# SPEEDING UP PROBABILISTIC ALGORITHMS APPLIED TO MOBILE ROBOTICS USING FPGA BASED RECONFIGURABLE ARCHITECTURES

*Sérgio Messias Cruz, Janier Arias García, Carlos Humberto Llanos*

Mechanical Engineer Department
Mechatronics Graduate Program
University of Brasilia
70.910-900, Brasilia, D.F., Brazil
E-mail: {sergiocruz, janier, llanos}@unb.br

**ABSTRACT**

This work describes a masters project proposal that aims to improve the probabilistic algorithm performance that are suitable to mobile robotics area. One hopes a better performance by using reconfigurable architectures based on FPGAs in order to speed up the matrix operations, such as those involved on these kind of algorithms. It is a well-known fact that some algorithms related to autonomous locomotion of robots spend a large time for matrix operations such as multiplication and inversion. These operations allow the improvement of algorithms performance whenever the availability of their respective hardware implementations. In critical applications, such as mobile robotics, such processes must be executed in real time so that the robot will not collide with the obstacles. Then, how faster the process can be executed better it will be for the application.

## 1. INTRODUCTION

Nowadays, the use of FPGA (Field Programmable Gate Array) is obtaining a special attention of the scientific community for solving computational problems that involve hard algebraic calculations [1]. To prove this, in a simple search on Google site for the words FPGA and algorithms more than seven millions hits were found. In this context it is pertinent to focus on the speeding up of the algorithm execution by using an FPGA (as a hardware accelerator), in part or in whole of some probabilistic algorithms applied to mobile robotics, for example, Markov chains, Kalman filters, Monte Carlo based methods, SLAM algorithm, among others. Such probabilistic algorithms have been historically used in mobile robotics in order to became the localization, mapping and navigation tasks more robust with regard to the uncertainties and/or noise generated by the sensors and also the intrinsic uncertainties of the environment [2].

## 2. MATRIX CALCULATIONS

The matrix calculations can be seen as a combination of various linear algebraic operations with a hierarchy. For example, the scalar product (or dot product) are additions and multiplications, matrix/vector multiplications are series of dot products and

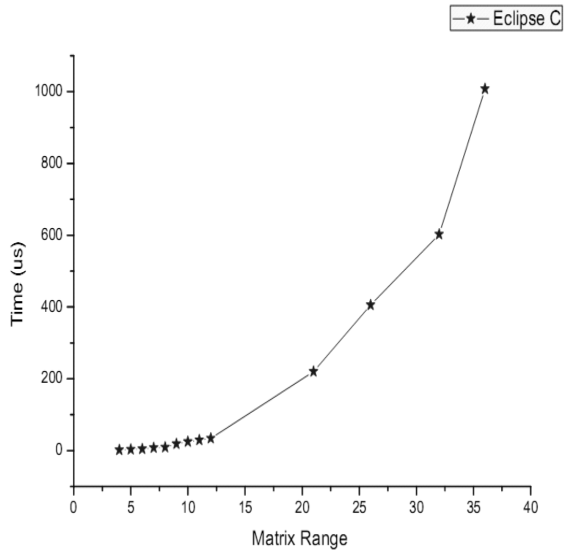


**Figure 1**. Time consumed by a computer to perform operations of matrix inversion wrote in C [1 modified]

matrix/ matrix multiplications are equivalent to a set of matrix/vector multiplication [1].

One of the most important and computationally expensive operations is the matrix inversion for which several well-known numerical approaches have been applied mostly in software platforms. Although the computational complexity of matrix inversion is an open question, popular sub cubic software solutions such as the $O(n^{2.807})$ Strassen and the $O(n^{2.376})$ Coppersmith-Winograd based approaches, applied to invert $n \times n$ matrices have a high theoretical interest, but the same are not used in practice because of the speed-up has only impact for very large matrices, which makes their hardware implementations unsuitable. Algorithmically simple methods such as Gauss-Jordan (GJ) elimination, although of higher complexity $O(n^3)$, are very important for developing practical architectural implementations. The hardware implementation of these algorithms is very interesting due to the fact that the von Neumann bottleneck can be avoided, mainly with regard to the writing/reading instructions to/from RAM memory. In

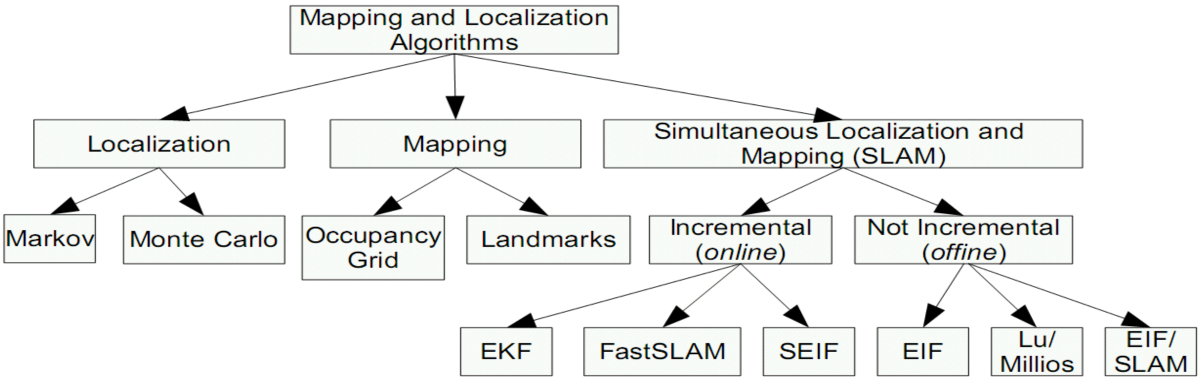


**Figure 2**. Classification of some algorithms applied to mobile robotics [2 modified]

the case of matrix operations that are implemented in hardware only write/read data operations need to be executed, without the complex decodification/execution steps related to the instruction executions, which are strongly related to the von Neumann model [1].

The computational operations developed in matrix calculations are fundamental, or rather, the heart of many scientific computing algorithms, for example those involved in signal processing, computer vision, robotics, among others. These types of algorithms that are developed using matrix calculations, such as multiplication or matrix inversion, end up being computationally expensive tasks, and their hardware implementations require great efforts and development time (see Figure 1). Therefore, there is a growing demand for architectures that allow matrix calculations, thus providing fast and efficient solutions to these problems [3].

## 3. ALGORITHMS APPLIED TO MOBILE ROBOTICS

For many years the development in robotics area has become an important area of study by researchers due to the fact the important challenges in programming and control, which are very complex, specially in mobile robotics. Additionally, the mobile robotics is a multidisciplinary research field in which knowledge of mechanisms, control theory, signal analysis, computer algorithms, artificial intelligence, probability theory etc. is required [4]. On the other hand, in a quick review it can be observed that several studies and projects are being developed in the area of mobile robotics.

In order to move a robot independently around a defined space, good location, mapping and navigation algorithms must be applied. The use of these algorithms is essential for allowing the robot to develop complex tasks. In this context, Figure 2 depicts a classification of some of the more important algorithms used in mobile robotics.

### 3.1. Localization

The localization is the capability of the robot to estimate its position in the environment. This is extremely important for a robot to execute an autonomous task. In simple terms, the environment map is known and the localization is estimated with the data provided by the sensors.

The most used methods for this task are based on Markov and Monte-Carlo techniques[2].

### 3.2. Mapping

Although the location (in the environment) can be previously known, in the mapping function the robot does not have a map of the environment for finding itself. The construction of this map is automatically accomplished. These maps can be classified into two types: (a) metric and (b) topological. Additionally, two well-known forms of metric representation are available: (a) the occupancy-grid and (b) landmarks [2].

One topic that has received special attention from the scientific community in robotics is the technique of simultaneous localization and mapping. This technique is known as SLAM (Simultaneous Localization And Mapping) or CML (Concurrent Mapping and Localization). The SLAM assumes that the robot does not have a known map of the environment and much less its position on it. The task of SLAM is exactly to build this map and also simultaneously estimating the position of the robot [4]. Table 1 shows a comparison between the algorithms of Kalman Filter (KF) and Extended Kalman Filter (EKF) and FastSLAM.

The purpose of this work focuses on the area of probabilistic computational algorithms applied to mobile robotics in what one of the function of the robot is to move independently around a area for making navigation more robust and reliable.

#### *3.2.1. Kalman Filter*

In statistics area, the Kalman filter is a mathematical method, which purpose is to use the observed measurements over time, containing noise (random variations) and other inaccuracies, and producing in such a way values that tend to be closer to the true values of the measurements and their associated

**Table 1.** Comparison between mapping and localization algorithms [2 modified]

| **Algorithm** | **Online Mapping** | **Information on the map** | **Require sensor location informed by the user** | **Computational Complexity (n= landmark and p = particle)** | **Noise type** |
|---|---|---|---|---|---|
| KF e EKF | Yes | Landmark | No | $\Phi(n^2)$ | Gaussian |
| FastSLAM | Yes | Landmark or Raw data | No | $\Phi(p \log n)$ | Gaussian for Landmark and any position for the sensor |

calculated values. The Kalman filter has many applications in technology and the respective extensions and generalizations of the method have also been developed, for instance Extended Kalman Filter (EKF).

The Kalman filter produces estimations of the true values of measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, as well as computing a weighted average of the predicted value and the measured value (See Figure 3). The larger weight is given to the value with the least uncertainty. The estimation produced by the method tends to be closer to the true values than the original measurements, because of the weighted average has a better estimated uncertainty than either of the values that carry into the weighted average [7].

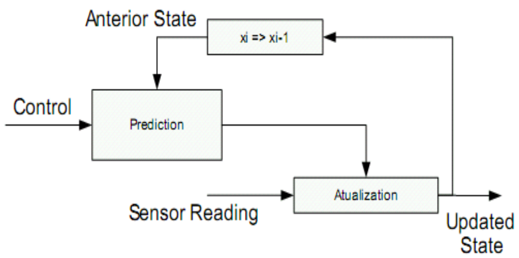


**Figure 3**. General Structure of KF

A difference between Kalman Filter and Extended Kalman Filter is that KF is used for linear dynamical systems as long as EKF is used for non-linear dynamical systems. Below, [7] shows a pseudo-code for Kalman Filter and it can be observed that there are many operations such as additions, subtractions, multiplications as well as matrix inversion.

Algorithm Kalman filter ($n_{t-1}, \Sigma_{t-1}, u_t, z_t$):
$\bar{n}_t = A_t\, n_{t-1} + B_t\, u_t$ (1)
$\bar{\Sigma}_t = A_t\ \Sigma_{t-1}\, A_t^T + R_t$ (2)
$K_t = \bar{\Sigma}_t\, C_t^T\, (C_t\ \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$ (3)
$n_t = \bar{n}_t + K_t\, (z_t - C_t\, \bar{n}_t)$ (4)
$\Sigma_t = (I - K_t\, C_t)\ \bar{\Sigma}_t$ (5)
return $n_t, \Sigma_t$

### 3.3. Navigation

The navigation problem is the motion process of the robot from its current position to an new position, which has been estimated safely and efficiently, being one of the fundamental problems in mobile robotics. The obstacle avoidance is the major navigation problems. This problem usually is solved making use of distance sensors in order to detect objects that may collide with the robot. After finding a possible obstacle, the robot should change its trajectory so that the collision can be avoided. Once the risk of collision disappears, the robot must return to its previous path [5]. In navigation issue, two algorithms are well-known: (a) potential fields and (b) VFH (Vector Field Histogram) [2].

## 4. RECONFIGURABLE ARCHITECTURES BASED ON FPGA

The FPGAs are programmable logic devices that consist basically of three types of components (see Figure 4): input and output blocks or cells (IOB), configurable logic blocks (CLB) and interconnect switches or switch matrix [6].

One of the major advantage of an FPGA is in the velocity to process a function. Instead of running a function in software, it can be ran in the hardware itself. This brings good results for some specific applications.

In a quick literature review, it can observed some works using FPGAs applied to mobile robotics: [1] implemented in a Xilinx Virtex-5 FPGA a floating-point matrix inversion. This can be useful to speed up probabilistic algorithms applied to mobile robotics; [2] presents an architecture implemented in hardware to solve SLAM problems. The system is completely embedded on an Altera Stratix II FPGA and its performance is at least one order of magnitude better than a high end PC-based implementation; [9] using a Xilinx Virtex-5 FPGA proposed a hardware implementation of the Particle Swarm Optimization algorithm with passive congregation (HPPSOpc), which was developed using several floating-point arithmetic libraries. This can be useful for collaborative robotics; [11] presents a pipelined hardware architecture for image processing using a low cost omnidirectional vision system, which was calibrated using a three-order polynomial interpolation. An Altera Cyclone II FPGA device was used for implementing the hardware architectures.

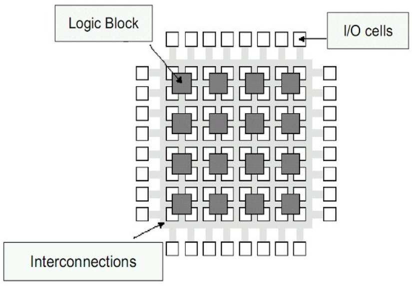


**Figure 4**. General structure of an FPGA [6 modified]

The two major manufacturers of FPGAs are Xilinx and Altera, and together they dominate 80% of the market. In this context, the Laboratory of parallel architectures (LAPAR), from the Department of Mechanical Engineering of University of Brasilia works with FPGAs manufactured by these companies. Development kits are available for students in order to develop their research projects as well as a research mobile robot from the Adept Mobile Robotics (a Pioneer P3-AT as shown in Figure 5) [8]. Works such as [1], [9], [10] and [11] were all developed at LAPAR using FPGAs. These works will be usefull to the development of this masters project proposal.

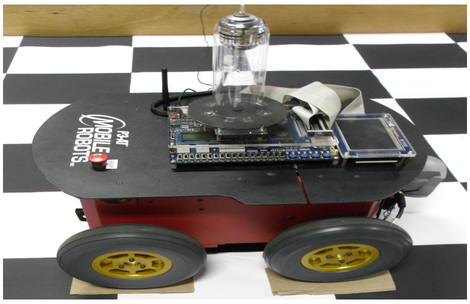

**Figure 5**. Altera FPGA development kit along with a Pioneer P3-AT robot used in LAPAR

## 5. CONCLUSION

The current stage of this work is still in the literature review, as well as the choice of the FPGA to be used for this project. As mentioned earlier, researchers at the LAPAR have been performed works with FPGAs of both companies, Xilinx and Altera. An attempt to migrate the library implemented by [1] is being performed (Xilinx to Altera), because the Altera´s software is more friendly to the user and has tools that speed up the process of development.

It is intended through this work to achieve improvements in the execution time and power consumption of such operations. In the context of stochastic based algorithms it is very important to guarantee that the arithmetic operations will be executed using floating-point representation, as discussed in [9] and [11]. It is also intended to compare the results of this study with the results obtained in other processing platforms, for example PC based ones as well as embedded microcontroller based platforms.

## 6. ACKNOWLEDGMENTS

The authors would like to thank Fundação de Apoio à Pesquisa do Distrito Federal - FAPDF for its financial support for the presentation of this work.

## 7. REFERENCES

[1] J. A. Garcia, R.P. Jacobi, C.A. Llanos, and M.A. Rincón, “A Suitable FPGA Implementation of Floating-Point Matrix Inversion Based on Gauss-Jordan Elimination”, In *IEEE Programmable Logic (SPL), 2011 VII Southern Conference on*, Cordoba, Argentina, p. 263-268, April, 2011.

[2] V. Bonato. (2007). Proposta de uma arquitetura de hardware em FPGA implementada para SLAM com multi-câmeras aplicada à robótica móvel. PhD Thesis in Computer Science and Computational Mathmatics. Institute of Computer Science and Computational Mathmatics, University of São Paulo, São Carlos, SP.

[3] J. A. Garcia. (2010). Implementação em FPGA de uma biblioteca parametrizável para inversão de matrizes baseada no algoritmo Gauss-Jordan, usando representação em ponto flutuante. Dissertation in Mechatronic Systems. Department of Mechanical Engineer, University of Brasilia, Brasilia, DF.

[4] Siegwart, R., Nourbakhsh I. R., *Introduction to Autonomous Mobile Robots*, USA: MIT Press, 2004.

[5] H. L. Polidoro, and D. F. Wolf, “Navegação e Monitoramento de Ambientes Internos Utilizando Robôs Móveis”, Institute of Computer Science and Computational Mathmatics, University of São Paulo, São Carlos, SP.

[6] M. L. L. Vallejo, and J. L. A. Rodrigo, “FPGA: Nociones básicas y Implementación”, Department of Electronic Engineer, Polytechnic University of Madrid, 2004.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[8] Adept MobileRobots, “Research Robots”, 2011. Available: *http://www.mobilerobots.com/ResearchRobots/ResearchRobots/PioneerP3DX.aspx*

[9] D. M. Muñoz, C. H. Llanos, L. S. Coelho, and M. A. Rincón, “Hardware Particle SWARM Optmization with Passive Congregation for Embedded Applications”, In *IEEE Programmable Logic (SPL), 2011 VII Southern Conference on*, Cordoba, Argentina, p. 173-178, April, 2011.

[10] J. Y. Mori, C. S. Ferreira, D. M. Muñoz, C. H. Llanos, and P. Berger, “An Unified Approach Convolution-Based Image Filtering on Reconfigurable Systems”, In *IEEE Programmable Logic (SPL), 2011 VII Southern Conference on*, Cordoba, Argentina, p. 63-68, April, 2011.

[11] J. Y. Mori, D. M. Muñoz, J. A. Garcia, and C. H. Llanos, “FPGA-Based image Processing for Omnidirectional Vision on Mobile Robots”, accepted to SBCCI 2011, Brasil.