48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015

Online prediction of cutting tool life in turning via cognitive decision making

Sara Karam*, Piera Centobelli, Doriana M. D'Addona, Roberto Teti

Fraunhofer Joint Laboratory of Excellence on Advanced Production Technology (Fh-J_LEAPT Naples)
Dept. of Chemical, Materials and Industrial Production Engineering, University of Naples Federico II, Piazzale Tecchio 80, Naples 80125, Italy

* Corresponding author. Tel.: +39-3420647890; fax: +039-0817682362. *E-mail address:* sara.karam@unina.it

Abstract

Multiple sensor monitoring of machining was investigated for online cutting tool life assessment through cognitive decision making based on signal processing for feature extraction and pattern recognition. Sensor signals obtained from sensor monitoring of turning operations were processed and analysed. The outcome was a set of extracted signal features correlated with the consumed tool life percentage. The aim of the work is to build an online cognitive system, based on artificial neural networks, able to predict the consumed tool life during turning operations. A preliminary experimental campaign was carried out for the construction of the sensorial knowledge database; the neural network type, architecture and training algorithm. After setting up the sensorial knowledge database and the neural network paradigm, the cognitive decision making system is ready to be implemented for online cutting tool life prediction during actual turning operations by exploiting the capacity of neural networks to constantly learn and improve through interaction with the sensorial data acquisition and processing system.

Keywords: Tool wear in turning; sensor monitoring of machining processes; neural network; online decision making systems;

1. Introduction

Monitoring and control of machining processes represent a major concern in industry and call for intense research and development in the field. One main research focus is given by the implementation of multiple sensor monitoring systems based on sensor fusion technology, allowing for the integration of information from sensors of different nature in order to improve the quality and robustness in the characterization of the process, cutting tool or operating machine [1]. This approach gives rise to novel requirements that need to be addressed, including the critical issue of handling ambiguous or noisy sensor signals.

Sensor monitoring systems can be effectively grounded on learning schemes, such as those based on cognitive paradigms, making use of sensorial data input to obtain output information about the machining operation, particularly when the process is so complex as to defy clear mathematical modelling. In recent years, research has emphasized the capability of intelligent sensing techniques applied to machining, which can be summarised as the detection of signals provided by multiple sensors during cutting, the conditioning and processing of these signals for cutting conditions control, and the utilisation of relevant signal features for cognitive decision making on the quality of the process. The capabilities of machining process monitoring through cognitive approaches comprise the continuous feedback of information on operative conditions to the machine tool numerical control. Examples of process conditions that can be monitored are the in-process evaluation of tool wear, the quick identification of tool fracture, the classification of chip form, and the verification of the nature and properties of the work material [2-4].

The final scope of the present research is tool life monitoring using a multiple sensor system provided with a cutting force sensor, a vibration sensor and an acoustic emission sensor mounted on a CNC turning machine. Sensor fusion technology investigations are carried out through advanced signal processing and feature extraction. The output relevant features are to be related to the tool condition via cognitive paradigms based on neural network (NN) pattern recognition aimed at tool life prediction [5,6]. This paper focuses on the building of the decision making part of the intelligent monitoring system, which requires experimental testing to generate the training set necessary for NN learning. These activities have been carried out within the EC FP7

Project "Real-Time In Situ Monitoring of Tool Wear in Precision Engineering Applications – REALISM" [7].

2. System setup

2.1 Multiple sensor monitoring system

Monitoring systems supplied with multiple sensors are highly appropriate for the detection of the diverse physical outputs from the machining process. These physical outputs can be used for the identification of correlations between sensorial data and cutting tool conditions. The multiple sensor system utilized in this work is composed of a cutting force sensor, a vibration sensor, and an acoustic emission sensor mounted on a CNC lathe (Fig. 1). During turning tests, signals are detected, digitized and stored on PC for post-processing.

2.2 Tool wear measurement

Cutting tool wear land was stepwise measured during turning tests through a small portable microscope positioned on the lathe tool holder to avoid cutting tool dismounting..

3. Sensorial knowledge database generation

To implement the NN decision making system for tool wear assessment, the execution of an experimental turning campaign to generate the sensorial knowledge database is required to make up the training set for NN learning.

3.1 Experimental turning campaign

Longitudinal turning tests on AISI 316 stainless steel using carbide inserts were carried out with multiple sensor signal detection for tool wear monitoring and stepwise measuring of tool flank wear. The criterion to consider the tool life fully consumed was the achievement of the maximum allowable wear land, upon which the turning test was concluded [8].

3.2 Sensor signal pre-processing

The detected sensor signals need to be pre-processed to be made suitable for signal analysis. Signal pre-processing consists of signal cleaning by which the head and tail of each signal are cut off to eliminate the transient periods, and signal segmentation by which homogeneous signal portions, corresponding to regime cutting conditions, are extracted.

3.3 Sensor signal processing for feature extraction

In the literature, several methods are proposed for sensor signal processing in order to extract features useful for NN pattern recognition and decision making: e.g. statistical feature extraction, Principle Component Analysis, and Wavelet feature extraction [2-4,6,8]. In the present work, the Wavelet feature extraction method was applied [9].

Fig. 1. Location of the three sensors on the tool holder.

3.4 Neural network based pattern recognition

NN pattern recognition based on feature patterns constructed through sensor fusion technology was utilized for decision making on tool life assessment.

NN are inspired by the biological nervous system [10]: they are composed of simple nodes that are interconnected and operate in parallel. The typology of the connections between the nodes determines the function of the network. NN are trained to perform a particular function by adjusting the weights of the connections between the nodes. One main function that NN are trained for is pattern recognition, i.e. distinguishing a pattern among the input features and relating this input pattern to the desired output. The NN architecture is typically made of input, hidden and output nodes, respectively contained in the input, hidden and output layers [10].

In this work, different three-layer feed-forward back-propagation NN configurations were implemented by varying the number of nodes contained in the hidden layer. For each NN architecture, the training algorithm utilized was the leave-k-out (LKO) method [11] whereby one homogeneous group of k patterns is removed from the full training set and held back in turn for testing, while the rest of the patterns are used for training. In the present case, k is equal to the number of feature patterns constructed for one fully consumed tool life and varies for each considered tool life.

4. Neural network decision making system

To implement the online NN decision making system for tool life assessment, a sensorial knowledge database must be generated and a suitable type of NN and an adequate training and testing algorithm need to be set up.

As regards the sensorial knowledge database, sensor signals detected during monitoring of turning operations were processed by the Warsaw University of Technology (WUT) within the EC FP7 REALISM project [7], providing 6-element pattern vectors, made of 6 extracted signal features (SF), related to a corresponding output given by the percentage of consumed tool life, estimated on the basis of the measured tool wear level, for seven identical cutting tool inserts. Table 1 reports the 6-SF pattern vectors with the

matching percentage of consumed tool life for three of the seven identical cutting tool inserts, the life of which was fully consumed by the same turning operation.

Thus, each set of 6-SF pattern vectors, identified in Table 1 by a block highlighted with the same grey tone, corresponds to the complete tool life of one cutting insert. Sensor signal acquisition and processing for SF extraction is interrupted when the tool life of the cutting insert is fully consumed, which is indicated in the table by 100%.

To carry out the prediction of consumed tool life based on cognitive processing of the 6-SF pattern vectors, diverse supervised NN paradigms were implemented using the Neural Networks Toolbox in MatLab and the Machine Learning Toolkit in LabVIEW.

In all cases, the NN was trained by a number of known instances identified by the couples constituted by one input feature pattern vector and one output quality value. The elements of the input feature pattern vector are the 6 SF extracted from the sensor signals, and the correspondent percentage consumed tool life is the output quality value. The trained NN is then tested with instances unknown to the network with the scope to assess the NN performance in consumed tool life prediction.

The implemented NN architectures are characterized by 3 layers: the input layer with 6 input nodes corresponding to the 6 SF, the hidden layer with a number of hidden nodes related to the number of input nodes, and the output layer with 1 output node yielding the predicted percentage of consumed tool life. The max number of epochs or iterations that the NN can perform during the training phase to achieve convergence is initially set. If convergence occurs before reaching the max number of epochs, NN training is halted.

The NN training and testing procedure was carried out as follows: the input 6-SF pattern vectors and the related percentage consumed tool life values for tool life # 1 were used for NN training, while the input 6-SF pattern vectors for tool life # 2 were used for NN testing. Next, the input 6-SF pattern vectors and the related percentage consumed tool life values for tool lives # 1 and # 2 were used for NN training, while the input 6-SF pattern vectors for tool life # 3 were used for NN testing. Finally, the input 6-SF pattern vectors and the related percentage consumed tool life values for tool lives # 1, # 2 and # 3 were used for NN training and, one at a time, tool lives # 4, # 5, # 6 and # 7 were used for NN testing.

4.1 Neural network implementation in MatLab

First approach using cascade-forward backpropagation NN

The cascade-forward backpropagation NN paradigm was applied in the first approach with 5 different NN architectures, built by varying the number of hidden nodes in the hidden layer, for NN learning: 6-3-1, 6-6-1, 6-9-1, 6-12-1, 6-15-1.

For each NN architecture, the MatLab function 'newcf' was used to create a cascade-forward backpropagation network [12]. The algorithm utilized for NN training was the symmetric sigmoid transfer function, converting the NN layer net input into its net output, known as 'tansig' in MatLab [13].

Fig. 2 reports the predicted consumed tool life values, obtained from the 6-3-1 MatLab implemented cascade-forward backpropagation NN architecture, $\Delta T_{calculated}$ (%), vs. the actual consumed tool life values, ΔT_{real} (%).

Table 2 reports the performance of the 5 MatLab implemented cascade forward backpropagation NN architectures for consumed tool life prediction.

Second approach using feed-forward backpropagation NN

The feed-forward backpropagation NN paradigm was used in the second approach. Feed-forward NN consist of a series of layers: the first layer has a connection from the NN input with weights from the input. Each subsequent layer has a connection from the previous layer and a weight coming from the previous layer. The final layer provides the NN output.

Utilizing the MatLab Neural Network Toolbox, feed-forward NN were configured with input layers employing the DOTPROD weight function, the NETSUM net input function, and the transfer function "purelin" [14]. Each layer's weights and biases are initialized with the INITNW layer initialization function. Adaption is done with the incremental training algorithm that updates weights with the learning function "trainlm", i.e. a NN training function that updates weight and bias values according to the Levenberg-Marquardt algorithm [15]. Performance is measured according to the specified performance function "mse", which measures the NN performance according to the mean of squared errors.

The architecture utilised for the feed-forward 3-layer NN was as follows: the input layer had 6 input nodes equal to the number of SF in the input pattern vectors, the hidden layer had 4 nodes, and the output layer had only one node. The number of hidden nodes was chosen according to a "feed learning" procedure: hidden units are added one at a time until an acceptable training speed is achieved.

Fig. 3 reports the predicted consumed tool life values, obtained from the 6-4-1 MatLab implemented feed-forward backpropagation NN architecture, $\Delta T_{calculated}$ (%), vs. the actual consumed tool life values, ΔT_{real} (%).

Table 2 reports the performance of the MatLab implemented feed-forward backpropagation NN for consumed tool life prediction.

Table 1. Extracted sensor signal features coupled with the corresponding tool wear value for three of the seven cutting tool inserts.

Tool life #	SF1	SF2	SF3	SF4	SF5	SF6	Consumed tool life (%)
1	-0.077	230	4.600	48.43	1231	-0.407	13
1	0.018	254	4.610	47.23	1312	0.039	25
1	-0.045	438	4.637	48.40	1330	0.736	38
1	0.028	584	4.724	47.68	1355	2.279	50
1	-0.025	569	4.815	48.45	1404	-0.165	63
1	0.241	753	4.769	49.15	1451	-0.223	75
1	0.047	1545	5.204	49.87	1425	-0.749	88
1	0.033	1187	4.975	50.67	1395	-0.794	100
2	-0.060	199	4.653	47.47	1306	0.236	20
2	0.016	259	4.655	47.65	1304	1.441	40
2	0.054	474	4.892	48.16	1342	1.016	60
2	0.092	550	4.773	47.63	1378	-0.439	80
2	0.051	782	4.817	49.15	1423	-0.834	100
3	-0.013	149	4.599	47.12	1318	0.386	17
3	0.009	190	4.652	47.73	1332	-0.016	33
3	0.039	313	4.689	47.63	1361	0.845	50
3	0.027	513	4.721	48.53	1391	0.178	67
3	0.050	732	4.820	48.59	1431	-1.629	83
3	0.084	1036	4.988	49.20	1434	1.165	100

4.2 Neural network implementation in LabVIEW

In the third approach, the LabVIEW platform was used to implement the same 5 cascade-forward backpropagation NN architectures as in the first MatLab approach (section 4.1): 6-3-1, 6-6-1, 6-9-1, 6-12-1, 6-15-1

The LabVIEW add-on library Machine Learning [16] was used for training and testing the NN architectures to compare the LabVIEW and MatLab platform effectiveness. Two VIs were used: back-propagation learn (BP Learn: training VI) and back-propagation evaluate (BP Evaluate: testing VI).

For each VI, the input data need to be set up. The input and output datasets are saved in text files: train input, train output, test input, test output. The output of the BP Learn VI is connected as input to the BP Evaluate VI. The output of the BP evaluate is a 1D vector containing the predicted outputs. The number of hidden neurons is specified in the BP Learn VI. The training algorithm is the Levenberg-Marquardt algorithm. By default, the number of epochs is set to 100.

Fig. 4 reports the predicted consumed tool life values, obtained from the LabVIEW implemented cascade-forward backpropagation NN architecture 6-3-1, $\Delta T_{calculated}$ (%), vs. the actual consumed tool life values, ΔT_{real} (%).

Table 2 reports the performance of the 5 LabVIEW implemented cascade forward backpropagation NN architectures for consumed tool life prediction.

5. Online cognitive decision making system implementation

After the creation of the sensorial knowledge database, the online decision making system for tool life assessment can be set up. As mentioned earlier, the NN training set is composed of sensorial features from three complete tool lives, i.e. three sets of 6-SF pattern vectors coupled with the corresponding percentages of consumed tool life.

For the proper implementation of the cognitive online system, a dedicated PC is required for the installation of the decision making software code and the connection to the CNC of the turning machine. Fig. 5 illustrates the information flow diagram of the cognitive decision making system throughout the machining process. The CNC controller sends the process code to the decision making system running on the PC (Fig. 5a). The system uses this code to identify the machining process and, accordingly, the appropriate learned NN, the NN trained with the relevant couples of sensorial feature pattern vectors and output quality values for the selected process.

Next, the turning operation starts, the identified NN identified through is ready for sensorial data processing, and the sensor signal data acquisition is initiated. A sensorial data acquisition time window of 1 second was considered sufficient to extract the signal features and send them as input to the learned NN in order to obtain as output the predicted consumed tool life value (Fig. 5b).

Once the predicted consumed tool life value is obtained, it is compared with 1 that represents the end of tool life and the need to change the cutting insert. If the predicted consumed

tool life value is less than 1, another 1 second of machining is carried out; otherwise, the turning operation is interrupted for tool change. For each successive 1 second of machining, the extracted signal features composed of the sensorial pattern vectors are coupled with the predicted consumed tool life value obtained by interrogating the learned NN.

These newly generated data, made up of the additional couples of feature pattern vectors and predicted consumed tool life values, are stored sequentially in a buffer (Fig. 5c).
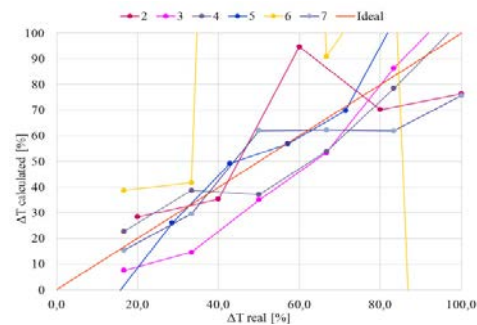

Fig. 2. Consumed tool life prediction results for the MatLab implemented 6-3-1 cascade-forward backpropagation NN architecture.
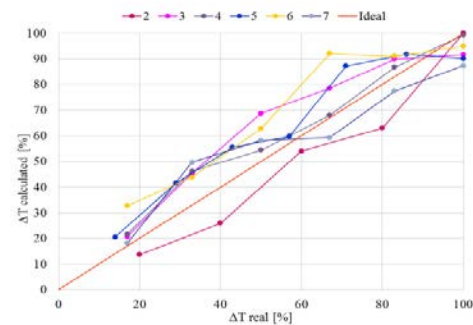

Fig. 3. Consumed tool life prediction results for the MatLab implemented 6-3-1 feed-forward backpropagation NN architecture.
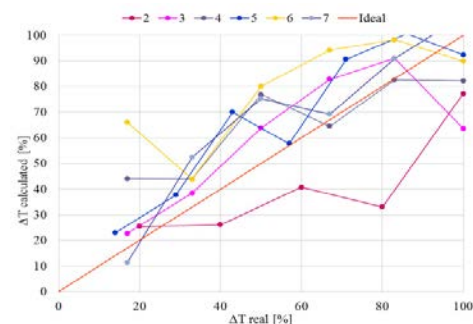

Fig. 4. Consumed tool life prediction results for the LabVIEW implemented 6-3-1 cascade-forward backpropagation NN architecture.

Table 2. Performance of the diverse NN approaches for consumed tool life prediction in terms of prediction error given by (ΔT_{real} - $\Delta T_{calculated}$) / ΔT_{real} (%), where ΔT_{real} is the actual consumed tool life and $\Delta T_{calculated}$ is the predicted consumed tool life.

|	Consumed tool life (%)	MatLab approach using cascade-forward backpropagation NN					MatLab approach using cascade-forward backpropagation NN	LabVIEW approach using cascade-forward backpropagation NN				
		3 nodes	6 nodes	9 nodes	12 nodes	15 nodes	3 nodes	3 nodes	6 nodes	9 nodes	12 nodes	15 nodes
TL 2	20	25.41	13.01	14.36	16.35	34.37	31.89	39.47	45.76	54.81	27.35	71.84
	40	26.09	16.12	15.29	16.36	48.32	35.27	1.96	52.60	8.82	0.22	20.79
	60	40.73	48.72	46.14	41.76	115.01	9.95	55.57	46.08	72.68	54.80	91.68
	80	32.99	66.56	63.69	59.32	85.25	21.22	13.51	7.42	4.00	73.76	6.56
	100	77.22	85.36	84.27	83.59	92.70	0.00	24.76	25.00	16.49	17.56	7.30
	Average error (%)	40.49	45.95	44.75	43.48	75.13	19.67	27.05	35.37	31.36	34.74	39.64
TL 3	0.17	22.63	17.32	19.22	20.56	8.89	21.76	55.94	51.53	58.15	111.15	47.68
	0.33	38.43	27.11	29.32	29.96	17.18	37.41	56.15	5.678	58.35	163.17	47.93
	0.50	63.84	46.76	46.41	45.78	40.92	37.44	31.09	15.42	34.53	17.43	18.16
	0.67	82.78	66.34	65.98	65.09	62.42	17.23	21.54	16.13	25.47	17.59	6.83
	0.83	90.94	94.30	96.56	99.90	101.12	8.33	2.59	8.99	2.54	31.99	21.83
	1.00	63.62	63.97	68.08	69.74	132.43	8.42	11.52	4.75	5.95	23.87	32.44
	Average error (%)	60.37	52.63	54.26	55.17	60.49	18.96	29.81	17.09	30.83	110.87	29.15
TL 4	0.17	44.02	27.83	31.33	45.07	23.29	27.49	31.96	63.69	84.29	74.01	145.04
	0.33	43.90	47.35	44.62	41.84	39.59	40.12	15.55	43.49	35.21	110.55	22.69
	0.50	76.89	78.93	79.29	72.62	37.97	8.78	26.86	57.85	58.59	64.56	45.79
	0.67	64.53	67.89	68.87	71.55	55.10	1.44	20.80	1.33	2.78	23.18	9.24
	0.83	82.48	82.91	82.62	82.07	80.28	4.51	6.85	0.10	0.45	74.55	0.28
	1.00	82.19	88.36	86.59	82.07	107.53	0.65	3.56	11.64	13.41	69.43	18.95
	Average error (%)	65.67	65.55	65.55	65.87	57.29	13.63	17.60	29.68	32.46	69.38	40.33
TL 5	0.14	22.89	1.52	1.34	29.46	2.82	46.27	121.69	15.14	120.02	128.16	120.12
	0.29	37.69	35.97	38.04	41.80	23.77	43.11	11.62	25.07	18.43	131.49	18.02
	0.43	70.01	77.29	72.99	63.84	44.93	29.21	12.63	102.81	3.96	33.90	4.48
	0.57	57.73	61.88	64.08	60.72	52.00	4.82	1.65	46.92	9.22	7.41	8.77
	0.71	90.58	95.43	94.97	92.73	63.76	22.89	3.20	29.30	10.65	64.22	10.20
	0.86	100.68	96.05	94.69	97.01	101.83	6.70	27.65	9.09	17.82	17.42	18.41
	1.00	92.29	96.17	88.48	65.80	95.07	9.87	2.49	19.86	5.40	40.99	4.93
	Average error (%)	67.41	66.33	64.94	64.48	54.88	20.45	25.85	35.46	26.50	89.08	26.42
TL 6	0.17	66.09	63.83	63.86	62.22	82.62	92.16	124.39	175.52	175.66	113.84	163.99
	0.33	43.87	46.06	46.92	45.41	84.47	32.99	24.47	39.57	42.17	37.49	33.48
	0.50	80.01	88.91	78.12	69.15	93.58	25.65	22.78	77.81	56.24	24.39	32.69
	0.67	94.09	98.13	96.26	93.83	120.71	37.33	33.66	46.46	43.67	50.01	36.80
	0.83	98.23	99.43	99.34	102.19	80.66	9.64	52.52	19.79	19.69	22.62	23.18
	1.00	89.87	94.72	87.98	65.94	65.52	5.11	181.09	5.28	12.02	34.31	30.44
	Average error (%)	78.69	81.84	78.74	73.12	87.92	32.11	139.82	77.40	74.91	63.78	70.10
TL 7	0.17	11.32	8.73	10.25	32.65	13.46	6.53	10.36	48.63	39.70	165.02	91.65
	0.33	52.38	56.92	51.54	49.39	25.76	50.49	11.68	72.49	56.17	109.49	51.60
	0.50	75.28	77.35	73.94	70.91	54.01	16.21	22.23	54.71	47.89	174.46	32.45
	0.67	69.12	71.78	71.87	68.48	54.29	11.55	8.31	7.13	7.27	18.89	5.19
	0.83	90.89	87.32	90.02	88.85	53.98	6.69	26.39	5.21	8.46	122.08	1.16
	1.00	107.68	97.59	102.91	89.57	65.79	12.67	25.55	2.41	2.91	154.37	23.89
	Average error (%)	67.78	66.61	66.76	66.64	44.55	7.05	17.42	31.76	27.07	124.05	34.32
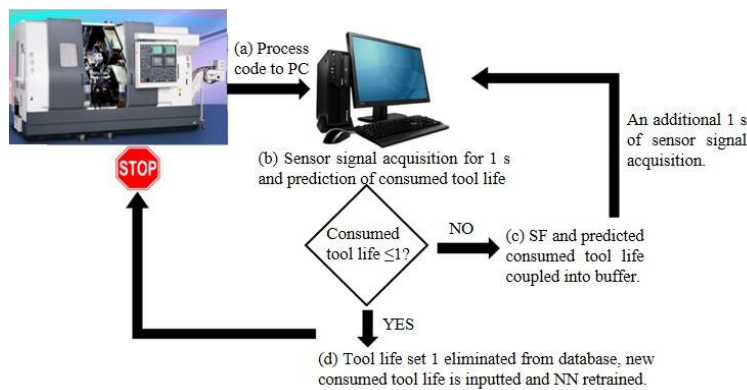
Fig. 5. Information flow diagram of the cognitive decision making system throughout the machining process.

This procedure is repeated till the predicted consumed tool life value reaches 1, attaining the maximum allowable tool wear and calling for the cutting insert change. Once this event is verified, an output STOP command is sent to the CNC controller in order to interrupt the turning operation, and an alert sign is displayed on the CNC controller screen warning the operator to change the cutting insert (Fig. 5d).

In the meantime, the original training set, initially employed for NN learning, is modified in the following manner. The first tool life sensorial dataset, composed of the coupled sensorial feature pattern vectors and consumed tool life values, is removed. The new tool life sensorial dataset is inserted and NN is retrained with the updated training set. After changing the cutting insert, the retrained NN is employed to predict the consumed tool life value of the new insert.

6. Conclusions

Sensor signal feature pattern vectors, obtained from experimental sensor monitoring of turning tests, were coupled with consumed tool life percentages and used to develop a cognitive decision making system for tool life assessment.

Neural network (NN) based pattern recognition for tool life consumption prediction was implemented on the basis of known instances given by the couples of input sensorial feature pattern vectors and correspondent output consumed tool life percentages.

Two NN types, a cascade-forward NN and a feed-forward NN, were investigated after implementation with the software packages: MatLab and LabVIEW. The preliminary results showed that the feed-forward NN implemented with MatLab provided the best performance in comparison with the cascade-forward NN implemented with MatLab and the cascade-forward NN implemented with LabVIEW.

On the grounds of the obtained results, an online cognitive decision making system was designed and set up for real-time tool life monitoring during turning. The system detects sensor signals from a given turning operation, processes them to extract relevant signal features, constructs input feature pattern vectors and forwards them to the appropriate learned NN for tool life prediction in the selected turning process.

If the predicted consumed tool life is lower than the usable tool life, a further machining term is performed. For each successive machining term, the extracted sensorial feature pattern vector is coupled with the predicted consumed tool life and stored sequentially in a buffer. When the predicted consumed tool life equals the usable tool life, the turning process is interrupted and the operator is warned for cutting tool change. Concurrently, the previous tool life dataset is removed, the new tool life dataset is inserted and the NN is retrained with the updated training set. The newly learned NN is then used to predict the consumed life of the new cutting tool. In this way, continuous learning in machine intelligence is realised by exploiting the NN capacity to constantly learn and improve through interaction with the sensorial data acquisition and processing system.

The most important aspect of the intelligence of the developed cognitive system is its capacity to constantly learn and improve through interaction with the environment in which it acts, i.e. continuous learning from the inputs to the cognitive system along time creating a continuous feedback in the machine learning algorithm.

References

[1] Teti, R., Jemielniak, K., O'Donnell, G., Dornfeld, D., 2010. Advanced monitoring of machining operations, CIRP Annals - Manufacturing Technology 59/2, p. 717-739.
[2] S. Karam, R. Teti, 2013. Wavelet transform feature extraction for chip form recognition during C steel turning, Procedia CIRP 12, p. 97-102.
[3] Segreto, T., Karam, S., Simeone, A., Teti, R., 2013. Residual stress assessment in Inconel 718 machining through wavelet sensor signal analysis and sensor fusion pattern recognition, Procedia CIRP 9, p. 103-8.
[4] Segreto, S. Karam, R. Teti, J. Ramsing, 2015. Feature Extraction and Pattern Recognition in Acoustic Emission Monitoring of Robot Assisted Polishing, Procedia CIRP 28, p. 22-27.
[5] H.V. Ravindra, Y.G. Srinivasa, R. Krishnamurthy, 1993. Tool wear monitoring in turning using a pattern recognition technique, Journal of Materials Processing Technology, 37, p. 731-740.
[6] Segreto, T., Simeone, A., Teti, R., 2012. Sensor fusion for tool state classification in nickel superalloy high performance cutting, Procedia CIRP 1, p. 593-598.
[7] EC FP7, Call for SME-2012, Real time in situ monitoring of tool wear in precision engineering applications - REALISM (2014-15).
[8] Segreto, T., Simeone, A., Teti, R., 2013. Multiple sensor monitoring in nickel alloy turning for tool wear assessment via sensor fusion, Procedia CIRP 12, p. 85-90.
[9] Krzysztof, J., Tomasz, U., Kossakowski, J., Bombiński, S., 2011. Tool condition monitoring based on numerous signal features. Int J Adv Manuf Technol, 59/1, p. 73-81.
[10] Jordan, M.I., Bishop, C.M., 2004. "Neural Networks". In Allen B. Tucker. Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, FL: Chapman & Hall/CRC Press LLC.
[11] Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14th International Joint Conference on Artificial Intelligence, p. 1137–1143.
[12] Prasad Mohapatra D, Patnaik S, Intelligent Computing, Networking, and Informatics, Springer India 2014; p.195.
[13] tansig, Neural Network Toolbox in Matlab, http://dali.feld.cvut.cz/matlab/nnet/tansig.html, retrieved on 18/06/2015.
[14] purelin, Neural Network Toolbox in Matlab, http://dali.feld.cvut.cz/matlab/nnet/purelin.html, retrieved on 18/06/2015.
[15] Lourakis, M.I.A., 2005. Brief Description of the Levenberg- Marquardt Algorithm Implemened by Levmar, Foundation of Research and Technology; 4, p. 1.
[16] LabVIEW Machine Learning Toolkit User Manual. https://decibel.ni.com/content/docs/DOC-19328, retrieved on 18/06/2015.