The definition of procedural knowledge in distributed mechatronic systems

Jerzy Świder, Mariusz Hetmańczyk

Faculty of Mechanical Engineering, The Silesian University of Technology

Abstract: The article describes software features of integrated control and diagnostic blocks dedicated to the ProfiBus DP network, in relation to the definition of the procedural knowledge enabling a diagnostic inference. Control and diagnostic blocks have been developed in order to an application in the distributed drives system, where amount of work connected with programming exceed skills of the average class programmer. The paper presents description of integrated control and diagnostic functions developed by the authors, internal structures of diagnostic functions that automatically detect faults of supervised units (MASTER or SLAVE of the ProfiBus network).

Keywords: diagnostic inference, procedural knowledge, distributed systems

1. Introduction

A maintenance of a failure-free operation of distributed mechatronic systems belongs to the very complex engineering tasks, containing all elements of the activities carried out in centralized systems, including: ensuring a continuity of work, providing of operational safety, rapid detection of errors and their sources, etc.

The definition of the useful procedural knowledge can be based on high order languages and PC computers [1]. The significant development of mechatronic devices allows for partial transfer of responsibility for the aggregation of knowledge on the elements of mechatronics (microprocessors of mechatronic devices).

2. An acquisition and a distribution of data based on Programmable Logic Controllers with fast processors

The implementation of control algorithms recorded in advanced Programmable Logic Controllers allows to force certain operating conditions that cause the occurrence of the corresponding diagnostic premises, during the realization of a active diagnostic experiment.

The combination of the declarative knowledge contained in the advisory system allows for comprehensive diagnosis of the problem of distributed systems, maintenance planning and prediction of future states, assuming the exchange of data via industrial bus networks.

Consideration of the problem connected with an integration of the control functionality of an industrial facility and its diagnosis is a matter of an automation, a construction and a operation of machines and a metrology.

Through the developed standards (IEC 61158, IEC61784) all units included in each configuration can be divided into classes, according to the following scheme:

- DP MASTER class 1 (DPM1) – in the form of a Programmable Logic Controller,
- DP MASTER class 2 (DPM2) – all the units used for programming, diagnostics and configuration of network elements,
- DP SLAVE class A – industrial sensors or actuators with the ProfiBus interface.

For the problem under consideration the whole structure of the data acquisition is based on dedicated software blocks, divided by types, characterized in terms of performing functions.

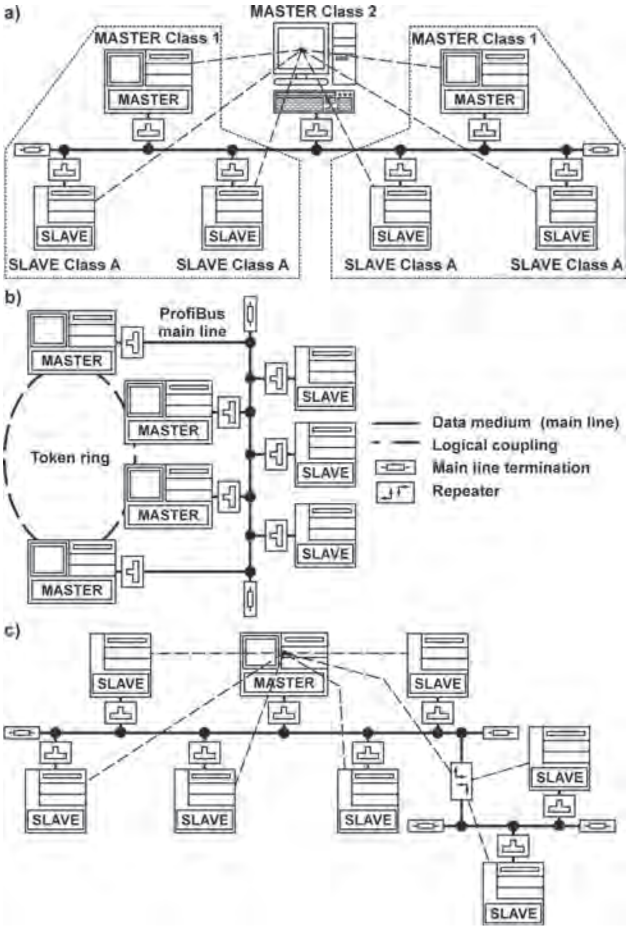


Fig. 1. Variations of the ProfiBus network topology: a) line (a Multi-MASTER type), b) ring, c) line (Mono-MASTER type)

Rys. 1. Odmiany topologii sieci ProfiBus, topologia: a) magistralna (w układzie Muli-MASTER), b) pierścieniowa, c) magistralna (typ Mono-MASTER)

Developed parametric blocks, allow to simplification of the programming stage of distributed drive system, and additionally, can extend the functionality of the program for ProfiBus network diagnostics and system units.

All blocks have been developed as elements characterized by a universal nature (PLC programming environments, a compliance with the standard IEC 61131 syntax). Authors assumed a universality of an implementation in different PLC environments and at the level of an application of the SLAVE units, with different functions of operation (i.e. the number of input and output parameters).

The basic factors that connect all the supported devices are:

- data exchange via the ProfiBus DP network (hardware network interface),
- a network Configuration MASTER – n × SLAVE type (Multi-MASTER configuration type has not been verified, but the convention of the algorithm syntax construction does not exclude such a system configuration).

Simplified configuration of the one program line is presented in fig. 2. Correct operation of data processing of defined algorithm requires remaining unchanged syntax.

Except of the proposed implementation of an individual program row, authors minimized the possibility of a configuration error by introducing the standard array data types DUT (Data Unit Types), which are useful at the stage of building the algorithmic structure of the integrated system of diagnostics and control.

Basic assumptions of the method can be characterized in several main groups, which include several points:

- each SLAVE station is identified by the ordinal number in the network,
- a proper operation requires the use of three basic blocks,
- additional features (additional diagnosis of unit, alarms control, etc.) needs to be expanded by additional function blocks,
- it is possible to program the system without the knowledge of the network structure and the whole system (part of the block is universal and can be used to a definition of the algorithm for the operation of any type of SLAVE units, appropriate control blocks have been developed to meet the individual requirements of selected units and have a narrow or a complete lack of universality of an application).

Configuration of DUT tables requires placing the program block, and then fulfillment the standard DUT tables. Thus avoiding the mistakes in the phase of an type assignation of the considered variable which reduces setup time of the system. The following assumptions have been adopted:

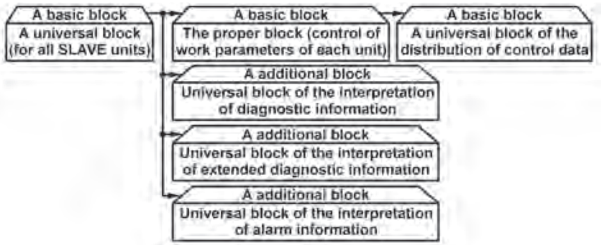
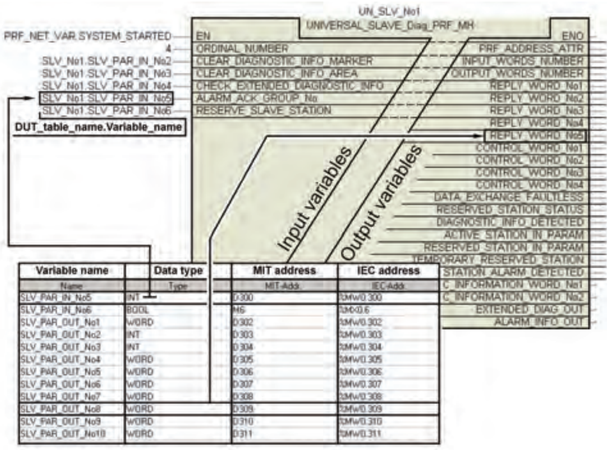

A basic block
A universal block (for all SLAVE units)
A basic block
The proper block (control of work parameters of each unit)
A basic block
A universal block of the distribution of control data
A additional block
Universal block of the interpretation of diagnostic information
A additional block
Universal block of the interpretation of extended diagnostic information
A additional block
Universal block of the interpretation of alarm information

Fig. 2. A simplified view of the organization of a single program line

Rys. 2. Uproszczony widok składni pojedynczej linii programu



Variable name	Data type	MIT address	IEC address
Name	Type	MIT-Addr.	IEC-Addr.
SLV_PAR_IN_No5	INT	D300	%MW0.300
SLV_PAR_IN_No6	BOOL	M6	%MX0.6
SLV_PAR_OUT_No1	WORD	D302	%MW0.302
SLV_PAR_OUT_No2	INT	D303	%MW0.303
SLV_PAR_OUT_No3	INT	D304	%MW0.304
SLV_PAR_OUT_No4	WORD	D305	%MW0.305
SLV_PAR_OUT_No5	WORD	D306	%MW0.306
SLV_PAR_OUT_No6	WORD	D307	%MW0.307
SLV_PAR_OUT_No7	WORD	D308	%MW0.308
SLV_PAR_OUT_No8	WORD	D309	%MW0.309
SLV_PAR_OUT_No9	WORD	D310	%MW0.310
SLV_PAR_OUT_No10	WORD	D311	%MW0.311

Fig. 3. Adopted implementation manner of DUT tables on the example of the universal diagnostic block of the SLAVE unit

Rys. 3. Zaimplementowana metoda tablic DUT, na przykładzie uniwersalnego bloku diagnostycznego jednostki SLAVE

- variables are divided into two groups (input and output variables),
- the order numbers are counted from the top, while the EN entrance and the ENO output are indexed with zero marker,
- types of arrays have been developed for all the blocks for control and diagnostic purposes of distributed drives,
- types of variables contained in DUT tables with appropriate serial numbers correspond to the types of variables assigned to the blocks.

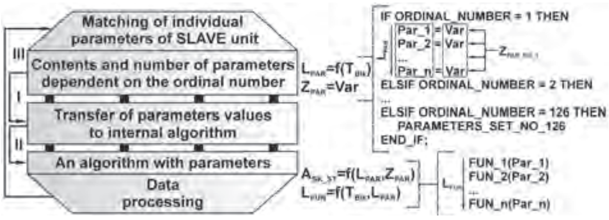

Matching of individual parameters of SLAVE unit
Contents and number of parameters dependent on the ordinal number
Transfer of parameters values to internal algorithm
An algorithm with parameters
Data processing
III
I
II
$L_{PAR}=f(T_{Blk})$
$Z_{PAR}=Var$
IF ORDINAL_NUMBER = 1 THEN
Par_1 = Var
Par_2 = Var
...
Par_n = Var
$Z_{PAR\ No\ 1}$
ELSIF ORDINAL_NUMBER = 2 THEN
...
ELSIF ORDINAL_NUMBER = 126 THEN
PARAMETERS_SET_NO_126
END_IF;
$A_{SK\_ST}=f(L_{PAR},Z_{PAR})$
$L_{FUN}=f(T_{Blk},L_{PAR})$
L_{FUN}
FUN_1(Par_1)
FUN_2(Par_2)
...
FUN_n(Par_n)

Fig. 4. The structure of modular blocks, where: L_{PAR} – the number of parameters, Z_{PAR} – the content parameters, T_{Blk} – block type, A_{SK_ST} – the internal structure of the algorithm, L_{FUN} – the number of functions

Rys. 4. Struktura bloków modułowych, gdzie: L_{PAR} – liczba parametrów, Z_{PAR} – zawartość parametrów, T_{Blk} – typ bloku, A_{SK_ST} – wewnętrzna struktura algorytmu, L_{FUN} – liczba funkcji

As an example of a parametric block with the modular structure the universal block of the SLAVE unit has been chosen. An internal structure of described block is composed of two basic functional components:

- a set of variables,
- an internal algorithm (i.e. program functions with attributes taken from the set of parameters).

The length of a set of parameters depends on the type of block, while internal values are assigned dynamically, depending on the ordinal number. The processing of internal data of parametric block is divided into steps (fig. 4):

I – the identification of the basic parameters (number and addresses stored in variables),

II – the transfer of ordered data cluster and the execution of the internal algorithm, III – returning to the module which identifying the basic parameters, with simultaneously checking of the content of the ordinal number.

The ordinal number of units is an equivalent of an identification number, but it is not always consistent with the numerical value of the network address. The concept of ordinal number On has been defined with the following assumptions [2, 3]:

$$\mathbf{O}_{\mathrm{n}} \in <\mathbf{1},\ \mathbf{125}> \quad (1)$$

$$\mathbf{O}_{\mathrm{n}} \in \mathbf{S}_{\mathbf{ADDR}} \quad (2)$$

$$\exists \mathbf{K}_{\mathrm{f}}(\mathbf{O}_{\mathrm{n}}) \in \mathbf{K}_{\mathbf{fdop}}\mathbf{:}\ \mathbf{O}_{\mathrm{n}} \neq \mathbf{A}_{\mathbf{PRF\_VAL}} \quad (3)$$

where: $\mathbf{O}_{\mathrm{n}}$ – the ordinal number, $\mathbf{S}_{\mathbf{ADDR}}$ – a set of numeric network addresses of individual units, $\mathbf{K}_{\mathrm{f}}(\mathbf{O}_{\mathrm{n}})$ – the program line configuration as a function of the ordinal number $\mathbf{O}_{\mathrm{n}}$, $\mathbf{K}_{\mathbf{fdop}}$ – a set of allowable configurations, $\mathbf{A}_{\mathbf{PRF\_VAL}}$ – the network address.

Condition 1 defines the number of network units, while the ordinal number is included in the set of network addresses (the relationship No 2). Condition 3 implies the possibility of any allocation of SLAVE network addresses. The maximum value of the ordinal number defines number of units in the configured system. The authors have identified three possible network configurations [3], the principle of an interpretation of the serial number defined in fig. 5.

Utilization of the described block enables an identification and programming of SLAVEs units without knowing the structure of the monitored system. The procedure of an identification of the SLAVE type unit comprises the following steps:

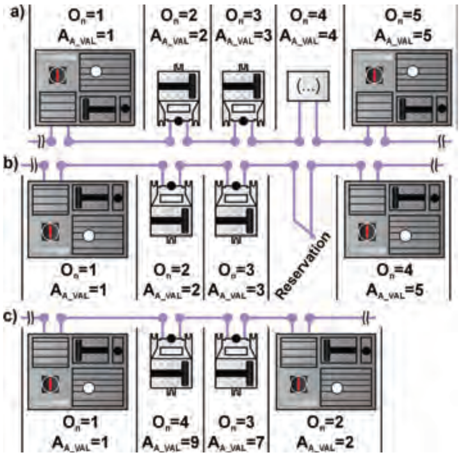


Fig. 5. Schematic interpretation of the ordinal number of the system: a) $O_n=A_{A\_VAL}$ and stations ordered according to a upward trend (step equal 1), b) with reservation of addresses numbers, c) $O_n \neq A_{A\_VAL}$ and stations arranged arbitrarily

Rys. 5. Schematyczna interpretacja liczby porządkowej w rozważanym systemie: a) $O_n=A_{A\_VAL}$ stacje uporządkowane zgodnie z trendem rosnącym (krok równy 1), b) z rezerwacją numerów adresów, c) $O_n \neq A_{A\_VAL}$ oraz stacje uporządkowane w sposób losowy

- the introduction of the universal block and configuration of all inputs and outputs (steps are simplified through the application of defined DUT tables),
- compilation and sending the program to a PLC controller,
- launching of the internal variables preview, which allows to the identification of a SLAVE unit on the basis of the number of words assigned to the parameters of I/Os.

List of parameters identifying the type of SLAVE units can be made by any concept, but the identification must be unambiguous and based on a set of individual parameters assigned to the unit, which can be defined as an ordered four:

$$\mathbf{F}_{\mathbf{int\_SLV}} = <\mathbf{A}_{\mathbf{PRF\_VAL}}, \mathbf{O}_{\mathrm{n}},\ \mathbf{L}_{\mathbf{SL\_Wej}},\ \mathbf{L}_{\mathbf{SL\_Wyj}}> \quad (4)$$

where: $\mathbf{L}_{\mathbf{SL\_Wej}}$ – the number of input words, $\mathbf{L}_{\mathbf{SL\_Wyj}}$ – number of output words.

On the basis of the interpretation of the contents of the $\mathrm{F}_{\mathrm{int\_SLV}}$ set it is possible an identification of any SLAVE type without the knowledge of the whole structure of the system. Comparing the ordered four of a diagnostic block and the configuration parameters it is easy to shown that the identification is correct, only if:

$$\begin{gathered} <\mathbf{A}_{\mathbf{PRF\_VAL}},\ \mathbf{O}_{\mathrm{n}},\ \mathbf{L}_{\mathbf{SL\_Wej}},\ \mathbf{L}_{\mathbf{SL\_Wyj}}> = \\ <\mathbf{A}_{\mathbf{PRF\_VAL\_konf}},\ \mathbf{O}_{\mathbf{n\_konf}},\ \mathbf{L}_{\mathbf{SL\_Wej\_konf}},\ \mathbf{L}_{\mathbf{SL\_Wyj\_konf}}> \Rightarrow \\ \mathbf{A}_{\mathbf{PRF\_VAL}} = \mathbf{A}_{\mathbf{PRF\_VAL\_konf}} \wedge \mathbf{O}_{\mathrm{n}} = \\ \mathbf{O}_{\mathbf{n\_konf}} \wedge \mathbf{L}_{\mathbf{SL\_Wej}} = \mathbf{L}_{\mathbf{SL\_Wej\_konf}} \wedge \mathbf{L}_{\mathbf{SL\_Wyj}} = \mathbf{L}_{\mathbf{SL\_Wej\_konf}} \end{gathered}$$

The need of a definition of interpretation rules relating to the type of unit results from the lack of a application universality of an appropriate diagnostic and control blocks.

Dual addressing errors which resulted in a two equal determinants sets, are detected by the universal diagnostic block and the diagnostic block of a MASTER station module.

In case of considered system configuration which performed a check of correctness of the data interpretation by internal algorithms, the following configurations were considered:

- ordered according to a upward trend – the addresses definition of step equals 1 and assumption $\mathrm{O}_{\mathrm{n}}=\mathrm{A}_{\mathrm{A\_VAL}}$,
- ordered according to a upward trend – the definition of addresses with a variable step (random omission of station addresses or reservation),
- random assignment of stations addresses (including reserved stations), in accordance with the condition specified by the formula:

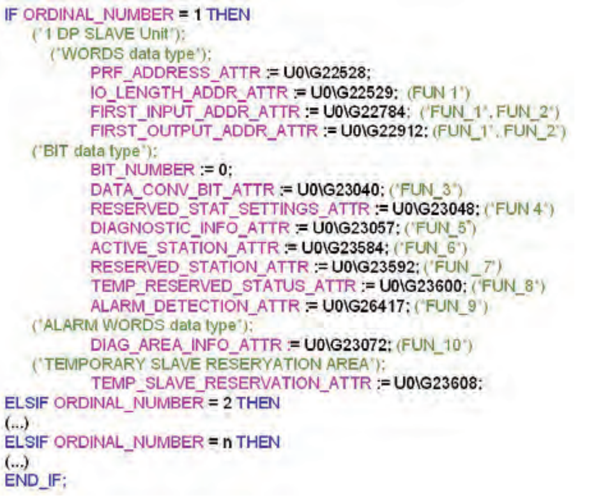
$$\begin{gathered} (\exists\ \mathbf{K}_{\mathrm{f}}\ ((\mathbf{O}_{\mathrm{n}}) \in \mathbf{K}_{\mathbf{Kfdop}}\mathbf{:}\ \mathbf{O}_{\mathrm{n}} \neq \mathbf{A}_{\mathbf{PRF\_VAL}}) \wedge \\ (\exists\ \mathbf{K}_{\mathrm{f}}\ ((\mathbf{O}_{\mathrm{n}}) \in \mathbf{K}_{\mathbf{Kfdop}}\mathbf{:}\ \mathbf{O}_{\mathrm{n}} = \mathbf{A}_{\mathbf{PRF\_VAL}}) \end{gathered} \quad (5)$$

The main problem is still the possibility of making a inappropriate configuration of single line of the control algorithm, which leads to a fault, in the form:

- improper formation of a set of control words: inconsistent number of input and output words,
- erroneous assignment of bits or values,
- incompatible content in relation to the words number,
- lack of correct performance of a complex algorithm of the drives system.

The described implementation recorded in ST language are shown in fig. 6.

Visible is the set of parameters taken each time from the global memory area (direct reference to the cluster data, a clear definition of the memory area independent on the type and the address assigned by the user variable).



```
IF ORDINAL_NUMBER = 1 THEN
   ('1 DP SLAVE Unit');
     ('WORDS data type');
           PRF_ADDRESS_ATTR := U0\G22528;
           IO_LENGTH_ADDR_ATTR := U0\G22529; (FUN 1')
           FIRST_INPUT_ADDR_ATTR := U0\G22784; ('FUN_1', FUN_2')
           FIRST_OUTPUT_ADDR_ATTR := U0\G22912: (FUN_1', FUN_2')
   ('BIT data type');
           BIT_NUMBER := 0;
           DATA_CONV_BIT_ATTR := U0\G23040; ('FUN_3')
           RESERVED_STAT_SETTINGS_ATTR := U0\G23048; ('FUN 4')
           DIAGNOSTIC_INFO_ATTR := U0\G23057; ('FUN_5')
           ACTIVE_STATION_ATTR := U0\G23584; ('FUN_6')
           RESERVED_STATION_ATTR := U0\G23592; ('FUN _7')
           TEMP_RESERVED_STATUS_ATTR := U0\G23600; ('FUN_8')
           ALARM_DETECTION_ATTR := U0\G26417; ('FUN_9')
   ('ALARM WORDS data type');
           DIAG_AREA_INFO_ATTR := U0\G23072; (FUN_10')
   ('TEMPORARY SLAVE RESERYATION AREA');
           TEMP_SLAVE_RESERVATION_ATTR := U0\G23608;
ELSIF ORDINAL_NUMBER = 2 THEN
(...)
ELSIF ORDINAL_NUMBER = n THEN
(...)
END_IF;
```

Fig. 6. The algorithm stored in ST language, with respect to the assumptions

Rys. 6. Algorytm zapisany w języku ST, zgodny z przedstawionymi założeniami

The presented notation is simple in the form and easy to read, but has a significant advantage in the form of the impossibility of making a mistake in the phase of a memory identification, which stores the values corresponding to the control and diagnostics of SLAVE units with the appropriate defined order number (network address).

3. Conclusions

In the study on usefulness procedural knowledge carriers has been defined the main drawback of distributed systems. Despite the diagnostic capabilities of individual SLAVE units, in certain specific cases, exist some problems of restoring the system to the state of a properly work. Occurrence of some errors that have been classified as complex, makes it necessary to re-configure the internal parameters of SLAVE units and their modification. Described action is possible only through the use of individual diagnostic socket of each station. Prioritizing of the system and the reference to the individual address assigned to the configuration of the ProfiBus network settings, ensures error-free data acquisition [4].

Bibliography

1. Świder J., Hetmańczyk M., *Computer aided diagnosis of frequency inverter states*, "Problems of Working Machines", Vol. 33, 2009, 57–66.
2. Świder J., Hetmańczyk M., *Adaptation of the expert system in diagnosis of the connection of the PLC user interface system and field level*, "Solid State Phenomena", Vol. 164, 2010, 201–206.
3. Świder J., Hetmańczyk M., *The computer integrated system of control and diagnosis of distributed drives*, 2011, Silesian Technical University Publishing (in Polish).
4. Świder J., Michalski P., *Diagnostic aid system for AS-Interface industrial networks*, 2008, Silesian Technical University Publishing (in Polish). ■

Definicja wiedzy proceduralnej w rozproszonych systemach mechatronicznych

Streszczenie: W artykule przedstawiono cechy programowe zintegrowanych bloków diagnostycznych i sterujących dedykowanych sieci ProfiBus DP, w odniesieniu do definicji wiedzy proceduralnej umożliwiającej wnioskowanie diagnostyczne. Bloki diagnostyczne i sterujące zostały opracowane pod kątem aplikacji w systemach napędów rozproszonych, w których ilość pracy związanej z wykonaniem struktury algorytmu sterującego przekracza możliwości przeciętnego programisty. Artykuł prezentuje opis zintegrowanych bloków diagnostycznych i sterujących, opracowanych przez autorów, oraz składnię funkcji diagnostycznych służących celom automatycznej detekcji błędów obsługiwanych jednostek (stacji MASTER lub SLAVE, sieci ProfiBus DP).

Słowa kluczowe: wnioskowanie diagnostyczne, wiedza proceduralna, systemy rozproszone

Prof. Jerzy Świder, PhD, DSc (Eng.)

He received an PhD degree in 1981 from the Silesian University of Technology, DSc. degree in 1992 and became a full professor in 2000. He is actually a head of the Institute of Engineering Processes Automation and Integrated Manufacturing Systems and director of the Congress – Education Center. His scientific interests are mechanics, mechatronics, processes automation, robotics and CAD/CAM systems. He is the author of numerous home and international publications in the field of robotics, mechanics, mechatronics, machine design and operation.

e-mail: jerzy.swider@polsl.pl

Mariusz Hetmańczyk, PhD

He received an PhD degree in 2011 from the Silesian University of Technology. He is actually a researchworker of the Institute of Engineering Processes Automation and Integrated Manufacturing Systems. His scientific interests are automatic control and engineering, mechatronics, industrial diagnostics, states prediction based on graph methods, MEMS technology. He is the author of numerous publications connected with computer aided diagnosis and prognosis of distributed mechatronic systems.

e-mail: mariusz.hetmanczyk@polsl.pl