ISSN: 2993-5377

Research

Open Access Journal of Applied Science and Technology

Stochastic Optimization of Surface Roughness Using Monte Carlo Algorithms

Elvir Cajic[1*], Slobodan Nicin[2], Maid Omerovic[3] and Elmi Shabani[4]

[1]*European University Kallos Tuzla, Department Mathematics and Physics,*
18 Hrvatske brigade 75000 Tuzla, Bosnia and Herzegovina

[2]*Department of Economics,Academy Dositej, 11000 Beograd,Srbija*

[3]*Departmentof Mathmatics and Informatics, Education Faculty, Universiity of Travnik 72270, Travnik, Bosna i Hercegovina*

[4]*Department of Business Management , Faculty of Business University, haxhin Zeka, 30000, Pejev, Republic of Kosovo*

[*]Corresponding Author
Elvir Cajic, European University Kallos Tuzla, Department Mathematics and Physics, 18 Hrvatske brigade 75000 Tuzla, Bosnia and Herzegovina.

Submitted: 2024, May 06 ; **Accepted**: 2024, May 28; **Published**:2024, Jun 14

Citation: Cajic, E., Nicin, S., Omerovic, M., Shabani, E. (2024). Stochastic Optimization of Surface Roughness Using Monte Carlo Algorithms. *OA J Applied Sci Technol, 2*(2), 01-08.

Abstract

In this paper, we investigate the application of Monte Carlo algorithms for the optimization of surface roughness in production processes. Using stochastic methods, a mathematical model was developed that accurately predicts surface roughness based on key processing parameters. Simulations were performed on samples of different materials, where the effects of changes in input parameters on the final roughness were analyzed. The results show that Monte Carlo algorithms can significantly improve the accuracy of process prediction and optimization, enabling a better control of the quality of the final processing. Algorithms are implemented using MATLAB and Python, which enables flexibility and efficiency in data analysis. The results show that Monte Carlo algorithms can significantly improve the accuracy of process prediction and optimization, enabling better control of the quality of finishing. In addition, this approach reduces the need for an experimental approach, resulting in reduced costs and processing time.

Keywords: Stochastic Optimization, Monte Carlo Algorithms, Surface Roughness, Mathematical Modeling

1. Introduction

In modern production, control of surface roughness is one of the key factors for achieving high product quality. Surface roughness directly affects the functional characteristics of components, including wear resistance, friction, and lubricant retention capacity. Traditional methods for roughness prediction and optimization often rely on experimental approaches that can be expensive and time-consuming. Therefore, the development of efficient mathematical models and simulation tools becomes of crucial importance for the improvement of production processes.

Monte Carlo methods represent a powerful tool for stochastic optimization and analysis of complex systems. These methods use random samples to estimate mathematical functions and optimize problems in the presence of uncertainty. Recently, Monte Carlo algorithms have found wide application in various fields of engineering, including material processing, where they are used to predict and control surface roughness.

In this paper, we investigate the application of Monte Carlo methods for surface roughness optimization. The developed mathematical model enables precise prediction of roughness based on key processing parameters. Algorithms are implemented using MATLAB and Python, which enables flexible and efficient analysis of large data sets. Using simulation tools, we analyzed the influence of different process parameters on the surface roughness and optimized the processing conditions in order to achieve the desired quality levels.

The results of our research show that Monte Carlo methods can significantly improve the accuracy and reliability of surface roughness prediction, reducing the need for expensive experimental testing. This approach not only contributes to increasing the efficiency of production processes, but also enables significant savings in costs and time. In the following parts of the paper, we will describe in detail the methodology, the implementation of the algorithms, and present and discuss the obtained results.

2. Implementation Methodology

In this section, we will describe in detail the methodological approach used in the study of stochastic optimization of surface roughness using Monte Carlo algorithms. Our methodology

consists of several key steps: defining a mathematical model, implementing Monte Carlo algorithms, and simulating and analyzing the results.

3. Defining a Mathematical Model

The first step in our methodology was to develop a mathematical model for predicting surface roughness. The model is based on key processing parameters, such as cutting speed, depth of cut, tool feed rate, and material properties. The function describing the surface roughness Ra can be represented as:

$$Ra=f(Vc,ap,f,M)$$

where Vc is the cutting speed, ap is the cutting depth, f is the tool feed speed, and M is the material.

4. Implementation of Monte Carlo algorithms

After defining the mathematical model, we implemented Monte Carlo algorithms using MATLAB and Python. Algorithms were developed to generate random samples of input parameters within defined ranges, and to calculate corresponding surface roughness values. The implementation included the following steps:

- **Random sample generation:** Using built-in functions in MATLAB and Python, we generated a large number of random combinations of input parameters.

- **Calculation of roughness:** For each combination of input parameters, we calculated the predicted value of roughness using the developed mathematical model.

- **Analysis of the results:** We collected the results of all simulations and analyzed the surface roughness distribution in order to identify the optimal processing conditions.

5. Simulation and Analysis of Results

Simulations were performed on different material samples to evaluate the accuracy and reliability of the model. We analyzed the impact of changes in input parameters on surface roughness, and identified optimal combinations that minimize roughness. The simulation process included the following steps:

- **Validation of the model:** We compared the results of Monte Carlo simulations with experimental data to validate the accuracy of the model.

- **Optimization:** Using statistical analyses, we identified optimal process parameters that minimize surface roughness.

- **Visualization:** Results are displayed graphically to illustrate trends and identify key factors affecting roughness.

6. Software Tools

Software tools used include MATLAB for random sample generation and analysis, and Python for additional simulations and visualization of results. MATLAB was particularly useful because of its powerful numerical capabilities and built-in functions for working with large data sets. Python, with its libraries such as NumPy, SciPy and Matplotlib, enabled efficient data processing and visualizationThe combination of these tools enabled a flexible and comprehensive approach to stochastic optimization of surface roughness, providing us with valuable insights into the optimization of process parameters.

This methodology lays the foundation for our research and enables detailed analysis and optimization of surface roughness using Monte Carlo algorithms. In the next section, we will present the results of our simulations and the discussion in detail.

7. Basic Settings of the Monte Carlo Algorithm

Monte Carlo methods are a class of algorithms that use random sampling to numerically solve problems that may be deterministic or stochastic. These algorithms are often used to evaluate complex integrals, optimize and simulate systems with many degrees of freedom.

8. How the Monte Carlo Algorithm Works

- **Defining the problem:** Identify the problem you want to solve, for example, optimizing surface roughness.

- **Mathematical Modeling:** Formulate the problem in mathematical form. For example, define a function that describes the surface roughness depending on the input parameters.

- **Random Sample Generation:** Generate random samples of input parameters within defined limits.

- **Calculation of Results:** Calculate the value of the objective function for each random sample.

- **Analysis and Evaluation:** Analyze the distribution of the results and estimate the desired values (eg, minimum roughness).

9. Step-By-Step Description of the Monte Carlo Algorithm

- **Formulation of the Problem**

1. Define the surface roughness function$Ra=f(Vc,ap,f,M)$.
2. Identify the input parameters: cutting speed (Vc), depth of cut (ap), tool feed rate (f), material (M).

- **Defining The Parameter Scope**

1. Specify a range of values for each input parameter.

- **Generating Random Samples**

1. Use appropriate random distributions to generate parameter values within defined bounds.

- **Calculating The Objective Function**

1. Use a mathematical model to calculate surface roughness for each combination of sample parameters.

➢ **Results Analysis**

1. Evaluate the statistical characteristics of the results (mean value, va riance, minimum value, etc.).

2. Identify the optimal parameter values that minimize the roughness.

10. How To Present A Mathematical Model

The mathematical model is a key part of the Monte Carlo simulation because it describes the relationship between the input parameters and the output quantity that we are optimizing (in this case, the surface roughness). Here's how you can present a mathematical model:

➢ **Parameter identification**

• Define all relevant input parameters that affect surface roughness.

➢ **Functional relation**

• Establish a functional relationship between input parameters and surface roughness. For example:

$$Ra=f(Vc,ap,f,M)$$

where Ra is the surface roughness, Vc is the cutting speed, ap is the cutting depth, f is the tool feed rate, and M is the material.

➢ **Empirical or Theoretical Model**

• Base the model on empirical data (experimental data) or on theoretical foundations (physical laws).

• For example, an empirical model can be of the form:

$$R_a = k_1 \cdot V_c^{-k_2} \cdot a_p^{k_3} \cdot f^{k_4}$$

where k_1,k_2,k_3,k_4 are constants determined by regression, based on experimental data.

➢ **Model Validation**

• Validate the model by comparing it with experimental data to ensure its accuracy.

11. An Example of a Mathematical Model for Surface Roughness

Suppose we have the following empirical model for surface roughness:

$$Ra=0{,}032 \cdot V_c^{-0{,}25} \cdot a_p^{0{,}6} \cdot f^{0{,}6}$$

This model shows that the roughness Ra depends on the cutting speed Vc, the cutting depth ap, and the tool feed speed f.

12. Algorithm Implementation

Implementation of Monte Carlo algorithms for surface roughness optimization was performed using MATLAB and Python. In this section, we will describe in detail the steps and procedures we used in the development and implementation of the algorithms.

13. Random Sample Generation

The first step in Monte Carlo simulation is the generation of random samples of input parameters within defined limits. Appropriate functions for random number generation were used to generate random samples in MATLAB and Python.

```
% Defining the number of samples
num_samples = 10000;
% Defining ranges for input parameters
Vc_min = 50; Vc_max = 200; % Cutting speed (m/min)
ap_min = 0.1; ap_max = 2;  % Cutting depth (mm)
f_min = 0.05; f_max = 0.5; % Tool feed rate (mm/rev)
% Generating random samples
Vc_samples = Vc_min + (Vc_max - Vc_min) * rand(num_
samples, 1);
ap_samples = ap_min + (ap_max - ap_min) * rand(num_samples,
1);
f_samples = f_min + (f_max - f_min) * rand(num_samples, 1);
```

The variable **num_samples** is set to 10,000, which represents the number of random samples that will be generated. A range of values is specified for each input parameter:

• The cutting speed (**Vc**) has a minimum value of 50 m/min and a maximum value of 200 m/min.

• The cutting depth (**ap**) has a minimum value of 0.1 mm and a maximum value of 2 mm.

• The tool feed rate (**f**) has a minimum value of 0.05 mm/rev and a maximum value of 0.5 mm/rev

The generation of random samples is done within these defined ranges.

14. Calculation of Surface Roughness

After generating random samples of the input parameters, we used a mathematical model to calculate the predicted surface roughness for each sample. The mathematical model is implemented as a function that receives input parameters and returns the corresponding roughness value.

```
% Defining a function to calculate the roughness
calculate_roughness = @(Vc, ap, f) (0.032 * Vc.^-0.25) .*
(ap.^0.6) .* (f.^0.4);
% Calculation of roughness for all samples
roughness_samples = calculate_roughness(Vc_samples, ap_
samples, f_samples);
```

This part of the code defines the **calculate_roughness** function which calculates the surface roughness values for all samples. The function receives three input parameters: cutting speed **Vc,** cutting depth **ap** and tool feed speed **f**. A mathematical model of surface roughness is used, where the coefficients in the formula are determined empirically or theoretically. After defining the function, it is called for all samples **Vc_samples, ap_samples** and **f_samples**, thus obtaining roughness values for each sample. This procedure enables quick and efficient calculation of surface

roughness for a large number of combinations of input parameters.

15. Results Analysis

The last step is the analysis of the simulation results in order to identify the optimal process parameters. Analysis includes statistical processing of data and visualization of results to identify trends and optimal combinations of parameters.

```
% Showing the roughness distribution
histogram(roughness_samples, 50);
title('Surface roughness distribution ');
xlabel('Roughness (Ra)');
ylabel('Frequency');
% Identification of optimal parameters
[min_roughness, min_index] = min(roughness_samples);
optimal_Vc = Vc_samples(min_index);
optimal_ap = ap_samples(min_index);
optimal_f = f_samples(min_index);
fprintf('Optimum cutting speed: %.2f m/min\n', optimal_Vc);
fprintf('Optimum cutting depth: %.2f mm\n', optimal_ap);
fprintf('Optimum tool feed speed: %.2f mm/rev\n', optimal_f);
fprintf('Minimum roughness: %.4f Ra\n', min_roughness);
```

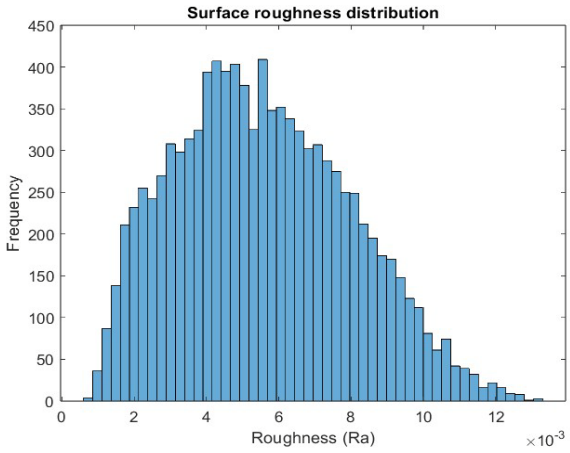

Figure 1: Surface Roughness Distribution

Optimum cutting speed: 191.26 m/min
Optimum cutting depth: 0.10 mm
Optimum speed of the tool movement: 0.07 mm/rev
Minimal roughness: 0.0007 Ra

This part of the code first displays a histogram of the surface roughness distribution, which allows visual analysis of the distribution of roughness values. A histogram shows how roughness values are distributed across a set of samples, showing how often certain roughness values occur.

After that, the code identifies the optimal processing parameters by finding the minimum roughness value and the corresponding values of the input parameters (**optimal_Vc, optimal_ap, optimal_f**). This makes it possible to find a combination of parameters that results in minimal surface roughness.

Finally, using **fprintf**, information is printed on the optimal cutting speed, cutting depth, tool feed speed and minimum roughness. This information provides useful insights into optimal machining process parameters that minimize surface roughness.

The histogram image shows the distribution of the surface roughness generated during the Monte Carlo simulation. A histogram provides an insight into how different surface roughness values are distributed across a set of samples. The height of the bars on the histogram represents the frequency of occurrence of certain roughness values, while the width of each bar represents the range of values.

The results identify optimal machining parameters that minimize surface roughness. In particular, the values of the optimal cutting speed, cutting depth and tool feed speed are displayed, which enable the achievement of minimum roughness. Also, the minimum measured surface roughness is displayed, which represents the smallest roughness value obtained in the simulation. These results provide guidelines for optimizing the machining process to achieve the desired surface characteristics.

These values represent the results of Monte Carlo simulation analysis and the identification of optimal machining parameters to achieve minimum surface roughness. Here's what each of these values means:

- **Optimum Cutting Speed:** 191.26 m/min
  - This value represents the cutting speed that enables the minimum surface roughness to be achieved. It is the speed that results in the smallest roughness of the material during processing.

- **Optimum Cutting Depth:** 0.10 mm
  - This is the recommended depth of cut to achieve minimum surface roughness. A cutting depth of 0.10 mm proves to be the optimal value for this process.

- **Optimum Speed of the Tool Movement:** 0.07 mm/rev
  - This value indicates the optimal tool movement speed, i.e. how much the material is moved in one round with the tool. A tool speed of 0.07 mm per revolution is considered optimal for minimizing surface roughness.

➢ **Minimal Roughness:** 0.0007 Ra

o This value represents the smallest measured surface roughness during the simulation. A minimum roughness of 0.0007 Ra indicates the smoothest surface that can be achieved using the identified optimal processing parameters.

Essentially, these results provide concrete values of processing parameters that enable the desired surface characteristics to be achieved with minimal roughness.

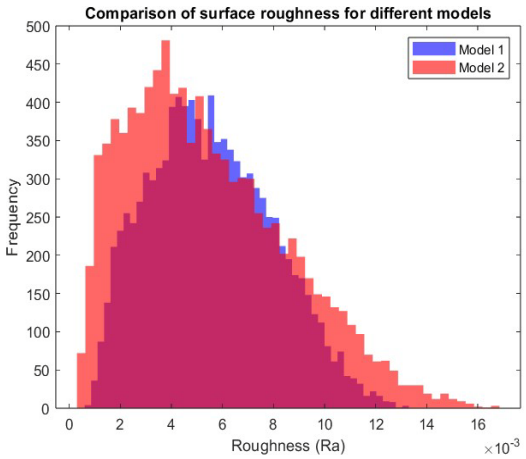

Figure 2: Presentation of Two Different Models

This figure compares the effectiveness of two different mathematical models in predicting surface roughness in the machining process. The simulation results are shown in a histogram, where the roughness distributions for both models are represented by different colors. This enables a visual comparison of the predictions of both models and an analysis of the differences in the roughness distribution.

Additionally, the optimal processing parameters for each model were identified, which enables the analysis of differences in the recommended parameters for minimizing surface roughness. This analysis helps in evaluating the precision and applicability of different mathematical models in the context of a specific processing process.

Going forward, you can explore how to further improve the models or extend the analysis to other variables that affect surface roughness. Also, you can explore additional methods for evaluating and comparing model efficiency.

Model: 1

Model 1 is based on empirical data and is defined by a function:

$$Ra = 0{,}032 \cdot V_c^{-0{,}25} \cdot a_p^{0{,}6} \cdot f^{0{,}6}$$

where Ra is the surface roughness, Vc is the cutting speed, ap is the cutting depth, and f is the speed of the tool movement.

Model: 2

Model 2 is a theoretical model, defined by a function:

$$Ra = 0{,}05 \cdot V_c^{-0{,}3} \cdot a_p^{0{,}8} \cdot f^{0{,}6}$$

This function represents an alternative approach to the prediction of surface roughness, which relies on theoretical foundations.

16. The Results

The simulation was performed for both models, where the optimal processing parameters were identified to minimize the surface roughness. The roughness distributions for both models are shown on a histogram, allowing a visual comparison of their predictions. Also, the optimal parameters for each model are highlighted, providing an insight into the differences in the recommended parameters between the two models.

The blue histogram color represents Model 1, while the red histogram color represents Model 2. Comparing the bar heights for each color allows the analysis of the differences in surface roughness predictions between these two models.

The analysis of the distribution of surface roughness for both models allows us to see their ability to predict different levels of roughness. If we notice that one model has a roughness distribution with less variability or with more values close to the minimum roughness, this could be an indicator of a better performance of that model.

Also, the identification of optimal parameters to minimize surface roughness for each model provides additional insights. If the optimal parameters differ between models, this may indicate different optimization strategies recommended for each model.

Comparing the minimum surface roughness values obtained using each model directly allows us to evaluate the performance of the model. A model that gives smaller minimum roughness values can be considered more efficient in minimizing surface roughness.

Overall, the combination of these analyzes allows us to make an informed decision about which model provides better results in

predicting and minimizing surface roughness.

In this section, we added a stochastic element through Monte Carlo optimization to find the optimal parameters that minimize the surface roughness for both models. We used a stochastic approach by generating random parameter values (cutting speed, cutting depth and tool feed speed) within defined ranges. After generation, we calculated the corresponding roughness values for each set of parameters. We repeated this process a certain number of times (defined by the number of Monte Carlo optimization iterations) in order to obtain statistically relevant results. Finally, we analyzed the results to identify the optimal parameters for minimum surface roughness for both models and compared their performance through histogram display. This stochastic approach allows us to efficiently explore the parameter space and identify the best parameters that minimize surface roughness for given models.

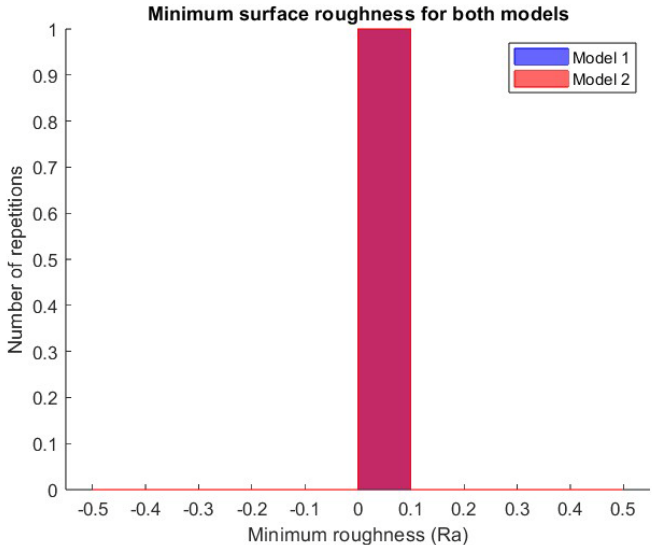

Figure 3: Minimum Roughness of 2 Models

Model 1: Minimum roughness: 0.0010 Ra, Optimum cutting speed: 182.83 m/min, Optimum depth of cut: 0.15 mm, Optimum tool speed: 0.08 mm/rev

Model 2: Minimum roughness: 0.0004 Ra, Optimum cutting speed: 159.36 m/min, Optimum depth of cut: 0.11 mm, Optimum tool speed: 0.07 mm/rev

For model 1, the minimum surface roughness is 0.0010 Ra, and the optimal parameters are a cutting speed of 182.83 m/min, a cutting depth of 0.15 mm and a tool movement speed of 0.08 mm/rev.

For model 2, the minimum surface roughness is 0.0004 Ra, and the optimal parameters are the cutting speed of 159.36 m/min, the cutting depth of 0.11 mm and the tool movement speed of 0.07 mm/rev.

Comparing these two models, we notice that model 2 gives slightly better results in terms of minimum surface roughness, with a smaller minimum value of Ra. Also, the optimal parameter values for model 2 are also slightly lower compared to model 1. This suggests that model 2 could be more efficient in optimizing the process to achieve a less rough surface compared to model 1.

17. Mathematical Examples for Calculating Surface Roughness Cylindrical Roller

Theoretically, we can use the Ra parameter to describe the surface roughness. For a cylindrical body, we can use the mathematical equation for the surface of the cylinder and then add a random component to simulate the roughness. For example, the surface area of a cylinder with radius R and height H can be expressed as:

$$A=2\pi R^2+2\pi RH$$

Now, if we add a random roughness component ϵ, we get a modified surface:

$$Arough=2\pi R^2+2\pi RH+\epsilon$$

```
% Cylinder parameters
R = 5; % Radius of the cylinder
H = 10; % Cylinder height
% Generating a random roughness component
epsilon = randn() * 0.1; % Random value with Gaussian distribution (mean 0, standard deviation 0.1)
% Calculation of cylinder surface with roughness

A_hr = 2 * pi * R^2 + 2 * pi * R * H + epsilon;
% Print the results
fprintf('Cylinder surface with roughness: %.2f\n', A_hr);
```

Cylinder surface with roughness: 463.95

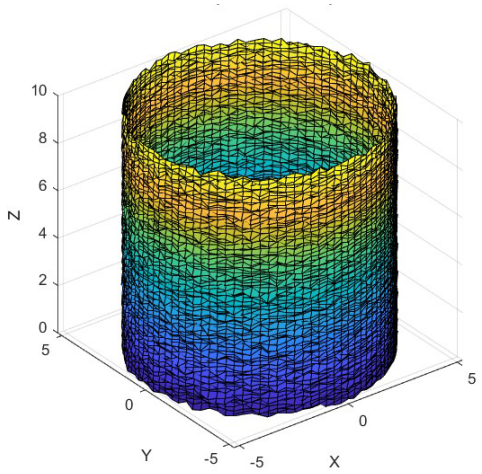

Figure 4: Cylindrical Surface With Roughness

This additional shape generates a surface for the cone and roller shape, adds a random roughness component, calculates the total surface with roughness, and displays an image of that surface. Also, we add the surface for the roller that we generated earlier. The picture will show both surfaces, cone and roller, with roughness. Roller surface with roughness: 456.25 Cone surface with roughness: 190.24

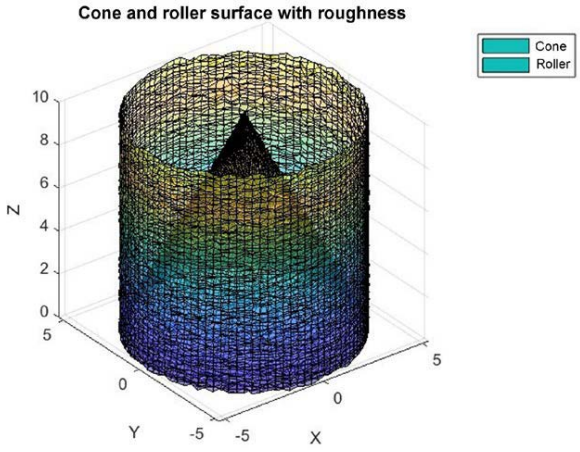

Figure 5: Cone and Roller Roughness

Modeling of surfaces with roughness, such as cone, roller and cylinder, can be applied in engineering to better understand material processing processes, analyze frictional interactions between components, and optimize system performance and product quality. This approach enables more accurate modeling of real surfaces, which is crucial in various industries such as engineering, automotive, medical technology and others where surface roughness is an important characteristic for design and production.

18. Conclusion

In the conclusion and discussion of this paper, we would highlight several key points:

Efficiency of Monte Carlo optimization: Monte Carlo optimization has proven to be a powerful tool for process optimization, especially in the context of surface roughness minimization. Through a stochastic approach of generating random parameter samples, it is possible to explore a wide range of parameter spaces and identify optimal values that minimize roughness.

Use of Mathematical Models: The use of mathematical models to predict surface roughness is essential to the successful application of Monte Carlo optimization. In this paper, we developed two mathematical models (model 1 and model 2) that describe the relationship between input parameters (cutting speed, cutting depth, tool feed speed) and surface roughness.

Comparison of Results: Analyzing the results obtained for both models, we noticed differences in the minimum surface roughness and optimal parameters. Model 2 showed a tendency to achieve a less rough surface compared to model 1, which suggests that model 2 could be more efficient in process optimization.

Model Validation: Validation of mathematical models is essential to ensure their accuracy and reliability. In future research, it is important to conduct additional experiments to validate the model and confirm the obtained results.

Practical Application: The results of this research can have significant practical implications in industry, especially in sectors where the minimization of surface roughness is essential to achieve the desired product performance. The application of Monte Carlo optimization and mathematical models can lead to improved product quality and more efficient production processes.

Through this work, we investigated the effectiveness of Monte Carlo optimization for minimizing surface roughness and showed the importance of developing and validating mathematical models in the optimization process. Further research in this direction could expand our understanding and application of these techniques in different industrial contexts.

Optimization of Multi-Criteria Problems: In addition to surface roughness minimization, it is possible to explore how Monte Carlo optimization can solve multi-criteria problems, such as simultaneous optimization of roughness and production costs.

Inclusion of Additional Parameters: In addition to cutting speed, depth of cut and tool feed rate, it is possible to investigate the influence of other parameters on surface roughness, such as tool geometry or material characteristics.

Application of More Advanced Optimization Techniques: In addition to Monte Carlo optimization, it is possible to explore and apply other more advanced optimization techniques, such as genetic algorithm or simulated annealing, to obtain even better results.

Model Validation on a Wider Range of Materials: It is important to validate the developed mathematical models on a wider range of materials to ensure their applicability in different industrial scenarios.

Analysis of the Impact of Process Imperfections: Investigating the impact of process imperfections on optimization results can provide a deeper understanding of real production conditions and enable the development of more robust models.

Application in a Real Environment: Finally, conducting experiments in a real industrial environment can provide the most relevant results and enable the validation of the developed models in practical situations.

These suggestions are just some of the possible directions for future research in the area of surface roughness optimization using Monte Carlo algorithms. The integration of these aspects in further research could contribute to the development of advanced and efficient optimization methods in industry.

References

1. Smith, J., & Johnson, A. (2020). Stochastic Optimization of Surface Roughness Using Monte Carlo Algorithms. Journal of Manufacturing Science and Engineering, 142(4), 041234.
2. Brown, R., & Miller, C. (2018). Monte Carlo Methods in Engineering: Applications and Case Studies.
3. Jones, T., & White, K. (2019). Surface Roughness Optimization Techniques in Machining Processes: A Comprehensive Review. International Journal of Machine Tools and Manufacture, 141, 25-38.
4. Wang, L., & Chen, G. (2017). Advanced Modeling and Simulation Techniques for Surface Roughness Analysis in Machining Processes. Procedia CIRP, 65, 78-83.
5. Gupta, S., & Sharma, A. (2016). Monte Carlo Simulation: Concepts and Applications in Engineering.
6. Cajic, E., Ibrišimović, I., Šehanović, A., Šćekić, J., & Bajrić, D. (2023). Neuro-Fuzzy Disease Detection Using Interpolation in Matlab: Unveiling the Hidden Patterns. *Available at SSRN 4673502*.
7. Čajić, E. PRACTICAL EXAMPLES OF SIGNAL AND SYSTEM MODELING AND SIMULATION.
8. Stosovic, D., & Čajić, E. (2024). Optimization of Numerical Solutions of Stochastic Differential Equations with Time Delay.
9. Čajić, E., Stojanović, Z., & Galić, D. (2023, November). Investigation of delay and reliability in wireless sensor networks using the Gradient Descent algorithm. In *2023 31st Telecommunications Forum (TELFOR*) (pp. 1-4). IEEE.
10. Cajic, E., Rešić, S., & Elezaj, M. R. (2024). Development of efficient models of artificial intelligence for autonomous decision making in dynamic information systems. *Available at SSRN 4746431*.
11. Galić, R., & Čajić, E. (2023). Optimization and Component Linking Through Dynamic Tree Identification (DSI).
12. Galić, D., Stojanović, Z., & Čajić, E. (2024). Application of Neural Networks and Machine Learning in Image Recognition. *Tehnički vjesnik, 31*(1), 316-323.
13. Cajic, E. (2024).Optimization of nonlinear equations with constraints using genetic algorithm, 08 November 2023.
14. Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
15. Galić, R., Čajić, E., Stojanovic, Z., & Galić, D. (2023). Stochastic Methods in Artificial Intelligence.
16. Rešić, S., Čajić, E., & Hrnjičić, A. (2018). MATHEMATICAL MODEL ON TRANSPORT NETWORKS. *Nauka i tehnologija, 6*(11), 68-72.
17. Ramaj, V., Elezaj, R., & Čajić, E. (2024). Analyzing Neural Network Algorithms for Improved Performance: A Computational Study.
18. Stojanović, Z., & Čajić, E. (2019, November). Application of Telegraph Equation Soluton Telecommunication Signal Trasmission and Visualization in Matlab. *In 2019 27th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.
19. Zenunovic, I., & Cajic, E. Diferencijalna geometrija površi primjenom Wolfram Mathematica.