

An Intelligent Machine Monitoring System for Energy Prediction Using a Gaussian Process Regression

Raunak Bhinge, Nishant Biswas, and David Dornfeld
Mechanical Engineering
University of California, Berkeley
Berkeley, CA, USA

Jinkyoo Park and Kincho H. Law
Civil and Environmental Engineering
Stanford University
Stanford, CA, USA

Moneer Helu and Sudarsan Rachuri
Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD, USA

Abstract—Recent advances in machine automation and sensing technology offer new opportunities for continuous condition monitoring of an operating machine. This paper describes an intelligent machine monitoring framework that integrates and utilizes data collection, management, and analytics to derive an adaptive predictive model for the energy usage of a milling machine. This model is designed using a Gaussian Process (GP) regression algorithm, which is a flexible regression method that also provides an uncertainty estimate. To improve computational efficiency, we propose a Collective Gaussian Process (CGP) in which the overall energy prediction is made by constructing local GP models weighted by probability distribution functions obtained using the Gaussian Mixture Model (GMM) technique. Finally, we demonstrate the ability of the proposed monitoring framework to construct an energy prediction model to predict the energy used to machine a part.

Keywords—Milling tool; Energy prediction; Gaussian Process regression; Data-driven manufacturing.

I. INTRODUCTION

Advances in sensor technologies and automated data acquisition systems provide new opportunities for real-time collection of valuable measurements on an operating machine. In the manufacturing sector, sensors have been instrumented to monitor machine operations and to assess the conditions of the machine [1]. To facilitate archiving, accessing, and retrieving machine operation data, MTConnect, an XML-based interoperability standard, has been developed [2]. MTConnect architecture consists of a set of devices (e.g., a machine tool), an agent (i.e., a function object that collects data from the device and delivers it to the applications), a client application (e.g., a user application that consumes the device's data), and a Lightweight Directory Access Protocol (LDAP) server (i.e., a server that translates device names for the agent). Using the operation data collected from a machine and stored in a standardized format, researchers have studied the influence of

machine operational parameters on the energy usage of the machine [3, 4]. Taking advantage of the automated data acquisition from sensor measurements, this paper discusses an intelligent machine monitoring framework that integrates data collection, management, and analytics to derive predictive models relating machine operating parameters to energy consumption.

As shown in Fig. 1, the intelligent machine monitoring system consists of two basic modules: (1) a data management and extraction agent, and (2) a data-driven machine learning and knowledge extraction agent. The data management and extraction agent retrieves the raw sensor data from the operating machine and systematically converts and organizes the data into semantically meaningful input features and response output. Using data analytics and machine learning techniques, the knowledge extraction agent analyzes the processed data and builds a prediction model that relates the input features and response output for the target machine. The prediction model is continuously updated with new measurement data. The prediction model can then be used to evaluate optimal process parameters for machine operations, such as minimizing the energy used to machine a designed part.

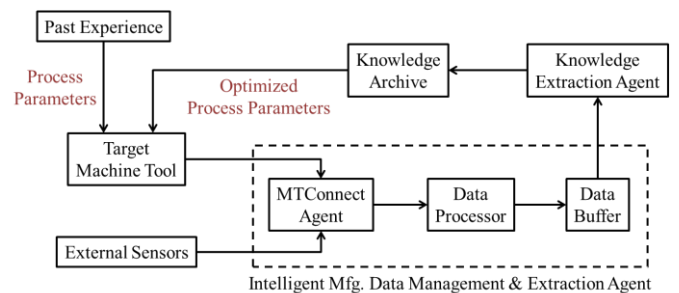


Fig 1. Intelligent machine monitoring system.

The data management and extraction module consists of the MTCConnect agent and a data post-processor. The MTCConnect agent collects time-synchronized (i.e., time-stamped) raw data, such as process control parameters and power, from the target machine. The post processor then proceeds to perform two basic tasks. Using simple calculations, it first converts the time sequenced raw data to derived values for a block of numerical control (NC) code that corresponds to a single cut or machining operation. To obtain useful feature values suitable for further analyses, the entire cutting process is simulated using the NC code and the derived data to generate material processing information, such as depth of cut, actual length of tool path, cutting strategy, and volume of material removal for each block of the NC code. The control parameters and the material processing parameters are synchronized and outputted in a format that is suitable for further data analysis.

The knowledge extraction agent uses the processed monitoring data to construct a data-driven prediction model for the target machine. Machine learning approaches have been applied to construct models for tool wear prediction [5, 6] and surface roughness prediction [7]. A machine learning module integrated in a big data infrastructure has been used to predict the energy of a machine for a cutting operation by Shin et al. [8]. In this study, we use the Gaussian Process (GP) [9] to build a nonlinear regression model that predicts the energy usage of a milling machine. As a non-parametric model, GP is particularly amendable to model complex relationships between the input and output data without the constraints for a pre-defined set of basis functions. As a probabilistic Bayesian approach, the regression model built using GP can quantify the uncertainties in the predicted values. Furthermore, GP is easy to implement because once the kernel function is specified by a user, all the parameters are automatically optimized based on the data.

Even with its advantages, GP has not been widely used in real-time monitoring applications because of the inherent computational cost and storage requirements for problems involving large data sets (e.g., training data). A number of approaches have been proposed to overcome the computational limitations. One approach is to represent the covariance matrix of GP using a sparse matrix with a limited number of base variables or basis vectors [10, 11, 12]. Another approach is to decompose the global GP regression model into a set of local GP regression models each built using a subset of training data to reduce the computational cost [13, 14, 15]. The prediction is then determined as a weighted sum of the predicted values by the local GPs. Both sparse GP and local GP models have been applied to various monitoring and tracking applications using real-time sensor measurement data [14, 16, 17]. In this study of a machine monitoring application, we employ the GP approach to build a data-driven model to characterize energy usage of a machine. Furthermore, for computational efficiency, we experiment with the local GP approach. For each local GP regression model, the probability density function for the input data (used for the local GP) is constructed based on the Gaussian Mixture Model (GMM). The collective GP prediction model that is constructed using the local GP models and the probability density functions (based on GMM) is then used to predict energy consumption. Using an automated milling machine as our prototype target, we demonstrate how the

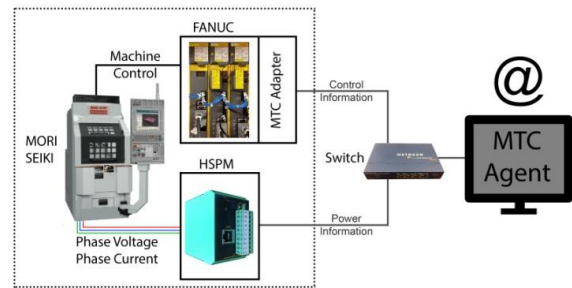


Fig. 2. Schematic representation of the hardware and software system utilized for online monitoring of the machine tool [18]. “MTC Agent” refers to the MTCConnect agent.

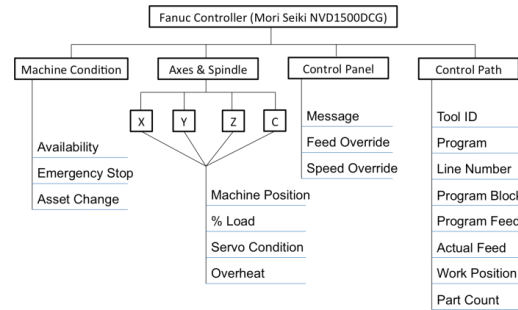


Fig. 3. Data provided by Fanuc controller on Mori Seiki [19]

proposed monitoring framework can be used to construct the energy prediction model, predict the energy usage for machining a part, and update the prediction model with real-time monitoring data.

This paper is organized as follows: First, the data acquisition system and the methods for post processing the raw sensor data are presented. We then discuss GP models and the local GP approach with GMM. The application of the GP approach developed for monitoring energy usage of a computer numerically controlled (CNC) milling machine (Mori Seiki NVD 1500DCG) is then described. The paper is concluded with a brief summary and discussion.

II. DATA ACQUISITION, MANAGEMENT, AND FEATURE EXTRACTION

This section describes the data acquisition, management, and feature extraction system developed for a machine tool using the MTCConnect standard and a data post-processor via cutting simulation. Instrumented with sensors for collecting real-time operational data, a machine tool can provide a source of valuable data sets that can be used for several purposes, such as condition-based machine monitoring, predictive modeling, and diagnostic and prognostic analyses of the machine’s condition [19]. For a machine monitoring system, it is important to ensure that the right data are measured and collected from the machine tool efficiently (with minimum number of sensors) and effectively (using the right sensors) [20]. As mentioned previously, the machine tool platform employed in this study is one developed by Helu et al. [19] on a Mori Seiki NVD1500DCG. Fig. 2 shows the schematic diagram depicting the hardware and software used to monitor the machine tool.

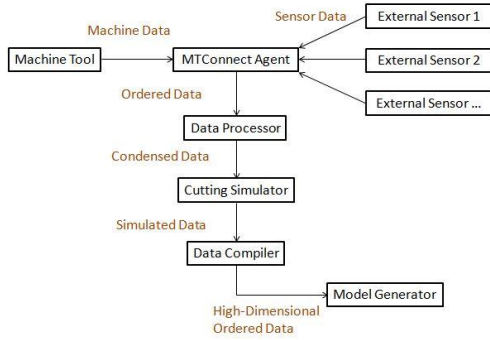


Fig. 4 Post-Processing of Machine Data Collected by the MTConnect.

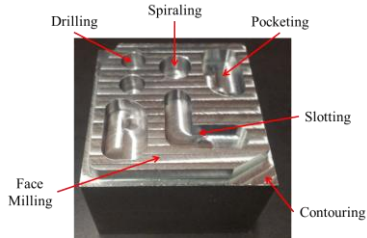


Fig. 5. Test part design for experimentation.

The two main sources of data on the machine tool platform are the Fanuc controller and the System Insights High Speed Power Meter (HSPM). Fig. 3 shows the variety of control and operation data that the Fanuc controller can provide via an Ethernet connection. The HSPM records power consumption data from the input power line to the machine tool at a frequency of 100 Hz. The HSPM and the Fanuc Controller collect a rich dataset that can be used to monitor the condition of the machine and to conduct analyses that provide a better understanding of its characteristics, such as energy consumption. The MTConnect standard is used to contextualize the data collected from the machine tool [2]. With the MTConnect agent, the synchronous monitoring data (with a common timestamp) is made available online.

To support data analysis, a post-processor converts the collected raw sensor data into “derived” data with features that can be used after further processing to develop an energy prediction model for the machine tool in this study. Fig. 4 summarizes the basic steps for the post-processing of the data collected from the MTConnect agent before it is used by the Model Generator (with any appropriate machine learning module). In the following, we first describe the experiments designed to assess the different operations by the machine tool. We then discuss the cutting simulation techniques that are used to convert the collected data into desirable features used in the development of the energy prediction model.

A. Experimental Design

Fig. 5 shows a test part that is specifically designed to generate machine operation data for the development of an energy prediction model. The parts are designed to be made on the Mori Seiki NVD1500DCG milling machine at the UC Berkeley Mechanical Engineering Machine Shop. Specific experimental details of the workpiece, machine tool, and cutting tool are shown in TABLE I. The test part requires six

TABLE II. Details of the experiments

Work piece Material	Cold Finish Mild Steel 1018
Work piece Dimensions	63.5mm x 63.5mm square cut to a length of 56mm
Machine Make	Mori Seiki NVD 1500
Machine Type	Micro NC Milling Machine
Tool Material	Solid Carbide
Tool Diameter	3/8" (9.525 mm)

TABLE II. Levels chosen for experimental operations

Level	Spindle Speed (RPM)	Chip Load (mm/tooth)	Depth of Cut (mm)
1	1500	0.0254	1
2	3000	0.0330	1.5
3	4500	0.0432	3
4	6000	0.0508	

different operations, as shown in Fig. 5: face milling, contouring, slotting, pocketing, spiraling, and drilling. Because multiple process parameters, such as feed rate, spindle speed, and depth of cut, could have affected energy consumption, nine test parts are produced with different combinations of machine parameters to investigate the relationship between the machine process parameters and the energy usage. A Taguchi technique [21] has been employed to design the experiments to ensure a fractional factorial combination for each of the process parameters in each operation. TABLE III shows the levels chosen for the depth of cut, chip load (thickness of chip removed by one cutting edge of the tool), and spindle speed used to machine the nine parts. The feed rate f (mm/min) is obtained as the product of the spindle speed (RPM), the number of tool teeth, and the chip load (mm/tooth).

B. Data Post-Processing for Energy Prediction Model

The data from the MTConnect agent needs to be processed into a condensed and more manageable data structure. Data obtained from the MTConnect agent can be categorized broadly into three sets as shown in Fig. 6: direct data, derived data, and simulated data. The direct data is obtained directly from the data sources (in this case, the Fanuc controller and the HSPM). The derived data is the data that can be easily computed from MTConnect data using simple calculations. Specifically, we compute the total energy, average feed rate, spindle speed, and length of tool path in x-and y-direction over the duration of a block of NC (numeric control) code, which corresponds to a single cut or a machining operation. From length of tool paths in the x- and the y-direction, the length of tool path and the direction of the cut can be easily computed. A sequence of cuts then constitutes a cutting process with a tool path. The direct and derived data sets are accurate even if there are interruptions in the data stream.

The third (simulated) data set consists of the data generated by simulating the cutting process. The simulated data includes (among others) the depth of cut and the cutting strategy, which are the features employed for the energy prediction model. Because the machine tool does not recognize the dimensions of the workpiece, these features are obtained using cutting simulations with a workpiece. The cutting simulation involves

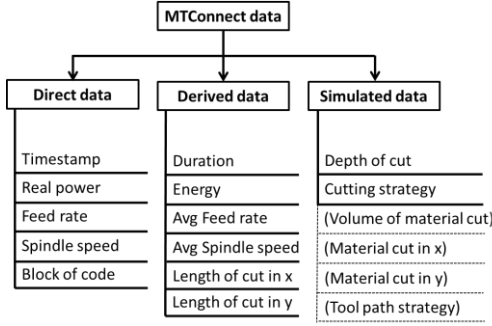


Fig. 6. Categorization of types of manufacturing data as obtained from MTConnect.

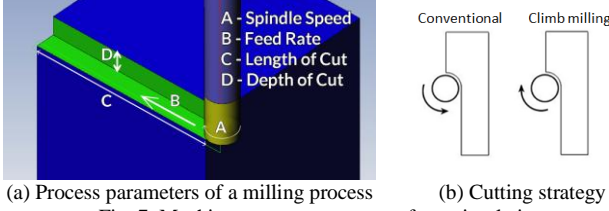


Fig. 7. Machine process parameters from simulation

forming a mesh of elements for the workpiece geometry and tracking the elements removed during each cut. The depth of cut is computed by tracking both the tool path recorded by MTConnect and the change in the dimension of workpiece in the simulator. The cutting strategy, namely climb milling or conventional milling, is determined from the cutting simulation by tracking the direction of angular rotation of the tool and the number of elements being cut on either side of the centerline of the tool. The cutting simulation requires an uninterrupted data stream during machining; otherwise, the simulated data generated would not be accurate. Fig. 7 shows the process parameters of a milling process.

The data generated from the cutting simulation is then combined with the direct and derived data for each block of the NC code. We use only a subset of the process parameters and data generated as input features for the energy prediction model. Particularly, we focus on the input features that are generated from the NC code with the aid of the simulator instead of the measurements from the sensors through MTConnect so that the knowledge model can be used to predict energy usage before actually operating the target machine. Each individual cutting operation is described by a set of machine parameters. The i th machine operation is represented by the input feature vector $\mathbf{x}^{(i)}$, which includes:

- $x_1 \in \mathbb{R}$ (B in Fig. 7) Feed rate: the velocity at which the cutter is fed, which can be retrieved from the controller data
- $x_2 \in \mathbb{R}$ Spindle speed (A in Fig. 7(a)): rotational speed of tool, which can be retrieved from the controller data
- $x_3 \in \{1, 2, 3, 4\}$ Active tool cutting direction (B in Fig. 7 (a)) (1 is for x -axis, 2 for y -axis 3 for z -axis and 4 for x - y axes): the indicator about the cutting direction of tool, which can be determined by the lengths of cut in the x -, y - and z -direction.

- $x_4 \in \mathbb{R}$ Depth of cut (D in Fig. 7(a)): the actual depth that the tool is removing material, and that can be obtained from cutting simulation
- $x_5 \in \{1, 2, 3\}$ Cutting strategy (1 is for conventional, 2 for climbing and 3 for both): the method for removing material, and obtained from cutting simulation.
- $x_6 \in \mathbb{R}$ Length of tool path (C in Fig. 7(a)): the total length of tool path in a single cut in an NC code block, which can be computed using the length of tool path in the x - and y -direction.

As an output response, the energy consumption for each NC block for each individual cut is used. The energy consumption, $y^{(i)}$, for block i , is calculated from the power consumption data obtained through MTConnect as:

$$y^{(i)} = \sum_{k=1}^{N_p} P_k^{(i)} \times t_k, \quad (1)$$

where $P_k^{(i)}$ denotes the k^{th} power consumption value during block i for a time duration of t_k and N_p denotes the number of power consumption values (i.e., number of data points in the retrieved power time series) obtained during the operation of that NC block. The time step t_k depends on the sampling rate of measuring the power time series in the block.

Fig. 8 shows the data obtained by the systematic data condensation and categorization process. Each line represents one NC code block containing the associated machine process parameters and the corresponding energy consumption. Using the input-output data set $\{(x^{(i)}, y^{(i)}) | i = 1, \dots, D\}$, where D is the number of NC code blocks, the goal of this study is to construct the energy prediction function $Y = F(\mathbf{x})$. Note that we first predict the energy per length of tool path $y = f(x_1, x_2, x_3, x_4, x_5)$, which will be discussed in the next section. The energy usage for a single cut in one NC code block is then computed as $Y = F(\mathbf{x}) = f(x_1, x_2, x_3, x_4, x_5) \times x_6$. By doing this, we can omit the obvious linear relationship between the energy usage and the length of tool path and focus on learning the influence of other machine parameters on energy usage. For brevity, we denote the feature vector as $\mathbf{x} = \{x_1, x_2, x_3, x_4, x_5\}$ (i.e., ignoring x_6 , the length of tool path) in the discussion below. TABLE III summarizes the data collected from the nine parts machined using different machine process parameters. A total of 1,650 pairs of input machine parameters and the corresponding energy output for the face

Timestamp	Code	Code2	Code3	Energy (J)	Duration (s)	Feed rate (mm/min)	Spindle speed (RPM)	Length of Cut (mm)	Depth of Cut (mm)
13.41	G01	G28	Z0	10.99	0.5	0	0	0	0
14.01	G04	X1.0000		43.25	2.36	0	0	0	0
16.37	G30	X0.0000	Y0.0000	10.62	0.58	0	0	0	0
16.95	G04	X1.0000		5.8	0.3	0	0	0	0
17.25	G90	G17	G00	39.49	0.88	17218.33708	0	48.32658916	0
18.13	G04	X1.0000		7.2	1.19	125	11.06666667	0	0
19.32	G43	Z1.0000	H2	1.8	0.59	14750	1411.35	0	0
19.91	G04	X1.0000		2.64	0.87	0	1499	0	0
20.78	Z-3.0000			8.12	3.3	69.54984894	1499	0	0
24.08	G04	X1.0000		2.81	1.22	0	1499	0	0
25.3	Y30.0000			461.25	52.31	75.98547401	1503.208716	66.574	0
77.61	G04	X1.0000		8.45	1.2	0	1499.082645	0	0
78.81	X-28.4130	Y31.5880		27.61	2.69	75.71851852	1503.407407	3.549981549	0
81.5	G04	X1.0000		8.28	1.17	0	1499.084746	0	0
82.67	F-31.5880			653.77	49.79	75.98473896	1503.280321	63.176	0
132.45	G04	X1.0000		5.07	0.9	0	1499	0	0
133.36	X-25.2380			23.23	2.7	74.00738007	1500.143911	3.175	0
136.06	G04	X1.0000		0.96	0.89	0	1502.111111	0	0
136.95	Y31.5880			482.96	49.88	75.74864702	1503.044899	63.176	0
186.83	G04	X1.0000		6.18	0.92	0	1499	0	0
187.75	X-22.0630			13.58	2.75	66.91304348	1501.210145	3.175	0

Fig. 8. Manufacturing process data after condensation and categorization.

TABLE III. Machining time and data size for the 9 experiments.

Part No	1	2	3	4	5	6	7	8	9
Cycle Time (sec)	6183.2	4208.3	2801.3	5219.3	3580.8	2308.9	4745.7	3259.9	1698.6
Data Size (MB)	25.26	17.16	11.32	21.66	16.24	9.623	23.73	13.75	7.05
Data Processing Time (sec)	5699.2	4186.6	2883.8	4962.3	3283.9	2574.7	4867.7	3212.5	1632.9
Number of Blocks (Entire process)	1415	1232	1071	1146	965	795	1007	831	528
Number of Blocks (Face milling)	226	146	78	232	152	78	234	151	78

milling process data are used for this study.

III. GAUSSIAN PROCES REGRESSION MODEL APPROACH FOR ENERGY USAGE PREDICTION

The data processed by the intelligent data management agent can be used to extract knowledge, which is represented in the form of regression model $y = f(\mathbf{x})$ about the target machine, by applying various machine learning algorithms. In this study, we predict the energy consumption Y (output) corresponding to the machine operational parameters \mathbf{x} (input feature vector) by constraining the prediction function $Y = F(\mathbf{x}) = f(\mathbf{x}) \times \text{length of tool path}$. We model the energy per length of tool path $f(\mathbf{x})$ using GP regression. For computational efficiency, we propose a Collective Gaussian Process (CGP) approach, in which the overall energy prediction function is represented by a set of local energy prediction functions constructed by GP regression.

The CGP extracts two types of information from each data set, $\{(\mathbf{x}^i, y^i) | i = 1, \dots, s\}$ where s is the number of data (in our case, the number of NC code blocks) in the training data set:

- The distribution of input machine parameter vector \mathbf{x} , characterizing machine operations conducted in the current data set. The distribution is represented by the probability density function $p(\mathbf{x})$ constructed using Gaussian Mixture Model.
- The energy prediction model from input machine parameter vector \mathbf{x} to the corresponding energy usage y for each NC code block. The energy prediction (per unit length of tool path) model $y = f(\mathbf{x})$ is constructed using Gaussian Process.

The CGP is a collection of pairwise models, $\{(f_i(\mathbf{x}), p_i(\mathbf{x})) | i = 1, \dots, N\}$ where N is the number of local GPs. Each pair $(f_i(\mathbf{x}), p_i(\mathbf{x}))$ represents the model extracted from the i^{th} data set; $f_i(\mathbf{x})$ is a local energy prediction model whose input domain is described by $p_i(\mathbf{x})$. CGP uses the pairwise models to predict the energy consumption for a new part from its input features.

A. Gaussian Mixture Model

The distribution of input feature vector \mathbf{x} (the machine operational parameter) for each data set is described by a probability density function (PDF) constructed by the Gaussian Mixture Model (GMM). A GMM can be regarded as a multivariate PDF written in terms of the weighted sum of the

Gaussian PDF (GPDF) For the machine control parameter vector \mathbf{x} , the multivariate PDF $p(\mathbf{x})$ is expressed in terms of a linear combination of K probability density functions:

$$p(\mathbf{x}) = \sum_{j=1}^K \phi_j g_j(\mathbf{x}), \quad (2)$$

where $g_j(\mathbf{x})$ is the GPDF for the j^{th} mixture component of the GMM and ϕ_j , $\sum_{j=1}^K \phi_j = 1$, is the corresponding mixture weight. The j^{th} GPDF $g_j(\mathbf{x})$ can be fully described by its mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$ [21]:

$$g_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_j|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)\right), \quad (3)$$

where $n (= 5$ in this study) is the dimension of the input feature vector \mathbf{x} . To construct the PDF $p(\mathbf{x})$, the parameters $\boldsymbol{\phi} = \{\phi_1, \dots, \phi_K\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}$ are determined from the measurement (training) data set $\{\mathbf{x}^i | i = 1, \dots, s\}$, where \mathbf{x}^i is the i^{th} input feature vector and s denotes the size (number of input feature vectors) of the training data set. The parameters are optimized to minimize the log-likelihood of the data by using the Expectation-Maximization (EM) algorithm [22, 23].

B. Gaussian Process

The energy per unit length of tool path prediction function $f(\mathbf{x})$ is modelled by Gaussian Process (GP) using each (supervised) training data set $\{(\mathbf{x}^i, y^i) | i = 1, \dots, s\}$. A GP is a collection of random variables, any finite set of which has a joint Gaussian distribution [9]. Treating the values of the unknown function $f(\cdot)$ as a collection of random variables, GP describes a distribution over functions [9]. Since a multivariate Gaussian distribution of a function $f(\cdot)$ can be fully specified by its mean and covariance, a GP is completely described by its mean function $m(\cdot)$ and the covariance function $k(\cdot, \cdot)$. Here, $m(\cdot)$ and $k(\cdot, \cdot)$ are not constant parameters but functions incorporating the prior knowledge about the unknown function $f(\mathbf{x})$. That is, the mean function $m(\cdot)$ captures the prior mean value at the corresponding data point, which is usually assumed to be zero. The covariance function $k(\cdot, \cdot)$ measures the similarity between the data points, through describing the underlying structure of the function that the GP tries to approximate.

In GP, given the measured energy consumptions $\mathbf{f}^{1:s} = \{f^1, \dots, f^s\}$, where s is the total number of NC code blocks in the training data, the energy consumption $f^{new} (= f(\mathbf{x}^{new}))$ for a new vector of machine process parameters \mathbf{x}^{new} follows a multivariate Gaussian distribution given as [9]:

$$\begin{bmatrix} \mathbf{f}^{1:s} \\ f^{new} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}^{new}, \mathbf{x}^{new}) \end{bmatrix}\right), \quad (4)$$

where \mathbf{K} is the covariance matrix (kernel matrix) whose (i, j) th entry is $\mathbf{K}_{ij} = k(\mathbf{x}^i, \mathbf{x}^j)$ for $i, j = 1, \dots, s$, and $\mathbf{k}^T = (k(\mathbf{x}^1, \mathbf{x}^{new}), \dots, k(\mathbf{x}^s, \mathbf{x}^{new}))$. Note that the vector \mathbf{k} contains the evaluations of the covariance functions between the new input parameter vector \mathbf{x}^{new} and every input vector in the training data, which are used to compute the prediction of energy consumption for f^{new} . We assume that the energy consumption is measured with additive Gaussian noise with zero mean.

The value of the covariance function $k(\mathbf{x}^i, \mathbf{x}^j)$ quantifies the similarity between two input vectors \mathbf{x}^i and \mathbf{x}^j . We use a squared exponential covariance function [24] whose evaluation between the two input machine parameters $\mathbf{x}^i = (x_1^i, \dots, x_q^i, \dots, x_n^i)$ and $\mathbf{x}^j = (x_1^j, \dots, x_q^j, \dots, x_n^j)$ is expressed as

$$k(\mathbf{x}^i, \mathbf{x}^j) = \tau^2 \exp\left(-\frac{1}{2}(\mathbf{x}^i - \mathbf{x}^j)^T \text{diag}(\boldsymbol{\lambda})^{-2}(\mathbf{x}^i - \mathbf{x}^j)\right). \quad (5)$$

The covariance function is described by the hyper parameters τ and $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_q, \dots, \lambda_n)$. The term τ denotes the amplitude of covariance function and the length scale λ_q quantifies the relevancy of the q^{th} input feature parameter x_q in predicting the energy consumption. A large length scale indicates weak relevance, while a small length scale implies strong relevance, of the corresponding machine parameter in predicting energy consumption. The hyper parameters can be learned from the data by maximizing the log-likelihood of data. In this study, we use a machine learning module, scikit-learn built using Python [25], to construct GP regression models. Once $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$ is determined, the importance of each input feature in the prediction can be studied.

GP predicts function value f^{new} for the new machine parameters \mathbf{x}^{new} in a probabilistic framework. Since the conditional distribution on any subset of data that is Gaussian is also Gaussian, the posterior distribution on f^{new} given the training data set $\mathbf{D}^{1:s} = \{(\mathbf{x}^i, y^i) | i = 1, \dots, s\}$ and the new machine parameter \mathbf{x}^{new} can be expressed as a 1-D Gaussian distribution as [8]:

$$\begin{aligned} p(f^{new} | \mathbf{D}^{1:s}, \mathbf{x}^{new}) \\ = N(f^{new}; \mu(\mathbf{x}^{new} | \mathbf{D}^{1:s}), \sigma^2(\mathbf{x}^{new} | \mathbf{D}^{1:s}) + \sigma_{error}^2). \end{aligned} \quad (6)$$

The posterior distribution $p(f^{new} | \mathbf{D}^{1:s}, \mathbf{x}^{new})$ can be fully described by the mean μ and the variance σ^2 , which are scalar parameters that can be expressed as [8]:

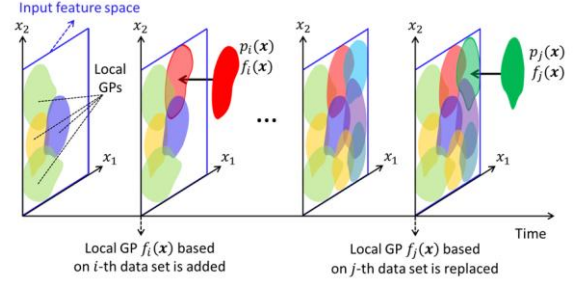


Fig. 9. Updating procedure of the collective Gaussian Process.

$$\mu(\mathbf{x}^{new} | \mathbf{D}^{1:s}) = \mathbf{k}^T (\mathbf{K} + \sigma_{error}^2 \mathbf{I})^{-1} \mathbf{f}^{1:s}, \quad (7)$$

$$\sigma^2(\mathbf{x}^{new} | \mathbf{D}^{1:s}) = k(\mathbf{x}^{new}, \mathbf{x}^{new}) - \mathbf{k}^T (\mathbf{K} + \sigma_{error}^2 \mathbf{I})^{-1} \mathbf{k}. \quad (8)$$

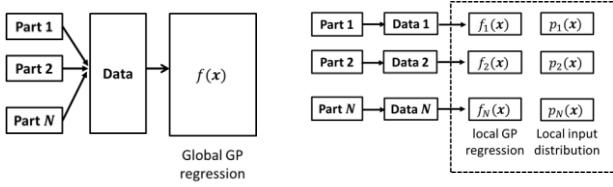
The mean $\mu(\mathbf{x}^{new} | \mathbf{D}^{1:s})$ can be interpreted as the predicted energy consumption for \mathbf{x}^{new} and the variance $\sigma^2(\mathbf{x}^{new} | \mathbf{D}^{1:s})$ as the associated uncertainty in the prediction. That is, the GP not only provides the prediction function but also quantifies the confidence level in the prediction.

C. Collective Gaussian Process (CGP)

Fig. 9 depicts how the overall energy prediction function is constructed by the collective Gaussian Process concept and updated with each new data set. Whenever the new data set (i^{th} data set) arrives from the data management agent, the local energy prediction function $y = f_i(\mathbf{x})$ is constructed based on GP regression (described in section II). In addition, to describe the range of input features in the i^{th} data set, the probability density function $p_i(\mathbf{x})$ for the input feature \mathbf{x} is constructed based on the Gaussian Mixture Model. The local GP regression models are added until the global input feature space is packed by the local GP models. If the overall input feature space is fully occupied by the local GP models, the newly constructed local GP based on the new measurement data replaces the outdated (old) local GPs whose input feature distribution is the closest to that of the new local GP. Therefore, we can continuously update the overall energy prediction function by managing the set of local energy prediction functions. Here we make a strong assumption that the local energy prediction function that is constructed based on a newly measured data set represented the most current energy consumption pattern of the target machine. This assumption essentially takes into consideration the changes of the machine condition over time.

Because each local GP regression model is constructed using only a subset of the entire data, the computational complexity of CGP is lower than that of the full GP model. While the computational complexity for learning the global GP model is $O(s^3)$ where s is the number of training data points, the computation cost for CGP is $O(s^3/N^3)$ where N is the number of local GPs. In addition, the CGP method replaces the old data points by simply changing the associated local GP function, which greatly simplifies the update of the regression model and the management of total complexity of the overall regression model.

Given the set of local energy prediction functions (local GPs), the prediction of the target value for a new input feature can be computed as a weighted sum of the predicted values by the local GPs. Different approaches have been reported for



(a) Global GP regression (b) Collective GP regression
 Fig. 10. Comparison between the global and the online collective GP regression models.

weighing the local GPs. For example, Shi et al. [15] used the mean of the predictions (equal weights) by the local GPs as the target prediction. Nguyen-thong and Peters [14] proposed the use of the geometric distance between the new input data and the center of the local data set as a weight for computing the target prediction value. The inverse of the variance computed by each local GP has also been used as a weight. Therefore, the predictions made by the local GPs with low variance, i.e., high confidence, are weighted more for the final prediction [13, 15, 17].

In this study, we use the PDFs constructed using GMM to find the weights for the local GPs. Given N pairs of energy prediction function $f_i(\mathbf{x})$ and the corresponding PDF $p_i(\mathbf{x})$ for the input feature vector \mathbf{x} , $\{(f_i(\mathbf{x}), p_i(\mathbf{x})) | i = 1, \dots, N\}$, the energy usage y^{new} for a new vector of machine parameters \mathbf{x}^{new} is computed as a weighted sum of the predicted values by all local energy prediction functions as

$$y^{new} = f(\mathbf{x}^{new}) = \sum_{i=1}^N w_i f_i(\mathbf{x}^{new}). \quad (8)$$

The weighting coefficient w_i for the i th local energy prediction function is determined as

$$w_i = \frac{p_i(\mathbf{x}^{new})}{\sum_{j=1}^N p_j(\mathbf{x}^{new})}, \quad (9)$$

which quantifies the closeness of the new data point \mathbf{x}^{new} to the i th data set. The use of probability in finding the weight is more robust than the use of geometric distance between the center of data set and \mathbf{x}^{new} , in that the computation of probability is independent of the scales in \mathbf{x}^{new} (the different scales in \mathbf{x}^{new} affect the geometric distance measure). Furthermore, the dispersion of input features (higher moments) can also be accounted for by $p_i(\mathbf{x})$. Note that if the entire data set is used (i.e., $N = 1$) in the Gaussian Process, the result becomes a global GP.

IV. APPLICATION TO ENERGY MONITORING

The proposed approach is demonstrated using the measured energy consumption data obtained from the Mori Seiki NVD1500DCG machine tool. The experimental data from machining the nine parts as described earlier in Section II are employed. Specifically, the data collected for the face milling operation is used in this demonstration study (Table III). The concept can be equivalently applied to other machine operation types. In total, there are 1,650 pairs of machine operation feature vector \mathbf{x} and energy consumption data y .

Fig. 10 depicts the procedure for constructing the global GP regression model and the collective GP regression model for predicting the energy consumption of the target machine. The global GP regression uses the entire experimental data for all nine test parts to construct the overall energy prediction function $f(\mathbf{x})$. For CGP model, the local energy prediction functions $f_i(\mathbf{x})$ and the probability distribution functions $p_i(\mathbf{x})$ using a subset of data coming from each test part are constructed.

A. Model Validation

In the following, we first assess the accuracy of models and compare the global GP and the collective GP approaches. We then illustrate the use of the GP model for the energy prediction application. The data collected from the nine test parts are used to assess the GP machine learning approaches. We use 80 % of the data for each part to construct the local GP regression model and the input feature probability density function. In addition, the training data (80 %) from each part is aggregated in a global training data set and are used to construct the global GP model. The remaining 20 % of the data for each part are then aggregated into the test data set and used to test the accuracies of the energy predictions by both the global GP and the collective local GP models.

The accuracy of the energy prediction is measured in terms of the relative absolute error (RAE) and the relative total error (RTE). The RAE is expressed as

$$\text{RAE} = \frac{\sum_{i=1}^t |f(\mathbf{x}^{(i)}) - y^{(i)}|}{\sum_{i=1}^t y^{(i)}}, \quad (10)$$

where t is the number of test data points. The RAE measures the averaged absolute error between the predicted energy usage $f(\mathbf{x}^{(i)})$ and thus quantifies the errors in the predictions of energy for every single machine operation by input feature vectors $\{\mathbf{x}^{(i)} | i = 1, \dots, t\}$ from the test data set. The RTE is expressed as

$$\text{RTE} = \frac{\sum_{i=1}^t f(\mathbf{x}^{(i)}) - \sum_{i=1}^t y^{(i)}}{\sum_{i=1}^t y^{(i)}}, \quad (11)$$

where $\sum_{i=1}^t f(\mathbf{x}^{(i)})$ is the predicted total energy consumption and $\sum_{i=1}^t y^{(i)}$ is the true total energy consumption for the machine operations. The RTE thus quantifies the error in the total energy consumption but not the errors in an individual energy value prediction by $f(\mathbf{x})$.

The predicted energy usage by the global GP and the collective GP models for the test data sets are shown in Fig. 11(a) and (b), respectively. As shown in the figures, the energy consumptions predicted by the global GP and the collective GP have generally good agreement with the true energy usage from the test data. Figures 12(a) and (b) show histograms for the errors between the predicted and the true energy consumptions for the two models. It can be seen from the distribution that the errors for most of the test data are zero. The averaged absolute errors for the global GP and the collective GP are 21.80J and 26.84J, respectively, which

correspond to 13.45 % and 16.56 % of the average energy usage (162.05J) for face milling operation in the test data set $\{\mathbf{x}^{(i)}|i = 1, \dots, t\}$. It is worth noting that the relative total error for the energy prediction is very small for both the global GP and the collective GP with 1.45 % and -1.55 %, respectively, of the total true energy consumption (44726.4J). In other words, the result illustrates that proposed approach predicts energy usage accurately since the predicted total energy consumption approaches the true total energy consumption as the relative total error goes to zero.

We repeat the experiments 100 times by randomly selecting 20 % of the data set from each part for testing the prediction models. TABLE IV summarizes the accuracy measures showing the relative absolute error and the relative total error. It can be seen that the collective GP approach compares fairly well with the global GP approach. On the other hand, as shown in TABLE IV, the computational time for the collective GP approach for constructing the local energy prediction functions and the probability density functions is significantly less than that of the global GP. The larger error rate for CGP is mainly due to the over-fitting of each local GP model into each local data set. In this case, the predicted energy corresponding to an input feature displaced far from local data sets ends up having a larger generalization

TABLE IV. Comparison of the averaged prediction accuracies and the computational times between the global and collective GP regression.

	Accuracy		Computational time	
	RAE (%)	RTE (%)	Learning (sec)	Testing (sec)
Global GP	13.95	1.34	28.27	0.06
CGP	16.64	1.93	3.76	0.06

error. The issue can be resolved if a larger number of local data sets are used, in which each test input feature will be closely located to one of the local data sets. The optimum number of N can be determined considering the data size being accumulated.

B. Comparison of Cutting Strategies

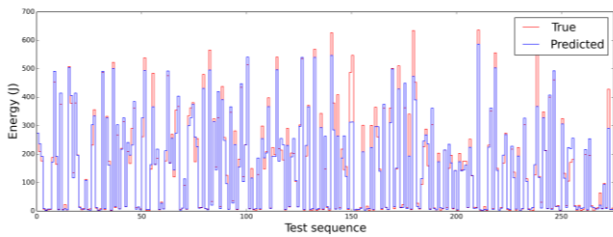
One application of the constructed energy prediction function is to compare the energy usages for machining a part with different strategies. A machining strategy is described by the NC code, from which the set of input feature vectors $\{\mathbf{x}^{(i)}|i = 1, \dots, s\}$ can be extracted.

To illustrate, we build the energy prediction function using the experimental data from Part 1 to Part 6 using the CGP model and compared energy usage for three different cutting strategies: (1) with small depth of cut (Part 7), (2) with moderate depth of cut (Part 8), and (3) large depth of cut (Part 9). As shown in TABLE V, the cutting strategy with large depth of cut results in the minimum energy consumption among the three strategies. The total energy usage of each part is predicted with relative total error of less than 5 %. Comparison among the predicted energy clearly shows that the cutting strategy with large depth of cut is efficient in terms of the energy usage. The result is straightforward in that the strategy with larger depth of cut requires a smaller number of NC code blocks and thus shorter total length of tool path for machining the same part. It is interesting to note that the mean energy per unit length (or per NC code block) for larger depth of cut is higher than that for the other cases, which possibly imply that a larger cutting force associated with larger depth of cut increases the energy density for unit operation.

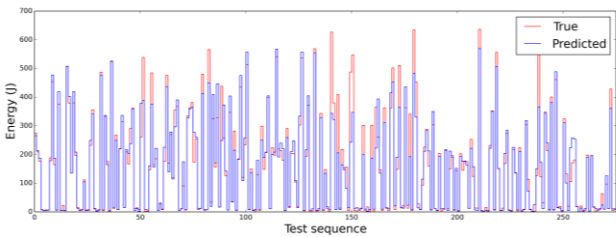
V. SUMMARY AND DISCUSSION

This paper describes a monitoring system architecture that is designed to acquire operation and power consumption data systematically from a milling machine using Gaussian process regression. The raw sensor data is collected by the automated data acquisition system via the MTConnect agent. A data post-processor, including cutting simulator, is developed to process the raw data and generate the operation parameters that can be used for further processing. Using the operation parameters as input features and the power consumption data as output, we apply Gaussian Process approaches to demonstrate the development of data-driven energy prediction model for the face milling operation.

In this study, nine experimental test artifacts has been manufactured using a Mori Seiki NVD1500DCG milling machine. As shown in TABLE III, depending on the operation parameters used, the data collected is in the order of 20 MB that can be generated through MTConnect for a single part over

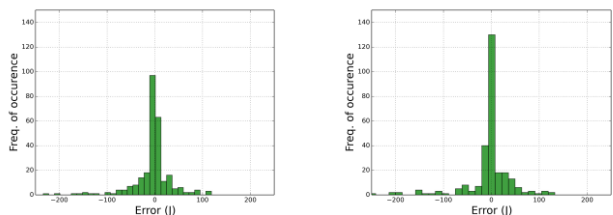


(a) Energy prediction by global GP



(b) Energy prediction by collective GP

Fig. 11. Comparison of energy predictions; true total energy is 44726.4J, the total predicted energy by the global GP model is 45376.1J (error is 1.45 %) and the total predicted energy by the collective GP model is 44035.0J (error is -1.55 %).



(a) Global GP

(b) Collective GP

Fig. 12. Comparison of errors in energy predictions; the averaged absolute error by the global model is 21.80J (RAE = 13.45 %), and the averaged absolute error by the collective model is 26.84J (RAE = 16.56 %).

TABLE V. Comparison of energy efficiencies for different cutting strategies.

Strategies	No. of blocks	Mean energy per length (J)	Mean energy per block (J)	True total Energy (J)	Predicted total energy (J)	RAE(%)	RTE(%)
Small depth of cut	234	4.12	140.38	32,850	32,692	9.88	-0.67
Medium depth of cut	151	4.41	150.16	23,426	23,672	18.10	1.05
Large depth of cut	78	4.63	157.76	12,305	11,766	20.13	-4.38

the time duration, ranging from about 1700 to 6200 seconds. Furthermore, the data post-processing and cutting simulation have taken almost the same amount of time to generate useful data.

To enable the use of predictive models derived by the described monitoring framework in industrial environments, future work will focus on developing a real-time online monitoring system that can directly integrate the data acquisition and data post-processing with the machine learning process, which is the aim of our current research efforts. Specifically, a data management strategy with a three level buffer schemes (for the raw data, the derived data, and the simulated data) is being designed to facilitate the management and efficient processing of the data. Furthermore, the collective Gaussian Process model, which processes a subset of the data set at a time, will be modified to integrate the real-time machine learning process that includes other machine operations, such as contouring, slotting, pocketing, spiraling, and drilling, as the data are being generated.

VI. ACKNOWLEDGEMENT AND DISCLAMIER

The authors acknowledge the support in part of the Smart Manufacturing Systems Design and Analysis Program at the National Institute of Standards and Technology (NIST), Grant Numbers 70NANB12H225 and 70NANB12H273 awarded to University of California, Berkeley, and to Stanford University respectively. In addition, the authors appreciate the support of the Machine Tool Technologies Research Foundation (MTTRF) and System Insights for the equipment used in this research. Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for the purpose.

REFERENCES

- [1] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, "Advanced monitoring of machine operations," *CIRP Annals-Manufacturing Technology*, 50, 2010, pp. 717-739
- [2] MTConnect Standard v1.2.0, MTConnect Insitute, 2012, <http://www.mtconnect.org/downloads/standard.aspx>.
- [3] A. Vijayaraghavan, and D. Dornfeld, "Automated energy monitoring of machine tools," *CIRP Annals - Manufacturing Technology*, 59, 2010, pp. 21-24.
- [4] N. Diaz, M. Helu, A. Jarvis, S. Tonissen, D. Dornfeld, and R. Schlosser, "Strategies for minimum energy operation for precision machining," *Proc. MTTRF 2009 Annual Meeting*, China, 2009.
- [5] E. Dimla, and S. Dimla, "Sensor signals for tool-wear monitoring in metal cutting operations—a review of methods," *International Journal of Machine Tools & Manufacture*, 40, 2000, pp. 1073-1098.
- [6] B. Sick, "On-line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research," *Mechanical Systems and Signal Processing*, 16(4), 2002, pp. 487-546.
- [7] P. Benardos, and G. Vosniakos, "Predicting surface roughness in machining: a review," *International Journal of Machine Tools & Manufacture*, 43, 2003, pp. 833-844.
- [8] S. Shin, J. Woo and S. Rachuri. "Predictive analytics model for power consumption in manufacturing," in *Proceeding of the 21st CIRP on Life Cycle Engineering*, 2014, pp. 153-158.
- [9] C. Rasmussen, and C. Williams, "Gaussian Process for machine learning", MIT Press, 2006.
- [10] J. Quiñero-Candela, and C. Rasmussen. "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, 6, 2005, pp. 1939-1959.
- [11] E. Snelson, and Z. Ghahramani., "Sparse Gaussian processes usingpseudo-inputs," *Advanced Neural Information Processing Systems*, 18, 2006, pp.1257-1264.
- [12] J. Hensman, N. Fusi, and N. Lawrence, "Gaussian Processes for big data," *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, Bellevue, WA, USA, 2013.
- [13] V. Tresp, "A Bayesian committee machine," *Neural Computation*, 12(11), 2000, pp. 2719-2741.
- [14] D. Nguyen-tuong, and J. Peters, "Local Gaussian process regression for real time online model learning and control," *Advances in Neural Information Processing Systems*, MIT Press, 2008.
- [15] J. Shi, R. Murray-Smith, and D. Titterton, "Hierarchical Gaussian process mixtures for regression," *Statistics and Computing*, 15, 2005, pp. 31-41.
- [16] A. Ranganathan, M. Yang, and J. Ho, "Online sparse Gaussian Process regression and its applications," *IEEE Transactions on Image Processing*, 20(2), 2011, pp. 391-404
- [17] H. Xiao, and C. Echert, "Lazy Gaussian Process Committee for Real-Time online regression," in *Proceeding of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13)*, Bellevue, WA, USA, 2013.
- [18] T. Bänziger, "Analyzing the relationship between the power demand of a CNC machine tool and the surface roughness imparted on a machined part," MS Thesis, ETH Zurich, 2014,
- [19] M. Helu, S. Robinson, R. Bhinge, T. Bänziger, D. Dornfeld. "Development of a Machine Tool Platform to Support Data Mining and Statistical Modeling of Machining Processes." *Proc MTTRF 2014 Annual Meeting*, San Francisco, CA, 2014.
- [20] T. Moriwaki, "Fundamentals: sensors in mechanical manufacturing-requirements, demands, boundary conditions, signal processing, communication technique, and man-machine interfaces," *Sensors in Manufacturing*, 1, 2011, pp. 24-45.
- [21] George EP. Box, J. Stuart Hunter and W. G. Hunter, "Statistics for Experimenters: Design, Inovation, and Discovery, WILEY, 1978.
- [22] A. D. Reynolds, F. T. Quatieri, and B. R. Dunn, "Speaker verification using adapted Gaussian Mixture Models," *Digital Signal Processing*, 10, 2000, pp. 19-41.
- [23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maxumum Likelihood from incomplete Data via the EM Algorithms," *Journal of the Royal Statistics Society, Series B*, 39(1), 1977, pp. 1-38.
- [24] R. M. Neal, "Bayesian learning for neural network," Springer, New York, 1996.
- [25] Pedregosa *et al.*, "Scikit-learn: machine learning in Python," *JMLR*, 12, 2011, pp. 2825-2830.