

Prediction of the Partial Reconfiguration Matrix Inverse Computations

Etienne Aubin Mbe Mbock *

Faculty of Mathematics and Computer Sciences
 Departement of Computer Engineering
 Ruprecht-Karls-Universität Heidelberg
 Heidelberg, Deutschland
 E.Mbe_Mbock@stud.uni-heidelberg.de

Abstract—Application of reconfigurable systems is very important in computer technology and computational sciences. The theoretical advantage of this concept is that we can reconstruct computations and create new hardware based on these computations. The discovery of the matrix inverse computations with help the provided concept denotes its importance in computer science. The problem that we need to solve is how to predict the two inverse matrices that have been generated via reconfiguration. This article will provide this approach and improve this method for the computation of any two $n \times n$ matrix inverse.

Index Terms—Partial, Reconfiguration, Linear, Matrix Recursion, Hardware, Control, Computations

I. Introduction

RECONFIGURATION of algorithms in computations is nowadays unavoidable in basic science fields. This concept is especially used in computer technology [1], [2], [8], [9] and is now being deployed in computational sciences. Classically this subject addresses Xilinx technologies and FPGAs related topics [2], [3], [10]. This concept has grown up with applications in control theory with the porting of the Kalman Filter algorithm on FPGA. Most recently many goals have been achieved. The recursive dynamic process creation with optimization goals. If partial reconfiguration technologies allow designers to change functionalities of their hardware devices, then this feature can be adapted in algorithm design and computing. The consequence in computer science will be algorithm optimization. The benefits of this concept are few, including among others performance development, hardware complexity reduction from technological point of view. For this research article, we suppose the existence of partial reconfiguration of algorithms [5], [6]. This reconfiguration is guaranteed by theorems on dynamic partial reconfiguration of algorithms. We admit that the general recursive linear process, specified by:

$$\left\{ q_1, \quad q_j = \sum_{i=1}^{j-1} \alpha_{ji} q_i, \quad j \in \{2, 3, \dots, N\} \right\}$$

*Ruprecht-Karls Universität Heidelberg, PHD-Candidate and Young Researcher at the Faculty of Mathematics and Computer Sciences. Teaching Assistant(Business Informatics University of Mannheim). Mailing Address:Eichendorffstrasse 35, 78120 Furtwangen, Germany.

is given. We admit the following inverse matrix hardware construction provided by [5]. The hardware solves the matrix

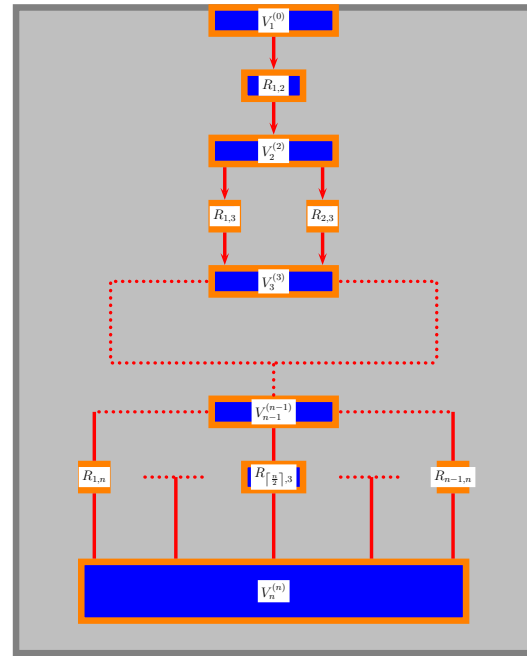


Fig. 1
Hardware Construction

inversion construction problem. That is, given two matrices A and B represented by their respective column entries

$$(A_j, B_j, j \in \{1, 2, \dots, n\})$$

construct with the partial reconfiguration, two matrices that are inverse to each other. The hardware will construct all $n \times n$ two matrices that are inverse to each other. According to this hardware upper or lower triangular matrices will be constructed. This research article assumes some results on re-configuration of algorithms, matrix analysis and computation basic results [13]–[18]. The research article will suppose the existence of the n -dimensional vector space and will extend

the hardware proposed by Mbock see [6]. The objective of our investigations is to construct all two inverse matrices. The inversion matrices will have the following expansion

$R_{1,1}$	$R_{1,2}$	\cdots	$R_{1,k}$	\cdots	$R_{1,n}$
$R_{2,1}$	$R_{2,2}$	\cdots	\cdots	\cdots	$R_{2,n}$
\vdots	\vdots	\ddots	\ddots	\ddots	\vdots
\vdots	\vdots	\ddots	$R_{k-1,k}$	\cdots	$R_{k-1,n}$
\vdots	\vdots	\ddots	\ddots	\ddots	\vdots
$R_{n,1}$	$R_{n,2}$	\cdots	$R_{n,k}$	\cdots	$R_{n,n}$

$V_{1,1}$	$V_{1,2}$	\cdots	$V_{1,k}$	\cdots	$V_{1,n}$
$V_{2,1}$	$V_{2,2}$	\cdots	\cdots	\cdots	$V_{2,n}$
\vdots	\vdots	\ddots	\ddots	\ddots	\vdots
\vdots	\vdots	\ddots	$V_{k-1,k}$	\cdots	$V_{k-1,n}$
\vdots	\vdots	\ddots	\ddots	\ddots	\vdots
$V_{n,1}$	$V_{n,2}$	\cdots	$V_{n,k}$	\cdots	$V_{n,n}$

With our investigations on that task we propose a scheme that will control the two inverse matrix output that is predicting the two computed matrix inverse.

II. Reconfiguration of Algorithms Overview

The study of reconfiguration of algorithm led to the discovery of the linear recursive process. This process could be implemented in hardware in the special case of integer input values. The Kalman Filter algorithm can also be optimized with this reconfiguration concept. The recent research about algorithm reconfiguration solves the matrix inverse computations. The provided inverse matrices will be all upper or lower triangular matrices with the main diagonal entries of the matrices not all being zero valued. This linear recursive process will have an impact on all matrix based computations. The reconfiguration analysis applied to this process can give a new vision on orthogonal matrix creation and [Q R]-decomposition. The reconfiguration time of the process is $\Theta(n^2)$. The process can be easily extended with new matrix computation functionalities. Some achievements of the reconfiguration of algorithms are:

- 1) Linear Recursive Process Creation and Optimization
- 2) Matrix [Q,R]-Decomposition
- 3) Reconfiguration Speed Problem Resolution
- 4) Real Vectors Coding

The improvements that we listed are more related to algorithms and computational optimization, this concept of reconfiguration has also been applied in robotics [19], [20]. The idea of partial reconfiguration of algorithms aims at the algorithms functionalities extension and this in turn still alters the true hardware complexity of the algorithm. The previously cited advantages of partial reconfiguration are very

significant in computer sciences and the recent matrix inverse computations are now a point of interest. In this research article we want to provide a way of controlling the inverse matrices that have been computed.

III. Principles and Theorems

Theorem III.1. *The control of the matrix inverse computations is given by any Gauß method under the existence of two matrices that are upper and lower matrices and inverse to each other, applied to the following given equations.*

$$\begin{bmatrix} \sum_{l=1}^n v_{j_l}^{(k)} \cdot v_{j_l}^{(k)} \cdot v_{j_l} \\ \sum_{l=1}^n v_{j_l}^{(k)} \cdot v_{j_2}^{(k)} \cdot v_{j_l} \\ \vdots \\ \sum_{l=1}^n v_{j_l}^{(k)} \cdot v_{j_n}^{(k)} \cdot v_{j_l} \end{bmatrix} = \begin{bmatrix} \alpha v_{j_1}^{(k)} + \beta v_{j_1}^{(k-1)} \\ \alpha v_{j_2}^{(k)} + \beta v_{j_2}^{(k-1)} \\ \vdots \\ \alpha v_{j_n}^{(k)} + \beta v_{j_n}^{(k-1)} \end{bmatrix}$$

for all

$$j \in \{1, 2, \dots, n\}.$$

There exists at least two $n \times n$ matrix classes for which the knowledge of the $n \times n$ inverse matrices $R_{i,j}$ and V_j determine all other $p \times p$ matrices $R_{i,j}$ and V_j with $p < n$

Proof:

- 1) We assume that there are two matrices that are upper or lower matrices and inverse to each other.
- 2) Given such $n \times n$ matrices R and V . The constructed hardware allows the following equations for all

$$j \in \{1, 2, \dots, n\}$$

$$V_j^{(k)} + V_j^{(k-1)} = V_j^{(k)} A_j^k \cdot V_i^{(k)}$$

for all $j \in \{1, 2, \dots, n\}$, since the matrices R with entries $R_{i,j}$ and the input column matrix A are related to each other according to the following scheme

$$R_{k,j} = \begin{cases} V_j^{(k)} \cdot A_j & \text{if } k \neq j \\ \|V_j^k\| & \text{if } k = j \end{cases}$$

- 3) Apply any Gaußalgorithm and stop
- 4) Assuming that there are such $n \times n$ matrices, according to the provided hardware construction the hardware construction, the following matrices will be

provided:

$$R|V = \begin{bmatrix} R_{1,1}|V_{1,1} & R_{1,2}|V_{1,2} & R_{1,3}|V_{1,3} & \cdots & R_{1,k-1}|V_{1,k-1} & \cdots & R_{1,n}|V_{1,n} \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & R_{k-1,k-1}|V_{k-1,k-1} & \cdots & R_{k-1,n}|V_{k-1,n} \\ \vdots & & & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & R_{n,n}|V_{n,n} \end{bmatrix}$$

5) Suppose that the last row

$$\begin{bmatrix} R_{1,n}|V_{1,n} \\ R_{2,n}|V_{2,n} \\ \vdots \\ R_{k-1,n}|V_{k-1,n} \\ \vdots \\ R_{n,n}|V_{n,n} \end{bmatrix}$$

has been computed. Knowing that these computations are based on the linear recursive process, the following submatrix computations must be stored by the process

$$R|V = \begin{bmatrix} R_{1,1}|V_{1,1} & R_{1,2}|V_{1,2} & R_{1,3}|V_{1,3} & \cdots & R_{1,k-1}|V_{1,k-1} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & R_{k-1,k-1}|V_{k-1,k-1} \\ \vdots & & & & 0 \\ 0 & \cdots & & & 0 \end{bmatrix}$$

6) This completes this proof

■

A. Simulations of the Computations and Devices Construction

The simulation shown in figure 4 is made of four tables. The last two tables represent the matrix inverse R and V . The two inverse matrices satisfy the specifications of the algorithm, they will be upper triangular. The values of these matrices are given by just considering the diagonal entries. Let us consider the last table of figure 4 the first diagonal concentrates on 1, all other concentration numbers are zero. The second diagonal will concentrate on negative real numbers closed to -0.5 and a positive value closed to 1 as pictured in the last table of figure 4. These tables represent the matrix V and R whose exact entries are given in the following triangle

$$V = \begin{bmatrix} 1 & & & & & \\ -0.4472 & 0.8944 & & & & \\ -0.2868 & -0.0637 & 0.9559 & & & \\ -0.1940 & -0.0540 & -0.0699 & 0.9770 & & \\ \vdots & \cdots & \cdots & \vdots & \cdots & \ddots \\ \vdots & \cdots & \vdots & \vdots & \cdots & \ddots \\ -0.0815 & -0.0319 & \cdots & \cdots & -0.0459 & 0.9918 \end{bmatrix}$$

Similarly the numerical values of the matrix R are given below:

TABLE I: Matrix Control Entries

A_1	A_2	A_3	A_4	A_5	A_6	A_7
1	0.5	0.33	0.25	0.20	0.1667	0.1429
0.5	0.333	0.25	0.20	0.1667	0.1429	0.125
0.333	0.25	0.20	0.1667	0.1429	0.125	0.1111
0.25	0.20	0.1667	0.1429	0.125	0.1111	0.1
0.2	0.1667	0.1429	0.1250	0.1111	0.1	0.0909
0.1667	0.1429	0.1250	0.111	0.100	0.0909	0.0833
0.1429	0.1250	0.1111	0.1	0.0909	0.833	0.0769

$$R = \begin{bmatrix} 1 & & & & & & \\ 0.5 & 0.8944 & & & & & \\ 0.33 & 1.1180 & 0.9559 & & & & \\ 0.25 & 0.074450 & 0.0671 & 0.0596 & & & \\ \vdots & \cdots & \cdots & \vdots & \cdots & \ddots & \\ \vdots & \cdots & \vdots & \vdots & \cdots & \ddots & \\ 0.1429 & 0.0479 & \cdots & \cdots & \cdots & 0.0468 & 1.0083 \end{bmatrix}$$

The simulated values of the control matrix are given in the following table. The second table of figure 4 is used to initialize the vector $V_j^{(0)}$ for $j \in \{1, 2, \dots, n\}$

IV. Conclusion

This research article extends the partial reconfiguration and the matrix inverse computation method. The computations that we provide in this paper propose a method to control the computed matrix inverse. This means, given two inverse matrices computed per “algorithm reconfiguration”, what are the entries of the starting $n \times n$ matrices $A|B$ used by the reconfiguration algorithm. Some of the devices constructed with this research have been presented in figure 3. This article also analyzes the case in which the knowledge of the real entries of $n \times n$ $R|V$ inverse matrices determines the $n-1 \times n-1$ $R|V$ inverse matrices. For this special case of the computations, the proof has taken advantage of the recursive dynamic process. The concept of partial reconfiguration that basically addresses hardware systems is now adapting to algorithms, processes and computations. The approach in this research article is computational and matrix based [21]–[29], the concept will be advantageous in all algorithm fields and computational optimizations [2], [5], [6]. The control of the matrix inverse computations will have the following advantages in addition to its theoretical foundation:

- 1) New vision of Matrix Computations Analysis
- 2) Computational Complexity $\Theta(n^2)$
- 3) Coding Matrices

This article is based on the previous resulting research [5], the inverse computation control is now provided. The main task of our research remains the hardware construction of this algorithm and the inverse matrices completeness. The analysis developed in this paper will be significant for computer scientists and computer engineers. The results will particularly be applied in computer algorithms and numerical matrix based computations.

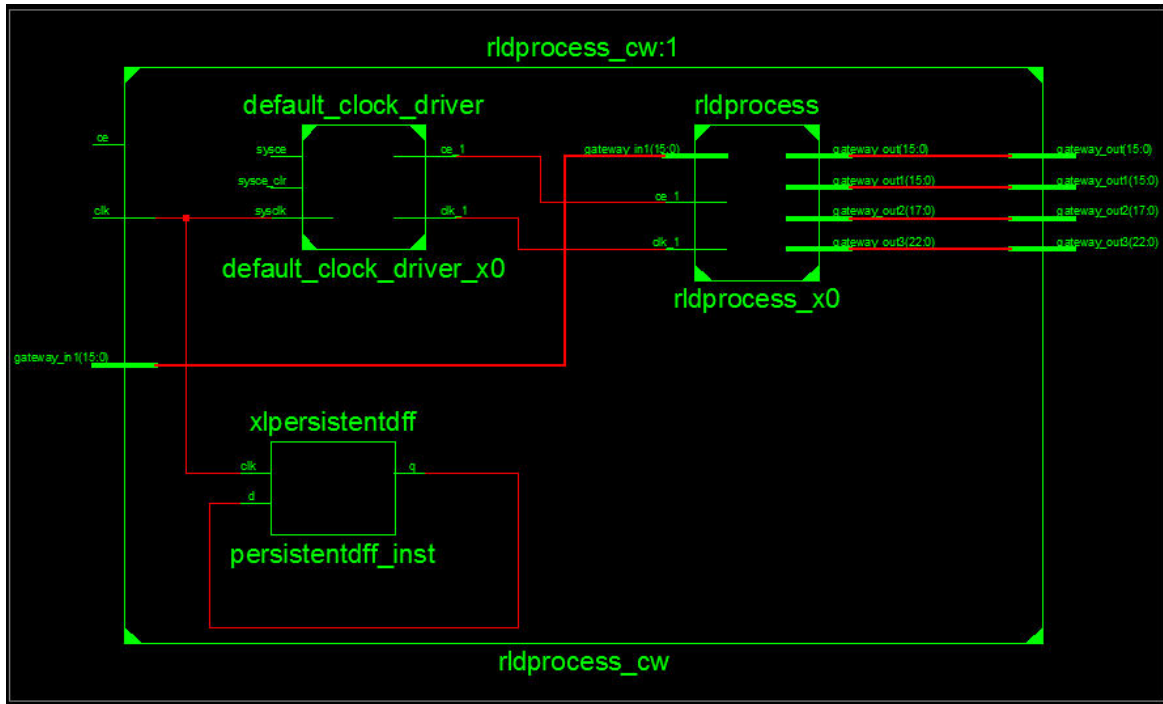


Fig. 2
Special Application

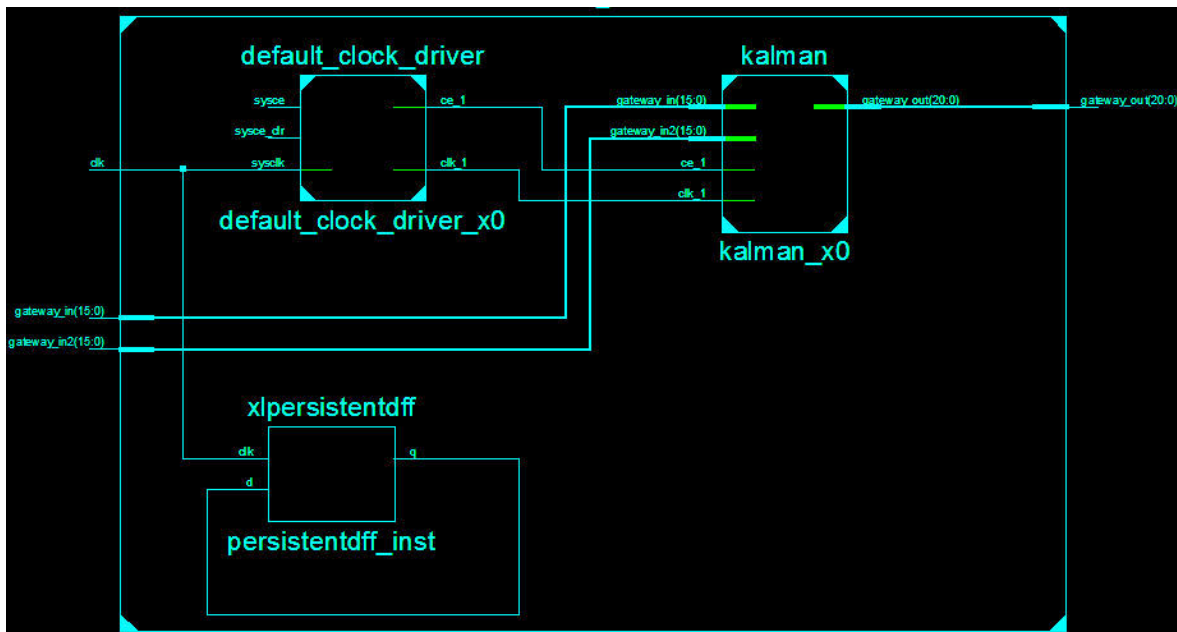


Fig. 3
Special Application

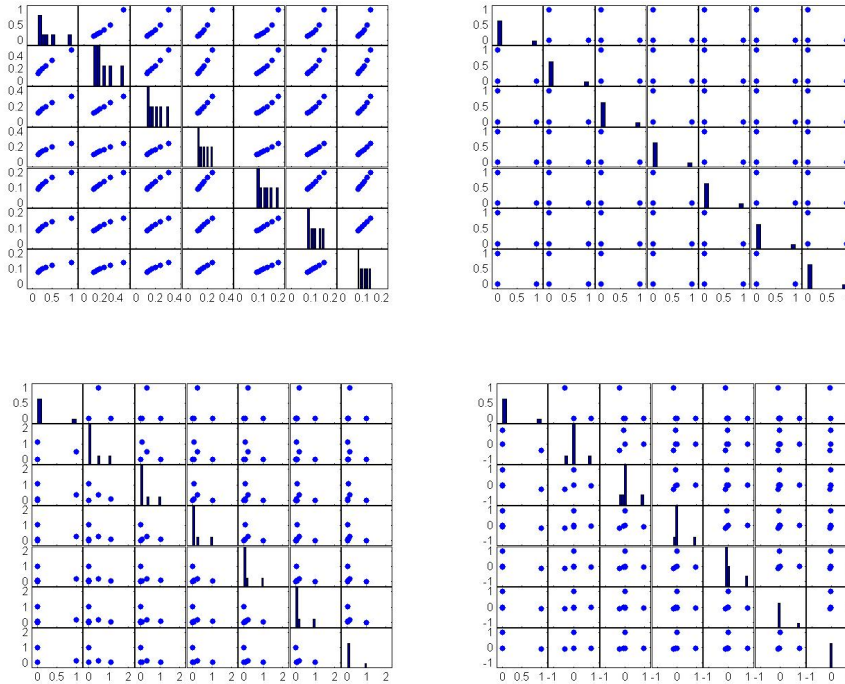


Fig. 4
Simulations for the Control Matrix

References

- [1] Xilinx, <http://www.xilinx.com/support/documentation/white-papers/wp374-partial-reconfig-xilinx-FPGAs.pdf>, 2012.
- [2] Osterloh, Michalik, Habinc, Fiethe, "Dynamic Partial Reconfiguration in Space Applications", *Conference Publication*, pp 336-343, 2009.
- [3] A. Donato, F. Ferrandi, M. D. Santambrogio, and D. Sciuto, "Operating System Support for Dynamically Reconfigurable Soc Architectures," *IEEE International Soc Conference*, pp 233-238, 2005.
- [4] E.A. Mbock, "fpga Implementation of the Kalman Filter": <http://www.dpg-verhandlungen.de/karlsruhe>, 2011.
- [5] E.A. Mbock, "Partial Reconfiguration and Novel Algorithm for Matrix Inverse Computations", http://www.actapress.com/Content_of_Proceeding.aspx?proceedingid=753 *Proc. of IAENG, San Francisco*, pp. 118-122, october 2013.
- [6] E.A. Mbock, "Dynamic Partial Reconfiguration based on the Kalman Filter Method", http://www.actapress.com/Content_of_Proceeding.aspx?proceedingid=753 *Proc. of Iasted on Control and Applications, Honolulu*, august, 2013.
- [7] C. Kao, "Benefits of Partial Reconfiguration", *Xcell Journal, fourth quarter*, pp 65-68, 2005.
- [8] Upegui, <http://islwww.epfl.ch/~upegui/docs/DPR.pdf>.
- [9] Xilinx, <http://www.xilinx.com/support/documentation/white-papers/374-partial-reconfig-xilinx-fpgas.pdf>, 2012.
- [10] S. Corbetta, F. Ferrandi, M. Morandi, M. Novati, M.D. Santambrogio, D. Sciuto "Two Novel Approaches to Online Partial Bitstream Relocation in a Dynamically Reconfigurable System", *Proc. of IEEE Computer Society Annual Symposium on vlsi*, pp. 457-458, may 2007.
- [11] E.A. Mbock, "Algorithms in Matlab, the Reality of Abstraction and the Power of Pseudocodes", *Optimus Verlag*, 2012. http://www.mathworks.de/support/books/index_by_categorytitle.html?category=6
- [12] J.W. Demmel, "Applied Numerical Linear Algebra", *Siam, Philadelphia, pa*, 1997.
- [13] G.H. Golub and Ch.F. Van Loan, "Matrix Computations", *second edition, Johns Hopkins University press, Baltimore, md*, 1989.
- [14] Forsythe, George E., M. A. Malcolm, and C. B. Moler, "Computer methods for Mathematical Computations", *Prentice Hall, Englewood Cliffs, nj*, 1977.
- [15] M.T. Heath, "Scientific Computing: an Introductory Survey", *Mcgraw-hill, Boston, ma*, 1997.
- [16] R.L. Burden, J.D. Faires, "Numerical analysis", *Brooks/Cole Publishing Company*, 1997-2001.
- [17] N.J. Higham, "Accuracy and Stability of Numerical Algorithms", *Siam, Philadelphia, pa*, 1996.
- [18] J.W. Demmel, "Applied Numerical Linear Algebra", *Siam, Philadelphia, pa*, 1997.
- [19] Meng, Y., "An agent-based mobile robot system using configurable soc technique", *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida*, 33683373, 2006.
- [20] Long, M., Gage, A., Murphy, R., Valavanis, K., "Application of the Distributed Field Robot Architecture to a Simulated Demining Task", *Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pp. 31933200, 2005.
- [21] G.H. Golub and Ch.F. Van Loan, "Matrix computations", *second edition, Johns Hopkins University Press, Baltimore, md*, 1989.
- [22] M.T. Heath, "Scientific Computing: an Introductory Survey", *Mcgraw-hill, Boston, ma*, 1997.
- [23] N.J. Higham, "Accuracy and Stability of Numerical Algorithms", *Siam, Philadelphia, pa*, 1996.
- [24] R.D. Skeel and J.B. Keiper, "Elementary numerical Computing with Mathematica", *Mcgraw-hill, New York, ny*, 1993.
- [25] B.D. Hahn, "Essential Matlab for Scientists and Engineers", *John Wiley Sons, New York, ny*, 1997.
- [26] D.R. Hill and D.E. Zitarella, "Linear Algebra Labs with Matlab", *second edition, Prentice Hall, Upper Saddle River, nj*, 1996.
- [27] R.E. Larson and B.H. Edwards, "Elementary Linear Algebra", *third edition, D.C. Heath and Company, Lexington, ma*, 1996.
- [28] S.J. Leon, "Linear Algebra with Applications", *fifth edition, Prentice Hall, Upper Saddle River, nj*, 1998.
- [29] G. Strang, "Linear Algebra and its Applications", *second edition, Academic Press, Orlando, fl*, 1980.