Robotics and Computer–Integrated Manufacturing 47 (2017) 102–111

Contents lists available at ScienceDirect

Robotics and Computer–Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Energy-optimal layout design of robotic work cells: Potential assessment on an industrial case study

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

[a] Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena 41125, Italy
[b] Department of Mechnical Engineering, Energy, Managment and Transportations, University of Genova, Genova 16145, Italy

A B S T R A C T

This paper presents a new method for optimizing the layout position of several Industrial Robots (IRs) placed within manufacturing work-cells, in order to execute a set of specified tasks with the minimum energy consumption. At first, a mechatronic model of an anthropomorphous IR is developed, by leveraging on the Modelica/Dymola built-in capabilities. The IR sub-system components (namely mechanical structure, actuators, power electronic and control logics) are modeled with the level of detail strictly necessary for an accurate prediction of the system power consumption, while assuring efficient computational efforts. Secondly, once each IR task is assigned, the optimal work-cell layout is computed by using proper optimization techniques, which numerically retrieve the IR base position corresponding to the minimum energy consumption. As an output to this second development stage, a set of color/contour maps is provided, that depicts both energy demand and time required for the task completion as function of the robot position in the cell to support the designer in the development of an energy-efficient layout. At last, two robotic manufacturing work-cells are set-up within the Delmia Robotics environment, in order to provide a benchmark case study for the evaluation of any energy saving potential. Numerical results confirm possible savings up to 20% with respect to state-of-the-art work-cell design practice.

1. Introduction

Industrial robotics may be envisaged as the most strategic technology that can practically enable *flexible automation* and *intelligent manufacturing processes*. Unfortunately, Industrial Robots (IRs) and related peripheral machines are intrinsically energy intensive, thus compromising the overall factory sustainability. In particular, as previously proven in several researches (e.g. [1,2]), IR massive adoption heavily impact both factories ecological footprint and overall production costs. In addition, many companies are currently facing severe issues related to the actual unavailability of electric energy supplier, which can withstand the peak power requirements as the number of IRs simultaneously employed within the same location exceeds a certain threshold. Naturally, owing to this restriction, any further industrial development may end up being heavily damped, or even impossible. Therefore, the industrial need towards possible strategies to reduce the energy consumption (EC) of single IRs and/or robotic work cells at factory level is unquestionable.

Within this scenario, current and past research activities concerning *mechatronic eco-design methods* have been rapidly gaining a strong foothold in the scientific arena, resulting in an increasing number of programs funded by both academia and industry [3]. For instance, energy optimization by means of IR electromechanical hardware replacement is addressed in [4]. At robotic cell and process design level, energy-optimal robot selection for specific operations is investigated in [5], whereas many past works deal with energy-optimal path generation, see e.g. [6]. In [7], novel methodologies aim at achieving an EC reduction while avoiding plant revision. These techniques are based on the optimization of the IR velocity profiles without affecting productivity and quality, thus providing interesting solutions that should aid programmers to develop more sustainable applications without affecting investment costs.

For what concerns IR positioning, the layout design of a robotic cell is a delicate task performed under conflicting requirements: on one hand, robots and peripheral equipment must be placed assuring best reachability and process performance; on the other hand, design rules that assure safety and ease of maintenance must be strictly observed. Owing to these concurrent needs, along with the lack of industrially-viable engineering tools, the design of robotic work-cells is currently tackled with a trial-and-error approach mostly based on the designer

* Corresponding author.
E-mail address: marcello.pellicciari@unimore.it (M. Pellicciari).

http://dx.doi.org/10.1016/j.rcim.2016.10.002
Received 8 December 2015; Received in revised form 6 October 2016; Accepted 6 October 2016
Available online 25 October 2016
0736-5845/

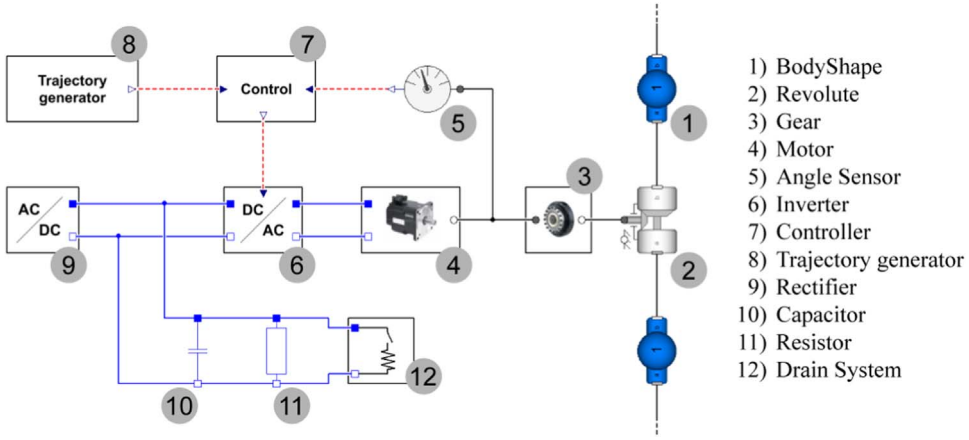

Fig. 1. Servomechanism mechatronic model realized in Dymola.

experience, while the EC performance is usually neglected and verified only during the final commissioning. On the other hand, even if the industrial design practice has been somehow unreceptive to the academic researches, the problem of finding the optimal IR location with respect to an assigned task has been investigated since the late 80 s, when Nelson et al. [8] studied the IR base positioning that would enhance a manipulability measure. Similarly, the cycle time reduction is addressed using different robot models in [9,10]. Chedmail and Wenger [11] introduced the obstacle problem, searching for an IR location permitting the task execution without collisions. In [12] the energy-optimal task placement inside the workspace of a 3-axis parallel robot was found by modeling the electrical motors consumption. In practice, the literature review furtherly underlines the importance and complexity of the layout positioning of the robot, although most of the past works focus on cycle time and manufacturing quality rather than EC reduction. On the basis of these considerations, we hereby propose a method, along with the related computer-aided tools, which supports the design engineers in the choice of an energy efficient cell layout. Extending the work presented in [13], at first, an object-oriented model of a common six-degrees-of-freedom anthropomorphous manipulator is created, focusing on accurate EC prediction and computational efficiency. Similarly to previous literature [14,15], the developed IR model comprises a description of the mechanical structure, actuation system and control architecture. Differently from previous works, the model also describes the electrical behavior of the entire electronic driver. Then, proper optimization algorithms are employed, to quickly and efficiently compute the energy-optimal IR base location for a specified task. In parallel, layout maps are generated, clearly showing the relation between robot position, energy consumption and cycle time, aiding an energy-efficient layout design. At last, the energy-saving potentials are assessed on industrial case studies optimizing the placement for three robots programmed for spot-welding and pick-and-place in automotive production lines.

The following software tools have been used for modeling, optimization, and subsequent numerical validation due to their built-in functions and wide spread usage in both academia and industry:

- *Dymola*, [16], a powerful environment for model based simulation relying on *Modelica* language [17], which offers significant advantages over other modeling techniques, namely graphical representation, reusability of code blocks and ease of modification & interpretation. The Dymola environment has been used for the IR model creation and for the subsequent optimization phase.
- *Delmia Robotics*, [18], one of the most widespread off-line programming and robotic simulation software. In particular, this virtual prototyping tool provides a 3D graphical interface where the plant designer can conceive/simulate/debug an entire work-cell or production line, finally generating the IR code to run the physical plant. In this context, *Delmia Robotics* has been employed for validating the final optimization results.

The paper is organized as follows: Section 2 describes the IR model within the Modelica/Dymola environment; Section 3 describes the proposed base-position optimization tools; Section 4 discusses the industrial case study and the achievable energy improvements; Section 5 reports the concluding remarks.

2. Modeling

2.1. The servomechanism model

Although rather accurate, the single models of each IR subsystem component found in the previous literature (see e.g. [19]) include a detailed description of several physical phenomena, thus relying on numerous parameters that are often unknown. On the contrary, the approach adopted in this paper employs simpler equations, that require a minimum number of parameters [20], yet accurate enough to predict the system power flow. In particular, the overall IR mechatronic model is obtained from the composition of more servomechanisms models including mechanical, electrical and control functional components described via the *Modelica/Dymola* language. The model layout is provided in Fig. 1, which shows the servomechanism model (along with functional components and connections) realized in *Dymola* environment. The mechanical structure of the system, reported on the rightmost part of the picture, is defined in a very efficient way using the standard *MultiBody* library: *BodyShape* and *Revolute* components respectively describe rigid bodies and revolute joints (characterized with their lengths, masses, inertias and rotation axis parameters).

Similarly to [21], the gear-box block (directly connected to the driving flange of the revolute joint) employs a simple but efficient model. The torques acting at two sides are related by the following equation:

$$\tau_{in}+\frac{\tau_{out}}{r}=\left(\alpha\left|\frac{\tau_{out}}{r}\right|+\beta\right)sign(\omega_{in})+f_v\,\omega_{in}+J_{in}\dot{\omega}_{in} \quad (1)$$

where r is the reduction ratio, α and β model the load dependent Coulombian friction, f_v is the viscous friction coefficient, J_{in}, ω_{in} and $\dot{\omega}_{in}$ are the gear inertia, the angular velocity and the angular acceleration referring at input side. Following modeling guidelines reported in [22], the IR Permanent Magnet Synchronous Motors (PMSMs) block diagram comprises one inductance, two resistors, the rotor inertia, and the electro-mechanical conversion block (via the back emf constant). For what concerns the dissipative elements, one resistor accounts for the

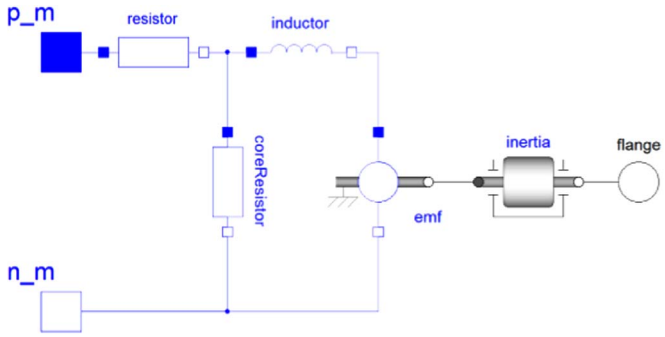

Fig. 2. Motor model.

armature losses, whereas the second resistor (referred to as *coreResistor* in Fig. 2) accounts for the iron losses, assuring a better accuracy of the overall EC prediction As for the control system (Fig. 1), it includes a classic cascade feedback structure with an outer proportional position loop and an inner proportional-integral velocity loop. In parallel, following the power flow from the AC mains, a rectifier creates a medium stage DC-bus from which an inverter generates the correct motor voltages (computed by the control logics), typically using the Pulse Width Modulation (PWM) technique, [19]. However, an accurate model of the PWM physical behavior is not rewarding for mere EC prediction purposes, so that an inverter model which is simply based on the power balance equation was chosen for the purpose, such that:

$$P_{inv}=P_m+P_{loss,inv} \tag{2}$$

where P_{inv} and P_m are respectively the input powers to the inverter and to the motor, whereas $P_{loss,inv}$ represents the power loss in the inverter. The latter can be computed as sum of two contributes, one related to the conduction losses, $P_{cond,inv}$, and the other accounting for the switching ones, $P_{sw,inv}$ [23]. Both contributions are related to the current flowing into the motor, i_m, and are computed as:

$$P_{cond,inv}=Ri_m^2 \tag{3}$$

$$P_{sw,inv}=k_{sw}|i_m| \tag{4}$$

where R is the conduction resistance and k_{sw} the switching losses constant. In conclusion, the inverter model is based on two parameters only, which can be identified using the datasheet [23] and/or more classical identification techniques [20]. The overall inverter block-diagram is presented in Fig. 3: the DC-bus connection pins can be found on the leftmost part of the picture, whereas the motor connections are depicted on the rightmost part. The *SignalVoltage* component imposes the motor voltage value computed by the control system and received through the *V_m* connector. The *SignalCurrent* component, instead, is used to draw the correct amount of power (as computed by means of Eq. (2)), enforcing a current flow that depends on the instantaneous DC-bus voltage. The inverter input current, i_{inv}, and the DC-bus voltage, V_{dc}, are simply related as follows:

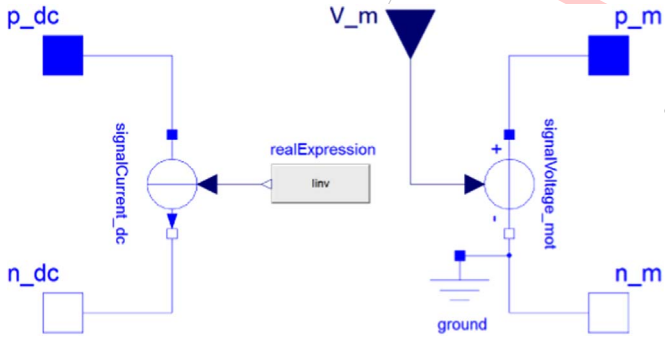

Fig. 3. Inverter model.

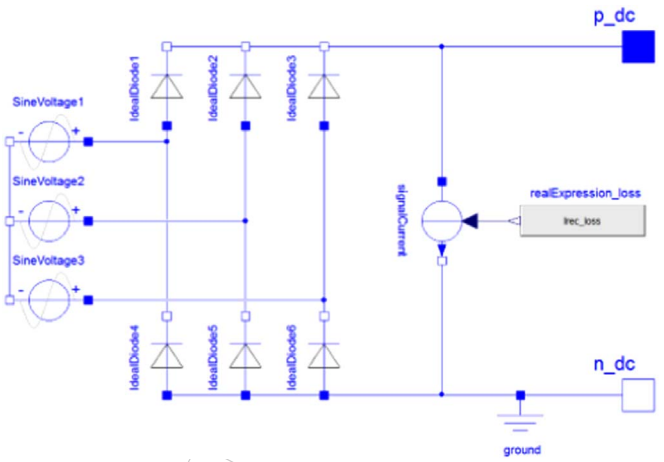

Fig. 4. Rectifier model.

$$i_{inv}=\frac{P_{inv}}{V_{dc}} \tag{5}$$

This inverter scheme enables power backflow from the motor to the DC-bus that can occur during motor braking. In this situation, the recovered energy is stored into the DC-bus capacitance causing a DC-bus voltage increase. For security issues, a drain system limits the voltage to be under a maximum value, dissipating the energy in excess. Fig. 4 presents the implemented Dymola scheme of a commonly employed rectifier [19], composed of six diodes connected in a full bridge configuration. Considerations on inverter losses are also valid for the rectifier: conduction and switching losses are computed in relation to the current flowing into the DC-bus and implemented through the *SignalCurrent* component. Owing to its simplicity and low costs, such rectifier architecture is very common in the industrial environment, although it produces a rectified voltage with a ripple of tens of Volts (dependent on electrical network characteristic). Such ripple is undesired, as it adversely affects the inverter performance. Therefore, the ripple is physically reduced principally through the DC-bus capacitance that accumulates energy when voltage increases and releases it when necessary [24]. The last component in the model is a resistance connected to the DC-bus, which accounts for the related current leakages.

2.2. Computational efficiency improvement

The described servomechanism model can accurately compute the electro-mechanical behavior of the entire system, retrieving a good accuracy in terms of power consumption. Nonetheless, computational efficiency should be further improved. In particular, major computational efforts are related to the full-bridge rectifier model, which includes the description of six diodes (see Fig. 4), forcing the Dymola solver to generate an event whenever a diode switches its state (from conduction to interdiction and vice versa). This modeling approach allows to compute the instantaneous DC-bus voltage ripple, although resulting in hundreds of events per second. Fortunately, for the purpose of this work, an accurate ripple prediction is not necessary, and a modeling approach that focuses on the medium DC-bus voltage value should be preferred. In practice, an approximated voltage value is imposed with a *SignalVoltage* component, as in Fig. 5, which implements an approximated formula previously proposed by the authors (see. Equation (38) in [23]). Current backflows towards the mains (not allowed by the intrinsic architecture of the considered rectifier) are finally prevented by the inclusion of a single diode.

In addition to the previous considerations, it should be highlighted that IR control systems are robust enough to realize the imposed trajectories with good accuracy. Therefore, since negligible position

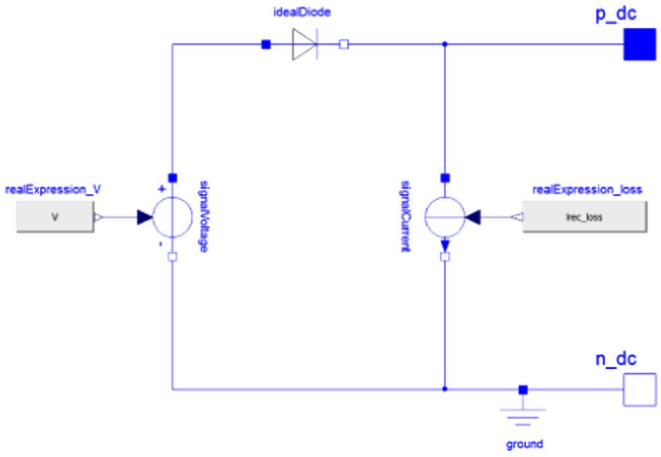

Fig. 5. Simplified rectifier model.

errors do not considerably influence the energy consumption, the control system contribution can be ignored. In order to explain the conceptual steps performed during model simplification, let one consider the sub-system depicted in Fig. 6a (composed of gear-box, motor, inverter and control system), that requires the reference trajectory as input, retrieving the real joint position as output. As long as the Modelica language is inherently a-causal [25] (i.e. models are described with equations without defining input and output variables), it is quite easy to invert the considered sub-system by directly enforcing the real joint position as input. The model inversion is obtained by means of the Modelica standard block *InverseBlockConstraints,* which encloses the model and is connected as in Fig. 6b. In this case, the Dymola solver computes the reference set-point required to exactly produce the imposed joint trajectory. As a further improvement, the control system block-diagram is removed, as in Fig. 6c, allowing the solver to compute only the required inverter input signal, corresponding to the required motor voltage. At last, Fig. 6d shows the resulting inverse model that will be used in the following section, the difference with Fig. 6c being only a rational re-organization of the model blocks for clarity purposes.

2.3. The complete industrial robot model

As widely known, typical robots employed in industries are anthropomorphous six joints serial manipulators. In particular, the complete IR mechatronic model is composed of six of the previously described servomechanisms with minor architectural difference, namely a single rectifier that creates a common DC-bus on which six

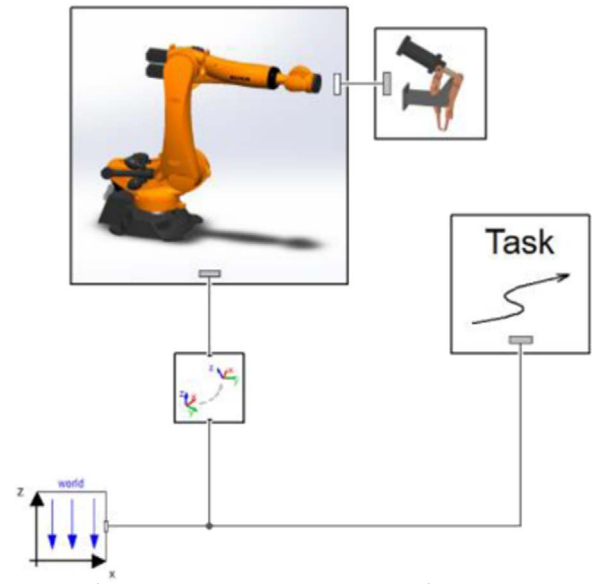

Fig. 7. Cell model.

motor inverters are simultaneously connected. Note that, even if the electrical power input is unidirectional, the IR multi-drive system allows for motor-to-motor energy exchange. Therefore, when an axis decelerates, the mechanical energy is converted back into electrical energy to be either used by other motors (if needed), stored in the DC bus capacitor, or simply dissipated via the abovementioned drain system. The overall Modelica block diagram is reported in Fig. 8. The rightmost part of the picture depicts a chain of rigid bodies and joints that define the IR mechanical structure, accounting also for the counterbalancing system mounted between the first and the second link. The first and the last mechanical structure reference frames, namely those of the IR base and flange, are used for external connections: the base frame defines the robot positioning, whereas the flange frame defines the end-effector pose. The total industrial robot EC is computed through an equation inserted in the model code which integrates the power on the cycle time, T, such that:

$$E_{ir} = \int_0^T P_{rec} + \int_0^T P_{ir,const} = E_{rec} + E_{ir,const} \tag{6}$$

where $P_{ir,const}$ and $E_{ir,const}$ are respectively the constant power consumption of the system (related to computers, fans, etc.) and the relative energy. The *trajectory generator block* computes the reference trajectory for each joint, starting from a table containing a list of motions

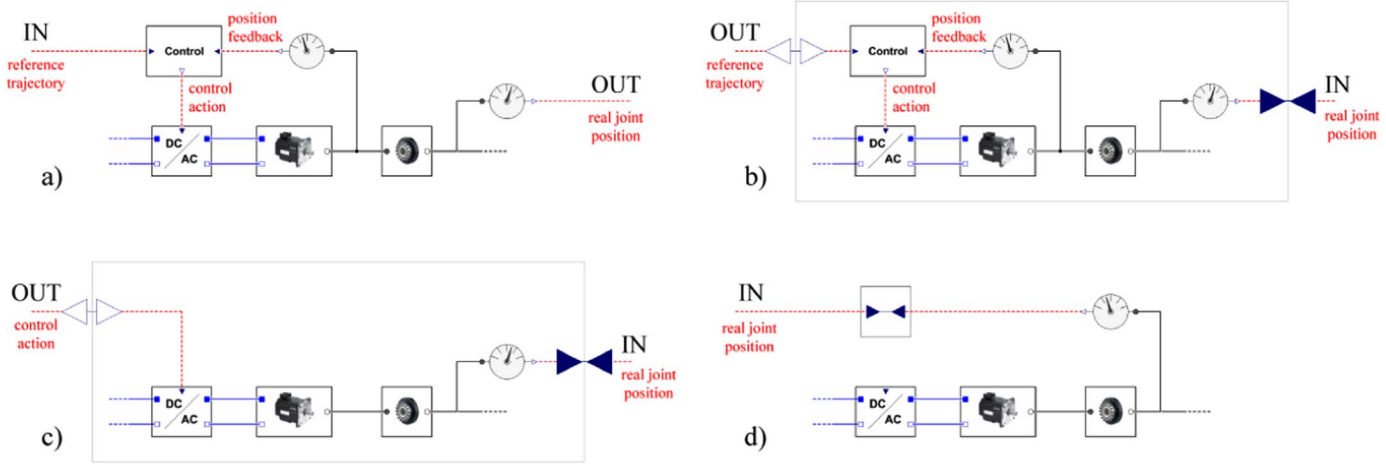

Fig. 6. Steps for model inversion.

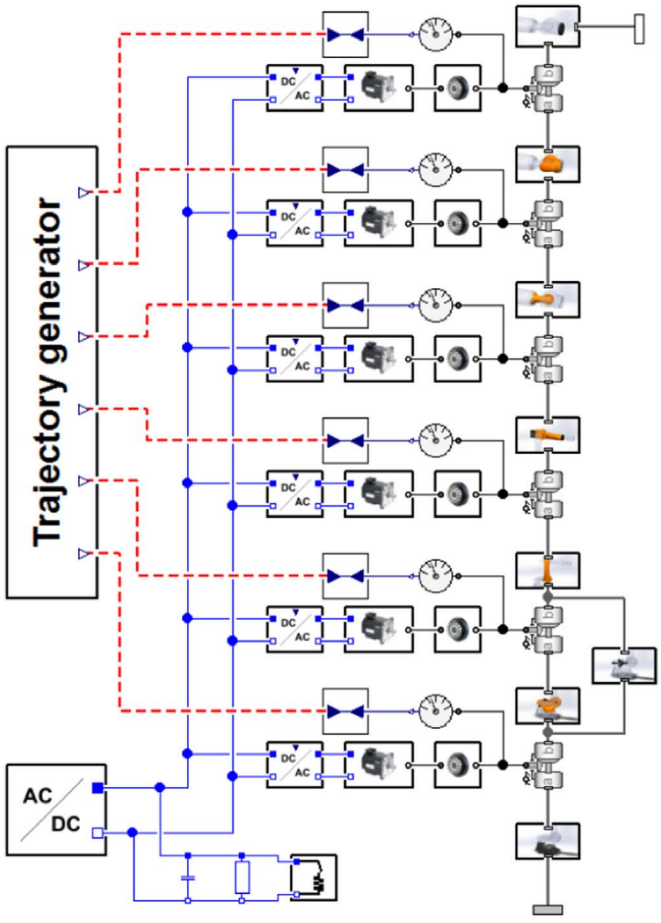

Fig. 8. Industrial robot mechatronic model.

instructions parameters defining the robot task. Target points are expressed in the operational space defining the Cartesian position and orientation with respect to the robot base frame, whereas the implemented inverse kinematic function computes the corresponding joint values (also accounting for their physical end limits). The algorithm implemented in the *trajectory generator block* computes the reference trajectory (in either operational or joint space) by enforcing the fastest possible motion under kinematical constraints of maximum velocity, acceleration and jerk. Very similar trajectory planners are commonly employed by the robot manufacturers, although the actual algorithms are hardly identifiable, proprietary data. The discrepancy between algorithms adopted in real robots and those implemented in simulation software is probably the most important source of error affecting simulation results.

In fact, the need for the computation of extremely accurate trajectories with real robots algorithms has led to the definition of the Realistic Robot Simulation (RRS) standard [26], which defines a standard interface for communicating with the so-called Robot Controller Simulation (RCS) module. The RCS module is a black box software, developed and provided by the robot manufacturer, which computes the reference trajectory with superior accuracy since it embeds proprietary algorithms used by real robots. In particular, each robot manufacturer provides an RCS module to improve the accuracy of the robot trajectories developed with robot offline programming and simulation tools, avoiding the need of experimental refinement. In current industrial practice, the use of such RCS module is accepted and validated [1]. On the other hand, it has not been implemented into this research work mainly because of two reasons: i) despite its effectiveness and widespread usage in industry, the communication protocol with the module is not yet an open standard, thus limiting its general applicability for research purposes; ii) its use inside the Dymola model would have drastically reduced the computational efficiency. Furthermore, industrial users expressly asked for optimization times below 5 min. In any case, for what concerns the trajectory generator employed in this work, its reliability and efficacy are proven by the numerical results presented hereafter.

3. Robot position optimization tools

The accurate and computationally efficient IR model presented in the previous chapter is used as the basis for the creation of innovative tools aimed at optimizing the cell layout, thus finding the robot base location that minimizes the EC needed to accomplish an assigned task. For this purpose, a *Cell* model has been created in the Dymola environment to approach the problem with high flexibility, see Fig. 7. The standard *World* block defines the absolute cell reference frame and the direction of the gravity vector. The IR base position is defined connecting its base flange to the world flange through an appositely created block, which (differently from the Modelica standard library components) allows the variation of the base coordinates after the model translation, thus increasing efficiency of the optimization procedure. Furthermore, the robot base is automatically rotated towards the central point of the task, avoiding unnecessary extra-rotations of the first robot joint that may occur in some base locations due to the end limits. The end effector block is connected to the IR tool flange defining mass, inertias and geometrical parameters of the end effector. The *Task* block permits the definition of the task in the cell reference frame; whereas a script automatically expresses target points in robot base coordinate system for the trajectories computation.

After the Cell model setup, that is the definition of robot, end effector and task data, it is possible to run the simulation: the time evolution of all the variables in the model is computed and retrieved as output by the Dymola software together with a useful 3D animation, see Fig. 9. This tool allows the user to obtain various interesting information about the robotic system, such as the power requirements and the subsequent energy costs. Instead, in order to find the best robot placement in function of the specified task, two approaches are proposed:

The first is based on a standard optimization algorithm, which looks for the robot base position that minimizes the assigned cost function. The optimization tool has been created using the Dymola *Optimization* library: the total robot EC has been set as the cost function to be minimized by tuning the robot base coordinates through the *Pattern Search* optimization algorithm.

The second approach consists on maps generation, with the intent

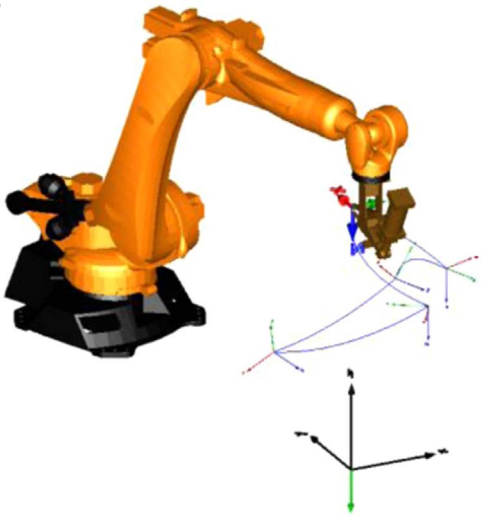

Fig. 9. 3D visualization.

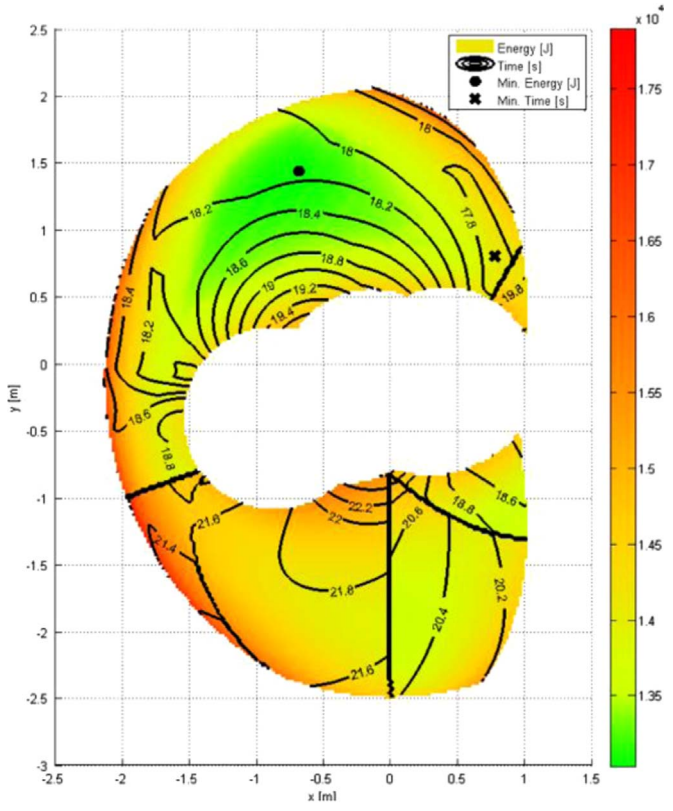

Fig. 10. Time/Energy map.

to provide more information to the designer, offering the possibility to graphically choose the best IR location, also evaluating un-modeled constraints (e.g. reachability, collisions or safety rules). Using Dymola built-in capability, a function for map generation has been created. Such function retrieves as input a grid of robot base coordinates, it executes the simulations placing the robot on each grid point, and subsequently stores relevant data into matrices; then a script is run, which exploit Matlab capability for image creation. In Fig. 10, an example is presented. The color map and the superimposed contour-line map respectively indicate the amount of energy and the time required for the task execution as function of the robot base position. The robot can be placed only in the colored zones, whereas the white areas depict prohibited zones where the robot would end up being incapable of performing the entire task due to its kinematic limits.

The first optimization approach has the advantage of being very fast, although retrieving as output only the best robot position; in case the robot may not be placed in this location due to un-modeled constraints, the designer does not retrieve any other useful information. Conversely, the map generation approach offers a complete view of the dependence between energy consumption, cycle time and robot position. Computational time for the map generation is in general higher than that required by the optimization algorithm, and it is strongly related to the desired accuracy (chosen by the user through the starting grid density). The map in Fig. 10 is very detailed (it is composed of 90.000) points and required approximately 2 h of computation on a standard PC. The optimization of the same problem was solved in less than 2 min leading to the same energetic optimal point. In practice, computational time can be usually reduced bounding the research area in smaller zone and asking for less detailed maps. The practical applicability of these approaches is validated in the following section on the basis of numerical simulations developed on an industrial case study.

4. Industrial application

Typically, in automotive production lines, a large number of robots are employed in various areas, e.g. for welding, assembly, gluing, painting, etc. The energy consumption of these robots heavily impacts the overall factory sustainability [1] and an optimization could save big amount of energy reducing the costs and the environmental impact of the plant. In this section, two generic projects of robotic stations are analyzed and optimized founding the optimal position for three industrial robots executing spot welding and pick-and-place operations.

4.1. Case studies overview

Original Delmia projects of the two considered robotic manufacturing cells are shown in Figs. 11 and 12: in the first picture, the robot (hereafter called *Robot 1*) is used for spot welding on a body side outer panel; in the second picture, the robot on the right (*Robot 2*) picks a sheetmetal part from a conveyor and places it on the chassis, while the other robot (*Robot 3*) performs the spot welding. All robots are KUKA KR210 R3100 ultra [27], chosen for their wide spread use in automotive production lines. The robots can be placed within the light blue areas. The yellow reference frames are the absolute stations reference frames. *Robot 1* and *Robot 3* mounts two different spot

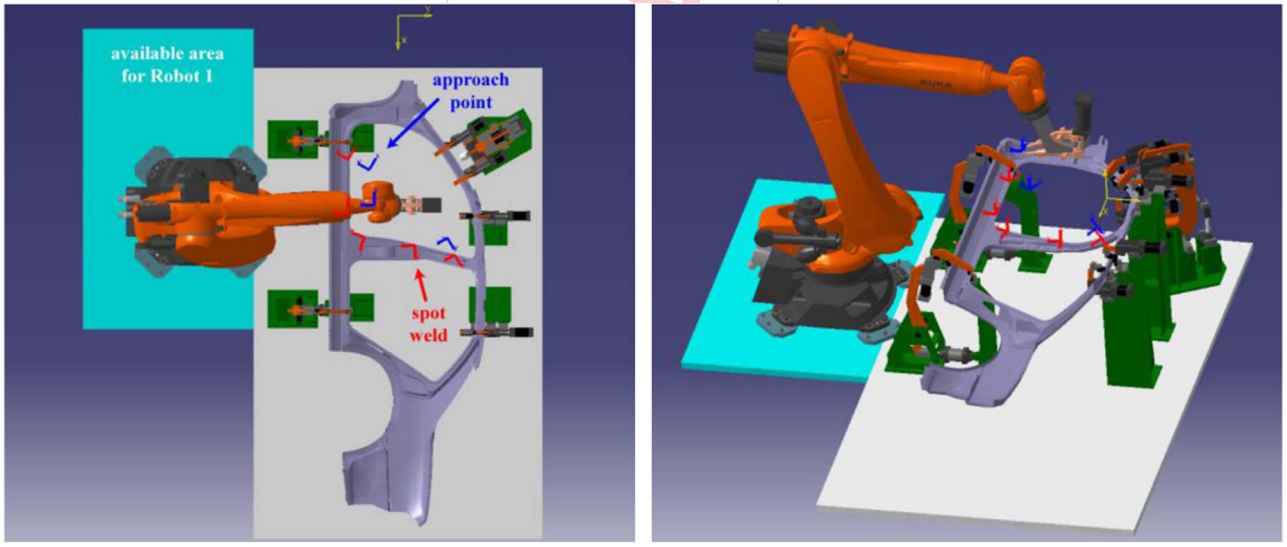

Fig. 11. Top-view and 3D-view of the first robotic station.

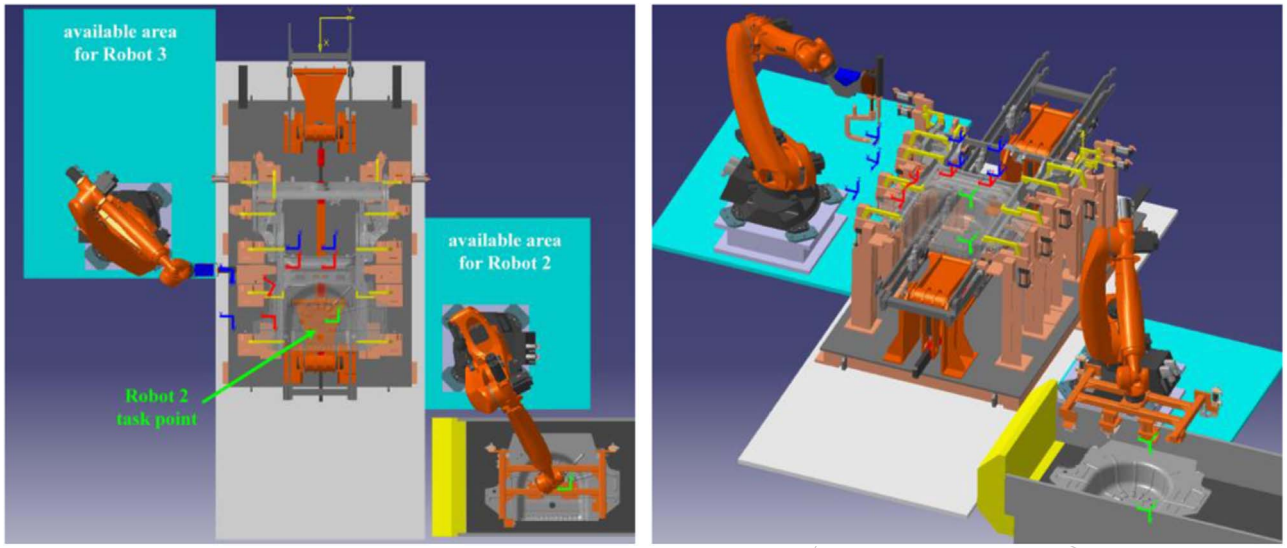

Fig. 12. Top-view and 3D-view of the second robotic station.

Table 1
Geometrical and dynamical parameters of the robots end-effectors.

	Robot 1	Robot 2	Robot 3
Tool frame position (x, y, z) [mm]	(220, 0, 263)	(105, −210, 410)	(0, 0, 835)
Tool frame orientation (φ, θ, ψ) [°]	(−160, 0, 90)	(0, 0, 90)	(−180, 40, 0)
Mass [kg]	100	150	140
Centre of mass (x, y, z) [mm]	(−50, 0, 110)	(0, 0, 200)	(−100, 0, 200)

welding guns, whereas *Robot 2* is equipped with a custom designed gripper. Geometrical and dynamical parameters of the three end effectors are reported in Table 1.

Since real robot programs are usually very complex, with thousands of motion instructions, for clarity purposes simplified programs are considered in this paper and explained hereafter. *Robot 1* and *Robot 3* are programmed for the spot welding, whereas *Robot 2* for the pick-and-place operation. The three programs are summarized in Table 2, Tables 3, 4 as lists of motions commands. The tables report: position and orientation of each target frame expressed referring to the station

Table 2
Robot 1 task.

Target Number	Position (x, y, z) [mm]	Orientation (φ, θ, ψ) [deg.]	Motion Type	Velocity	Pause time [s]
1	(1450, -180, 1300)	(0, 0, -180)	/	/	/
2	(1700, 375, 900)	(0, 30, -60)	Joint	50 %	0
3	(1830, 430, 900)	(0, 30, -60)	Linear	0.3 m/s	2.1
4	(1755, 125, 795)	(0, 0, -85)	Linear	0.3 m/s	2.1
5	(1695, -255, 800)	(0, 0, -110)	Linear	0.3 m/s	2.1
6	(1515, -380, 815)	(0, 0, -180)	Linear	0.3 m/s	2.1
7	(1100, -380, 815)	(0, 0, 150)	Linear	0.3 m/s	2.1
8	(1180, -220, 815)	(0, 0, 150)	Linear	0.3 m/s	2.1
1	(1450, -180, 1300)	(0, 0, -180)	Joint	50 %	0

Table 3
Robot 2 task.

Target Number	Position (x, y, z) [mm]	Orientation (φ, θ, ψ) [deg.]	Motion Type	Velocity	Pause time [s]
1	(5420, 2875, 1015)	(180, 0, −90)	/	/	/
2	(5420, 2875, 185)	(180, 0, −90)	Linear	0.3 m/s	1
1	(5420, 2875, 1015)	(180, 0, −90)	Linear	1 m/s	0
3	(3500, 210, 1930)	(180, 0, −90)	Joint	50 %	0
4	(3500, 210, 1385)	(180, 0, −90)	Linear	0.3 m/s	1
3	(3500, 210, 1930)	(180, 0, −90)	Linear	1 m/s	0
1	(5420, 2875, 1015)	(180, 0, −90)	Joint	100 %	0

absolute frame; type and velocity of the motion for reaching the target; the pause time during which the robot is stopped on the target for performing required operations (spot weld, pick or place), if any. Joint type and higher velocities are used for collision-safe motions, whereas linear type and lower velocities for risky motions requiring a fixed Cartesian path. Each robot starts and ends the cycle with its end-effector tool frame coincident with the first target, placed in a safety position permitting the product displacement without collisions. Figs. 11 and 12 show the robots on their initial configurations; blue and red frames indicate respectively approach and welding points defining the task for *Robot 1* and *3*; green frames indicates *Robot 2* task targets.

4.2. Optimization and results

The goal is to find the best robots layout position reducing the energy consumption required to accomplish the assigned tasks without any modification. The robot layout position is defined with the position of the robot base frame, with respect to the absolute station reference frame. As a common design practice, the robot base is imposed to be parallel to the floor, i.e. the first joint axis is always parallel to the gravity vector. This assumption is commonly adopted since industrial robots are usually optimized for this configuration and since inclined mounting plates are considered unjustified extra-costs. Moreover, the remaining possible rotation of the robot base around the z axis does not

Table 4
Robot 3 task.

Target Number	Position (x, y, z) [mm]	Orientation (φ, θ, ψ) [deg.]	Motion Type	Velocity	Pause time [s]
1	(2900, −1000, 1900)	(0, 0, −90)	/	/	/
2	(3430, −1000, 1600)	(0, 0, −90)	Joint	50 %	0
3	(3430, −520, 1630)	(0, 0, −90)	Linear	0.3 m/s	1
2	(3430, −1000, 1600)	(0, 0, −90)	Linear	0.3 m/s	0
4	(2900, −1000, 1600)	(0, 0, −100)	Linear	0.5 m/s	0
5	(3070, −520, 1630)	(0, 0, −120)	Linear	0.3 m/s	1
4	(2900, −1000, 1600)	(0, 0, −100)	Linear	0.3 m/s	0
1	(2900, −1000, 1900)	(0, 0, −90)	Joint	50 %	0
6	(2650, −240, 1900)	(0, 0, −180)	Joint	50 %	0
7	(2650, −240, 1600)	(0, 0, −180)	Linear	0.5 m/s	0
8	(2870, −240, 1630)	(0, 0, −180)	Linear	0.3 m/s	1
7	(2650, −240, 1600)	(0, 0, −180)	Linear	0.3 m/s	0
9	(2650, 170, 1600)	(0, 0, −180)	Linear	0.5 m/s	0
10	(2870, 170, 1630)	(0, 0, −180)	Linear	0.3 m/s	1
9	(2650, 170, 1600)	(0, 0, −180)	Linear	0.3 m/s	0
11	(2650, 170, 1900)	(0, 0, −180)	Linear	0.5 m/s	0
1	(2900, −1000, 1900)	(0, 0, −90)	Joint	50 %	0

Table 5
Vertices of rectangular prisms defining the robots bases permissible volumes.

	Robot 1	Robot 2	Robot 3
Prism vertex 1 (x, y, z) [mm]	(500, −2000, 0)	(2800, 1700, 0)	(500, −2700, 0)
Prism vertex 2 (x, y, z) [mm]	(2000, −1500, 600)	(4000, 2600, 1000)	(2500, −1700, 1000)

influences the energy consumption. These assumptions reduce the dimensionality of the problems from six to three: the optimal robots locations can be searched in the three-dimensional available Cartesian space. Considering the size of the robots bases (1 m×1 m) and imposing limits at possible elevations, the volumes in which the robots base frames can be placed and where to search for the optimal locations are restricted to rectangular prisms defined in Table 5 through their vertices.

From a fast preliminary analysis of the considered case studies, one can easily see that they are well suited to be solved using the optimization tool, since it should be possible to place the robots everywhere in the available volumes without collisions risks: the only solution retrieved by the optimization tool should be really implementable without problems. Nevertheless, both proposed tools (optimization and map generation) have been used to highlight their potentialities. Table 6 summarizes optimizations results, whereas Fig. 13 depicts the EC maps generated at different heights. Results from the two tools are congruent, highlighting the existence of energy-optimal positions for the three robots with significant energy savings potentials. Maps are obtained executing the simulations on 100 mm spaced grids, a suitable choice for the state of the art industrial practice, which allows to obtain good results in just few minutes of computation. The same level of detail was required for the optimization algorithm, which provides significant results almost instantly (< 5 s).

Table 6
Optimizations results.

	Robot 1	Robot 2	Robot 3
Initial Position (x, y, z) [mm]	(1500, −1500, 0)	(3800, 1900, 400)	(2400, −2200, 400)
Optimal Position (x, y, z) [mm]	(500, −1500, 500)	(2800, 2600, 1000)	(900, −2000, 1000)
Initial Energy	19,053 J	27,821 J	30,206 J
Optimal Energy	17,777 J	22,223 J	23,674 J
Energy reduction	−1276 J (−6.7%)	−5598 J (−20.1%)	−6532 J (−21.6%)

4.3. Results validation

Due to the very high costs related with a layout change during production, which practically inhibit any reliable experimental validation on real cells, the tool has been validated following the latest industrial best practices. As previously said, robotic cells designers in industry currently adopt robot offline simulation software compliant with the RRS standard in order to retrieve reliable and trusted data on trajectories and cycle times of robotic processes. Recently, due to the growing interest on energy consumption, RRS has been extended for providing also information related to the robot energy consumption, calculated by robot manufacturers' proprietary models embedded into RCS modules, generally named "RCS Energy". Dassault Systemes Delmia Robotics and the KUKA Roboter RCS Energy module have been extensively tested and validated by both system integrators and final users of the automotive robotic manufacturing industry, fully satisfying industrial requirements. In any case, as previously mentioned, RCS modules are very accurate but computationally inefficient, so that they could not be effectively used for optimization tasks, but only for result validation. To validate the results, state of the art industrial practice has been adopted: two simulations for each robot have been executed in Delmia, with robots on their initial (Figs. 11, 12) and optimal (Fig. 14) positions proving the absence of collisions. During the simulations, the energy consumption computed by the RCS module has been monitored and then exported. Fig. 15 compares initial and optimal power curves for each robot, whereas Table 7 reports total energy values. As it can be seen, energy values obtained with the KUKA RCS module are pretty closed to values obtained with our model (Table 6) confirming the validity and efficacy of the tools proposed in this paper.

5. Conclusions

The paper proposes a novel engineering method for reducing the energy consumption of robotic cells, achieved by optimizing the layout position of industrial robots in relation to the assigned tasks. An accurate object-oriented IR model with higher computational efficiency has been developed using the *Modelica* language in *Dymola* environment. All mechatronic system components influencing the power flow from the AC mains to the mechanical structure are defined using simple but effective models. The model has been implemented into an innovative and flexible engineering tool, able to quickly provide the optimal robot base location computed through an optimization process.

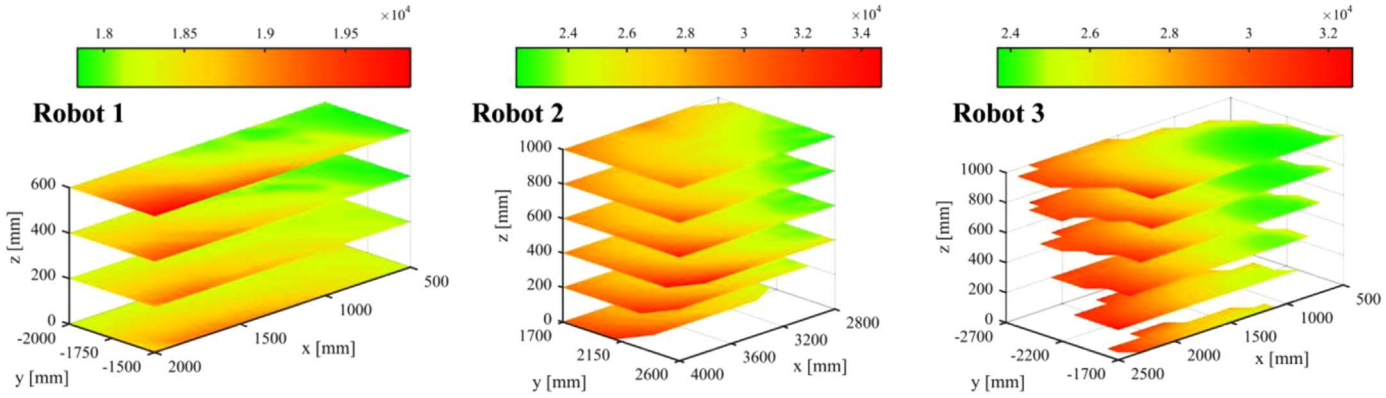

Fig. 13. Energy maps computed at different quotes.

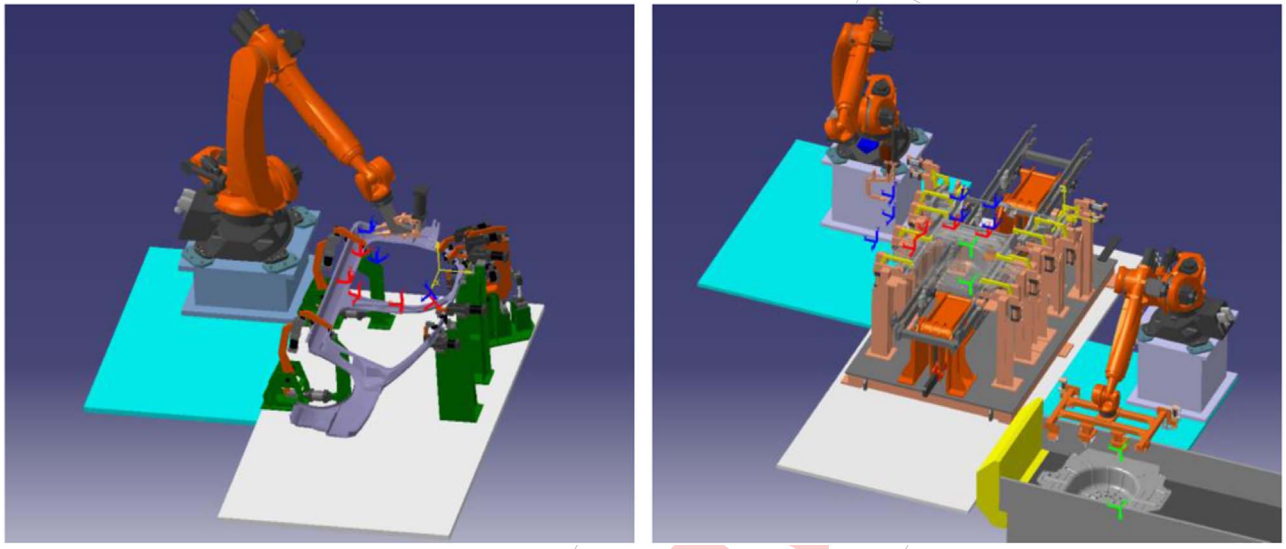

Fig. 14. Stations with robots on their energy-optimal positions.

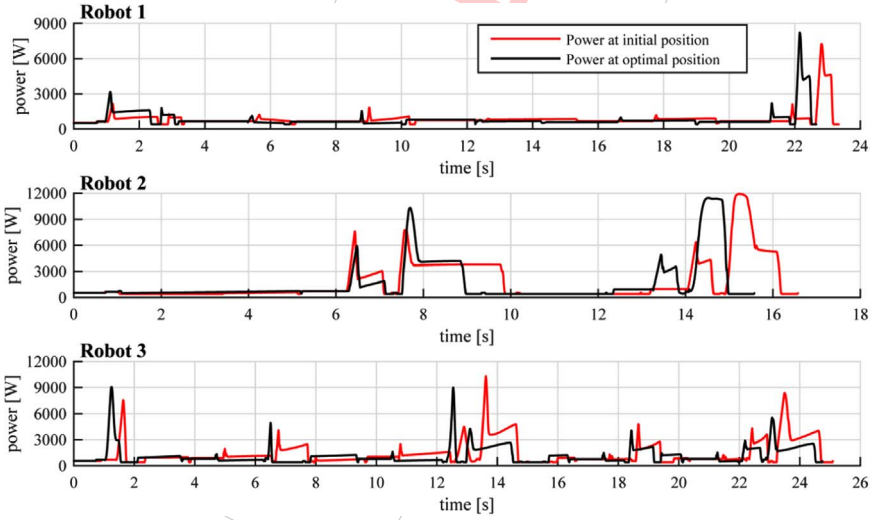

Fig. 15. Robots power consumption on their initial and optimal positions.

Furthermore, a set of color/contour maps depicts the energy demand along with the time required for the task completion in function of the robot base location. Following the industrial requirements, the proposed approach has been numerically tested, optimizing the base position of three robots working in general industrial stations. By placing the robots on the positions computed by the tool allows to achieve a reduction of the energy consumption in the order of 6÷20% as compared to that of the initial positions firstly proposed by experienced design engineers. Moreover, generated energy-maps evidently show very hard-to-predict relations between robot position and energy consumption, highlighting the benefits provided by this engineering tool, namely a quick and clear understanding of the best

Table 7
Energy consumption computed with KUKA RCS module.

	Robot 1	Robot 2	Robot 3
Initial Energy	19,442 J	29,343 J	35,507 J
Optimal Energy	18,156 J	24,532 J	28,285 J
Energy reduction	−1286 J (**−6.6%**)	−4811 J (**−16.4%**)	−7222 J (**−20.3%**)

areas in which the robot should be placed, that could be easily integrated in current robotic cell design practices. With the proposed engineering tool and method, design engineers can now address also the energy efficiency and sustainability of the robotic cell layout.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No. 609391 (AREUS). The authors would like to thank AREUS Project partners DAIMLER AG, KUKA Roboter, SIR Spa and EngRoTec Consulting for their support.

References

[1] D. Meike, Increasing Energy Efficiency of Robotized Production Systems in Automobile Manufacturing (Ph.D Thesis), Riga Technical University, 2013.
[2] P. Waide, C. Brunner, Energy-efficiency policy opportunities for electric motor-driven systems, Int. Energy Agency (2011) 1–132 (Available online).
[3] M. Pellicciari, A. Avotins, K. Bengtsson, G. Berselli, N. Bey, B. Lennartson, D. Meike, AREUS – innovative hardware and software for sustainable industrial robotics, in: Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), 2015,pp. 1325–1332.
[4] R. Saidur, A review on electrical motors energy use and energy savings, Renew. Sust. Energy Rev. 14 (2010) 877–898.
[5] O. Maimon, E. Profeta, S. Singer, Energy analysis of robot task motions, J. Intell. Robot Syst. 4 (1991) 175–198.
[6] E. Sergaki, G. Stavrakakis, A. Pouliezos, Optimal robot speed trajectory by minimization of the actuator motor electromechanical losses, J. Intell. Robot. Syst. 33 (2002) 187–207.
[7] M. Pellicciari, G. Berselli, F. Leali, A. Vergnano, A method for reducing the energy consumption of pick-and-place industrial robots, Mechatronics 23 (3) (2013) 326–334.
[8] B. Nelson, K. Pederson, M. Donath, Locating assembly tasks in a manipulator's workspace, in: Proceedings of the 1987 IEEE International Conference on Robotics and Automation, vol.4, 1987, pp. 1367–1372.
[9] B. Fardanesh, J.Rastegar, Minimum cycle time location of a task in the workspace of a robot arm, in: Proceedings of the 27th IEEE Conference on Decision and Control, vol.3, 1988, pp. 2280–2283.
[10] J. T. Feddema, Kinematically optimal robot placement for minimum time coordinated motion, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, vol. 4, 1996, pp. 3395–3400.
[11] P. Chedmail, P. Wenger, Design and positioning of a robot in an environment with obstacles using optimal research, in: Proceedings of the 1989 IEEE International Conference on Robotics and Automation, vol. 2, 1989, pp. 1069–1074.
[12] R. Ur-Rehman, S. Caro, D. Chablat, P. Wenger, Path placement optimization of manipulators based on energy consumption: application to the orthoglide 3-axis, Trans. Can. Soc. Mech. Eng. (2009).
[13] M. Gadaleta, G. Berselli, M. Pellicciari, A. O. Andrisano, Towards Energy-Optimal Layout Design of Robotic Work Cells, in: Proceedings of the International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), June 2015, pp. 3–26.
[14] H. Elmqvist, S. E. Mattsson, M. Otter, Modelica – a language for physical system modeling, visualization and interaction, in: Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design, Hawaii, USA, 1999
[15] A. Kazi, G. Merk, M. Otter, H.Fan, Design optimisation of industrial robots using the Modelica multi-physics modeling language, in: Proceedings of the 33rd ISR (International Symposium on Robotics), 2002
[16] ⟨http://www.3ds.com/products-services/catia/products/dymola⟩ (accessed 15.11.15).
[17] ⟨http://www.modelica.org/⟩ (accessed 15.11.15).
[18] ⟨http://www.3ds.com/products-services/delmia/⟩ (accessed 15.01.15).
[19] B.K. Bose, Modern Power Electronics and AC Drives, Prentice-Hall, Englewood Cliffs, NJ, USA, 2002.
[20] E. Oliva, G. Berselli, M. Pellicciari, A.O. Andrisano, An engineering method for the power flow assessment in servo-actuated automated machinery: mechatronic modeling and experimental evaluation, Robot. Comput.-Integr. Manuf. 38 (2016) 31–41.
[21] P. Hamon, M. Gautier, P. Garrec, A. Janot, Dynamic modeling and identification of joint drive with load-dependent friction model, in: Proceedings of the 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2010, pp. 902–907.
[22] F. Fernandez-Bernal, A. Garcia-Cerrada, R. Faure, Determination of parameters in interior permanent-magnet synchronous motors with iron losses without torque measurement, IEEE Trans. Ind. Appl. 37 (5) (2001) 1265–1272.
[23] D. Meike, M. Pellicciari, G. Berselli, Energy efficient use of multi-robot production lines in the automotive industry: detailed system modeling and optimization, Autom. Sci. Eng. IEEE Trans. 11 (3) (2014) 798–809.
[24] D. Meike, L. Ribickis, Recuperated energy savings potential and approaches in industrial robotics, in: Proceedings of the IEEE Conference on Automation Science and Engineering (CASE), 2011, pp. 299–303.
[25] P. Fritzson, Principles of Object-oriented Modeling and Simulation with Modelica 2.1, John Wiley & Sons, 2010.
[26] ⟨http://www.realistic-robot-simulation.org/⟩ (accessed 20.11.15).
[27] KUKA, KR QUANTEC ultra Specification, 2013.