Optimization of Power Consumption for Robotic Lines in Automotive Industry

Pavel Burget, Libor Bukata, Přemysl Šůcha, Martin Ron, and Zdeněk Hanzálek

1 Introduction

The utilisation of robots in production lines has become a very important aspect to increase the productivity, throughput and efficiency of the production. Especially car manufacturers have been investing big effort in obtaining reliable, precise and high-throughput robotic production lines. Hand in hand with devoting higher ratio of the production work to the robots and thus increasing the number of robots participating in the production, the amount of energy consumed by the robots increases as well. Therefore it is of great importance to search for ways how to improve the energy efficiency of the robot operations. In this chapter the focus is put on robotic welding lines, which already exist and which have been in production. Because of this reason there are only limited possibilities in adding additional sensors or in performing changes in the robotic programs and in the programs of the superordinate controllers. However, such existing lines can still be improved in terms of their energy consumption. This contribution concentrates on methods how to optimise the robotic operations and how to get the necessary energy models of the robots that are needed for the optimisation. All this while keeping in mind the above stated requirements to utilise the existing production line infrastructure. Moreover, the underlying optimisation model together with the energy model of the robots are general enough to be used also during the design phase of new production lines.

The core of this contribution is a novel mathematical formulation of the energy optimisation problem for robotic lines. Contrary to the existing works the proposed solution considers different trajectories of robots, gravity and order of robot operations from the global point of view of the whole robotic cell. In fact, it may also be enhanced to a series of cells but it is out of the scope here. Moreover, the presented mathematical formulation takes into account the robots' power saving modes such as staying on brakes or "falling asleep", to which robot in a stationary position

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Control Engineering, e-mail: {burgetpa,bukatlib,suchap,ronmarti,hanzalek}@fel.cvut.cz

can switch to save even more energy. The optimal solution to the problem is the one which is both the most energy-efficient and meets the desired production cycle time.

The basic characteristics of the robots with respect to their energy consumption is their so-called energy function, which represents the dependency of the consumed electrical energy on the duration of given robotic operations. Such an energy function may be obtained using electrical power measurements on the real robots, by simulations or by analytical computations based on a physical model of the robots. Moreover the robotic operations must be clearly separated in order to get their boundaries to be entered into the optimisation model, as well as to be able to compare the energy function of the individual robots with their real behaviour.

The physical modelling of the robots is based on graphically-oriented computer-aided concept that exploits CAD software such as NX or Solidworks and simulation environment Matlab Simulink with SimMechanics and SimPowerSystems libraries. These software tools are used for the composition of a dynamical simulation model that represents both mechanical robot structure and robot drives during the robot motions. Thus the power needed for the robot movements can be obtained and the energy functions can be calculated.

The robotic operations may also be simulated in another environment, which is used to design and simulate complete robotic lines such as Process Simulate. Next to the design of the robot trajectories it is possible to simulate the robot controller itself if an appropriate Robot Controller Software (RCS) and Realistic Robot Simulation (RRS) modules are available. Recently, RCS and RRS modules that allow simulating the energy consumption have been provided by robot manufacturers.

1.1 Contribution

In this work the global optimisation of the robotic lines is devised with respect to the identified energy aspects which resulted from measurements and simulations. Compared with the existing works such as [24, 25, 22, 23], the presented solution is more general by considering the robot power saving modes such as brakes, bus-power-off or hibernation, and different locations are taken into account where a robot operation can be performed. Moreover, the presented formulation enables the robotic line designers to specify path alternatives, i.e. selecting the best order in accordance with precedence relations. The achieved results have revealed that a significant energy saving is possible.

An important part of the optimisation model is the energy function of the robot movements, which allows choosing the optimal speed of the robotic operations and thus minimising the energy consumption of the robotic cell not only from a local point of view but also from the point of view of the cell or even series of cells. This work presents several ways how to obtain the energy function.

1.2 Related works

The current research on energy optimisation of robotic lines can be categorised into two groups. The first one is the optimisation of individual robot trajectories with respect to physical limitations of robots and obstacles to be avoided. The second one, rarely occurring in the literature, is the optimisation of the robotic line as a whole using mathematical models. Both the groups are not necessarily disjunctive and there are a few papers dealing with both of the aspects.

As an example from the first group the following works are worth to be mentioned. In the work of Saramago et al. [16] both the time of the robot movement and mechanical energy consumed by actuators are taken into account. The multi-objective optimisation problem was solved by using the DOT program (Design Optimisation Tools Program) and tested on three and six degrees of freedom manipulator arms.

A real-time planning of energy efficient trajectories for the robot catching small flying objects was proposed by Lampariello [9]. The authors formulated the non-linear constrained optimisation problem, nevertheless, to be able to find good trajectories in a real-time the global planner was generalised using the learning methods, such as nearest neighbour, Support Vector Machines, and Gaussian process regression. The proposed approach has shown to be efficient on the ball-catching task.

Michna *et al.* [12] developed an algorithm for the generation of time optimal trajectories for wheeled robots. The trajectory is interpolated by the cubic Hermite spline curves and a speed profile is determined by the algorithm. To accelerate the calculation of collision-free trajectories the authors propose to use neural network. Nevertheless, the energy consumption is not considered and the approach was only tested on a hypothetical example.

The following works perceive the robotic lines as a whole to find globally good solutions. In the work of M. Mashaei and B. Lennartson [11] an energy model of the Pallet-Constrained Flow Shop problem was formulated to find an optimal switching control strategy leading to the desired throughput and minimal energy consumption. Idle states of machines were also taken into account to reduce energy consumption if the machine is not working. However, the model requires a line with special structure, i.e. closed-loop pallet system, and therefore it is not generally applicable to the robotic lines.

There are a few similar papers [24, 25, 22, 23] focusing on both the local and global optimisation of the robotic lines. For example, in the work of Wigström et al. [24] a physical model of a robot with AC synchronous motors is created and optimal control problem, determining how to control the robot moving along the specified partial trajectory in an energy efficient way, was solved using Dynamic Programming. Nevertheless, the geometry path was fixed and the initial time optimal trajectory obtained from ABB Robot Studio was required. Afterwards, the locally optimised trajectories were used as an input for the global solver (Mixed Integer Non-Linear Programming) to find a solution that is energy efficient and satisfying demanded production cycle time. Although the model is the first model considering the global energy aspects, there are a few drawbacks limiting the possible energy saving – the robot power saving modes are not taken into account and different positions of robots during the work are not considered. The same authors provide more details about the formulation in [23]. As a verification of the energy-aware solution [25] suggests to use Hybrid Cost Automata .

Riazi *et al.* in [14] combine the optimisation of individual trajectories with ordering the robot operations. The decrease of energy consumption during movements is achieved by minimising the sum of square roots of accelerations over the trajectory. Worth mentioning is the fact that no model of the robot is needed because the acceleration vector is obtained by sampling the existing movements and the optimised trajectories are uploaded back into the robot afterwards.

As part of the movement optimisation in order to reduce the energy consumption it is necessary to calculate, measure or at least assess the energy consumption of the robot under different circumstances. Several methods exist that are based on mathematical analysis, i.e. on modelling the kinematics and dynamics of the robots such as in [3], [17] or [18] where a specific expression of the energy consumption equation in dependence on a given robot trajectory is presented. Papers such as [4], [5], or [21] focus on processing and analysing real data obtained from physical measurements.

The way how measurements can be done differs if the robot is in a laboratory environment or if the measurements must be done in a production environment on a robot that is usually part of a robotic cell. In such a case pattern matching or machine learning techniques are used to process the data and identify the robot operations. In [6] region-based segmentation stemming from frequency analysis of the original signal is used. [10] relates to a state estimation and a corresponding energy audit of injection moulding machines and the focus is given to identifying the production state of the machine using a two-level neuron network to classify the states. Paper [20] focuses on pattern recognition of 1-D signal in industrial batch dryer with a goal to slice the measured data of pressure into time windows of the periodic batch processing intervals using supervised learning of a Takagi-Sugeno fuzzy model. Ron *et al.* in [15] use 1-D pattern matching with correlation and feature extraction techniques.

1.3 Outline of the chapter

The following sections provide the details about the individual parts of the energy optimisation problem. Specifically, Section 2 defines the problem formally and shows examples of a simple robotic line, schedule of operations and corresponding energy function for a robot movement.

Section 3 deals with different ways how to obtain data for modelling the energy function such as measurements of the power consumption on a robot in a laboratory, creation of a kinematic and dynamic model of a robot and its electrical drives to get an equation for the energy consumption, and how to simulate the robotic movements to get the energy function from simulation. A special attention is put to the identification of the robotic operations at a production line, which completes the whole picture in such a way that it is possible to evaluate the obtained energy functions of the robot movements with respect to their real energy consumption during production operations.

Section 4 shows how to use the energy functions to optimise the energy consumption. The mathematical model in terms of Integer Linear Programming is defined there and a way is proposed how to compute a lower bound using Langrangian relaxation. Section 5 describes the results of the optimisation, which has been performed on generated problem instances as well as on an industrial use case from Škoda Auto car manufacturer. Section 6 summarises the results and concludes the chapter.

2 Problem statement

The following aspects of energy saving at robots are crucial: (a) *selection of stationary positions* represented by different robot configurations, which take into account the robot energy consumption, (b) *power saving modes*, whose utilisation may result in significant energy demand decrease[1], (c) *trajectory selection and alternatives* again with respect to their energy consumption, and (d) *speed of the movement*, which is dual to the duration of the movement. A detailed analysis supporting this statement is provided in Section 3.1.

Energy aspects are illustrated by a simple robotic line depicted in Fig. 1. In this example the first robot takes the weldment, performs a welding operation, and puts it on the bench where the second robot takes it, carries out a welding operation, and

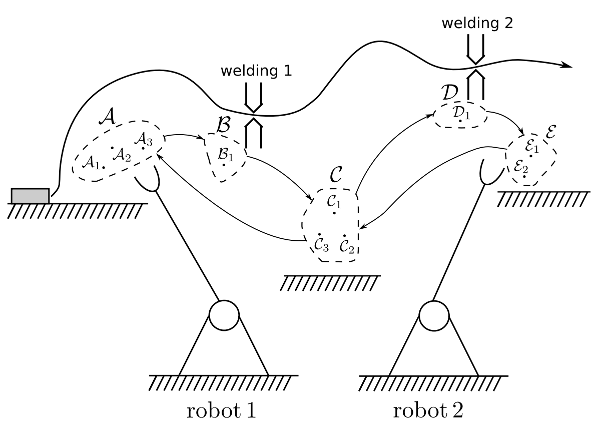

Fig. 1 Example of the line with two robots.

[1] The more energy-saving mode is the longer time is required to have the robot back in a ready-to-operate mode.

finally puts the weldment on the conveyor belt. In each subspace, denoted as a calligraphy letter without subscript, there are possible points (i.e. gun coordinates), in which the robot can conduct a task, e.g. welding, assembling, taking the workpiece, putting it on the bench, or handing it over to another robot. Between subspaces the robot can move in a direction indicated by the arrow that corresponds to the set of point-to-point movements. From the set no more than one movement is selected and

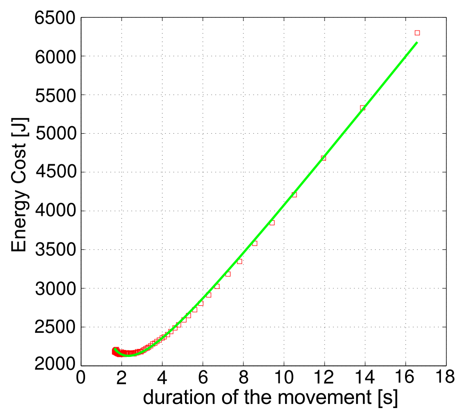

Fig. 2 Relation between the duration of the movement and energy consumption.

the duration of motion and required energy is determined according to the energy function obtained either from the measurement or from the energy model of the robot. In Fig. 2 the measured movements corresponding to points were interpolated with function

$$f_E^{s_i}(d_i) = a_{-1}d_i^{-1} + a_0 + a_1 d_i \tag{1}$$

where d_i is the duration of the movement, $f_E^{s_i}(d_i)$ consumed energy, and finally a_{-1}, a_0, a_1 are coefficients calculated by e.g. the Gauss-Newton algorithm. The subscript letter E means energy and superscript letter s represents the fact that the function is parametrized by the trajectory the robot moves along. The function was empirically proposed using the following ideas and supported by the measurement results from Section 3.1. As the duration tends towards 0_+ the power consumption increases to ∞, i.e. due to the first term $a_{-1}{d_i}^{-1}$. On the other hand, if the duration is very lengthy, only the gravity part can be considered and the consumption increases linearly, which is represented by the third term $a_1 d_i$. Finally, term a_0 is a constant offset of the function. The function is convex provided that coefficient $a_{-1} \geq 0$. It is also possible to use higher degrees of the approximation polynomial if it fits better the robot behaviour but it is subject to further evaluation in the particular case.

The alternatives are illustrated in Fig. 3. There are two possible paths that the robot can take and one of them is probably more energy efficient. However, both paths have a bit different timing (synchronisation between robots) and order of operations.

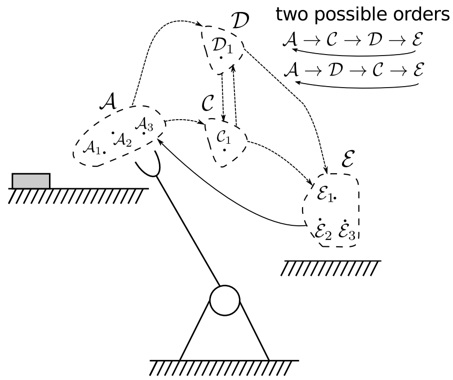

Fig. 3 Illustration of alternatives.

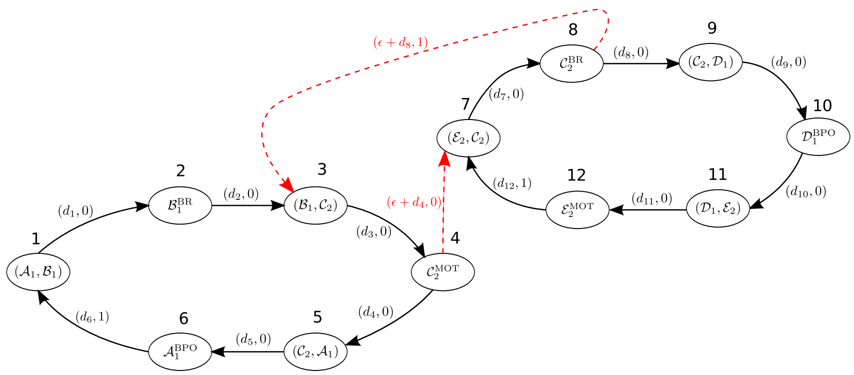

Fig. 4 Graph representation of the robotic line.

To demonstrate how the problem can be represented by a graph and how the final schedule looks like the robotic line from Fig. 1 was used for the example in Fig. 4, which shows the structure of the robotic line in terms of operations and movements. The red dashed arrows are synchronisation edges $\mathscr{L}$, i.e. time lags, that ensure the correct handover of the weldment to the second robot using a turntable. The black ones guarantee that operations and movements are performed in a desired order (orders in case of alternatives). All edges are weighted by length $L(e_{i,j})$ and

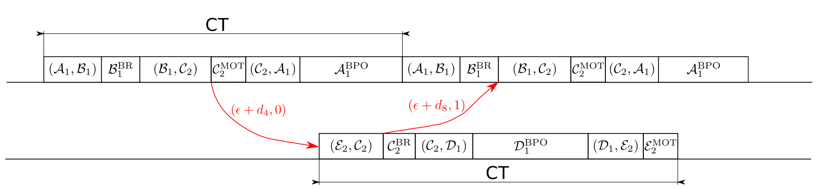

Fig. 5 An example of a schedule.

height $H(e_{i,j})$ where the length corresponds to the duration or time offset, and the height binds the previous or future cycles with the current one. For more information about time lags please refer to [7]. Finally, the selected robot positions, power saving modes (i.e. brakes – BR, motors – MOT, and bus-power-off – BPO), and movements are indicated in the graph nodes. One of the possible schedules is depicted in Fig. 5 where CT is the production cycle time.

As it can be seen the problem is similar to cyclic scheduling, however, there are a few differences. At fist, there is a synchronisation between robots, and as a consequence rotated schedules are not equally good as it is in cyclic scheduling. Secondly, durations are not fixed (energy functions) as it is the case of cyclic scheduling. If only one robot is taken into account, the problem is equivalent to the Travelling Salesman Problem with the exception of non-constant edge weights.

3 Energy function of the robot movements

There are several ways how the energy function of the robots can be obtained. The following subsections propose three ways, i.e. measurement at a real robot, physical modelling of the robot kinematics and dynamics, and simulation based also on a software model of the robot controller. Subsection 3.4 deals with a way how to identify the individual robotic operations automatically from the power measurement data obtained at a production cell with multiple robots. This procedure allows evaluating the energy model of the robot against the robot behaviour in the cell.

3.1 Power measurements

Detailed measurements were performed at industrial robot *KUKA KR 5 arc* for different speeds of movements, trajectories and robot positions to find out the energy-saving potential. Such a measurement cannot be done in a production cell typically. However, it is presented here to support the hypothesis about using the energy function. A brief description of the robot, which has been used, can be found in Table 1.

In the experiment the *static consumption* and *dynamic consumption* were measured. The static consumption is perceived as an amount of energy consumed by a robot in a stationary position. A non-moving robot can also get to a power saving mode (brakes, bus-power-off, hibernate) to save even more energy. The dynamic consumption corresponds to energy consumed during the robot movement.

The measured profile of active power is shown in Fig. 6. The left part of the graph (up to 80 seconds) can be used to evaluate the static consumption for the robot being held on brakes or motors. The rest of the graph is designated for the measurement of the dynamic consumption. For each speed, denoted as '`T2: X %`' where `X` is a relative speed of the robot, the sequence of movements (peaks in the graph) p_1 — p_2 — $p_1 \searrow p_3 \nearrow p_1$ is executed. 'p_i — p_j' is a movement between points p_i and

Table 1 Basic parameters of KUKA KR 5 arc (KUKA Industrial Robots [8]).

working range	1412 mm
maximal load	5 kg
weight	127 kg
idle power: held on brakes	cca 180 W
idle power: held on motors	cca 350 W
idle power: bus-power-off	cca 134 W

Table 2 Analysis of the measured data.

interval	t_1	t_2	Δt [s]	input power [W]	energy consumption [J]
Idle - brakes	25.0	30.0	5.0	180.3	901.3
Idle - motors	57.6	58.2	0.6	347.9	208.7
T2: 30 % (p_1—p_2)	192.7	194.8	2.1	649.5	1364.0
T2: 30 % (p_2—p_1)	197.7	199.8	2.1	641.4	1346.9
T2: 30 % ($p_1 \searrow p_3$)	202.5	205.0	2.5	583.8	1459.6
T2: 30 % ($p_3 \nearrow p_1$)	208.0	210.8	2.8	755.8	2116.2
T2: 50 % (p_1—p_2)	252.2	253.8	1.6	858.9	1374.2
T2: 50 % (p_2—p_1)	256.5	258.0	1.5	874.0	1310.9
T2: 50 % ($p_1 \searrow p_3$)	260.7	262.3	1.6	860.0	1376.0
T2: 50 % ($p_3 \nearrow p_1$)	265.2	267.2	2.0	1015.7	2031.3
T2: 70 % (p_1—p_2)	328.2	329.4	1.2	1204.1	1444.9
T2: 70 % (p_2—p_1)	332.0	333.4	1.4	1001.1	1401.5
T2: 70 % ($p_1 \searrow p_3$)	336.0	337.4	1.4	1102.9	1544.0
T2: 70 % ($p_3 \nearrow p_1$)	340.3	341.8	1.5	1420.6	2130.8
T2: 90 % (p_1—p_2)	385.0	386.3	1.3	1198.1	1557.5
T2: 90 % (p_2—p_1)	388.8	390.0	1.2	1233.9	1480.7
T2: 90 % ($p_1 \searrow p_3$)	392.5	393.8	1.3	1220.1	1586.1
T2: 90 % ($p_3 \nearrow p_1$)	396.5	398.0	1.5	1492.7	2239.0
T2: 100 % (p_1—p_2)	433.3	434.5	1.2	1270.7	1524.8
T2: 100 % (p_2—p_1)	437.0	438.2	1.2	1182.6	1419.2
T2: 100 % ($p_1 \searrow p_3$)	440.7	441.9	1.2	1411.2	1693.5
T2: 100 % ($p_3 \nearrow p_1$)	444.5	446.0	1.5	1494.5	2241.7

p_j which are at the same height, '$p_i \searrow p_j$' is a descending movement, and finally '$p_i \nearrow p_j$' is ascending one. Before each sequence the robot is moved from the home position, i.e. an initial position of the robot determined by the robotic cell designer, to p_1 and after the sequence from p_1 to the home position.

The measured data (see Fig. 6) were analysed and the results are presented in Table 2. From the static consumption point of view it can be deduced that it is pos-

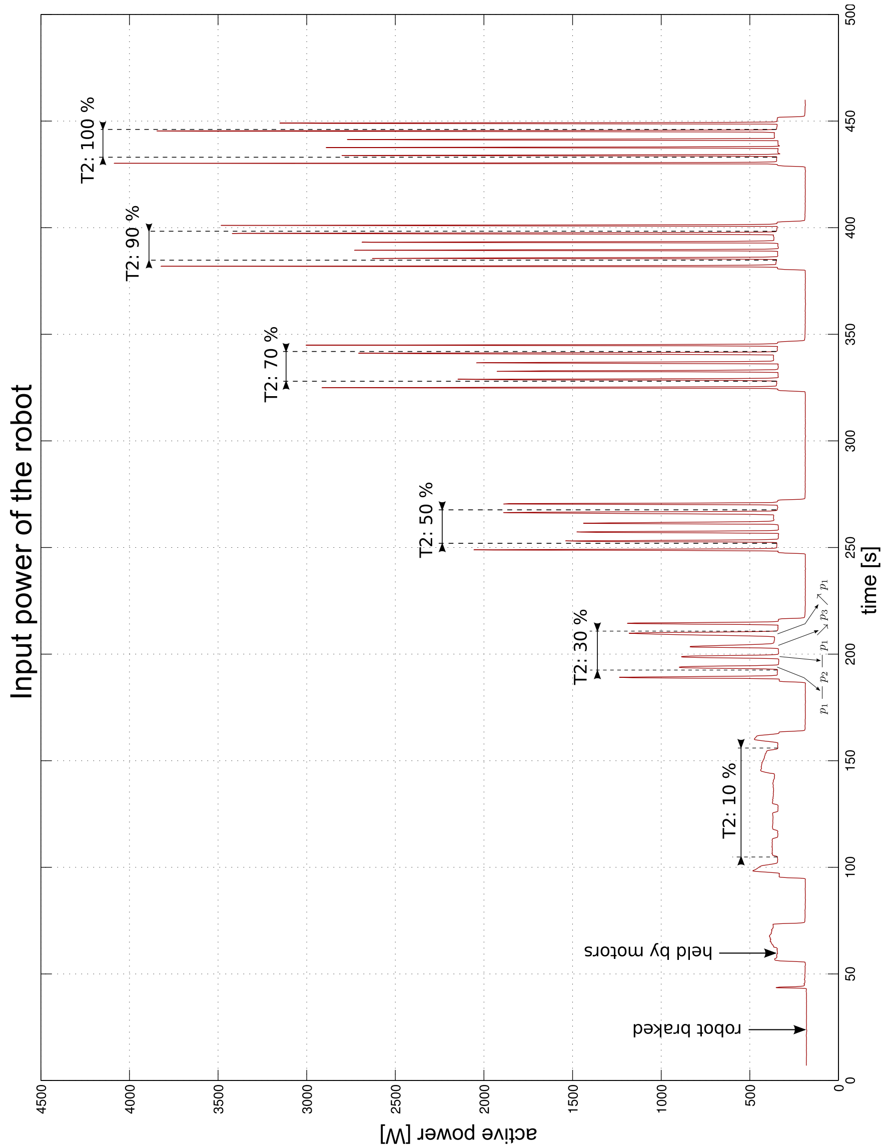

Fig. 6 Energy profile of the robot power consumption.

sible to save about 45 % energy if the robot is braked instead of being held in the position by motors. The difference could be even bigger if the robot was loaded or a less energy efficient configuration (i.e. a position of the robot) was selected. Another experiment, not mentioned before, was related to the measurement of how the robot configuration influences the power consumption. It was found out that the robot vertically stretched out required 344 W compared to 366 W for the robot horizontally stretched out. A relatively small difference was caused by using a small industrial robot without load. Along with the experiment, it was measured that the robot consumes 134 W and 30 W in the bus-power-off and hibernate modes, respectively.

To evaluate the effect of different movement speeds on the energy consumption the average input power and total consumption were calculated (last two columns in Table 2). From the results it is no surprise that the energy consumption was confirmed to be higher for $p_1 \searrow p_3$ movement than for $p_3 \nearrow p_1$ one. With respect to the speed of the robot it was shown that there is no need to consider too slow movements as the gravity part constitutes huge loss of energy. In a similar way, too fast movements increase the consumption dramatically while the duration of the movement decreases only little. For instance, there is a fall of about 6.3 % in energy for the p_1—p_2—p_1 movement if the relative speed is set to 70 % instead of 90 %.

3.2 Physical model of the robot

A complete modelling approach has been presented in [13], which deals with a description of the fully graphically-oriented computer-aided modelling and its mathematical analysis that is used for the determination of the robot energy consumption. The computer-aided modelling follows from a pure geometrical 3D model of the robot that is split by CAD software such as NX or Solidworks into particular robot components. They are supplemented with appropriate physical parameters like volumes, masses and moments of inertia. Such a component model, which represents physically a mechanical robot structure, can be converted into a simulation model operated in the Matlab/Simulink environment as shown in Fig. 7. The blocks in the figure have the following meaning. (A) is a World frame block, which represents the global reference frame; (B) is a Mechanism configuration block of general parameters used in the simulation; (D) is a Rigid transform block representing a transformation matrix that allows a following mechanical robot element to move with respect to the basic frame; (E) is a Link block, which represents the rigid body with its Denavit-Hartenberg frame and appropriate information about the body mass, moment of inertia related to its center of gravity; (F) is the Revolute joint block with one DOF, where the information about its angle, angular velocity, angular acceleration and actuating torque are obtained from a built-in joint sensor; and (G) is the next Link connected to the next revolute robot joint. The meaning of the other blocks is straightforward. The mechanical model from Fig. 7 is completed by blocks representing the robot drives, which allows getting equation (3) of torque equilibrium.

On the basis of the mathematical analysis (see [13]), equation (2) gives the input power for a single electrical motor with stator resistance R_{S_i} and current i_{q_i} for axis i in q-coordinate of the $d-q$ system.

$$P = \frac{3}{2}\sum_{i=1}^{n} R_{S_i} i_{q_i}^2 + \sum_{i=1}^{n} \omega_{m_i}(\tau_i + B_i \omega_{m_i} + J_i \frac{d}{dt}\, \omega_{m_i}) \qquad (2)$$

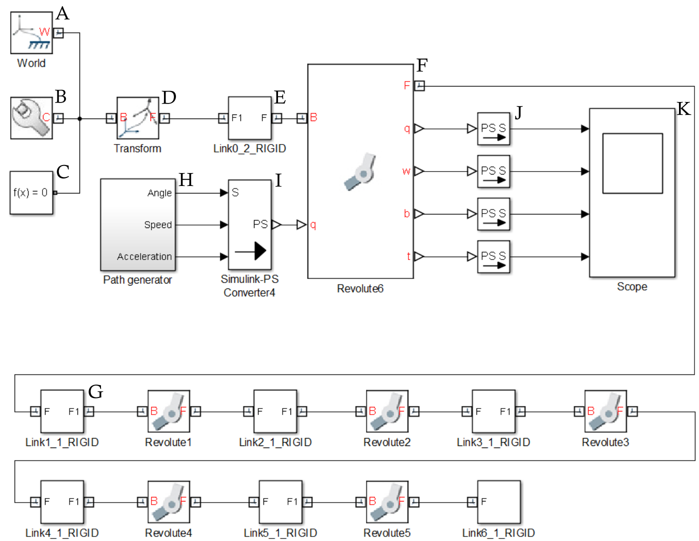

Fig. 7 Block diagram of robot structure in Simulink.

This equation is based on the equilibrium between electromagnetic torque and mechanical torque for the individual components, i.e. for the axes of the robot, as expressed in equation (3)

$$\tau_{e_i} = \tau_i + B_i\,\omega_{m_i} + J_i\,\frac{d}{dt}\,\omega_{m_i} \tag{3}$$

where J_i is the inertia and B_i is the friction of the motor and the load; ω_{m_i} is mechanical rotor speed related to electrical speed as $\omega_{m_i} = \omega_{e_i}/p$ considering p for a number of pole pairs; τ_i is a one load torque component of the torque vector τ from the dynamic model as expressed in equation (4); and τ_{e_i} is electromagnetic torque.

$$B(\vartheta)\,\ddot{\vartheta} + C(\vartheta,\dot{\vartheta})\,\dot{\vartheta} + g(\vartheta) = \tau \tag{4}$$

where $B(\vartheta)$ is an inertia matrix, $C(\vartheta,\dot{\vartheta})$ is a coefficient matrix of Coriolis and centrifugal force effects, $g(\vartheta)$ is a vector of gravitational effects, $\vartheta = [\,\vartheta_1\ \vartheta_2\ \cdots\ \vartheta_n\,]^T$ is a vector of joint angles and $\tau = [\,\tau_1\ \tau_2\ \cdots\ \tau_n\,]^T$ is a vector of torques acting on appropriate joints as shown e.g. in [19].

By integrating the total input power over the interval corresponding to the considered motion trajectory the robot energy consumption is obtained in equation (5), where Γ is the duration of the robotic movement and the other variables come from equations (2) – (4).

$$E = \int_0^{\Gamma} \left[\frac{3}{2} \sum_{i=1}^{n} R i_{q_i}^2 + \sum_{i=1}^{n} \omega_{m_i} \left(\tau_i + B_i\,\omega_{m_i} + J_i\,\frac{d}{dt}\,\omega_{m_i} \right) \right] dt \tag{5}$$

By evaluating the energy consumption for different durations of the motion it is possible to construct the energy function for the considered trajectory.

3.3 Simulated energy function

In some situations it is not possible to perform all necessary experiments and measurements with real equipment because of some physical or organizational limitations. As an example, a robotic cell being part of a regular production may be named. Thus, it may not be possible to change the robotic paths to perform any additional movements, which are not part of productive robotic operations, to identify a set of parameters relating to the dependency of the power consumption on a specific robot trajectory, to the robots' dynamic parameters etc.

Therefore a simulation environment can be used such as Tecnomatix Process Simulate. This environment contains robot controllers implemented according to the Robot Controller Simulation (RCS) specification, which means that the robot controllers perform Realistic Robot Simulation (RRS). One of the features of the recent Process Simulate version is the possibility to simulate power consumption of the robot movements. By summation of the total energy used by a robot for one particular movement performed repeatedly with different speed settings energy function $f_E^s(x)$ can be constructed – see equation (1).

3.4 Robotic operations

As mentioned above, knowing the robotic operations and the consumption of the energy for each of them helps evaluate the energy function for particular robot movements. Moreover, it is also used to group robotic operations together to form the activities that are used in the mathematical model to describe the behaviour of the robots (see section 4). According to this model, it is necessary to differentiate operations, which represent movement trajectories, and operations representing work such as welding, holding a part in a specific position, etc. A set of subsequent e.g. welding operations that are executed in a given order and are next to each other, is typically represented as a single activity because it is not expected that changing the trajectories between the welding points would mean any significant savings. Fig. 8 shows a sequence of operations of one robot in the welding cell. The blue horizontal bars represent the operations, which may be grouped into activities that are later for the optimisation represented as single dynamic or static activities (see section 4). The yellow horizontal bar is the length of one production cycle, i.e. the time after which the robot repeats its set of operations for the next part. Interval T_R shows the time remaining between the last operation and the beginning of the next production cycle. Thus, operation $o1$ forms activity 1, which is dynamic, and represents a movement of the robot to its first position. Operations $o2$ – $o6$ represent a set of

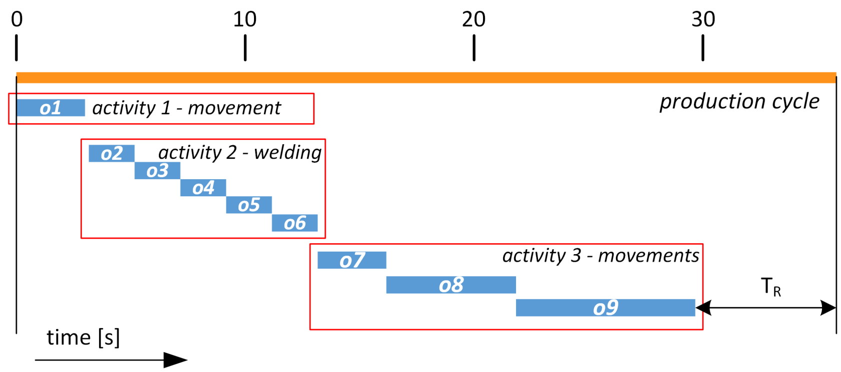

Fig. 8 Operations of one robot in the cell

welding operations and short movements between the neighbouring welding spots and forms static activity 2. Subsequent operations $o7 – o9$ correspond to gripping the part and moving it to another position (such as a turn table) and all together form activity 3. This is to demonstrate that also several movements, i.e. movement operations, can be formed into a single activity, which in case of activity 3 is dynamic. Such a sequence of operations is specified for each robot in the cell.

In the following text, which relates to [15], the problem of identifying individual robotic operations from the actual power needed by the robots to execute the movements is described.

The basic idea is to label sections of the data of power measurements, which correspond to particular operations of each robot in the cell. This may be done by observing the robots working in the cell and placing marks in the data. After that patterns are marked in the labelled data as model patterns, i.e. one pattern as a representative of one robotic operation, that will be searched for in the whole set of the power consumption data.

The procedure of identifying the operations consists of (a) *data filtering* to suppress the noise and errors stemming from misplaced samples, (b) *evaluating the similarity* of all segments of the power consumption data, and finally (c) taking the *measure of similarity as a threshold* to search for local maxima, which correspond to the location of the model patterns in the data.

Data filtering: To get rid of the error (noise, misplaced samples) a median filter is used. Its filtering window length has been chosen to filter out random signal errors with a big amplitude difference such as unsynchronised neighbouring samples, where two neighbours are swapped. The filter still conserves high peaks that are used as classification feature during the detection phase.

Similarity of segments: To assess the similarity of two same-length vectors of one-dimensional data signals the analysis in frequency or in time domain may be done. According to the measurements, which had been performed at the robots in a production cell, the frequency analysis has proven not to be suitable because the signals are very similar in their spectra. There may be two instances of the same operation but from different time points during the day, whereas one time point is at the end of the working shift and the other is at the beginning of the next one, whose correlation is 83 %. This value is too low contrary to the fact that both spectra correspond to the same operation. Moreover the frequency spectra of two different operations may be correlated up to 85 %. Based on this observation the frequency analysis is not suitable.

The get the similarity measure in the time domain Pearson's correlation coefficient is computed for its simplicity and suitability for this intended case. It is defined as follows.

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{cov(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \tag{6}$$

$$cov(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\mathbf{x}})(y_i - \mu_{\mathbf{y}}) \tag{7}$$

$$\sigma_{\mathbf{x}} = \sqrt{\frac{(\mathbf{x}-\mu_{\mathbf{x}})^T(\mathbf{x}-\mu_{\mathbf{x}})}{N}} \tag{8}$$

Coefficients $\sigma_{\mathbf{y}}$ and $\sigma_{\mathbf{y}}$ are the standard deviations of values of vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. The $cov(\mathbf{x},\mathbf{y})$ is the cross-covariance of the vectors $\mathbf{x}$ and $\mathbf{y}$. The $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{y}}$ are the mean values of $\mathbf{x},\mathbf{y}$, respectively, and N is the length of the model pattern vector. However, the mean values $\mu_{\mathbf{x}}, \mu_{\mathbf{y}}$ are not known exactly and thus their approximations by computing the mean values of $\bar{x},\bar{y}$ from the measured data are used.

Measure of similarity: After having established the measure of a good match of two segments for the power consumption data it must be considered, which segments to compare. A straightforward procedure is to compare every possible vector, i.e. a window in the data, which has the same length as the model pattern vector. Such a strategy guarantees a good precision of localization in time but is computationally demanding. Nevertheless, the procedure lies in picking a vector of length K of samples from the power consumption data where K is the same as the length of the model pattern vector. Then the Pearson's correlation coefficient is computed and stored in a vector of results $\mathbf{r}$, whose length can be expressed as

$$\dim(\mathbf{r}) = \dim(\mathbf{d}) - K + 1 \tag{9}$$

where $\mathbf{d}$ is vector of energy consumption data. The bigger K is the less correlations are needed to be computed but the more multiplications must be performed to compute each of them. Roughly $3K$ multiplications need to be done for each window on the power consumption data vector.

To reduce the length of the model pattern and thus to lower the computational cost to compute the correlation vector, distinctive features can be extracted from the power consumption data and the correlation can be computed on them. Local maxima have been chosen as these features because they can be detected during one iteration over the power consumption data and they provide good measure to match the patterns. Undesirable peaks are avoided by applying a threshold to choose only dominant peaks, which contain enough information for classification. In case more peaks occur in the defined neighbourhood only the biggest one is picked and in case of same-valued peaks the first one is prioritised. Fig. 9 shows the model pattern and peaks chosen according to the rules above. However, there are also peaks under the dashed threshold line that may get above the threshold during the peak extraction

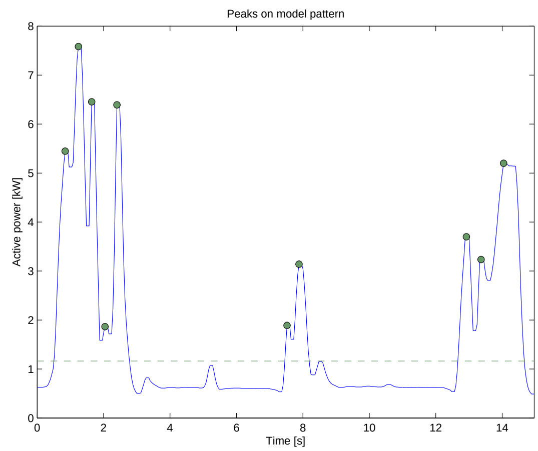

Fig. 9 Peaks extracted from the model pattern

process because of the drifts in offset during the working shifts. This fact would cause that some patterns physically generated by the same type of operation would have more dominant peaks than others. Such a situation is solved as follows.

The model pattern vector and the vector, in which the matching is done, must have the same length to be able to be compared. In fact, by the extraction of the feature vector a new down-sampled data vector with adaptive sampling time is created. Basically three distinctive situations may occur if different peaks, i.e. not all peaks are correctly recognised, are extracted. There may be (1) a perfect match, (2) there is one peak missing, and (3) there is one additional peak in the power consumption data. To avoid decreasing the correlation coefficient because of the described situation the corresponding timestamps must be paired and the peaks relating to unpaired timestamps must be dropped. Thus, because of the fact that the duration of the model pattern is known and the patterns that are searched for should not deviate much from it, a window being at least as long as the model pattern is chosen. Then each sample of model pattern vector is assigned a sample from the data vector based on the shortest Euclidean distance of timestamps. This procedure equalises the sizes of the compared vectors. Finally the correlation of the vectors of the aligned timestamps is evaluated and only such vectors that are correlated enough, are passed further for value correlation. Thanks to this procedure the correlation not only of the order of samples, but also of their position is considered. Moreover, situations when the robot operations are interrupted abruptly are also coped well with. The reason for

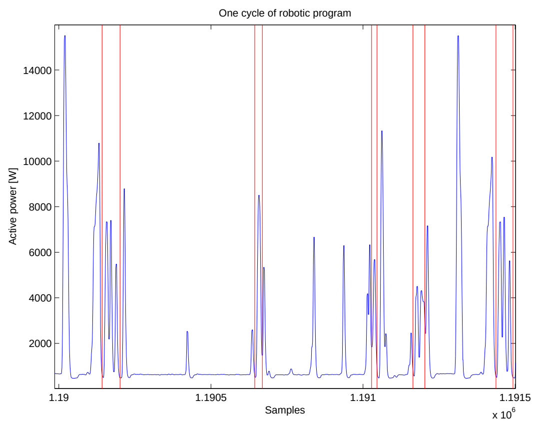

Fig. 10 Identified operations' boundaries

interruptions, which are in fact unplanned pauses in the production, may mean a failure in the equipment, interruption of the material flow, etc. Fig. 10 shows an example how the boundaries (i.e. the red vertical lines) of the operations of one robot are identified.

4 Optimisation

The following terms are used in the formulation of the optimisation problem as an Integer Linear Programming problem. The first one is the set of *static activities* $V_{\mathscr{S}}$ where static activity $i \in V_{\mathscr{S}}$ corresponds to a robot operation (i.e. subspace) such as e.g. welding or assembling. The set of *dynamic activities* $V_{\mathscr{D}}$ corresponds with all the movements where dynamic activity $i \in V_{\mathscr{D}}$ consists of all possible point-to-point movements between two subspaces, i.e. black arrows in Fig. 1 in section 2. Both the static and dynamic activities are multi-mode activities; it means that there are different states that an activity can attain. Mode $t \in T_i$ of dynamic activity i selects one of the point-to-point movements between related subspaces. In case of static activity i it is the selected position $p \in P_i$ (6 coordinates – x, y, z, rx, ry, rz) and power saving mode $m \in M_i$ of the robot. It is obvious that the set of activities $V = V_{\mathscr{S}} \cup V_{\mathscr{D}}$. Activities associated with robot $r \in R$ will be denoted by set V_r.

Static activities $V_{\mathscr{S}}$ can be further divided into three sets — $V_{\mathrm{IN}}, V_{\mathrm{OP}}, V_{\mathrm{OUT}}$. In set V_{IN} there are *input activities*, i.e. activities related to taking a workpiece. In a similar way set V_{OUT} consists of *output activities* related to passing a workpiece to another robot or machine. And finally, the rest of robot operations, i.e. $V_{\mathscr{S}} \setminus \{V_{\mathrm{IN}} \cup V_{\mathrm{OUT}}\}$, are activities such as welding, assembling, disassembling, cutting, etc.

In activity set V there are mandatory and optional activities. *Mandatory activities* $V_{\mathscr{M}}$ have to be carried out in all cases, whereas *optional activities* $V_{\mathscr{O}}$ are not necessary to be performed. The optional activities were introduced by considering alternatives where different paths can be taken, and as a result there may exist dynamic activities (i.e. $V_{\mathscr{O}} \subseteq V_{\mathscr{D}}$) conditionally executed. Robot operations $V_{\mathscr{S}}$, however, have to be performed every time, and therefore $V_{\mathscr{S}} \subseteq V_{\mathscr{M}}$. It is evident that $V = V_{\mathscr{M}} \cup V_{\mathscr{O}}$.

4.1 Integer Linear Programming Model

The objective is to minimise the overall energy consumption of activities. Note that not all activities in $V_{\mathscr{O}}$ have to be performed. In that case their W_i and d_i are set to zero due to the criterion.

Table 3 Model variables

W_i	required energy by activity i
s_i	start time of activity i
d_i	duration of activity i
x_i^p	true if robot position $p \in P_i$ for static activity i is selected, otherwise false
z_i^m	true if robot power saving mode $m \in M_i$ is selected in static activity i, otherwise false
y_i^t	true if movement $t \in T_i$ of dynamic activity i is selected, otherwise false
$h_{i,j,r}^*, w_{i,j}$	decide the order of activities

Equations (10) and (11) bind activity duration d_i with its power consumption W_i. Both the equations can be enabled or disabled depending on selected activity modes where $\overline{W}$ is an upper bound on energy. Equation (10) is proposed for static activities[2], whose power demand $a_{i,p}^m$ depends on robot configuration p and power saving mode m. In case of dynamic activities, i.e. equation (11), the energy function was approximated by a set of linear functions with coefficients $a_{i,k}^t$ and $b_{i,k}^t$ where $k \in K$ is the k-th segment of the energy function for movement t. The energy function has to be convex to ensure validity of the model.

$$\text{minimise} \sum_{\forall i \in V} W_i$$

$$\text{s.t.} \quad a_{i,p}^m d_i - \overline{W}\left(2 - z_i^m - x_i^p\right) \leq W_i \qquad \forall i \in V_{\mathscr{S}}, \forall p \in P_i, \forall m \in M_i \tag{10}$$

$$a_{i,k}^t d_i + b_{i,k}^t - \overline{W}\left(1 - y_i^t\right) \leq W_i \qquad \forall i \in V_{\mathscr{D}}, \forall t \in T_i, \forall k \in K \tag{11}$$

[2] Each activity can be performed by only one assigned robot.

Equations (12) and (13) state that each static activity $i \in V_{\mathscr{S}}$ has the position and robot power saving mode assigned. In a similar way, equation (14) ensures that one of the movements is selected for each mandatory activity.

$$\sum_{\forall p \in P_i} x_i^p = 1 \qquad \forall i \in V_{\mathscr{S}} \tag{12}$$

$$\sum_{\forall m \in M_i} z_i^m = 1 \qquad \forall i \in V_{\mathscr{S}} \tag{13}$$

$$\sum_{\forall t \in T_i} y_i^t = 1 \qquad \forall i \in V_{\mathscr{D}} \cap V_{\mathscr{M}} \tag{14}$$

$$\tag{15}$$

Flow preservation constraints (16) and (17) mean that if the robot moves to position p it also has to move away from p. In other words, if a movement to position p is selected then a movement from p is selected as well. Inward and outward movements are found by enumerating predecessors and successors respectively.

$$\sum_{\forall j \in \mathrm{pred}(i)} \sum_{\forall t \in T_j(p_{\mathrm{from}}, p)} y_j^t = x_i^p \qquad \forall i \in V_{\mathscr{S}}, \forall p \in P_i \tag{16}$$

$$\sum_{\forall j \in \mathrm{suc}(i)} \sum_{\forall t \in T_j(p, p_{\mathrm{to}})} y_j^t = x_i^p \qquad \forall i \in V_{\mathscr{S}}, \forall p \in P_i \tag{17}$$

Equations from (18) to (23) are related to activity ordering. Equation (18) sets time relations for mandatory activities ($V_{\mathscr{M}} \subseteq V_{\mathscr{D}}$). Alternatives are taken into account in equations (19), (20), and (21) where $w_{i,j}$ is a decision variable determining whether dynamic[3]activity $i \in V_{\mathscr{O}}$ with movements to static activity j will be selected or not. Binary variables $h^*_{i,j,r}$ decide which activity $i \in V_{\mathscr{D}}$ is closing (i.e. is the last one) the production cycle for each robot (see equations (22) and (23)) as it was found out that rotated schedules have to be taken into consideration due to time lags.

$$s_j - s_i = d_i - \mathrm{CT} h^*_{i,j,r} \tag{18}$$
$$\forall r \in R, \forall i \in V_r \cap V_{\mathscr{M}}, \forall j \in \mathrm{suc}(i)$$

$$\sum_{\forall t \in T_i} y_i^t = w_{i,\mathrm{suc}(i)} \qquad \forall i \in V_{\mathscr{O}} \cap V_{\mathscr{D}} \tag{19}$$

$$s_j - s_i + (1 - w_{i,j})\,\mathrm{CT} \geq d_i - \mathrm{CT} h^*_{i,j,r} \tag{20}$$
$$\forall r \in R, \forall i \in V_{\mathscr{O}} \cap V_r \cap V_{\mathscr{D}}, \forall j \in \mathrm{suc}(i)$$

$$s_j - s_i - (1 - w_{i,j})\,\mathrm{CT} \leq d_i - \mathrm{CT} h^*_{i,j,r} \tag{21}$$
$$\forall r \in R, \forall i \in V_{\mathscr{O}} \cap V_r \cap V_{\mathscr{D}}, \forall j \in \mathrm{suc}(i)$$

$$\sum_{\forall i,j} h^*_{i,j,r} = 1 \qquad \forall r \in R \tag{22}$$

$$h^*_{i,j,r} = 0 \qquad \forall r \in R, \forall i \in V, \forall j \notin V_{\mathrm{IN}} \tag{23}$$

The duration of the activity is bound in equations (24) and (25). Minimal time of staying in a static activity $\underline{d}^m$ is determined by the selected robot power saving mode. Maximal duration $\overline{d}_i$ can be limited by a robot operation, e.g. immersion of a workpiece in paint to get a protective coating. The duration of dynamic activity i is influenced by selected trajectory t lasting from $\underline{d}_i^t$ to $\overline{d_i^t}$.

$$\underline{d}^m z_i^m \leq d_i \leq \overline{d}_i \qquad \forall i \in V_{\mathscr{S}}, \forall m \in M_i \tag{24}$$

$$\underline{d}_i^t y_i^t \leq d_i \leq \overline{d}_i^t + \mathrm{CT}\left(1 - y_i^t\right) \qquad \forall i \in V_{\mathscr{D}}, \forall t \in T_i \tag{25}$$

Finally, the last two equations (26) and (27) ensure the correct synchronisation between robots. Equation (26) guarantees time constraints, e.g. the workpiece is taken away after it has been put on the bench, whereas equation (26) warrants proper handovers in terms of robot configurations. For each position p of activity $i \in V_{\mathrm{OUT}}$ one of compatible positions $p' \in \mathrm{CP}(i,p)$ of activity $j \in V_{\mathrm{IN}}$ can be selected. Both the equations can be perceived as the global ones because they link robots to each other.

$$s_j - s_i \geq l_{i,j} - \mathrm{CT} h_{i,j} \qquad \forall (l_{i,j}, h_{i,j}) \in \mathscr{L} \tag{26}$$

$$x_i^p \leq \sum_{\forall p' \in \mathrm{CP}(i,p)} x_j^{p'} \qquad \forall i,j \subseteq V_{\mathrm{OUT}} \times V_{\mathrm{IN}} \tag{27}$$

$$W_i, s_i, d_i \in \mathbb{R}_0^+ \quad x_i^p, z_i^m, y_i^t, h_{i,j,r}^*, w_{i,j} \in \{0,1\}$$

4.2 Lagrangian Relaxation

As the first attempt to get a good lower bound it was decided to use Lagrangian relaxation (for details see e.g. [1]) that is based on relaxing difficult constraints and moving them to the criterion where they are multiplied by Lagrange multipliers. The global constraints seem to be the best candidates for the relaxation, i.e. (26) and (27), as without them the problem decomposes to subproblems where each subproblem corresponds to one robot. Applying the relaxation the following lower bound is obtained.

---

[3] The dynamic activity has exactly one successor and one predecessor.

$$\underset{\substack{\lambda_e \geq 0 \\ \alpha \geq 0}}{\text{maximise}} \quad \underset{\substack{W_i, s_i, d_i \in \mathbb{R}_0^+ \\ x_i^p, z_i^m, y_i^t, h_{i,j,r}^*, w_{i,j} \in \mathbb{B}}}{\text{minimise}} \quad \sum_{\forall i \in V} W_i + \sum_{\forall e \in E} \lambda_e \left(l_{i,j} - \mathrm{CT} h_{i,j} + s_i - s_j \right)$$
$$+ \sum_{\forall i,p} \alpha_{i,p} \left(x_i^p - \sum_{\forall p' \in \mathrm{CP}(i,p)} x_j^{p'} \right)$$
$$\text{subject to (10) to (25)}$$

Primal and dual Lagrangian problems are iteratively solved to get the lower bound. The primal problem minimises the value of the modified criterion with the fixed multipliers for the original problem without relaxed constraints. The aim of the dual problem is to set the multipliers to such values that optimal criterion value of the primal problem is maximised. The dual problem is usually solved by using the subgradient method, however, other methods such as the Bundle method are also possible (see [2]). The maximal criterion value of the primal problem is a valid lower bound for the original problem.

5 Experimental Results

To verify the validity of the proposed optimisation model 17 problem instances were generated, each of them corresponds to a robotic cell with 5 co-operating robots where each robot has up to 3 power saving modes (motors, brakes, bus power off). From 1 to 4 robot configurations are considered for each static activity and in average there are approximately 150 activities per instance. The production cycle time is got by multiplying a lower bound by a factor from 1.05 to 1.40.

The energy optimisation problem was formulated as an Integer Linear Programming problem and solved by using IBM Ilog Cplex 12.6. Gentoo Linux server equipped with 2 x Intel Xeon E5-2620 v2 @ 2.10 GHz processors and 64 GB memory was used for benchmarks.

As the first experiment the influence of Cplex time limit on quality of solutions was investigated as shown in Table 4. It was found out that if the solver is given 2 hours instead of 100 seconds the quality of solutions improves about 3.3 % in average. An average gap, which is a relative distance between the best found upper bound (the best solution) and the lower bound, was 27.5 % for the two-hour limit. The size of the model was roughly about 10000 constraints and 1000 variables.

Table 4 Statistics of the energy consumption for feasible instances.

	time limit 100 s	time limit 7200 s
min UB	28038.8 J	27656.2 J
avg UB	33796.7 J	32681.6 J
max UB	43043.0 J	40849.9 J

The Lagrangian relaxation was tested on 4 selected instances as it had been shown that other feasible problems are too computationally expensive. Nonetheless, it took a few hours to find a high quality lower bounds by using the subgradient method and ILP solver since solving even one robot to optimality usually takes more than a minute even though all 12 CPU cores are utilised. The subgradient algorithm stops if more than 200 consecutive deteriorations were reached. Results reveal that the Lagrangian relaxation provides much tighter lower bounds than Cplex solver since average gap was 3.5 % in comparison with 16.7 % gap proved by Cplex.

An industrial use case has also been used as an instance for the optimisation problem. This use case represents a robotic cell with six robots and other pieces of equipment such as turn tables, conveyors, gluing machine and welding guns. A

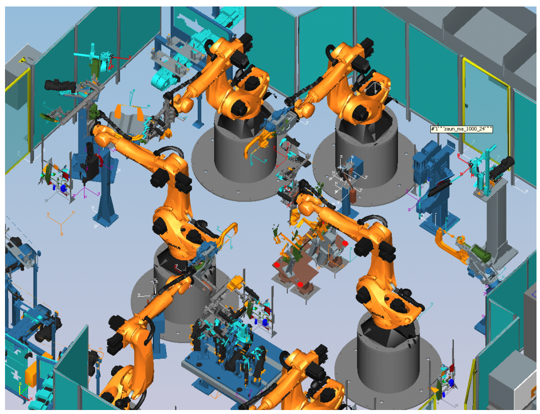

Fig. 11 Structure of the welding cell

general structure of the welding cell is depicted in Fig. 11. The behaviour of the cell can be described shortly as follows. The basic part together with two smaller parts are put onto the turn table by the operator. This turn table cannot be seen in Fig. 11 as it is hidden below its bottom. The turn table turns and the first robot (which is visible only partially in the figure) performs the welding to mount the small parts to the basic one. After this welding is finished the robot, which besides of the servo gun possesses also a gripper, takes the part and moves it to the following static table. Here, the next robot places another parts onto the basic one and performs the welding afterwards. Then, the part is taken by the next robot from the table and is moved to the next table. In the meanwhile, the next robot prepares another part and has a glue put on it by the gluing machine. This part is placed onto the basic part, which has been moved already, and the welding is performed. Subsequently, the part is taken by the next robot and brought to the static welding gun that performs the next welding operations. Finally, the last robot takes the part, performs additional welding operations at the last static welding gun and puts the resulting part onto the outgoing conveyor, which conveys the part out of the cell. The timing of the operations was retrieved from the robot programs and each trajectory's energy function f_E

has been interpolated from the points, obtained from simulations in Tecnomatix Process Simulate. Welding, glueing, and assembling operations have not been changed by the optimisation to ensure repeatability of the production process. Only the robot speeds and power saving modes (at home position) were addressed in the optimization since the minimal intervention is desirable for existing robotic cells. The results show that the original energy consumption 430 kJ (maximal speeds, without power saving modes) can be decreased to 320 kJ (reduced speeds, power saving modes), which makes up roughly 25 % of energy saving.

6 Conclusion and future work

In this work a general mathematical model was proposed to optimise the energy consumption of robotic lines that allows taking into account robot power saving modes and different locations of operations. The optimisation problem was formulated as an Integer Linear Programming problem and solved using Cplex solver. The achieved results confirm the correctness of the model and show that a significant reduction of the energy consumption can be achieved. In addition to the mathematical model the Lagrangian relaxation was used to devise a very tight lower bound.

Each of the presented methods to calculate the energy function has its pros and cons and their usage depends on specific conditions for a given robotic cell to be optimised. The measurement of the energy consumption of the robots moving on different trajectories with various speeds is usually not possible at an existing robotic cell, which participates in the production. Mathematical modelling depends on having the right 3D models not only of the robots but also of the parts that the robots carry. Realistic Robotic Simulation depends on the precision of the simulation model whereas an exact or even an approximate value is usually not known. Thus in a typical situation a combination of more approaches must be used to obtain the energy function of all the robots.

The outcomes of the industrial use-case optimisation show a significant potential to reduce energy consumption of existing robotic cells and even more can be expected for planned robotic cells as the full potential of the optimisation algorithm can be exploited.

The future work will thus concentrate mainly on making the mathematical model of the robots more precise and on decreasing the uncertainties in the models by providing further information e.g. from measurements. The robot models will probably always contain uncertainties because of the lack of publicly available information. Therefore measurements must be performed to supplement the missing information and to complete the mathematical models. Last but not least more stress is going to be put on evaluating the optimisation results with industrial use cases to prove their viability.

Glossary

CT	production Cycle Time
$\mathscr{L}$	time lags
$\mathrm{L}(e_{i,j})$	see $l_{i,j}$
$\mathrm{H}(e_{i,j})$	see $h_{i,j}$
$E(d_i)$	energy function linking the consumption with time of movement
d_i	duration of activity i
s_i	start time of activity i
W_i	energy consumed by activity i
$\mathrm{pred}(i)$	predecessors of activity i
$\mathrm{suc}(i)$	successors of activity i
$V_{\mathscr{S}}$	set of static activities, i.e. robot operations
$V_{\mathscr{D}}$	set of dynamic activities, i.e. robot movements
$V_{\mathscr{M}}$	set of activities that have to be executed
$V_{\mathscr{O}}$	set of activities ($\subseteq V_{\mathscr{D}}$) that can optionally be executed
P_i	set of possible robot configurations for activity $i \in V_{\mathscr{S}}$
M_i	set of the robot power saving modes that can be used in activity $i \in V_{\mathscr{S}}$
T_i	set of possible movements of activity $i \in V_{\mathscr{D}}$
x_i^p	binary variable set to true iff the robot configuration p was selected for activity $i \in V_{\mathscr{S}}$
z_i^m	binary variable set to true iff the robot power saving mode m was selected for activity $i \in V_{\mathscr{S}}$
y_i^t	binary variable set to true iff movement t was selected for activity $i \in V_{\mathscr{D}}$
$l_{i,j}$	the length of the edge in cyclic scheduling
$h_{i,j}$	the height of the edge in cyclic scheduling

Acknowledgements This work has been conducted in cooperation with Skoda Auto within contract 830-8301343/13135 and with support of the Grant Agency of the Czech Technical University in Prague, grant No. SGS13/209/OHK3/3T/13.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)
2. Bard, J.F., Purnomo, H.W.: Cyclic preference scheduling of nurses using a Lagrangian-based heuristic. Journal of Scheduling **10**(1), 5–23 (2007)
3. Bjorkenstam, S., Gleeson, D., Bohlin, R., Carlson, J., Lennartson, B.: Energy efficient and collision free motion of industrial robots using optimal control. In: IEEE Int. Conf. on Automation Science and Engineering, pp. 510–515 (2013)

4. Bryan, C., Grenwalt, M., Stienecker, A.: Energy consumption reduction in industrial robots. In: Proc. of ASEE North Conf., pp. 1–4 (2010)
5. Chemnitz, M., Schreck, G., Kruger, J.: Analyzing energy consumption of industrial robots. In: Proc. of IEEE Conf. on Emerging Technologies & Factory Automation, pp. 1–4 (2011)
6. Gavrovska, A.M., Paskaš, M.P., Dujković, D., Reljin, I.S.: Region-based phonocardiogram event segmentation in spectrogram image. 10th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2010 - Proceedings pp. 69–72 (2010). DOI 10.1109/NEUREL.2010.5644108
7. Hanen, C., Munier, A.: A study of the cyclic scheduling problem on parallel processors. Discrete Applied Mathematics **57**(23), 167 – 192 (1995). Combinatorial optimization 1992
8. KUKA: KUKA Industrial Robots (2014). Available at http://www.kuka-robotics.com/en/, accessed 3.6.2014
9. Lampariello, R., Nguyen-Tuong, D., Castellini, C., Hirzinger, G., Peters, J.: Trajectory planning for optimal robot catching in real-time. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 3719–3726 (2011)
10. Le, C.V., Pang, C.K., Gan, O.P., Chee, X.M., Zhang, D.H., Luo, M., Chan, H.L., Lewis, F.L.: Classification of energy consumption patterns for energy audit and machine scheduling in industrial manufacturing systems. Transactions of the Institute of Measurement and Control **35**(5), 583–592 (2012). DOI 10.1177/0142331212460883
11. Mashaei, M., Lennartson, B.: Energy Reduction in a Pallet-Constrained Flow Shop Through On–Off Control of Idle Machines. IEEE Transactions on Automation Science and Engineering **10**(1), 45–56 (2013)
12. Michna, V., Wagner, P., Cernohorsky, J.: Constrained Optimization of Robot Trajectory and Obstacle Avoidance. In: IEEE Conference on: Emerging Technologies and Factory Automation (ETFA), pp. 1–4 (2010)
13. Othman, A., Belda, K., Burget, P.: Physical modelling of energy consumption of industrial articulated robots. In: 15^th ICCAS International Conference on Control, Automation and Systems (2015)
14. Riazi, S., Bengtsson, K., Wigström, O., Vidarsson, E., Lennartson, B.: Energy optimization of multi-robot systems. In: 20^th IEEE CASE Conference on Automation Science and Engineering, pp. 1345–1350 (2015)
15. Ron, M., Burget, P., Fiala, O.: Identification of operations at robotic welding lines. In: 20^th IEEE CASE Conference on Automation Science and Engineering, pp. 470–476 (2015)
16. Saramago, S., Jr, V.S.: Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. Mechanism and Machine Theory **33**(7), 883 – 894 (1998)
17. Saravanan, R., Ramabalan, S., Balamurugan, C.: Evolutionary optimal trajectory planning for industrial robot with payload constraints. The Int. J. of Advanced Manufacturing Technology **38**(11-12), 1213–1226 (2008)
18. Sharma, G.: Optimization of energy in robotic arm using genetic algorithm. Int. J. of Comp. Sci. and Technology **2**(2), 315–317 (2011)
19. Siciliano, B., Sciavicco et al., L.: Robotics - Modelling, Planning and Control. Springer (2009)
20. Simon, L., Hungerbuehler, K.: Real time takagi-sugeno fuzzy model based pattern recognition in the batch chemical industry. In: Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on, pp. 779–782 (2008). DOI 10.1109/FUZZY.2008.4630459
21. Smetanová, A.: Optimization of energy by robot movement. Modern Machinery Science Journal **3**(1), 172–176 (2010)
22. Vergnano, A., Thorstensson, C., Lennartson, B., Falkman, P., Pellicciari, M., Leali, F., Biller, S.: Modeling and Optimization of Energy Consumption in Cooperative Multi-Robot Systems. IEEE Transactions on Automation Science and Engineering **9**(2), 423–428 (2012)
23. Wigstrom, O., Lennartson, B.: Integrated OR/CP optimization for Discrete Event Systems with nonlinear cost. In: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 7627–7633 (2013)

24. Wigstrom, O., Lennartson, B., Vergnano, A., Breitholtz, C.: High-Level Scheduling of Energy Optimal Trajectories. Automation Science and Engineering, IEEE Transactions on **10**(1), 57–64 (2013)
25. Wigstrom, O., Sundstrom, N., Lennartson, B.: Optimization of Hybrid Systems with Known Paths. In: 4 th IFAC Conference on Analysis and Design of Hybrid Systems, 2012, pp. 39–45 (2012). DOI 10.3182/20120606-3-NL-3011.00007