

Hardware and Software Co-design for the EKF Applied to the Mobile Robotics Localization Problem

Luis Contreras, Sérgio Cruz, J. M. S. T. Motta, and Carlos H. Llanos

Abstract—This paper describes a Hardware/Software Co-design approach for the Extended Kalman Filter (EKF) applied to the localization problem in mobile robotics. The EKF algorithm has been implemented and run on an Altera Cyclone IV FPGA with a Nios II embedded processor jointly with specific hardware modules, being adapted and applied to the mobile platform Pioneer 3AT (P3AT). In order to achieve this, we developed both the model of the mobile robot and its measurement systems previously to obtain the respective EKF equations. In the *prediction step* of the EKF algorithm, a system model based on concepts of dead-reckoning has been used and its implementation was achieved in software, using the Nios II processor. Conversely, in the *estimation step* of the EKF algorithm the respective equations have been implemented directly in hardware, producing an overall balanced implementation.

Index Terms—Dead-reckoning, extended Kalman filter, FPGA, hardware/software co-design, Pioneer 3AT.

I. INTRODUCTION

There are several works describing the implementation of probabilistic algorithm in FPGAs such as described in [1]-[3]. Among them, the Kalman Filter (KF) is one of the most useful algorithms for estimating the state of a dynamic system [3]. A special case of KF is the Extended Kalman Filter, which application is oriented to nonlinear systems [4].

This work presents a *hardware/software co-design* approach in FPGA of the Extended Kalman Filter Algorithm aiming at solving the Localization Problem in Mobile Robotics. The algorithm implementation was achieved considering the following approach: 1) a *software implementation* in the Nios II processor of the *prediction stage* (namely, 1st and 2nd EKF algorithm equations) and 2) a *hardware implementation* in Altera Cyclone IV FPGA of the *estimation stage* (namely 3rd, 4th and 5th EKF algorithm equations). For this case, it has been considered that the mobile robot is located within an environment with a known map and it moves using data processing *on line*. In this case, the mobile platform used in this project was the Pioneer 3-AT [5].

In this context, the contributions of this paper include: 1) the implementation of a sequential EKF algorithm in FPGA based on an approach of hardware/software co-design for the stages of *prediction* and *estimation*, using floating-point

representation, and 2) the validation of the results in terms of hardware resource consumption, performance and functionality in a real environment using the robot platform.

II. THE EKF ALGORITHM

The KF is based on a recursive method to estimate the state variable (x_k) of a linear dynamic system with the process noise (w_k) and measurement noise (v_k) [6]. An extension of the Kalman Filter is the EKF algorithm (Extended Kalman Filter), which is used for nonlinear dynamical systems. The EKF tries to linearize the system around the current state. For this case, we will consider a nonlinear system represented by:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}), \quad z_k = h(x_k) + v_k, \quad (1)$$

where $f(\cdot)$ is the nonlinear function of the process system, $h(\cdot)$ is the nonlinear function of the measurement system, x_{k-1} is the state vector, u_{k-1} is the input (or control) vector and z_k the output vector.

For discrete time, the *EKF prediction stage* will be defined by (2) and (3). The *EKF estimation stage* is defined by (4), (5) and (6). A detailed description about the symbols and matrix dimensions used in this paper is presented in Table I.

TABLE I: SYMBOL DESCRIPTION AND MATRIX DIMENSIONS FOR EKF

Symbols	Dimension	Description
X	3×1	State vector – Robot position
u	2×1	Input vector – provided by the encoders
e	2×1	Input error vector
P	3×3	Robot position covariance
A	3×3	Robot motion Jacobian
W	3×2	Robot motion noise Jacobian
Q	2×2	Covariance of permanent process noise
G	3×2	KF Gain
H	2×3	Measurement Jacobian
R	2×2	Covariance of permanent measurement noise
Z	2×1	Sensor measurement – provided by LRF
$f(\cdot)$	3×1	Process Nonlinear Function
$h(\cdot)$	2×1	Measurement Nonlinear Function

Prediction:

$$X_k^- = f(X_{k-1}^+, u_{k-1}, e_{k-1}) \quad (2)$$

$$P_k^- = A \cdot P_{k-1}^+ \cdot A^T + W \cdot Q_{k-1}^+ \cdot W^T \quad (3)$$

Estimation:

$$G_k = P_k^- \cdot H^T (H \cdot P_k^- \cdot H^T + R)^{-1} \quad (4)$$

$$X_k^+ = X_k^- + G_k (Z_k - h(X_k^-)) \quad (5)$$

$$P_k^+ = P_k^- - G_k \cdot H P_k^- \quad (6)$$

Manuscript received August 26, 2014; revised November 10, 2014. This work is supported by the Brazilian institutions CAPES, CNPq and PPMEC.

L. Contreras, S. Cruz, C. H. Llanos, and J. M. S. T. Motta are with the Graduate Program of Mechatronics Systems, Department of Mechanical Engineering, University of Brasilia – UnB, Brasilia, Brazil (e-mail: lcontreras@aluno.unb.br, sergio@unb.br, jmmotta@unb.br, and llanos@unb.br).

In some variables, the superscript ‘-’ means predicted value and the superscript ‘+’ means estimated value. The actual and previous time is denoted by the symbols K and $K-1$ respectively. Moreover, A is the *Jacobian matrix* of $f(\cdot)$ in respect to X^+ , W is the *Jacobian matrix* of $f(\cdot)$ in respect to the error vector e and H is the *Jacobian matrix* of $h(\cdot)$ in respect to X^+ .

From now on, the development of all EKF equations adapted to the system can be carried out considering that the mobile robot is a 4-wheel skid-steering one (SSMR). The steering of an SSMR is based on the differentially driving wheel pairs on each side (relative velocities of left and right side) of the mobile robot [7].

A. Prediction Stage

Since we are interested in the position x , y and the angular orientation θ of the robot, these variables are set as the state variables, namely: $X = [x \ y \ \theta]^T$.

Based on the SSMR models proposed in [7], [8] and by using the Euler integration method one can obtain the discrete-time system model for (2), i.e. the 1st EKF equation, which can be written as (7).

1st EKF Equation:

$$X_{K+1}^- = f(X_K^+, u_K, e_K),$$

$$\begin{bmatrix} x_{K+1}^- \\ y_{K+1}^- \\ \theta_{K+1}^- \end{bmatrix}_{3 \times 1} = \begin{bmatrix} x_K^+ + \Delta T \cdot \left(\frac{V_L + V_R}{2} \right) \cdot \cos \theta_K^+ \\ y_K^+ + \Delta T \cdot \left(\frac{V_L + V_R}{2} \right) \cdot \sin \theta_K^+ \\ \theta_K^+ + \Delta T \cdot \left(\frac{V_R - V_L}{D} \right) \end{bmatrix}_{3 \times 1}, \quad (7)$$

where ΔT is the sample time and $u = [V_L, V_R]^T$ is the input vector (provided by the encoders). The parameter D is the effective spacing between the wheels. The error associated to the linear tread speeds of V_L and V_R can be defined such as described in (8).

$$V_L = V_{Lc} + V_{Le}, \quad V_R = V_{Rc} + V_{Re},$$

$$e = [V_{Le}, V_{Re}]^T, \quad (8)$$

where V_{Lc} and V_{Rc} are the correct linear speeds of the left and right wheels respectively and the error vector e represents the uncertainties of the odometric model, assuming that it is a Gaussian noise with zero mean.

To construct a model equation from (3), i.e. the 2nd EKF equation, it is firstly described the *Jacobian matrices* A and W , whose respective representations are described by (9) and (10).

$$A_{K+1} = \frac{\partial f}{\partial X_K^+} = \begin{bmatrix} 1 & 0 & -\Delta T \cdot \left(\alpha \frac{V_L + V_R}{2} \right) \cdot \sin \theta_K^+ \\ 0 & 1 & \Delta T \cdot \left(\alpha \frac{V_L + V_R}{2} \right) \cdot \cos \theta_K^+ \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3} \quad (9)$$

$$W_{K+1} = \frac{\partial f}{\partial e_K} = \begin{bmatrix} \left(\alpha \frac{\Delta T}{2} \right) \cdot \cos \theta_K^+ & \left(\alpha \frac{\Delta T}{2} \right) \cdot \cos \theta_K^+ \\ \left(\alpha \frac{\Delta T}{2} \right) \cdot \sin \theta_K^+ & \left(\alpha \frac{\Delta T}{2} \right) \cdot \sin \theta_K^+ \\ -\frac{\Delta T}{D} & \frac{\Delta T}{D} \end{bmatrix}_{3 \times 2} \quad (10)$$

Then, the 2nd EKF equation can be described as in (11):

2nd EKF Equation:

$$P_{K+1}^- = A_{K+1} \cdot P_K^+ \cdot A_{K+1}^T + W_{K+1} \cdot Q_K^+ \cdot W_{K+1}^T,$$

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_\theta^2 \end{bmatrix}_{3 \times 3}, \quad Q = \begin{bmatrix} \sigma_{V_L}^2 & 0 \\ 0 & \sigma_{V_R}^2 \end{bmatrix}_{2 \times 2} \quad (11)$$

where P is the error covariance matrix of the robot pose and Q represents the covariance matrix of the permanent noise process.

B. Estimation Stage

To design the Estimation EKF Stage, it is assumed that the mobile robot is located within a known environment, which has a particular shape (e.g., a wall) showed in the Fig. 1. The symbol representation is shown in the Table II. Then, two arrays can be defined as in (12).

$$Z = \begin{bmatrix} u_i \\ v_i \end{bmatrix}_{2 \times 1}, \quad R = \begin{bmatrix} \sigma_{u_i}^2 & 0 \\ 0 & \sigma_{v_i}^2 \end{bmatrix}_{2 \times 2} \quad (12)$$

where Z is the measurement vector, whose parameters u_i and v_i are used as inputs at this stage to update the mobile state. Besides, R represents the measurement covariance matrix of the parameters (u_i, v_i). It is important to mention that R has two states: (a) when the robot is standstill the entries of the R matrix are minor compared to the entries of the Q matrix, and (b) when the robot is in motion the relationship is reversed, since the accuracy of the LRF sensor is reduced when the robot is moving.

During each measurement, the O -object global position (coordinates l in X and m in Y) is input to the EKF algorithm. By using analytic geometry the parameters (l, m) can be found and described such as in (13).

$$l_K = \frac{a - (x_{si} \sin \theta_K + y_{si} \cos \theta_K) + (x_{si} \cos \theta_K - y_{si} \sin \theta_K) \cdot \tan(\theta_K + \beta_i)}{\tan(\theta_K + \beta_i) + 1},$$

$$m_K = a - l_K, \quad (13)$$

where a is related to the point $(a, 0)$ in the Global Coordinate System (GCS), corresponding to the intersection of the environment line with the X_g -axis.

The other array necessary for this stage is $h(\cdot)$, and it can be obtained from a coordinate transform of the feature O from GCS to the Sensor S_i Coordinate System (SiCS), yielding (14).

Measurement model:

$$h(X_K^-) = \left\{ \begin{bmatrix} \cos \theta_K^- & \sin \theta_K^- \\ -\sin \theta_K^- & \cos \theta_K^- \end{bmatrix} \cdot \left[\begin{bmatrix} l_K \\ m_K \end{bmatrix} - \begin{bmatrix} x_K^- \\ y_K^- \end{bmatrix} \right] \right\} - \begin{bmatrix} x_{si} \\ y_{si} \end{bmatrix}$$

or

$$h(X_K^-) = \begin{bmatrix} (l_K - x_K^-) \cos \theta_K^- + (m_K - y_K^-) \sin \theta_K^- - x_{si} \\ (m_K - y_K^-) \cos \theta_K^- - (l_K - x_K^-) \sin \theta_K^- - y_{si} \end{bmatrix}_{2 \times 1} \quad (14)$$

From (14), the *Jacobian matrix* H can be calculated as in (15).

$$H = \begin{bmatrix} -\cos \theta_K^- & -\sin \theta_K^- & -(l_K - x_K^-) \sin \theta_K^- + (m_K - y_K^-) \cos \theta_K^- \\ \sin \theta_K^- & -\cos \theta_K^- & -(l_K - x_K^-) \cos \theta_K^- - (m_K - y_K^-) \sin \theta_K^- \end{bmatrix}_{2 \times 3} \quad (15)$$

TABLE II: DESCRIPTION OF THE ROBOT POSITION SYMBOLS

Symbols	Description
(X_g, Y_g)	Global Coordinate System – GCS
(x, y)	Robot Position into GCS
θ	Robot Orientation into GCS
(X_s, Y_s)	Robot Coordinate System – RCS
(X_{si}, Y_{si})	Sensor i Position into RCS
(U, V)	Sensor S_i Coordinate System – SiCS
γ_i	Distance between O-object (detected) and sensor S_i
β_i	Angle of measurement
(u_i, v_i)	O-object Position into SiCS
(l, m)	O-object Position into GCS

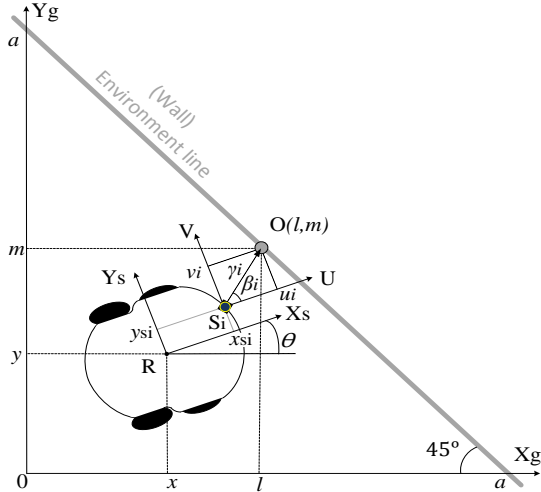


Fig. 1. Mobile robot positioned on a specific environment.

So, all the matrices needed to implement (4), (5) and (6) of the EKF *estimation stage* have been described from (7) to (15)

III. FPGA IMPLEMENTATION

This section presents the integration of the two EKF Stages through a *Hardware/Software Co-design* in the FPGA. The algorithm implementation was accomplished considering the following approach: 1) a *software implementation* in the Nios II processor (written in *C language*) for the EKF Prediction Stage and 2) a *hardware implementation* [2] in the Altera Cyclone IV FPGA (written in *VHDL code*) for the EKF Estimation Stage. Fig. 2 shows the general diagram of the project architecture that has been developed.

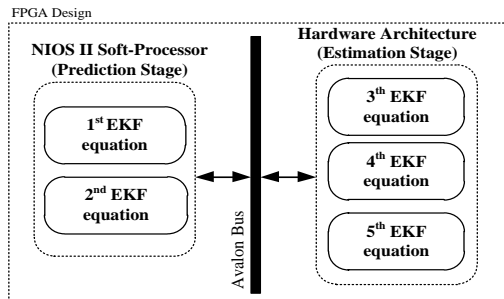


Fig. 2. Diagram of the general architecture.

A. Software Implementation

The EKF *Prediction Stage* requires matrices operations such as sum and multiplication. Algorithms (1) and (2) are shown with their pseudo-codes used at this stage of the project.

Algorithm 1: Code for algorithm of matrix multiplication

Matrix $C = \text{Matrix A} \times \text{Matrix B}$

```

1: for (i = 0; i < #RowA; i++) do
2:   for (j = 0; j < #ColumnB; j++) do
3:     for (k = 0; k < #ColumnA; k++) do
4:       C[i][j] = C[i][j] + A[i][k] * B[k][j];
5:     end for
6:   end for
7: end for
    
```

Algorithm 2: Code for algorithm of matrix addition

Matrix $C = \text{Matrix A} + \text{Matrix B}$

```

1: for (i = 0; i < #RowA; i++) do
2:   for (j = 0; j < #ColumnB; j++) do
3:     C[i][j] = A[i][j] + B[i][j];
4:   end for
5: end for
    
```

where #Row is the number of matrix rows and #Column is the number of matrix columns.

B. Hardware Implementation

At the EKF *estimation stage*, matrices operations such as multiplication, addition and inversion are used. To implement those operations, a Finite State Machine (FSM) [2] was designed based on floating-point arithmetic operators [9]. The FSM structure is shown in Fig. 3, composed by six states, namely: *waiting*, *mult1*, *mult2*, *multadd*, *add* and *div1*. Fig. 4 shows the data path [2] for implementing the *estimation stage* architecture. The EMB unit is composed of the arrays elements P_K^- , H , R , Z , $h(\cdot)$ and X_K^- ; the XMB unit is composed of the arrays elements G , X_K^+ and P_K^+ , and the IMB unit is composed of temporary arrays elements O .

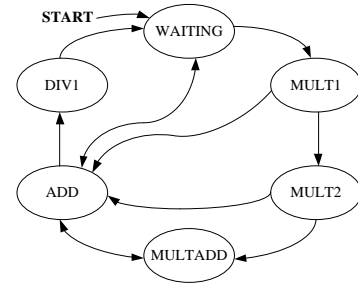


Fig. 3. FSM used in the architecture to perform the EKF estimation stage.

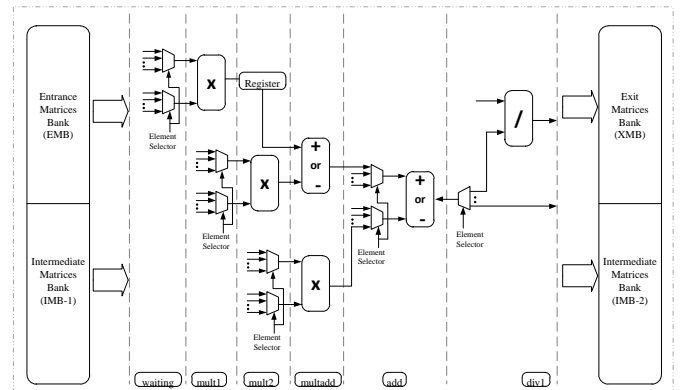


Fig. 4. Data path used for implementing the EKF estimation stage.

In the implementation of this EKF stage was necessary to divide in eleven steps as shown in (16) to (26).

$$1th: O_{3 \times 2}^1 = P_K^- \cdot H^T \quad (16)$$

$$2th: O_{2 \times 2}^2 = H \cdot O_{3 \times 2}^1 \quad (17)$$

$$3th: O_{2 \times 2}^3 = O_{2 \times 2}^2 + R_K \quad (18)$$

$$4th: O_{2 \times 2}^4 = inv(O_{2 \times 2}^3) \quad (19)$$

$$5th: G_K = O_{3 \times 2}^1 \cdot O_{2 \times 2}^4 \quad (20)$$

$$6th: O_{2 \times 1}^5 = Z_K - h(X_K^-) \quad (21)$$

$$7th: O_{3 \times 1}^6 = G_K \cdot O_{2 \times 1}^5 \quad (22)$$

$$8th: X_K^+ = X_K^- + O_{3 \times 1}^6 \quad (23)$$

$$9th: O_{3 \times 3}^7 = G_K \cdot H \quad (24)$$

$$10th: O_{3 \times 3}^8 = O_{3 \times 3}^7 \cdot P_K^- \quad (25)$$

$$11th: P_K^+ = P_K^- - O_{3 \times 3}^8, \quad (26)$$

where the 3rd EKF equation (4) is divided in five steps such as in (16) to (20). So, (21) to (23) are the corresponding steps to compute the 4th EKF equation (5). Finally, the 5th EKF equation (6) is referred to (24), (25) and (26).

It can be seen that (16), which is the first step to compute the 3rd EKF equation, is a matrix multiplication ($P_K^- \cdot H^T$). The result of the multiplication is a new matrix O^1 with dimension 3×2 , whose first entry is described by (27).

$$O_{11}^1 = p_{11}^- \cdot H_{11}^T + p_{12}^- \cdot H_{21}^T + p_{13}^+ \cdot H_{31}^T \quad (27)$$

In the case of (27), the FSM was designed to perform the three multiplications and the two additions, producing a FSM structure composed by five states: *waiting*, *mult1*, *mult2*, *multadd* and *add*.

The calculation of the *Jacobian matrix* H and the nonlinear function $h(\cdot)$ is performed previously in the Nios II processor and then their results are addressed to the *hardware architecture*.

IV. RESULTS

After the implementation of the EKF algorithm in the Altera DE2-115 board [10] we proceeded to perform some tests using the Pioneer 3-AT mobile platform (see Fig. 5). One of tests consisted in that the robot had to follow the path shown in the Fig. 6.



Fig. 5. Pioneer 3-AT with FPGA and LRF sensor.

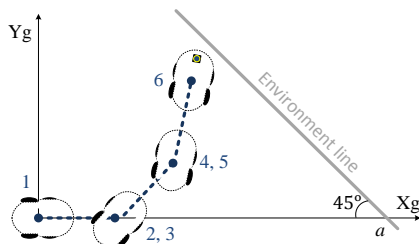


Fig. 6. Robot trajectory for the test.

For the test, the environment map was designed to have the parameter $a = 117\text{cm}$ and with initial values as $X_0^+ = [2\text{cm}, 0.1\text{cm}, 0^\circ]^T$. The test results are shown in Table III, where it is shown a comparison between the actual robot position (x, y, θ) and the estimated one, using the EKF algorithm developed in this paper. In Fig. 6, the instants 1 until 6 are shown along the robot path. As an example, the robot performs a straight motion between the instants 1-2, 3-4 and 5-6. In the instants 2-3 and 4-5 the robot turns $+47.5^\circ$ and $+61.7^\circ$ respectively, both angles with respect to the X_g -axis. During this test, the LRF sensor was located on the X_s - axis, i.e. the LRF laser beam was oriented to the robot front ($x_{si} = 17\text{cm}$, $y_{si} = 0$ and $\beta_i = 0$).

TABLE III: RESULTS OF THE TEST

Instant	Real robot position			Estimated robot position using EKF		
	x (cm)	y (cm)	θ ($^\circ$)	x (cm)	y (cm)	θ ($^\circ$)
1	0	0	0	-0.05	0.1	-0.001
2	27.36	0	0	24.821	0.092	0.27
3	27.36	0	47.5	24.17	-1.36	49.14
4	38.38	11.88	47.5	34.75	10.92	49.24
5	38.38	11.88	61.7	35.01	11.52	63.81
6	46.47	27.7	61.7	44.18	30.13	63.64

Regarding the hardware synthesis tasks, the respective results are shown in Table IV, which is related to the consumption of hardware resources of the EP4CCE115F29C7 FPGA device (Cyclone IV E), where it can be observed the little use of FPGA resources.

TABLE IV: HARDWARE SYNTHESIS RESULTS

Architecture	Total Logic Elements (LEs)	DSPs	Clk (Mhz)	Power (mW)
EKF Estimation Stage	22089 (19%)	175 (33%)	49,08	201,13

V. CONCLUSIONS

A Hardware/Software Co-design of the EKF algorithm applied to the Mobile Robotics Localization Problem was presented in this paper. The resulting system runs on an Altera Cyclone IV FPGA with a Nios II processor, adapted to the mobile platform Pioneer 3AT. The test results showed an appropriate performance of the FPGA implementation, consuming few resources and estimating the position and orientation of the robot along the path. However, it was observed that the algorithm (especially EKF Prediction Stage) was affected in long periods of use due to the error propagation generated by the odometric model. Moreover, it was presented the hardware and power consumptions resources for the project implementation.

As future works will be included a group of various LRF laser beams, each one with different β_i angle to improve the Estimation Stage. Besides, a hardware architecture in FPGA for the Prediction Stage will be developed.

ACKNOWLEDGMENT

The authors would like to thank to the Brazilian

institutions CAPES (Coordination for Higher Education Staff Development) and CNPq (National Council of Scientific and Technological Development) for the financial support and also to the Altera University Program.

REFERENCES

- [1] J. Arias-Garcia, R. P. Jacobi, C. Llanos, and M. Ayala-Rincon, "A suitable FPGA implementation of floating-point matrix inversion based on Gauss-Jordan elimination," in *Proc. 2011 VII Southern Conference on Programmable Logic (SPL)*, April 2011, pp. 263-268.
- [2] S. Cruz, D. M. Munoz, M. E. Conde, C. H. Llanos, and G. A. Borges, "A hardware approach for solving the robot localization problem using a sequential EKF," in *Proc. 2013 IEEE 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 20-24 May, 2013, pp. 306-313.
- [3] M. E. Conde, S. Cruz, D. M. Munoz, C. H. Llanos, and E. L. F. Fortaleza, "An efficient data fusion architecture for infrared and ultrasonic sensors, using FPGA," in *Proc. 2013 IEEE Fourth Latin American Symposium on Circuits and Systems (LASCAS)*, 2013, pp. 1-4.
- [4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Cambridge, MA, USA: MIT Press, 2005.
- [5] AdeptMobilerobots. (2012). "P3-AT datasheet". [Online]. Available: <http://www.mobilerobots.com/Libraries/Downloads>
- [6] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, Cambridge, MA, USA: The MIT Press, 2005.
- [7] K. Kozłowski and D. Pazderski, "Modelling and control of a 4-wheel skid-steering mobile robot," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477-496, 2004.
- [8] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1222-1227.
- [9] D. Munoz, D. Sánchez, C. Llanos, and M. Ayala-Rincón, "Tradeoff of FPGA design of a floating-point library for arithmetic operators," *Journal of Integrated Circuits and Systems*, vol. 5, no. 1, pp. 42-52, 2010.
- [10] Terasic. (2012). "Altera DE-115 development and education board". [Online]. Available: <http://www.terasic.com.tw>



Luis Federico Contreras Samam received his B.S. degree in electronic engineering from the Department of Electrical and Electronic Engineering from the National University of Engineering (UNI), Lima, Peru, in December 2010. He is currently working for his M.S. degree in mechatronics systems in the Department of Mechanical Engineering at University of Brasilia – UnB, Brasilia, Brazil. His research interests are reconfigurable systems for automation, mobile robotics and embedded systems.



Sérgio Messias Cruz received his M.S. degree, and is currently working for his Ph.D. degree in mechatronic systems in the Department of Mechanical Engineering at University of Brasilia – UnB, Brasilia, Brazil. He has technology degree in industrial mechatronics at the Federal Institute of Education, Science and Technology of Ceara – IFCE, Fortaleza, Brazil. His current research interests are reconfigurable systems, mobile robotics and embedded systems.



José Maurício S. T. Motta received his B.S. and M.S. degrees in mechanical engineering from the University of Brasilia – UnB, Brasilia, Brazil in 1986 and his Ph.D. degree in robotics technology from Cranfield University, Cranfield, UK in 1999. He teaches in the Graduate Program in Mechatronics Systems, Department of Mechanical Engineering at University of Brasilia, Brazil since 1990. He is a member of the IEEE. His current research interests include robot calibration, robot vision and robot off-line programming.



Carlos H. Llanos received his B.S. degree in electrical engineering from the University of Valle, Cali, Colombia in 1983, M.S. degree in computer science from the University of Minas Gerais – UFMG, Minas Gerais, Brazil in 1990 and a Ph.D. degree in electrical engineering from the University of São Paulo – USP, São Paulo, Brazil, in 1998. He teaches in the graduate program in Mechatronics Systems in the Department of Mechanical Engineering at University of Brasilia – UnB Brasilia, Brazil since 2003. He is a member of the IEEE. His current research interests include reconfigurable systems for automation, intelligent systems, instrumentation and embedded systems design.