Equational Model Guided by Real-time Sensor Data to Monitor Industrial Robots

Aomar Osmani[1], Pegah Alizadeh [2] and Christophe Rodrigues[2]

Abstract— **The monitoring of industrial robots is often ensured by generic simulators which model the equational aspect of the target machines. We propose an original approach to complete the equational simulator of a milling machine using the accumulated data from the employed sensors in the real workshops. This approach creates a specific simulator for each machining situation by taking the triplet (material, cutting tool, workpiece) into account. This improvement brings great added value to the industrial experts and improves the efficiency of industrial robots. It allows them to better follow and interpret the behavior of machines during the milling process. Our proposed method does not only correct the simulator w.r.t. the real observed data, it also detects the anomalies during the real manufacturing performance. It also fixes the minor bugs along the observed real data during its continuous simulation mimicry. The additional interest of our model remains the precise definition of the complementary model between the real system and the equational simulator. This makes it possible, by using an inductive approach to search for regularities in the model in order to better interpret the structural differences between the model and the system and to better understand the situations linked to their functionalities or undesirable situations. The intensive experiments on real data validate our model and open up many perspectives for future works.**

I. INTRODUCTION

Industrial robotics presents significant challenges, especially when it associates with the automation of the robot environment [7]. Milling process optimization is one of these important challenges that aims to increase the quality of products and process flexibility or to reduce the machines operating time and material wastes or tool damages. The milling process is mainly controlled by Computer Numerical Control (CNC) [11] while the cutting robot movements w.r.t. the used metal for the manufacturing (e.g. tool's position and its angle w.r.t. the material, tool speed and orientation, ...) are controlled by specific and complex command programs (see Fig. 1). Designing a new CNC program to produce a specific manufactured workpiece involves selecting a lot of milling parameters, which is difficult even for the machining experts with several years of practice. To reduce several risks, such as tool breakage and tool vibration in the aerospace industry [17] or material damage [16] in industry 4.0, the machinist uses some simulator softwares beforehand. Nowadays, there exist several numerical control machining simulation [18] for simulating, verifying, and optimizing CNC machining, including the simulator used in the context of our industrial application: NCSimul[1] [2].

The NCSimul reads a given post-processed G-code language that tells computerized robots where to move, how fast to move, and what path to follow. This G-code identifies the tool path, and replicates the material removals by cutting automatically volumes (see Fig. 2). By this, the NCSimul then identifies all syntax errors in the code, crashes in the machining environment, and deviations from the given geometrical form for a to-be-manufactured workpiece. The NCSimul simulator mainly assists the machinists to tune a set of parameters related to the robots, materials, speed and etc during the manufacturing; these parameters are explained by details in Sec. II. The behavior of NCSimul is not completely faithful to the real manufacturing workshops and the used industrial robots, because it is not possible to model all the parameters of the machining environment mathematically. However, some works show most of these imperfections can be detected by studying and analyzing the generated signal powers [5], [9]. For example, during manufacturing, if the used metal is a soft metal as aluminium, the machinist requires less power inside the material for following the G-code. On the other hand, keeping the power stable and low generates many oscillations during the manufacturing process. The simulators such as NCSimul, are normally designed based on mechanical and physical laws, which do not take totally into account the material type, tool and machine dynamics, the environmental situations, initial material and tools conditions or any post noises/errors diagnostic [10] (the formulas are explained in Sec. II-B).

One of the main limitations of equational simulator-based approaches to monitor dynamic systems behavior is the time management. Indeed, while it is easy to model the perfect temporal behavior using equations, it is not possible to predict all the recurring damage, unforeseen events or abnormal situations due to operators and interactions with other tools and machines. Our solution aims to solve these problems by an inductive model in addition to the equational deductive model and using the data from the real system to continuously correct the simulator. The obtained results are very satisfactory as will be explained later. Some available works in the literature propose hybrid models for mixing simulator and online sensors in order to predict tool wear and cutting forces [6], [3]. A simulator can be used to describe non-measurable features by the sensors. They show

[1]Aomar Osmani is with LIPN-UMR CNRS 7030, Université Sorbonne Paris Nord, Villetaneuse, France aomar.osmani@lipn.univ-paris13.fr

[2]Pegah Alizadeh and Christophe Rodrigues are with Léonard de Vinci Pôle Universitaire, Research Center, 92 916 Paris, La Défense, France

[1]https://www.ncsimul.com/

[2]https://www.roboris.it/

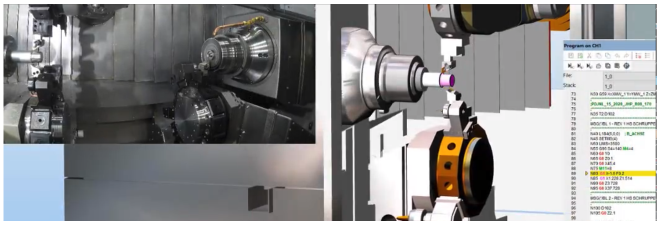

Fig. 1: The left part shows the real milling industrial robot [2] in real environment. The right one shows the G-code generated by the equational model. Our work is to improve the equational model by adding both the environment and the human constraints, only guided by the sensors data.

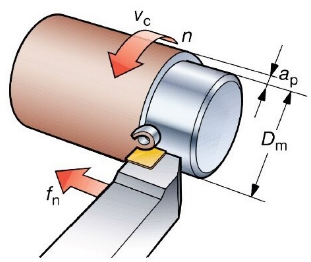

Fig. 2: Example of removed volumes of the material by the five axis milling robot. D_m, f_n v_c, n and a_p are respectively a machined diameter, a feed per revolution, a cutting speed, a spindle speed and a radial cutting depth.

that mixing simulator and real data improves the accuracy of a machine learning model in comparison to the machine learning model which relies only on simulated data or only on real data. In opposite to some works [6] that surcharge some measured parameters manually to the system, we focus on a general approach adaptable to any CNC machine without manual intervention. Among them, some works use the dedicated sensors for monitoring the energy consumption, because it allows them to reuse process parameters on similar machining via a case-based reasoning [14].

The issues of adjusting equational-based simulators are not only limited to the simple data observation, but they concern also the capitalization and analysis of the useful industrial knowledge. Knowledge models and meta-models are useful to support semantic interoperability between the observed data bases within the whole augmented simulation chain. To achieve this goal, Meskini et al. [13] introduce a knowledge-based framework that aims to support intelligent reasoning and decision making as an assistant for the industrial experts.

To improve the imperfection and difference between the real and simulation conditions during the milling process, we propose a method based on a Temporal Probably Approximately Correction (TPAC) in Sec. III and by studying phase and amplitude variations of temporal sequences between equational simulated and observed real power signals. The method is, for example, able to readjust the power signal and the pre-defined manufacturing parameters on fly. In summary the main contributions of this paper are: (1) We propose an original approach for simulating the milling and manufacturing process based on the mechanical formula and by observing real data in the industrial workshops. Our simulation concerns milling any material, tool and workpiece combination, (2) Our proposed method does not only correct the simulator w.r.t. real observed data, it can also be used for detecting anomalies during real manufacturing performances, (3) An experimental study on around 1.7 GB data provides strong empirical evidence that the simulator accuracy can be improved using sensor and real data, (4) Our experiments show that our approach is noise resistant, and it fixes minor bugs happening in real observed data during its continuous adjustment simulation.

The paper is organized as follows: we describe the problem concerning the equational simulators and observed data in Sec. II. In Sec. III, we present our probably approximately correction-based model. The paper is followed by discussing the experimental results and conclusions in Sec. IV and V.

II. Problem Description

A geometrical trajectory (G-code) of an industrial workpiece such as the axis positions of the cutting tool w.r.t. the material or the spindle torque is first designed using a software such as CAD. To conduct a successful milling process [8], [15] and avoiding any risk (e.g. material or machine damage and tool breaking), several complex process characteristics and parameters related to tool, cutting forces and accelerations should be properly tuned beforehand (see Sec. II-A). Usually, tuning these complex parameters is done by highly experienced machinists and using tools and machines designed based on mechanical and milling laws. However, any slight changes in the environment or expert situation can affect their tuning phases.

Nowadays, there exist several softwares that allow the machinists to simulate a milling example and tune its milling parameters by observing the result performances using several trials. The main problem with the current simulators is that they are designed based on the milling mechanical and physics formulas. Thus, they are not close enough to the real milling performances and therefore, they are not completely reliable for the machinists. In our proposed augmented simulator, we are interested in adapting the formula-based simulators with real-time experiments and complete their missing information into the simulated model. Some of the missing information that can not be taken into account using the milling formulas are as follows. For a given workpiece, the milling performance regarding various materials and tool types are not dependent on tool-material changing with the milling formulas. Or the tools and material conditions are not observable when the cutting tool is outside the material.

We are also interested in designing a simulator that adapts itself to correct and acceptable real-time observations. As observations are done in real workshops, the used real performances may contain some flawed, noisy data or some special conditions. For this reason, we are interested in an adapted simulator that detects flaws and anomalies in real observed data. In order to design an augmented simulation model closer to the real milling conditions, we focus on power signals in the real-time manufacturing and the power signals generated by the formula-based simulators. The power plays a key role in evaluating the productivity and manufacturing quality [9], [5]. It reflects the complex behavior of the material-cutting tool-workpiece assembly for each time step and contains the manufacturing performance history.

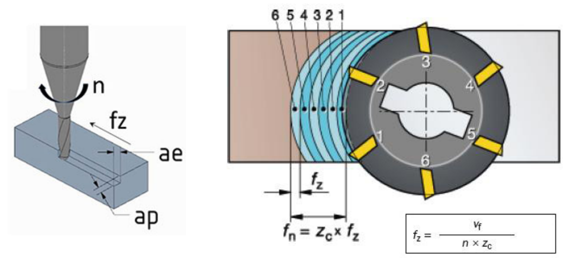

Fig. 3: **Left:** example of radial (a_e) and axial (a_p) depth for the generated cut using the cutting tool. **Right:** example of a cutter tool with $z_c = 6$ teeth, spindle speed n, feed per tooth f_z and feed tool per minute v_f in relation to the material.

A. *Data Sources*

Two data sources are available in the case of our applications, including the data that come from the observed real power during the milling experiments and simulation data generated using the NCSimul simulator. The observed real-time power is collected each second during real milling fabrications by only one sensor (a digital watt-meter). The sensor measures the consumed power of machines and cutting tools in KW with three aspects: voltage, strength and dephasing.

Consequently, the total consumed power is available in each real-time step. On the other hand, the simulated data collected from NCSimul includes not only the consumed power but also the features and shape deviations of the workpiece in each tool-material-machine combination. These parameters indicate various measuring during manufacturing, such as the tool and machine positions w.r.t. the material, feed directions, tool orientation and removed volume from the material. The material removal process changes the workpiece stiffness during machining, while feed directions change the direction of the cutting force vector. Tool orientation changes the norm and direction of the cutting force. These parameters are introduced with more details in Sec. II-B. Because of the limitation of implementing sensors during the real manufacturing process, power is the only registered parameter in our real data-set. The rest of parameters are only registered using the NCSimul and are used for computing the power value. The studied database of this paper for observed and simulated environments is available on limited industrial workpieces and material types, tools and machines.

B. *Simulator Description*

The industrial system we are studying uses NCSimul [2] simulator. It uses a set of milling and mechanical formulations for the manufacturing process, namely *Sandvic* [1]. This section introduces the main formulas used in our work. The first one is: $v_f = n \times f_z \times z_c$. It shows the relation between the feed per tooth (f_z) and table feed per minute (feed rate) (v_f) w.r.t. the number of teeth (z_c) and the spindle speed (n) (see Fig. 3). According to this figure, spindle speed (n (rpm)) is the number of revolutions the milling tool makes per minute on the spindle. This is a machine oriented value, calculated from the recommended cutting speed value for an operation. Feed per minute (v_f (mm/min)) is the feed of the tool in relation to the workpiece in distance per time-unit related to feed per tooth and number of teeth in the cutter. The number of available cutter teeth in the tool varies considerably and is used to determine the table feed while the effective number of teeth (z_c) is the number of effective teeth in cut.

The relation between the feed per revolution (f_n) and the feed per minute (v_f) is given by: $f_n = \frac{v_f}{n}$, where f_n in mm/rev is a value used specifically for feed calculations and often to determine the finishing capability of a cutter. Fig. 3 shows what are feed per minute and feed per revolution. The volume of metal removed (Q in cm^3/min) is established using the values for cutting depth, cutting width (a_e and a_p in mm), and feed per minute (v_f) within $Q = \frac{a_e \times a_p \times v_f}{1000}$. Fig. 3 (left) shows the two cutting parameters a_e and a_p denoted radial depth and axial depth of the cut, respectively. Fig. 2 shows the removed particles of metal by the appropriate tool. The cutting depth (a_p) is the difference between the uncut and the cut surface in axial direction. Maximum a_p is primarily limited by the insert size and machine power. a_e is the radial width of the cutter engaged in cut.

Formula 1 calculates the net power to ensure that the machine can handle the cutter and operation. It has a direct relation with cutting depth and width parameters, feeding and a constant k_c. k_c is a material constant which is a factor used for power calculations, expressed in N/mm^2.

$$\text{net power, } P_c \text{ (kW)}: \quad P_c = \frac{a_e \times a_p \times v_f}{60 \times 10^6} \times k_c \qquad (1)$$

Another critical factor in roughing operations is torque (M_c). It is important in finishing operations and vibrations. Torque is the measure of how much immediate rotational force a spindle drive motor can generate. Formula 2 shows that the torque is dependent on the net power and the spindle speed.

$$\text{torque, } M_c \text{ (Nm)}: \quad M_c = \frac{P_c \times 3 \times 10^3}{\pi \times n} \qquad (2)$$

In order to maximize the speeds and feeds, the machining experts should have a good understanding of how the CNC machine develops and holds torque.

C. *Connection between Real system and Simulation*

In order to machine a workpiece, there are mainly two distinct stages: design, which is done using Computer-Aided Design (CAD), and the workshop process ensuring the same workpiece realization. These two separate environments can be considered as one with a computer (white collar), the other with a manufacturing machine (blue collar). This dichotomy also persists between data simulated with the computer and real data measured with real machines. The machining simulation allows the industrial experts to reduce the round trips between design and production by detecting geometrical path errors, possible collisions between the tool and the workpiece or the machine. On the other hand, analyzing and observing the consumed power by the machines allow the machinists to check if there has been too much effort that could destroy the produced workpiece quality or the machine and tool duration. This information exists but is not coordinated and is partially used. We propose a method that feed back this information the equational simulators in order to detect bad choices of upstream parameters.

III. Proposed Approach

Designing a new program to produce a specific manufactured piecework involves choosing a large number of mechanical parameters and taking into account the actual operating context to reduce risks such as tool breakage and tool vibrations. Equational models are essential to simulate operation and to prevent these problems. But it remains largely imperfect and insufficient as we will show in this study on real industrial tools. To better assist industrial experts to adjust a set of parameters related to tools, material, speed and better adapt to wear, malfunction and adjust these parameters during manufacturing, we start from the postulate that a continuous adjustment of the equational simulator is possible by analyzing the real data coming from the various sensors and by analyzing especially the power of the signal generated by the industrial machine. Our numerous experiments show that the adjustment of this power signal guided by the real data allows the equational simulator to considerably improve the quality of its predictions compared to the real system. It is therefore a concrete success of the use of the data-driven approach.

Our approach is therefore concerned with both aligning the power signals generated by the simulator and the real system. We adjust the simulator by observing the real data and the dynamics of the machines in the context of industrial manufacturing. Since each triplet (material, tool, workpiece) generates a power signal, we also propose an approach that generalizes this combination for each piecework with different tools, materials and machines. The readjustment of the simulator power signal allows better prediction of anomalies by following a given geometric trajectory (G-code). Using a corrected simulator, industrial experts can also define and adjust manufacturing parameters beforehand.

Therefore, the problem to be solved can be summarized as follows. According to the parameters defined in Table I, how an augmented equational simulator S_{AE} (to-be-learned) can be computed such that following a time line defined by R_S, for each criterion β, the probability to have a difference between R_S and S_{AE} greater than ϵ in machining time is close to zero. Assuming each criterion's contribution to the whole model is defined by a simple linear combination, then:

Notation	Explanation	Notation	Explanation
R_S	real dynamic system	S_E	equational simulator
β	each criterion	MT	machining time
S_{AE}	augmented equational simulator	n	number of criteria
D	set of data driven by R_S under nominal operating conditions		

TABLE I: List of parameters used in this section.

$$\forall t \in MT, \left(\frac{1}{n}\Sigma_{\beta=0}^{\beta=n-1}\alpha_\beta P_{MT}(|R_S^\beta(t) - S_{AE}^\beta(t)| > \epsilon\right) < \sigma \quad (3)$$

The correction of the simulator is empirically done in several steps. First, it's necessary to fix and tune the hyper-parameters allowing us to verify empirically that the model is close to the real system. Second, we propose a linear fit-by-part approach using the Legendre principle [12]: the least square regression to calculate the coefficients of the linear function by part. Finally, we compute a set of control parameters to analyse and interpret a reverse engineering process for monitoring, diagnosis or other industrial needs.

A. Tuning hyper-parameters

The main parameters to consider are: n, ϵ, σ and MT. As the only parameter concerned by the studied industrial application is power, the previous formula will be simplified:

$$\forall t \in MT, P_{PT}(|R_S^\beta(t) - S_{AE}^\beta(t)| > \epsilon) < \sigma \quad (4)$$

where ϵ, β and MT are fixed by the user and P_{PT} is the probability function. From an empirical point of view, sequences of events are known step by step, thereby it's not possible to compute the probability in the whole MT time period. In order to respect the previous Formula, we define several hyper-parameters as below:

- W_0: even if there is a difference between R_S and S_{AE} in a short period (e.g. a Dirac delta distribution with uncertainty [4]), it's not necessary to correct the simulator. We denote this period with W_0. It represents the minimum period of time where the difference is important to apply the modification in S_{AE}.
- W: as S_{AE} must follow R_S in the real time, we define a maximal window where it's possible to modify the current behavior of the simulator w.r.t. the previous mechanical equations. We denote this window with W.

The optimal value of W_0 and W can be learned using the real data D and a machine learning algorithm. This step is not developed in this work, because using an one step parameters calculation, the obtained results are close to the desired ones. By fixing the hyper-parameters, we propose a power alignment approach to build an augmented simulator.

B. Building Augmented Simulator for Power Signals

The complex application presented in Sec. II is only governed by one order criterion: the electric power. The real and simulated power signals are generated for a given workpiece and by following a unique geometrical trajectory. To see some examples, all figures in Sec. IV demonstrate how the two real and equation-based simulated signals perform.

A set of physic and mechanical formulas, partially presented in Sec. II, are used in simulating the milling process. Tuning these parameters reduces the risk and anomalies and contributes to optimize the associated industrial process. However, the theoretical equations are not sufficient to follow the behavior of the real system precisely without any drifts. This is because actual operations introduce biases and parameters that are impossible to model and predict, such as human operator pauses, programming error, premature wear and any exogenous events to the normal process. Hence, we need to provide an approach guided by real data to ensure continuous adjustment of the simulator to the real system in order to ensure a continuous monitoring. In this section, we propose a new algorithm where the existing electric power by simulator $S_E^{\texttt{power}}$ (represented by Equations 1 and 2) is guided by real data D to follow the real signal $R_S^{\texttt{power}}$ (simplified as S_E and R_S notations, respectively). Notice that we have only one criterion i.e. $\beta = \texttt{power}$, regarding Equation 3. With intensive experiments on the related data, our algorithm shows that the augmented simulator built with the equational model and adapted with the real data avoids the drifts effects and takes the parameters that are difficult to describe in a theoretical model into the account.

As data in D are labeled with the real time associated to the power measurement ($t \in \{1, \dots, MT\}$) and according to Equation 4, the main idea of our approximation algorithm is to find a mapping function f, defined step by step in window W (contiguous subset of $\{0, 1, \dots, MT\}$). It means for a given current step started at time t_c and a given window W, find $f : \{t_c - W, \cdots, t_c + W\} \longrightarrow \{t_c - W, \cdots t_c + W\}$, such that $\forall\ t_c - W \leq t \leq t_c + W$, $S_E(f(t))$ aligns $R_S(t)$. In the other hand, it minimizes the gap as follows:

$$\min_{f,W_0} \sum_{[t_i, t_i+W_0] \subset [t_c - W, t_c + W]} |a(t_i, W_0) - 1.0| + |b(t_i, W_0)| \tag{5}$$

where a and b as functions of t_i and W_0 are defined as:

$$a(t_i, W_0) = \frac{cov(A, B)}{var([S_E(f(t_i)), S_E(f(t_i + W_0))])} \tag{6}$$

$$\begin{aligned} b(t_i, W_0) = &\ mean([R_S(t_i), R_S(t_i + W_0)]) - \\ & a(t_i, W_0) \cdot mean([S_E(f(t_i)), S_E(f(t_i + W_0))]) \end{aligned} \tag{7}$$

Where $B = [S_E(f(t_i)), S_E(f(t_i + W_0))]$ and $A = [R_S(t_i), R_S(t_i + W_0)]$[3].

The following algorithms give additional details about this approach. In addition to the implementation and tuning details, Algorithm 1 presents the Continuous Readjustment Process (CAP) of the simulator to follow the real system behavior regarding the implementation of formulas 4 to 7. For the sake of convenience, Algorithm 3 (CAP$_{\texttt{partial}}$) treats part of Algorithm 1 by calculating a modification of a single step of the simulator given all the inputs, respecting all Equations 4 to 7 in approach. Algorithm 2 calculates the linear deviation per part (PLD), to be applied to the simulator, to bring it closer to the real model over a short period of time. It computes Equations 6 and 7. Finally, Algorithm 3 implements the minimization problem, presented in 5. An important variable to recover after applying the whole approach and getting the augmented simulated signal, is the difference between this achieved signal and the real signal which is calculable as the following. This can be useful in diagnosis and risk prediction tasks: $V_{\texttt{diff}} = \{(t, R_S(t_R) - S_{AE}(t_S)) \mid \forall (t_R, t_S) \in RS_{\texttt{matches}}\}$

[3] $[R_S(t_i), R_S(t_i + W_0)] = \{R_S(t_i), R_S(t_i + 1), \cdots, R_S(t_i + W_0 - 1), R_S(t_i + W_0)\}$

Algorithm 1 CAP (**In**: R_S, S_E, W, W_0, α, ϵ, δ, σ **Out**: $RS_{\texttt{matches}}$, S_{AE}, $V_{\texttt{change}}$)

$S_{AE} \leftarrow |S_E| \times [-inf]$ adjusted simulator signal
$t_S \leftarrow 0$ pointer on the simulator signal
$t_R \leftarrow 0$ pointer on the real signal
$RS_{\texttt{matches}} \leftarrow []$
$V_{\texttt{change}} \leftarrow []$ simulated signal form changes
repeat
  $R_W \leftarrow R[t_R, t_R + W]$
  $S_W \leftarrow S[t_S, t_S + W]$
  $t_1, t_2, V, RS, S_A \leftarrow$ CAP$_{\texttt{Partial}}(R_W, S_W, S_A, W_0, \alpha, \epsilon, \delta, \text{real})$
  $V_{\texttt{change}} \leftarrow V_{\texttt{change}}.\texttt{concatenate}(V)$
  $RS_{\texttt{matches}} \leftarrow RS_{\texttt{matches}}.\texttt{concatenate}(RS)$
  $t'_1, t'_2, V, RS, S_A \leftarrow$ CAP$_{\texttt{Partial}}(R_W, S_W, S_A, W_0, \alpha, \epsilon, \delta, \text{simulated})$
  $RS_{\texttt{matches}} \leftarrow RS_{\texttt{matches}}.\texttt{concatenate}(RS)$
  $t_S \leftarrow \texttt{max}(t_1, t'_1)$; $t_R \leftarrow \texttt{max}(t_2, t'_2)$
  $t_S \leftarrow t_S + W$; $t_R \leftarrow t_R + W$
until $t_S \leq |S_E|$ and $t_R \leq |R_S|$
return $RS_{\texttt{matches}}, S_{AE}, V_{\texttt{change}}$

Algorithm 2 PLD (**In**: R_S, S_E **Out**: a, b)

$a \leftarrow cov(S_E, R_S)/var(S_E)$
$b \leftarrow mean(R_S) - a \times mean(S_E)$
return a, b

IV. Experimentation

We have carried out intensive experiments on a data-set of our industrial application. All these experiments show the efficiency of our proposed approach regardless of material types (steel or aluminium), or different used tools or machines for the manufacturing process. Given the variety of data and parameters of the application, this section will present the structure of extracted data-set used by our algorithms and show some promising curves obtained to build a robust and effective augmented simulator in real time.

A. Data Description

The simulated and real data are required for various material-cutting tool-workpiece combinations. The first one is generated by the NCSimul software, and the second one is generated by the real observed data in a real factory[4]. In these experiments, we extract around 1.7 *GB* data in total. We use data for manufacturing two workpieces: GP2R and 5axes, two types of cutting tools: long (2 different cutting tools) and short (2 different cutting tools)[5], and two materials: steel and aluminium.

For each (material, cutting tool, workpiece) triplet, a large data-set indexed by process time steps is created. It contains

[4] In order to respect the anonymous submission rules, we cannot give more details on this system, it will be done if the paper is accepted

[5] The tools are: Walter F4042R.T22.025.Z03.10, Drill GUHRING ø12 ref 3470, Drill GUHRING ø12 R2 ref 3599 and Drill GUHRING ø12ref 3891.

Algorithm 3 CAP$_{\texttt{partial}}$ (**In**: R_S, S_E, S_{AE}, W_0, α, ϵ, δ, move **Out**: t_S, t_R, V_{change}, $RS_{\texttt{matches}}$, S_{AE})

$RS_{\texttt{matches}} \leftarrow []$ time matches between real and simulated signals
$V_{\texttt{change}} \leftarrow []$ simulated form changes
step $\leftarrow \alpha W_0$ hyper-parameter
$t_S, t_R \leftarrow 0$
repeat
  $S_{\text{seg}} \leftarrow S_E[t_S, t_S + W_0]$
  $R_{\text{seg}} \leftarrow R_S[t_R, t_R + W_0]$
  $a, b \leftarrow \text{PLD}(R_{\text{seg}}, S_{\text{seg}})$
  if $a = 1 \pm \epsilon$ and $S_{\text{seg}}[-1] - R_{\text{seg}}[-1] = b \pm \delta$ **then**
    for $t' = 0$ **to** $|S_{\text{seg}}|$ **do**
      $RS_{\texttt{matches}} \leftarrow RS_{\texttt{matches}}.\texttt{concatenate}(t_R + t', t_S + t')$
    end for
    $t_S \leftarrow t_S + W_0$
    $t_R \leftarrow t_R + W_0$
  else if $b \neq S_{\text{seg}}[-1] - R_{\text{seg}}[-1] \pm \delta$ **then**
    $V_{\text{change}} \leftarrow V_{\text{change}}.\texttt{concatenate}((t_S + W_0, R_{\text{seg}}[-1]))$
    $S_{AE}[t_S - W_0, t_S] \leftarrow |W_0| \times [b]$
  else if $a \neq 1 \pm \epsilon$ **then**
    if move = real **then**
      $t_R \leftarrow t_R + W \times step$
    else if move = simulated **then**
      $t_S \leftarrow t_S + W \times step$
    end if
  end if
until $t_S \leq |S_E|$ or $t_R \leq |R_S|$
return $t_S, t_R, V_{\text{change}}, RS_{\texttt{matches}}, S_{AE}$

several characteristics including tool name, machining cycle, tool (x,y,z) axis, type of intersection, type of movement, interaction mode, bloc number, relative times for several local milling processes, global time, tool family reference and several other technical characteristics. An example of the simulated data is given in Table II. Even if the data structures are modeled by the engineers for an operational system, we noticed the lack of rigor in the choice of characteristics and especially the redundancy, the duplication of certain data and lack of useful information during the manufacturing process. The generated data is far from being canonical or respecting database basic normal forms.

On the other hand, because of some limitations for applying the sensors in the real workshop, the only registered data for the real observations is the consumed electrical power (P_c). An example of the simulated and real data for the same steel-GPR2-REF3891D12 is given in Table III. It contains the real power in millisecond time-steps and three power parameters: voltage (P_v), strength (P_s) and dephasing (P_d).

B. Experimental results

From the intensive experiments carried out on the industrial manufacturing data-set presented in Sec. IV-A, we summarize, in this section, the experiments for which our approach brings real added value from an industrial point of view. The augmented simulator is used for assisting the machining experts during the workpiece manufacturing for several tasks among which:

The main additional value of our proposed augmented simulator S_{AE} is that it includes material and tools specificity of the current experiments by taking the real data into account. While S_E uses only generic equations. As shown by Figures 4 and 5, with the following values of parameters $W = 15000, W_0 = 500, \text{step} = 50$ and $\epsilon = 0.0$ for steel material and $W = 45000, W_0 = 500, \text{step} = 50$ and $\epsilon = 0.6$ for the aluminium, we can observe that by using real data, S_{AE} readjusts the temporal and amplitude desynchronization of S_E. By this way, it assists the machinists to better drive the manufacturing process. We can observe the same result when we change the material as shown by Figures 6 and 7. Notice that in all presented figures, the vertical axes represent the power, and the horizontal axes represent time steps. According to Algorithm 1 notations, the green curves correspond to R_S, the blue ones to S_E, the golden ones to S_{AE} and the red ones to $\overline{S}_E$. The step by step deviation between R_S and S_{AE} is given by the grey signals.

In the previous system S_E, initial conditions of material and tools are added manually to the system, because the used equations in S_E don't use concrete information concerning these parameters. S_{AE} includes these parameters and computes nominal power value. In particular, S_E is not able to detect the fact that the cutting tool turns in the void. Fig. 7 shows that when the cutting tool is outside of material (not cutting), the simulation returns a zero energy consumption because the mechanical formulas are independent from the command machine center and can not predict the consumed energy by the machine (see Equation 1). Our results show how the missing information can be completed by our augmented simulation.

Our proposed method does not only correct the simulator w.r.t. the real observed data, it also detects the anomalies during the real manufacturing performance. For instance, if the machine is off during the real manufacturing for a few seconds, the augmented simulator S_{AE} keeps its flow without taking the anomalies into account (see Fig. 6). In this case, the machine has been off for a while before the 20,000th time step, however our method is able to detect this behavior as an anomaly and it does not take it into account for the final results.

Furthermore, we can observe in all our experiments and particularly in Fig. 6 that our approach is noise resistant. It is for example the case if the machinist manufactures a soft material such as aluminium, it requires more time to follow the G-code. Our augmented simulator corrects the machinist's generated noise in its results.

V. Conclusions and Perspectives

In this paper, we propose and formalize an approach that integrates real and environmental constraints through sensor data analysis into equational models of simulators. This

num bloc	real-time	relative-time	tool-time	tool-seq	x	y	z	move-type	P_c
G1 X18. Z-4.017	404.6771638	0.1241704	16.0783144	2	0	0	0	CPwFO	33.1137074536456
G1 X18.093 Y8.503 Z-4.018	404.6869077	0.0097438	16.0880582	2	0	0	0	CPwFO	27.406235294754
G1 X18.189 Y8.512 Z-4.02	404.6970063	0.0100986	16.0981569	2	0	0	0	CPwFO	9.4408062581948
G1 X18.285 Y8.527 Z-4.021	404.7071812	0.0101749	16.1083317	2	0	0	0	CPwFO	27.0868594783492

TABLE II: An example of simulated signal for the GP2R workpiece, steel material and using RF100U REF 3891 D12 tool-ref. CPwFO means change of position with fixed orientation.

time	P_c	P_v	P_s	P_d
0	407	3	9	-8
1.0	403	4	6	2
2.0	400	5	5	15
3.0	393	4	5	0

TABLE III: An example of observed real power signal for the GP2R workpiece, steel material and using RF100U REF 3891 D12 tool-ref.

approach performs a continuous adjustment of the simulator w.r.t. the real system to ensure a continuous monitoring of industrial robots. We empirically evaluate our proposed approach on real industrial workpieces and show significant improvements with respect to the equational simulator model alone. The experiments show that our proposed augmented simulator has several advantages: it takes into account material stiffness by adding real data and more generally, it is adaptable to the used materials (e.g. steel or aluminium), cutting tool workpiece and tools. It can also correct the simulation results when the cutting-robot is outside the material or when the industrial process is stopped temporally for any external reason. Our proposed approach is robust toward the noisy real conditions and can solve difficult cases where the real and simulated values have a complex and non-linear differences. Moreover, it has a direct effect for the machinists to improve the continuous monitoring of industrial robots in real time and consequently speeding up the manufacturing process.

In the evaluated industrial problem, experimental results show that we obtain an appropriate industrial robot precision only by manually tuning some hyper-parameters. However, our approach is more general. It gives us a possibility to learn the hyper-parameters automatically by using machine learning approaches on the same available data. Another interesting perspective for this work is the using of cumulative changes for diagnostic and any anomaly detection. Even if this model is applied in the case of milling robots, controlled mainly by one parameter/sensor, it can be equally applied to the industrial cases requiring several control parameters.

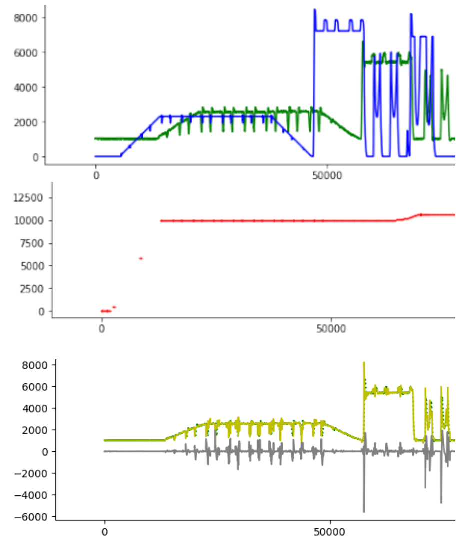

Fig. 4: Part of the power signals for the pocket milling of the **GPR2** with **aluminium** material and a **short tool**.

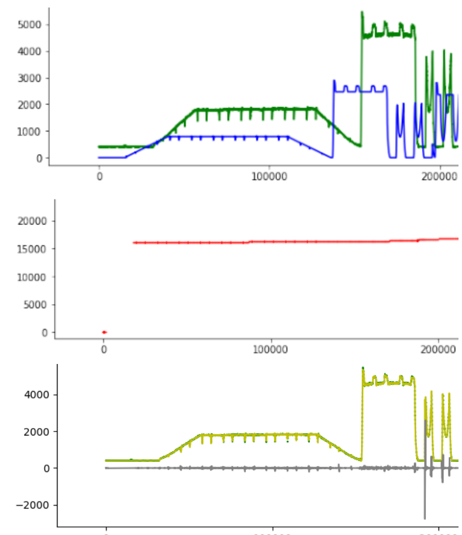

Fig. 5: Part of the power signals for the pocket milling of the **GPR2** with **steel** material and a **short tool**.

References

[1] *Sandvic Cromont*, 1942.
[2] *ncsimul software*, 2000 (accessed 2020).
[3] Y. Altintas and D. Aslan. Integration of virtual and on-line machining process control and monitoring. *CIRP Annals*, 66(1):349 – 352, 2017.
[4] George B Arfken and Hans J Weber. Mathematical methods for physicists, 1999.
[5] Mattia Carletti, Chiara Masiero, Alessandro Beghi, and Gian Antonio Susto. Explainable machine learning in industry 4.0: evaluating feature importance in anomaly detection to enable root cause analysis. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 21–26. IEEE, 2019.
[6] Felix Finkeldey, Amal Saadallah, Petra Wiederkehr, and Katharina Morik. Real-time prediction of process forces in milling operations using synchronized data fusion of simulation and sensor data. *Engineering Applications of Artificial Intelligence*, 94:103753, 2020.
[7] Antoni Grau, Marina Indri, Lucia Lo Bello, and Thilo Sauter. Industrial robotics in factory automation: From the early stage to the internet of things. In *IECON 2017-43rd Annual Conference of the*

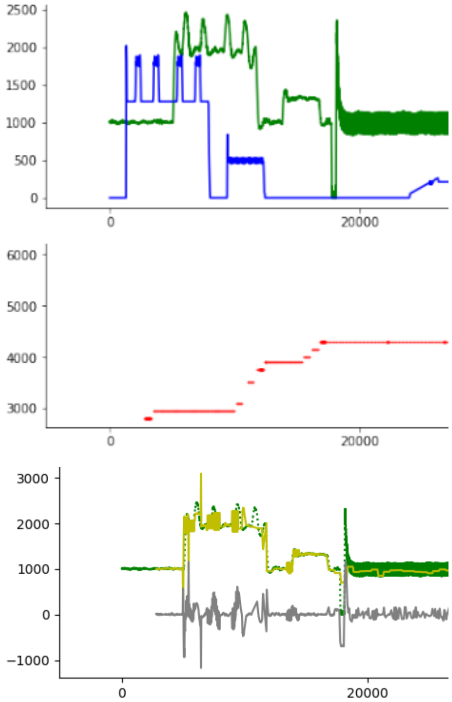

Fig. 6: Part of the power signals for the pocket end milling of the **GPR2** with **aluminium** material and a **short tool**.

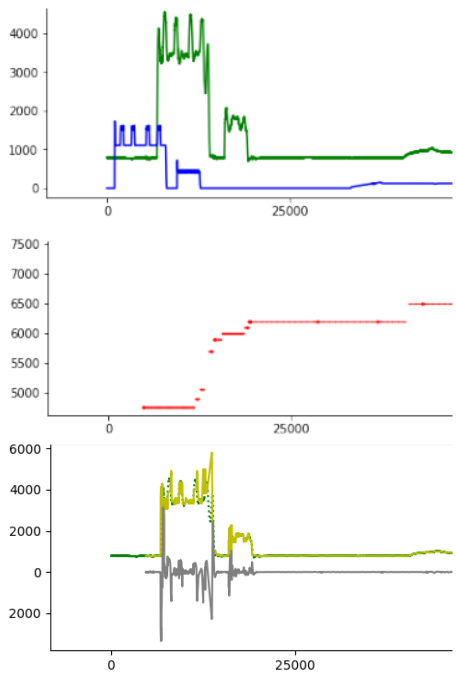

Fig. 7: Part of the power signals for the pocket end milling of the **GPR2** with **steel** material and a **short tool**.

IEEE Industrial Electronics Society, pages 6159–6164. IEEE, 2017.

[8] Nasser Jazdi. Cyber physical systems in the context of industry 4.0. pages 1 – 4, 05 2014.

[9] VVRPV Jyothsna, VV Rama Prasad, and K Munivara Prasad. A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7):26–35, 2011.

[10] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018.

[11] Lex Lennings. Selecting either layered manufacturing or cnc machining to build your prototype. *SME Technical Paper, Rapid Prototyping Association, PE00-171*, pages 1–10, 2000.

[12] Mansfield Merriman. *A List of Writings Relating to the Method of Least Squares: With Historical and Critical Notes*, volume 4. Academy, 1877.

[13] Oussama Meski, Farouk Belkadi, Florent Laroche, and Benoit Furet. Towards a knowledge-based framework for digital chain monitoring within the industry 4.0 paradigm. *Procedia CIRP*, 84:118–123, 2019.

[14] Dimitris Mourtzis, Ekaterini Vlachou, Nikolaos Milas, and George Dimitrakopoulos. Energy consumption estimation for machining processes based on real-time shop floor monitoring via wireless sensor networks. *Procedia CIRP*, 57:637 – 642, 2016.

[15] Fadi Shrouf, Joaquín Ordieres-Meré, and Giovanni Miragliotta. Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm. *IEEE International Conference on Industrial Engineering and Engineering Management*, 2015, 12 2014.

[16] Reddy Sreenivasulu. Optimization of surface roughness and delamination damage of gfrp composite material in end milling using taguchi design method and artificial neural network. *Procedia Engineering*, 64:785–794, 2013.

[17] P Wiederkehr and T Siebrecht. Virtual machining: capabilities and challenges of process simulations in the aerospace industry. *Procedia Manufacturing*, 6(Supplement C):80–87, 2016.

[18] Yu Zhang, Xun Xu, and Yongxian Liu. Numerical control machining simulation: a comprehensive survey. *International Journal of Computer Integrated Manufacturing*, 24(7):593–609, 2011.

APPENDIX

A. Reverse Engineering to Control Parameters

The analysis of the difference between the real system (R_S) and the existing simulation model (S_E), used to manufacture the workpieces exhibits two main drifts[6] between the two systems: temporal alignment and amplitude. We defined two hyper-parameters vectors $V_{\texttt{diff}}$ and $V_{\texttt{change}}$, respectively, for dealing with this issue. Both of them are indexed by the time of the real industrial system and not the simulator's time. $V_{\texttt{diff}}$ concerns the case where the system and the simulator have rigorously the same behavior, but are not temporally synchronized. The time correction is done step by step and for S_{AE}. Moreover, each value added to the simulator to be close to the real system is stored in $V_{\texttt{diff}}$. Most of the time, the difference is also in the amplitude of the signal. In this case, the correction coefficients, used for aligning the simulator according to the Legendre principle, are saved in the vector $V_{\texttt{change}}$ indexed by the time of the real system.

These two vectors are important for the correction of the simulator. But their main importance is in the a posteriori analysis of the gap between the equational model and the real system. Initially, these vectors permit us to calculate the inverse function of the simulator's correction step by step (see Equation 8 below). This can be used for several analyses, for example, to diagnose the faults or, to interpret the possible anomalies which led to the differences between the two systems. But the most important role of these two vectors is to study the regularities of the changing behavior, in order to understand if the difference is not because of some problems caused by the real system performances. For example, due to an incomplete equational modeling process of the real system. As a prospective work, this will be studied using some machine learning approaches in the future.

Let us consider $c(t)$ at time t, indexes all the time changes included in $V_{\texttt{diff}}$. If we denote $\overline{S}_E$ as a cumulative change between S_E and S_{AE}:

$$\forall t \in \{0, \ldots, MT\} \ \ \overline{S}_E(t) = S_E(t) + \Sigma_{i=0}^{i=c(t)} V_{\texttt{change}}(t) \tag{8}$$

[6]Notice that, if the output signal slowly changes independent of the measured property, is defined as drift.