

DEVELOPMENT OF A SOFTWARE APPLICATION FOR AUTOMATED CALCULATION OF CUTTING PARAMETERS IN MACHINING PROCESSES

Krasimir Kanev¹[0009-0007-6464-252X], Marin Zhilevski¹[0000-0002-4545-9718], Mikho Mikhov¹[0000-0002-7768-5525]

Oleg Krol²[0000-0003-0193-2750], Vladimir Sokolov²[0000-0003-0459-1824]

¹Technical University of Sofia, 8 Kliment Ohridski Blvd, Sofia 1000, Bulgaria

²Volodymyr Dahl East Ukrainian National University, 17 Ioanna Pavla II str., Kyiv 01042, Ukraine
Email: krols@i.ua

Abstract - The selection of optimal cutting parameters is one of the most important stages in the planning of the machining process, as it directly affects the quality of the machining, productivity, tool life and production costs. Conventional engineering calculations are often performed manually using machining manuals and manufacturers' recommendations, which slows down the process and increases the possibility of calculation errors. This article presents the development of an Android-based software application designed for automated calculation and, hence, optimization of cutting parameters for drilling, boring and milling operations. The application was developed using Java in the Android Studio environment and includes analytical models for calculating spindle speed, feed rate, cutting power, torque and other important machining parameters. A modular software architecture is implemented to ensure flexibility and easy integration of additional machining operations. The developed software performs automatic verification of input data, performs the necessary calculations in real time and presents the results via a graphical user interface. Compared to the conventional manual calculation procedure, the proposed solution significantly reduces the time required for process planning, while minimizing the likelihood of operator errors. The developed software represents a practical engineering tool to support machining process planning and demonstrates the potential of mobile applications for digital transformation in manufacturing engineering. Unlike conventional machine calculators, the proposed software offers a unified computational framework for multiple machine operations within a single mobile platform.

Keywords: Android application, Machining processes, Cutting parameters, Drilling, Boring, Milling, Manufacturing engineering, Process planning.

1. Introduction

The development of modern manufacturing technologies and the transition toward Industry 4.0 have significantly changed contemporary machining processes. The increasing integration of digital technologies into manufacturing systems has created new opportunities for process monitoring, planning, optimization, and decision support [1]. The Industrial Internet of Things (IIoT) and Digital Twins have become important components of this transition [2]. The integration of Digital Twins with artificial intelligence further extends the possibilities for advanced manufacturing applications [3]. These developments have increased the need for effective

computational approaches that can support technological decision-making during machining.

Among the numerous factors affecting machining performance, the appropriate selection of cutting parameters remains an important task because it directly influences machinability, surface quality, productivity, energy consumption, and production costs. Machinability evaluation is important for understanding and improving machining performance and for selecting appropriate process conditions [4]. Sustainable CNC machining requires consideration of process conditions and their influence on energy consumption and resource efficiency [5]. Energy efficiency is also strongly influenced by the selection of appropriate machining parameters [6]. Surface quality represents another

important consideration in machining, particularly during finishing operations [7]. These aspects demonstrate that the determination of appropriate machining parameters requires consideration of several technological factors rather than a single process variable.

Traditionally, cutting parameters are determined using analytical equations, machining manuals, manufacturers' recommendations, and accumulated practical experience. Analytical modelling approaches remain relevant for describing machining processes and establishing relationships between technological parameters [8]. A major advantage of analytical approaches is that their calculation procedures are transparent and can be directly interpreted and verified by an engineer. Analytical models also provide a relatively straightforward basis for implementing computational tools because the relationships between input parameters and calculated machining characteristics are explicitly defined.

Despite these advantages, conventional analytical calculations normally require the user to select the appropriate equations, obtain the necessary technological data, perform the calculations, and verify the obtained values against the limitations of the machine tool and cutting tool. When several machining operations or alternative parameter combinations have to be considered, this procedure can become repetitive and time-consuming. Manual data entry and repeated calculations may also increase the possibility of operator errors. Therefore, analytical methods provide a high degree of transparency, but their conventional manual implementation is less suitable when rapid calculation is required during machine setup or process adjustment.

Another important research direction is represented by optimization and artificial-intelligence-based approaches. AI-based optimization has demonstrated potential for addressing multiple machining objectives [9]. Optimization algorithms provide a broad range of computational methods that can be applied to manufacturing process optimization [10]. Intelligent machining approaches have also been increasingly integrated into CNC machine-tool systems [11].

The main advantage of AI- and optimization-based approaches is their ability to process large amounts of experimental or historical data and identify parameter combinations that may improve machining performance. Artificial intelligence, machine learning, and deep learning techniques have been applied in machining operations for prediction, monitoring, and optimization [12]. Historical machining data can also be used for intelligent adjustment of process parameters in batch machining [13]. A wide range of AI techniques has been investigated for machining parameter optimization, demonstrating the potential of data-

driven approaches for complex manufacturing problems [14].

However, AI- and optimization-based solutions also have important limitations. Their practical implementation may require sufficiently large and representative datasets, appropriate model training and validation procedures, and additional computational resources. In addition, data-driven models may provide less direct transparency than analytical equations, which can be important in engineering applications where the user needs to understand and verify the calculation procedure. AI-based approaches for adaptive manufacturing process monitoring also involve implementation challenges related to the integration of intelligent methods into practical manufacturing environments [15]. Consequently, AI- and optimization-based methods are not necessarily replacements for analytical calculations in every engineering application. Their suitability depends on the available data, optimization objectives, computational requirements, and required level of automation.

Computer-aided manufacturing (CAM) software represents another established approach for supporting machining process planning. Industrial software solutions increasingly provide comprehensive functionality for smart manufacturing and digital process planning [16]. CAM-based solutions can integrate several functions, including machining simulation, toolpath generation, NC program generation, tool selection, and process optimization. Dynamic toolpath generation and CAM-based process optimization have also been investigated for CNC machining [17].

The major advantage of CAM systems is their comprehensive functionality and their ability to integrate several stages of machining process planning within a single software environment. However, this broad functionality is also associated with limitations. CAM systems are generally designed for workstation-based use and are frequently integrated into complex CAD/CAM environments. Their implementation may require commercial software licenses, specialized training, and relatively complex configuration procedures. Consequently, although CAM systems are highly effective for comprehensive process planning, their functionality may be excessive when the immediate task is limited to the rapid calculation or verification of a relatively small number of cutting parameters during machine setup or process adjustment.

Mobile engineering applications provide another approach, particularly for rapid calculations directly in the production environment. Intelligent mobile applications incorporating advanced manufacturing-related functionality demonstrate the potential of mobile technologies for supporting manufacturing activities [18]. The implementation of Industry 4.0 technologies in manufacturing also emphasizes the

importance of selecting digital solutions according to specific manufacturing requirements and operational contexts [19]. Application-based approaches have additionally demonstrated the potential of mobile software as practical support tools for error detection and improvement in manufacturing operations [20].

The main advantages of mobile engineering solutions are portability, immediate availability, relatively simple graphical interfaces, and the possibility of performing calculations without direct access to a workstation-based engineering system. Modern mobile devices provide sufficient computational capabilities for many engineering calculations while offering convenient touch-based interaction. However, mobile engineering applications may also have limitations. Many solutions are designed as stand-alone calculators focused on individual equations or a limited number of technological operations. As a result, they may provide rapid numerical calculations but lack integrated input-data validation, a common computational architecture for several machining operations, and possibilities for systematic extension of the underlying analytical models.

From the perspective of the specific task addressed in this study, the mobile platform provides several relative advantages over the alternative approaches. Compared with conventional manual calculations, the application reduces repetitive computational work and provides immediate results while retaining the analytical basis of the calculations. Compared with AI- and optimization-based solutions, it does not require a training dataset or model development and can therefore be used with limited computational and data requirements. Compared with comprehensive CAM systems, the application is focused specifically on the rapid calculation of machining parameters and can be used independently of a workstation-based CAD/CAM environment. In comparison with stand-alone calculators, the proposed application integrates several machining operations within a common software architecture and incorporates input-data validation. These characteristics make the mobile platform particularly suitable for rapid preliminary calculation and verification of machining parameters in situations where a comprehensive CAM system or data-intensive optimization approach would be unnecessary.

The analysis of these approaches demonstrates that each solution provides specific advantages while also involving particular limitations. Conventional analytical calculations offer transparency and ease of verification, but their manual execution is repetitive and time-consuming. AI- and optimization-based methods provide advanced data-driven capabilities, but their implementation depends on suitable datasets, model development, and validation. CAM systems offer comprehensive process-planning

functionality, but their complexity and workstation-oriented implementation may be unnecessary for simple parameter calculations. Mobile applications provide portability and rapid interaction, but existing solutions may have limited technological scope and insufficient integration between different machining operations.

These observations indicate a practical need for an engineering solution that combines the transparency of analytical calculation methods with the accessibility and rapid operation of a mobile platform. Such a solution should also provide automatic input-data validation and support several machining operations within a unified environment. In response to this need, this paper presents the development of an Android-based software application for the automated calculation of cutting parameters in machining processes. The proposed system integrates analytical models for drilling, boring, and milling operations within a unified computational environment developed using Java and Android Studio. The application allows the user to enter the required technological parameters, automatically validates the input data, selects the appropriate analytical relationships, and calculates key machining characteristics, including spindle speed, feed rate, cutting power, machining time, and torque.

The modular software architecture enables the individual calculation modules to be maintained and extended without changing the overall structure of the application. The proposed solution is therefore intended as a complementary engineering tool for rapid calculation and preliminary selection of machining parameters, particularly in situations where a full CAD/CAM environment or a data-intensive AI-based optimization system is not required. The application is not intended to replace comprehensive CAM or AI-based optimization systems, but rather to provide an accessible and lightweight solution for analytical calculation and preliminary technological assessment during machining process planning.

2. Materials and Methods

2.1. Analytical Framework for Cutting Parameter Calculation

The developed software uses analytical relationships for calculating cutting parameters for drilling, milling, and boring operations. For drilling, the calculation procedure incorporates relationships associated with the electric-drive selection methodology for drilling machines [21]. The corresponding relationships for milling are based on the spindle-drive selection methodology described in [22], while the boring calculations incorporate the feed-drive selection methodology described in [23]. These methodologies provide the basis for relating

the calculated machining conditions to the corresponding drive requirements.

The application processes the technological input parameters, automatically validates the input data, selects the relevant analytical equations according to the machining operation and calculates the required cutting parameters. The calculated results are subsequently displayed through the graphical user interface, which allows operators to perform quick technological assessments without manual calculations.

The main input parameters are related to setting parameters such as the workpiece material, cutting tool diameter, cutting speed, feed per revolution or feed per tooth, machining length, number of cutting edges, spindle speed (if predefined) and other operation-specific technological data. When preparing the calculation and optimization procedure, it is necessary to consider kinematic factors. In particular, when selecting the feed per revolution of the rotary table, the characteristics of the worm gear [24] and input belt drive of the machine are decisive [25]. Depending on the selected machining operation, the application automatically determines the relevant output parameters.

To improve the reliability of the operation, the developed software performs automatic validation of all technological input parameters before performing the calculations. The implemented validation procedure checks the completeness of the input data, the numerical correctness of the parameter values, and the correspondence between the selected machining operation and the required technological variables. Invalid or incomplete data is reported to the user, thus preventing incorrect calculations.

Unlike conventional engineering calculators, which usually perform isolated numerical calculations, the proposed application integrates machining operations typical of a machining center into a unified analytical environment. The modular implementation allows additional machining operations to be added or analytical models to be updated without changing the overall software architecture. This facilitates future development of the software in accordance with evolving manufacturing technologies.

2.2. Computational Algorithm

The application's operating algorithm is designed to automate the determination of cutting parameters, while providing a sequential step for calculating various machining operations. It follows a sequential workflow, starting with the selection of the machining operation and ending with the visualization of the calculated technological parameters.

Table 1 summarizes the main steps of the computational algorithm implemented in the

developed system. Each step has a specific function within the overall process, ensuring the correct sequence of operations and minimizing the possibility of calculation errors.

The process begins with the selection of the machining operation. Depending on the user's selection, the application activates the corresponding calculation module for drilling, boring or milling. This step determines the set of technological parameters required for the subsequent calculations.

Table 1. Basic steps of the algorithm

Step	Process	Function
1	Machining operation selection	Activates the corresponding calculation module
2	Data input	Collects technological parameters
3	Data validation	Verifies correctness and completeness of input data
4	Parameter calculation	Computes machining parameters
5	Results visualization	Displays calculated values to the user
6	Repeat or exit	Starts a new calculation or terminates the application

In the second step, the user enters the required input data, including parameters such as workpiece material, cutting tool diameter, cutting speed, feed, machining length and other operation-specific variables.

Before any calculations are performed, the software automatically validates the input data. The validation procedure checks whether all required fields are filled in, whether the numerical values are within the allowable ranges, and whether invalid or inconsistent input data has been provided. If an error is detected, the application immediately informs the user and requests correction of the input data. After successful validation, the software automatically selects the appropriate equations according to the selected machining operation and determines the necessary machining parameters.

After all calculations are completed, the results are presented via the graphical user interface. The user can review the calculated machining parameters, change the input values if necessary, or perform a new calculation without restarting the system.

The modular implementation of the algorithm allows the inclusion of additional machining operations or computational models without changing the overall workflow. This improves the scalability and maintainability of the application and facilitates future software development.

2.3. Software Implementation

The developed mobile software is implemented using the Android Studio integrated development environment, which provides a complete platform for designing, developing, testing and deploying Android applications. Android Studio was chosen due to its broad support for Android SDK components, efficient debugging tools, graphical layout editor and compatibility with modern Android devices [26]. The application is developed using the Java programming language, which offers high portability, object-oriented programming capabilities and extensive software libraries for mobile application development [27]. The graphical user interface (GUI) was designed using XML layout files, which allows for a clear separation between the application logic and the user interface. This approach improves software maintainability and facilitates future modifications of the system. The computational procedures are implemented as independent software modules corresponding to the supported processing operations. Each module performs the necessary calculations based on the technological input data provided by the user and returns the calculated processing parameters to the graphical interface.

Although the developed application automates the calculation procedure, it is not intended to eliminate the user's involvement in the technological decision-making process. Traditional manual calculations have an important educational advantage because they require the user to work directly with the analytical relationships and understand how the calculated parameters are obtained. Automated calculation reduces repetitive work and the possibility of arithmetic errors; however, excessive reliance on automated results may reduce the user's understanding of the underlying calculation procedure. Therefore, the developed application is intended to support rather than replace the user's engineering judgement. The calculated results are based on the analytical relationships described in this work and can be interpreted and verified by the user.

3. Results and Discussion

3.1. Program Logic

The architecture of the software application is built around the "Activity" cycle and XML-based views (Layout), providing a separation between the user interface and the computational logic.

- Home screen

After starting the application, the home screen is visualized, which represents the main navigation environment of the developed program system. Through it, the user selects the type of technological operation for which the necessary calculations will

be performed. Three main functional modules have been implemented, corresponding to the mechanical operations – milling, drilling and boring.

Fig. 1 presents the graphical interface of the home screen. To facilitate the user's work, graphic buttons (*ImageButton*) have been used, through which the selection of the corresponding operation is performed without the need for additional settings.



Figure 1: The graphical interface of the home screen

Fig. 2 presents a fragment of the program code that implements the initialization of the graphical components and navigation between the individual screens of the application.

```
public class MainActivity extends AppCompatActivity {
    ImageButton boring, drilling, milling;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        boring = findViewById(R.id.boring);
        drilling = findViewById(R.id.drilling);
        milling = findViewById(R.id.milling);
        milling.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent millingIntent = new Intent(MainActivity.this, milling.class);
                startActivity(millingIntent);
            }
        });
        boring.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, boring.class);
                startActivity(intent);
            }
        });
        drilling.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent boringIntent = new Intent(MainActivity.this, drilling.class);
                startActivity(boringIntent);
            }
        });
    }
}
```

Figure 2: Initialize the main screen and control navigation

The adopted modular software architecture improves maintainability, facilitates future software extensions, and enables the integration of additional machining operations without modifying the core application structure.

- Input and processing of input data

After selecting a technological operation, the software loads the corresponding screen for entering the necessary technological parameters. The input data is entered through “EditText” components, each of which is intended for a specific parameter. Depending on the selected operation, the user fills in data characterizing the processed material, the tool used and the technological mode. To ensure correct operation, a mechanism for validating input data has been implemented. The application monitors the completion of all mandatory fields and does not allow calculations to be performed in the absence of the necessary parameters. This prevents the possibility of obtaining incorrect results.

Fig. 3 presents a fragment of the program code that implements the verification of the entered values and the control of the visualization of the interface. After successfully filling in the mandatory parameters, the remaining elements on the screen necessary for performing the calculations are automatically visualized.

```
paramEditTexts[0] = HB;
paramEditTexts[1] = Dmb_max;
paramEditTexts[2] = Dcb_max;
paramEditTexts[3] = aPb_max;
paramEditTexts[4] = mMax;
paramEditTexts[5] = Kspg;
paramEditTexts[6] = nB;
paramEditTexts[7] = vb_max;
paramEditTexts[8] = ab_max;

for (EditText editText : paramEditTexts) {
    editText.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence,
                                      int i, int i1, int i2) {
        }
        @Override
        public void onTextChanged(CharSequence charSequence,
                                   int i, int i1, int i2) {
            boolean allParamsFilled = true;
            for (EditText editText : paramEditTexts) {
                if (editText.getText().toString().isEmpty()) {
                    allParamsFilled = false;
                    break;
                }
            }
            layoutOfParams.setVisibility(allParamsFilled ?
                                         View.VISIBLE : View.GONE);
        }
        @Override
        public void afterTextChanged(Editable editable) {
        }
    });
}
```

Figure 3: A fragment of the program code that implements the verification of the entered values and the control of the visualization of the interface

Additionally, the logic has been implemented for automatic processing of part of the input data. When entering certain values, the application checks the parameters and, if necessary, updates the information displayed in the interface. In this way, the user's work is supported and the probability of making errors when entering the technological characteristics is reduced.

- Computing module

After validation of the technological input parameters, the calculation process is initiated by pressing the “Calculate” button. The software converts the input values into numerical form and sequentially executes the computational algorithm.

The computational module is implemented as an independent software component responsible for processing the validated input data and calculating the technological parameters corresponding to the selected machining operation. Depending on the selected machining operation, the software calculates spindle speed, feed rate, cutting power, feed drive power, specific cutting force, torque, and other process-related technological parameters.

Fig. 4 presents a fragment of the program code implementing the mathematical dependencies and the algorithm for calculating the technological parameters.

```
setOnClickListener(new View.OnClickListener() {
    onClick(View v) {
        double frbValue = Double.parseDouble(Frb.getText().toString());
        double frb = frbValue * Math.pow(10, -5);
        double vb_max = Double.parseDouble(VB_max.getText().toString());
        double dmb_max = Double.parseDouble(Dmb_max.getText().toString());
        double vcb = Double.parseDouble(Vcb.getText().toString());
        double apb_max = Double.parseDouble(aPb_max.getText().toString());
        double wB = (vcb * 2) / dmb_max;
        double kcfzb = Double.parseDouble(Kcfzb.getText().toString());
        double nb = Double.parseDouble(nB.getText().toString());
        double dcb_max = Double.parseDouble(Dcb_max.getText().toString());
        double hbi = Double.parseDouble(Hbi.getText().toString());
        double abmax = Double.parseDouble(ab_max.getText().toString());
        double mmax = Double.parseDouble(mMax.getText().toString());
        double mug = Double.parseDouble(Mg.getText().toString());
        double dgb = Double.parseDouble(Dgb.getText().toString());
        double kspg = Double.parseDouble(Kspg.getText().toString());
        double Wbms = wB * kspg;

        double vfbValue = (frb * wB) * Math.pow(10, -3);
        double pcb_max = (vcb * apb_max * frb * kcfzb * 2 * Math.PI *
                          * Math.pow(10, 6)) / (nb * (1 - (apb_max / dcb_max)

        double pcbw = (1.1 / 1.3) * pcb_max;
        double pspb = (95.0 / 99) * pcbw;
        double pfb = (1 / 5.0) * pcbw;
        double ffb = pfb / vfbValue;
        double wfbi = ((0.05 * 2 * Math.PI) / hbi) * 10;
        double fg = mmax * abmax * mug;
        double pfbgc = (fg + ffb) * vfbValue;
        double pfbg_max = fg * vb_max;
        double mfbi = pfbgc / wfbi;
        double mfv_max = pfbg_max / wfbi;
    }
});
```

Figure 4: A code implementing the mathematical dependencies and the algorithm for calculating the technological parameters

Once the calculations have been completed, the obtained results are automatically transferred to the graphical user interface, where they are presented in a structured format suitable for engineering analysis.

To increase the reliability of the application, processing of incorrectly entered data has been implemented. If an error is detected when converting the input values or the lack of necessary information, the application notifies the user with an appropriate message and prevents incorrect calculations.

The computational module constitutes the core functional component of the developed application, providing reliable and automated determination of machining parameters for various types of mechanical processing.

3.2. Implementation of Machining Modules

The developed application includes three specialized functional modules designed for automated calculation of the main technological parameters in the operations of boring, drilling and milling.

The boring module (Fig. 5) enables the user to enter the technological parameters required for boring operations.

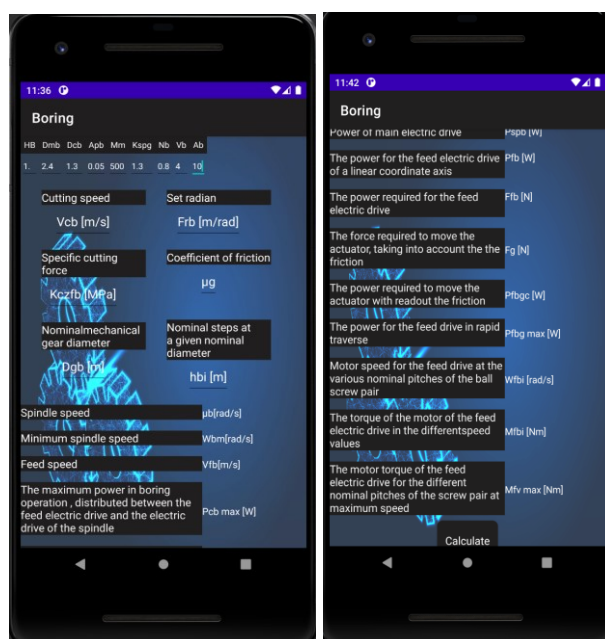


Figure 5: Boring module user interface

After the input data have been validated, the software automatically determines spindle speed, feed rate, cutting power, torque, and other machining characteristics.

The drilling module (Fig. 6) follows the same software architecture while incorporating mathematical models specific to drilling operations. As a result, the software determines the main operating modes and the necessary parameters for performing the technological operation.

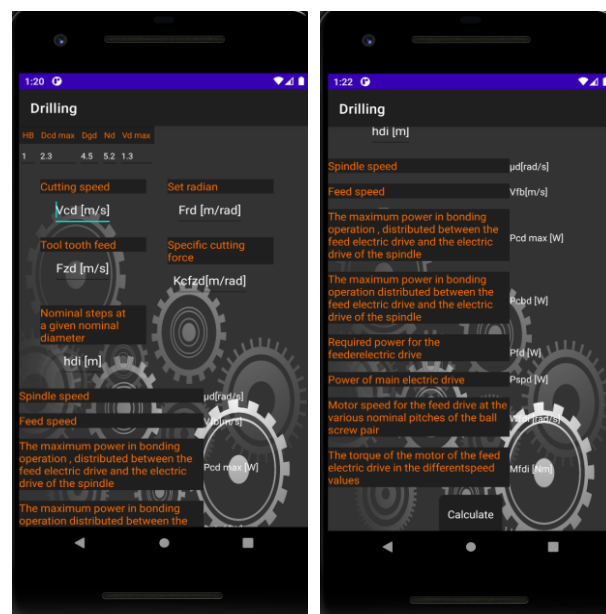


Figure 6: Drilling module user interface

The milling module (Fig. 7) is based on the common application architecture, with computational algorithms adapted to the characteristics of milling processes. After processing the entered technological data, the software calculates the necessary cutting modes and visualizes the obtained results in the user interface.

The use of a common software architecture in the three modules ensures a unified user interface, a uniform way of working and easy maintenance of the application. At the same time, specific algorithms for each technological operation guarantee the correct determination of cutting modes according to the specifics of the respective process.

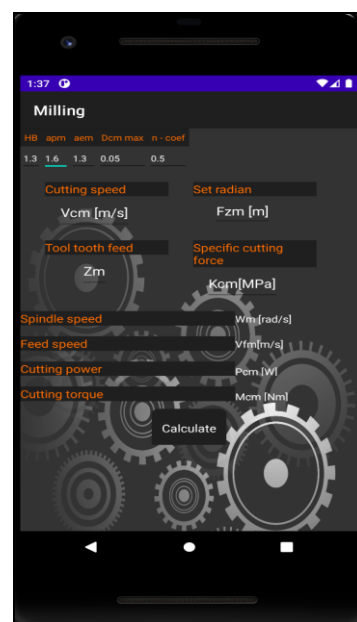


Figure 7: Milling module user interface

Overall, the developed mobile application enables automated determination of technological

parameters for drilling, boring, and milling operations. By eliminating repetitive manual calculations, the proposed solution reduces process planning time, minimizes the probability of human errors, and assists engineers in selecting appropriate machining conditions. The modular software architecture further facilitates future extension of the application through the integration of additional machining operations and computational models.

3.3. Comparative Evaluation of the Developed Application

Table 2 presents a comparative analysis between the conventional manual procedure for determining the processing parameters and the proposed Android application.

Table 2. Comparison of manual calculation and the proposed Android application

Criterion	Conventional manual calculation	Proposed Android application
Input data validation	Manual	Automatic
Formula selection	Manual	Automatic
Calculation execution	Manual	Automatic
Average calculation time	Approximately 3–5 min	Approximately 10–15 s
Risk of calculation errors	Medium	Low
Result presentation	Manual interpretation	Automatic visualization
Recalculation after parameter changes	Repeated manual calculations	Immediate recalculation
User guidance	Reference tables required	Integrated graphical interface

The comparison shows that the developed software significantly simplifies the calculation workflow by automating data processing, selecting appropriate equations, and visualizing the results. While manual calculations typically require between 3 and 5 minutes, the proposed application performs the same task within approximately 10–15 seconds. Furthermore, automatic validation of input data reduces the probability of calculation errors, thereby improving the reliability of the obtained results.

These advantages make the application a practical engineering tool for rapid process planning and support the digitalization of the selection of processing parameters.

The proposed application significantly reduces the time required for determining machining parameters, decreasing it from several minutes to only a few seconds while simultaneously minimizing the likelihood of human errors through automated validation and computational processing.

4. Conclusions

This article presents the development of an Android-based software application for automated calculation of machining parameters in drilling, boring and milling. The software is implemented using the Java language in the Android Studio programming environment and uses analytical models to determine spindle speed, feed rate, cutting power, torque and other important machining parameters. A modular software architecture is used, which provides a clear separation between the graphical user interface and the computational algorithms, thus facilitating software maintenance and future expansion.

The developed system automates the validation of input process data, the selection of appropriate computational models and the presentation of calculated results. Compared to the conventional manual calculation procedure, the proposed solution significantly reduces the time required for process planning, while minimizing the likelihood of errors through automated data validation and computational processing. The integration of drilling, boring and milling modules within a single mobile platform provides a practical engineering tool that supports the planning of machining processes in industrial and educational environments.

Future work will focus on expanding the application by including additional machining operations, expanding the database of workpiece and tool materials, and integrating optimization algorithms for automatic selection of cutting parameters. Further development may also include cloud-based data synchronization and direct communication with CNC systems, which will allow for broader integration into Industry 4.0 and smart manufacturing environments.

Acknowledgement

This research was funded by the Research and Development Sector at the Technical University of Sofia, Bulgaria under Project No. 261PD0001-08/2026.

References

- [1] Patel, H., Chotaliya, M., Bhalodia, D. & Lakum, P. (2025). Industry 4.0 and Smart Manufacturing: A Review. *International Journal of Manufacturing and Production Engineering*, 3(2), 14–17. <https://journals.stmjournals.com/ijmpe/article=2025/view=234930>
- [2] Bleistein, T., Paulus, M., Gani, K., Becker, R. & Werth, D. (2025). IIoT and Digital Twin: A Systematic Literature Review and Looking Beyond the State. *Applied Stochastic Models in Business and Industry*, 41, e2923. <https://doi.org/10.1002/asmb.2923>
- [3] Ahmmed, Md, Khan, L., Mahmood, M. & Liou, F. (2025). Digital Twins, AI, and Cybersecurity in Additive Manufacturing: A Comprehensive Review of Current Trends and Challenges. *Machines*, 13(8), 691. <https://doi.org/10.3390/machines13080691>
- [4] Liao, Z., Schoop, J., Saelzer, J., Bergmann, B., Priarone, P., Splettstößer, A., Bedekar, V., Zanger, F. & Kaynak, Y. (2024). Review of current best-practices in machinability evaluation and understanding for improving machining performance. *CIRP Journal of Manufacturing Science and Technology*, 50, 151–184. <https://doi.org/10.1016/j.cirpj.2024.02.008>
- [5] Soori, M., Jough, F., Dastres, R. & Arezoo, B. (2024). Sustainable CNC machining operations, a review. *Sustainable Operations and Computers* 5, 73–87. <https://doi.org/10.1016/j.susoc.2024.01.001>
- [6] Pawanr, S., & Gupta, K. (2024). A Review on Recent Advances in the Energy Efficiency of Machining Processes for Sustainability. *Energies*, 17(15), <https://doi.org/10.3390/en17153659>
- [7] Trinh, V.-L. (2024). A Review of the Surface Roughness Prediction Methods in Finishing Machining. *Engineering, Technology & Applied Science Research*, 14(4), 15297–15304. <https://doi.org/10.48084/etasr.7710>
- [8] Korkmaz, M., Gupta, M., Sarikaya, M., Günay, M., Boy, M., Yaşar, N., Demirsöz, R. & Pehlivan, F. (2024). Analytical Modeling Methods in Machining: A State of the Art on Application, Recent Challenges, and Future Trends. *Arabian Journal for Science and Engineering* 49, 10287–10326. <https://doi.org/10.1007/s13369-024-09163-7>
- [9] Ullrich, K., Elling, M., Gutzeit, K., Dix, M., Weigold, M., Aurich, J., Wertheim, R., Jawahir, I. & Ghadbeigi, H. (2024). AI-based optimisation of total machining performance: A review. *CIRP Journal of Manufacturing Science and Technology*, Volume 50, 40–54, ISSN 1755-5817. <https://doi.org/10.1016/j.cirpj.2024.01.012>
- [10] Song, J., Wang, B. & Hao, X. (2024). Optimization Algorithms and Their Applications and Prospects in Manufacturing Engineering. *Materials*, 17(16), 4093. <https://doi.org/10.3390/ma17164093>
- [11] Yoon, J., Park, I. & Lee, D. (2025). A Review of Intelligent Machining Process in CNC Machine Tool Systems. *International Journal of Precision Engineering and Manufacturing*, 26, 2243–2268. <https://doi.org/10.1007/s12541-025-01343-1>
- [12] Kaur, R., Kumar, R. & Aggarwal, H. (2025). Systematic Review of Artificial Intelligence, Machine Learning, and Deep Learning in Machining Operations: Advancements, Challenges, and Future Directions. *Archives of Computational Methods in Engineering*, 32, 4983–5036. <https://doi.org/10.1007/s11831-025-10290-z>
- [13] Lu, J., Tu, S., Li, Y., Zhang, L. & Liao, X. (2025). Process planning of parameter intelligent adjustment for batch machining based on historical data segmented modeling. *Engineering Applications of Artificial Intelligence*, 145, 110180. <https://doi.org/10.1016/j.engappai.2025.110180>
- [14] Patil, C., Trikal, S. & Shahade, A. K. (2026). A taxonomy of artificial intelligence techniques for machining parameter optimization. *Multiscale and Multidisciplinary Modeling, Experiments and Design*, 9, 155. <https://doi.org/10.1007/s41939-026-01239-1>
- [15] Belmouadden, M., Dadouchi, C. & Pellerin, R. (2025). Artificial intelligence applied in adaptive manufacturing process monitoring: A state-of-the-art in the era of automation. *Procedia Computer Science*, 256, 47–54. <https://doi.org/10.1016/j.procs.2025.02.094>
- [16] Gao, T., Wang, L., Song, W., Cheng, Y., Zuo, Y., Xiang, F., Zhang, H. & Tao, F. (2025). Ten industrial software towards smart manufacturing. *Journal of Manufacturing Systems*, 79, 255–285. <https://doi.org/10.1016/j.jmsy.2025.01.012>
- [17] Zhang, K., Xiao, W., Fan, X. & Zhao, G. (2025). CAM as a Service with dynamic toolpath generation ability for process optimization in STEP-NC compliant CNC machining. *Journal of Manufacturing Systems*, 80, 294–308. <https://doi.org/10.1016/j.jmsy.2025.03.004>
- [18] Hrehova, S., Husár, J., Lazorík, P. & Trojanowski, P. (2024). The concept of an intelligent mobile application with elements of augmented reality and PLM software for value stream mapping. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-024-06006-4>
- [19] Maretto, L., Faccio, M. & Battini, D. (2025). Industry 4.0 technology implementation in manufacturing: a selection method and real case applications. *International Journal of Production Research*, 63, 3142–3174. <https://doi.org/10.1080/00207543.2024.2430439>
- [20] Elshourbagy, S. (2025). Enhancing error detection and improvement in operations and manufacturing through app designers.

- International Journal of Environmental Science and Technology*, 22, 12083–12090. <https://doi.org/10.1007/s13762-025-06392-7>
- [21] Zhilevski, M. & Mikhov, M. (2018). Optimization of the Drive System Choice for a Class of Drilling Machines. *European Journal of Electrical and Computer Engineering*, 2(6), 12–16. <https://doi.org/10.24018/ejece.2018.2.6.42>
- [22] Mikhov, M. & Zhilevski, M. (2014). Methodology for Selection of Spindle Drives for Milling Machines. *International Journal of Engineering and Computer Science*, 3(5), 5948–5953. ISSN 2319-7242
- [23] Zhilevski, M. & Mikhov, M. (2019). Optimal selection of feed electric drives for boring machines. *IOP Conference Series: Materials Science and Engineering*, 618(1), 012003. <https://doi.org/10.1088/1757-899X/618/1/012003>
- [24] Krol, O. & Sokolov, V. (2021). Selection of worm gearing optimal structure for machine rotary table. *Diagnostyka*, 22(1), 3–10. <https://doi.org/10.29354/diag/129949>
- [25] Shevchenko, S., Mukhovaty, A. & Krol, O. (2022). Research of the modified tooth-belt drive for the machining center. In: Radionov, A. A., Gasiyarov, V. R. (Eds.), Proceedings of the 7th International Conference on Industrial Engineering (ICIE 2021). ICIE 2021. Lecture Notes in Mechanical Engineering (pp. 32–41). Springer, Cham. https://doi.org/10.1007/978-3-030-85233-7_4
- [26] Android Studio, 2026, [online] Available: <https://developer.android.com/studio>
- [27] Java Documentation, 2026, [online] Available: <https://docs.oracle.com/en/java/>