8th IEEE International Conference on Automation Science and Engineering
August 20-24, 2012, Seoul, Korea

Increasing the Energy Efficiency of Multi-robot Production Lines in the Automotive Industry

Davis Meike, Marcello Pellicciari, Giovanni Berselli, Alberto Vergnano, Leonids Ribickis

Abstract— **This paper quantitatively reports about potential energy savings on robotic assembly lines for the automotive industry. The key aspect of the proposed approach is that both cell production rate and robot hardware limitations are considered as strict constraints, so that no plant revision is needed. The methodology relies on: a) calculation of energy-optimal trajectories, by means of time scaling, concerning the robots' motion from the last process point to the home positions; b) reduction of the energy consumption via earlier release of the actuator brake when the robots are kept stationary. Simulation results are presented, which are based on the production timing characteristics measured on a real plant.**

Index Terms— **Industrial Robots, Energy Efficiency, Trajectory Scaling, Production Planning.**

I. Introduction

Energy efficiency is becoming a topic of primary importance in industrial robotics due to either immediate potential savings or forthcoming Energy Consumption (EC) standards [1]. Focusing on automobile manufacturing, which is typically highly automated with robot systems, it is interesting to note that 15-28% of a vehicle's EC during its overall lifecycle occurs within the production phase, whereas the electrical energy consumed by the Industrial Robots (IRs) on the assembly lines amounts to about 8% on average. Therefore, as previously highlighted in [2], it is self-evident that EC minimization in existing automated manufacturing systems impacts both production costs and total CO_2 emissions.

Concerning EC minimization in robotics, most of the past literature describes effective methodologies that rely on either efficient equipment selection/design (e.g. [3]) or path re-planning (e.g. [4]), thus involving considerable system modifications. On the other hand, when a robotic production system is in its mature lifecycle phase (i.e. it is already technologically optimized), it is desirable to reduce the EC while avoiding any substantial revision. Within this scenario, previous work by the authors [5] reports about energy-optimal scheduling of robotic cells by means of trajectory time scaling, the cell production rate and the hardware limitations being considered as strict constraints. In particular, the tool center point (TCP) position profile is assumed as given since any modified path might be practically unfeasible due to the well-established process constraints or intrinsic limitations of the industrial controllers.

Following the same direction, this paper reports about a case study applied on a industrial robot cell of a vehicle body-shop production line. The cell is composed of four anthropomorphic IRs (210kg payload) with unshared workspace and employed in spot-welding or handling applications. On the basis of measured data (i.e. process schedule/constraints), two saving effects are simulated, namely: a) time scaling concerning the robot's motion from the last process point to the robot's home position; b) reduced release times concerning the robot mechanical brakes.

The paper is organized as follows: Sec. II-III extend previously published results [5] and report an EC model for a single IR. The model captures the EC dependency on release time of the mechanical brakes and task execution times. In Sec. IV, the proposed saving method is applied on the mentioned multi-robot cell, the EC values being now evaluated via proprietary software [6]. Results confirm that noticeable energy savings are possible without any investments on new equipment or negative consequences on the cell production rate.

II. Energy Consumption Of Industrial Robots

A. Manipulator model

A typical 6-dof anthropomorphic IR is actuated by six Permanent Magnet (PM) synchronous machines.During the robot unconstrained motion, the vector of actuation torques, $\boldsymbol{\tau} \in \mathbb{R}^6$, can be expressed as the sum of the following terms:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_I(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \mathbf{D}\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \qquad (1)$$

where $\boldsymbol{\tau}_I$, $\mathbf{G}$ and $\mathbf{D}\dot{\mathbf{q}}$ respectively represent the portions of actuator torques that are used to counteract the manipulator own inertia, the manipulator own weight and the viscous frictions, whereas $\mathbf{q}(t)$ is the vector of joint positions as a function of time t. The matrix $\mathbf{D}$ is a constant diagonal matrix of friction coefficients whereas, for each actuator, the rotor inertia is accounted for in the term $\boldsymbol{\tau}_I$. As for the PM machines' dynamics, it can be described by

$$\mathbf{v} = \mathbf{R}\mathbf{i} + \mathbf{K}_v\mathbf{K}_r\dot{\mathbf{q}}; \qquad \mathbf{i} = \mathbf{K}_T^{-1}\mathbf{K}_r^{-1}\boldsymbol{\tau} \qquad (2)$$

where, for $j = 1...6$, $\mathbf{v} = [v_j]$ and $\mathbf{i} = [i_j]$ are column vectors of the supplied armature voltages and currents respectively, $\mathbf{R} = diag\{R_j\}$, $\mathbf{K}_v = diag\{K_{v,j}\}$, $\mathbf{K}_T = diag\{K_{T,j}\}$ and $\mathbf{K}_r = diag\{K_{r,j}\}$ are constant diagonal matrices of the different armature electric resistances, gear ratios, back emf constants and motor torque constants respectively ($K_{T,j}/K_{v,j} \approx 1$ for trapezoidal type AC brushless [4]). From Eq. 2, the overall

D. Meike is with Daimler AG, Sindelfingen, Germany, e-mail: davis.meike@daimler.com; M. Pellicciari, G. Berselli and A. Vergnano are with the University of Modena and Reggio Emilia, Modena, Italy, e-mail: name.surname@unimore.it; L. Ribickis is with the Riga Technical University, Riga, Latvia, e-mail: leonids.ribickis@rtu.lv.

978-1-4673-0430-6/12/$31.00 ©2012 IEEE

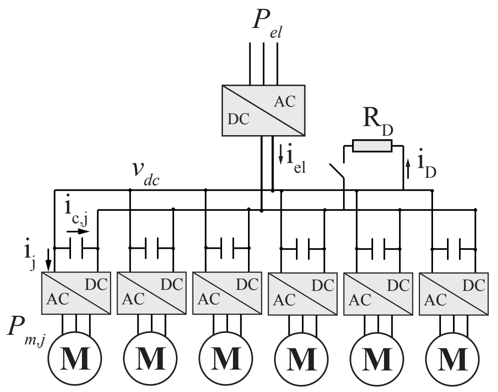


Fig. 1. Schematic of a multidrive system for industrial applications.

instantaneous power delivered to the motors, P_m, is given by

$$P_m = \sum_{j=1}^{6} P_{m,j} = \mathbf{i}^T \mathbf{v} = \boldsymbol{\tau}^T \overline{\mathbf{R}} \boldsymbol{\tau} + \boldsymbol{\tau}^T \dot{\mathbf{q}} \tag{3}$$

where $P_{m,j}$ is the power requested by the single motor and $\overline{\mathbf{R}} = \mathbf{R}\left(\mathbf{K}_r \mathbf{K}_T\right)^{-2}$.

B. Electrical drive model

A conceptual scheme of a multidrive servo-system comprising the six PM machines and the power converters (AC/DC module + DC link + DC/AC modules) is depicted in Fig. 1. Normally, the energy flow towards the AC network is undesired [7]. On the other hand, even if the electrical power flow is unidirectional (i.e. $P_{el} \geq 0$), most multidrive systems allow for motor-to-motor energy exchange within the common DC-bus. However, when several axes brake simultaneously, the DC-bus voltage increases above the rectified AC supply voltage (commonly $\sim 565V$) and excessive over-voltages are dissipated in a drain resistor, R_D. For instance, past measurements [2] highlight possible peaks of $\sim 690V$ which are lowered by means of R_D.

In particular, with reference to Fig. 1, considering the current flow from the AC/DC converter, i_{el}, to the resistor, i_D, to each DC/AC module, i_j, and to each capacitor, $i_{c,j}$, the current balance equations can be written as

$$i_{el} = \sum_{j=1}^{6} i_j + \overline{C}\frac{dv_{dc}}{dt} + \alpha i_D \tag{4}$$

where v_{dc} is the voltage over the DC-link, C_i is the capacitance connected to each DC/AC module, $\overline{C} = \sum_{j=1}^{6} C_i$ is the overall DC-Bus capacitance, and α is either 0 or 1 for activated/deactivated drain resistor.

If any sort of energy loss within the DC/AC modules is neglected, the overall power delivered to the motors is simply $P_m = v_{dc} \sum_{j=1}^{6} i_j$. In addition, let now assume that the AC/DC converter is an ideal full bridge rectifier (i.e. no energy losses, $P_{el} = i_{el} v_{dc}$) and let $\overline{v}_{dc}$ be the average voltage value over the DC-bus when the motors operate at constant velocity (i.e. when $\ddot{\mathbf{q}} = 0$). Resorting to Eq. 4, the power flow equation for the multidrive system can be written as

$$P_{el} - P_m = \overline{C} v_{dc} \frac{dv_{dc}}{dt} - \alpha v_{dc}^2 / R_D \tag{5}$$

Finally, the power flow from the network is given by

$$P_{el} = \begin{cases} P_m, & \text{if } v_{dc} = \overline{v}_{dc} \text{ AND } P_{act} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The overall model block diagram is depicted in Fig. 2.In practice, considering the rectifier as an ideal voltage source, if the motors are accelerating, the required amount of power is instantaneously taken from the network. On the other hand, when the motors are decelerating, the power backflows from the motors causes the DC-bus voltage to increase. In this case, useful energy is stored in the capacitors and $P_{el} = 0$. Finally, if v_{dc} exceeds a certain over-voltage threshold v_{max}, the drain resistor is activated ($\alpha = 1$) and some of the power backflow is dissipated for Joule effect. In case the value R_D is unknown, the drain resistor contribution may be modeled via a saturation block with lower and upper limits equalling $\overline{v}_{dc}$ and v_{max} respectively.
As a qualitative example, Fig. 3(a) depicts the values of P_{el} and P_m for a typical robot operation. The simulation refers to an unloaded anthropomorphic arm (model parameters from [8]) performing a semicircular path in the cartesian space. Power measures have been normalized with respect to the peak actuation power, P_{max}, whereas times have been normalized with respect to the reference time, t_h.

C. Influence of the mechanical brakes

The IR motors are usually equipped with six normally-closed mechanical brakes, which are synchronously released/unreleased [2] and which require an overall power, $P_b = \sum_{j=1}^{6} P_{b,j}$, to be kept open ($P_{b,j}$ being the power required by the single brake). The overall power required by the actuation system is then given by

$$P_{act} = \gamma(P_{el} + P_b) \tag{7}$$

where γ is either 0 or 1 for released or unreleased brakes respectively. In particular, when the robot is stationary, P_{act} reduces to

$$P_{act}^* = \gamma\left(\mathbf{G}^T \overline{\mathbf{R}} \mathbf{G} + P_b\right) \tag{8}$$

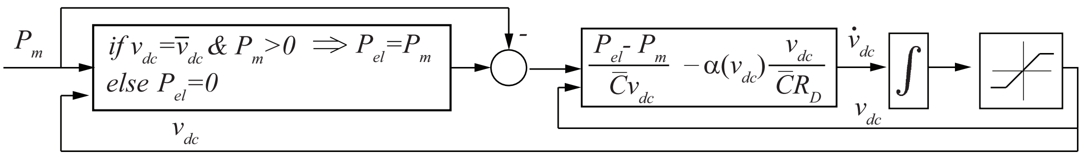
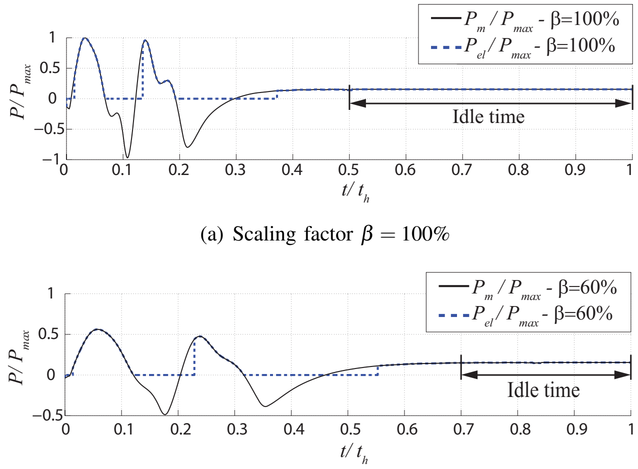

P_m
$if\, v_{dc}=\overline{v}_{dc}\ \&\ P_m>0 \Rightarrow P_{el}=P_m$
$else\ P_{el}=0$
-
$\frac{P_{el}-P_m}{\overline{C}v_{dc}} - \alpha(v_{dc})\frac{v_{dc}}{\overline{C}R_D}$
$\dot{v}_{dc}$
$\int$
v_{dc}
v_{dc}

Fig. 2. Block diagram of the simplified drive dynamics.



(a) Scaling factor $\beta = 100\%$

(b) Scaling factor $\beta = 60\%$

Fig. 3. Normalized actuation power, P_{act}/P_{max} and electrical power, P_{el}/P_{max} for different scaling factors.

and power is used to simply counteract gravity loads (see the idle time in Fig. 3(a)). Equation 8 trivially shows that an early brakes' release is always beneficial in terms of power consumption. Nonetheless, this action significantly increases the brake switching cycles, the consequences of which have to be considered in real plants [2].

III. Trajectory Time Scaling Applied To Energy Consumption Minimization

With reference to a given time interval $t \in [0, t_h]$, an IR is usually assumed to operate at its maximum speed whenever allowed by the scheduling/technological constraints, and stand still otherwise (idle times). However, this reference trajectory, denoted as $\mathbf{q}_O(t), \dot{\mathbf{q}}_O(t), \ddot{\mathbf{q}}_O(t)$[1], may not be energy-optimal due to high power-consuming accelerations and energy wasted to counteract gravity (if mechanical brakes are opened).

Similarly to [5], [9], let alternatively assume that the aforementioned reference path is followed with a trajectory whose position profile is given by:

$$\mathbf{q}(t) = \mathbf{q}_O(\beta t) = \mathbf{q}_O(t') \tag{9}$$

where $t' = \beta t$ and $\beta \leq 1$ are denoted as *scaled time* and *scaling factor* respectively. The scaled velocities and accelerations can be expressed as function of the parameter β and of time t, that is

$$\begin{aligned} \dot{\mathbf{q}}(\beta,t) &= \frac{d\mathbf{q}_O(t')}{\beta^{-1}dt'} = \beta\dot{\mathbf{q}}_O(t') \\ \ddot{\mathbf{q}}(\beta,t) &= \frac{d(\beta\mathbf{q}_O(t'))}{\beta^{-1}dt'} = \beta^2\ddot{\mathbf{q}}_O(t') \end{aligned} \tag{10}$$

In the same manner, recalling the notation defined in Eq. 1, the scaled torques as function of β and t, are given by:

$$\boldsymbol{\tau}(\beta,t) = \beta^2\boldsymbol{\tau}_O(t') + \beta\mathbf{D}\dot{\mathbf{q}}_O(t') + \mathbf{G}_O(t') \tag{11}$$

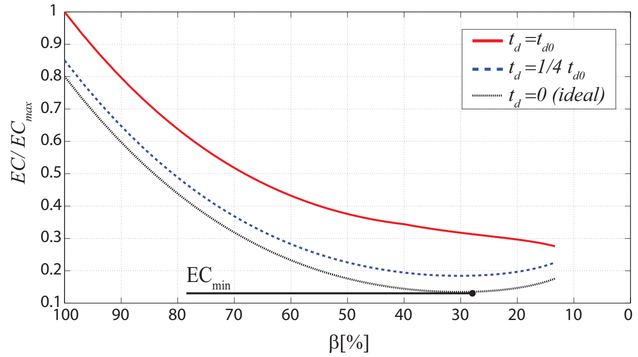


Fig. 4. Energy Consumption as a function of scaling factor and release time delay of the mechanical brakes.

Finally, the scaled actuation power is given by

$$P_{act}(\beta,t) = \gamma\left[\boldsymbol{\tau}^T(\beta,t)\cdot\overline{\mathbf{R}}\cdot\boldsymbol{\tau}(\beta,t) + \beta\boldsymbol{\tau}^T(\beta,t)\cdot\dot{\mathbf{q}}(\beta,t) + P_b\right] \tag{12}$$

which reduces to Eq. 3 for $\beta = 1$. This instantaneous power is fed into the model depicted in Fig. 2 to compute the parametric electrical power for the scaled trajectory, $P_{el}(\beta,t)$. Finally, for each β value, the overall EC (including possible standstill electrical power, P_{el}^*) can be numerically computed as:

$$E(\beta) = \int_0^{t_h} P_{el}(\beta,t)dt \tag{13}$$

Refereing to the case study discussed in Sec. IIb, Fig. 3(b) reports the values of P_{el} and P_m for a scaled trajectory (i.e. $\beta = 60\%$). In addition, Fig. 4 reports normalized EC plots in terms of the scaling factor (EC measures have been normalized with respect to the peak EC, E_{max}).Three release time delays for the mechanical brakes are considered, namely $t_d = 0s$, $t_d = 1/4 t_{d0}$ and $t_d = t_{d0}$, where t_{d0} is the state-of-the-art brake release delay. The condition $t_d = 0s$ models a purely ideal case where the brakes are instantaneously released at the end of the robot motion. For all the β values, the figure highlights that the EC decreases by decreasing the brake release delay. In addition, for a given t_d value, there might exist an EC minimum for some intermediate scaling factor which can be found by simulation once the model parameters are known.

IV. Minimization Of Energy Consumption On The Automobile Production Line

The described approach has been applied to a real robot cell of the Mercedes-Benz plant in Sindelfingen, Germany. The EC has been simulated using proprietary software [6], the input trajectories (reference as well as scaled ones) being generated using the Realistic Controller Simulator (RCS) module and the interpreter of the manufacturer's programming language (KRL). As for the robot programs, they are partly scaled, affecting only the movement from the last process point to the standard waiting (HOME) position. Finally, the timing characteristics of the robot cell are experimentally measured. In the following, the value $t_{x,i,k}$ refers to the time period x concerning the i-th process

[1] In the following, all quantities referring to this reference trajectory will be denoted with O subscript.

performed by the k-th robot, whereas $t_{x,k}$ is the average value over $i = z$ processes and t_x is the average value over $i = z$ processes and $k = n$ robots.

A. Analysis of the multi-robot cell

As a case study, a cell consisting of $n = 4$ IRs without overlapping workspaces has been selected. The process related to robot R_1 is the one that imposes the global cycle time within the cell, whereas the processes related to robots $R_{2\ldots4}$ are completed in a shorter time span. Hence, denoting $t_{c,i,k}$ as the single robot cycle time, the global cycle for the given process is $t_{gc,i} = t_{c,i,1}$. In particular, Fig. 5 represents the various states of each robot R_k as a function of time (normalized with respect to $t_{gc,i}$). A nonzero value for an ordinate indicates that the state is present, the possible robot states being defined as follows:

- APPL_RUN: active at the start and at the end of the cycle time. In Fig. 5, the continuous line forms a block of duration $t_{c,i,k}$ for the particular robot's program, whereas the dashed line (refer, for instance, to robot R_2) represents the duration of the global cycle time minus t_t, i.e. the transfer duration of the car body between the robot cells (t_t proved to be constant over the measured processes).
- IN_HOME: active when the robot is in its default waiting (*HOME*) position;
- STOPPED: represents the state when the robot manipulator is at standstill, e.g. because of welding or waiting.

A zero value of the STOPPED bit allows the movement phases between process standstills to be identified. A time span between the falling edge of the IN_HOME block and the rising edge of the STOPPED block represents the duration of the movement from the default waiting position to the first process point. Vice versa, it is the movement from the process point to the default home position. If this phase is followed by a falling APPL_RUN, it has to be recognised as the last cycle movement of the robot R_k with duration $t_{h,i,k}$. Excluding all the process-related standstills, t_h is typically around 20% of the total movement duration [2].

A measurement has been performed over a weeks production of a total of $i = z$ cycles and $k = n$ robots of the selected cell, and the global average cycle length has been computed as:

$$t_{gc} = max\left(\frac{1}{z}\sum_{i=1}^{z} t_{c,i,1} \ldots \frac{1}{z}\sum_{i=1}^{z} t_{c,i,n}\right) + t_t \tag{14}$$

where, as said, the time $t_t = t_{c,i+1} - t_{c,i}$ proved to be constant. Due to confidentiality, the exact values are not revealed. However, to give the reader some clue, it can be assumed that $t_{gc} \approx 60s$. Also, the robot R_1 appears to be the *leading one*. However, that does not have to be the case for every cell. Next, the average process duration $t_{c,k}$ related to robot R_k and the average actual duration $t_{h,k}$ of the last movement has been statistically calculated. Since the robots do not share a common workspace, the available time $t_{ah,k}$ for the scaling of the last movement can be calculated as follows:

$$t_{ah,k} = t_{gc} - \left(t_{c,k} - t_{h,k}\right) \tag{15}$$

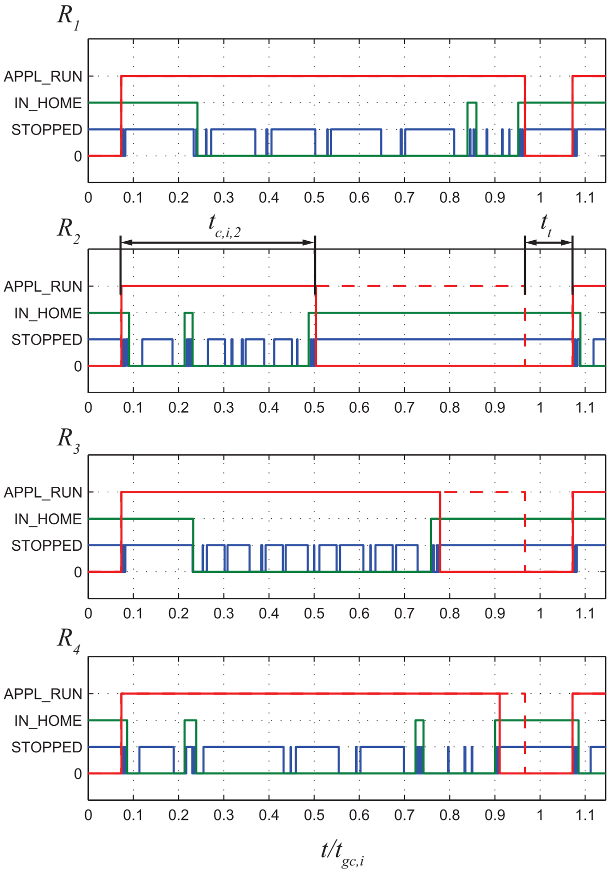


Fig. 5. Process states within the robot cell for the i-th process and for $R_{1\ldots4}$ robots (Note: bit states are differentiated by block heights)

The average scaling factor β_k of each robot R_k for the last movement is chosen within a range proportional to the available time, as:

$$\beta_k \in \left[\frac{t_{h,k}}{t_{ah,k}}, 100\%\right] \tag{16}$$

Hence, the duration of the unchanged and scaled part of the robot program, $t_{O,k}$ and $t_{s,k}$ respectively, are found as:

$$\begin{aligned} t_{O,k} &= t_{c,k} - t_h \\ t_{s,k} &= \beta_k t_{ah,k} \end{aligned} \tag{17}$$

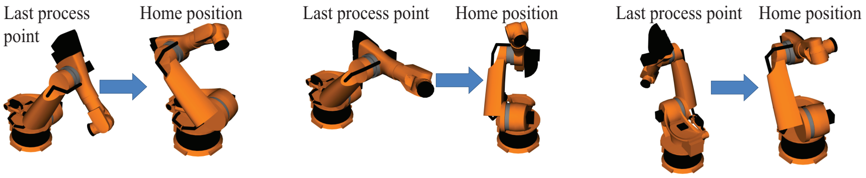
As already stated, the mechanical brakes of a typical IR are kept open during the robot's motion and close after a standstill delay time, t_d. Depending on this value, the waiting time in the home position after the process completion is:

$$t_{wait,k} = \begin{cases} t_d, & \text{for } t_{gc} - t_{c,k} \geq t_d \\ t_{gc} - t_{O,k} - t_{s,k}, & \text{for } t_{gc} - t_{c,k} < t_d. \end{cases} \tag{18}$$

As an example, Fig. 6 depicts the pose of robots $R_{2\ldots4}$ at the last process point and at the HOME position. This movement phase is the one to be optimized by means of time scaling.

B. Computation of the energy consumption within the cell

Defining β_k as the scaling factor applied to the k-th robot within the cell, the overall EC of the k-th robot can be split into four main components:



(a) Robot R_2 (b) Robot R_3 (c) Robot R_4

Fig. 6. Robot last and home positions concerning scaled movements.

- Energy required for the movement and process standstills of a non-scaled program part, that is

$$E_{O,k} = \int_{0}^{t_{O,k}} P_{el,k}(t)dt \tag{19}$$

- Energy required for the last movement phase. Recalling Eq. 13, this term can be split into:
  - Energy required for the scaled program part of the last movement, that is

$$E_{s,k} = \int_{t_{O,k}}^{t_{O,k}+t_{s,k}} P_{el,k}(\beta_k, t)\, dt \tag{20}$$

  - Energy required for the idle time after the program is finished but motors are still in operation, that is

$$E_{wait,k} = \int_{t_{O,k}+t_{s,k}}^{t_{O,k}+t_{s,k}+t_{wait,k}} P^{*}_{el,k}(\beta_k, t)dt \tag{21}$$

- Static loads like PC controller, control panel, controller board, cooling[2];

$$E_{st} = P_{st}\, t_{gc} \tag{22}$$

Assuming that all robots have the same type of control cabinets, the energy requirement for all n robots within the cell is:

$$E_{cell} = \sum_{k=1}^{n} \left(E_{O,k} + E_{s,k} + E_{wait,k}\right) + n\, E_{st} \tag{23}$$

As stated above, the two components $E_{wait,k}$ and $E_{s,k}$ are dependent on β_k, thus (23) can be expressed as a function of the scaling factors:

$$E_{cell} = f(\boldsymbol{\beta}) \tag{24}$$

where $\boldsymbol{\beta} = [\beta_k]$ is a vector of the scaling factors for the multi-robot cell. Refer to [10], [2] for typical EC values.

C. Optimization results

Three energy saving approaches have been tested:

1) A static reduction of the brake release time $t_d^{o1} = 1/4\, t_{d0}$, where t_{d0} is the state-of-the-art brake release delay.
2) Reduced brake release time $t_d^{o2} = 1/4\, t_{d0}$ and time scaling of the last movement of the robot program.
3) Time scaling of the last movement and *dynamic brakes release*. Dynamic brake release is an optional override function to release the brakes on command within the robot program code. In all those cases where the beginning of the upcoming standstill is known, e.g. at the end of the cycle, the brakes are set to release right after reaching the home position. However, they are not released in case of process standstills, so that the maximum switching cycles for brakes (as set by the manufacturer) are not exceeded. A minimum time delay $t_d^{o3} = 0.5$s is still considered.

Let us denote these three types with o1, o2 and o3 respectively. Results concerning the second optimization method are shown in Fig. 7 (EC expressed in energy per cycle). Over the given time t_{gc}, the curves are plotted as a function of the actual scaling factors, β_k, of the last movement to be set in the robot controller. As for the third optimization method, since $t_d^{o3} < t_d^{o2} < t_t$, the EC profiles are found from curves depicted in Fig. 7 subtracting the offset $\left(t_d^{o3}/t_d^{o2}\right) \cdot E_{wait,k}$. In addition, note that these EC profiles are given only in the acceptable $\boldsymbol{\beta}$-range as defined by Eq. (16). Thus, since R_4 has a very low available time $t_{ah,4}$ (see Fig. 5), its EC minimum is at the longest allowable duration (achieved for $\beta = 30\%$). In contrast, R_2 presents a rough increase of the EC at a low β, due to a largely extended movement duration and, therefore, a growing proportion of the mechanical brakes' consumption and gravity loads. In this case, similarly to the curve depicted in Fig. 4, there exist a practically achievable EC minimum for a given scaling factor.

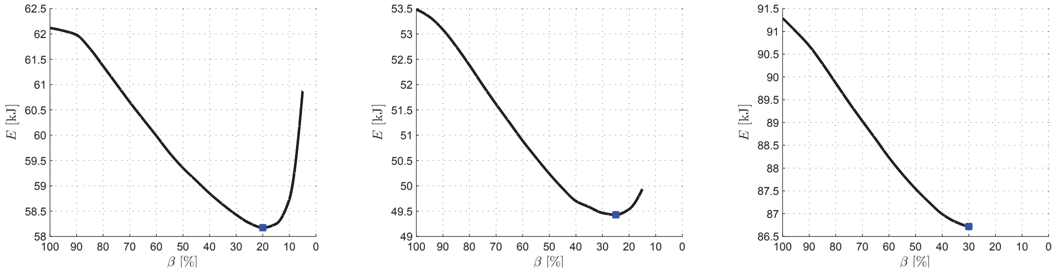
A summary of the average R_k usage in relation to the global cycle time, t_{gc}, together with the actual, minimum and optimal scaling factors (β_{actual}, β_{min}, β_{opt}, respectively) for the last robot movement is given in Table I.

The results for each robot and for each of the three optimization approaches are shown in Fig. 8. The same figure also reports: a) the estimated percentage of EC savings with respect to the state of the art; b) the duration of the various power states, i.e. the process movement, standstill in the HOME position with unreleased brakes, standstill in the

Robot	Average usage $[t_{c,k}/(t_{gc}-t_t)]$	β_{actual} [%]	β_{min} [%]	β_{opt} [%]
R_1	1	100	100	100
R_2	0.48	100	5	20
R_3	0.79	100	14	25
R_4	0.91	100	27	30

TABLE I
POSSIBLE SCALING FACTORS FOR THE LAST MOVEMENT TO THE HOME POSITION

[2] Some latest generation controllers do have cooling adjusted to actual temperature.



(a) Robot R_2

(b) Robot R_3

(c) Robot R_4

Fig. 7. Energy consumption over the cycle t_{gc} depending on the execution velocity of the last movement (optimization method 2).

	E_{cell}[kJ]	EC savings
ref	288.6	-
o1	277.2	-3,9%
o2	264.6	-8.3%
o3	258.0	-10.6%

TABLE II

EC OF THE MULTI-ROBOT CELL OVER THE CYCLE TIME t_{gc}

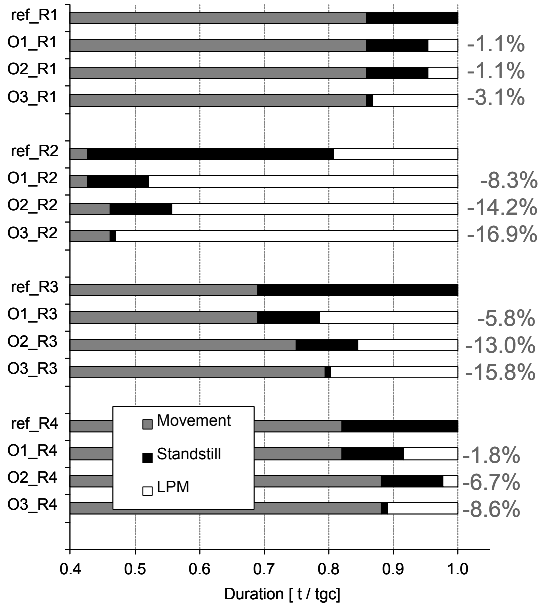


Fig. 8. Comparison of robot consumption and duration of various power states at each optimization method.

HOME position with released brakes (referred to as Low Power Mode or LPM). The EC numerical results per cycle time for the multi-robot cell are summarized in Table II. In order to calculate the annual percentage savings, typical production standstills must be considered [10].

V. CONCLUSIONS

The present paper discusses an energy consumption optimization method for multi-robot production systems. The optimization is constrained by externally given or currently existing robot hardware limitations and production rates. Previous research was extended considering a more detailed robot model, new optimization criteria and other operation constraints. The robot dynamics model is extended to consider other devices in the system, such as the motor brakes, the capacitors and the drain resistance, to deliver a partially regenerative behavior for the motors. The criteria concern different release time delays for the mechanical brakes and time scaling of the last movement of each respective robot. Energy simulation results, based on experimentally measured production timing characteristics on a real automobile production cell, show that the execution time for a robot task is often not optimally synchronized with the others in the cell, so that robots have an excess time after task completion. Following the proposed methods, simulation results show that those idle times can be used to achieve significant EC savings while still respecting robot dynamics limitations, production constraints and cycle times. The method is implementation-close, since it requires very little modification of robot programs and can be easily introduced into existing systems or integrated into *CAR-Tools* for initial production planning. Finally, the results shown energy saving possibilities up to 10,6%.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the contribution of Michael Lebrecht and Thomas Schneider (Daimler AG) and the support by the European Social Fund within the project *Support for the implementation of doctoral studies at Riga Technical University*.

REFERENCES

[1] European Commission. *Energy Efficiency Plan*. Brussels, 2011.

[2] D. Meike and L. Ribickis. Energy efficient use of robotics in the automobile industry. In *The IEEE 15th Int. Conf. on Advanced Robotics*, pages 507–511, 2011.

[3] R. Saidur. A review on electrical motors energy use and energy savings. *Renew. Sust. Energ. Rev.*, 14:877–898, 2010.

[4] J.S. Park. Motion profile planning of repetitive point-to-point control for maximum energy conversion under acceleration conditions. *Mechatronics*, 6 (6):649–663, 1996.

[5] M. Pellicciari et al. A minimal touch approach for optimizing energy efficiency in pick-and-place manipulators. In *The IEEE 15th Int. Conf. on Advanced Robotics*, pages 100–105, 2011.

[6] KUKA Roboter. *http://www.kuka-robotics.com/*, 2011.

[7] B.K. Bose. *Modern Power Electronics and AC Drives*. Prentice Hall PTR, 2002.

[8] P.I. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1):24–32, 1996.

[9] John M. Hollerbach. Dynamic scaling of manipulator trajectories. In *American Control Conference*, pages 752 –756, june 1983.

[10] D. Meike and L. Ribickis. Recuperated energy savings potential and approaches in industrial robotics. In *The IEEE 15th Int. Conf. on Automation Science and Engineering*, pages 299–303, 2011.