

# Energy Efficient Use of Multirobot Production Lines in the Automotive Industry: Detailed System Modeling and Optimization

Davis Meike, Marcello Pellicciari, and Giovanni Berselli

**Abstract**—This paper quantitatively reports about potential energy savings on robotic assembly lines for the automotive industry. At first, a detailed system model is described, which improves previously published results by explicitly considering both manipulator and electrical drive dynamics. The model closely captures experimental data in terms of actuation torques and servodrive voltages, which are directly used to derive the plant input power. Two practical methods are then evaluated for reducing the overall energy consumption. The methods rely on: 1) implementation of energy-optimal trajectories obtained by means of time scaling, concerning the robots' motion from the last process point to the home positions and 2) reduction of energy consumption by releasing the actuator brakes earlier when the robots are kept stationary. Simulation results, based on the production timing characteristics measured at a real plant, clearly shows that the system energy consumption can be effectively reduced without negative effects on the production rate.

**Note to Practitioners**—The global industry trend towards sustainability demands energy optimization as a primary goal. Currently, industrial multirobot systems are not efficiently programmed as long as effective simulation tools are mostly lacking. In this context, a detailed model of the production line is proposed, which can be readily integrated into commercial tools for robot programming. Furthermore, the impact of practical optimization methods which are readily applicable to existing equipment is quantitatively highlighted. The key aspect of the proposed approach is that both the production rate of each cell and the robot hardware limitations are considered as strict constraints, so that no significant changes to the plant are required.

**Index Terms**—Energy efficient robotics, robotic manufacturing, trajectory planning, virtual prototyping.

## 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 robotic systems, it is interesting to note that 15%–28% of a vehicle's EC during its overall lifecycle

occurs during the production phase, whereas the electrical energy consumed by the Industrial Robots (IR) 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 on production costs and total CO<sub>2</sub> emissions.

Concerning EC minimization in robotics, most of the past literature describes effective methods that involve considerable system modifications. For instance, energy optimization achieved by means of hardware replacement is investigated in, e.g., [3]–[5]. Energy-optimal robot selection for specific operations is described in [6]. Many past researches concern energy-optimal paths and motion profiles, also considering the system dynamics and control, see, e.g., [7]–[11]. At last, in the field of autonomous robots for service applications (see, e.g., [12]–[14]), EC can be regarded as a possible cost function to be minimized in order to determine optimal design parameters, such as gear reduction ratios and/or actuator placements [15], [16].

Differently from the aforementioned literature, the methodologies presented in [17]–[19] aim at achieving EC reduction, while avoiding any substantial revision of the industrial plant. These approaches are clearly interesting for application to established robotic systems in a mature lifecycle phase (i.e., already technologically optimized). In particular, [17] describes energy-optimal scheduling of robotic cells by means of trajectory time-scaling with constant scaling factor, the cell production rate and the hardware limitations being considered as strict constraints. Similarly, [18] applies the same method to parallel kinematic machines. At last, [19] extends the time scaling procedure by introducing a dynamically varying scaling factor (i.e., a generic monotonically increasing function of time), which provides better simulation results but is hardly implementable on the real system due to the intrinsic limitations of standard industrial controllers.

Following the same direction, the aim of the present paper is to improve upon [17]–[20], where simplified dynamic models are employed and no experimental measures on real industrial plants are presented. In particular, a case study performed on an industrial robot cell of a vehicle body-shop production line is reported. The cell is composed of four anthropomorphic IRs (210 Kg payload) that do not share a common workspace and are employed in spot-welding or handling applications. The production timing characteristics are first measured, highlighting that IRs are commonly programmed to operate at high speed whether this is necessary or not. Then, a detailed model of a single robot is derived, which closely captures experimental

Manuscript received March 28, 2013; revised August 04, 2013; accepted September 21, 2013. This paper was recommended for publication by Associate Editor T. Bretl and Editor D. Tilbury upon evaluation of the reviewers' comments. (Corresponding author: G. Berselli.)

D. Meike is with Daimler AG, Sindelfingen 71063, Germany (e-mail: davis.meike@daimler.com).

M. Pellicciari and G. Berselli are with the University of Modena and Reggio Emilia, Modena 41100, Italy (e-mail: name.surname@unimore.it; giovanni.berselli@unimore.it).

Digital Object Identifier 10.1109/TASE.2013.2285813

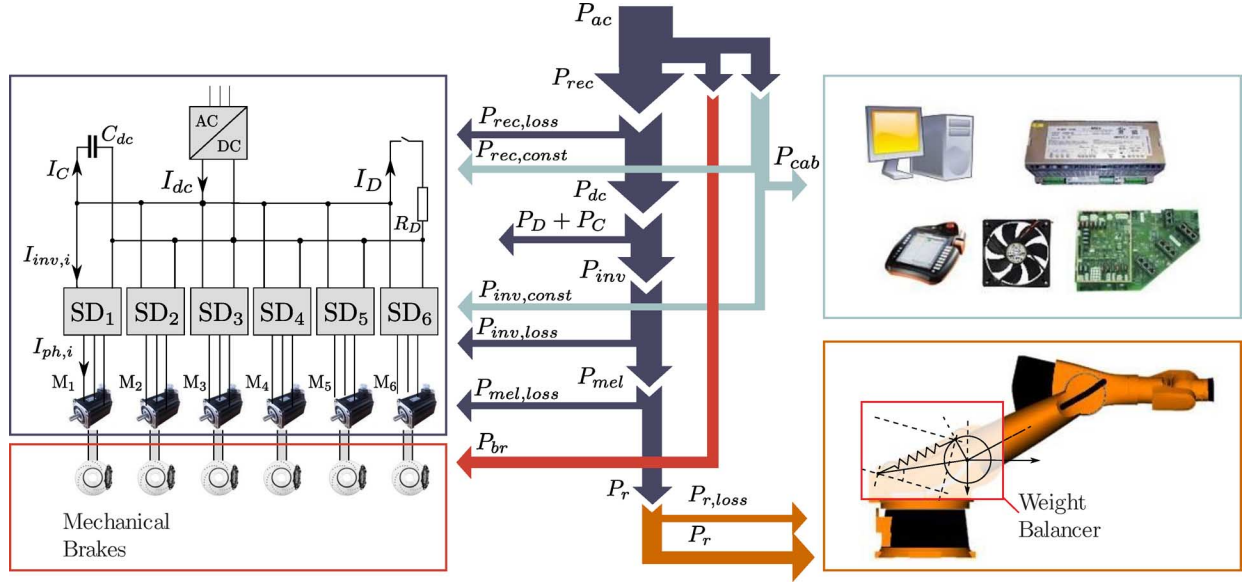


Fig. 1. Schematic of the power flow within the multidrive system of an industrial robot.

data both in terms of actuation torques and servodrive voltages (directly used to derive the input power). The model improves the ones described in previous literature, by explicitly including the contribution of weight balancing mechanisms, electric drive dynamics, and several sources of energy loss within the actuation system. Then, two saving effects are evaluated on the basis of the measured data (i.e., process schedule/constraints), namely:

- Time scaling concerning the robot's motion from the last process point to the robot's home position. In fact, the last robot motion is not process relevant and can be varied without affecting the production rate. A constant scaling factor (also referred hereafter as *override ratio*) is employed, as long as it can be readily implemented in the industrial controller.
- A reduction in the release time of the robot's brakes. The IR actuators are usually equipped with normally closed brakes which prevent the IR from moving when released. Note that, following standard convention, brakes are said to be "closed" or "released" when they are engaged. Also, these mechanical brakes require a certain amount of power to be kept open (i.e., disengaged). An earlier brake release allows to reduce both brake power consumption and the energy wasted to counteract gravity when the IR is idling. The overall experiential EC is finally quantified, confirming that noticeable energy savings are possible without any additional investments on new equipment or negative consequences on the cell production rate.

## II. ENERGY CONSUMPTION MODELING AND OPTIMIZATION APPROACH

A typical 6-dof anthropomorphic IR is actuated by a multidrive servo system which comprises six Permanent Magnet Synchronous Machines (PMSM) and the related power converters, composed of AC/DC module (rectifier), DC bus, and DC/AC modules (inverters, also referred to as servodrives).

With reference to Fig. 1,  $SD_i$  and  $M_i$ ,  $i = 1 \dots 6$ , refer to servodrives and electric motors, whereas the power flow through the system has been conceptually decomposed into several terms, namely:

- Total electrical power delivered to the system,  $P_{ac}$ , to the central AC/DC converter,  $P_{rec}$ , to the DC bus,  $P_{dc}$ , to the six inverters,  $P_{inv}$ , or overall electric power input of the six PMSM,  $P_{mel}$ ;
- Power,  $P_C$ , to charge/discharge the DC bus capacitance,  $C_{dc}$ ;
- Overall mechanical power delivered to the PMSM shafts,  $P_m$ , and subsequently delivered to the robot mechanical structure and external users,  $P_r$ ;
- Electrical power consumed when the electromechanical brakes are opened,  $P_{br}$ ;
- Load-independent power,  $P_{cab}$ , delivered to cabinet's PC, cooling, and IR control panel. The term  $P_{cab}$  also includes the load-independent losses  $P_{rec,const}$  and  $P_{inv,const}$ , respectively due to the permanent power requirements of the control electronics within the rectifier and the six inverters (active also when  $P_{rec}$  and  $P_{inv}$  are null).

In parallel, the overall power loss can be described by the following terms:

- Power loss due to mechanical friction in the IR structure,  $P_{r,loss}$ , mainly due to the presence of gear reducers;
- Electric power loss in the six PMSM,  $P_{mel,loss}$ , on the six inverters,  $P_{inv,loss}$ , and on the rectifier,  $P_{rec,loss}$ ;
- Electric power loss,  $P_D$ , due to the activation of the drain resistor,  $R_D$  (hereafter also referred to as brake chopper).

With the exception of the load-independent terms, assuming the system electromechanical parameters and the Tool Center Point (TCP) path in the Cartesian space as given, the amount of EC strongly depends on the programmed IR motion law at joint level and on the brake release time delay,  $T_b$  (i.e., the amount of time for which the IR is kept stationary by actively controlling its actuators). In particular, with reference to a given time interval  $t \in [0, t_{ah}]$ , an IR is usually assumed to operate at

its maximum speed whenever allowed by the scheduling/technological constraints and, otherwise, to remain stationary (idle times). However, the corresponding joint space time-optimal trajectory [21], [22], which is taken without loss of generality as the reference trajectory denoted as  $q_{O,i}(t)$ , may not be energy-optimal due to high power-consuming accelerations and longer idle times where energy is wasted to counteract gravity (if mechanical brakes are opened). Similarly to [18] and [19], let one alternatively assume that the aforementioned reference path is followed with a trajectory whose  $i$ th joint position profile is given by:

$$q_i(t) = q_{O,i}(\beta t) = q_{O,i}(t') \quad (1)$$

where  $t' = \beta t$  and  $\beta \leq 1$  are denoted as *scaled time* [23], [24] and *override ratio*, respectively. Note that, as previously said, the override ratio may be a generic function of time, as in [19]. On the other hand, the use of dynamic time-scaling is extremely interesting from a theoretical point of view but less interesting in a practical scenario. In fact, linear time-scaling is readily applicable on current industrial controllers by simply changing the override ratio, whereas the implementation of the motion laws resulting from nonuniform time-scaling requires controllers different from the industrial ones (such as, for instance, expensive research interfaces which are used in robotic labs only). Also, note that any TCP trajectory may be taken as reference. In case the time-optimal trajectory is chosen as reference, the time-scaling factor coincides with the override ratio implemented in the IR controller (an override ratio  $\beta = 100\%$  simply representing the time-optimal trajectory).

The electrical input power,  $P_{ac}$ , can then be computed as a function of both the user-defined parameters  $\beta$  and  $T_b$ , such that  $P_{ac} = P_{ac}(\beta, T_b, t)$ . For each  $\beta$  and  $T_b$  values, the overall EC is then given by

$$E(\beta, T_b) = \int_0^{t_{ah}} P_{ac}(\beta, T_b, t) dt. \quad (2)$$

Clearly, the correct computation of the robot EC and, consequently, of the energy-optimal parameters  $\beta$  and  $T_b$ , require a detailed description of the system energy flow as described in the following subsections. The model parameters have been acquired from multiple sources. Manipulator inertial parameters (see Section II-A) are confidential data from the IR manufacturer which, in any case, may also be retrieved resorting to well-known identification techniques, e.g., [25] and [26]. The parameters of PMSM and drive DC bus at rated operation (see Section II-B) are provided in the datasheets from the respective manufacturers. The remaining parameters (i.e., load-dependent and static electrical losses as well as resistances of various component) have been directly acquired from direct measurements (see, e.g., methods described in [27]).

#### A. Mechanical Power

The mechanical power delivered to the  $i$ th PMSM shaft,  $P_{m,i}$ , can be computed as

$$P_{m,i} = P_{r,i} + P_{r,loss,i} = \tau_i \dot{q}_i + f_i \dot{q}_i \quad (3)$$

where  $P_{r,i}$  and  $\tau_i$  are, respectively,  $i$ th axis power and torque counteracting gravity and inertial loadings,  $P_{r,loss,i}$  and  $f_i$  are

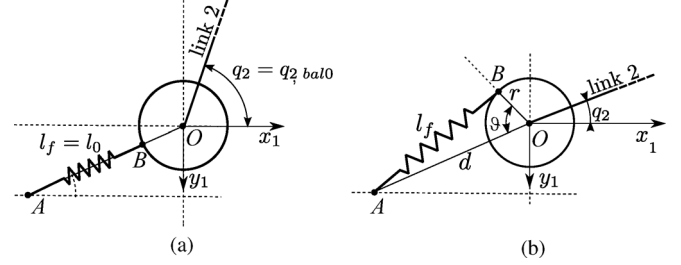


Fig. 2. Weight balancing system of axis 2. (a) Rest position. (b) Active position.

respectively  $i$ th axis mechanical power loss and torque caused by friction, whereas  $\dot{q}_i$  is the  $i$ th axis joint velocity. Resorting to the well known IR dynamic model [28], the torque vector  $\tau = [\tau_i]^T$ , during the robot unconstrained motion, can be expressed as the sum of the following terms:

$$\underbrace{D(q)\ddot{q} + C(q,\dot{q})}_{\tau_I} + \underbrace{g(q) + e(q)}_{\tau_g} \quad (4)$$

where  $D(q)$  is the symmetric joint-space inertia matrix (also accounting for PMSM rotors inertia),  $C(q, \dot{q})$  describes the Coriolis/centripetal torques,  $g(q)$  is the gravity loading,  $e(q)$  is a torque vector accounting for the presence of a weight balancing system (see Fig. 1), whereas  $q = [q_i]^T$ ,  $\dot{q} = [\dot{q}_i]^T$ ,  $\ddot{q} = [\ddot{q}_i]^T$  are joint angular position, velocity and acceleration vectors, respectively.

For the purpose of the following sections, it is convenient to isolate the inertial,  $\tau_I = [\tau_{I,i}]^T$ , and the gravitational,  $\tau_g = [\tau_{g,i}]^T$ , contributions of the torque vector  $\tau$ . Also, the  $i$ th motor shaft torque,  $\tau_{m,i}$ , and angular velocity,  $\omega_{m,i}$ , can be computed as

$$\tau_{m,i} = (\tau_i + f_i)/G \quad (5a)$$

$$\omega_{m,i} = G\dot{q}_i \quad (5b)$$

where  $G_i$  is the  $i$ th PMSM gear ratio. Equation (5a) trivially shows that the mechanical power  $P_{m,i}$  of (3) can also be computed as

$$P_{m,i} = \tau_{m,i} \omega_{m,i}. \quad (6)$$

1) *Friction Losses*: The torque component due to friction on the PMSM shaft,  $f_i$ , are computed by means of a linear model, where friction is dependent on angular velocity only. The frictional torques can then be formulated, for each axis individually as

$$f_i(\dot{q}_i) = K_{c,i} \text{sgn}(\dot{q}_i) + K_{v,i} \dot{q}_i \quad (7)$$

where  $K_{c,i}$  and  $K_{v,i}$  are the Coulomb and viscous friction coefficients of the  $i$ th axis, respectively.

2) *Weight Balancing System*: As depicted in Figs. 1 and 2, a spring weight balancing system is considered. The corresponding torque vector is  $e(q) = [\tau_{b,i}]^T$ , where  $\tau_{b,i}$  is the torque component due to the balancer and exerted on the  $i$ th axis. Since the weight balancing is used for axis  $i = 2$  only, all elements except  $\tau_{b,2}$  are zeros and  $e(q)$  is function of  $q_2$  only.

With reference to Fig. 2, where the coordinate frame  $x_1 - y_1$  is depicted, let one denote the distances of spring connection points,  $A$  and  $B$ , from the joint revolution axis trace,  $O$ , as  $d$

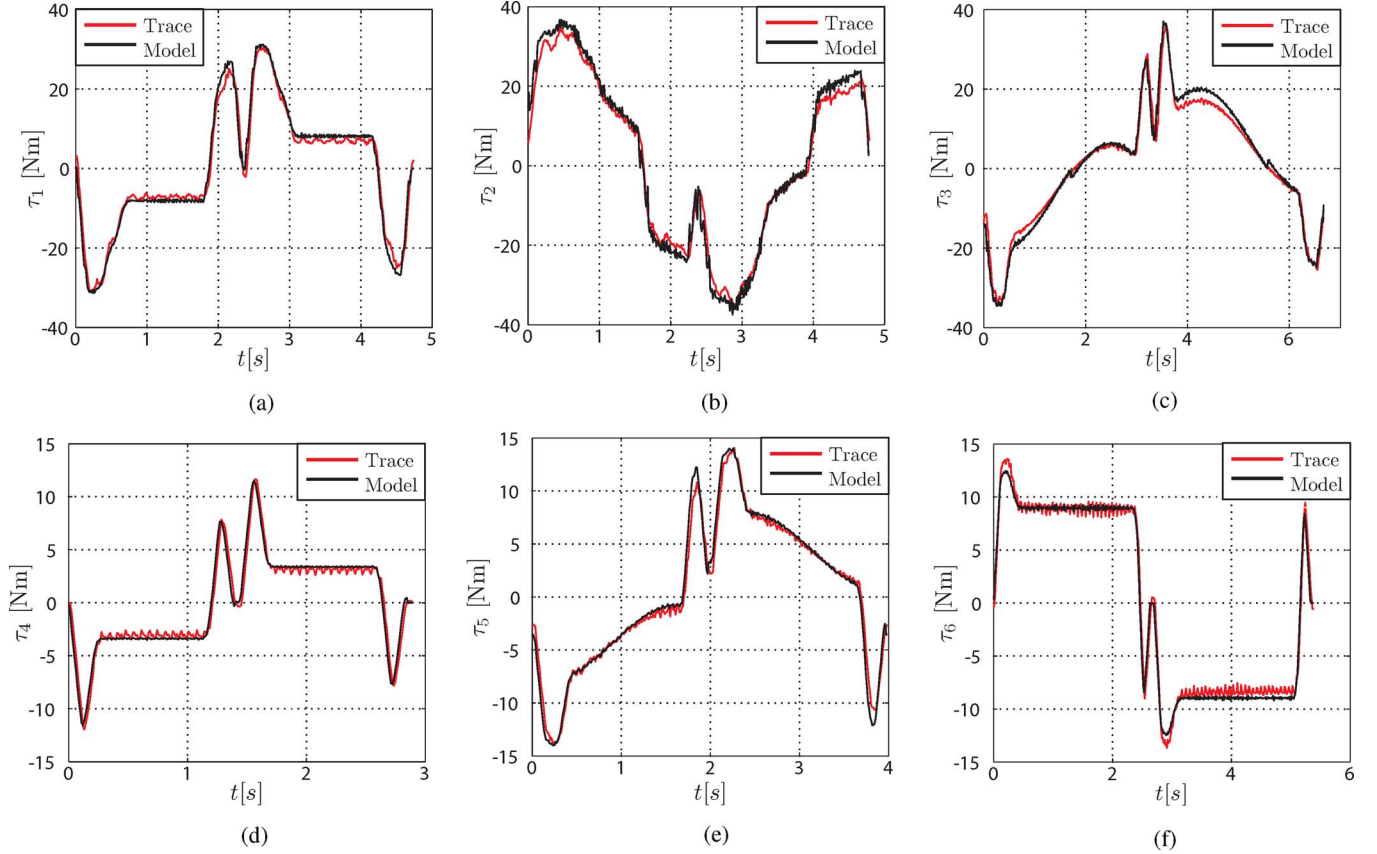


Fig. 3. Torque comparison, measurement and simulation, load = 154 Kg,  $\beta = 100\%$  override. (a) Axis 1,  $\Delta\tau_1 = 2.6\%$ . (b) Axis 2,  $\Delta\tau_2 = 3.0\%$ . (c) Axis 3,  $\Delta\tau_3 = 2.1\%$ . (d) Axis 4,  $\Delta\tau_4 = 2.4\%$ . (e) Axis 5,  $\Delta\tau_5 = 2.0\%$ . (f) Axis 6,  $\Delta\tau_6 = 2.3\%$ .

and  $r$ , respectively. In parallel, let one define the axis angle at the configuration of balancer full backstroke [i.e., when the moment acting on the axis is zero, see Fig. 2(a)] as  $q_{2,bal0}$ . The base spring length,  $l_0$ , and the actual spring length,  $l_f$  are, respectively, given by

$$\begin{aligned} l_0 &= d - r \\ l_f &= [(l_0 + r(1 - \cos \vartheta))^2 + (r \sin \vartheta)^2]^{1/2} \end{aligned} \quad (8)$$

having defined  $\vartheta = q_{2,bal0} - q_2$  (see Fig. 2(b)). The spring force modulus is then given by

$$f_{sp} = k_{sp}(l_f - l_0) + f_0 \quad (9)$$

where  $k_{sp}$  is a spring constant and  $f_0$  is the initial spring force when  $q_2 = q_{2,bal0}$ . Finally, the moment acting on the axis is

$$\tau_{b,2} = -f_{sp}dr \sin \vartheta / l_f. \quad (10)$$

**3) Dynamic Model Validation:** A comparison of measured and simulated PMSM torques for each axis  $M_1, \dots, M_6$  of sample robot programs of different duration on a KR210.2 manipulator is shown in Fig. 3. A difference error between simulated and measured torques is given below each subfigure (a)–(f). The simulation deviation,  $\Delta\tau_i$  is calculated according to maximal torque amplitude in the given measuring cycle as

$$\Delta\tau_i = 100 \frac{\frac{1}{t} \int_0^{t_{ah}} |\tau_{meas,i}(t) - \tau_{simul,i}(t)| dt}{\max(\tau_i) - \min(\tau_i)} \quad (11)$$

where  $\tau_{meas,i}$  and  $\tau_{simul,i}$  are the measured and simulated torques for axis  $i$ , respectively. In particular, the figure highlights that the overall torques (including the contribution of friction) are closely captured. Consequently, the proposed model provides adequate accuracy for the prediction of the mechanical power flow.

### B. PMSM Electrical Power

The PMSM electrical power can be estimated by referring to the electrical machine models described in [29], which relates the PMSM behavior at a generic operating point with the PMSM behavior at rated (nominal) operation. In the following, an additional index  $N$  will refer to this nominal behavior. The PMSM's parameters that are typically given for rated operation are summarized in Table I. For the  $i$ th axis, the total active electrical power flow,  $P_{mel,N,i}$ , at rated operation can be expressed as

$$P_{mel,N,i} = P_{m,N,i} + P_{mel,loss,i} = \frac{P_{m,N,i}}{\eta_{N,i}} \quad (12)$$

where

$$\begin{aligned} P_{m,N,i} &= \tau_{m,N,i} \omega_{m,N,i} \\ P_{mel,loss,N,i} &= \frac{1 - \eta_{N,i}}{\eta_{N,i}} P_{m,N,i}. \end{aligned} \quad (13)$$

As for the term  $P_{mel,loss,i}$  in (12), it is conceptually composed of the following terms.

- **Stator winding resistive losses**,  $P_{Rs,i}$ , which are proportional to winding resistances and depends on stator current (and, therefore, on torque).

TABLE I  
NOMINAL MOTOR PARAMETERS REFERRING TO THE  $i$ TH AXIS

Parameter	Symbol	Unit
Torque	$\tau_{m,N,i}$	Nm
Motor shaft angular speed	$\omega_{m,N,i}$	rpm
Phase RMS current	$I_{ph,N,i}$	A
Efficiency at rated operation	$\eta_{N,i}$	-
Number of pole pairs	$p_i$	-

- **Iron losses**,  $P_{Fe,i}$ , due to magnetization and demagnetization of the windings.
- **Additional losses**,  $P_{add,i}$ , which represent miscellaneous losses and possible calculation error according to standard norms (i.e., [30]).

Therefore, the total electrical losses at rated operation can be expressed as

$$P_{mel,loss,N,i} = P_{Rs,N,i} + P_{Fe,N,i} + P_{add,N,i}. \quad (14)$$

In particular, also recalling (12), the additional losses of the total electrical consumption can be written as

$$P_{add,N,i} = k_{\eta,add} P_{mel,N,i} = k_{\eta,add} \frac{P_{m,N,i}}{\eta_{N,i}} \quad (15)$$

where  $k_{\eta,add} = 0.005$  according to norm IEC 60034-30 [30]. The motor phase current, outside the rated operation, for a given generic torque is

$$I_{ph,i} = \frac{\tau_{m,i}}{k_{t,i}} \quad (16)$$

where

$$k_{t,i} = \frac{\tau_{m,N,i}}{I_{ph,N,i}}. \quad (17)$$

The stator resistive losses can then be calculated as

$$P_{Rs,i} = 3I_{ph,i}^2 R_{s0} (1 + \alpha_{Cu} (\vartheta_i - \vartheta_0)) \quad (18)$$

where  $R_{s0}$  is a measured stator resistance at ambient temperature,  $\vartheta_0$  is the ambient temperature,  $\vartheta_i$  is the actual motor temperature ( $\vartheta_i = 100$  °C in the following), and  $\alpha_{Cu}$  is a temperature-related coefficient of copper ( $3.93 \cdot 10^{-3}/^\circ\text{C}$ ). Note that, the stator resistive losses at rated operation,  $P_{Rs,N,i}$ , can be simply computed resorting to (18) by imposing  $I_{ph,i} = I_{ph,N,i}$  and  $\vartheta_i = \vartheta_{N,i}$ , the value  $\vartheta_{N,i}$  being the motor temperature at rated operation. Concerning the iron losses, in classical motor designs, they are usually modeled as strongly dependent on the PMSM magnetic induction and its frequency of variation (see, e.g., [27]). Therefore, an exact computation of the term  $P_{Fe,i}$  would require a complete simulation of the PMSM electromagnetic field, which is possible only with proprietary data from motor manufacturer. Hence, similarly to [31], a simplified yet reliable model is used and the motor iron losses are calculated as proportional to the shaft angular velocity, such that

$$\begin{aligned} P_{Fe,i} &= \frac{|\omega_{m,i}|}{\omega_{m,N,i}} P_{Fe,N,i} \\ &= \frac{|\omega_{m,i}|}{\omega_{m,N,i}} (P_{mel,loss,N,i} - P_{add,N,i} - P_{Rs,N,i}) \end{aligned} \quad (19)$$

At last, additional losses at a generic operating point can be written, similarly to (15) as

$$P_{add,i} = k_{\eta,add} P_{mel,i}. \quad (20)$$

The electrical power requirement,  $P_{mel,i}$ , is then given by the sum of mechanical power,  $P_{m,i}$ , and electrical losses,  $P_{mel,loss,i}$ , the latter being computed as a sum of the previously calculated power terms in (18)–(20). Therefore, the following equation holds:

$$P_{mel,i} = \frac{P_{m,i} + P_{Rs,i} + P_{Fe,i}}{1 - k_{\eta,add}} \quad (21)$$

where  $P_{m,i}$  is the mechanical power requirement for the  $i$ th axis acquired from the dynamic model [see (6)].

### C. Electrical Drive Power Flow

A conceptual scheme of the electrical drive (inverters + rectifier + DC bus) is depicted in Fig. 1. In particular, the energy flow towards the AC network is normally prevented, unless the multidrive system is equipped with a regenerative module [32]. On the other hand, even if the electrical power input is unidirectional (i.e.,  $P_{ac} \geq 0$ ), most multidrive systems allow for motor-to-motor energy exchange within the common DC bus. Therefore, when an axis decelerates, the mechanical energy is converted back into electrical energy to be either used by other PMSM if needed, or stored in the DC bus capacitor  $C_{dc}$  (see Fig. 1). In the latter case the DC bus voltage increases above the rectified AC supply voltage. At the same time, when several axes brake simultaneously, possible excessive over-voltages are dissipated by the aforementioned drain resistor,  $R_D$ . For instance, past measurements [2] highlight possible peaks of  $\sim 690$  V which are lowered by means of  $R_D$ .

1) *Inverter Models*: First, in order to correctly model the electrical drive behavior, the power flow through each inverter is considered. Using a method similar to that applied in the previous subsection, the  $i$ th inverter behavior at a generic operating point is related to its nominal behavior. In particular, the overall  $i$ th inverter losses at rated operation are split into a load-dependent term,  $P_{inv,loss,N,i}$ , and a constant term,  $P_{inv,const,i}$ , the latter being accounted for in the term  $P_{cab}$  (as depicted in Fig. 1). Then, an experimentally determined factor  $k_{igbt}$  splits the load-dependent losses into resistive losses due to IGBT switching [32], capacitor balancer resistances and circuit boards active resistance that are proportional to the square of the current [first term in (22)]. The remaining losses [second term in (22)] are the consumption of the cooling fan, whose speed is proportional to actual inverter load. Since thermal processes in this context are much slower, a simplified model is used and losses are assumed proportional to an average inverter current,  $I_{ph,i,avg} = t_{ah}^{-1} \int_0^{t_{ah}} I_{ph,i} dt$ , computed over the whole simulation duration  $t_{ah}$  in the PMSM submodel [Section II-B]. Therefore, the following relation holds:

$$\begin{aligned} P_{inv,loss,i} &= P_{inv,loss,N,i} \left( k_{igbt} \frac{I_{ph,i}^2}{I_{inv,N,i}^2} \right. \\ &\quad \left. + (1 - k_{igbt}) \frac{|I_{ph,i,avg}|}{I_{inv,N,i}} \right). \end{aligned} \quad (22)$$



The overall electrical energy delivered to the six inverters is then given by

$$P_{\text{inv}} = \sum_{i=1}^n (P_{\text{mel},i} + P_{\text{inv,loss},i}) \quad (23)$$

where  $P_{\text{mel},i}$  is the electrical power requirement for motor  $i$  taken from (21).

2) *DC-Bus*: As depicted in Fig. 1, the model of the DC-bus power flow comprises the terms  $P_{\text{dc}}$ ,  $P_C$ , and  $P_D$ . In order to correctly compute these power terms, it is necessary to numerically evaluate the DC bus voltage variation. The calculation is based on a differential model, the derivatives being approximated by finite differences. In the following, the index  $j-1$  for any variable refers to the value of the previous integration step, the index  $j$  (current integration step) being omitted for clarity. If  $U_{ph}$  is a phase-to-phase input supply voltage of the AC network, peak and idle RMS DC bus voltages have constant values of

$$U_{\text{dc},0} = \sqrt{2}U_{ph} \quad (24a)$$

$$U_{\text{dc},d} = \frac{3\sqrt{2}}{\pi}U_{ph} \quad (24b)$$

respectively. During IR functioning, the DC bus voltage can vary above and below the RMS voltage  $U_{\text{dc},d}$ , the corresponding electric drive states being referred to as *recuperation* or *loading condition*, respectively. In particular, the following equation holds:

$$\left. \begin{aligned} U_{\text{dc}} &= U_{\text{dc},d} - U_{\text{dc},r} \\ P_{\text{dc}} &= P_{\text{inv}} \end{aligned} \right\} \quad \text{if } U_{\text{dc},[j-1]} < U_{\text{dc},d} \wedge P_{\text{inv}} > 0$$

$$\left. \begin{aligned} U_{\text{dc}} &= \sqrt{\frac{2W_C}{C_{\text{dc}}}} \\ P_{\text{dc}} &= 0 \end{aligned} \right\} \quad \text{otherwise} \quad (25)$$

where  $U_{\text{dc},r}$  is the voltage ripple,  $W_C$  is the actual energy in the capacitor  $C_{\text{dc}}$ , and  $P_{\text{dc}}$  is the power requirement of the DC bus.

The first case in (25) models the *loading condition*. When a robot manipulator accelerates, the sum of power requirement of all inverters is positive (i.e.,  $P_{\text{inv}} > 0$ ) and power is required from the DC bus input, if additional energy is not present in the capacitor ( $U_{\text{dc},[j-1]} < U_{\text{dc},d}$ ). At load change a DC bus voltage ripple is present, such that

$$U_{\text{dc},r} = \frac{U_{\text{dc},d}}{12 f C_{\text{dc}} X_{\text{dc},[j-1]}} \quad (26)$$

where

$$X_{\text{dc},[j-1]} = \frac{U_{\text{dc},[j-1]}}{I_{\text{dc},[j-1]}} \quad (27)$$

is the total input impedance of the DC bus at the previous integration step, whereas  $f$  is the AC network frequency. Therefore, the initial DC bus conditions in the simulation must be imposed as

$$I_{\text{dc}}(0) = I_{\text{dc,init}} \quad (28a)$$

$$U_{\text{dc}}(0) = U_{\text{dc,init}} \quad (28b)$$

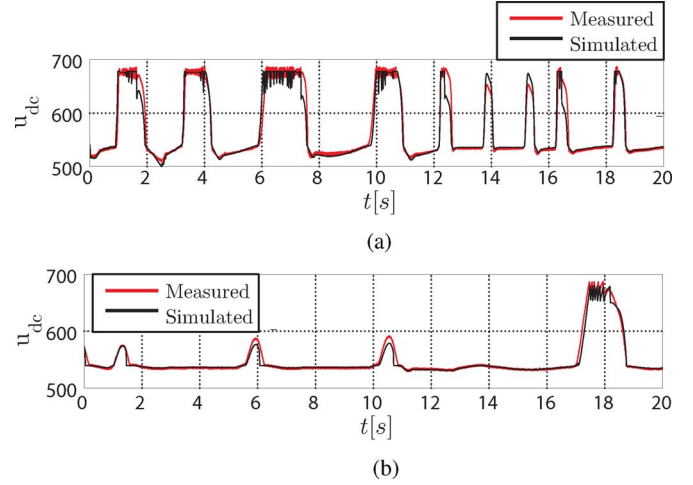


Fig. 4. Comparison of measured and simulated DC bus voltages. (a) DC bus voltage at tool load = 154 Kg,  $\beta = 100\%$ . (b) DC bus voltage at tool load = 154 Kg,  $\beta = 50\%$ .

In particular, in (28a) and (28b), it can be assumed that  $I_{\text{dc,init}} = 0$  and  $U_{\text{dc,init}} = U_{\text{dc},0}$  if the IR torques are initially zero and no additional energy is stored in the DC bus.

As for the second case in (25), it models the *recuperation condition* occurring when regenerative energy charges the capacitor  $C_{\text{dc}}$ . In particular, since a full bridge diode rectifier is used [32], the *recuperation condition* states that no energy is taken from the AC network if the DC bus voltage is higher than the idle voltage. In general, the energy,  $W_C$ , and the power,  $P_C$ , stored in the capacitor can be computed as

$$W_C = W_{C,[j-1]} + (P_{\text{dc}} - P_{\text{inv}} - P_D)\Delta t \quad (29)$$

$$P_C = (W_{C,[j-1]} - W_C)/\Delta t \quad (30)$$

where  $W_C(0)$  can be retrieved from (28b),  $\Delta t$  is the duration of the integration step, whereas  $P_{\text{dc}} = 0$  as in the second case of (25). Due to the over-voltage protection, some of the energy is dissipated on the drain resistor  $R_D$ . Brake chopper working range is defined as  $[U_{\text{off}} \dots U_{\text{on}}]$  and it is triggered only when these limit values are exceeded, such that

$$\gamma_D = \begin{cases} 1, & \text{if } U_{\text{dc},[j-1]} \geq U_{\text{on}} \\ 0, & \text{if } U_{\text{dc},[j-1]} \leq U_{\text{off}} \end{cases} \quad (31)$$

The dissipated power on the drain resistor is

$$P_D = \frac{\gamma_D d U_{\text{dc},[j-1]}^2}{R_D} \quad (32)$$

where  $d$  is the duty cycle of the brake chopper switch. An ability to model the DC bus voltage fluctuation is crucial for an exact robot model. Therefore, a comparison between measured and simulated voltages is provided in Fig. 4. Two modes are shown: high load (a), and no load at reduced speed (b). The average deviation, when calculated in a similar manner to (11), does not exceed 3%.

3) *Rectifier, Mechanical Brakes, and Total Input Power*: As the next step, the input power of the rectifier is derived as

$$P_{\text{rec}} = P_{\text{dc}} + P_{\text{rec,loss}} \quad (33)$$

where  $P_{\text{rec,loss}}$  is computed, identically as in (22), on the basis of rated current,  $I_{\text{rec},N}$ , and rated rectifier losses, which are split into load-dependent losses,  $P_{\text{rec,loss},N}$ , and load-independent losses,  $P_{\text{rec,const}}$ . The total consumption of a robot system is the sum of movement energy, loss model and static loads, such that

$$P_{\text{ac}} = \gamma_{\text{br}}(P_{\text{rec}} + P_{\text{br}}) + P_{\text{cab}} \quad (34)$$

where, as said,  $P_{\text{br}}$  is the measured power requirement of all actuator mechanical brakes, whereas  $P_{\text{cab}}$  is the power requirement for cabinet's PC, cooling, control panel, and other electronics, including load-independent consumption of rectifier,  $P_{\text{rec,const}}$ , and inverters,  $P_{\text{inv,const}}$ . In parallel, the brake coefficient,  $\gamma_{\text{br}}$ , is defined as

$$\gamma_{\text{br}}(T_b) = \begin{cases} 1, & \text{for opened brakes} \\ 0, & \text{for closed brakes} \end{cases} \quad (35)$$

where  $T_b$  is the brake release time delay. In practice, the brake coefficient simply depicts the state whether the manipulator is kept stationary by means of the mechanical brakes or it is actively stabilized by its actuators. After the end of the IR motion, the brakes will be kept opened for a time period equalling the user-defined parameter  $T_b$ .

### III. EFFECT OF TRAJECTORY TIME SCALING ON ROBOT ENERGY CONSUMPTION

In order to clarify the effect of trajectory time scaling on the robot EC, let us consider only the overall power delivered to the robot actuators, namely,  $P_{\text{mel}} = \sum_{i=1}^6 P_{\text{mel},i}$  and  $P_{\text{br}}$ , while momentarily disregarding any other power term (i.e., the power flows within the IR cabinet, inverters, DC bus, and rectifier). In the following, recalling the notation defined in (1), all quantities referring to the reference time-optimal trajectory (obtained when  $\beta = 100\%$ ) will be denoted with  $O$  subscript.

As shown in, e.g., [17] and [18], the scaled velocities and accelerations can be expressed as a function of the parameter  $\beta$  and of time  $t$ , that is

$$\begin{aligned} \dot{q}_i(\beta, t) &= \frac{dq_{O,i}(t')}{\beta^{-1}dt'} = \beta \dot{q}_{O,i}(t') \\ \ddot{q}_i(\beta, t) &= \frac{d(\beta \dot{q}_{O,i}(t'))}{\beta^{-1}dt'} = \beta^2 \ddot{q}_{O,i}(t') \end{aligned} \quad (36)$$

In the same manner, recalling the notation defined in (4) and (5a), the scaled  $i$ th motor shaft torque and angular velocity as function of  $\beta$  and  $t$ , are given by [17]

$$\begin{aligned} \tau_{m,i}(\beta, t) &= G^{-1}[\beta^2 \tau_{IO,i}(t') + \beta K_{v,i} \dot{q}_{O,i}(t') \\ &\quad + K_{c,i} \text{sgn}(\dot{q}_{O,i}(t')) + \tau_{gO,i}(t')] \\ \omega_i(\beta, t) &= G \dot{q}_{O,i}(t') \end{aligned} \quad (37)$$

Henceforth, with reference to a given time interval  $t \in [0, t_{ah}]$ , let one suppose that the IR is in motion for  $t \in [0, t_O/\beta]$ , whereas it is kept stationary for  $t \in [t_O/\beta, t_{ah}]$ . Resorting to

(16), (18), (19), (21), the scaled PMSM electrical power input,  $P_{\text{mel}}$ , can be computed as

$$P_{\text{mel}}(\beta, t) = \sum_{i=1}^6 [(1 - k_{\eta,add})^{-1} [\tau_{m,i}(\beta, t) \omega_{m,i}(\beta, t) + \frac{3R_s}{k_{t,i}} \tau_{m,i}^2(\beta, t) + \frac{P_{Fe,N,i}}{\omega_{m,N,i}} \omega_{m,i}(\beta, t)]] \quad (38)$$

This power term  $P_{\text{mel}}$  reduces to  $P_{\text{mel}}^*$  when the robot is idling in a configuration defined by  $q_i = q_i(t_{ah})$ , such that

$$P_{\text{mel}}^*(\beta) = \sum_{i=1}^6 [(1 - k_{\eta,add})^{-1} \left[ \frac{3R_s}{k_{t,i}} \tau_{gO,i}^2(t_{ah}) \right]] \quad (39)$$

Hence, the overall energy consumption of the IR actuators will be given by

$$E(\beta, T_b) = E_m(\beta) + E_{\text{idle}}(\beta, T_b) + E_{\text{br}}(\beta, T_b) \quad (40)$$

where

$$\begin{aligned} E_m(\beta) &= \int_0^{t_O/\beta} P_{\text{mel}}(\beta, t) dt \\ E_{\text{idle}}(\beta, T_b) &= (t_{ah} - t_O/\beta) \cdot P_{\text{mel}}^*(\beta) \cdot \gamma_{\text{br}}(T_b) \\ E_{\text{br}}(\beta, T_b) &= t_{ah} \cdot P_{\text{br}} \cdot \gamma_{\text{br}}(T_b). \end{aligned} \quad (41)$$

In particular, the terms  $E_m$  and  $E_{\text{idle}}$  represent the EC when the IR is either in motion or idling, whereas the term  $E_{\text{br}}$  represents the EC for keeping the brakes opened. As said, the brake coefficient  $\gamma_{\text{br}}(T_b)$  [see (35)] depends on the user-defined parameter  $T_b$  and, naturally, an earlier brake release (i.e., low  $T_b$  values) is always beneficial in terms of energy consumption.

As an example, Fig. 5 illustrates the normalized EC profiles concerning the energy terms defined in (40) and (41) where, with obvious notations, the values  $E_{\text{max}}, E_{m,\text{max}}, E_{\text{idle,max}}, E_{\text{br,max}}$  denote the maximum values of each energy term. In all the figures, the *forbidden zone* simply represents an area where the IR motion would end for  $t > t_{ah}$  due to extremely low  $\beta$  values. Concerning the brake release time delay  $T_b$ , the three most interesting practical cases are considered, namely:

- Release time  $T_b = 0$ , which models a purely ideal case where the brakes are instantaneously released at the end of the robot motion (the lowest values actually allowed by the real controller being  $T_b = 0.1$  s, see Section IV);
- Release time  $T_b = \infty$ , which models a purely ideal case where the brakes are always kept opened;
- Release time  $T_b = T_{b0}$ , which models a practical case where, for a given IR lifetime, the brake release time  $T_{b0}$  is computed in order not to exceed the maximum number of brake switching cycles allowed by the IR warranty limits (see, e.g., [33]). The interested reader should refer to [2] for an exact description on how a decreasing  $T_{b0}$  influences the maximum number of brake switching cycles in a case study application.

In the following discussion, the forbidden zone will be neglected for clarity.

Concerning Fig. 5(a), it depicts EC when the robot is moving and, obviously, the brakes are opened. The contributions of the

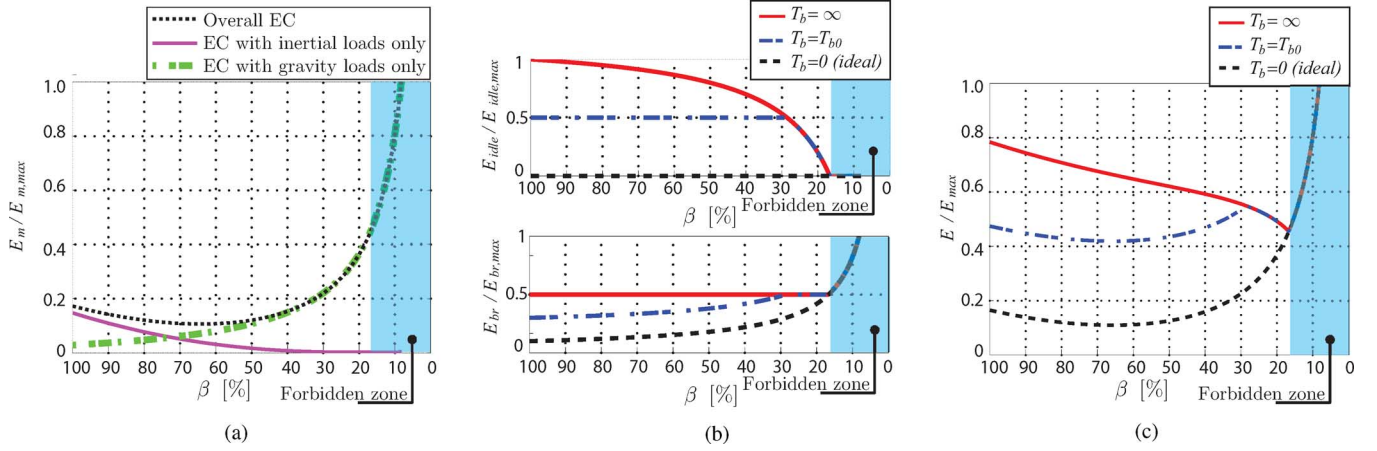


Fig. 5. Contribution to the actuators' EC as function of the override ratio [refer to (40), (41)]. Normalized graphs. (a) Energy term  $E_m$ . Contribution of gravitational and inertial loads. (b) Energy terms  $E_{idle}$  (upper graph) and  $E_{br}$  (lower graph). (c) Overall actuators' EC.

EC due to either purely inertial or purely gravitational loads are highlighted. As for the EC due to inertial loads, it is a monotonically decreasing function of  $\beta$  which approaches zero when the motion time approaches infinity. On the other hand, the EC due to gravitational loads is a monotonically increasing function approaching infinity as the motion time increases. Therefore, the term  $E_m$  is given by the combination of two monotonic functions with opposite trends, highlighting that there may exist an EC minimum for some  $\beta$  value.

Fig. 5(b), in the upper graph, illustrates the energy required to keep the robot still. If  $T_b = 0$ , the idling energy is zero. On the other hand, if  $T_b = \infty$ , the idling energy monotonically decreases as long as the IR keeps still for a shorter amount of time for decreasing override ratios. At last, in case  $T_b = T_{b0}$ , the idling energy has a constant value  $E_{idle} = T_{b0} \cdot P_{me1}^*$  if the IR is idling for a time  $t > T_{b0}$ , whereas it coincides with the previous case otherwise (there is not enough time for the brakes to ever close before the end of the cycle).

Fig. 5(b), in the lower graph, depicts the energy required to keep the brakes opened. If  $T_b = 0$ , the brake energy term monotonically increases for decreasing override ratios as long as the brakes are opened only if the IR is moving. On other hand, if  $T_b = \infty$ , the brake energy term has a constant value  $E_{br} = t_{ah} \cdot P_{br}^*$ . At last, in case  $T_b = T_{b0}$ , the brake energy has the same constant value as the previous case, i.e.,  $E_{br} = t_{ah} \cdot P_{br}^*$ , if  $T_b + t_O/\beta > t_{ah}$ , whereas it monotonically increases otherwise, i.e.,  $E_{br} = (T_{b0} + t_O/\beta) \cdot P_{br}^*$ .

Finally, Fig. 5(c) depicts the overall EC as function of  $\beta$  for the three aforementioned brake release times. In all the depicted cases, it is possible to define a single  $\beta$  value that minimizes the system EC. In parallel, both (40) and Fig. 5(a), trivially show that an early brake release is always beneficial in terms of energy consumption.

#### IV. MINIMIZATION OF ENERGY CONSUMPTION ON ROBOTIC CELLS FOR AUTOMOBILE PRODUCTION

The described approach has been used to model an existing robot cell located at the Mercedes-Benz plant in Sindelfingen, Germany. Differently from Section III, the EC has been simulated using the complete model described in Section II and no

energy term has been neglected. The input trajectories (reference as well as scaled ones) have been generated using the Realistic Controller Simulator (RCS) module [34] and the interpreter of the manufacturer's programming language (KRL, see, e.g., [35]). As for the robot programs, they are partly scaled, affecting only the movement from the last process point to the standard waiting *home position*. The actual process standstills of the robot applications are then experimentally measured, and both actual tool and load data as used in the real plant are acquired. Finally, with reference to cell global cycle time,  $t_{gc}$ , the overall cell EC is computed as

$$E_{cell}(\beta, T_b) = \sum_{k=1}^m \int_0^{t_{gc}} P_{ac,k}(\beta_k, T_b, t) dt \quad (42)$$

where  $P_{ac,k}$  and  $\beta = [\beta_k]^T$ , for  $k = 1 \dots m$  are, respectively, overall electrical input power of the  $k$ th IR and vector of the override ratios for the  $m$  robots in the cell. Note that the same value for the parameter  $T_b$  has been assumed for each robot.

In the following, a generic variable  $t_{j,k}$  will refer to the time period  $j$  performed by the  $k$ th robot.

##### A. Analysis of the Multirobot Cell

As a case study, a cell consisting of  $m = 4$  IR 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  $R_{2...4}$  are completed in a shorter time span. Hence, denoting  $t_{c,k}$  as the single robot cycle time, the global cycle for the given process is

$$t_{gc} = t_{c,1} > t_{c,2}, \dots, t_{c,4}. \quad (43)$$

In particular, Fig. 6 represents the various states of each robot  $R_k$  as a function of time. For each IR state, only binary values are allowed such that a positive value for an ordinate indicates that the corresponding state is active, whereas a zero value indicates that the corresponding state is inactive. Three states are relevant for time-scaling, namely:

- APPL.RUN, which represents a state when a robot application is running. In Fig. 6, a continuous line forms a block of duration  $t_{c,k}$  for the particular robot's program, whereas



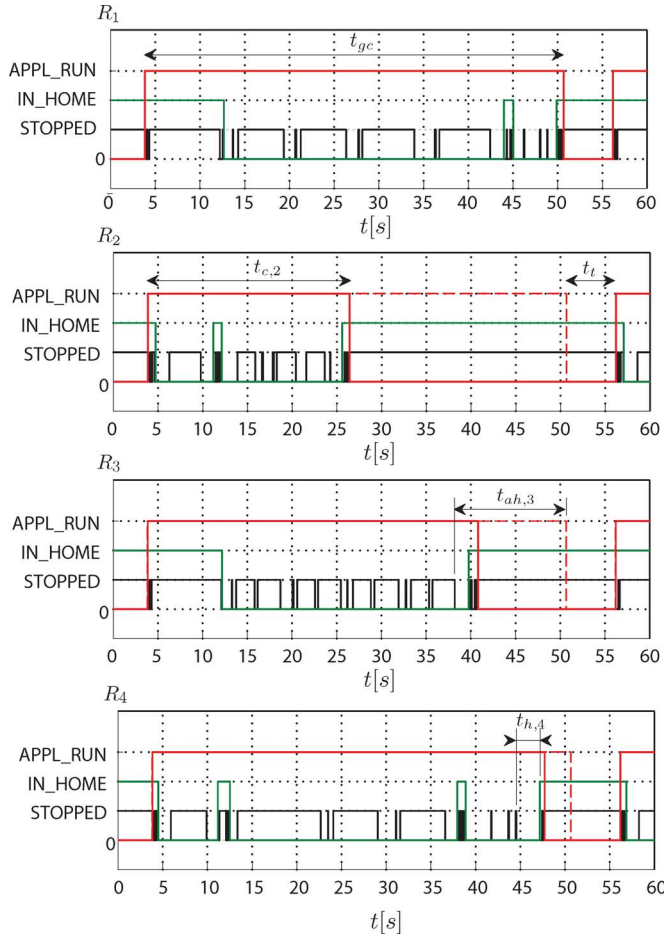


Fig. 6. Process states within the robot cell for the  $i$ th process and for  $R_1 \dots R_4$  robots (Note: bit states are differentiated by block heights).

the dashed line (refer, for instance, to robot  $R_2$ ) represents the duration of the global cycle.

- IN\_HOME, which represents a state when the robot is in its default waiting *home position*.
- STOPPED, which represents the state when the robot manipulator is stationary for any reason, e.g., when welding or waiting.

In addition, a transfer duration  $t_t$  is introduced, representing the time frame during which the car body is transferred between the robot cells ( $t_t$  is typically constant). Thus, whenever the robot is in the IN\_HOME state, it is also in the STOPPED state, but not always *vice versa*. 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 of the robot task. *Vice versa*, a time span between the falling edge of last STOPPED block within the APPL\_RUN block and the rising edge of the IN\_HOME block represents the duration of the movement from the last process point to the default home position. This movement is also recognized as the *last cycle movement* of the robot  $R_k$ , having a duration  $t_{h,k}$ . Excluding all the process-related standstills,  $t_{h,k}$  is typically around 20% of the total movement

duration [2]. As an example, Fig. 7(a–c) depicts the pose of robots  $R_2, \dots, R_4$  at the last process point and at the HOME position. This movement phase is the one to be optimized by means of time scaling.

In this particular case study, the robot  $R_1$  appears to be the *leading one*. Since the selected robots do not share a common workspace, the available time  $t_{ah,k}$  for the last cycle movement can be calculated as follows:

$$t_{ah,k} = t_{gc} - t_{c,k} + t_{h,k}. \quad (44)$$

The override ratio  $\beta_k$  of robot  $R_k$  might be chosen as any positive real number. Nonetheless, the duration of the robot application will be within the global cycle and no kinematic constraints will be violated, only if  $\beta_k$  is chosen within the vector of validity boundaries,  $\beta_{k,\text{valid}}$ , which is defined as

$$\beta_{k,\text{valid}} = \left[ \frac{t_{h,k}}{t_{ah,k}}, 1 \right]. \quad (45)$$

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

$$t_{o,k} = t_{c,k} - t_{h,k}, \quad (46)$$

$$t_{s,k} = \frac{t_{h,k}}{\beta_k}. \quad (47)$$

Concerning the brake release time delay, the value  $T_b$  can be pre-programmed, but cannot be changed dynamically (at least, for many common robot controllers). As said, the lower the  $T_b$  value, the lower the amount of energy that is required to counteract gravity loads. Consequently, the higher the amount of brake usage, the shorter its life [2].

As a further step, the waiting duration with opened brakes ( $\gamma_{br} = 1$ ) and closed brakes ( $\gamma_{br} = 0$ ) is calculated. Two cases are considered in the simulation model. Concerning the first case, if the scaled program does not violate the cycle time, that is

$$\beta_k \in \beta_{k,\text{valid}} \quad (48)$$

then the following conditions hold:

$$\left. \begin{aligned} t_{\gamma 1,k} &= t_r \\ t_{\gamma 0,k} &= 0 \end{aligned} \right\} \quad \text{if } T_b \geq t_r, \quad (49)$$

$$\left. \begin{aligned} t_{\gamma 1,k} &= T_b \\ t_{\gamma 0,k} &= t_r - T_b \end{aligned} \right\} \quad \text{if } T_b < t_r$$

where  $t_r = t_{gc} + t_t - t_{o,k} - t_{s,k}$ . As for the second case, if  $0 < \beta_k < (t_{h,k})/(t_{ah,k})$ , then the scaled program duration is longer than global cycle time and the following conditions hold:

$$\left. \begin{aligned} t_{\gamma 1,k} &= t_t \\ t_{\gamma 0,k} &= 0 \end{aligned} \right\} \quad \text{if } T_b \geq t_t, \quad (50)$$

$$\left. \begin{aligned} t_{\gamma 1,k} &= t_d \\ t_{\gamma 0,k} &= t_t - T_b \end{aligned} \right\} \quad \text{if } T_b < t_t.$$

If (48) holds, then the total simulation duration will always be  $t_{gc} + t_t$  and the numerical comparison is done for the actual task and waiting time. If, however, (48) is not fulfilled, the cycle time is exceeded (i.e., the IR enter the forbidden zone depicted in

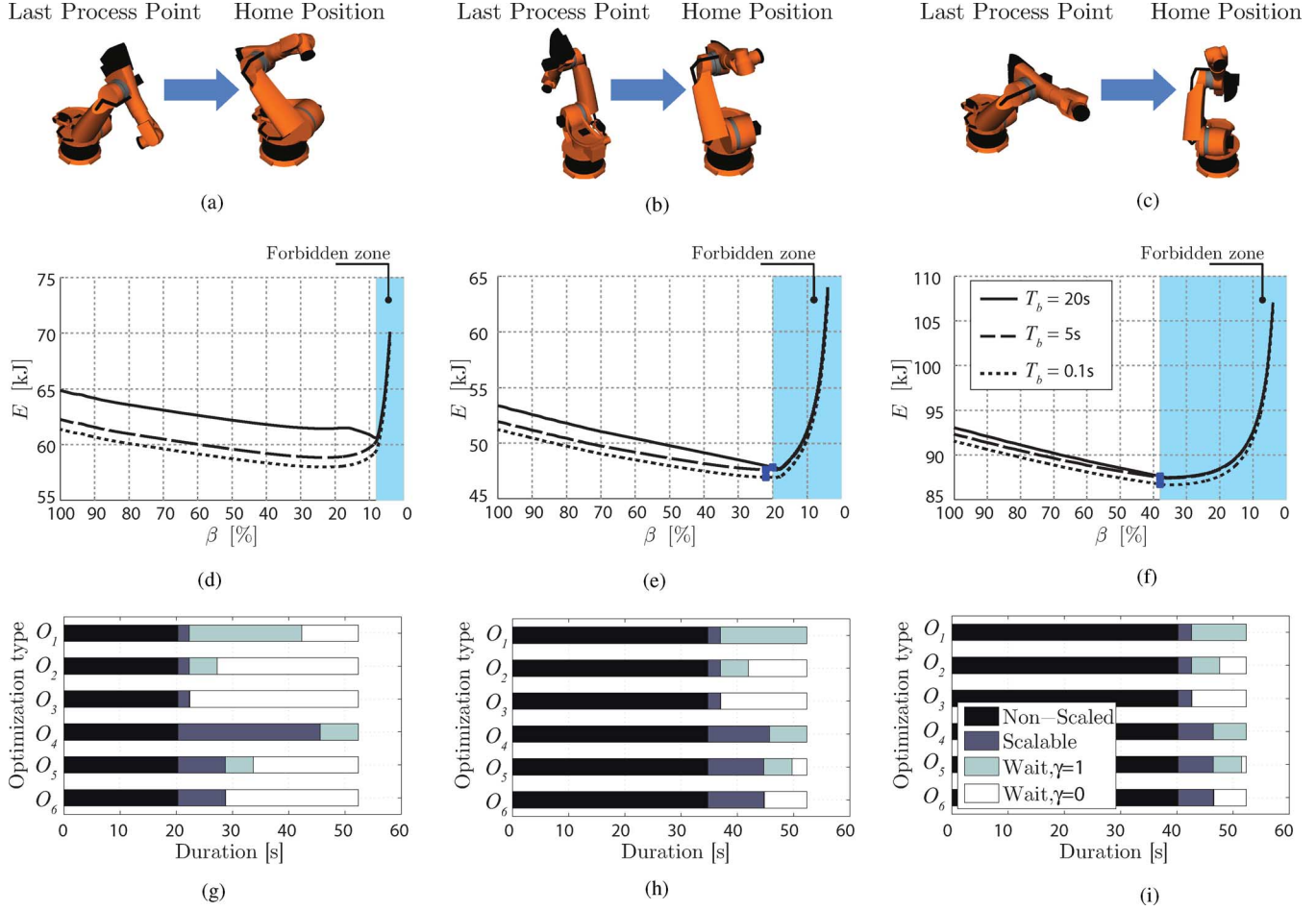


Fig. 7. (a–c) Robot last and home positions concerning scaled movements, (d–f) energy signatures, and (g–i) program/standstill comparison. (a) Scalable movement frame of robot  $R_2$ . (b) Scalable movement frame of robot  $R_3$ . (c) Scalable movement frame of robot  $R_4$ . (d) EC signature of robot  $R_2$ . (e) EC signature of robot  $R_3$ . (f) EC signature of robot  $R_4$ . (g) Program duration frames of robot  $R_2$ . (h) Program duration frames of robot  $R_3$ . (i) Program duration frames of robot  $R_4$ .

Fig. 5), and the increase of EC is mostly due to longer program durations, the maximum idle time after reaching the last *home position* being the transfer duration  $t_t$  and the next cycle starting right after. Finally, the energy-optimal override ratio,  $\beta_{k,opt}$ , is introduced, corresponding to the minimum energy consumption of a robot application within a cycle time, so that

$$\min E_k = E_k(\beta_{k,opt}), \quad \text{where } \beta_{k,opt} \in \beta_{k,valid}. \quad (51)$$

### B. Optimization Results

In the following, six optimization approaches have been compared, which combine energy-optimal time scaling and various brake handling modes, namely:

- $O_1$ : Reference mode where  $T_b = 20$  s and  $\beta_k = 1$ .
- $O_2$ : Reduction of the brake release time to  $T_b = 5$  s, while maintaining  $\beta_k = 1$ .
- $O_3$ : Dynamic brake release, namely,  $T_d = 0.1$  s, while maintaining  $\beta_k = 1$ .
- $O_4$ : No brake release time reduction, namely,  $T_b = 20$  s, and energy-optimal time scaling with  $\beta_k = \beta_{k,opt}$ .
- $O_5$ : Reduction of the brake release time to  $T_b = 5$  s and energy-optimal time scaling with  $\beta_k = \beta_{k,opt}$ .

- $O_6$ : Dynamic brake release  $T_b = 0.1$  s and energy-optimal time scaling with  $\beta_k = \beta_{k,opt}$ .

As previously said, due to various production standstills, a shorter brake release time lead to an increase in the brakes switching cycles, whose consequences have to be considered in real plants. For instance, [33] provides a source (press release) from an IR manufacturer stating that the maximum brake switching cycles within the warranty limits have recently been increased to 5 million. In the present case study, an IR lifetime of 14 years and 5 million brake switches are considered, ultimately leading to  $T_b = 5$  s or higher, according to the analysis described in [2]. Here, the *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 (i.e., at the end of the cycle) the brakes are set to release right after reaching the home position. On the other hand, in case of process standstills (i.e., intermediate standstills), the brakes are not released in order not to exceed the maximum switching cycles set by the manufacturer. A minimum time delay  $t_d = 0.1$  s is still considered. In particular, approaches  $O_1$ ,  $O_2$  and  $O_3$  consider only various brake release times, whereas modes  $O_4$ ,  $O_5$ , and  $O_6$  represent a combination of the different brake release times with the energy-optimal time scaling approach. Fig. 7(d)–(f) illustrate

TABLE II  
POSSIBLE OVERRIDE RATIOS FOR THE LAST MOVEMENT  
TO THE HOME POSITION

Robot	Usage [ $t_c/t_{gc}$ ]	$\beta_{min}$ [%]	$\beta_{opt}$ [%] $t_d = 20s$	$\beta_{opt}$ [%] $t_d = 5s$	$\beta_{opt}$ [%] $t_d = 0.1s$
$R_1$	1	100	100	100	100
$R_2$	0.48	8	8	24	24
$R_3$	0.79	20	20	22	22
$R_4$	0.91	38	38	38	38

TABLE III  
EC OF THE MULTIROBOT CELL OVER THE CYCLE TIME  $t_{gc}$

R	Reference [kJ]	Optimization [%]					
		2	3	4	5	6	
$R_1$	68.5	-0.2	-1.2	0	-0.2	-1.2	
$R_2$	64.9	-4.0	-5.4	-6.6	-9.3	-10.6	
$R_3$	53.4	-2.7	-4.0	-10.4	-10.8	-12.1	
$R_4$	93.0	-0.8	-1.6	-5.9	-6.0	-6.8	
$Cell$	279.8	-1.7	-2.9	-5.4	-6.2	-7.3	

EC signatures of particular robots during global cycle time  $t_{gc}$  and transfer time  $t_t$ . Each type of curve (continuous, dashed, dotted) represents a different  $T_b$  selection. Similarly to Fig. 5, the colored area represents the forbidden zone where (48) is not fulfilled. The numerical results of approaches  $O_1$ ,  $O_2$ , and  $O_3$  can be found by evaluating the first value on the respective curve (i.e., when  $\beta = 100\%$ ). In parallel, the numerical results of approaches  $O_4$ ,  $O_5$ , and  $O_6$  are illustrated with a dot on the respective curve type and can be found by assessing the EC minimum obtained for  $\beta_k = \beta_{k,opt}$  as in (51).

Referring to Figs. 6 and 7, some analysis of the resulting EC signatures may be done. The robot  $R_4$  has a relatively low available time  $t_{ah,4}$ , its EC minimum being at the longest allowable duration (achieved for  $\beta_4 = (t_{h,4})/(t_{ah,4})$ ). In contrast,  $R_2$  presents an increase of the EC at a low  $\beta_2$  values, due to a largely extended movement duration and, therefore, an increased contribution to EC due to mechanical brakes and gravity loads. As for the EC for robot  $R_2$  obtained by means of approach  $O_4$ , two local minima are found, where the peak between both local minima represents a state where  $T_b = t_r$  [see (49)]. In general, if  $t/\beta_k$ , for  $\beta_{k,opt} \in \beta_{k,valid}$ , never reaches  $T_b = t_r$ , then there is no second local minimum. The rapid EC increase in the forbidden zone in all cases is due to increased total movement and standstill duration that is greater than  $t_{gc} + t_t$ . In Table II, a robot usage ratio  $t_{c,k}/t_{gc,k}$  is shown. The lower the initial ratio, the higher time scaling range may be evaluated, and, therefore, the optimum selected. Table II also shows the numerical values of the minimum,  $\beta_{k,min}$ , and optimal,  $\beta_{k,opt}$ , override ratios at various  $T_b$  values. The numerical EC results for each robot per optimization method are summarized in Table III, whereas the breakdown of various durations  $t_{o,k}$ ,  $t_{s,k}$ ,  $t_{\gamma 1,k}$ ,  $t_{\gamma 0,k}$  is shown in Fig. 7(g)–(i), respectively. In order to calculate the annual percentage savings, typical production standstills must be considered [2].

## V. CONCLUSION

This paper discusses an energy consumption optimization method for multirobot 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 relate to 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, resulting in robots having excessive idle times after task completion. Following the proposed methods, simulation results show that those idle times can be used to achieve significant energy savings while still taking into account robot dynamic 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 *Computer Aided Robotic Tools* for initial production planning. Finally, the results show a potential energy saving up to 12,1% for the single industrial robot and 7,3% for the overall cell.

## REFERENCES

- [1] *Energy Efficiency Plan*. Brussels: E. Commission, 2011.
- [2] D. Meike and L. Ribickis, "Energy efficient use of robotics in the automobile industry," in *Proc. IEEE 15th Int. Conf. Advanced Robot.*, 2011, pp. 507–511.
- [3] R. Saidur, "A review on electrical motors energy use and energy savings," *Renew. Sust. Energ. Rev.*, vol. 14, pp. 877–898, 2010.
- [4] R. Visinka, "Ch. 2 - Energy Efficient Three-Phase AC Motor Drives for Appliance and Industrial Applications," in *Green Electronics/Green Bottom Line*, L. H. Goldberg and W. Middleton, Eds. Woburn, MA: Newnes, 2000.
- [5] A. Yang, J. Pu, C. Wong, and P. Moore, "By-pass valve control to improve energy efficiency of pneumatic drive system," *Control Eng. Pract.*, vol. 17, pp. 623–628, 2009.
- [6] O. Maimon, E. Profeta, and S. Singer, "Energy analysis of robot task motions," *J. Intell. Robot Syst.*, vol. 4, pp. 175–198, 1991.
- [7] H. Diken, "Energy efficient sinusoidal path planning of robot manipulators," *Mech. Mach. Theory*, vol. 29, no. 6, pp. 785–792, 1994.
- [8] J. Park, "Motion profile planning of repetitive point-to-point control for maximum energy conversion under acceleration conditions," *Mechatronics*, vol. 6, no. 6, pp. 649–663, 1996.
- [9] E. Sergaki, G. Stavrakakis, and A. Pouliezios, "Optimal robot speed trajectory by minimization of the actuator motor electromechanical losses," *J. Intell. Robot. Syst.*, vol. 33, pp. 187–207, 2002.
- [10] M.-S. Huang, Y.-L. H., and R. F. Fung, "Minimum-energy point-to-point trajectory planning for a motor-toggle servomechanism," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 2, pp. 337–344, 2012.
- [11] G. Field and Y. Stepanenko, "Iterative dynamic programming: An approach to minimum energy trajectory planning for robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, vol. 3, pp. 2755–2760.
- [12] C. H. Kim and B. K. Kim, X.-J. Jing, Ed., "Minimum-energy motion planning for differential-driven wheeled mobile robots," *Motion Planning*, pp. 192–226, 2008, ISBN: 978-953-7619-01-5, InTech.
- [13] A. Barili, M. Ceresa, and C. Parisi, "Energy-saving motion control for an autonomous mobile robot," in *Proc. IEEE Int. Symp. Ind. Electron., ISIE '95*, vol. 2, no. 10–14, pp. 674–676.
- [14] M. Yongguo, L. Yung-Hsiang, Y. Hu, and C. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 507–522, 2006.

- [15] F. Roos, H. Johansson, and J. Wikander, "Optimal selection of motor and gearhead in mechatronic applications," *Mechatronics*, vol. 16, pp. 63–72, 2006.
- [16] T. Izumi, H. Zhou, and Z. Li, "Optimal design of gear ratios and offset for energy conservation of an articulated manipulator," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 3, pp. 551–557, 2006.
- [17] A. Vergnano, C. Thorstensson, B. Lennartson, P. Falkman, M. Pellicciari, C. Yuan, S. Biller, and F. Leali, "Embedding detailed robot energy optimization into high-level scheduling," in *Proc. IEEE Int. Conf. Autom. Sci. Eng., CASE'10*, 2010, pp. 386–392.
- [18] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano, "A minimal touch approach for optimizing energy efficiency in pick-and-place manipulators," in *Proc. IEEE 15th Int. Conf. Adv. Robot.*, 2011, pp. 100–105.
- [19] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-level scheduling of energy optimal trajectories," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 57–64, Jan. 2013.
- [20] D. Meike, M. Pellicciari, G. Berselli, A. Vergnano, and L. Ribickis, "Increasing the energy efficiency of multi-robot production lines in the automotive industry," in *Proc. IEEE Int. Conf. Autom. Sci. Eng., CASE*, 2012, pp. 700–705.
- [21] A. Kobetski and M. Fabian, "Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming," *J. Discrete Event Dynamic Syst.*, vol. 19, no. 3, pp. 287–315, 2009.
- [22] S. B. Moon and S. Ahmad, "Time-optimal trajectories for cooperative multi-manipulator systems," *IEEE Trans. Syst., Man, Cybern.-Part B*, vol. 27, pp. 343–353, 1997.
- [23] J. Hollerbach, "Dynamic scaling of manipulator trajectories," in *Amer. Control Conf.*, San Francisco, CA, USA, 1983, pp. 752–756.
- [24] J. Moreno-Valenzuela, "Time-scaling of trajectories for point-to-point robotic tasks," *ISA Trans.*, vol. 45, no. 3, pp. 407–418, 2006.
- [25] M. Grotjahn, M. Daemi, and B. Heimann, "Friction and rigid body identification of robot dynamics," *Int. J. Solids and Structures*, vol. 38, no. 1013, pp. 1889–1902, 2001.
- [26] E. Oliva, G. Berselli, and F. Pini, "Dynamic identification of industrial robots from low-sampled data," *Appl. Mech. Mater.*, vol. 328, pp. 644–650, 2013.
- [27] C. Mi, "Modelling of Iron Losses of Permanent Magnet Synchronous Motors," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 2000.
- [28] J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2005.
- [29] E. Kiel, *Drive Solutions—Mechatronics for Production and Logistics*. New York, NY, USA: Springer, 2008.
- [30] "Ec 60034-30 efficiency classes of single-speed, three-phase, cage-induction motors," 2012. [Online]. Available: <http://www.iec.ch/>
- [31] D. Meike, C. Hansen, J. Öltjen, and T. Ortmaier, "Enhanced approach for energy-efficient trajectory generation of industrial robots," in *Proc. IEEE 8th Int. Conf. Autom. Sci. Eng.*, 2012, pp. 1–7.
- [32] B. Bose, *Modern Power Electronics and AC Drives*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

- [33] P. Klüger, "Die grüne robotik," *Automatica Special*, pp. 29–34, 2012.
- [34] Fraunhofer IPK, 2009. [Online]. Available: <http://www.realistic-robot-simulation.org/>
- [35] KUKA Roboter, 2011. [Online]. Available: <http://www.kuka-robotics.com/>



robot systems.

**Davis Meike** received the M.Sc. degree in electrical engineering from Riga Technical University, Riga, Latvia, in 2010.

He is presently a Researcher at the Daimler AG Branch of Automation and Control Engineering, team Robotics and Technology Units, Sindelfingen, Germany. He is currently working on his Ph.D. dissertation on energy efficient robotics. His research interests include new production planning and optimization methods and intelligent control systems, smart factories, and cooperative industrial



**Marcello Pellicciari** joined the University of Modena and Reggio Emilia as a Researcher in the field of engineering design methods in 2001. He is currently Chair of the Engineering Design course at the same university and member of the IEEE technical committee on sustainable production automation. His current research interests are integrated design and virtual prototyping of mechatronic systems, industrial robotics, and sustainable manufacturing.



**Giovanni Berselli** received the Laurea degree (*cum laude*) in mechanical engineering from the University of Modena, Modena, Italy, and the Ph.D. degree in mechanics of machines at the University of Bologna, Bologna, Italy, in 2004 and 2009, respectively.

He is currently an Assistant Professor at the University of Modena. He was a Research Assistant with the Department of Mechanical Engineering, Monash University, Melbourne, Australia, and with CEIT, Escuela Superior de Ingenieros de la Universidad de Navarra, San Sebastian, Spain. His research activity

is focused on the development of engineering methods for sustainable robotics, and on the design of compliant robotic systems.