

# Parallel implementation of linear repetitive processes identification using subspace algorithms

Dominik Kujawa

Institute of Control and Computation Engineering University of Zielona Góra  
ul. Podgórna 50, 65-246 Zielona Góra, Poland  
Email: dkujawa@weit.uz.zgora.pl, dominikinf@gmail.com

**Abstract**—This paper presents a new parallel approach to identification of linear repetitive processes based on subspace algorithms. Parallel realizations of these algorithms are tested on various graphic cards that use NVIDIA CUDA technology. The paper describes implementation of subspace identification algorithms and their parallel speedup, efficiency, throughput, and delay. The parallel approach to the identification of linear repetitive processes based on subspace methods, presented in this paper, is very useful not only for time invariant LRPs but also for processes with dynamics that changes rapidly from pass to pass. A simulation example is included to illustrate the effectiveness of the proposed approach.

**Index Terms**—state space models, subspace methods, identification algorithms, parameter estimation. state space models, subspace methods, identification algorithms, parameter estimation.

## I. INTRODUCTION

FOR SEVERAL years, the growth of maximum clock frequency of digital implementations has stopped, particularly in general purpose processors. This phenomenon is presented in [1]. It seems that further progress in computer system performance is still possible despite reaching the limit of switching frequency of digital integrated circuits. The further increase in computing power can be achieved by multiplying the number of computational units in multi-core general purpose Digital Signal Processors (DSP), Graphics Processing Units (GPU), and reconfiguration Field Programmable Gate Array (FPGA) chips [2].

Subspace identification methods are an attractive alternative to the well-known prediction error methods due to simple and general parameterizations in the MIMO case. They do not need any canonical parameterizations as well. Moreover, no nonlinear optimization is performed and reliable state-space models for complex multi-input multi-output dynamical systems are derived directly from the input-output data. Also, computational complexity of subspace identification methods is modest in comparison with the well-known prediction error methods [3], [4], [5], [10]. Subspace identification algorithms consist of two steps. In the first step, based on the input-output data, the repetitive process order is determined and the repetitive process sequence of states (N4SID algorithm) or the extended observability matrix of system (MOESP algorithm) are computed [7]. In the second step, the unknown system matrices are determined.

The repetitive control theory has been an area of intense research since the beginning of 1990's and considerable results have been achieved both in the analysis and synthesis problems, (see: [6]) for the actual state of art. Contrary to the LRP control theory, the identification of LRPs has attracted very limited attention. The aim of this paper is to propose a new approach to the identification of the LRPs based on subspace algorithms. In this approach, the order of a LRP and the unknown process matrices are determined based on the input and output sequences of the actual pass and the output sequence of the previous pass using parallel implementations of N4SID and MOESP identification algorithms.

Parallelization of algorithm for repetitive processes subspace identification requires creating and processing Hankel matrices. The number of processors that is simultaneously busy depends on the size of matrices storing information defining the spatial variable, and the time variable determining the position on the pass, while the length of each pass is finite [6].

This paper considers the number of processors used, speedup, efficiency, throughput and computing time of parallel implementations of subspace identification algorithms and compares them to the modified sequential versions of MOESP N4SID algorithms. They were both quantitative and qualitative indicators describing the identification repetitive processes. These algorithms are examined using the input-output data generated by the repetitive process simulator. The paper is organized as follows. Section 2 presents an introduction to the identification of discrete deterministic repetitive processes. Parallel problem identification was formulated, and its solution based on subspace algorithms is given in Sections 3 and 4. Section 5 presents the simulation results. The conclusions are given in Section 6.

## II. IDENTIFICATION ALGORITHMS

### A. Discrete linear repetitive process model

Consider the state-space model of a discrete linear repetitive process of the following form:

$$x_{k+1}(p+1) = Ax_{k+1}(p) + B_0y_k(p) + Bu_{k+1}(p) \quad (1)$$

$$y_{k+1}(p) = Cx_{k+1}(p) + D_0y_k(p) + Du_{k+1}(p) \quad (2)$$

where:

$0 \leq p \leq \alpha - 1 \in Z_+$  is the independent spatial or temporal variable,

$k \in Z_+$  is the current pass number,

$x_k(p) \in R^n$  is the state vector,

$y_k(p) \in R^l$  is the pass profile (output) vector,

$u_k(p) \in R^m$  is the input vector,

$A, B, B_0, C, D, D_0$  are matrices of appropriate dimensions.

To complete process description, it is necessary to specify the boundary conditions:

$$x_{k+1}(0) = d_{k+1}$$

$$y_0(p) = f(p)$$

where  $d_{k+1} \in R^n$  is a vector with known constant entries and  $f(p) \in R^l$

are known functions of  $p$ .

Define the following input Hankel block matrix

$$U_{0|2i-1} = \begin{bmatrix} u_{k+1}(0) & \dots & u_{k+1}(j-1) \\ y_k(0) & \dots & y_k(j-1) \\ \vdots & \vdots & \vdots \\ u_{k+1}(i-1) & \dots & u_{k+1}(i+j-2) \\ y_k(i-1) & \dots & y_k(i+j-2) \\ \hline u_{k+1}(i) & \dots & u_{k+1}(i+j-1) \\ y_k(i) & \dots & y_k(i+j-1) \\ u_{k+1}(i+1) & \dots & u_{k+1}(i+j) \\ y_k(i+1) & \dots & y_k(i+j) \\ \vdots & \vdots & \vdots \\ u_{k+1}(2i-1) & \dots & u_{k+1}(2i+j-1) \\ y_k(2i-1) & \dots & y_k(2i+j-2) \end{bmatrix} = \begin{bmatrix} U_p \\ U_f \end{bmatrix}$$

$$U_{0|2i-1} = \begin{bmatrix} u_{k+1}(0) & \dots & u_{k+1}(j-1) \\ y_k(0) & \dots & y_k(j-1) \\ \vdots & \vdots & \vdots \\ u_{k+1}(i-1) & \dots & u_{k+1}(i+j-2) \\ y_k(i-1) & \dots & y_k(i+j-2) \\ u_{k+1}(i) & \dots & u_{k+1}(i+j-1) \\ y_k(i) & \dots & y_k(i+j-1) \\ \hline u_{k+1}(i+1) & \dots & u_{k+1}(i+j) \\ y_k(i+1) & \dots & y_k(i+j) \\ \vdots & \vdots & \vdots \\ u_{k+1}(2i-1) & \dots & u_{k+1}(2i+j-1) \\ h_j y_k(2i-1) & \dots & y_k(2i+j-2) \end{bmatrix} = \begin{bmatrix} U_p^+ \\ U_f \end{bmatrix}$$

Define also the output block matrix  $Y_{0|2i-1}$

$$y_{0|2i-1} = \begin{bmatrix} y_{k+1}(0) & \dots & y_{k+1}(j-1) \\ \vdots & \vdots & \vdots \\ y_{k+1}(i-1) & \dots & y_{k+1}(i+j-2) \\ y_k(i) & \dots & y_{k+1}(i+j-1) \\ y_{k+1}(i+1) & \dots & y_{k+1}(i+j) \\ \vdots & \vdots & \vdots \\ y_{k+1}(2i-1) & \dots & y_{k+1}(2i+j-1) \end{bmatrix} = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}$$

$$y_{0|2i-1} = \begin{bmatrix} y_{k+1}(0) & \dots & y_{k+1}(j-1) \\ \vdots & \vdots & \vdots \\ y_{k+1}(i-1) & \dots & y_{k+1}(i+j-2) \\ y_{k+1}(i) & \dots & y_{k+1}(i+j-1) \\ y_{k+1}(i+1) & \dots & y_{k+1}(i+j) \\ \vdots & \vdots & \vdots \\ y_{k+1}(2i+1) & \dots & y_{k+1}(2i+j-1) \end{bmatrix} = \begin{bmatrix} Y_p^+ \\ Y_f \end{bmatrix}$$

The number of block rows  $i$  should be larger than the maximum order of the LRP. Define block Hankel matrices  $W_p$  and  $W_p^+$  consisting of  $U_p$  and  $Y_p$  and  $U_p^+$ , respectively:

$$W_p = \begin{bmatrix} U_p \\ Y_p \end{bmatrix} \quad (3)$$

$$W_p^+ = \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix} \quad (4)$$

The state-sequence matrix  $X_i$  is defined as

$$X_i = [x_{k+1}(i) \dots x_{k+1}(i+j-1)] \quad (5)$$

Define the extended observability matrix  $\Gamma_i$  and the reversed extended controllability matrix  $\Delta_i$ :

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \quad (6)$$

$$\Delta_i = [A^{i-1}[BB_0] \dots A[BB_0] [BB_0]] \quad (7)$$

Assume also that the pair  $\{A, C\}$  is observable and the pair  $\{A, [BB_0]\}$  is controllable. Define the lower block triangular Toeplitz matrix  $H_i$

$$H_i = \begin{bmatrix} [DD_0] & 0 & \dots & 0 \\ C[BB_0] & [DD_0] & \dots & 0 \\ CA[BB_0] & C[BB_0] & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{i-2}[BB_0] & CA^{i-3}[BB_0] & \dots & [DD_0] \end{bmatrix} \quad (8)$$

The above block Hankel matrices along with the extended observability matrix, the reversed extended controllability matrix and the lower block triangular Toeplitz matrix (3)–(8) play an important role in the development of subspace identification

methods. Following Theorem 1 (see: [8]), the state-space model (1) – (2) can be reformulated in a matrix form:

$$Y_p = \Gamma_i X_p + H_i U_p \quad (9)$$

$$Y_f = \Gamma_i X_f + H_i U_f \quad (10)$$

$$X_f = A_i X_p + \Delta_i U_p \quad (11)$$

where  $X_f = X_i$  and  $X_p = X_0$ .

### B. Identification Problem

Given  $\alpha$  measurements of the input  $u_{k+1}(p)$  and the outputs  $y_k(p)$  and  $y_{k+1}(p)$  measurements generated by the LRP (1) – (2) determine its order and the matrices  $A, B, B_0, C, D$  and  $D_0$  up to within a similarity transformation.

Assuming that the augmented input  $[u_{k+1}(p) \ y_k(p)]$  is persistently exciting of order  $2i$  and the intersection of the row space of  $U_f$  and the row space of  $X_p$  is empty, the unknown LRP matrices  $A, B, B_0, C, D$  and  $D_0$  can be computed based on the results of Theorem 2 (see: [8]). It can be done in two different ways using N4SID or MOESP. In both these algorithms, the order of the process (1)-(2) can be found based on inspection of the singular value decomposition of the matrix  $W_1 \theta_i W_2$

$$W_1 \theta_i W_2 = [U_1 U_2] = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \quad (12)$$

where  $W_1 \in R^{lixi}$  and  $W_2 \in R^{jxj}$  are the user defined weighting matrices and  $\theta_i$  is the oblique projection

$$\theta_i = Y_f / U_f W_p \quad (13)$$

The order of LRP is equal to the number of non-zero singular values in  $S_1$ . The extended observability matrix  $\Gamma_i$  is computed from the the following equation

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T \quad (14)$$

where  $T \in R^{n \times n}$  is an arbitrary non-singular similarity transformation matrix. N4SID calculates the state sequence  $X_i$  from the equation

$$X_i = \Gamma_i^\dagger \theta_i \quad (15)$$

where  $\Gamma_i^\dagger$  denotes the Moore-Penrose pseudo-inverse of the matrix  $\Gamma_i$ . Finally, the matrices  $A, B, B_0, C, D$  and  $D_0$  are determined solving the following set of equations

$$\begin{bmatrix} A & [BB_0] \\ C & [DD_0] \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} \quad (16)$$

The state sequence  $X_{i+1}$  is calculated from the equation

$$X_{i+1} = \Gamma_{i-1}^\dagger \theta_{i-1} \quad (17)$$

where  $\Gamma_{i-1}$  is the extended observability matrix  $\Gamma_i$  without the last  $l$  rows and  $\theta_{i-1}$  is the oblique projection

$$\theta_{i-1} = Y_f^- / U_f^- W_p^+ \quad (18)$$

In MOESP, the unknown LRP matrices are determined in two steps. In the first step, A and C are calculated from the extended observability matrix  $\Gamma_i$ . In the other step,  $B, B_0, D$ , and  $D_0$  are calculated solving a set of equations.

### III. PARALLEL IMPLEMENTATION OF MODIFIED N4SID AND MOESP ALGORITHMS

The modified N4SID and MOESP algorithms can be partitioned into the following three modules:

- Input module
- Matrix calculation modul
- Model testing modul

The data input module performs pre-processing operations in spatial context in fine-grained pipelined architecture. The detailed time analysis including technological factors can be found in [9]. The delay in the operations of fine-grained data (in cycles) can be expressed as

$$F_t = \frac{\alpha - 1}{2} + r \quad (19)$$

where  $r$  is the delay of additional variables in the passes. The transport delay can be determined from (19), on basis of the pass length. The component  $r$  corresponds to the delay resulting from the use of additional variables, synchronizing data. They are used to distribute evenly the propagation time between the elements in the matrices. The other modules are parallelized in a coarse-grained architecture. The transport delay for operation with coarse-grained data (in cycles) is

$$L_t = \alpha + r \quad (20)$$

Modules of subspace identification algorithms can be partitioned into the following classes performing parallel operations:

- Class Hankel
- Class oblique projection
- Class SVD
- Class observability
- Class controllability
- Class ABCD

The UML activity diagram (Fig. 1) presents parallelization of block Hankel matrices for subspace identification of repetitive processes.

### IV. PARALLELIZATION LEVEL INDICATORS

System architecture can be adapted to a particular computational task in itself is not a determinant of performance or efficiency.

$$y_{k+1}(p) = F^l(u_k(p)) || F^{l-1}(u_k(p)) || \dots || F^1(u_k(p)) \quad (21)$$

|| – executing operations simultaneity

$F^\sigma$  – operator,  $\sigma \in (1, \dots, l)$ .

Many operations can be performed simultaneously at the inputs (21).

$$S_m = \frac{T_1}{T_m} \quad (22)$$

where:

$S_m$  – the speedup ,

$T_1$  – the execution time of the sequential algorithm,

$T_m$  – the execution time of the parallel algorithm with  $m$  processors

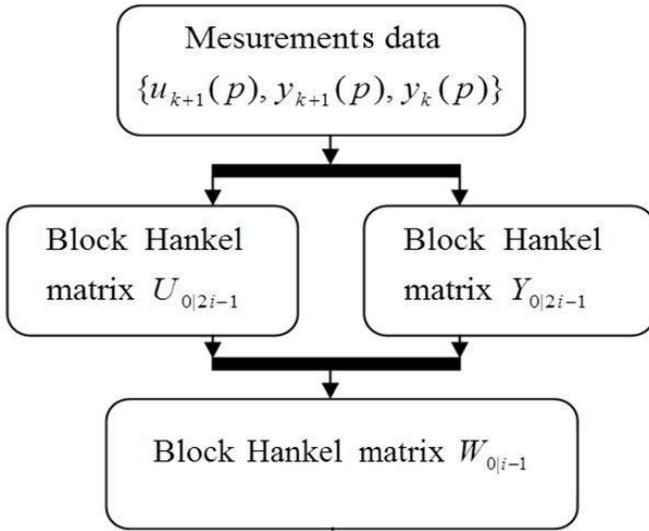


Fig. 1. UML activity diagram parallelized block Hankel matrices

For the evaluation of computational tasks performance in multi-core systems speedup formula was used, (22).

In multiprocessor systems, the actual value of the speedup is less than the theoretical (based on the number of processors) due to communication overheads and the need to share resources such as memory and buses. This property describes a model of efficiency, which is a measure of concurrent use of resources

$$E_m = \frac{S_m}{m} \quad (23)$$

where:

$E_m$  – the speedup efficiency with  $m$  processors  $E_m \in (0;1)$ .

In the ideal case, the efficiency (23) is 1.0 which means that the speedup is proportional to the number of processor or computing elements.

In the assessment of computing activity is often used as a criterion of time needed to process a specific task or quantum computing. The concept of throughput was introduced, which corresponds to the number of data processed per time unit.

$$P = \frac{D_n}{t} \quad (24)$$

where:

$P$  – throughput,

$D_n$  – the number of computed data,

$t$  – measurement time.

Relative to the computing system has the ability to compute the limit, usually in the long term. Throughput (24) relative to the computing system has the ability to compute the limit, usually in the long term.

## V. SIMULATION EXAMPLE

Consider a simple example to illustrate the proposed approach. The following LRP of the fourth order is identified:

$$A = \begin{bmatrix} 1.516 & -0.755 & 0.125 & -0.001 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.048 & 0.072 & -0.006 & -0.001 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad B_0 = \begin{bmatrix} -0.571 \\ -0.142 \\ -0.453 \\ -0.258 \end{bmatrix} \quad D = [0] \quad D_0 = [0.2]$$

with zero boundary conditions  $y_0(p) = 0, p = 1, \dots, 4 \times 10^5$ , and the initial conditions  $x_{k+1}(0), k = 1, \dots, 20$ , defined as a uniformly distributed pseudorandom sequence on the interval  $[0,2.5]$ . The LRP is exited with a uniformly distributed pseudorandom sequence on the interval  $[0,1]$  and its output disturbed by a pseudorandom sequence of normal distribution with mean 0 and standard deviation 0.0001. To identify selected data from two consecutive passes and N4SID and MOESP parallelized algorithms were used. Computing schemes of parallelized algorithm is:

Stage A - Input-output block Hankel matrices

Stage B - SVD, observability matrix, controllability matrix, calculation of  $A, B, B_0, C, D, D_0$  and  $QSR$  matrices

Stage C - Test module

The tables 1,2,3,4 show the comparison of the indicators of speedup, throughput, delay of parallel performance algorithm for the sequential algorithm.

TABLE I

EVALUATION OF THE PARALLEL ALGORITHM—SPEEDUP FOR  $m = 255$

Parallel algorithm			
Stage	A	B	C
	0,086[s]	0,105[s]	0,049[s]

Sequential algorithm			
Stage	A	B	C
	0,61[s]	0,42[s]	0,34[s]

TABLE II

EVALUATION OF THE PARALLEL ALGORITHM—EFFICIENCY FOR  $m = 255$

Stage	A	B	C
	0,028[s]	0,015[s]	0,027[s]

TABLE III

EVALUATION OF THE PARALLEL ALGORITHM—TRANSPORT DELAY FOR  $m = 255$

Sequential algorithm	Parallel algorithm
1,2[s]	0,17[s]

TABLE IV

EVALUATION OF THE PARALLEL ALGORITHM—THROUGHPUT FOR  $m = 255$

Sequential algorithm	Parallel algorithm
0,6[GFlop/s]	18,79[GFlop/s]

The algorithms were tested on a PC computer with Genuine Intel CPU T2300 @ 1,66 GHz, and 1536 MB RAM NVIDIA GTX 460 graphic card. Figure 2 shows the speedup of a parallelized algorithm N4SID and in Figure 3 MOESP

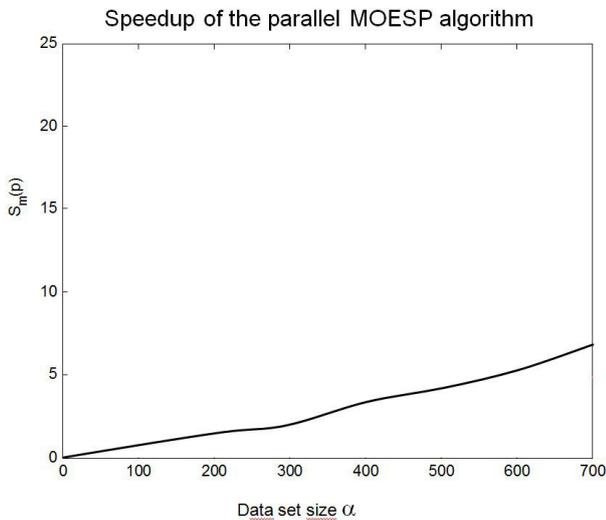


Fig. 2. Speedup of the parallel MOESP algorithm

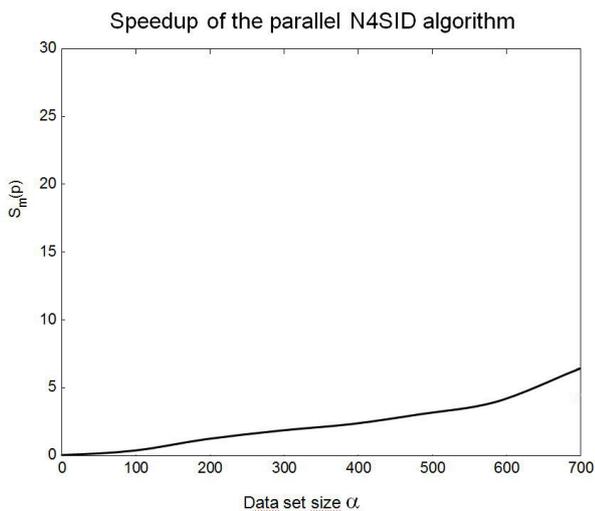


Fig. 3. Speedup of the parallel N4SID algorithm

parallelized algorithm speedup as a function of the number of measurements of  $\alpha$ . The pass length was changed from 5 to 700 and the run time for the sequential modified algorithm N4SID and MOESP and their parallel versions was measured. Based on these measurements, the speed up was evaluated. Speedups, shown in Fig. 2 and 3, are equal to formally calcu-

lated speedups of parallel algorithms. Parallel implementations run about 5-6 times faster than their sequential versions.

## VI. CONCLUSIONS

For Stages 1 and 3, the following indicators: speedup  $S_m$ , throughput  $P$  and computational efficiency  $E_m$  increase. At the same time, the transport delay  $L$  decreases. For Module 2, speedup, throughput  $P$ , and efficiency decrease while the delay transport increases. This comes from the calculation of the system order using SVD decomposition. Speedups, shown in Fig. 2 and 3, are equal to formally calculated speedups of parallel algorithms. Parallel implementations run about 5-6 times faster than their sequential versions. Transport delay is interpreted as the time of calculation. The speedup and throughput efficiency are good candidates for indices to evaluate the degree of parallelization. A parallel algorithm may be useful in selecting fast and using less resources algorithm for identification of stationary linear repetitive processes.

## ACKNOWLEDGMENT

The author is a scholar participating in Sub-measure 8.2.2. Regional Innovation Strategies, Measure 8.2. Transfer of Knowledge, Priority VIII Regional Human Resources for the Economy Human Capital Operational Programme co-financed by the European Union Social Fund and the State Budget of Poland.

## REFERENCES

- [1] R. Hartenstein, "Are we really ready for the breakthrough? ", In Parallel and Distributed Processing Symposium, page 7, April 2003
- [2] Z. Baker, M. Gokhale, and J. Tripp, "Matched filter computation on fpga, cell and gpu", In 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pages 207-218, USA, April 2007
- [3] A. Janczak, D. Kujawa, E. Rogers, Z. Cai, "Subspace identification of process dynamics for iterative learning control", Proceedings of the 8th International Workshop on Multidimensional Systems, Erlangen, Niemcy, 2013
- [4] Tomasz Zawadzki, Dominik Kujawa, "The hybrid algorithm for procedural generation of virtual scene components", Lecture Notes in Computer Science: 8th International Symposium, 2012
- [5] A. Chiuso and G. Picci, "Some algorithmic aspects of subspace identification with inputs", In Int. J. Appl. Math. Comp. Sci., 2001, Vol. 11, No. 1, 55-75, 2001
- [6] E. Rogers, K. Galkowski and D. H. Owens "Control Systems Theory and Applications for Linear Repetitive Processes", Springer, 2007
- [7] T. Katayama, "Subspace methods for system identification", Springer, 2008
- [8] P. Van Overschee and B. De Moor, "Subspace Identification for Linear Systems", Kluwer Academic Publishers, 2001
- [9] D. Lyons and I. Giselle, "Evaluation of a Parallel Architecture and Algorithm for Mapping and Localization", In Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pages 546-551, June, 2007
- [10] J. Stój, "Temporal aspects of redundancy in distributed real-time computer systems (original title: Wpływ redundancji na zależności czasowe w rozproszonych informatycznych systemach czasu rzeczywistego)", PHD Thesis, Silesian University of Technology, Gliwice, Poland, 2009