

DEM4RTS: SOFTWARE DEVELOPMENT METHODOLOGY FOR SPECIAL CASE OF REAL-TIME CLOSED-LOOP CONTROL SYSTEMS

Kristina Blašković & Sanja Čandrlić



This Publication has to be referred as: Blaskovic, K[ristina] & Candrlic, S[anja] (2018). DEM4RTS: Software Development Methodology for Special Case of Real-Time Closed-Loop Control Systems, Proceedings of the 29th DAAAM International Symposium, pp.0993-1000, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-20-4, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/29th.daaam.proceedings.142

Abstract

The paper presents a novel case specific software development methodology oriented toward managing research and development of control software algorithms for modern marine electric propulsion systems. Focus of our research are highly complex technical closed-loop control systems where software performs process control algorithm in real-time. Specific software requirements (like efficient code and close compatibility with hardware on which it is implemented) are satisfied by implementing control mechanisms into development process. In addition, the methodology provides necessary documentation and traceability.

Keywords: software development methodology; real-time closed-loop control systems; permanent magnet electric motor drives

1. Introduction

Generally, we can define a control system to be a set of processes with input, output and internal state variables [26]. If inner processes of the system are unknown but the relationship between inputs and output variables is defined, then by changing input, the output variables can be controlled. This kind of system is called Open-loop control system [28]. In a case where realization of output variables must be more precisely controlled, matching of output values with the pre-set conditions via feedback information, must be checked. Application of these systems requires simultaneous input correction when disagreement with specified conditions happens. In digital control systems, the software executing this algorithm must react instantly when variables change. In that case, we are dealing with real-time closed-loop control systems. Significant importance of these technical systems stems from their wide application in industry, transport, household and different application of special purpose [21]. Modern marine electric propulsion system is a special case of real-time closed-loop control system. This system is composed of:

1. an electric motor connected to the propeller power train
2. electronic control unit for motor torque and speed regulation
3. power sources supplying motor with electric energy

Electric motor is a device that converts electrical energy into mechanical work, while the device that converts mechanical into electrical energy is called the electric generator [22]. Electric motor can run in two directions, clockwise (CW) and counter-clockwise (CCW), delivering torque to propeller. In the moment when the propeller changes rotation direction, the energy stored in rotating mechanical parts should be regenerated back to the power source and in this case, the motor is working as a generator.

Regarding electric motor behaviour, we basically distinguish two groups of electric machines [20]:

1. DC motor (Direct current motor) – electric motor which uses direct current energy source and transforms it into mechanical work
2. AC motor (Alternating current motor) – electric motor where the polyphase alternating current energy source is connected to stationary windings generating rotating magnetic field resulting in rotor CW or CCW rotation. We can divide them into two groups:
 - a) synchronous – rotor rotating speed is in correspondence with the frequency of the power supply
 - b) asynchronous – or so-called induction motor where rotor speed slightly defers from the frequency of the power supply

Today's modern marine electric propulsion systems are based on high efficiency AC motor drives and future premium efficiency systems are based on specific synchronous motors in permanent magnet technology. This motor control system implements real-time full digital close loop control algorithms commonly known as "vector control drives" [30].

Control Algorithm constantly analyses output values on the propeller and makes decisions about changes of electric motor input values, concurrently with the change of internal variables of process state or new input setting values, forming real-time closed-loop control system. Application of those systems permits possibility of motor revolutions per minute (rpm) regulation and ship propulsion optimisation. For AC motors to fulfil this requirement it is necessary to implement complex software and hardware device also known as inverter which can generate polyphase system with alternating frequency and voltage from energy provider as battery system or, in general, on board DC bus [22].

It is possible to supply electric motor with electric energy through several sources. The most common use is battery storage that can be charged via solar photovoltaic cells, wind generator, fuel cells or in case of hybrid systems, directly from diesel-generators [27], [29].

In this work software development methodology for the described specific real-time closed-loop control system will be proposed implementing control mechanisms into the development process with an intention to guarantee efficient program code, full software and hardware compatibility, an adequate testing environment, good documentation and transparent traceability.

2. Related Work

Complexity of real-time closed-loop control systems and difficulties with system comprehension were recognized in [1] when authors presented RCS Refence Model Architecture with guidelines for definition of modular hierarchical structure of the system. The main idea was that if we focus on just one small part of the complex system, it is sufficient to understand functionality of that unity part and define its interaction with the complex system. In that sense we can reduce the complexity by observing separately different unity parts of the system. Today, modular design is accepted in many projects regarding control systems, which facilitates future upgrades and maintenance of the system [31], [32].

Quintero and Barbera [2] were first to establish activities in the development process of real-time closed-loop system (RCS) based on the mentioned architecture. RCS Methodology covers software development process and its implementation and testing on corresponding simulators and general purpose actual hardware. Thanks to software modularity requirement, methodology is based on rapid prototyping incremental software process model. Huang, Scott, Messina, Juberts and Quintero [3], Albus and Barbera [4] and Horst [5] applied those defined activities for development of real projects.

Klee [6] and Pintos [7] propose improvements in verification phase with use of MATLAB and Simulink tool for system simulation. The advantage of this approach is that the need for developing specific software simulators is eliminated, thus resulting in reducing development time and cost.

Overviews of linear and incremental software development process models like Pressman [9], Abrahamsson, Salo, Ronkainen and Warsta [23], Tsui, Karam and Bernal [24], Fitzgerald and Stol [25] and Martin [18] cover development phases and activities, but since real-time closed-loop control systems encompass hardware and software, the development methodologies are not fully applicable. Consulting specifically oriented works according to the case study [8] and [33], it is obvious that case oriented tools are much more efficient in specific software development process. Actual industrial needs require specific development methodology and accordingly, we propose a development methodology designed for special case of real-time closed-loop control systems.

Modern marine propulsion systems are oriented toward electric concepts. A lot of actual work and effort is placed in developing new technology for the future marine projects and one of the most interesting is using permanent magnet technology in propulsion machinery. These new marine concepts require strong real-time closed-loop control software development support. Proposing a new, case oriented, expanded software development methodology would be highly appreciated in project management tools resources.

Here we propose Expanded Development Methodology DEM4RTS, based on RCS Methodology which generally systematizes the development process but implies the use of commercial hardware and doesn't prescribe steps for its development. Commercial, general-purpose, hardware contains standard set of modules that can support most common applications. For special applications, general-purpose hardware mostly lacks specific modules while some existing modules are irrelevant and even useless. In this case hardware should be specifically designed and developed parallel to the software. Since our case study requires application specific hardware, we prescribe life-cycle phases and activities specialized for development of complete control system (hardware and software), managing this new generation electric motor drives, which are classified as real-time closed-loop control systems.

3. Advance Permanent Magnet Electric Motor Drive Software Structure

AC synchronous electric motor with permanent magnets is a highly sophisticated technical system where processes are controlled through software in real-time by combining part of the functionality of the electric machine into a digital algorithm commonly known as "electronic commutation". The whole system is composed of electromechanical components (electric motor) and electronic control unit [20].

The electronic control unit is composed of: power electronic circuits (rectifier and inverter), signal conditioning circuits, sensors and embedded industrial computer hardware structure (CPU). The control software managing these systems is the focus of our research and case oriented interest.

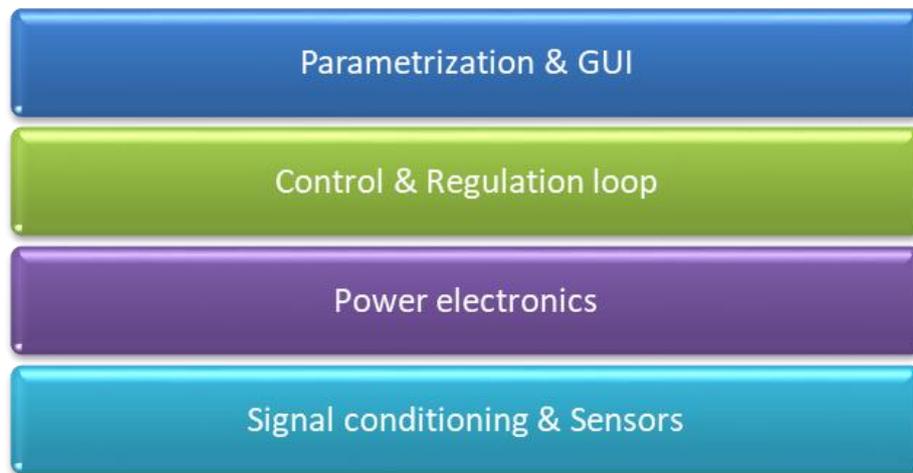


Fig. 1. Structure of control software

Structure of the software managing the described systems is firmly tied to hardware specifics on which it is implemented. Therefore, the development process must include tracking hardware and software compatibility. Specific requirements set on system management software, except for the basic functionalities, are:

1. extremely compact program code
2. high speed of execution
3. code modularity

Development process of permanent magnet electric motor drive system can be divided in two functional units: development of electric motor and development of electric control unit. They are concerned with development of different system parts (Table 1.).

Development	Development of electric motor (Figure 2.)	Electrotechnical part
		Mechanical part
	Development of electronic control unit (Figure 3.)	Power electronic circuits
		Measuring and sensors electronics
		CPU Hardware
		Software

Table 1. Development process of permanent magnet electric motor drive

Even though the electric motor is an important and inseparable system component, we do not deal with its analysis and management of development in this paper, since these activities require other specialities and competencies.



Fig. 2. Electric motor



Fig. 3. Electric control unit

4. Life-cycle Phases and Development Activities

As previously mentioned, controlling the Permanent magnet electric motor drives is carried out through electric control unit consisting of application specific hardware and control software. Thus, development process covering the entire life-cycle of the system can be observed through 7 phases (Figure 4.).

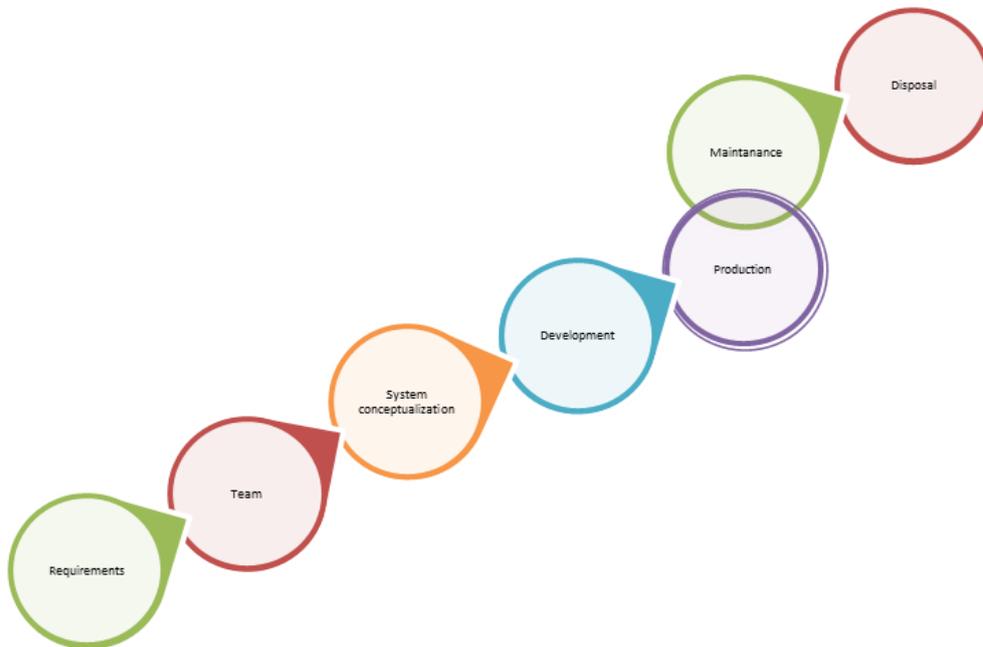


Fig. 4. ECU life-cycle phases

Each phase is further divided into activities (Table 2.).

1. Specification of project requirements
1.1. Definition of project requirements and decomposition into tasks
1.2. Setting time line, priorities, deadlines and planning task execution
1.3. Documentation
2. Development teams
2.1. Definition of competencies, skills and knowledge needed for system development
2.2. Formation of development teams
2.3. Responsibilities assignment for each team
3. Software and hardware conceptualization
3.1. Identification of hardware and software modules specified by project requirements
3.2. Definition of bounds and way of communication between software and hardware
3.3. Setting control points on the time line

4. Hardware and software development
4.1. Production of software modules and hardware
4.2. Tracking of software and hardware compatibility based on control points schedule
4.3. Filling out accompanying forms
4.4. Testing software modules on software simulators
4.5. Documentation
4.6. Implementing and testing software modules on prepared hardware
4.7. Documentation
4.8. Testing software modules in their real processes
4.9. Documentation
5. Production
5.1. Development of control tools for specific working place
5.2. Filling out testing report
5.3. Documentation
6. Maintenance
6.1. Development of maintenance software modules
6.2. Filling out maintenance report
6.3. Documentation
7. Disposal
7.1. System shut down and proper disposal of harmful parts

Table 2. Activities of the ECU development process

4.1. Specification of Project Requirements

In classic software development process where requirements are set and defined by interviewing users and stakeholders, it is of utmost importance to collect, analyse, document, validate and manage them in the correct manner. The discipline of Requirements engineering (RE) is concerned about this issue [19]. In the described systems, project task is defined by market needs, the state of competitiveness and development of technological background that set requirements for needed hardware and indirectly for implemented management software algorithm. To satisfy the defined requirements through the development process, hardware is specifically designed and developed parallel to the software.

Project tasks are divided in subtasks. By defining subtasks, it is possible to determine subtasks dependencies allowing the assessment if comparative or sequential execution of certain tasks is possible. This makes the foundation for priority assignment and deadlines setting that can be displayed on the time line for clear visualisation and tracking of the state of completion. Defined requirements are documented in requirements specification which is in the form of scenarios and serves as a reference document in development process.

4.2. Development Teams

The development of these complex systems requires different skills, competencies and knowledge defined by requirements specification and project domain. Therefore, the team consists of experts in a given area. To ensure quality development and future maintenance, process must be supported by bug tracking and versioning systems [10] which are fundamental for team co-operation. These supporting systems provide tracking of programming code versions taking care of adequate merging and task assignment, state and solution.

Team members can often be dislocated, and their collaboration supported by these systems makes it possible for the whole team to be informed about every assignment and its status [11]. Additionally, over historical data, one can apply classification algorithms like Vector Space Model, Support Vector Machine, Regression and Classification Trees and k-nearest neighbour [12] to obtain an intelligent agent which takes care of automatic task distribution.

4.3. Software and Hardware Conceptualization

This is a phase in which the model of the system is generated by identification of necessary hardware. Since those systems require high efficiency code, software modules are directly tied to and implemented on specific hardware structure, developed parallel to the software. For that reason, compatibility checks are integrated into development process through compliance tests which take place in set control points. For purpose of setting the preconditions for traceability implementation, the results of the test are recorded in the prescribed forms.

4.4. Hardware and Software Development

Development process is based on the system model designed in the previous phase. During the development process, specific development tools installed on standard computers are used. Tools usually combine text editor, linker, compiler, debugger, software emulator and In-circuit emulator (ICE) data communication module.

In early development stages when hardware structure is not yet developed, evaluation boards can be used. These boards support hardware functionality of the processor itself, which means that they support development of initial software modules like: initialization, basic processor parametrisation (A/D and D/A converters), serial communication modules, timers, etc. For further development, it is necessary to work with specific hardware modules defined in requirements specification.

Because of software being implemented on specific hardware, testing their interaction is very important. For this purpose, before actual implementation, testing is performed on software emulators which simulate the processor performance and give access to the current state of the variables as well as parameters to the developer. By placing breakpoints in program code, it is possible to track real-time state of variables and parameters to ensure correct execution timing.

Since the simulation does not demonstrate the real process completely, software implemented on specified hardware structure needs to be tested in real working environment and in real-time. To be able to conduct this test, special In-circuit emulator (ICE) should be used via standard JTAG (Join Test Action Group) connector - derived on all modern processors. By connecting ICE to the processor, it is possible to monitor processors internal data and analyse software execution in real-time through adequate software tools installed on working station.

However, this statement should be taken with caution especially in time-critical processes. Because of data sampling break points, the testing module execution time in some cases is not completely identical to the execution time of the same module in real process. After conducting all tests and getting positive testing results, the prepared hardware modules with implemented software become a part of the system.

4.5. Production

After the system is developed and tested, the production can start. During this phase, the specified hardware is produced and control software implemented. Every produced unit must be tested by a specialized engineer. Testing is performed within a proclaimed parameter for establishing the limits of endurance.

For testing purposes, assisting software tools like micromodules, one-time programming logical structures are being developed. Every engineer in a different working place is testing a different part of the system. Thus, for this assisting software tools, custom work place oriented graphical user interface (GUI) with built-in controls and allowed parameter range is required.

In this phase, the results of performed tests must be recorded for future easy access, creation of statistical reports and implementation of traceability. For that reason, test engineers fill out predefined digital forms with necessary data and store them in the database.

4.6. Maintenance

Once the produced system is ready to function in its real environment, the maintenance phase begins. In the maintenance phase, we distinguish three kinds of actions:

1. On-site commissioning - when system is starting up for the first time, the user specific parameters are set
2. In application field, possible system malfunction may occur, so part of the system needs to be repaired.
3. Periodical checking and servicing of the system

Before any action can be taken, the system is connected to the computer with installed specifically developed testing software tools. Unlike testing tools in the production phase, these tools test functionality of the system, considering the specific application, rather than endurance. During testing the whole system must be active.

After connecting, the test engineer can access system parameters through custom GUI having:

1. a good overview of all parameters
2. data logging capability
3. overview of error history

Upon the completion of the testing, a decision regarding the type of action is made. Since the results of testing are a useful source for different analysis and future predictions and overall system improvement, they are stored in database for availability and consistency.

4.7. Disposal

At the end of system life-cycle when maintenance actions are no longer worthwhile, the system must be extinguished and replaced. The strategy and the accompanying documentation have to be predicted regarding the remaining components and waste disposal. Special attention should be paid to certain physical components of the system which cannot be reused. If some of them are harmful to the environment, it is very important to take care of their proper disposal.

The Law on Sustainable Waste Management [13], which introduces measures to eliminate the harmful effects of waste on human life and the environment, prescribes principles, objectives, methods and obligations in the process of disposal of hazardous waste and defines the accompanying documentation [14].

Upgrading the support system in the development process with program support in the process of waste management [17] would enable manufacturers of such systems to keep records of generated waste and to create all necessary reports, in accordance with the adopted law [15], [16].

5. Conclusion

AC synchronous electric motor with permanent magnets is a highly sophisticated technical system and due to its complexity, it strives for systemized development approach. So far, it is known that the development process of these systems consists of software and hardware development, but previously conducted research and analysis resulted in defining only general activities of the development process based on commercial general-purpose hardware.

In this paper we propose a specific methodology for development of special case of real-time closed-loop control systems. By introducing the Extended Methodology DEM4RTS the following implications can be observed: better collaboration and organisation for dislocated team of experts, encouragement of team members' specialization in a narrow area, well documented development activities, facilitated monitoring of the development process through achieved traceability, systematization of ad-hoc approach of these complex systems. Additionally, as a negative outcome, one can single out that, in order to achieve traceability, team members need to invest additional effort in the described activities, and from their perspective that might seem as extra work unrelated to project tasks.

Prescribing life-cycle phases in development process where application specific hardware integrates layered software with very specific requirements, efficient case-oriented development methodology is obtained with implemented necessary control mechanisms.

Methodology covers the entire life-cycle of the system. However, its limitation is related to its primal focus on the software development phase and its tight connection to a specific hardware. In our further research, we will propose defining steps that track application specific hardware development, which, in combination with our Extended Methodology, would offer a complete managing software solution for business companies developing real-time closed-loop control systems.

6. References

- [1] Barbera, T., Fitzgerald, M. L., & Albus, J. S. (1982, April). Concepts for a real-time sensory-interactive control system architecture. In Fourteenth Southeastern Symposium on System Theory.
- [2] Quintero, R., & Barbera, T. (1992). A Real-Time Control System Methodology for Developing Intelligent Control Systems (No. NIST Interagency/Internal Report (NISTIR)-4936).
- [3] Huang, H. M., Scott, H., Messina, E., Juberts, M., & Quintero, R. (2002). Intelligent System Control: A Unified Approach and Applications. In *Expert Systems* (pp. 197-265).
- [4] Albus, J. S., & Barbera, A. J. (2005). RCS: A cognitive architecture for intelligent multi-agent systems. *Annual Reviews in Control*, 29(1), 87-99.
- [5] Horst, J. A. (2000). Architecture, design methodology, and component-based tools for a real-time inspection system. In *Object-Oriented Real-Time Distributed Computing, (ISORC 2000) Proceedings. Third IEEE International Symposium on* (pp. 296-303). IEEE
- [6] Klee, A. (2005). Development of a motor speed control system using matlab and simulink, implemented with a digital signal processor, MSc. Thesis, Electrical and Computer Engineering, College of Engineering and Computer Science, Orlando, Florida
- [7] Pintos, B. (2015.). New approach for engine control system software development. 6. CIMAC CASCADES, AVL Software and Functions
- [8] Pavlić, M., Pošćić, P., & Marinović, M. (2006). Education model for MIRIS methodology. *Informatologija*, 39(4), 251-255.
- [9] R. S. Pressman (2010). *Software engineering: a practitioner's approach*, The McGraww-Hill Companies Inc., ISBN 73375977
- [10] Canfora, G., & Cerulo, L. (2005, September). Impact analysis by mining software and change request repositories. In *Software Metrics, 2005. 11th IEEE International Symposium* (pp. 9-pp). IEEE.
- [11] Di Lucca, G. A., Di Penta, M., & Gradara, S. (2002, October). An approach to classify software maintenance requests. In *Software Maintenance, 2002. Proceedings. International Conference on*(pp. 93-102). IEEE
- [12] Harrington, P. (2012). *Machine learning in action (Vol. 5)*. Greenwich, CT: Manning.

- [13] <http://www.mzoip.hr/hr/otpad/odrzivo-gospodarenje-otpadom.html> (2017.), Ministarstvo zaštite okoliša i energetike, Accessed on: 2018-06-19
- [14] http://www.fzoeu.hr/hr/gospodarenje_otpadom/ (2017.), Fond za zaštitu okoliša i energetske učinkovitost, Accessed on: 2018-06-19
- [15] http://www.fzoeu.hr/hr/propisi/propisi_iz_podrucja_zastite_okolisa_i_gospodarenja_otpadom/ (2017.), Fond za zaštitu okoliša i energetske učinkovitost, Accessed on: 2018-06-19
- [16] <http://www.mzoip.hr/hr/otpad/obraci-i-upute.html> (2017.), Ministarstvo zaštite okoliša i energetike, Accessed on: 2018-06-19.
- [17] Blašković, K. (2012) Computer supported ecologic waste management, SORTA 2012. Proceedings ISBN 978-953-7738-20-4
- [18] Martin, R. C. (2002). Agile software development: principles, patterns, and practices. Prentice Hall.
- [19] Ambreen, T., Ikram, N., Usman, M., & Niazi, M. (2016). Empirical research in requirements engineering: trends and opportunities. *Requirements Engineering*, 23(1), 63-95.
- [20] Bose, B. K., & Bose, B. K. (Eds.). (1997). Power electronics and variable frequency drives: technology and applications (Vol. 2, pp. 1-8). New York: IEEE press
- [21] Bose, B. K. (2009). Power electronics and motor drives recent progress and perspective. *IEEE Transactions on Industrial Electronics*, 56(2), 581-588.
- [22] Leonhard, W. (2012). Control of electrical drives. Springer Science & Business Media.
- [23] Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.
- [24] Tsui, F., Karam, O., & Bernal, B. (2016). Essentials of software engineering. Jones & Bartlett Learning.
- [25] Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176-189.
- [26] Mele, C., Pels, J., & Polese, F. (2010). A brief review of systems theories and their managerial applications. *Service Science*, 2(1-2), 126-135.
- [27] Firoozian, R. (2014). Servo motors and industrial control theory. Springer.
- [28] Mi, C., & Masrur, M. A. (2017). Hybrid electric vehicles: principles and applications with practical perspectives. John Wiley & Sons.
- [29] Bose, B. K. (2009). Power electronics and motor drives recent progress and perspective. *IEEE Transactions on Industrial Electronics*, 56(2), 581-588.
- [30] Kaewjinda, W., & Konghirun, M. (2007). Vector control drive of permanent magnet synchronous motor using resolver sensor.
- [31] Andreev, V. & Kim, V. (2016). Control System and Design of the Motion Module of a Heterogeneous Modular Mobile Robot, Proceedings of the 27th DAAAM International Symposium, pp.0586-0594, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-08-2, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/27th.daaam.proceedings.086.
- [32] Solovyev, M., Vorotnikov, A., Klimov, D., Kovalskii, V. & Poduraev, Y. V. (2017). Control System of the Articulated Arm Braking Mechatronic Machine (AABMM), Proceedings of the 28th DAAAM International Symposium, pp.1002-1009, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/28th.daaam.proceedings.139.
- [33] Davydov, D.; Eprikov, S.; Kirsanov, K. & Pryanichnikov, V. (2017). Service Robots Integrating Software 4nd Remote Reprogramming, Proceedings of the 28th DAAAM International Symposium, pp.1234-1240, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734- 11-2, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/28th.daaam.proceedings.172.