Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 11 (2017) 319 – 328

Procedia MANUFACTURING

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

A Simulation Tool for Computing Energy Optimal Motion Parameters of Industrial Robots

Michele Gadaleta[a]*, Giovanni Berselli[b], Marcello Pellicciari[a], Mario Sposato[a]

[a] *"Enzo Ferrari" Department of Engineering, University of Modena and Reggio Emilia, Via P. Vivarelli, 10, 41125 Modena, Italy*
[b] *Department of Mechanics, Energetics, Management and Transportation,University of Genova, Via All'Opera Pia, 15/A, 16145 Genova, Italy*

Abstract

This paper presents a novel robot simulation tool, fully interfaced with a common Robot Offline Programming software (i.e. *Delmia Robotics*), which allows to automatically compute energy-optimal motion parameters, for a given end-effector path, by tuning the joint speed/acceleration during point-to-point motions whenever allowed by the manufacturing constraints. The main advantage of this method, as compared to other optimization routines that are not conceived for a seamless integration with commercial industrial manipulators, is that the computed parameters are the same required by the robot controls, so that the results can generate ready-to-use energy-optimal robot code.

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing

Keywords: Computer-Aided Robotic tools, Industrial Robots, Energy Optimization, Delmia Robotics, Sustainable Manufacturing

1. Introduction

Sustainable manufacturing may be defined as a production model where both present and future needs are contemporarily accounted for, implying that social, ecological and economic impacts should be quantitatively

* Corresponding author. Tel.: (+39) 059-2056278;
E-mail address: michele.gadaleta@unimore.it

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing
doi:10.1016/j.promfg.2017.07.114

assessed and optimized [1-3]. From an industrial point of view, and focusing on the latter two aspects, manufacturing companies shall aim at reducing the consumption of resources and the global expenses, while possibly improving the overall process production rate. Naturally, in the latest decades, the requirements for low costs, high accuracy and high flexibility led to a huge adoption of automated production lines and Industrial Robots (IR), which, unfortunately, are rather energy intensive. Present researches dealing with energy efficiency improvement in this engineering field, basically follow two different approaches:

- In a medium-long term perspective, the development of innovative technologies that would ensure a reduced energy consumption (EC) as compared to the actuals ones. This approach includes the introduction of renewable/green energy sources [4,5] or energy-efficient equipment (such as low consumption electric motors and drives [6]), and the improvement of the plant architecture so as to minimize the rejection rate of products which may not comply with quality standards [7]. Such strategies have been classified as *eco-efficient design methods* [8], since they all require substantial modifications of existing plants;
- In a short term perspective, the development of innovative methods and tools for an efficient use of the actual technologies, offering the possibility to reduce the EC also on existing plants, which are liable of small possibilities for adjustments/modifications. This approach includes the optimization of the overall production planning and/or operations scheduling [9-11] and the energy-optimal IR control [12-14]. Such strategies have been classified as *eco-efficient programming methods* [8], since they do not require the introduction of new components.

In the following, we focus on the latter approach and, in particular, on IR smart programming. The first consideration to be underlined is that many factories still do not employ the latest generation of energy-optimized and (possibly) freely programmable robot arms. Most small/medium enterprises are surely willing to employ/re-use their in-house IR until the end of their overall lifecycle. The second consideration, which has been clearly underlined in the part literature [13], is that these IR may consume on the order of hundreds of kWh per day, so that a very slight EC reduction may lead to a very substantial cost decrease in the long term.

For what concerns IR programming tools and engineering practices commonly employed by IR programmers, the following aspects should be taken into account:

- In most cases, IR motions are programmed offline via dedicated robot simulation packages, such as *Delmia Robotics* (from *Dassault Systèmes* [15]), *RobotStudio* (from *ABB* [16]) or *KUKA.Sim* (from *KUKA AG* [17]), which can simulate the IR movement in a virtual plant and, subsequently, provide (as a direct output) the robot code that can be readily uploaded on the physical system. Some of these software are vendor-specific tools (i.e. they can simulate only IR produced and sold by one vendor), whereas others are conceived as general purpose simulators. In the latter case, these tools actually provide a rough approximation of the real IR end-effector trajectory. On the other hand, the simulations can be highly improved by employing a vendor-specific plug-in (the so called RCS module [19]), which basically acts as a black-box model that computes the IR trajectories with the same proprietary algorithms employed in the physical controller. In addition, (although the trend is rapidly changing) several robot simulation tools are mainly kinematic in nature, so that the dynamic IR behaviour and any possible knowledge about EC remains hidden to the end-user.
- Commonly, IR programmers are not totally free to impose a complex end-effector trajectory, the IR motions mostly employed in practice being Point-to-Point (PtP), Linear and Circular movements. For what concerns PtP commands, the robot simulation package computes an end-effector path that cannot be adjusted in any other way rather than dividing the path itself into series of small PtP motions.
- For what concerns the IR motion laws, the only tool available to the programmer is a control on the percentage of maximum velocity and acceleration imposed to the end-effector. Since any information about the IR energy consumption may not be available, also skilled operators usually program their robots to follow a path in the quickest possible way, despite the fact that such execution speed may be actually unnecessary. For instance, it has been shown that, in a robotic cell composed of several robots, many IR spend most of their time in standstill mode of operation, thus wasting electrical energy that is used to provide the motor torque that counterbalance the IR own weight [19]. This strategy is surely not energy-optimal nor required to actually increase the plant production rate. The same consideration also applies to a single IR performing a series of PtP motions. An energy aware robot programming would employ the correct value of velocity and acceleration parameters that allow to perform each operation within the right amount of time (e.g. reducing standstills to a minimum).

As for related literature dealing with optimal IR motions, several theoretical studies have been presented in the past

(see e.g. [20-22]). In particular, a practical method to determine the energy-optimal scaling parameter (termed override ratio) has been proposed by the authors in [19] on the basis of a simplified IR dynamic model. Outcome of this approach is the so-called IR *Energy Signature*, namely a function describing the robot EC at varying override ratio. The IR energy signature shows that, for a given and fixed geometrical end-effector path, it is possible to quickly compute an energy optimal scaling parameter. This paper build upon this previous results by providing three critical advancements:

- An accurate IR system model, which accounts for all the chain of components from the IR end-effector to the electrical energy source;
- The possibility to vary both velocity and acceleration parameters;
- An optimization method which is interfaced with a commercial robot simulation tool, along with the RCS module, thus allowing to effectively generate a robot code that can be practically employed in the physical system.

In summary, the difference of the method, as compared to other approaches for determining IR optimal motions (e.g. [19]), lays primarily in the fact that the optimization results can be seamlessly applied also on those IR (still largely employed in the factories) with a closed controller which allows a limited parameters setting.

The rest of this paper is organized as follows: Section 2 presents an overall IR energy-model, built on the basis of past literature [19, 23, 24]; Section 3 describes how the EC is computed, leveraging on the restricted capabilities of the robot simulation tool *Delmia Robotics*; Section 4 reports about the optimization method and provides a numerical example; Section 6 reports the concluding remarks and discusses about future directions of improvement.

2. Overall IR system modelling

Industrial robots are mechatronic machines constituted by mechanic, electric, and electronic components, all determining the total power consumption. For an accurate estimation of the total energy flow, a complete mechatronic model of the IR system is essential, modelling all influential peripheral devices. As depicted in Fig. 1 and 2, components which are taken into account are *i)* manipulator mechanical structure (including the weight balancer), *ii)* Permament Magnet Synchronous Motor (PMSM), along with their gearboxes and electromechanical brakes, drive systems devices (rectifier, inverters, DC-bus and electrical brake resistor), *iii)* any constant loss of auxiliary devices. The model described hereafter is accurate enough for retrieving an accurate EC prediction but also general enough to be used for most robots (i.e.: with small or large payloads). The employed IR model computes the EC starting from the robot joints trajectories (computed within a robot simulation tool after the imposition of an end-effector path) and a series of data defining the mechatronic system. In the following, the dynamic model of each mechatronic component is described following back the system power flow.

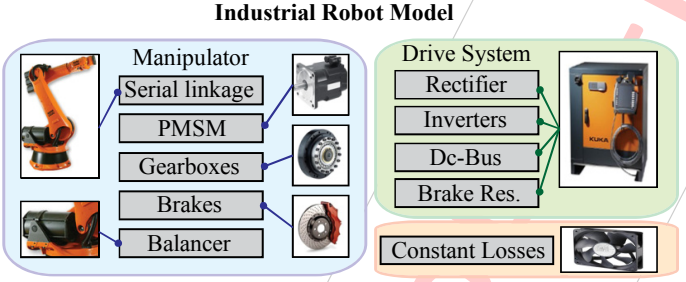

Fig. 1. Components contributing to IR energy consumption

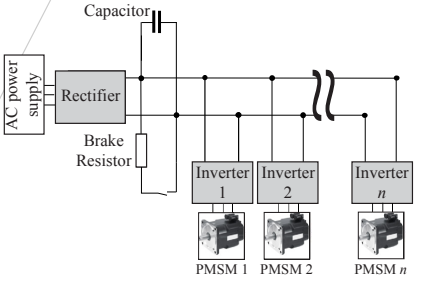

Fig. 2. Schematic of the drive system components

2.1. Mechanical components

The procedure for the EC computation starts within a robot simulation tool, where the user specifies an end-effector geometric path and imposes certain values for the velocity and acceleration parameters. As a simple example, let us consider an IR performing a linear movement. In this case, Fig. 3 reports an example of end-effector

speed for different values of velocity and acceleration limits: the black curve depicts the original trajectory; whereas the blue and the red curves respectively show a case where either velocity limit only or acceleration limit only are reduced. Given the end-effector path and its velocity/acceleration limits, the robot simulation tool computes the IR joint trajectories by inverse kinematics. As an example, Fig. 4 reports the angular velocity of the first robot joint when executing the linear motions depicted in Fig. 3. Once the joint trajectories are known, the motor torque can be computed with standard procedures. In particular, a typical IR mechanical structure is composed by a series of rigid links connected by revolute joints whose torques can be computed using the well-known Lagrange formulation [25]. Indicating with $\boldsymbol{\tau_j}$ the 6×1 vector of joint torques and with $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ the 6×1 vectors of joint position, velocity and acceleration, for each time instant, it is possible to write:

$$\boldsymbol{\tau_j} = \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau_g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f} \quad (1)$$

where $\boldsymbol{H}$ is the 6×6 symmetric, positive-definite inertia matrix. $\boldsymbol{C}$ is the 6×6 matrix such that $\boldsymbol{C}\dot{\boldsymbol{q}}$ is the vector of Coriolis and centrifugal terms. $\boldsymbol{\tau_g}$ is the 6×1 vector of gravity terms including also effects due to eventual counterbalancing systems. $\boldsymbol{J}^T$ is the transpose of the 6×6 IR Jacobian matrix, $\boldsymbol{J}$, which is multiplied for the 6×1 vector, $\boldsymbol{f}$,of the external forces and toques applied on the end-effector.

The required joints' torque characteristics are not compatible with common electrical motors, so that gear reducers (generally with high reduction ratios) are employed. Henceforth, the torque, $\boldsymbol{\tau_m}$, and velocity, $\dot{\boldsymbol{q}}_{\boldsymbol{m}}$, required to the six motors can be computed as:

$$\boldsymbol{\tau_m} = \boldsymbol{G}^{-1}\boldsymbol{\tau_j} + \boldsymbol{\tau_i} + \boldsymbol{\tau_f} \quad (2a) \qquad \dot{\boldsymbol{q}}_{\boldsymbol{m}} = \boldsymbol{G}\dot{\boldsymbol{q}} \quad (2b)$$

where $\boldsymbol{G}$ is a 6×6 gear reduction ratio diagonal matrix. The terms $\boldsymbol{\tau_i}$ and $\boldsymbol{\tau_f}$ stand for 6×1 vectors of torque contributes due to inertias and frictions, which can be computed as:

$$\boldsymbol{\tau_i} = (\boldsymbol{J_g} + \boldsymbol{J_m})\ddot{\boldsymbol{q}}_{\boldsymbol{m}} \quad (3a) \qquad \boldsymbol{\tau_f} = \boldsymbol{K_c}\,\mathrm{sign}(\dot{\boldsymbol{q}}_{\boldsymbol{m}}) + \boldsymbol{K_v}\dot{\boldsymbol{q}}_{\boldsymbol{m}} \quad (3b)$$

where $\boldsymbol{J_g}$ and $\boldsymbol{J_m}$ are, respectively, 6×6 diagonal matrices of the gears and the motors inertias measured on the motors shaft axis, whereas $\boldsymbol{K_c}$ and $\boldsymbol{K_v}$ are 6×6 diagonal matrices of Coulomb and viscous friction coefficients also referred to the motors shaft.

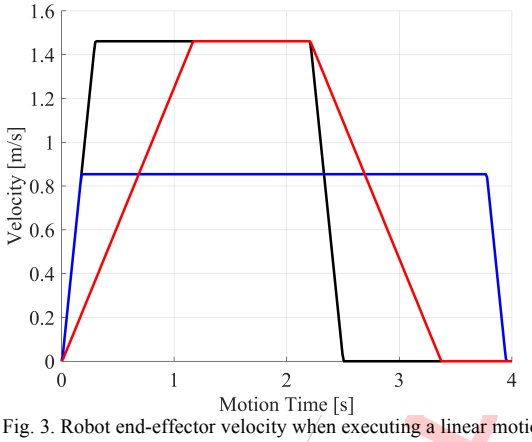

Fig. 3. Robot end-effector velocity when executing a linear motion with different velocity and acceleration limits.

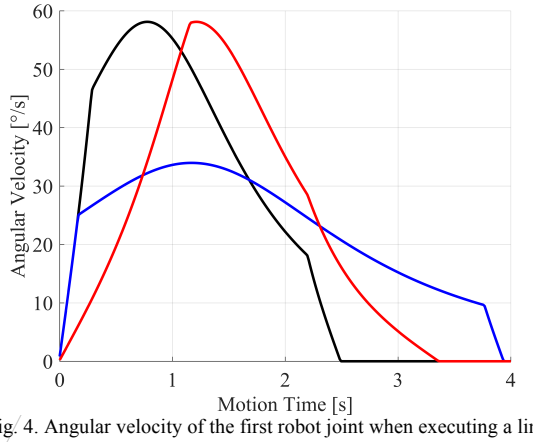

Fig. 4. Angular velocity of the first robot joint when executing a linear motion with different velocity and acceleration limits.

2.2. Electric motors

Due to their high dynamic characteristics, Permanent Magnet Synchronous Motors (PMSM) are typically used in industrial robots [26]. These motors generally present permanent magnets on the rotor and three-phase windings on the stator, which require to be correctly controlled by the drive system. For purposes of this work, a lumped parameter model of an equivalent DC-motor is used, whose dynamic behaviour is governed by these equation

systems:

$$\begin{cases} \boldsymbol{V_m} = \boldsymbol{R}\boldsymbol{I_m} + \boldsymbol{R_c}\boldsymbol{I_{m,c}} \\ \boldsymbol{V_m} = \boldsymbol{R}\boldsymbol{I_m} + \boldsymbol{L}\dfrac{d\boldsymbol{I_{m,t}}}{dt} + \boldsymbol{p}\boldsymbol{K_v}\dot{\boldsymbol{q}}_{\boldsymbol{m}} \\ \boldsymbol{\tau_m} = \boldsymbol{K_t}\boldsymbol{I_{m,t}} \\ \boldsymbol{I_m} = \boldsymbol{I_{m,c}} + \boldsymbol{I_{m,t}} \end{cases} \tag{4}$$

where $\boldsymbol{V_m}$ and $\boldsymbol{I_m}$ are the 6×1 vectors of motors equivalent voltage and current, $\boldsymbol{R}$, $\boldsymbol{L}$ and $\boldsymbol{R_c}$ are 6×6 diagonal matrices of the stator resistances, the stator inductances and the core resistances used for efficiently model PMSM core losses, similar to [24]. The terms $\boldsymbol{I_{m,c}}$ and $\boldsymbol{I_{m,t}}$ indicate 6×1 vectors of currents flowing through the core resistances and the currents generating the torque. At last, the terms $\boldsymbol{K_v}$, $\boldsymbol{K_t}$ and $\boldsymbol{p}$ respectively indicate 6×6 diagonal matrices of beck emf constants, torque constants and numbers of pole pairs. As shown in [24], this equivalent model parameters are related to the real PMSM phase parameters by a multiplying factor of $3/2$.

Starting from required motor shafts torques and velocities computed from Eqs. 2a and 2b, it is possible to calculate the required motor equivalent voltage and current solving the system of Eq. 4. The 6×1 vector of instantaneous powers, $\boldsymbol{P_m}$, delivered to each motor is simply given by:

$$\boldsymbol{P_m} = \mathrm{diag}(\boldsymbol{V_m})\,\boldsymbol{I_m} \tag{5}$$

For security issues, common IR motors are equipped with normally closed brakes, that are kept opened during the motions and released (after a predefined period) whenever the robot is stationary. When the brakes are released, the power required by the motors becomes null. The power consumption to keep the brakes opened is indicated as P_{br} and will be taken into account hereafter for the computation of total IR energy consumption.

2.3. Drive system

The drive system comprises the group of components between the mains (i.e. the electric energy source) and the motors, with the aim of correctly modulating the electrical power for motors alimentation. A schematic diagram for the drive components and their connections for an actual IR is given in Fig. 2 and explained in the following. Starting from the three-phase mains, a rectifier creates a common DC-bus from which the six motors draw the required power modulated through six inverters. This architecture allows for energy exchange between motors, in fact during braking phases, energy is pumped into the DC-bus and used by other motors or stored into the capacitance causing a DC-bus voltage increasing. For security issues, a braking resistor intervenes whenever the DC-bus voltage exceeds a predefined limit, thus dissipating energy and restoring the voltage within a secure threshold. The abovementioned drive components dynamic behaviour can be predicted using complex and very accurate models [27, 28], which however require a deep knowledge of their electrical and logical architecture generally unknown for industrial robots. For the purpose of EC prediction, simpler yet accurate models are well described in [24] and briefly reported hereafter.

The inverter model is based on the simple power balance:

$$\boldsymbol{P_{inv}} = \boldsymbol{P_m} + \boldsymbol{P_{loss,inv}} \tag{6}$$

where the inverter input powers, $\boldsymbol{P_{inv}}$, is the sum of the power required by the connected motor, $\boldsymbol{P_m}$, computed with (5), and the inverter losses, $\boldsymbol{P_{loss,inv}}$, computed as:

$$\boldsymbol{P_{loss,inv}} = \boldsymbol{K_c}\boldsymbol{I_m^2} + \boldsymbol{K_{sw}}|\boldsymbol{I_m}| \tag{7}$$

the terms $\boldsymbol{K_c}$ and $\boldsymbol{K_{sw}}$ being 6×6 diagonal matrices of conduction and switching-losses coefficients used to compute the respective losses terms, both related to the motor current computed in (4). With the knowledge of the powers required by the inverters and using the algorithm described in [24], is possible to estimate the behaviour of the DC-bus, namely its voltage, the energy stored into the capacitance and the energy dissipated through the braking resistance. Now, similar to the inverters, the rectifier input power, P_{rec}, is computed correlating its losses to the current flowing into the DC-bus.

Finally, the total power consumption of the industrial robot, P_{ir}, is obtained as:

$$P_{ir} = P_{rec} + P_{const} + P_{br} \tag{8}$$

where P_{const} is a term accounting for all the system constant power consumption such as for PCs, cooling fans, etc. The total energy that the industrial robot drawn from the mains in the time interval $[0, T]$ is, then, given by:

$$E_{ir} = \int_0^T P_{ir}(t)\, dt \tag{9}$$

3. Computation of a single motion energy consumption

3.1. The generalized motion

Usually, a production process requires robot motions with fixed duration, e.g. when the preceding and following motions are process relevant and not modifiable. In case of a motion duration shorter than the required duration the robot waits on its finial configuration for the remaining time and the motors have to produce the required holding torque until the brakes are released. The energy consumed during the waiting time must be taken into account for a correct energy optimization, so that a convenient approach is to conceptually consider each robot motion as a sequence of a moving phase and a steady phase. Doing so, the EC of this generalized motion can be computed as the sum of the two phases, such that:

$$E_{ir,gmov} = \underbrace{\int_0^{T_{mov}} P_{ir}(t)\, dt}_{E_{ir,mov}} + \underbrace{\int_{T_{mov}}^{T_{fix}} P_{ir}(t)\, dt}_{E_{ir,ste}} \tag{10}$$

where T_{mov} and T_{fix} are, respectively, the duration of the moving phase and the total fixed available time for the motion. Starting form this general case, motions without the steady phase or without the moving phase can be easily obtained respectively setting $T_{fix} = T_{mov}$ or $T_{mov} = 0$. Obviously, the condition $T_{mov} \leq T_{fix}$ must be always verified, otherwise the motion is considered infeasible.

3.2. Trajectory computation

Due to typical restrictions of actual IR controllers, it is not possible to execute a generic trajectory. The programmers make use of the basic commands provided by the robot manufacturers for defining the spatial path that the robot end-effector has to follow, although having no possibilities to exactly impose the evolution in time of such motion (trajectory). The trajectory can be approximately modified changing some parameters, such as maximum velocity and acceleration, but the exact relation of these parameters with the trajectory formula is unknown and protected by the robot manufacturer. In the past, robot programmers considered this restriction as a simplification; common practice was to run the robot at the maximum speed allowed by the process with the only intent to comply with the imposed cycle time. In parallel, the predictive simulation of the production line became more and more important leading to the definition of the Realistic Robot Simulation (RRS) interface, [18]. Each manufacturer provides a *Robot Controller Simulation* (RCS) specific module, which works as a black-box computing the trajectories with the same algorithms used in the real controller. Today the RRS-interface is the world-wide de-facto standard for precise simulation of robot motion behaviour. In this work, *Delmia Robotics V5* (from *Dassault Systèmes*) was used as simulation environment taking advantage of the accurate trajectory computation retrieved by the RCS module relative to a *KUKA KR C4* controller connected to a *KUKA Quantec KR 210 R2700 prime* robot. After the simulation within *Delmia*, information about IR joint trajectory can be exported into a file.

3.3. Energy consumption computation

With reference to Fig. 5a, a Matlab implementation of the IR model described in Section 2 uses the *Delmia*

Robotics simulation data to compute the IR moving phase EC and correct it summing the steady phase EC basing on the available time for the motion. Since a single motion is analysed, a correct initial condition of the system must be imposed: it is assumed that at the initial instant no usable energy is stored into the DC-bus capacitance so that the system is stationary. This hypothesis is the most logical one if only one motion is considered since it is impossible to determine the initial state without simulating the entire robot cycle (planned in future works). Furthermore, it is possible that at the final instant some recuperated energy remains into the system capacitance, available for the subsequent motions. This energy was draw from the mains but only temporally used by the robot so that it cannot be considered part of the generalized motion EC and for that it is properly removed.

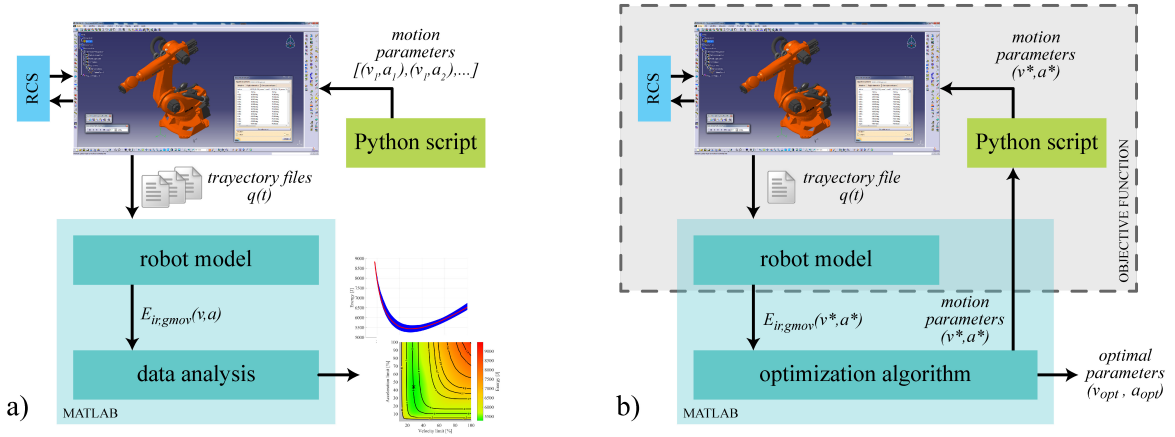

Fig. 5. Schematic of the procedure for motion parameter analysis (a) and optimization (b).

4. Determining optimal motion parameters

As mentioned before, robot tasks are actually programmed using motion commands provided by the manufacturer, which define the interpolating trajectory between determined targets. Each command requires the definition of some parameters to be set by the programmer in function of the desired motion characteristics. The actual programming trend is uniquely based on a time criterion, so that the task motions parameters are chosen as to complete the desired operations within the imposed cycle time. No attention is usually paid to the robot EC mainly because of the lack of adequate tools for assisting the designer on this purpose. Only in last years, with the continuously increasing attention to energy efficiency, some simulation software (i.e. *ABB RobotStudio* [16]) implemented the computation of an estimate of the robot energy consumption, however without providing any built-in optimization routine. In order to fill this lack, a preliminary tool is developed which integrates with state of the art simulation software, with the aim of assisting the programmer in an accurate choice of the motion parameters. As said, two commonly settable parameters are considered here, namely the velocity and acceleration limits of the motion, although the method can be easily extended considering every other freely adjustable parameter.

4.1. Motion parameters analysis

As an example, a standard linear motion has been analyzed varying the percentage of maximum velocity v, and acceleration, a. At first, a series of simulations has been executed in *Delmia Robotics* changing both v and a on a predefined grid of combinations. The process has been automated using a Python script, which repetitively sets the parameters within *Delmia Robotics*, runs the simulation and exports the results into a file. Secondly, for each generated file the generalized motion EC has been computed as explained in Section 3. Thirdly, the obtained information has been analyzed and graphically arranged to be better interpretable. In Fig. 6, data are presented as a

combination of a color map indicating the motion EC and an isoline contour map showing the moving phase time duration. The corresponding limits of velocity and acceleration are expressed on the axes as percentage of the robot maximum ones. This kind of plot clearly shows the dependence of the energy consumption on the imposed parameters and permit to find the optimal ones, indicated by the "X".

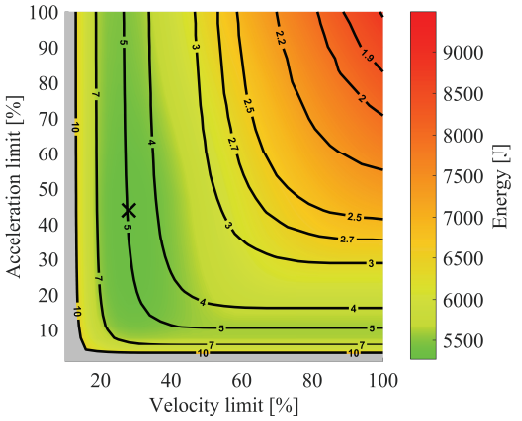

Fig. 6. Energy consumption (colour map) and motion time (isolines) as function of velocity and acceleration limits.

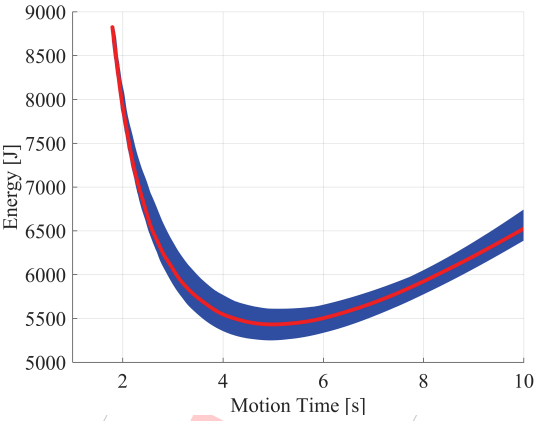

Fig. 7. Energy Signature (red line) compared to possible EC variations when varying velocity & acceleration (blue area).

For comparison purposes, in Fig. 7, the data are also arranged into another form. The red line is the *Energy Signature* of the analyzed motion obtained with a linear trajectory scaling of the fastest possible motion [19]. The blue area indicates the motion times and energy consumptions obtainable changing the velocity and acceleration limits. It is evident that a better minimum of energy can be reached tuning the motion parameter instead of linearly scaling the trajectory.

4.2. The optimization tool

Whit the just described motion analysis an enormous quantity of information is retrieved, which permits an accurate understanding of the motion EC. However, usual practical cases only require the knowledge of the energy-optimal parameters. To reach this intent with the minimal computational effort, an optimization procedure has been created, the conceptual scheme being illustrated in Fig. 5b. The software package *Matlab* controls the optimization process through a pattern search algorithm aimed at minimizing the generalized motion EC with the nonlinear constraint of the fixed motion duration. The objective function involves a Python script for setting the optimization parameters in *Delmia Robotics*, run the simulation and export the trajectory data file, which is used for the energy computation (as described in Section 3). At the end of the process, the optimal motion parameters are retrieved.

The optimization of the motion previously analyzed requires 33 simulation loops and produces results matching those derivable from the map in Fig. 7 confirming the validity of the approach.

It's interesting to underline that 98% of the computational time is spent for the computation of the trajectory. This is principally due to the limited functionalities offered by *Delmia Robotics* in the automation of the simulation process. Furthermore, not much attention has been devoted to the script efficiency, since this objective goes beyond the purposes of this work. On the other hand, an integration of these procedures directly into the simulation software can enormously increase the computational efficiency offering a useful tool for an energy efficient programming.

5. Conclusions

In this paper a novel energy consumption optimization method, based on the intensive use of a complete and accurate industrial robot model, has been proposed. In particular, starting from accurate trajectories exported from simulations in *Delmia Robotics* environment, with the use of the robot specific RCS module, a robot motion has been analyzed, obtaining the relation between the adjustable motion parameters and the energy consumption.

Results show that an accurate choice of the motion velocity and acceleration limits can lead to meaningful energy consumption reduction. Furthermore, a practical tool has been developed for assisting the designer in the robot energy efficient programming, by retrieving the optimal motion parameter combination also considering available time constraints. Future works will concern experimental validations of the obtained results and the extension of the method for multi-robot cell optimization with constraints on cycle time, sequencing and collision avoidance.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No. 609391 (AREUS).

References

[1] Brundlandt, G.H. Our Common Future. Report of the World Commission on Environment and Development. Oxford University Press, 1987.
[2] Jayal, A.D., Badurdeen, F., Dillon Jr., O.W., Jawahir, I.S. Sustainable manufacturing: Modeling and optimization challenges at the product, process and system levels. CIRP Journal of Manufacturing Science and Technology, 2(3), pp. 144-152, 2010.
[3] Jovane, F., et al. The incoming global technological and industrial revolution towards competitive sustainable manufacturing. CIRP Annals - Manufacturing Technology, 57(2), pp. 641-659, 2008.
[4] Edenhofer, O., et al. On the economics of renewable energy sources. Energy Economics, 40(1), pp.12-23, 2013.
[5] Twidell, J., Weir. T. Renewable Energy Resources. Routlege, Taylor and Francis Group, 2015.
[6] Masters, G.M. Renewable and Efficient Electric Power Systems. Wiley and IEEE press, 2013.
[7] VDMA–Packaging Machinery Association: DIN 8743. Packaging machines and packaging installations, time related definitions, reference factors and calculation fundamentals, available online, 2005.
[8] Berselli, G., Balugani, F., Pellicciari, M., Gadaleta, M. Energy-optimal motions for Servo-Systems: A comparison of spline interpolants and performance indexes using a CAD-based approach. Robotics and Computer-Integrated Manufacturing, 40, pp. 55-65, 2016.
[9] Sundstrom, N., Wigstrom, O., Lennartson, B. Conflict between Energy, Stability, and Robustness in Production Schedules. IEEE Transactions on Automation Science and Engineering, 14 (2), pp. 658-668, 2017.
[10] M. Pellicciari, A. Avotins, K. Bengtsson, B. Lennartson, G. Berselli, N. Bey, D. Meike. AREUS - Innovative hardware and software for sustainable industrial robotics. IEEE International Conference on Automation Science and Engineering, vol. 2015-October (2015)
[11] Wigstrom, O., Lennartson, B., Vergnano, A., Breitholtz, C. High-Level Scheduling of Energy Optimal Trajectories. IEEE Transactions on Automation Science and Engineering, 10(1), pp. 57-64, 2013.
[12] Mohammed, A., Schmidt, B., Wang, L., Gao, L. Minimizing Energy Consumption for Robot Arm Movement. Procedia CIRP, 25, pp. 400-405, 2014.
[13] Bryan, C., Grenwalt, M., Stienecker, A. Energy consumption reduction in industrial robots. In Proceedings of ASEE North Conference, pp. 1-4, 2010.
[14] Björkenstam, S., Gleeson, D., Bohlin, R., Carlson, J. S., Lennartson, B. Energy efficient and collision free motion of industrial robots using optimal control. IEEE CASE International Conference on Automation Science and Engineering, pp. 510-515, 2013.
[15] ⟨https://www.3ds.com/products-services/delmia/products/v6/portfolio/d/digital-manufacturing-and-production/s/robotics-programmers/p/robotics-offline-programming/⟩ (accessed 30.05.17).
[16] ⟨http://new.abb.com/products/robotics/robotstudio/downloads⟩ (accessed 30.05.17).
[17] ⟨https://www.kuka.com/en-gb/products/robotics-systems/software/simulation-planning-optimization/kuka_sim⟩ (accessed 30.05.17).
[18] ⟨http://www.realistic-robot-simulation.org/⟩ (accessed 30.05.17).
[19] Pellicciari, M., Berselli, G., Leali, F., Vergnano, A. A method for reducing the energy consumption of pick-and-place industrial robots. Mechatronics, 23(3), pp-. 326-334, 2013.
[20] Gregory, J., Olivares, A., Staffetti, E. Energy-optimal trajectory planning for robot manipulators with holonomic constraints. Systems & Control Letters, 61(2), pp. 279–291, 2012.
[21] Ata, A. A. Optimal trajectory planning of manipulators: A review. Journal of Engineering Science and Technology, 2(1), pp. 32–54, 2007.
[22] Sergaki, S.E., Stavrakakis, G.S., Pouliezos, A.D. . Optimal Robot Speed Trajectory by Minimization of the Actuator Motor Electromechanical Losses. Journal of Intelligent and Robotic Systems, 33(2), pp. 187-207, 2002.
[23] Oliva, E., Berselli, G., Pellicciari, M., Andrisano, A.O. An engineering method for the power flow assessment in servo-actuated automated machinery: Mechatronic modeling and experimental evaluation. Robotics and Computer-Integrated Manufacturing, 38, pp. 31-41, 2016.
[24] Pellicciari, M., Berselli, G., Balugani, F. On Designing Optimal Trajectories for Servo-Actuated Mechanisms: Detailed Virtual Prototyping. IEEE/ASME Transaction on Mechatronics, 20(5), pp. 2039-2052, 2014.
[25] Siciliano, B., Khatib, O. (Eds.). Springer Handbook of Robotics. Springer-Verlag Berlin Heidelberg, 2008.
[26] Bose, B.K. Modern Power Electronics and AC Drives. Prentice Hall, Upper Saddle River, NewJersey, 2002.
[27] Malesani, L., Rossetto, L., Tenti, P., Tomasin, P. AC/DC/AC PWM Converter with Reduced Energy Storage in the DC Link. IEEE Transaction on Industry Applications, 31(2), pp. 287-292, 1995.

[28] Pillay, P., Krishnan, R. Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part I: The Permanent-Magnet Synchronous Motor Drive. IEEE Transaction on Industry Applications, 25(2), pp. 265-273, 1989.