


Autonomous and data-efficient optimization of turning processes using expert knowledge and transfer learning

Journal Article**Author(s):**

Maier, Markus; Kunstmann, Hannes; Zwicker, Ruben; [Rupenyan-Vasileva, Alisa Bohos](#) ; Wegener, Konrad

Publication date:

2022-05

Permanent link:

<https://doi.org/10.3929/ethz-b-000536080>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Originally published in:

Journal of Materials Processing Technology 303, <https://doi.org/10.1016/j.jmatprotec.2022.117540>



Autonomous and data-efficient optimization of turning processes using expert knowledge and transfer learning

Markus Maier^{a,*}, Hannes Kunstmann^b, Ruben Zwicker^a, Alisa Rupenyan^{a,c}, Konrad Wegener^b

^a Inspire AG, Technoparkstrasse 1, 8005 Zurich, Switzerland

^b Institute of Machine Tools and Manufacturing (IWF), ETH Zurich, Leonhardstrasse 21, 8092 Zurich, Switzerland

^c Automatic Control Laboratory, ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland

ARTICLE INFO

Associate Editor: Markus Bambach

Keywords:

Turning
Transfer learning
Expert knowledge
Process optimization
Bayesian optimization
Gaussian process models
Machining

ABSTRACT

Process parameters in machining are predominantly selected by following manual tuning procedures. Using data from the system and dedicated performance indicators combined with learning-based approaches enables automating these procedures while reducing the costs of the machining process. This study investigates efficient data-driven approaches for autonomous parameter selection in turning. The number of experimental trials for finding optimal process parameters is reduced by incorporating expert knowledge and transferring knowledge between different tasks. The turning process costs are modeled using Gaussian process models, and the selection of informative experiments is achieved by Bayesian optimization. In this study, all tested methods using expert knowledge or transfer of knowledge reduced the number of experiments by at least 40% compared to a standard approach for parameter selection based on Bayesian optimization without expert knowledge, confirming the efficiency of the applied methods.

1. Introduction

Turning is a common machining process, requiring the selection of various parameters such as feed per revolution, cutting speed, and depth of cut, depending on the specific machining requirements. A mis-specification of these parameters results in uneconomical processes, or in non-compliant final parts. The choice of the parameter values depends on the specific cutting tool, workpiece, machine, and cooling lubricant. These categories can be further subdivided into specific properties, such as material type or geometry, which leads to an enormous variety of individual optimization tasks.

Today, in industrial environments, the parameter selection is mainly performed by machine operators based on trial and error and experience. The operators follow individual strategies for parameter selection, thus influencing the efficiency and the consistency of the parameter selection procedure. Expert operators will find (near-) optimal parameters with minimal experimental effort, while novice operators typically require significantly more experiments. Autonomous optimization-based parameter selection holds the potential to improve productivity and product quality, and to ensure consistency by reducing the variance in the parameter selection.

Existing approaches for autonomous optimization often rely on

process modeling in combination with selection of experiments, as shown in Table 1. A classical approach in process modeling relies on empirical models, or on polynomial function approximations, where the corresponding coefficients in the candidate model function, or in the polynomial are identified from experimental trials. Standard approaches such as the Taylor equation, the Kienzle equation or the kinematic surface roughness model can be found in Klocke (2011). Polynomial functions have been used in Khamel et al. (2012) to model the tool life, the surface roughness and the cutting forces in hard turning. Empirical models and polynomial functions are beneficial in terms of simplicity and interpretability, but suffer from limited generalizability. For example, the kinematic surface roughness model does not consider surface roughness increase due to built-up edge, which is known to exist and described in Klocke (2011). To overcome the limited generalizability of these models, Homami et al. (2014) used artificial neural networks (ANNs) for the prediction of surface roughness and flank wear in turning. ANNs achieved impressive results in image recognition (Krizhevsky et al., 2012), where deep convolutional neural networks perform significantly better than standard feature-based methods. In manufacturing, a deep convolutional long short-term memory (LSTM) neural network autoencoder has been demonstrated for predicting multi-step machine speeds by Essien and Giannetti (2020). Fan et al.

* Corresponding author.

E-mail address: maier@inspire.ethz.ch (M. Maier).

<https://doi.org/10.1016/j.jmatprotec.2022.117540>

Received 30 September 2021; Received in revised form 6 February 2022; Accepted 25 February 2022

Available online 2 March 2022

0924-0136/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Table 1
Methods for autonomous parameter selection in turning.

Source	Method for selection of experiments	Modeling techniques	Objective & Constraints on output
Yang and Targ (1998)	Taguchi method (orthogonal array)	None	Optimization of tool life and surface roughness individually
Nian et al. (1999)	Taguchi method (orthogonal array)	None	Multi objective optimization of tool life, cutting force and surface finish
Homami et al. (2014)	Full factorial design	Neural networks	Optimization of flank wear constrained to surface roughness
Aramesh et al. (2013)	Full factorial design	Gaussian process regression	Multi objective optimization of surface roughness, tool wear, and productivity
Abbas et al. (2016) with details provided in Sadek et al. (2015)	1) Full factorial design 2) m-EGO (adaptive sampling)	Gaussian process regression	Multi-objective optimization of surface quality and material removal rate
Maier et al. (2019)	Bayesian optimization	Gaussian process regression	Optimization of production costs with surface roughness constraints

(2020) demonstrated anomaly detection for semiconductor manufacturing processes utilizing a neural network denoising autoencoder. In contrast to these data-rich applications, the available number of experiments for process parameter selection in turning is typically small because the costs for each experiment are very high, especially for tool life measurements. A disadvantage of ANNs in combination with only a few data points is that ANN models are typically not able to specify the confidence in their predictions. Intuitively, one would expect lower uncertainties of the model's predictions near measurements and higher uncertainties far away from measurements, which is typically not captured by neural networks. Recently, Maier et al. (2019) used probabilistic Gaussian process (GP) models to model a longitudinal turning process. Gaussian process models are more flexible than empirical models or second order polynomial functions, and at the same time provide confidence intervals of the associated predictions, which makes them suitable for modeling in the presented study.

Process parameter optimization studies in turning are often based on statistical analysis of classical design of experiments (DoE), as reported in Homami et al. (2014). Alternatively, Yang and Targ (1998) used the Taguchi design for experiment selection. Another approach for parameter optimization is adaptive sampling, where the experiments are determined iteratively, based on an existing model. Bayesian optimization is a form of adaptive sampling and has been demonstrated for turning in Maier et al. (2019). In Bayesian optimization, the algorithm trades off exploration and exploitation, examining points close to a local optimum (exploitation), but at the same time samples at points with high uncertainty to discover previously unknown optima (exploration), as described in Shahriari et al. (2016). As shown in Gardner et al. (2014), Bayesian optimization is a data-efficient method for experiment selection by leveraging model knowledge, which makes it the method of choice in this study.

Whenever large variability in the production scenarios is present, the parameter selection process can be enhanced by incorporating expert knowledge and by transferring learned knowledge from one scenario to another related one, instead of re-optimizing from scratch. Gaussian process models are well suited for incorporating both types of knowledge as they incorporate the Bayes rule, which naturally allows for the specification of prior knowledge, mimicking the expert knowledge-based approach of operators in practice. Nevertheless, the number of studies using expert knowledge or transfer of knowledge combined with Gaussian processes or Bayesian optimization in turning is limited. The use of a Gaussian process model combined with an analytical model to predict the surface roughness in turning was demonstrated in Misaka et al. (2020), where the model accuracy was improved, especially for cases with only a few training samples. Guidetti et al. (2021) have also demonstrated incorporating empirical process relations in the Gaussian process model for plasma spray coating. The results of Misaka et al. (2020) are very encouraging, but the work is limited to surface roughness modeling and does not consider autonomous parameter selection and data-efficient adaptive sampling. The present study investigates and

compares different methods to incorporate expert knowledge, and to transfer knowledge in turning with the aim to reduce the number of experiments for autonomous parameter selection.

This paper is organized as follows: A short introduction to Gaussian process models and Bayesian optimization is provided, followed by an explanation of the experimental turning setup, the optimization task, and the implementation of the different parameter selection approaches. The parameter selection approaches mainly differ in the applied Gaussian process model. A standard Gaussian process model without expert knowledge, similar to Maier et al. (2019), is compared to Gaussian process models utilizing expert knowledge, either incorporated through predetermined hyperparameters or through a predefined prior mean function, and transfer of knowledge, established by multi-task learning. Finally, the paper reports and compares the results of the individual approaches for cutting speed selection in turning.

2. Gaussian process models and Bayesian optimization

A detailed introduction to Gaussian processes is provided in Rasmussen and Williams (2006). An overview on Bayesian optimization is provided in Shahriari et al. (2016). An introduction related to Gaussian processes and Bayesian optimization applied to optimize the process parameter set-up in manufacturing is provided in Maier et al. (2020) for grinding.

2.1. Gaussian process regression

As described in Rasmussen and Williams (2006), a Gaussian process is a collection of random variables, which have a joint Gaussian distribution, and is fully defined by a mean function $m(\underline{x})$ and a covariance function $k(\underline{x}, \underline{x}')$, where \underline{x} and \underline{x}' are two different inputs, in our particular case they correspond to points in the parameter space of the process parameters. Gaussian process models can be used to represent a distribution over functions (Rasmussen and Williams, 2006), and are used in this work to provide a model of the process costs as a function of the process parameters. Each process cost value, measured at a given process parameter, corresponds to a noisy realization of the Gaussian process. The cost function GP model consists of the mean of the GP at each process parameter configuration, and the associated covariance, specified with a kernel function. A Matérn 5 covariance function is selected in this study, in line with previous work in Maier et al. (2019), where it showed good performance in modeling a turning process. The Matérn 5 covariance function is defined following Rasmussen and Williams (2006),

$$k(\underline{x}, \underline{x}') = \sigma_f^2 \exp\left(-\sqrt{5}r\right) \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \quad (1)$$

$$r = \sqrt{(\underline{x} - \underline{x}')^T \underline{\underline{P}}(\underline{x} - \underline{x}')} \quad (2)$$

$$\underline{P} = \text{diag}(l_1^{-2}, l_2^{-2}, \dots, l_D^{-2}) \quad (3)$$

where σ_f^2 is the signal variance, and r is the Euclidean distance between the process parameter points \underline{x} and \underline{x}' scaled by the length scale parameters l_i^2 for each process parameter up to dimension D . In this study, the process parameter corresponds to the cutting speed, which needs to be optimized. Based on Eq. (1) it can be seen that the covariance is equal to σ_f^2 for parameter values $\underline{x} = \underline{x}'$ and declines for a larger distance between the parameter values \underline{x} and \underline{x}' . Therefore, the model outputs corresponding to input parameters close to each other, are assumed to be strongly correlated, whereas model outputs for parameters far away from each other are assumed to show a greater variation. Based on the Gaussian process and t available measurements y_t at process parameter points $\underline{x}_{1:t}$, predictions for an arbitrary process parameter point \underline{x}_* can be made following Rasmussen and Williams (2006),

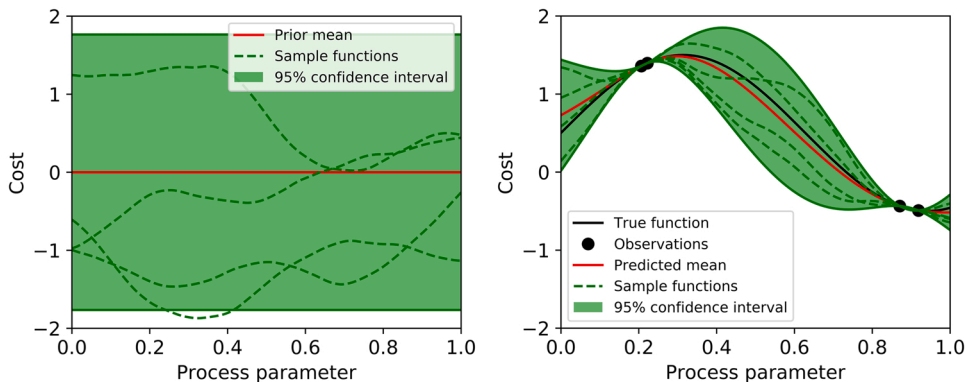
$$\mu_t(\underline{x}_*) = m(\underline{x}_*) + k^T(\underline{x}_*) (\underline{K} + \sigma_N^2 \underline{I})^{-1} (\underline{y}_t - \underline{m}(\underline{x}_{1:t})) \quad (4)$$

$$\sigma_t^2(\underline{x}_*) = k(\underline{x}_*, \underline{x}_*) - k^T(\underline{x}_*) (\underline{K} + \sigma_N^2 \underline{I})^{-1} \underline{k}(\underline{x}_*) \quad (5)$$

$$\underline{K} = \begin{pmatrix} k(\underline{x}_1, \underline{x}_1) & \dots & k(\underline{x}_1, \underline{x}_t) \\ \vdots & \ddots & \vdots \\ k(\underline{x}_t, \underline{x}_1) & \dots & k(\underline{x}_t, \underline{x}_t) \end{pmatrix} \quad (6)$$

where μ_t is the predicted mean value, σ_t^2 is the predicted variance, \underline{I} is the identity matrix, $\underline{m}(\underline{x}_{1:t}) = (m(\underline{x}_1) \dots m(\underline{x}_t))^T$ is the mean function vector, \underline{K} is the covariance matrix, $\underline{k}(\underline{x}_*) = (k(\underline{x}_*, \underline{x}_1) \dots k(\underline{x}_*, \underline{x}_t))^T$ is the covariance vector, and $k^T(\underline{x}_*)$ is the transpose of $\underline{k}(\underline{x}_*)$. Furthermore, it is assumed that the measurements are corrupted by Gaussian noise $N(0, \sigma_N^2)$. Fig. 1 illustrates the Gaussian process regression. The Gaussian process prior is shown on the left side, where no observations are available. The red line represents the prior mean function $m(\underline{x})$, which for this case is zero. Specifying a zero prior mean function is a common choice if no expert knowledge is available but it is also possible to select non-zero prior mean functions. The picture also shows four arbitrary sample functions of the Gaussian process prior. It can be seen that different functions can result from the same Gaussian process. The distribution over all functions is specified by the 95% confidence interval (shaded green area). By conditioning the Gaussian process on available measurements, predictions can be obtained using Eqs. (4) and (5). The result of the prediction is visualized in the right panel of Fig. 1. It can be seen that the prediction shows a low uncertainty close to the observations and higher uncertainties for parameters further away from available observations. While the true function is unknown for typical optimization tasks, it is shown in this example for comparison.

The hyperparameters σ_f^2 , l_i^2 , and σ_N^2 can be specified based on expert



knowledge. When such a specification is not available a priori, the hyperparameters can be determined by maximizing the marginal log likelihood $\log p(\underline{y}_t | \underline{\theta})$ based on available experiments, as reported in Rasmussen and Williams (2006).

$$\log p(\underline{y}_t | \underline{\theta}) = -\frac{1}{2} \underline{y}_t^T (\underline{K}_\theta + \sigma_N^2 \underline{I})^{-1} \underline{y}_t - \frac{1}{2} \log |\underline{K}_\theta + \sigma_N^2 \underline{I}| - \frac{t}{2} \log 2\pi \quad (7)$$

$$\underline{\theta}^* = \arg \max \log p(\underline{y}_t | \underline{\theta}) \quad (8)$$

Maximizing the marginal log likelihood provides a natural trade-off between the quality of the model fit and the model complexity, as described in Rasmussen and Williams (2006).

2.2. Gaussian processes for multi-task learning

In the conventional single-task/single-output approach, each task is learned individually without exploiting correlations between tasks. Gaussian process models can be extended to learn multiple-tasks/multiple-outputs simultaneously, as investigated in detail in Alvarez et al. (2012). The motivation for such an approach is to exploit relations between tasks with the aim of improving the individual model accuracy, without increasing the amount of data for the specific task. For turning, a task corresponds to the optimization of a specific turning setup. For example, in a first task the coating of the cutting tool might be from type A and in a second task it might be from type B, while the rest of the setup remains unchanged. In multi-task learning, the tasks are learned simultaneously, which allows to exploit correlations between the task's outputs. Similar to the single task case, a Gaussian process for multi-task learning is defined following Alvarez et al. (2012),

$$\underline{f} \sim GP(\underline{m}, \underline{K}) \quad (9)$$

where \underline{m} is now a vector of mean functions for each task, and \underline{K} expresses the covariance between the different tasks.

The main difference between Gaussian processes for a single task and multiple-tasks is the specification of the covariance matrix. For the multi-task case, it is necessary to find a suitable expression for the matrix \underline{K} , which specifies the relation between the different tasks. The linear model of coregionalization (LMC) allows the specification of a valid covariance matrix \underline{K} by expressing the different output functions $f_d(\underline{x})$ as linear combinations of random functions as reported in Journel and Huijbregts (1978),

$$f_d(\underline{x}) = \sum_{q=1}^Q \sum_{i=1}^{R_q} a_{d,q}^i u_q^i(\underline{x}) \quad (10)$$

where $u_q^i(\underline{x})$ are latent functions (hidden functions), $a_{d,q}^i$ are scalar coefficients, Q is the number of kernels, and R_q is the number of latent

Fig. 1. Illustration of the Gaussian process regression with a Matérn 5 kernel and the hyperparameters $\sigma_f^2 = 0.81$, $l_1 = 0.38$, and $\sigma_N^2 = 0$. The left panel shows four arbitrary functions of the Gaussian process prior depicted as dashed lines. The GP posterior distribution after four observations is shown on the right panel, where the dashed lines again correspond to four arbitrary functions and the solid line represents the predicted mean $\mu_t(\underline{x}_*)$. The shaded area shows the 95% confidence interval, which is $0 \pm 2\sigma_f$ for the prior distribution and $\mu_t(\underline{x}_*) \pm 2\sigma_t(\underline{x}_*)$ for the posterior distribution. The predicted mean $\mu_t(\underline{x}_*)$ and predicted variance $\sigma_t^2(\underline{x}_*)$ are calculated using Eqs. (4) and (5).

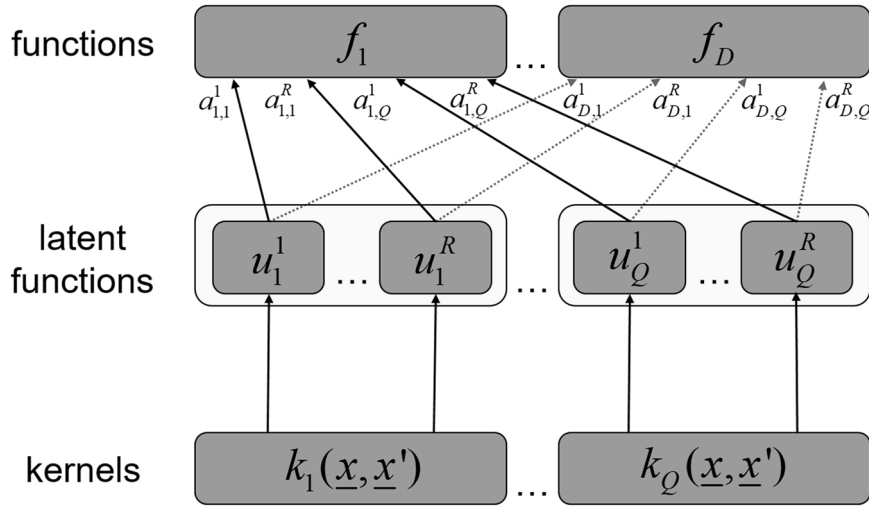


Fig. 2. Illustration of the LMC, as described in [Journal and Huijbregts \(1978\)](#).

functions in each group with the same kernel. Fig. 2 shows a schematic representation of Eq. (10). Several latent functions $u_q^i(\underline{x})$ are determined based on different kernels $k_q(\underline{x}, \underline{x}')$. Afterwards, the final functions $f_d(\underline{x})$ are calculated by linear combination of the latent functions. The latent functions $u_q^i(\underline{x})$ are used for model construction but cannot be directly measured and therefore are hidden. Only the output functions $f_d(\underline{x})$ are directly measurable and observable. In the turning example, the output function $f_1(\underline{x})$ corresponds to the production costs of the first turning task with cutting tool coating A, whereas the output function $f_2(\underline{x})$ corresponds to the production costs of the second turning task with cutting tool coating B.

By using the LMC, the matrix $\underline{K}(\underline{x}, \underline{x}')$ can be stated as in [Journal and Huijbregts \(1978\)](#),

$$\underline{K}(\underline{x}, \underline{x}') = \sum_{q=1}^Q \underline{B}_q k_q(\underline{x}, \underline{x}') \quad (11)$$

where \underline{B}_q is the coregionalization matrix with entries $b_{d,d'}^q = \sum_{i=1}^R a_{d,i}^q a_{d',i}^q$ and rank R_q .

As shown in [Alvarez et al. \(2012\)](#), by setting $Q = 1$ the general LMC is reduced to the intrinsic coregionalization model (ICM) and by setting $R_q = 1$ the general LMC is reduced to the semiparametric latent factor

model (SLFM). Hence, both the ICM and the SLFM are special cases of the general LMC. As described in [Cohn and Specia \(2013\)](#), different special cases can be derived from the ICM, which are illustrated in Fig. 3. If \underline{B}_q is equal to the identity matrix as shown in the left panel of Fig. 3, the tasks are modeled independently, but share the same kernel. The kernel only restricts the considered function space of the outputs but has no impact on the relation of the outputs. On the other hand, if all entries of the matrix \underline{B}_q are equal to one, the outputs are perfectly related and only have different noise levels.

As described in [Alvarez et al. \(2012\)](#), based on the data and the model, predictions for an arbitrary process parameter point \underline{x}_* follow a joint normal distribution $N(\underline{f}_*(\underline{x}_*), \underline{K}_*(\underline{x}_*, \underline{x}_*))$.

$$\underline{f}_*(\underline{x}_*) = \underline{K}_*^T (\underline{K}(\underline{X}, \underline{X}) + \text{diag}(\sigma_N^2) \otimes \underline{I}_N)^{-1} \underline{y} \quad (12)$$

$$\underline{K}_*(\underline{x}_*, \underline{x}_*) = \underline{K}(\underline{x}_*, \underline{x}_*) - \underline{K}_* (\underline{K}(\underline{X}, \underline{X}) + \text{diag}(\sigma_N^2) \otimes \underline{I}_N)^{-1} \underline{K}_*^T \quad (13)$$

For simplicity, the prior mean functions \underline{m} are assumed to be constants with a value of zero. The measured output vector is $\underline{y} = (y_{1,1}, \dots, y_{N,1}, \dots, y_{1,D}, \dots, y_{N,D})^T$, where D is the number of tasks and N is the number of data samples per task, assumed to be the same for all tasks for simplicity. The measurements of each output are assumed to be

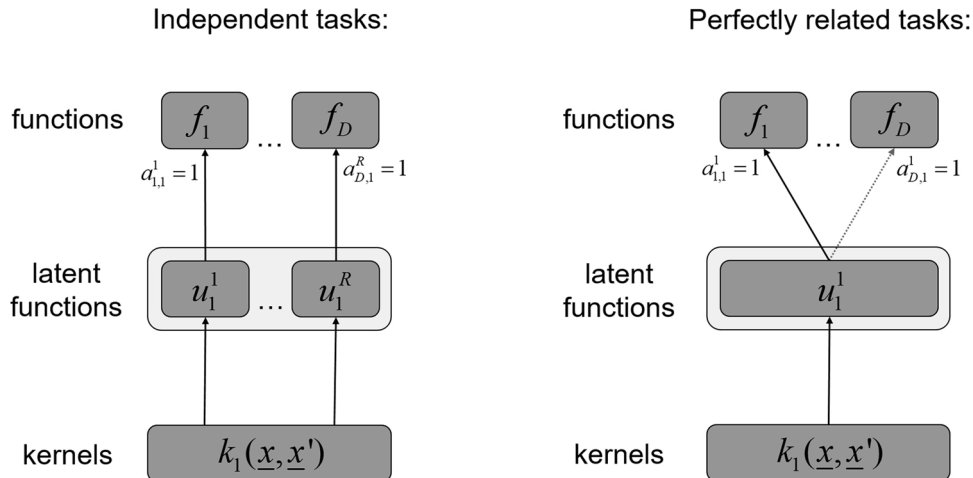


Fig. 3. Illustration of special cases of the ICM, as described in [Cohn and Specia \(2013\)](#). The left panel shows the case where the tasks are modeled independently but share the same kernel. The right panel shows the case where all tasks are perfectly related.

corrupted with an individual Gaussian noise with zero mean and variance $(\sigma_N^2)_d$. The matrix $\underline{K}(\underline{X}, \underline{X})$ is of dimension $ND \times ND$, and the matrix \underline{K}_x is of dimension $D \times ND$. Both matrices are calculated based on Eq. (11). Further details on their calculation are provided in the Appendix. \underline{I}_N is the identity matrix with dimensions $N \times N$, \underline{X} are the input points for all measurements of all tasks, and \otimes is the Kronecker product. The Kronecker product of $\text{diag}(\sigma_N^2) \otimes \underline{I}_N$ multiplies each element of the matrix $\text{diag}(\sigma_N^2)$ (dimensions $D \times D$) with matrix \underline{I}_N (dimensions $N \times N$), which results in a matrix of $ND \times ND$ dimension. For multi-task learning, the prediction in Eqs. (12) and (13) is given as a joint normal distribution. The prediction of a single function $f_d(\underline{x}_*)$ at a test parameter point \underline{x}_* , as for the single output case, corresponds to the marginal distribution of the joint normal distribution. For the prediction of the single function $f_d(\underline{x}_*)$, the mean is $\mu_d(\underline{x}_*) = (f_{\rightarrow}(\underline{x}_*))_d$, and the variance is $\sigma_d^2(\underline{x}_*) = (\underline{K}_{\rightarrow}(\underline{x}_*, \underline{x}_*))_{d,d}$, which directly follows from the properties of a joint normal distribution, as listed in Rencher and Christensen (2012).

The LMC has a number of hyperparameters collected in the vector $\underline{\theta}$, such as the elements of each coregionalization matrix \underline{B}_q , the signal noise vector σ_N^2 , and the parameters for each kernel $k_q(\underline{x}, \underline{x}')$. By using Matérn kernels, each kernel is characterized by the length scale parameters l_i^2 for each input dimension and a signal variance parameter σ_f^2 . The hyperparameters can be computed by maximizing the marginal log likelihood, similar to Eq. (8) for the single output case, as described in Alvarez et al. (2012).

$$\log p(\underline{y} | \underline{X}, \underline{\theta}) = -\frac{1}{2} \underline{y}^T \left(\underline{K}(\underline{X}, \underline{X}) + \text{diag}(\sigma_N^2) \otimes \underline{I}_N \right)^{-1} \underline{y} - \frac{1}{2} \log \left| \underline{K}(\underline{X}, \underline{X}) + \text{diag}(\sigma_N^2) \otimes \underline{I}_N \right| - \frac{ND}{2} \log 2\pi \quad (14)$$

$$\underline{\theta}^* = \arg \max \log p(\underline{y} | \underline{X}, \underline{\theta}) \quad (15)$$

2.3. Bayesian optimization

Having obtained the prediction of the Gaussian process model, Bayesian optimization can be used to select the next experimental parameters by maximizing an acquisition function. Within the Bayesian optimization framework, the acquisition function ensures the trade-off between exploration and exploitation, according to the specifics of the optimization problem (Mockus et al., 1978). The most straight-forward acquisition function, *probability of improvement* (Kushner, 1964) selects the location of the next experimental evaluation, based on the probability to improve the costs beyond the current lowest costs. Another way to choose promising points for experimental evaluation is the *upper confidence bound* acquisition function (Srinivas et al., 2010), which uses the predicted costs and the uncertainty estimation for the selection. An information-theoretical approach is used by the *predictive entropy search* acquisition function (Hernández-Lobato et al., 2014) for the selection of the next experiment.

In this study, the acquisition function of choice is *expected improvement*, as in Maier et al. (2019), which selects the point with largest (predicted) improvement compared to the last evaluated optimum as the location of the next experimental evaluation. The advantage of the expected improvement acquisition function is that it easily allows extension to constrained Bayesian optimization, while still providing a very good performance, as shown in Gardner et al. (2014). Following Mockus et al. (1978), the expected improvement acquisition function is

calculated as follows,

$$a_{EI}(\underline{x}) = (C_{min} - \mu_{t,co}(\underline{x}))F(Z) + \sigma_{t,co}(\underline{x})\phi(Z) \quad (16)$$

$$Z = \frac{C_{min} - \mu_{t,co}(\underline{x})}{\sigma_{t,co}(\underline{x})} \quad (17)$$

where C_{min} corresponds to the lowest measured cost so far, $F(Z)$ is the cumulative standard normal distribution $F(Z) = 1/\sqrt{2\pi} \int_{-\infty}^Z \exp(-t^2/2)dt$, and $\phi(Z)$ is the probability density function of a standard normal distribution $\phi(Z) = 1/\sqrt{2\pi} \exp(-Z^2/2)$.

3. Methodology

3.1. Experimental setup

The experiments are performed for longitudinal turning on a Swiss GT 32 turning machine from Tornos, shown in Fig. 4. The machine is equipped with an automatic bar feeder Robobar SBF 326 from Tornos, which handles round bars made of 1.4125 martensitic stainless steel with an initial diameter of 20 mm and a length of 3 m. During the cutting operation, the diameter of the bar is reduced from 20 to 7 mm in 13 steps, with a fixed depth of cut of 0.5 mm over a length of 20 mm. In this study, the feed per revolution is fixed to 0.05 mm/rev, and only the cutting speed is optimized. By changing the cutting speed of the turning operation, the machine controller may internally change other parameters such as the spindle acceleration (not considered for optimization in this study). During cutting, Blasomill 15 from Blaser Swisslube is used as

a cooling lubricant with a constant flow rate for all experiments. The cutting operations are performed using carbide inserts from Diametal (carbide M10/30, coating D30, geometry DCGX-FR070301, corner radius of 0.1 mm, and article number 236157), mounted on a right-hand tool holder Topdec SDACR from Diametal. The tool life of the cutting tool is determined based on VB_{max} measurements using a Leica Wild M10 microscope, whereby a tool with $VB_{max} \geq 100 \mu\text{m}$ is considered worn out.

3.2. Optimization task and cost calculation

In this study, the aim of the optimization is to find the cutting speed $v_{c,min}$ which minimizes the individual production costs.

$$v_{c,min} = \arg \min(\ln(C_{FE}(v_c))) \quad (18)$$

The cutting speed v_c is optimized over a large domain between 10 and 175 m/min. The upper bound of the cutting speed is selected based on the maximum rotational speed of the turning machine, reached during the final longitudinal cutting step. A wide range of cutting speeds results in costs that are different by orders of magnitudes. Having these large cost differences, accurate modeling typically requires short length scale parameters of the Gaussian process kernel, which generally slows down the optimization. Therefore, the logarithm of the individual production costs is modeled and optimized. Note that by optimizing in the log space the Gaussian noise is also changed from a normal to a log-normal distribution. The advantage of the log-normal distribution is that the predicted costs are always positive, which matches reality. The individual production costs C_{FE} are calculated as follows,



Fig. 4. Picture of turning machine.

$$C_{FE} = t_c \left(C_{MH} + \frac{C_f}{T} \right) \quad (19)$$

and the cutting time t_c for multiple cuts is calculated as,

$$t_c = \sum_{i=1}^{n_{cut}} \frac{D_i l \pi}{v_c f} \quad (20)$$

where T is the tool life, l is the length of the cut, D_i is the diameter at each cut, v_c is the cutting speed, and f is the feed per revolution. In this study, the machine hour-rate C_{MH} is assumed to be 90 U/h and the costs per cutting edge C_f are assumed to be 10 U. The cost unit is denoted with U for generality but corresponds to Swiss francs.

3.3. Modeling

The optimization is implemented in Python and uses the GPy library (GPy, 2012) for Gaussian process models. The aim of all Gaussian process models is to accurately model the production costs as a function of cutting speed. Four Gaussian process model versions are tested:

- Model 1 - standard model without expert knowledge (benchmark)
- Model 2 - model with fixed hyperparameters, determined from similar optimization tasks to transfer knowledge
- Model 3 - model with non-zero prior mean function, incorporating knowledge from available empirical or analytical models
- Model 4 - model based on multi-task learning, modeling several tasks together and transfer knowledge by utilizing correlations between different tasks.

Fig. 5 illustrates the four Gaussian process model variants.

The standard Gaussian process model (Model 1) uses a zero prior mean function and determines the hyperparameters by maximizing the marginal log likelihood based on available data points, as specified in Eq. (8). Throughout this study the maximization of the marginal log likelihood is performed using the quasi-Newton method by Broyden, Fletcher, Goldfarb, and Shanno (BFGS) with 1000 restarts. Details of the BFGS method are given in Nocedal and Wright (2006).

Instead of determining the hyperparameters by maximizing the marginal log likelihood, the hyperparameters can be fixed based on

expert knowledge, which is investigated as a possible alternative in Model 2.

The informative prior mean function of Model 3 can be specified based on known physical or empirical models. In this study, the Taylor equation is used to model the tool life as a function of the cutting speed, which according to Klocke (2011) is specified as follows,

$$T = C_v \cdot v_c^k \quad (21)$$

where C_v and k are model coefficients. The tool life predictions are then used in combination with Eq. (19) to calculate the individual production costs. The resulting deviation between the measured cost and the prediction based on the Taylor equation is modeled by a standard Gaussian process model using a Matérn 5 kernel. This corresponds to a two-step approach. The parameters of the Taylor equation are determined first by ordinary least squares using the *scikit-learn* library (Pedregosa et al., 2011). Afterwards the hyperparameters of the Gaussian process model are determined by maximizing the marginal log likelihood, following Eq. (8). The two-step approach is chosen to make the results comparable to standard curve fitting techniques based solely on the Taylor equation, which is identical to the first modeling step.

For transferring knowledge between different tasks, Model 4 uses multi-task learning. As introduced in Section 2.2, different multi-task learning models exist. In this study, the general idea of the used multi-task approach is to determine a common model for all tasks and model individual deviations to this model independently. Thus, a linear model of coregionalization based on two Matérn 5 kernels is used for multi-task learning, resulting in two coregionalization matrices \underline{B}_1 and \underline{B}_2 . The first coregionalization matrix \underline{B}_1 is filled with ones, and the hyperparameters of the associated Matérn 5 kernel are free. In this way, the combination of coregionalization matrix \underline{B}_1 and the corresponding kernel models the fully related share of the task's outputs. For the second coregionalization matrix and associated kernel, the coregionalization matrix is diagonal $\underline{B}_2 = \text{diag}(\kappa)$, the length scale hyperparameter of the associated kernel is free, and the signal variance of the kernel is fixed to one, as the signal variance is fully captured by the coregionalization matrix \underline{B}_2 . By specifying a diagonal coregionalization matrix, the outputs of each task are modeled independently, but share the same hyperparameters. Note that the specifications of \underline{B}_1 and \underline{B}_2 are identical

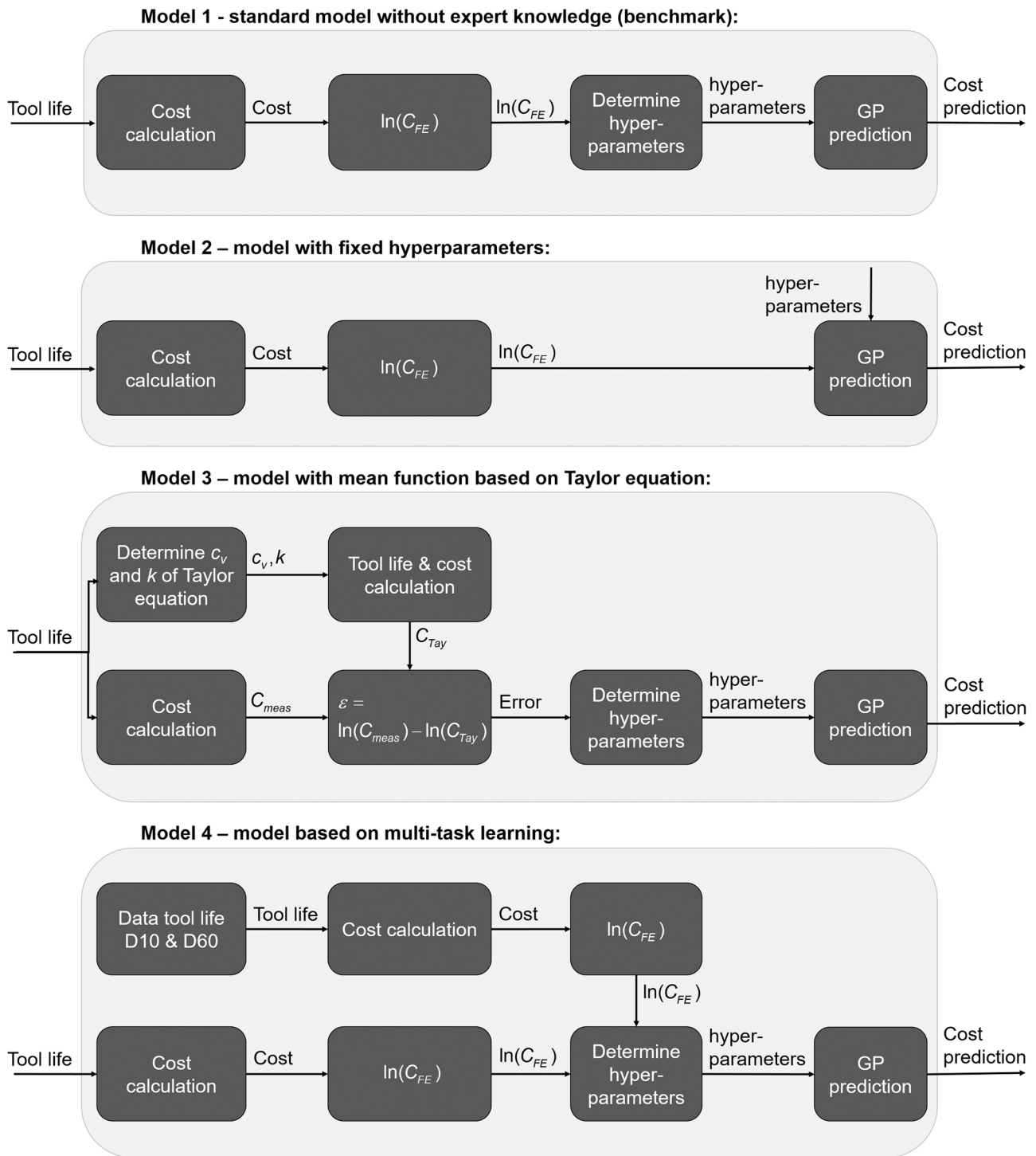


Fig. 5. Overview of tested Gaussian process model variants. The different Gaussian process variants are a standard Gaussian process, a Gaussian process with fixed hyperparameters, a Gaussian process with a prior mean function based on the Taylor equation, and a multi-task Gaussian process.

to the special cases shown in Fig. 3. The hyperparameters of the multi-task model are determined by maximizing the marginal log likelihood, following Eq. (15).

The benchmark model (Model 1) and the model based on a non-zero prior mean function (Model 3) do not need additional information. For the multi-task learning model (Model 4), measurements from other tasks are necessary, and the model with fixed hyperparameters (Model 2) requires the specification of the hyperparameters, which can also be achieved by using measurements from other tasks. Fig. 6 shows available results for optimization tasks using carbide cutting inserts with

coating D10 and D60, which serve as prior knowledge for the current optimization task. The current optimization task is slightly different than the previous optimization tasks because in the current setup the carbide insert is coated with D30 instead of D10 or D60 – the rest of the setup remained unchanged. The nomenclature of the tool coating is only a classification of the tool manufacturer and does not correspond to specific physical properties of the tool, as specified in Diametal (2017). This situation is typical to an industrial environment where the operator of the turning machines does not know the exact physical properties of the tool, due to trade secrets of the tool manufacture. For the fixed

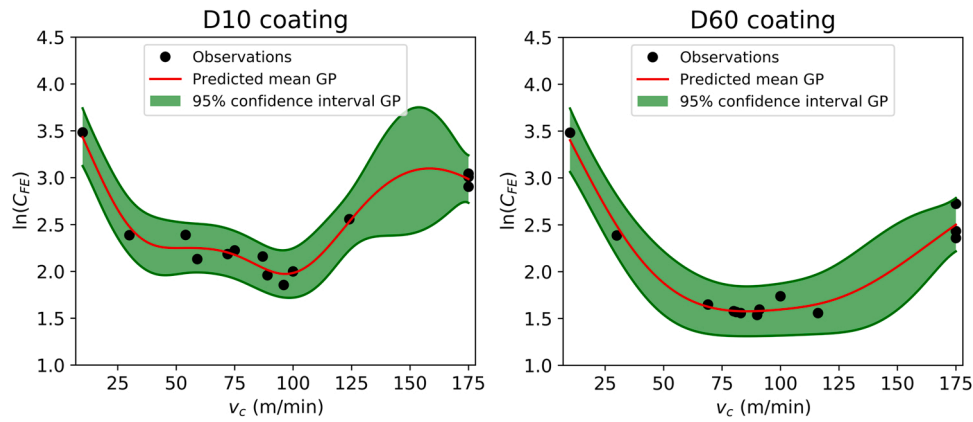


Fig. 6. Gaussian process regression for coating D10 and D60.

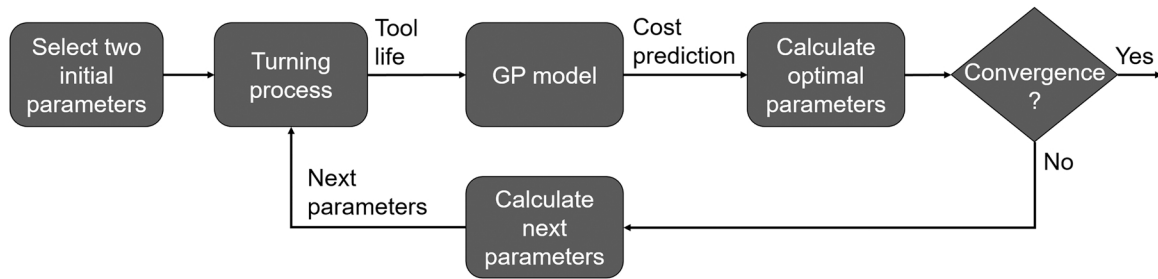


Fig. 7. Flowchart of the general optimization procedure. The GP model can be represented by one of the four model variants as illustrated in Fig. 5.

hyperparameter case, the hyperparameters of the D60 coating determined by maximizing the marginal log likelihood utilizing all D60 measurements are reused for the D30 case, which are $l_{1,D60} = 0.975$, $\sigma_f^2_{D60} = 17.02$, and $\sigma_N^2_{D60} = 0.015$. Note that these hyperparameters correspond to a normalized cutting speed between zero and one. For the multi-task learning approach, the available observations from the D10 and the D60 coating are used together with the new observations from the D30 coating for model updating, as specified in Eq. (15), and for predictions, as specified in Eqs. (12) and (13).

3.4. Optimization procedure

The different Gaussian process models are tested within the general optimization procedure, as shown in Fig. 7. Each optimization is started with two initial experiments. In the cases where the hyperparameters

are determined by maximizing the marginal log likelihood, the initial experiments should be chosen in an area, which allows a good characterization of the function shape. In Bayesian optimization, knowing the fastest function change allows to determine the smallest length scale of the GP model, which improves robustness of the optimization. Thus, the initial cutting speeds are set to 10 m/min and 30 m/min because this region is the expected maximum slope of the cost. This selection is especially useful for the standard Gaussian process model.

Each experiment is started with a new cutting insert that is then used for manufacturing, until it is considered worn out following the VB_{max} criterion. For low cutting speeds it might take very long until the tool reaches the VB_{max} criterion, while the cost for the machine time is already very high. To avoid unnecessarily long experiments, the experiments are stopped when the contribution of the cost for the machine time exceeds 95% of the total cost. In this case, the cost is calculated as

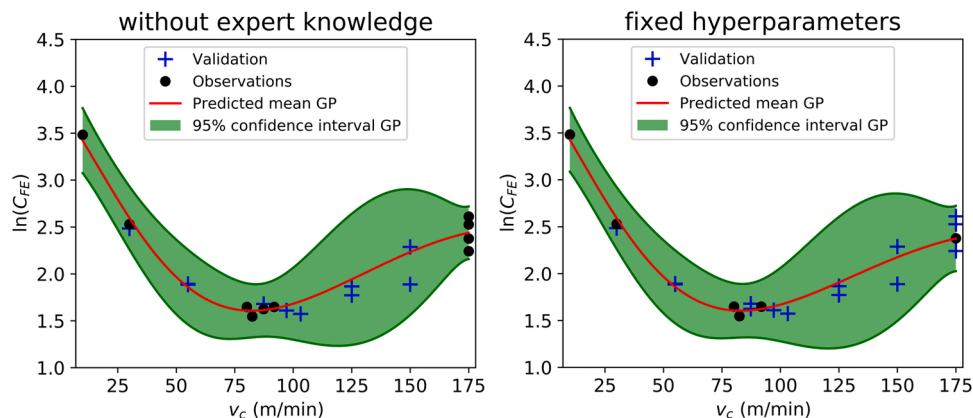


Fig. 8. Left: Result of Bayesian optimization after 10 experiments using a standard Gaussian process model without expert knowledge (Model 1). Right: Result of Bayesian optimization after 6 experiments using fixed hyperparameters (Model 2).

the average between the machine time cost alone, assuming an infinite tool life, and the cost, which would occur if the insert is assumed worn out directly after the next workpiece is manufactured.

Having obtained the measurements, the Gaussian process model can be used to calculate the optimal parameters and to assess convergence of the optimization. Similar to Maier et al. (2019), convergence is reached when for three consecutive iterations the change in the optimal predicted cost $\mu(x_{opt})$ is less than 5%, the variance at the optimal predicted parameter is $2\sigma(x_{opt})/\mu(x_{opt}) < 25\%$, and the optimal cutting speed varies less than 15 m/min. If convergence is not reached, the cutting speed maximizing the expected improvement acquisition function, as specified in Eq. (16), is used as the next test parameter. In total, four individual optimization runs have been performed. To reduce the total number of experiments for the optimization runs, measurements from previous optimization runs were reused when a measurement is available within ± 2 m/min of the requested cutting speed value given that the measurement was not used during this run yet.

4. Experimental results

Fig. 8 left shows the model prediction after 10 experiments for the standard Gaussian process model without expert or transferred knowledge (Model 1). The observations (black dots) are acquired by the Bayesian optimization procedure, as specified in Fig. 7. After the two initial experiments at cutting speeds of 10 m/min and 30 m/min, the algorithm requests an experiment at the maximum cutting speed of 175 m/min because based on the two available experiments the model expects a decrease in cost for an increase in cutting speed. Having these three data points, the algorithm explains the data by a nearly linear model with a high length scale parameter and a high noise level, where the cost decreases slightly for higher cutting speeds. Considering only these three data points, without additional information, the prediction is reasonable, but the model is too simple for the investigated process. As a consequence of the simple initial model, a cutting speed of 175 m/min is investigated four times until the model reduces the noise estimation to an adequate level and is able to distinguish between noise and signal. The following experiments are selected close to the optimum. After 10 experiments the algorithm reaches convergence.

In addition to the observations used by the Gaussian process regression, several validation points are shown and used to assess the performance of the Gaussian process regression. For each cutting speed, the prediction of the Gaussian process regression follows a Gaussian distribution with a mean and a 95% confidence interval. In the ideal case, the validation points should be centered around the mean function and the variation of the data points should match the 95% confidence interval of the prediction. Indeed, all validation points are within the predicted confidence interval. The uncertainty prediction close to the data points is mainly explained by noise, whereas the uncertainty

between measurement points is a combination of noise and uncertainty due to missing data. It can be observed that for high cutting speeds above 150 m/min, the measured cost values are more scattered. In the Gaussian process model the noise is assumed to be identical for all cutting speeds. Therefore, this approximation causes the model to overestimate the noise for lower cutting speeds. It would be possible to use a different likelihood which reflects the heteroscedasticity of the data (different noise levels for different process parameters), as shown in Muñoz-González et al. (2011). However, such approaches usually increase the model complexity and are typically analytically intractable and need approximation methods such as expectation propagation, as shown in Muñoz-González et al. (2011). Alternatively, they require computationally expensive sampling methods such as Markov chain Monte Carlo (MCMC), as reported in Goldberg et al. (1997). Due to the increase of the model complexity by using a tailored likelihood and the good performance obtained by the simple model no attempts have been made to improve the model. However, the use of a tailored likelihood might be an interesting direction for future research.

The right panel of Fig. 8 shows the result for the model with fixed hyperparameters (Model 2). The two initial experiments are again at cutting speeds of 10 m/min and 30 m/min. As for the case without prior knowledge the next experiment is performed at the highest cutting speed of 175 m/min. However, the algorithm does not test high cutting speeds again because it is able to directly distinguish between noise and signal. Afterwards the algorithm samples close to the optimum and reaches convergence after 6 experiments. The resulting posterior prediction of the standard Gaussian process model and the model with fixed hyperparameters is very similar after convergence but the model with fixed hyperparameters needs 6 experiments instead of 10 experiments, which reduces the experimental effort significantly.

The next investigated approach is the specification of a non-zero prior mean function (Model 3), as displayed in Fig. 9. At least two tool life measurements at different cutting speeds are required to fit the Taylor equation. The optimization is again initialized with experiments at cutting speeds of 10 m/min and 30 m/min. However, the experiment at the lowest cutting speed of 10 m/min takes very long without reaching the end of the tool life. Hence, the 95% criterion for the cost is reached before the insert is worn out and no tool life is measured for this experiment. To obtain a second measurement for the tool life the third experiment is conducted at 175 m/min as requested by the optimization with the standard Gaussian process model (Model 1). After these three experiments, two tool life measurements are available and a prior mean function based on the Taylor equation can be calculated. Afterwards the algorithm starts to sample close to the optimum and converges after 6 experiments. As shown on the left of Fig. 9, the Taylor equation shows a high error between measurements and predictions. Therefore, using the predicted tool life by the Taylor equation for costs calculated alone leads to high model errors, as shown on the right of Fig. 9. Although the Taylor

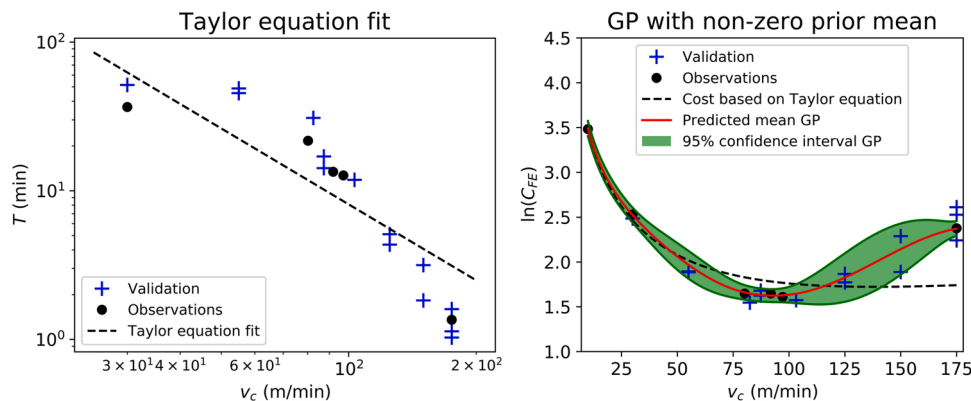


Fig. 9. Taylor equation fit and Gaussian process regression with prior mean based on Taylor equation (Model 3) after 6 experiments.

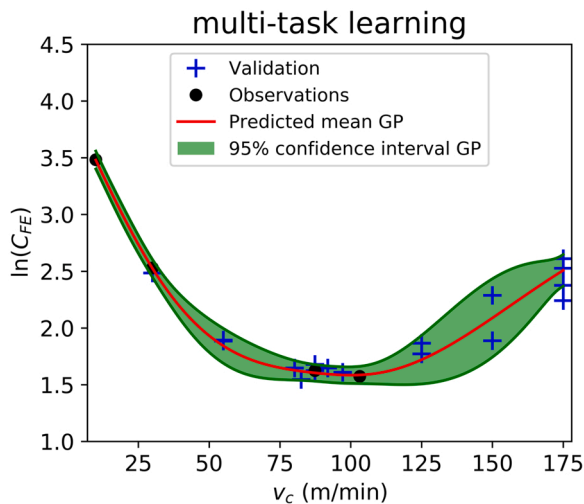


Fig. 10. Results for multi-task learning (Model 4) after 4 experiments.

equation fit is not very good, the data is explained well by the Gaussian process model using the costs calculated by the Taylor equation as a mean function. Therefore, the deviation between the costs predicted by the Taylor equation and the costs determined experimentally is modeled accurately by the Gaussian process model. Moreover, compared to the standard approach, the data-efficiency of the optimization is improved by using a Gaussian process model with a mean function based on the Taylor equation. Similarly to the previous approaches, some of the validation points are outside the 95% confidence interval for cutting speeds of 175 m/min. This is again a result of the Gaussian likelihood, assuming identical noise for all cutting speeds. However, for this case, the model predicts the data very well near the optimum because more data is available in this range, whereas a cutting speed of 175 m/min is only sampled once.

The last investigated approach is multi-task learning (Model 4), as shown in Fig. 10. As before, the multi-task learning case is started with two initial experiments at 10 m/min and 30 m/min. In addition to these two starting points, the multi-task model incorporates the results of the previous measurements with the D10 and D60 coatings. As a consequence, the algorithm requests points close to the optimum and reaches convergence after 4 experiments. After these 4 experiments the model is able to predict the data very well, especially close to the optimum. Only the variation at high cutting speed is again slightly underestimated due to the assumption of Gaussian noise that is modeled independently of

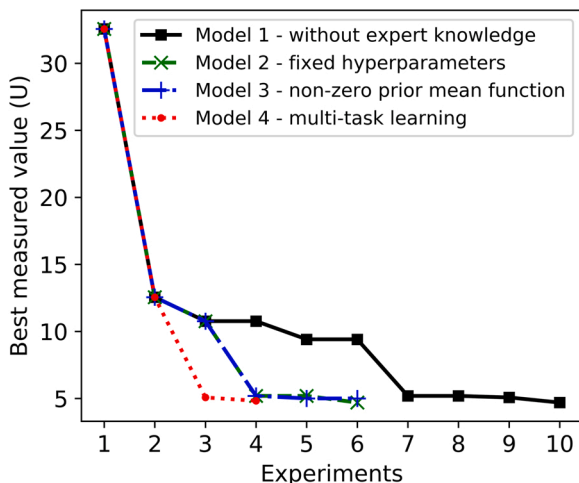


Fig. 11. Performance comparison of different Gaussian process models.

the cutting speed. Note that the model's results for the D30 coating are very similar to the results of the D60 coating. This similarity is exploited by the multi-task learning approach, leading to a fast convergence.

Fig. 11 shows a comparison of all tested models. It can be seen that the standard Gaussian process model without expert knowledge (Model 1) starts to sample close to the optimum only after 7 experiments and converges after 10 experiments. The performance of the Gaussian process model with fixed hyperparameters (Model 2) and the non-zero prior mean function (Model 3) behave very similar. They start to sample parameters close to the optimum after 4 experiments and converge after 6 experiments. The best performance is achieved with the multi-task learning approach (Model 4), which first samples a parameter close to the optimum after 3 experiments and converges after 4 experiments. In summary, the three tested methods that include expert knowledge or share knowledge are suited to significantly improve the sample efficiency compared to the standard Gaussian process model.

5. Conclusion

In this study, different methods to include expert knowledge and transfer knowledge were investigated and compared to a standard data-driven approach for the minimization of the production costs by adjusting the cutting speed in turning. For all demonstrated cases, the combination of Gaussian process models with Bayesian optimization proved successful for modeling of the optimization objective and for experiment selection. A direction for future research is to investigate tailored likelihood functions to consider the different noise levels for different cutting speeds. Expert knowledge and transfer learning were introduced by specifying Gaussian process hyperparameters a priori, using an empirical mean function based on the Taylor equation for the Gaussian process models, and by utilizing multi-task learning. While the model with an empirical mean function does not need additional data from related tasks, the approaches based on a model with fixed hyperparameters and multi-task learning model require data from previous experiments. Thus, the fixed hyperparameters and multi-task approach are especially useful if data from previous similar experiments is available. All approaches using expert knowledge or transfer learning reduced the number of experiments to find optimal cutting speeds by at least 40% compared to the standard approach, confirming that the presented approaches improve the sample efficiency of parameter selection. Future work is necessary to investigate the data efficiency and transferability of data for other manufacturing cases. Furthermore, the incorporation of expert knowledge and transfer of knowledge is not limited to the demonstrated methods. For example, it might be possible to also use prior knowledge to select the initial experiments.

CRedit authorship contribution statement

Markus Maier: Conceptualization, Methodology, Software, Investigation, Validation, Visualization, Writing – original draft, Writing – review & editing. **Hannes Kunstmann:** Software, Investigation, Validation, Writing – review & editing. **Ruben Zwicker:** Investigation, Writing – review & editing. **Alisa Rupenyan:** Resources, Project administration, Writing – review & editing. **Konrad Wegener:** Supervision, Writing – review & editing.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Competing Interest

The authors report no declarations of interest.

Appendix

Equations for multi-task learning

In this section the calculation of $\underline{K}(\underline{X}, \underline{X})$ and \underline{K}_{x_*} is shown based on Alvarez et al. (2012). The Eq. (11) can be written componentwise as follows,

$$(K(x, x'))_{d,d'} = \sum_{q=1}^Q b_{d,d'}^q k_q(x, x') \quad (\text{A.1})$$

where $b_{d,d'}^q$ are the coefficients of the coregionalization matrix \underline{B}_q modeling the covariance between the different outputs $f_d(x)$, and $k_q(x, x')$ are the kernels modeling the covariance between different process parameter points. For D different outputs and N data points per output, The matrix $\underline{K}(\underline{X}, \underline{X})$ of dimension $ND \times ND$, is calculated as follows,

$$\underline{K}(\underline{X}, \underline{X}) = \begin{pmatrix} (\underline{K}(\underline{X}_1, \underline{X}_1))_{1,1} & \cdots & (\underline{K}(\underline{X}_1, \underline{X}_D))_{1,D} & \cdots & (\underline{K}(\underline{X}_D, \underline{X}_1))_{D,1} & \cdots & (\underline{K}(\underline{X}_D, \underline{X}_D))_{D,D} \end{pmatrix} \quad (\text{A.2})$$

where \underline{X}_d is the input training data for output d . By using equation (A.1), the blocks $(\underline{K}(\underline{X}_d, \underline{X}_{d'}))_{d,d'}$ of the matrix $\underline{K}(\underline{X}, \underline{X})$ can be calculated as follows.

$$(\underline{K}(\underline{X}_d, \underline{X}_{d'}))_{d,d'} = \begin{pmatrix} (K(x_1, x_1))_{d,d'} & \cdots & (K(x_1, x_N))_{d,d'} \\ \vdots & \ddots & \vdots \\ (K(x_N, x_1))_{d,d'} & \cdots & (K(x_N, x_N))_{d,d'} \end{pmatrix} \quad (\text{A.3})$$

The matrix \underline{K}_{x_*} of dimension $D \times ND$ and entries $(\underline{K}(x_*, x_j))_{d,d'}$ can be calculated as follows.

$$\underline{K}_{x_*} = \begin{pmatrix} (\underline{K}(x_*, x_j))_{1,1} & \cdots & (\underline{K}(x_*, x_j))_{1,D} \\ \vdots & \ddots & \vdots \\ (\underline{K}(x_*, x_j))_{D,1} & \cdots & (\underline{K}(x_*, x_j))_{D,D} \end{pmatrix} \quad (\text{A.4})$$

$$(\underline{K}(x_*, x_j))_{d,d'} = \begin{pmatrix} (K(x_*, x_1))_{d,d'} & \cdots & (K(x_*, x_N))_{d,d'} \end{pmatrix} \quad (\text{A.5})$$

References

- Abbas, A.T., Hamza, K., Aly, M.F., Al-Bahkali, E.A., 2016. Multiobjective optimization of turning cutting parameters for J-steel material. *Adv. Mater. Sci. Eng.*, 6429160
- Alvarez, M.A., Rosasco, L., Lawrence, N.D., 2012. Kernels for vector-valued functions: a review. *Found. Trends Mach. Learn.* 4, 195–266.
- Aramesh, M., Shi, B., Nassef, A.O., Attia, H., Balazinski, M., Kishawy, H.A., 2013. Meta-modeling optimization of the cutting process during turning titanium metal matrix composites (Ti-MMCs). *Procedia CIRP* 8, 576–581.
- Cohn, T., Specia, L., 2013. Modelling annotator bias with multi-task Gaussian processes: an application to machine translation quality estimation. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32–42.
- Diametal, A.G., 2017. Precision turning tools catalog 5, Biel/Bienne, Switzerland.
- Essien, A., Giannetti, C., 2020. A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. *IEEE Trans. Ind. Inform.* 16, 6069–6078.
- Fan, S.-K.S., Hsu, C.-Y., Jen, C.-H., Chen, K.-L., Juan, L.-T., 2020. Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes. *Adv. Eng. Inform.* 46, 101166.
- Gardner, J.R., Kusner, M.J., Xu, Z.E., Weinberger, K.Q., Cunningham, J.P., 2014. Bayesian optimization with inequality constraints. In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 937–945.
- Goldberg, P.W., Williams, C.K.I., Bishop, C.M., 1997. Regression with input-dependent noise: a Gaussian process treatment. *Adv. Neural Inf. Process. Syst.* 10, 493–499.
- GPy, since 2012. GPy: A Gaussian process framework in python. [cited 04.08.2021]; Available from: (<http://github.com/SheffieldML/GPy>).
- Guidetti, X., Rupenyan, A., Fassl, L., Nabavi, M., Lygeros, J., 2021. Sample-efficient plasma spray process configuration with constrained bayesian optimization. *arXiv Prepr. arXiv 2103.13881*.
- Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z., 2014. Predictive entropy search for efficient global optimization of black-box functions, *Advances in Neural Information Processing Systems*.
- Homami, R.M., Tehrani, A.F., Mirzadeh, H., Movahedi, B., Azimifard, F., 2014. Optimization of turning process using artificial intelligence technology. *Int. J. Adv. Manuf. Technol.* 70, 1205–1217.
- Journal, A.G., Huijbregts, C.J., 1978. *Mining Geostatistics*. Academic Press, London.
- Khamel, S., Ouelaa, N., Bouacha, K., 2012. Analysis and prediction of tool wear, surface roughness and cutting forces in hard turning with CBN tool. *J. Mech. Sci. Technol.* 26, 3605–3616.
- Klocke, F., 2011. *Manufacturing Processes 1: Cutting*. Springer, Berlin, Heidelberg, Germany.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 1097–1105.
- Kushner, H.J., 1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Eng.* 86, 97–106.
- Maier, M., Rupenyan, A., Bobst, C., Wegener, K., 2020. Self-optimizing grinding machines using Gaussian process models and constrained Bayesian optimization. *Int. J. Adv. Manuf. Technol.* 108, 539–552.
- Maier, M., Zwicker, R., Akbari, M., Rupenyan, A., Wegener, K., 2019. Bayesian optimization for autonomous process set-up in turning. *CIRP J. Manuf. Sci. Technol.* 26, 81–87.
- Misaka, T., Herwan, J., Ryabov, O., Kano, S., Sawada, H., Kasashima, N., Furukawa, Y., 2020. Prediction of surface roughness in CNC turning by model-assisted response surface method. *Precis. Eng.* 62, 196–203.
- Mockus, J., Tiesis, V., Zilinskas, A., 1978. The application of Bayesian methods for seeking the extremum. In: Dixon, L.C.W., Szegő, G.P. (Eds.), *Towards Global Optimisation 2*. North-Holland, Amsterdam, pp. 117–129.
- Muñoz-González, L., Lázaro-Gredilla, M., Figueiras-Vidal, A.R., 2011. Heteroscedastic Gaussian process regression using expectation propagation, 2011 IEEE International Workshop on Machine Learning for Signal Process. 1–6.
- Nian, C.Y., Yang, W.H., Tarn, Y.S., 1999. Optimization of turning operations with multiple performance characteristics. *J. Mater. Process. Technol.* 95, 90–96.
- Nocedal, J., Wright, S.J., 2006. *Numerical Optimization (2nd edition)*. Springer, New York, NY.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., 2011. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Rasmussen, C.E., Williams, C.K.I., 2006. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts.
- Rencher, A.C., Christensen, W.F., 2012. *Methods of Multivariate Analysis (3rd edition)*. John Wiley & Sons, Hoboken, New Jersey.
- Sadek, A., Aly, M., Hamza, K., Meshreki, M., Nassef, A.O., Attia, H., 2015. Optimization of cutting conditions in vibration assisted drilling of composites via a multi-objective EGO implementation. In: *Proceedings of the ASME 2015 International Design*

- Engineering Technical Conferences and Computers and Information in Engineering Conference.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., Freitas, Nd, 2016. Taking the human out of the loop: a review of Bayesian optimization. Proc. IEEE 104, 148–175.
- Srinivas, N., Krause, A., Kakade, S., Seeger, M., 2010. Gaussian process optimization in the bandit setting: no regret and experimental design. In: Proceedings of the 27th International Conference on Machine Learning, pp. 1015–1022.
- Yang, W.H., Tarn, Y.S., 1998. Design optimization of cutting parameters for turning operations based on the Taguchi method. J. Mater. Process. Technol. 84, 122–129.