*Journal of Machine Engineering, 2019, Vol. 19, No. 4, 70–81*
*ISSN 1895-7595 (Print)   ISSN 2391-8071 (Online)*

Received: *12 November 2019* / Accepted: *03 December 2019* / Published online: *20 December 2019*

*machine tool error, compensation,*
*control system*

Pawel MEDYK[1*]
Jerzy JEDRZEJEWSKI[1]
Marcin KASPRZAK[1]

# CNC MACHINE TOOL ERROR COMPENSATION SYSTEM IMPLEMENTATION STRATEGIES AND THEIR CONSTRAINTS

This paper deals with the constraints imposed on error compensation systems by their implementation strategies. Constraints stemming from the structure of the control system with an NCK architecture of the ADCBI type are presented. Practical realizations of compensation and potential solutions offered by contemporary commercial CNC controllers are discussed.

## 1. INTRODUCTION

Because of the strong competition on the CNC machine tool market, machine producers need to continuously improve their machines. Therefore solutions improving selected properties of machine tools, but without increasing their production costs, are sought. A property which significantly improves machine tool competitiveness is its precision, i.e. its ability to reproduce the user preset trajectory of tool motion with the possibly smallest error. A machine tool error is usually understood as a difference between the actual tooltip center point position and the point preset by the control program. This definition is a commonly used simplification, very much justified because it directs one's attention to the phenomena which can cause machining precision to deteriorate, and to measures which should be taken to improve this precision. The problem of a measure of machine tool precision is described in more detail in [1].

Extensive research into methods of improving machine tool precision has been conducted for years because of the general trend towards increasing the productivity of machine tools and the demand for high-precision products. Generally, the methods of improving machine tool precision can be divided into two groups: hardware methods and software methods. The hardware methods include, i.a., design modifications, including

[1] Wroclaw University of Science and Technology, Department of Machine Tools and Mechanical Technologies, Wroclaw, Poland
[*] E-mail: pawel.medyk@pwr.edu.pl
https://doi.org/ 10.5604/01.3001.0013.6231

modifications of feed drive system [2], using higher quality assembly components and equipping machine tools with additional position sensors for the realization of a hybrid loop (Fig. 1). The latter uses motor shaft position measurements to achieve a fast system response, and direct ball screw nut position measurements to ensure higher positioning precision [3]. Considering that the use of hardware solutions as a rule entails considerably higher machine tool production costs and that improving precision by means of such methods is possible only to a limited extent, software methods, such as error compensation systems and advanced feed drive control algorithms [2], have been developed for many years to improve precision. The fast technological progress, especially the growing computing power of computers (which directly contributes to the development of advanced CNC control systems), is conducive to the development of software methods. Thanks to the higher computing power of processors, increasingly more advanced error compensation systems based on highly complex error models can be implemented. This creates possibilities for the use of new error modelling tools, e.g. convolutional neural networks (characterized by better knowledge generalization than the commonly used methods [4]), and for creating systems performing optimization in time closer to real time. Nevertheless, it is worth noticing that implementing error compensation system on poorly designed machine with noticeable hysteresis may have negative effects, such as deterioration of surface quality [5].

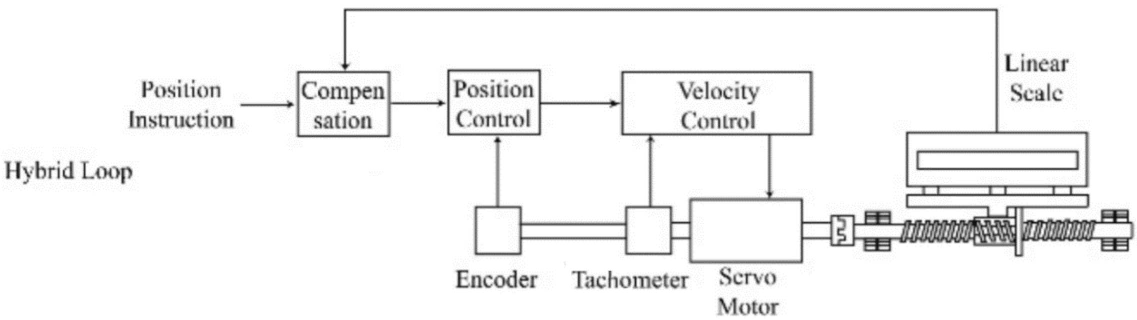


Fig. 1. Hybrid loop for controlling drive system in CNC machine tools [3]

By and large, contemporary control systems seem to be incapable of the compensation of errors caused by changes in fast-variable quantities, such as the cutting force. Attempts were made to compensate tool deformation and displacement caused by cutting forces by predicting the force value in the given point using G-code and its later modification taking into account the displacements caused by the cutting force [6]. A similar approach using a G-code modification including corrections for errors caused by cutting forces was presented in [7], where the cutting force was predicted on the basis of three feed and depth-of-cut components. However, such methods do not realize the real-time compensation of errors generated by the cutting force, but use error prediction regardless of the hardness of the stock material, without taking into account random factors causing changes in the cutting force in real time. Because of the partial randomness of the errors generated by the cutting force and since flexible solutions, which could be used in small lot production and to improve many different machines, are sought, research on compensation performed in real time or in time possibly closest to real time is needed.

The present paper discusses the implementation of an error compensation system and the associated difficulties stemming from the structure of the control system and the way it operates. The NCK (Numerical Control Kernel) unit is briefly described and then the problem of compensation implementation in the control system is discussed.

## 2. CONTROL SYSTEM STRUCTURE

Contemporary CNC control systems commonly use an NC system kernel of the ADCBI type (acceleration/deceleration control before interpolation). A block diagram of such a kernel is shown in Fig. 2. Thanks to the ADCBI motion planning architecture a deviation of the actual tool path from the preset one can be avoided. In the case of ADCAI (acceleration/ deceleration control after interpolation) the deviation is due to the fact that the Acc/Dec control is realized separately for each of the axes after the displacement which at the preset feed rate is to occur in one interpolator clock pulse is determined. For NCK of the ADCBI type the theoretical tool path generation error connected with changes in the feed rate amounts to zero. But such an architecture is more difficult to implement than ADCAI and requires greater computing power and more memory. Currently, NCK units have sufficient memory and computing power reserves for ADCBI architecture implementation. The benefits resulting from the use of this architecture include not only the elimination of the error connected with changes in the velocity of the feed axes, but also the possibility of implementing additional functions, such as the Look Ahead algorithm.

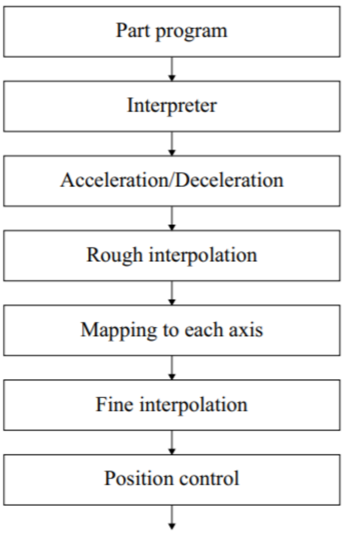


Fig. 2. Block diagram of NC kernel of ADCBI type [3]

For better understanding of the difficulties in implementing error compensation systems in contemporary CNC machine tools the particular ADCBI NCK modules and the way in which data flow between them are briefly described below. This is also presented in Fig. 3. This subject is dealt with in detail in [3].

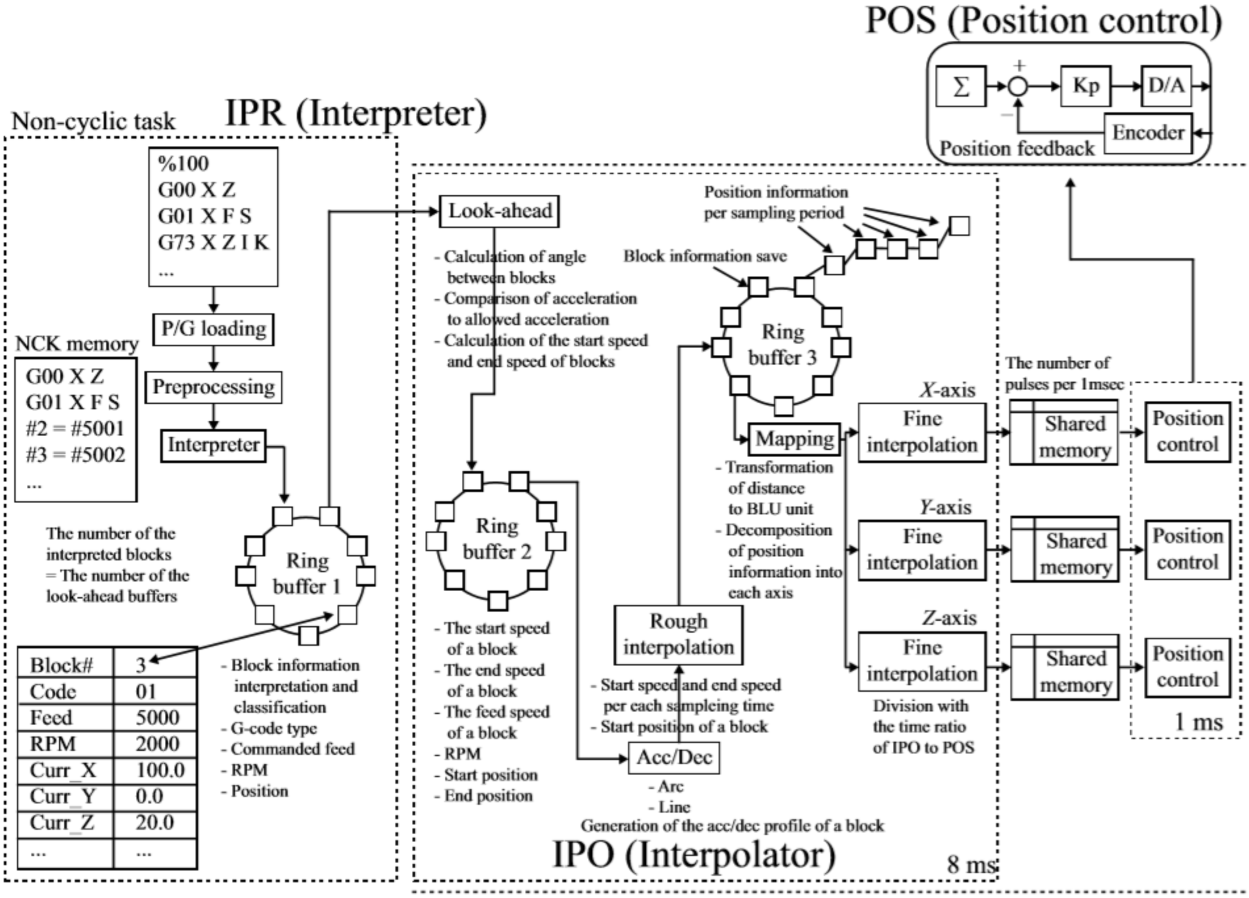


Fig. 3. Structure of NCK of ADCBI type with flow of data between modules [3]

After the NC program is executed it is interpreted by the control system. For this purpose the interpreter module loads all the necessary system and user variables and then cyclically reads the program block by block. For each of the blocks the interpreter performs appropriate computations and transformations to determine the initial and target positions of the tooltip central point in the basic coordinate system for the given block. Then it performs a tool radius and length correction and checks if the calculated velocity and position values do not exceed the allowable boundary values. Then the interpreter module writes the results in the buffer used by the Look Ahead function (if the latter is implemented in the control system and is turned on).

The Look Ahead algorithm significantly increases machining productivity when the program consists of many short blocks (blocks describing small displacements). It is mainly applicable to the machining of free-form surfaces since it is for the latter that the tool path is usually described by a sequence of very short blocks. The aim of the Look Ahead algorithm is to avoid a situation when the rate of feed is severely limited by the length of the blocks and to prevent tool stoppage after the execution of each short block. The function executing the algorithm draws the necessary information from many (from a few to a few hundred or even a few thousand) blocks which follow the currently processed block and on this basis determines the possibly highest feed rate for the given block.

For the processed block the number of iterations to be performed by the rough interpolator in order to move to the given block's preset end position and a table of values of the (linear or angular) velocity with which the tool should move for the particular interpolator iterations are determined in the acceleration control module. Determining speeds for the successive interpolation iterations the acceleration controller attempts to bring the tool acceleration as close as possible to the nearest maximum value dictated by the drives and/or machine constraints.

The rough interpolator determines the target position for each interpolation iteration, using information about the speed and the remaining distance or fragment of the arc (considering that the acceleration, steady speed and deceleration phases must be realized in a time being the multiple of the interpolator clock pulse time) and taking into account the compensational displacements. The rough interpolator's output is a table containing information about the position to which the central tooltip point is to move in the given interpolation clock pulse. This output is written in the buffer from which it is later retrieved by the mapping-to-axis module.

After retrieving the data from the buffer the mapping-to-axis module determines the displacement value for each of the axes in the given interpolation clock pulse and then sends the data to the fine interpolator. Generally, the mapping-to-axis module can be regarded as a part of the rough interpolator.

In general, the fine interpolator module prepares data for the position controller. The preparation consists in dividing the distance which the given axis is to cover during an interpolation clock pulse into a table of displacements of this axis in the next position controller clock pulses. This is so because the position controller cycle time is many times shorter than the interpolator cycle time.

## 3. ERROR COMPENSATION SYSTEM CONSTRAINTS DUE TO ADOPTED IMPLEMENTATION STRATEGY

One of the ways in which an error compensation implementation strategy can be divided is the division into on-line and off-line compensation. On-line compensation is performed during the execution of the NC program within the particular NCK modules, whereas off-line compensation is done prior to machining. In general, off-line strategies consist in reconstructing the control program. Since the usability of off-line compensation methods is not limited by the control system, only limitations in the implementation of on-line compensation methods are discussed below.

### 3.1. ERROR COMPENSATION IN INTERPRETER

Since the interpreter writes data on the initial and end position of the whole block, without determining the intermediate positions, the use of compensation in the interpretation process results in only the determination of a new initial or end position of the block, with no change in the position of the intermediate points determined in the rough interpolation module. Moreover, data are interpreted much ahead of time, stored in the buffer and then processed by the next modules. The number of buffer elements varies depending on the length of the blocks and can reach from a few to a few hundred or even a few thousand. For this reason the time which passes from the instant a compensation correction is entered into the interpreter module to the instant when the motion is executed is relatively long and impossible to determine. Therefore entering compensation values into the interpreter module makes sense only in the case of static or quasi-static errors, i.e. such whose value does not change or changes very little in time and is independent of the position within a single block. If error values change significantly between the block’s initial point and end point it is possible to prepare earlier the G-code in such a way that the block will consist of many short (linear of circular) segments. This, however, can significantly reduce machining efficiency, especially in the case of NCK without the Look-Ahead module. The shorter the blocks, the lower the machining efficiency.

### 3.2. ERROR COMPENSATION IN ROUGH INTERPOLATOR

Error compensation can be performed at the rough interpolation stage in order to increase the precision of the compensation of position-dependent errors and to reduce the lag between entering a correction and its execution. Owing to this solution, during the generation of a position for the next interpolation clock pulses the displacement values determined for this position by the compensation system can be taken into account. The distances between the positions determined for the successive interpolation iterations are very small, whereby position-dependent errors can be compensated with the possibly maximum precision (the same with which the interpolator determines the successive nodes). Moreover, the rough interpolator module processes blocks in advance and stores the processed data in the buffer, whereby no so severe constraints connected with acting in real time as the ones occurring within the fine interpolator are imposed on it. Unfortunately, the consequence is that the error compensation performed as part of rough interpolation is asynchronous compensation.

The time passing from the writing of an instruction in the form of a preset position to the instant when the motion described in this way is executed is indeterminate. This is due to the fact that the number of instructions being already in the buffer from which the mapping module and then the fine interpolator read data depends on very many factors and is not constant. Consequently, also the execution time of all the instructions which have not been executed yet, but are present in the buffer is variable and furthermore, much longer than the interpolator cycle time.

Because the lag between the instant when compensation is introduced into the path and the instant of its execution is unknown, it is not possible to introduce the compensation of time-dependent errors if the changes do not occur slow enough. One can assume that for time-dependent errors it makes sense to introduce compensation during rough interpolation if the time in which a significant change in the error value occurs is greater than the product of the interpolator cycle time and the buffer size (the maximum number of buffer elements containing information about the position for one sampling cycle) between the rough interpolator and the mapping module.

It is worth noting here that reducing the buffer size cannot be the solution to this problem. The buffer enables the fluid motion of the tool if computations for the next block are not completed during the interpolator clock pulse. If the buffer is too small, there is a risk that before the interpolation of the next block is performed all the instructions present in the buffer will be executed and the motion of the tool will be halted because of the absence of execution instructions. The higher the rate of feed, the faster the buffer is emptied (at the same interpolator clock pulse) and so the larger the buffer should be.

Each machine tool has certain kinematic properties which determine the maximum allowable axis velocity values and the maximum acceleration and jerk values. This imposes certain constraints on drive motion generation, which are taken into account in the interpreter and acceleration controller modules.

As a result of the introduction of error compensation into rough interpolation the determined tool displacement for the given interpolation iteration can differ from the one which follows from the speed determined by the acceleration controller and is not subject to verification by the modules preceding the interpolation. Therefore when implementing error compensation systems one should bear in mind that there is a certain limitation on the value of compensation corrections and one should check whether after the corrections the generated motion does not result in the exceedance of the allowable velocity and acceleration values.

It is worth noting that both the compensation performed during the rough interpolator clock pulse, by adding compensation corrections when determining the preset points for the next interpolation iterations, and the compensation realized through a modification of the data within the rough interpolator buffer are implementations of asynchronous on-line compensation. The two implementations are characterized by almost the same lag between the introduction of a compensation correction and the execution of the motion. Therefore, simplifying one can assume that the two methods of implementation are methods of compensation in rough interpolation.

The greatest difficulty in realizing compensation in the rough interpolator is that it is necessary to modify the operation of the rough interpolator or to have access to the interpolator buffer, which is not possible in the case of commercial CNC controllers. Thus compensation in rough interpolation is possible only in open CNC systems [8] or using ready-made functions of commercial CNC controllers.

### 3.3. ERROR COMPENSATION IN FINE INTERPOLATOR

Owing to compensation performed in the fine interpolator it would be possible to compensate position- and time-dependent errors since compensation corrections would be executed in the same interpolator clock pulse. The fine interpolator has no information about the preset position, but only the preset displacement (in basic units of length) for the preset axis. Therefore the compensation of position-dependent errors within this module would require access to measurements of the tool position in the controllable axes. Such a solution, shown in Fig. 4, was presented in [8].

At the same time it seems that the strongest barrier to the implementation of error compensation in the fine interpolator is that the latter must operate in real time. Besides reading measurements off sensors, such a system would have to process the error model during one interpolator clock pulse to determine the compensation value and then read data off the shared memory (off which data are read by the position controller) in order to overwrite them after corrections are added. Because of the constraints due to the necessity of performing all the operations in one interpolator pulse, no solution of error compensation in fine interpolation has been presented so far.

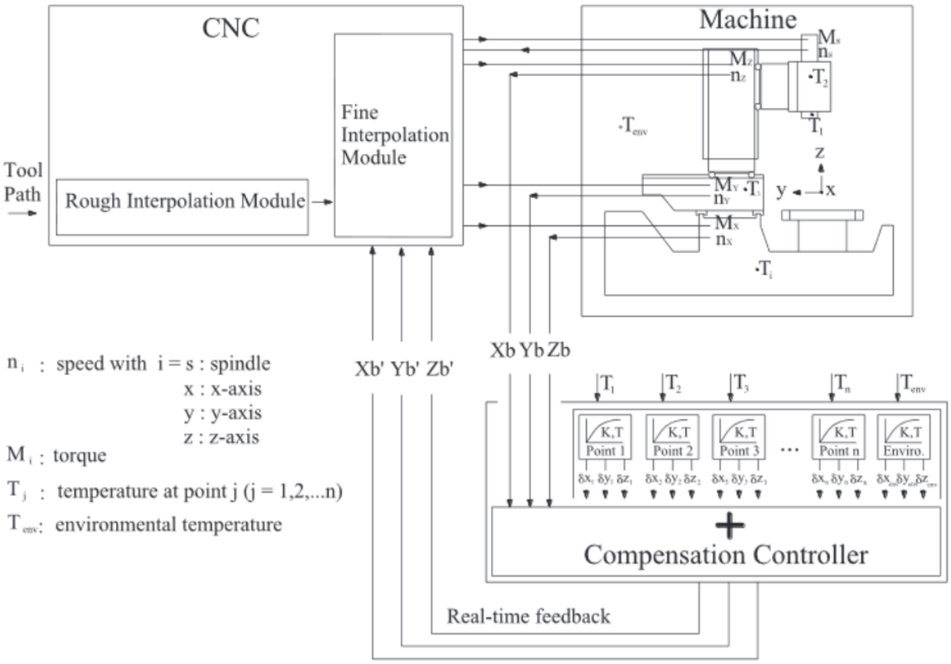


Fig. 4. Thermal error compensation in fine interpolation [8]

## 4. ERROR COMPENSATION STRATEGIES – SURVEY OF SOLUTIONS

### 4.1. OFF-LINE COMPENSATION

The most popular direction of research on error compensation is still off-line compensation consisting in modifying G-code. This method can be used if the error is repeatable or if it can be predicted. The most serious drawback of this method is that compensation is effected through a change of the position of the program end blocks. Because of this, the path has to be divided into short segments and arcs in order to achieve good results. This can adversely affect machining efficiency. At the same time it is not possible to introduce compensation in intermediate points between the block's initial and end position. Nojodeh et al. carried out the compensation of the geometric errors of a 3-axis milling machine, using the method of dividing the nominal path into short segments and arcs [9], which resulted in a 50% and 12% machining error reduction in two tests. Zhue et al. [10] compensated the geometric errors of a 5-axis machine tool by modifying G-code consisting of short blocks, which resulted in a 40-50% improvement in machining precision. Ratchew et al. [11] proposed a strategy for the multilevel compensation of errors caused by the cutting force during milling, through error prediction and G-code modification, showing that it is possible to reduce the total machining error by as much as 70% using a single-level compensation algorithm. The use of full multilevel compensation can yield even better results.

### 4.2. ON-LINE COMPENSATION

#### 4.2.1. COMPENSATION IN INTERPRETER MODULE

Considering that compensation in the interpreter module generally comes down to block end position modification, for time-invariable errors the compensation works the same as off-line compensation and alike requires the program consisting of short blocks. The difference between the two strategies is that compensation in the interpreter module makes it possible to take into account the variation of errors over time.

The above approach was used by Vinod et al. [12] to realize the on-line compensation of the positioning error of a turning lathe by introducing a compensation correction as a zero point shift or a tool off-set (possible selection of a command). The cycle time for determining a new correction value was set to 1 second. This solution did not use any external devices, but only a program processing the error model on the same processing unit. This was made possible by the open architecture of the CNC controller. A positioning error reduction of about 75% was achieved.

The same method of introducing compensation was used in [13] for thermal error compensation and the error was reduced from 50 μm to 3 μm.

#### 4.2.2. COMPENSATION IN ROUGH INTERPOLATOR

Compensation performed within rough interpolation can be effected by means of, e.g., the functions available in commercial CNC controllers, such as: temperature compensation, sag compensation and the volumetric compensation system.

The error model used by the temperature compensation function [16] is stored in PLC. At specified intervals (of a few minutes) PLC processes the temperature profile and performs the linear approximation of the error function for the measured temperature. The function approximating the error function is as follows:

$$K_x(T) = K_0(T) + K_P(T)(P_X - P_0)$$

where:

$K_x(P_x, T) -$ temperature compensation value of axis at position $P_x$ and temperature $T$,
$P_X -$ actual position of axis,
$P_0 -$ reference position of axis,
$K_P(T) -$ coefficient for the position-dependent temperature compensation,
$K_0(T) -$ position-independent temperature compensation value of axis.

Then linear function coefficients $K_0$ and $K_p$, determined through approximation, are sent to NCK. As the target positions for the successive interpolation iterations are determined in the rough interpolation process, the compensation correction value is computed. In order to avoid problems connected with a too high value of the compensation correction, the latter is distributed over a few successive interpolator clock pulses. In this way a new position preset for the given interpolation iteration is obtained. It is worth noting that if the temperature compensation function is used, the asynchronous on-line compensation of position-dependent errors is possible, provided they are successfully approximated in segments, with the linear function. Also position-independent errors in the form of off-set can be compensated by setting coefficient $K_p(T)$ to zero.

The volumetric compensation system (VCS) [16] requires for its operation that all 21 geometric error components of a cartesian machine tool are determined for the whole workspace. The data are written, in the form of tables, to a file from which they are read in by the CNC controller at the program start. It is possible to create many files containing data on the volumetric error for different temperatures and/or a different weight of the tool or the workpiece.

The latest versions of VCS enable interpolation between compensation files (volumetric error value determination for intermediate temperature or weight values) and compensation which takes the direction of tool motion into account. After appropriate data are read in, the volumetric compensation system computes compensation values which are taken into account in the rough interpolation process [14]. However, error model of VCS cannot be modified when the program is running. Hence this method can be used only to compensate static errors. Compensation based on VCS can increase precision by as much as 75–80%.

The sag compensation function [15] operates on the basis of the preset compensation table which contains pairs of the preset datum axis position and the compensational axis compensation value. The function enables the use of various tables, depending on the direction of motion of the given axis, and the selection of the same axis as simultaneously the datum axis and the compensational axis. Similarly as in the case of VCS, the compensation tables cannot be modified during machining, which means that only static errors can be modified by means of this function.

Other methods of introducing compensation corrections into the interpolation process are interpolatory corrections [15], such as: the external shift of the zero point, the DRF (Differential Resolver Function) offset, and the on-line tool correction, available in contemporary CNC systems.

A strategy of implementing the compensation of position-dependent geometric errors and thermal errors, based on the use of the DRF offset, was implemented by Cui et al. [16]. An error model was processed on an external device which after computing the compensation value sent it to the CNC controller, imitating the hand-wheel signal. The experimental results showed a geometric error reduction in axis $x$, $y$, $w$, of respectively 65%, 70% and 46%.

Shen et al. [8] presented a method of compensating static/quasi-static errors through the modification of the data in the rough interpolator buffer. The compensation values were determined on the basis of a compensation table which included: the preset position in the given axis, the post-compensation position for forward motion and the post-compensation position for backward motion. An attempt was made to verify the proposed method through simulation and the method was found to correctly map the values contained in the compensation tables in the modified rough interpolator data. However, this strategy can be implemented only in open CNC systems.

As part of the above study also the compensation of the position-independent error was carried out, using the PLC-CNC interface for the asynchronous determination of the off-set by means of an inbuilt temperature compensation function (the method was discussed above). The error reduction amounted to over 70%.

## 5. CONCLUSION

High precision is one of the main criteria which a modern machine tool must satisfy. Striving to maximally exploit the machining precision potential of machine tools wide research into software methods of improving machine tool precision is being conducted. Contemporary control systems are capable of handling various error compensation system implementation strategies. In this paper, compensation system constraints resulting from the adopted compensation implementation strategy and whose source is the structure of the control system were discussed.

The most popular error compensation strategy is the off-line method consisting in G-code reconstruction. However, using this method one cannot take into account the effects of changes in temperature and force-induced dynamically variable deformations on machining precision. Only the end positions of program blocks can be corrected in this way. The effect of changeable machining conditions can be taken into account through error compensation in the interpreter module, but with a considerable lag. For this reason such a strategy can be used for temperature compensation, but not for the compensation of deformations induced by forces. A compensation correction can be introduced only by modifying the block’s end position, which limits the achievable compensation precision.

Different strategies for implementing compensation in the rough interpolation module were presented in this paper. Thanks to compensation performed within this module the occurring changes can be responded to with a shorter delay than in the case of compensation taking place in the interpreter and corrections can be introduced in the intermediate points determined in the interpolation process.

So far no method of performing compensation in real time has been developed. Such compensation could be realized in the fine interpolator module, but the contemporary CNC controllers make this impossible. Further research in this area is needed.

### ACKNOWLEDGEMENT

*The authors express his thanks to NCBiR for supporting the research on intelligent machine tools, Project number POIR.04.01.04-00-0067/17.*

### REFERENCES

[1] JÓZWIK J., 2018, *Experimental methods of error identification in CNC machine tool operation*, Lublin University of Technology.
[2] ALTINTAS Y., VERL A., BRECHER C., URIARTE L., & PRITSCHOW G., 2011, *Machine tool feed drives*, CIRP Annals, 60/2, 779–796.
[3] SUH S.H., 2008, *Theory and design of CNC systems,* Springer Science & Business Media.

[4] ZHANG X., YANG L., LOU P., et al., 2019, *Thermal Error Modelling for Heavy Duty CNC Machine Tool Based on Convolution Neural Network*, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference ITNEC, IEEE, 665–669.
[5] SCHWENKE H., KNAPP W., HAITJEMA H., et al., 2008, *Geometric error measurement and compensation of machines – an update*, CIRP Annals, 57/2, 660–675.
[6] SOORI M., AREZOO B., HABIBI M., 2014, *Virtual machining considering dimensional, geometrical and tool deflection errors in three-axis CNC milling machines*, Journal of Manufacturing Systems, 33/4, 498–507.
[7] RAKSIRI C., PARNICHKUN M., 2004, *Geometric and force errors compensation in a 3-axis CNC milling machine*, International Journal of Machine Tools and Manufacture, 44/12–13, 1283–1291.
[8] SHEN H., FU J., HE Y., YAO X., 2012, *On-line asynchronous compensation methods for static/quasi-static error implemented on CNC machine tools*, International Journal of Machine Tools and Manufacture, 60, 14–26.
[9] NOJEDEH M.V., HABIBI M., AREZOO B., 2011, *Tool path accuracy enhancement through geometrical error compensation*, International Journal of Machine Tools and Manufacture, 51/6, 471–482.
[10] ZHU S., DING G., QIN S., et al., 2012, *Integrated geometric error modelling, identification and compensation of CNC machine tools*, International journal of machine tools and manufacture, 52/1, 24–29.
[11] RATCHEV S., LIU S., HUANG W., BECKER A.A., 2006, *An advanced FEA based force induced error compensation strategy in milling*, International Journal of Machine Tools and Manufacture, 46/5, 542–551.
[12] VINOD P., REDDY T.N. SAJIN S., et al., 2014, *Real-time positioning error compensation for a turning machine using neural network*, Procedia Materials Science, 5, 2293–2300.
[13] REDDY T.N., SHANMUGARAJ V., PRAKASH V., et al., 2014, *Real-time thermal error compensation module for intelligent Ultra Precision Turning Machine (iUPTM)*, Procedia materials science, 6, 1981–1988.
[14] TUREK P., JEDRZEJEWSKI J., MODRZYCKI W., 2010, *Methods of machine tool error compensation*, Journal of Machine Engineering, 10/4, 5–25.
[15] Siemens Company publications, i.a.: Extended Functions, SINUMERIK 840D sl/840DE sl. Version 4.91, Extended Functions, SINUMERIK 840D sl/828D, Version 4.7 SP2.
[16] CUI G., LU Y., GAO D., YAO Y., 2012, *A novel error compensation implementing strategy and realizing on Siemens 840D CNC systems*, The International Journal of Advanced Manufacturing Technology, 61/5–8, 595–608.