

Development of Energy Consumption Model for Virtual Commissioning Software

Oskars Bormanis (M.Sc.ing., Riga Technical University, Institute of Industrial Electronics and Electrical Engineering)

Abstract – Two different industrial robot electrical energy consumption modeling applications with various modeling approach are discussed. Application programming interface between MATLAB and external C++ or VB.NET application is introduced for energy consumption data at virtual commissioning software. Validation of the developed model is performed. The main identified problem is speed of the application programming interface (API). Future development and application suggestions are discussed.

Keywords – Virtual manufacturing; multi-robot systems; prediction models;

I. INTRODUCTION

By use of virtual commissioning software (VCS) it is possible to simulate automation solutions for entire production lines of factories [1]. VCS provides an opportunity for extensive testing, validation of automation configurations, operator training and factory level communication interfaces from the very early production development stages [2].

Introducing VCS, larger projects are realized in reduced time, providing real-world installations in a virtual environment. Configuration, production time, risk assessment and plant start-up routine – all are evaluated extensively before construction of the production line.

During planning of production, plant exists only in descriptions of manufacturing process and other engineering documents, therefore any data inconsistencies, design flaws or bugs are revealed late, compared to testing of physical solutions in VCS, with less time-stress. Corrections require less effort, resources and reduced implementation costs.

Environment of VCS supports testing of multiple mechatronic elements, including industrial robots, conveyers, lifts, whose operation is controlled from PLC, connected to a computer, instead of an actual actuator as described in [3]. Movement and operation of components is sequenced in virtual environment to meet requirements of production cycle.

In case of complex mechatronic units, such as industrial robots, robot movement program is developed and tested in VCS, including simulations of connected PLC inputs, outputs. Tool of the robot is included in simulation.

When the control algorithm is verified [4], it is uploaded on the main controller at production site. Calibration is still required for precision and coordinate specific processes. Industrial robots require calibration mainly for tool center points, movement limits and precision processing point coordinates.

By use of VCS, it is possible to control production in an advanced manner, as exact time requirements for various

production events are revealed. Time management in production planning increases the output product quality and provides more balanced production rates [5].

Operation rates of various mechatronic units are arranged according to maximal speed, tested in VCS with PLC synchronization. Energy is saved by decrease of production rates before time-consuming stations, avoiding creation of production queries. Production speed is manipulated according to analysis at the VCS.

VCS is a visualization and commissioning software. It has no advanced mathematical (Wolfram Alpha) or physics engine (Bullet Physics) capabilities. External sources must be used, when high level calculations are required, such as energy consumption by any included mechatronic component.

II. EXISTING OFFLINE ENERGY CONSUMPTION MODEL

Introduction of external sources for energy consumption approximations provide multiple innovative opportunities, such as energy efficient production planning. Methods for energy consumption calculations of an industrial robot have been presented in [6].

By use of industrial robot energy consumption model, movement path, processing speed and acceleration profiles of programmed trajectories are altered to achieve the least possible energy consumption with the required production speed and quality parameters.

To improve precision of the model output data, tool of the industrial robot is included as an additional mechanical load with mass center and inertia values. Tool parameter modification flexibility matters, since the load weight of handling robots may vary throughout the manufacturing process.

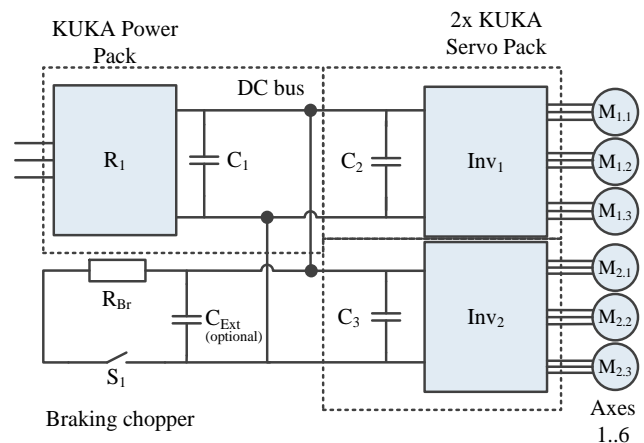


Fig. 1. Layout of the main components included in the model of an industrial robot with an addition of an external capacitor and a brake chopper resistance.

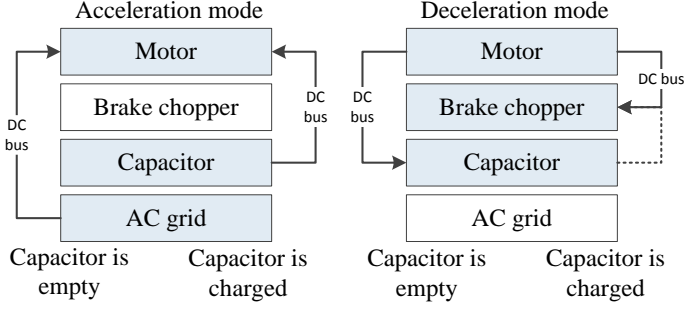


Fig. 2. Model of energy flow in the model during acceleration and deceleration phase, with both – empty and charged capacitors of the DC bus.

Setup is connected to industrial AC voltage source. Model includes a rectifier, a DC bus link with a brake chopper resistance, multiple voltage filtering capacitors and an inverter for each of the six permanent magnet synchronous motors (PMSM), as shown in Fig. 1.

Additional capacitor is included in the model by entering exact value of external capacitance. External capacitor is used to store recuperated energy of the robot, which is consumed during the acceleration of the same robot, as shown in Fig. 2. Since external capacitance is an addition to standard industrial robot unit, it is included in the model as an optional feature. When modeled without additional capacitor, recuperated energy is dissipated in the brake chopper and total consumption of the industrial robot system is increased.

The previously developed software, which is a base of the version for VCS integration, supports multiple industrial robot types and a realistic trajectory generation module.

In the offline model of the industrial robot system, trajectory of the robot is imported and processed as an array, executing required mathematical functions in a single calculation loop. DC bus is modeled and electrical energy consumption value is reported as an output array with electrical parameters of the system included. These parameters are power dissipation in the brake chopper resistance, DC bus voltage, power losses in the diodes of the rectifier.

After the modeling is complete, output data are analyzed and required modifications are applied to code of industrial robot movement in KUKA Robot Language (KRL). Model is executed again, and if the output data meets the requirements, previously modified code is ready to be verified for install on industrial robot, to reduce electrical energy consumption.

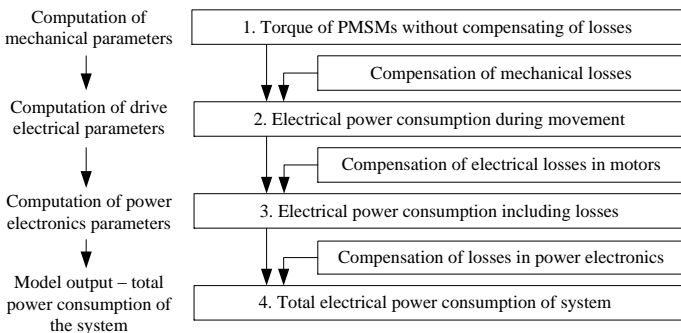


Fig. 3. Functional structure of the electrical energy consumption model, including computation of both – mechanical and electrical robot system data.

Functional schematic of the power consumption modeling offline version software is shown in Fig. 3. The software is created according to conversion methods presented in [6].

III. APPLIED MODIFICATIONS

Development of the online version is a scheduled upgrade to the previously created offline model. Goal of this configuration is to develop an interface between two independent software environments. The developed online model is a tool for combining functionality of VCS (visualization, commissioning) and Mathworks MATLAB (mathematical engine, energy consumption calculations).

Embedding the engine of mathematical calculations in virtual commissioning environment results in extended research possibilities. These possibilities include trajectory optimization directly by the first programming engineer, modeling of peak electrical loads and implementation of efficient sequencing, to avoid overlay of multiple industrial robot acceleration and therefore power consumption peaks, which are demanding for the power supply system, similar to [7].

The offline model is intended for analysis and optimization of already existing sample research trajectories, to develop algorithms for testing and development. Offline model has a graphical user interface (GUI), it has an integrated trajectory generation module, a PDF generation module, an user friendly interface.

On the other hand, the online model is intended for embedded use in parent software, receiving operation instructions through application programming interface (API). The online model is a MATLAB compiled library, therefore GUI is created at the VCS and included features are executed throughout the limited possibilities of the API. Additional features, such as trajectory generation are removed from the online model, since trajectory is generated by VCS and generation of PDF reports is not required for this version.

Interfacing MATLAB with VCS requires development of different approach to data processing, compared to previous version, since, previously, trajectory was known for entire movement and analyzed with complete movement path. In second model, trajectory data points are added each time step, separating single movement points, which are generated by VCS module.

In VCS environment, movement path is generated by embedded realistic movement path generator. VCS trajectory data points are generated with time step in milliseconds range, therefore data exchange between the original software and external calculation application is required with time period in milliseconds as well.

Time requirements for communication and calculations could exceed the originally required value of milliseconds, therefore further modifications are possible, reducing the energy consumption value update frequency, demanding less time for API execution.

Time consumption of the mathematical model is a matter of optimization of the required computation parameters.

TABLE I

FORMATTING SAMPLE OF 6-AXIS INDUSTRIAL ROBOT TRAJECTORY DATA

Time, ms	A1 pos., °	A2 pos., °	A3 pos., °	A4 pos., °	A5 pos., °	A6 pos., °
0	0	-90	90	0	0	0
0,012x	0,001	-89,99	90	0	0,001	0
0,024	0,008	-89,93	90,01	0	0,010	0

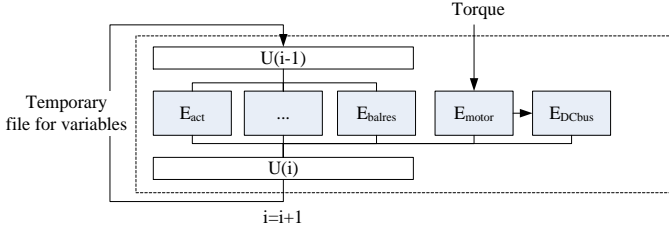


Fig. 4. The most recent DC Bus parameters and previous location values for operation with VCS are stored in a temporary variable file and loaded during each inquiry from VCS, creating a feedback loop.

Less processing data and functions reduce time consumption. If required, 22 electrical parameter data output is available, but requires more time, compared to output of single variable, for example, total power consumption.

For each time value of the trajectory, series of mathematical functions are applied, by input of variables at the external calculation software. According to construction of an industrial robot, trajectory position include data values of six axis and time step as shown in Table 1.

Model output data is computed and output for each received trajectory data value set. Typical time period values for single trajectory point range from 10 to 12 ms. Duration of total movement depends on type of industrial robot. If the robot is assigned and programmed for handling – program is short and fast, however, if the task of industrial robot involve applying paint or glue – program is long and executed with limited speed, to process the product with caution and quality.

Functional modifications of the offline model include introduction of temporary variable, to store previously received input data values together with previously obtained parameter values, such as DC bus voltage or energy stored in capacitors.

Upon the receipt of input data at the compiled library, most recent available trajectory values are loaded from the previously generated temporary variable storage file. This file is created during the first inquiry of the compiled library. There are other methods available to operate with the previously generated variables essential for the system, such as the return of all data to VCS as an output. All required data would be received through a single inquiry. However, during the testing phase, creation of temporary file proved the less programming demanding solution.

To obtain final data, multiple computation loops are required. With a sample time step of 12ms, five seconds long handling movement require 418 computation loops. During each loop, variables of the previous cycle are loaded from the previously mentioned temporary variable file. Final values of the active loop are calculated at the end of processing each cycle as shown in Fig. 4.

IV. THE DEVELOPED ONLINE MODEL

The developed online model is intended to model KUKA KR C2 industrial robot system with a single robot in VCS environment. There are no major differences in calculation of movement dynamics between the previous offline and the currently developed online model versions. The main difference between these two models is a different data processing algorithm. At the offline model data is processed in a single loop, whereas in the online model – data processing is done in an increasing number of loops, depending on trajectory length.

At the offline model, initialization values are set at the beginning of the process. For comparison, initialization value must be set to zero throughout whole execution time of the online model, with an exception of the first inquiry, at which the file of temporary variables is created and initialization voltage values are set.

Due to modifications in data processing algorithm, total number of computation function inquiries is increased in the online model. Both models were analyzed with built-in MATLAB profiler for detailed data of function inquiries and total time consumption. In the offline model computing functions were requested once, compared to multiple hundred function inquiries in the online model. The offline model managed to complete analysis of sample movement in 2.77 seconds, compared to 22 seconds of the online model.

In order to operate the model in VCS environment, major requirement is that the computation time is smaller than the period between two inquiries of the VCS.

Performance of the online model is in a balance between operation speed and frequency of data updates. By pre-declaration of the model parameters, the best approach for integration of VCS is obtained, modifying the time step, update frequency from 1ms to 1s or more, according to the software test results.

Sample scripts were created in MATLAB, C++ and VB.NET programming environments. Each script emulated a VCS environment by loading required variables. Trajectory was loaded from an external file and transformed into an array. Additional parameters include robot type, robot tool data, path to the robot data and initialization variable.

When inquiry of the VCS is received at the compiled library, the most recent trajectory and robot data is loaded to execute the energy consumption model. Data of more than 20 various parameters are returned as an output and processed by VCS environment, as shown in Fig. 5.

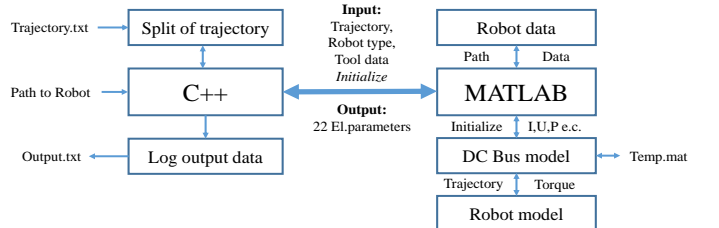


Fig. 5. Structure of the offline energy consumption model in sample C++ script and the developed MATLAB model.

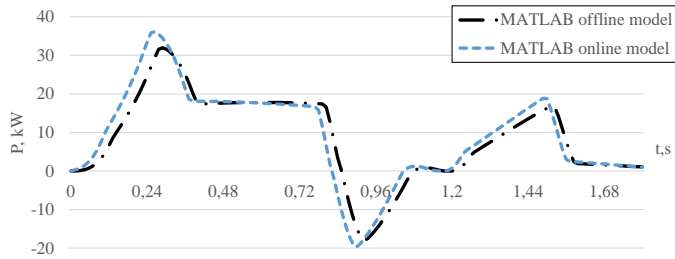


Fig. 6. MATLAB offline energy consumption model output data of first axis (A1) of an industrial robot, compared to the online model energy consumption model output data of the same robot following the same trajectory.

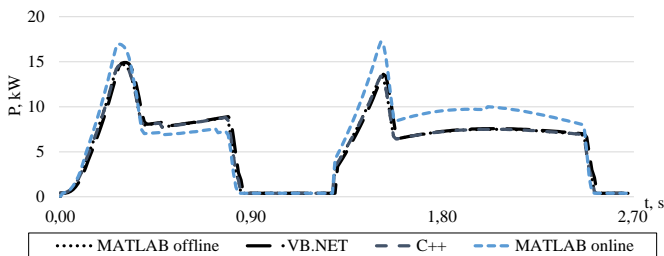


Fig. 7. Total power consumption data of an industrial robot system during sample movement, analyzed in MATLAB, VB.NET and C++ environments.

V. VALIDATION OF THE ONLINE MODEL

Multiple tests were completed to validate data output of the online model. One of such tests included analysis of KUKA KR570PA electrical energy consumption during a sample movement. Program performed a standard analysis by the previously developed and validated offline model and the newly developed online model. Visual comparison of data output for both models is shown in Fig. 6. Average difference between the offline and the online model output in this scenario is 4,951%.

Difference between the two developed models is introduced by disparity of variable processing during the dynamics and energy computations. At the online model, temporary data array of 5 most recent movement values was available for speed, acceleration and inertia calculation, stored in the temporary external file of variables. At the offline model, all movement data values are processed as single array. Since speed and acceleration is calculated as a derivative from location, more location data values provide more accurate data. Research is required how increase of available data points in an array affects precision and total modeling rate, including communication.

Before deployment of the compiled online model final version, an additional test revealed that transfer between MATLAB environment and that of other programming languages is possible without additional loss of precision. In average, an error of 0.009W was introduced by compiling the offline model code to different programming languages, as shown in Fig. 7.

Programming languages were chosen following the request from VCS developer companies. Multiple sample scripts and compilation tests were required, since developers of VCS have based their software on various programming environments.

VI. CONCLUSIONS

Prototype of a new modification of the previously designed standalone industrial robot electrical energy consumption modeling application has been created and validated. Features include multiple industrial robot types, dynamic industrial robot tool parameters.

By future development of this application, introduction of MATLAB computations in VCS will allow high precision estimation of the total electrical energy consumption value. It will include modeling of consumption by addition of more electrical drives, in cooperation with equipment manufacturers, to develop precise models with actual values of mechanical and electrical equipment parameters. Integration of additional electrical energy consumers is also possible with known values of power consumption in time and known operation parameters.

With known consumption values and total electrical load estimation, design of electrical energy supply infrastructure can be improved.

Integration of renewable energy sources would assist during the design phase, allowing to evaluate the effect and planned output power, with respect to the electrical grid of the factory.

ACKNOWLEDGEMENTS

This research is done with support of the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No 609391.

REFERENCES

- [1] M. Oppelt and L. Urbas, "Integrated Virtual Commissioning an essential Activity in the Automation Engineering Process From virtual commissioning to simulation supported engineering," in *Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE*, 2014, pp. 2564 – 2570.
- [2] H. Wang, N. Li, X. Fang, and J. Hofschulte, "An architecture for robot application simulation systems and its industry practice," *IEEE Conf. Robot. Autom. Mechatronics, RAM - Proc.*, pp. 143–147, 2011.
- [3] R. Drath, P. Weber, and N. Mauser, "An evolutionary approach for the industrial introduction of virtual commissioning," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, pp. 5–8, 2008.
- [4] H. Carlsson, B. Svensson, F. Danielsson, and B. Lennartson, "Methods for reliable simulation-based PLC code verification," *IEEE Trans. Ind. Informatics*, vol. 8, no. 2, pp. 267–278, 2012.
- [5] S. Seidel, U. Donath, and J. Haufe, "Towards an integrated simulation and virtual commissioning environment for controls of material handling systems," *Proc. - Winter Simul. Conf.*, 2012.
- [6] D. Meike, M. Pellicciari, and G. Berselli, "Energy Efficient Use of Multirobot Production Lines in the Automotive Industry: Detailed System Modeling and Optimization," *Autom. Sci. Eng. IEEE Trans.*, vol. 11, no. 3, pp. 798 – 809, 2013.
- [7] A. Ranjan, P. Khargonekar, and S. Sahni, "Offline Preemptive Scheduling of Power Demands to Minimize Peak Power in Smart Grids *," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, 2014, pp. 1–6.

Oskars Bormanis. He received his B.Sc and M.Sc degrees in electrical engineering from Riga Technical University, Riga, Latvia in years 2014 and 2015. Since 2011 he is employed by Institute of industrial electronics and electrical engineering of RTU. His current position is technical assistant. Main research interests include industrial robots, industrial DC grids. He is a member of IEEE Industry Applications Society.
E-mail: Oskars.Bormanis_1@rtu.lv.