Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 80 (2019) 144–149

www.elsevier.com/locate/procedia

26th CIRP Life Cycle Engineering (LCE) Conference

Multiple Sleeping States for Energy Saving in CNC Machining Centers

Matteo Squeo[a], Nicla Frigerio[a,*], Andrea Matta[a]

[a]Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy

Abstract

Energy efficient control policies that switch off/on the machine have been proposed in recent literature. However, such policies control the whole machine simultaneously and consider only one stand-by state. This paper extends a time-based threshold policy by modeling several sleeping states and controlling machine components. Machine idle time is stochastic and the expected value of the energy consumed per produced part is minimized while assuring a certain productivity target. Technological constraints for control applicability are discussed. Results are based on a 5-axes Computer Numerical Control (CNC) machining center and savings are significant compared with that of state-of-the-art policies.

Peer-review under responsibility of the scientific committee of the 26th CIRP Life Cycle Engineering (LCE) Conference.

Keywords: Machine Tools; Energy Efficiency; Switching Control; Multiple Sleeping States.

1. Introduction

A key towards more sustainable manufacturing industries is applying measures to improve resource energy efficiency. At machine level, improvements can be achieved through a better use of machine auxiliary equipment. According to several preceding works regarding energy assessment in manufacturing environments, e.g., [1] [2], the base load or non-processing energy (NPE), which indicates whenever energy is used in not-operative states, is separated from the processing energy (PE) that is required while the machine tool is working on parts. The selection of proper process parameters aims at optimizing PE. This work is focused on the reduction of NPE that can be achieved through the control of machine states. This control aims to reduce the energy demanded when the machines are idle by start/stop features such that the machine is switched off/on according to certain rules. Other measures can be found in recent and comprehensive overviews, e.g. [2] [3].

1.1. Brief Literature Review

Relevant articles with the objective of reducing the NPE with state control can be classified in two groups: energy-efficient scheduling (EES) and energy-efficient control (EEC). EES plans off/on modes over a specific period of time given the parts assigned to the machine, while EEC focuses on the control level providing policies during production progress, often without knowing when the next part arrives. A recent and complete review on EES literature has been developed by Gahm et al. [4], whereas herewith we provide a brief review of EEC literature.

The machine is usually modeled with several energy consumption states. Beside the *busy state* –i.e. the machine is processing a part– and the *idle*–i.e., the machine is starved or blocked but it can immediately start processing parts when needed, intermediate low energy consumption states–i.e., *standby* or *sleep*– are introduced. The control policies proposed in the literature decide if a transition from the idle state to an energy saving mode is advantageous in terms of energy consumption knowing that the machine might require a compulsory *startup* transitory, in order to restore the service after a sojourn in a *sleeping state*. The startup time is sometimes considered and assumed to be constant (e.g.,[5][6][7][8][9]) or randomly distributed (e.g.,[10][11]). As an exception, Frigerio and Matta [12] analyze a startup that is dependent on the time period the resource stays in standby.

Time-based control policies use arrival information to choose in which instant to switch the machine before next part arrives (e.g.,[5][13][12][7][19]). In buffer-based policies, the energy saving potential comes from the blocking/starvation phenomena: whenever a machine starves (or it is blocked), it could be switched-off for energy saving purposes.

* Corresponding author. Tel.: +39-02-23998559 ; fax: +39-02-23998585.
E-mail address: nicla.frigerio@polimi.it (Nicla Frigerio).

2212-8271
Peer-review under responsibility of the scientific committee of the 26th CIRP Life Cycle Engineering (LCE) Conference.
10.1016/j.procir.2018.12.020

When a certain number of parts accumulates in the buffer (or a certain number of buffer slot empties), the machine can be switched-on. Effectiveness of the control depends on buffer thresholds. Several policies based of buffer thresholds have been proposed in the literature (e.g., for production systems: [14][8][9]; for single machines: [10][11]). A combination of time and buffer information is used in a third group of studies (e.g., [15][16])

1.2. Contribution

According to the literature, only one standby state is considered. As exceptions, multiple sleeping states are defined in [17] and [6]. In Mashaei *et al.* [17], machines are assumed to have two energy saving states: *Hot Idle* and *Cold Idle* modes. However, only the *Cold Idle* mode requires a transitory; therefore, the problem degenerates into a single-sleeping state problem. Moreover, the paper deals with EES problems. Li and Sun [6] assume that machines might have several hibernation (H_q) states and a transitory is required to enter in a H_q state and to resume the service after a sojourn in H_q state. When a machine is starving or blocked, the control might trigger the machine in an hibernation state. Blocking and starvation periods are estimated to decide which among the H_q states is optimal. Whenever the input buffer is not empty and the downstream buffer is not full, the machine immediately enters in transitory from one hibernation state. Therefore only the switch-off action is optimized identifying a single optimal state when the machine is not operational. However, since a transitory is required, a different policy might perform better. For example, a policy that allows to accumulate more than one part, or a policy that allows to pass from one sleeping state to another sequentially.

In this work, we propose a *Multi-Sleep* (MS) control policy that uses time-based thresholds $\tau_{\text{off},i}$ and $\tau_{\text{on},i}$ to control each machine component i:

> *Switch-off component i when $\tau_{off,i}$ has elapsed from the last departure. Switch-on component i when $\tau_{on,i} > \tau_{off,i}$ has elapsed from the last departure or when next part arrives.*

MS policy can be seen as an extension of the *Switching* (SP) policy [7] that controls the machine with two time-thresholds τ_{off} and τ_{on}. Therefore, the contribution with respect to the literature is twofold. First, the technological feasibility of multi-state control has been investigated for CNC machining centers. A state model of a 5-axis machining center for automotive purposes is created by explicitly modeling component states, e.g. the chiller units, the hydraulic unit, the chip conveyor, etc. Beside technological precedence constraints, each component can be individually controlled creating several sleeping states from the machine point of view. As a second contribution, the proposed MS policy chooses which component is advantageous to switch. The startup transitory required to resume the readiness is different among components such that the selection of which components to switch is not trivial. The new model considers several sleeping modes for the machine and the policy enables an optimal sequence of transitions between saving modes. Discrete event simulation is used to evaluate machine performance in terms of energy consumption and productivity. The optimal solution is obtained and discussed for a set of numerical cases.

2. Multi-Sleep Model

The recent standard ISO 14955-1:2014 defines *machine tool component as mechanical, electrical, hydraulic, or pneumatic device of a machine tool, or a combination thereof.* A machine tool is composed by assemblies of components ensuring a specific function. A methodology to build energy state-based models of complex machine tools with a bottom-up approach and the automata formalism has been proposed in [18]. The starting point is the machine decomposition in components. Then, by synchronizing individual component models, it is possible to create a unique model for the machine. Instead of analyzing the whole machine, that might be complex, the approach enables machine modelling at component level with minor effort.

2.1. Components Modelling

Machine tool components can be classified upon their power behaviour over time as *constant*, *periodic*, or *intermittent* (Figure 1). The states *Off*, *OnService (OnS)*, and *Run* respectively represent the component not connected to the grid, connected to the grid but not working, and connected and executing its function. Each component k has a power request $P_{k,s}$ when in state s. Trivially, $P_{k,\text{off}} = 0$. The *Run* state has a significant power request such that $P_{k,run} > P_{k,ons} \geq 0$. Mostly, the *OnS* request is almost negligible because the component is not executing any function. Furthermore, components might require a certain *Startup* transitory before entering into the *OnS/Run* state. Specific procedures are executed during component startup and the power request is significant. Also, $P_{k,su} > P_{k,ons}$. Transition from *OnS/Run* to *Off* happens almost instantaneously for all analyzed components. The events *MainOn* and *MainOff* are common among components and represent the connection of the machine to the grid by manually moving the machine main switch.

Constant components are always in *Run* state after the startup ends. The control cabinet belongs to this class.

Intermittent components are required accordingly to the part program executed by the machine. Events *Function Request* and *Request End* represent the need of component's functioning and its completion. Spindle and servo drivers are intermittent.

The *periodic* behaviour is typical of components depending on machine condition of pressure or temperature. As an example, the hydraulic unit has a periodic power consumption due to the dependence on the oil pressure level. It runs when the oil pressure drops below a certain limit p_ℓ and stops when the required pressure p_u is reached. The periodic visit of *OnS* and *Run* states can be simplified in an *Active* state. Therefore, the power $P_{k,a}$ request while *Active* is calculated as an average con-

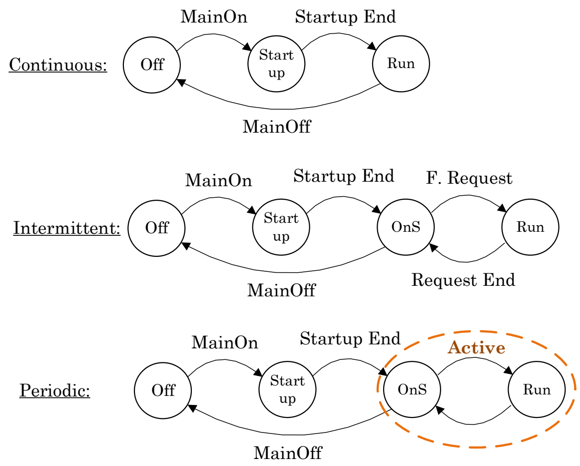

Figure 1. State models representing the behavior of not controlled components.

sidering the time $t_{k,s}$ spent in *OnS* and *Run* states:

$$P_{k,a} = \frac{P_{k,run} \cdot t_{k,run} + P_{k,ons} \cdot t_{k,ons}}{t_{k,ons} + t_{k,run}}. \quad (1)$$

Since during the startup, the component is always running, for periodic components we assume $P_{k,su} > P_{k,a}$.

Moreover, power request might depend on some parametes $\mathbf{x}$ such that $P_{k,s}(\mathbf{x})$. For example, the spindle starts accelerate when the G-Code instruction requires a certain speed and the power *P*spindle, $run(\cdot)$ depends on process parameters. The hydraulic unit require $P_{HS,run}(\cdot)$ to reach the appropriate pressure p_u. During the startup, the hydraulic unit is also reaching the required p_u consuming $P_{HS,su}(\cdot) \approx P_{HS,run}(\cdot)$, but the sojourn in startup is longer that the period commonly spent in *Run* because the starting pressure level can be below p_ℓ.

Herewith, we consider a CNC-machining center with 5-axis and the following list of components: 1) *continuous*: control cabinet (Computer&PLC and fans), display and lights, air compressor, low pressure pump for cutting coolant fluid, chip conveyor, exhaust air system; 2) *intermittent*: spindle, servo drives (e.g. X, Y, Z axis linear motors, B-axis and tilting motors on the table, tool changer motor), high pressure coolant pump, and coolant filter pump; 3) *periodic*: hydraulic pump, two cooling circuits for axes and spindle (each composed by a pump, a compressor and a fan).

2.2. Component Control

Controllability comes from the addition of an auxiliary state *Inactive* that has a negligible power request. Two controls trigger the transitions: *Activate* (or switch on) and *Deactivate* (or switch off). The *Main On* and *MainOff* events trigger each component in the *Inactive* state instead of into the *OnS*. The hit of the *OnS* state relies on the *Activate* control action, as well as the way back to *Inactive* state on the *Deactivate* command. Transition from *OnS* to *Inactive* happens instantaneously, whilst the *Activate* event starts the *Startup* when necessary. Notice that, without any control policy, all components are activated simultaneously with the *MainOn*; therefore, all components execute simultaneously their startup. With individual controls, startups might start at different points in time.

Machine State	Hydraulic Unit	Axis Cooling Unit	Spindle Cooling Unit	Chip Conveyor	...
1	Off	Off	Off	Off	...
2	Inactive	Inactive	Inactive	Inactive	...
3	Transitory	Inactive	Inactive	Inactive	...
4	OnService	Inactive	Inactive	Inactive	...
5	OnService	Transitory	Inactive	Inactive	...
6	Run	Transitory	Inactive	Inactive	...
...	...	...	...	...	...

Table 1. Portion of machine state matrix: each machine state corresponds to a certain vector (row) of component states.

Considering the technological and functional relationships among components, some of them have to be activated (deactivated) consistently with others: simultaneously, or one after the other. For example, the spindle drive cannot be activated before its cooling unit to prevent an excessive heat storage. The components of a cooling unit must be activated simultaneously to allow a proper functioning. Therefore, precedence constraints need to be defined among *Activate/Deactivate* commands.

2.3. Energy-State Model of a Controlled Machine

The machine model is built by synchronizing the models of all components. Precedence should be included in an automaton representing machine supervisor. This approach is in general creating a state-model of a machine tool. The machine state can be represented by a vector containing the state of each component. A portion of machine state is reported in Table 1.

In the following, we simplify the model without having an impact on the objective function and we do not include the *Main On/Off* commands because they are common among components and with strict precedence for all other transitions:

A. Intermittent components. *Intermittent* components are only required while the part program is executed and do not affect NPE. Therefore, spindle and servo drives, and high pressure coolant pump are not included in this model.

B. Simultaneous control. Continuous and periodical components do not have any sequential precedence (i.e., one after the other). However, some of them must be activated simultaneously. Therefore, we can group and control them with the same commands to reduce problem dimension. This is the case of cooling units where pump, compressor and fan must be controlled with the same $\tau_{\text{off},i}$ and $\tau_{\text{on},i}$.

C. Negligible startup. At optimality, components with negligible startup duration should be controlled simultaneously. Therefore, problem dimension can be further reduced.

For the analyzed machine, the control problem is reduced to the control of four groups of components: the hydraulic unit $i = 1$, the spindle cooling unit $i = 2$, the axis cooling unit $i = 3$, and a group $i = 4$ of components with negligible startup. Group 4 is composed by the following components: low pressure coolant pump, air compressor, display, lights, chip conveyor, exhaust air system and coolant filter pump. The power request associated with control group i is the sum of requests by component belonging to group i and they are denoted with $P_a^{(i)}$ and $P_{su}^{(i)}$. In this way, machine power is equal to the sum of group powers. In the analyzed case, we obtain 34 machine states where machine sleeping states come from separated control of each group.

3. Optimization Problem

Let us define a *cycle* as the time interval starting from the departure of a part and the departure of the next one. Machine idle time T_w is a random variable distributed accordingly to a certain density function $f_w(\cdot)$. The control parameters $\tau_{\text{off},i}$ and $\tau_{\text{on},i}$ represent the deactivation and activation instants of group i in a certain cycle. The control policy also states that when $\tau_{\text{off},i} \to \infty$ the component is not controlled, and when $\tau_{\text{on},i} \to \infty$ the component is activated upon the arrival. The optimization problem is formulated in equations (2)-(5) where the vector of decision variables $\boldsymbol{\tau} = \{\tau_{\text{off},i}, \tau_{\text{on},i}\}|i = 1, 2, 3, 4$:

$$\min_{\boldsymbol{\tau}} \mathbb{E}[E_{cy}(\boldsymbol{\tau})] = \mathbb{E}[E_{NPE}(\boldsymbol{\tau})] + \mathbb{E}[E_h(\boldsymbol{\tau})] \quad (2)$$

$$\text{Subject to:} \quad \mathbb{E}[TH(\boldsymbol{\tau})] \geq (1 - \varepsilon)\mathbb{E}[TH_\infty] \quad (3)$$

$$\tau_{\text{on},i} > \tau_{\text{off},i} \quad \forall i \quad (4)$$

$$\tau_{\text{off},i}, \tau_{\text{on},i} \in \mathbb{R}_0^+ \quad \forall i \quad (5)$$

Problem objective in equation (2) is the minimization of the expected value of the energy E_{cy} consumed by the machine in a cycle. The energy consumed during part process (PE) is not dependent on the control strategy and it is not included in the objective function. $E_{NPE}(\boldsymbol{\tau})$ is the energy consumed waiting for the new part: $E_{NPE}(\boldsymbol{\tau}) = \sum_i E_a^{(i)}(\boldsymbol{\tau}) + \sum_i E_{su}^{(i)}(\boldsymbol{\tau})$. Energy $\mathbb{E}[E_a^{(i)}(\tau)]$ spent while group i is active is computed as $P_a^{(i)} \cdot \mathbb{E}[T_a^{(i)}(\tau)]$, where $T_a^{(i)}(\tau)$ is the expected time period while group i is in active state. Time $T_a^{(i)}(\tau)$ is stochastic since depending on T_w. Similarly for $\mathbb{E}[E_{su}^{(i)}]$ computed as $P_{su}^{(i)} \cdot \mathbb{E}[T_{su}^{(i)} \cdot I_{T_w > \tau_{\text{off},i}}]$, where $T_{su}^{(i)}$ is the startup duration of group i and $I_{T_w > \tau_{\text{off},i}}$ is an indicator function that marks whether the transitory has been visited within the cycle. $T_{su}^{(i)}$ can be stochastic. Energy $\mathbb{E}[E_h]$ expresses a penalty for holding the part in front of the machine while some components have to complete the startup. It can be calculated as the product between a penalty (P_h) and the expected holding time $\mathbb{E}[T_h(\tau)]$ which is stochastic because relying on T_w.

Equation 3 represents a minimum throughput target that can be seen as a maximum reduction ε of the expected throughput $\mathbb{E}[TH_\infty]$ obtained without any control policy. It can be expressed also as:

$$\frac{t_p + t_w}{t_p + t_w + t_h} \geq 1 - \varepsilon \quad (6)$$

where $t_p = \mathbb{E}[T_p]$ is the mean processing time, $t_w = \mathbb{E}[T_w]$ is the expected waiting time and $t_h = \mathbb{E}[T_h]$ is the expected holding time. Constraints (4) represent the control feasibility between control parameters of the same component (i.e., each *switch-on* must happen after the *switch-off*). Constraints (5) define the domain of decision variables.

The resulting *Multi-Sleep* (MS) control might allow transitions from a sleeping state to another because of the sequential activation/deactivation commands. Also, since components of group 4 do not require any significant startup, the following property holds trivially:

Property. The optimal control parameters are $\tau^*_{off,4} = 0$ *and* $\tau^*_{on,4} = \infty$.

Therefore, problem dimension is reduced.

4. Numerical Analysis of the Multi-Sleep Policy

The power requests of component groups while executing the startup procedure are: $P_{su}^{(1)} = 1.75\ kW$, $P_{su}^{(2)} = 2.5\ kW$, and $P_{su}^{(3)} = 2.25\ kW$. Similarly, while active groups require on the average: $P_a^{(1)} = 0.875\ kW$, $P_a^{(2)} = 1\ kW$, $P_a^{(3)} = 0.9\ kW$, and $P_a^{(4)} = 0.8\ kW$. The startup times are deterministic: $t_{su}^{(1)} = 10\ s$, $t_{su}^{(2)} = 30\ s$ and $t_{su}^{(3)} = 20\ s$. Also, the processing time is $t_p = 180\ s$. Machine parameters and the penalty for holding parts ($P_h = 1\ kW$) do not change in the experimental plan. In order to analyze the machine subject to different conditions, we consider different distributions of waiting time. In the experiments, we consider two Weibull distributions with $k = 0.5$ and $k = 3$ to respectively represent unimodal distribution $f_w(\cdot)$ with Decreasing and Increasing Hazard Rate. Also, we vary the mean of waiting times $t_w \in [5, 120]\ s$. Therefore, we model situations where the machine is utilized from 60% to 97%.

Results of the MS policy are compared with that of (1) a not-controlled machine–i.e., *Always On* (AO)– and (2) the single-sleep SP policy. Section 4.1 analyzes the problem numerically. The expected energy per part $\mathbb{E}[E_{cy}(\tau)]$ and the expected throughput $\mathbb{E}[TH(\tau)]$ are obtained by simulation considering a certain number of observations $n = 6000$ to guarantee the half-width of the interval of confidence IC95% with 10 replications at most equal to 1% of the mean. Optimization results obtained are in Section 4.2. A gradient-based stochastic solver embedded in *Matlab* is used, i.e., *Global Search*. Although the optimization problem is solved by relaxing the throughput constraint (3), the throughput reduction is evaluated.

4.1. Properties of the Function and Solver Performance

The shape of the objective function has been analyzed numerically by varying the controls $\boldsymbol{\tau}$. The existence of a global minimum can be spotted from numerical results, as well as the energy convergence to the AO solution as $\tau_{\text{off},i}|\forall i$ increase. Also, the dependence between decision variables is investigated numerically. The expected energy required in a cycle $\mathbb{E}[E_{cy}]$ is represented in Figure 2 for a single scenario ($k = 0.5$ and $t_w = 120\ s$). It can be noticed that there exist a dependency between decision variables $\tau_{\text{off},1}$ and $\tau_{\text{off},3}$. The same conclusions can be drawn with similar analyses onto other decision variables. The reason of such dependency is that although group i is ready, it must wait for group j ($j \neq i$) whenever group j has to complete its own startup. This situation happens, for example, when $\tau_{\text{off},i} > \tau_{\text{off},j}$ and the part arrives at t before group i is switched off (i.e., $\tau_{\text{off},i} > t > \tau_{\text{off},j}$).

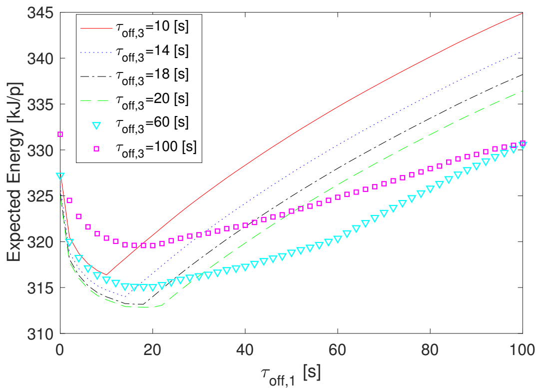

Figure 2. Expected energy per part varying $\tau_{\text{off},1}$ and $\tau_{\text{off},3}$ (k=0.5, t_w = 120 s).

Weibull(k, t_w)	$Weib(3, 120)$	$Weib(0.5, 120)$
$\tau_{\text{off},1}$ [s]	0.022 ± 0.02	12.4 ± 1.5
$\tau_{\text{off},2}$ [s]	0.019 ± 0.02	19.4 ± 0.6
$\tau_{\text{off},3}$ [s]	0.021 ± 0.02	18.6 ± 0.8
$\tau_{\text{on},1}$ [s]	157.4 ± 0.3	1946 ± 31
$\tau_{\text{on},2}$ [s]	137.4 ± 0.3	1907 ± 75
$\tau_{\text{on},3}$ [s]	147.4 ± 0.3	1897 ± 90
$\mathbb{E}[E_{cy}]$ [kJ/p]	207.2 ± 0.04	160.1 ± 0.24
$\mathbb{E}[TH]$ [p/s]	7.3 ± 0.01	5.5 ± 0.08

Table 2. Confidence interval of sample-path solution found by GS in two scenarios (5 replications, n = 6000).

In order to show the performance of the solver, a single scenario has been optimized 5 times and results in Table 2 confirm that GS provides very narrow ranges on the optimal solution $\boldsymbol{\tau}^*$ when dealing with a Weibull with $k = 3$ ($t_w = 120$ s). Whereas, the results obtained from a Weibull with $k = 0.5$ ($t_w = 120$ s) shows a wider range due to a higher variance. Nevertheless, the confidence interval on the optimal objective function achieved is narrow, showing that the objective function is flat in this case.

4.2. Optimization Results

In this section, GS is used to find the sample-path solution (n = 6000) by varying t_w. For the search, the solution space has been limited up to the 98-th percentile of the waiting time distribution with mean $t_w = 120/s$: [0, 870] s for $k = 0.5$ and [0, 320] s for $k = 3$. Therefore, it is implicitly assumed that if $\tau_{\text{off,i}}$ is equal to the upper bound, group i is never switched off. Similarly for $\tau_{\text{on,i}}$, group i is switched on at part arrival.

Results in Figure 3 and Figure 4 show that controlling the machine with the MS policy is advantageous compared to the SP policy and to the uncontrolled machine (AO). For low values of mean waiting time t_w, components with long startup are not controlled. As t_w increases, all components are controlled. The maximum throughput reduction is $\varepsilon = 0.093$ and the MS performs better than SP for several scenarios. By constraining the reduction, similar results can be obtained.

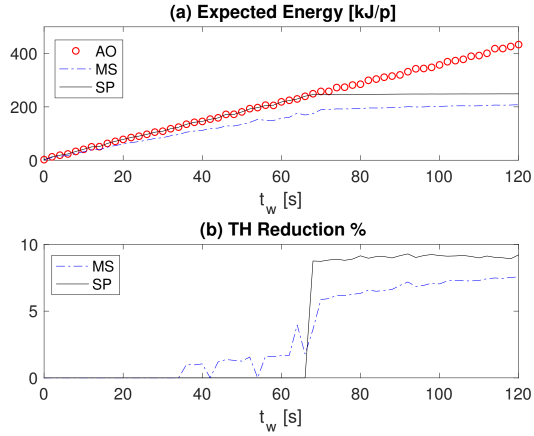

Figure 3. Optimization result comparison for (Weibull k = 3): expected energy per part (a) and throughput reduction in % (b).

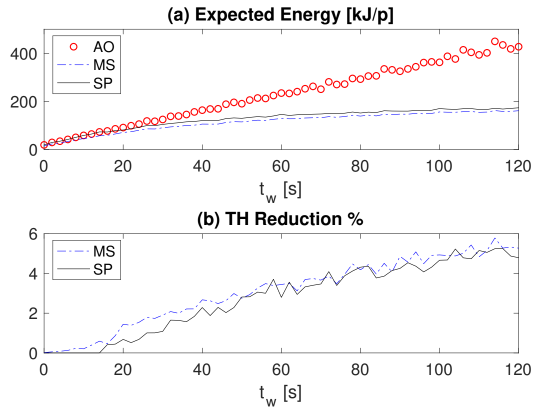

Figure 4. Optimization result comparison for (Weibull k = 0.5): expected energy per part (a) and throughput reduction in % (b).

Scenarios with $k = 3$ show it is always optimal to immediately switch off all components ($\tau^*_{\text{off},i} = 0$—$\forall i$). Although components with a long startup are kept active. An example is in Figure 5. The hydraulic unit (i = 1) and the axes cooling unit (i = 3) are switched off; although, it is advantageous to keep active the spindle cooling unit i = 3. The switch-on commands are launched in advance to prepare components for the arrival of parts and they occur with different $\tau_{\text{on},i}$, implying a progressive startup procedure where the passage among sleeping states comes from the different switch-on instants of components. As t_w increases, the optimal $\tau^*_{\text{on},i}$ increases. The effect of t_w onto optimal controls is similarly to that of the SP as in [7].

Results obtained with $k = 0.5$ show it is always optimal to switch on components when the part arrives ($\tau^*_{\text{on},i} \rightarrow \infty | \forall i$ and to wait before switching off. An example is in Figure 6. The switch off commands occur with different $\tau_{\text{off},i}$, implying a progressive passage from a higher power consumption sleeping state to lower ones. Also, as t_w increase, the optimal $\tau^*_{\text{off},i}$ decreases. As t_w increases, it might happen that a component is kept active.

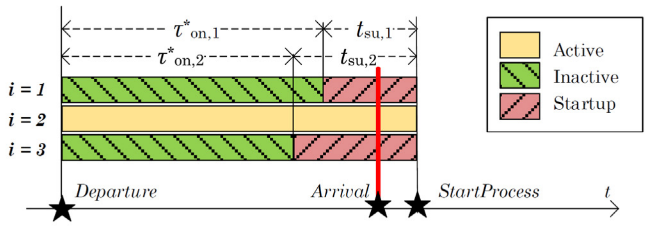

Figure 5. Optimization results for Weibull $t_w = 78\ s, k = 3$. Optimal control parameters $\tau^*_{\text{off},i} = 0|\forall i$ and $\tau^*_{\text{on},i}$ vary: $\tau^*_{\text{on},1} = 79.1s$, $\tau^*_{\text{on},2} \rightarrow \infty$ and $\tau^*_{\text{on},3} = 69.3\ s$. In this case, the arrival occurs while $i = 1, 2$ are executing the startup.

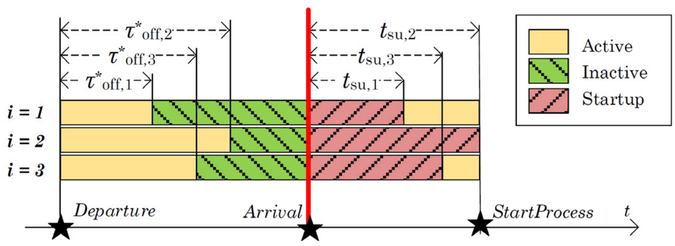

Figure 6. Optimization results for Weibull $t_w = 49, k = 0.5$. Optimal control parameters $\tau^*_{\text{on},i} \rightarrow \infty|\forall i$ and $\tau^*_{\text{off},i}$ vary: $\tau^*_{\text{off},1} = 30.5s$, $\tau^*_{\text{off},2} = 82.6s$, $\tau^*_{\text{off},3} = 55.1s$. In this case, the arrival occurs when all components are inactive and triggers the startups.

4.3. Computational Times

The computational time required by the solver is significant. Results have been obtained on a laptop with i5 Intel Core @1.3GHz and RAM-4GB. Distribution $f_w(\cdot)$ significantly affects the computational time required by GS which are longer for $k = 0.5$. Also, when the objective function is flat, it is more difficult to find the optimum. In the analyzed cases, GS requires on the average for each optimization: 2.6 minutes (up to 3.2) for $k = 3$ and 20 minutes (up to 34) for $k = 0.5$.

5. Conclusions

Results show that including multiple sleeping states in EEC can improve the performance of switching policies thanks to the selection of which component to switch and avoiding long startups. Future effort will be devoted to analyze the structural properties of the problem and to design more efficient optimization algorithms. An extended sensitivity analysis on other significant parameters, e.g., startup duration and penalty, will be performed in future works. Although the MS policy performs efficiently for unimodal distributions, a different control policy should be formulated in order to efficiently address not-unimodal distributions as objective of future studies. Also, the effect of problem parameters will be performed, e.g., startup duration and penalty. A critical barrier for implementation is the knowledge of the waiting time distribution; thus, learning methods should be included in order to increase applicability for practitioners.

ACKNOWLEDGEMENTS

The research has been partially funded by Regione Emilia under the project: Hi-score.it - Progetto POR-FESR 2014-2020, ASSE 1 Ricerca e Innovazione - Azione 1.2.2.

References

[1] Dahmus, J.B., Gutowski, T.G., 2004. An environmental analysis of machining, in: ASME Int. Mech. Eng. Congress and Exposition, 643-652.
[2] Yoon, H.S., Kim, E.S., Kim, M.S., Lee, J.Y., Lee, G.B. and Ahn, S.H., 2015. Towards greener machine tools a review on energy saving strategies and technologies. Renewable and Sustainable Energy Reviews, 48, 870-891.
[3] Zhou, L., Li, J., Li, F., Meng, Q., Li, J., and Xu, X., 2016. Energy consumption model and energy efficiency of machine tools: a comprehensive literature review, J. of Cleaner Prod., 112(5), 3721-3734.
[4] Gahm, C. Denz, F., Dirr, M. and Tuma, A., 2016. Energy-efficient scheduling in manufacturing companies: a review and research framework. European J. of Operational Research, 248(3), 744-757.
[5] Mouzon, G., Yildirim,M.B. and Twomey, J., 2007. Operational methods for minimization of energy consumption of manufacturing equipment. Int. J. of Prod. Research (IJPR), 45(18-19), 4247-4271.
[6] Li, L., and Sun, Z., 2013. Dynamic energy control for energy efficiency improvement of sustainable manufacturing systems using Markov decision process. IEEE Trans. on Systems, Man, and Cybernetics, 43(5), 11951205.
[7] Frigerio, N. and Matta, A. (2015). Energy-efficient control strategies for machine tools with stochastic arrivals. IEEE Trans. on Automation Sc. and Eng., 12(1), 50-61.
[8] Jia, Z., Zhang, L., Arinez, J. and Xiao, G., 2016. Performance analysis for serial production lines with Bernoulli machines and real-time WIP-based machine switch-on/off control. IJPR, 54(21), 6285-6301.
[9] Wang, J., Feng, Y., Fei, Z., Li, S., and Chang, Q., 2017. Markov chain based idle status control of stochastic machines for energy saving operation, Proceedings of IEEE Int. Conf. on Automation Sc. and Eng. (CASE), 10191023, Xian, P.R. China.
[10] Maccio, V.J. and Down, D.G., 2015. On optimal policies for energy-aware servers. Performance Evaluation, 90, 36-52.
[11] Frigerio, N. and Matta, A., 2016. Analysis on energy efficient switching of machine tool with stochastic arrivals and buffer information. Trans. on Automation Sc. and Eng., 13(1), 238-246.
[12] Frigerio, N. and Matta, A., 2014. Energy efficient control strategy for machine tools with stochastic arrivals and time dependent warm-up. Procedia CIRP, 15, 56-61.
[13] Prabhu, V.V., Jeon, H.W. and Taisch, M., 2012. Modeling green factory physics - an analytical approach. Proc. of IEEE Int. CASE, 46-51, Seoul, Korea.
[14] Brundage, M.P., Chang, Q., Li, Y., Xiao, G. and Arinez, J., 2014. Energy efficiency management of an integrated serial production line and HVAC system, IEEE TASE., 11(3), 789-797.
[15] Chen, G., Zhang, L., Arinez, J. and Biller, S. (2011). Feedback control of machine startup for energy-efficient manufacturing in Bernoulli serial lines. Proc. of IEEE Int. CASE, 666-671, Trieste, Italy.
[16] Guo, X., Zhou, S., Niu, Z., Kumar, P., 2013. Optimal wake-up mechanism for single base station with sleep mode, IEEE Int. Teletraffic Congress, 18.
[17] Mashaei, M., and Lennartson, B., 2013. Energy reduction in a pallet-constrained flow shop through onoff control of idle machines. Trans. on Automation Sc. and Eng., 10(1), 4556.
[18] Frigerio, N., Matta, A., Ferrero, L., and Rusina, F., 2013. Modeling Energy States in Machine Tools: An Automata Based Approach. Proc. of CIRP Conf. on Life Cycle Eng., Singapore, 203-208.
[19] Li, W., Zein, A., Kara, S., Herrmann, C., 2011. An Investigation into Fixed Energy Consumption of Machine Tools. 18th CIRP LCE Conference, Braunschweig,268-273.