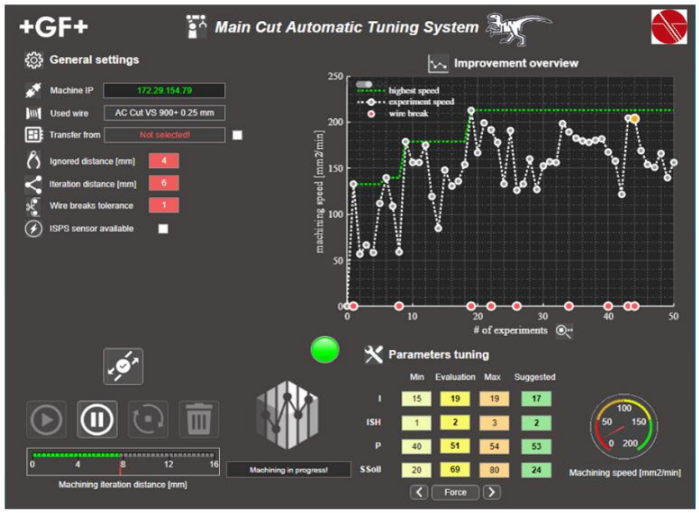

Fig. 9. GUI developed to maximize WEDM roughing machining speed through the definition of a multi-objective optimization problem by also considering several technical aspects such as feasibility of the parameters, wire break likelihood, process instability beside the technology parameters.

setting for roughing machining, meaning just cutting the workpiece according to the geometry regardless the surface quality and geometry precision which are supposed to be achieved in so called cut trims where less pulse energy is used to achieve such a degree of machining high precision. Therefore the main goal of roughing cutting is to be performed as fast as possible for boosting productivity as the aerospace use-case depicted in Fig. 8. and save time for more delicate trim cuts operations. So the optimization problem we want to solve is formulated as:

$$\min_{\theta \in \Theta} \left[-\mathrm{WEDM}_{\mathrm{speed}}(\theta) + \sum_{i=1}^{N} \lambda_i \varphi_i(\theta) \right] \quad (5)$$

where $\mathrm{WEDM}_{\mathrm{speed}}(\theta)$ is the roughing speed to maximize (main objective) and $\varphi_i(\theta)$ are secondary objectives introduced as soft constraints as wire break probability, instability of the WEDM process and pulse energy, used to correctly trade-off speed with robustness selecting the appropriate weights λ_i. The latter aspect is non-trivial because the behavior of the optimization can lead to dramatically different outcomes depending how hyperparameters are set.

5.3 WEDM roughing speed Digital Twin

During optimization sessions we collect iteratively data after each experiment which can be exploited offline to extract insights of the WEDM process.

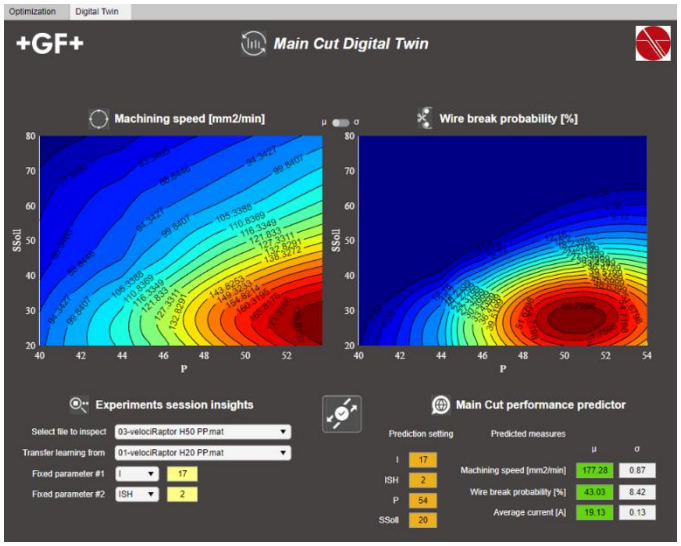

Fig. 10. GUI for accessing the Digital Twin of the WEDM roughing in terms of machining speed and wire break probability. This GUI allows the users to explore models in terms of predictions (with associated uncertainty) of machining performance, potential danger of wire breaks by moving on the contour plots to find safe areas and so saving time, material and machine allocation to redo machining if the model is confident about the predictions.

In particular we can model machining speed and wire break probability without having to evaluate on a real machine the actual performance of a parameter setting if the prediction confidence bound is sufficiently tight.
The latter aspect is a key feature in supporting application technology development tasks in terms of reducing trial and error efforts, machine tool commitment and waste material as workpieces and wire electrodes.

Conclusions

A comprehensive self-learning system is created to perform the tasks of a virtual machine tool operator. Bayesian Optimization with Gaussian processes is used for model-based optimization. The capabilities of the system are demonstrated within different contexts using EDM processes case studies. It is shown that the virtual operator is able to:

- perform multi-objective optimization and create a robust process model (Digital Twin) simultaneously, with lower required experiments,
- achieve complex optimization objectives to obtain application and business goal specific manufacturing process technology,
- achieve results through exploration in large multivariate parameter space, which can lead to new insights into the manufacturing processes.
- transfer prior knowledge of an existing optimization session to a new one if the task is similar but not identical and do inference about the new optimum point, reducing dramatically the number of experiments needed to converge to it.

Acknowledgements

This study has been provided with the support of the Swiss Innovation Agency - Innosuisse and La Scuola universitaria professionale della Svizzera italiana – SUPSI.

References

[1] Maradia U, Wegener K, EDM Modelling and Simulation, Electrical Discharge Machining (EDM): Types, Technologies and Applications. ISBN 978-1-63483-591-6; 2015. p. 67-121.
[2] Hinduja S, Kunieda M. Modelling of ECM and EDM processes. CIRP Annals- Manufacturing Technology 2013. 62(2): 775-797.
[3] Monostori L, Westkämper, E., Brussel, H. van, Markus, A. (1996) Machine learning approaches to manufacturing. CIRP Annals-Manufacturing Technology 45(2): 675-712.
[4] Rasmussen C, Williams C. Gaussian processes for machine learning. MIT Press; 2006.
[5] Yuan J, Wang K, Yu T, Fang M. Reliable multi-objective optimization of high-speed WEDM process based on Gaussian process regression. International Journal of Machine Tools and Manufacturing 2008. 48(1): 47-60.
[6] Jones D. A taxonomy of global optimization methods based on response surfaces. Journal of global optimization 2001. 21(4): 345-383.
[7] Snoek J, Larochelle H, Adams P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems 2012. p. 2951-2959.
[8. Alvarez AM, Rosasco L, Lawrence ND. Kernels for Vector-Valued Functions: A Review. ISBN 978-1-60198-558-3. Now Foundations and Trends; 2012.
[9] Shahriari B, Swersky K, Wang Z, Adams RP, Freita N. Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE 2016. 104(1): 148-175.