An Application of Reconfigurable Architectures to the Localization Problem in Mobile Robotics

Luis Contreras, Sérgio Cruz, J. M. S. T. Motta, Carlos H. Llanos
Department of Mechanical Engineering, Mechatronics Graduate Program
University of Brasilia
Brasília, D.F., Brazil, 70910-900
Email: lcontreras@aluno.unb.br and {sergio, jmmotta, llanos}@unb.br

ABSTRACT
This paper presents a Hardware/Software Co-design of a probabilistic algorithm, specifically the Kalman Filter (KF). In this case we present a solution for calculating the path length, which can be used for several navigation techniques used in mobile robotics. The KF algorithm has been implemented and run on an Altera Ciclone II FPGA with a Nios II processor. To achieve this, we have developed the model of our mobile system applying concepts of odometry, considering the readings of the encoders on our mobile platform Pioneer 3AT (P3AT). Then we proceeded to develop the probabilistic algorithm of the Kalman Filter from our system model using as additional measure the information of path length provided by the internal robot system. This probabilistic algorithm is based on the concept of sensor fusion, allowing us to obtain the better estimate of the path length value. Finally, we show a comparison of the algorithm performance developed in software and hardware independently.

Categories and Subject Descriptors
I.2.9 [**Mobile Robotics**]: Robotics - Commercial robots and applications

General Terms
Algorithms, Measurement, Performance, Design.

Keywords
Odometry; Hardware/Software Co-Design; Sensor Fusion; Kalman Filter; Encoder.

1. INTRODUCTION
It is well known that many algorithms developed in software improve its performance using reconfigurable architectures in hardware based on FPGAs, reducing the execution time of the algorithm and thus increasing its speed [1], [2]. This is very important in real-time systems, like occurs in the area of mobile robotics. Since these kind of systems include a large number of variables provided from sensors such as encoders, gyroscopes, accelerometers, among others, we have the necessity of receiving and processing all those data almost in real-time for knowing some specific information, for instance, to know how many meters of the path length our mobile robot performed for going a given task.

The last question mentioned, it will be the task that we attempt to calculate using the information of the measurement systems. The first information is about velocities that is provided by encoders corresponding to the right and left wheels; besides this we have the information of travel length provided by the internal system of the P3AT robot [3].

The problem here is that each of these measures has an uncertainty (σ) specific to each measurement system, which gives troubles for estimating the variable we want to know, without considering further that each of these uncertainties can be propagated over time due to our system and measurement models that we have considered.

It is here that goes into the concept of Kalman Filter, which is a stochastic algorithm that takes advantage of the fusion of these measurement data to provide a better estimate of our variable desired and of the final uncertainty, which is generated by the propagation of the errors of each measurement system [4].

This work presents an FPGA implementation using the Kalman Filter algorithm for the problem of calculating the path length by a mobile robot using data of the internal robot system and encoders.

In this context, novel contributions of this paper include: (a) the implementation of a KF algorithm in FPGA, using floating-point representation [5], and (b) the validation of the results in terms of use of resources in FPGA.

The importance of calculating the path length is that many systems in Mobile Robotics needing that information for solving other problems as is the case of localization, mapping, avoiding of obstacles, planning and optimization of path.

This work is focused to treat localization problem issues, which can be later applied for solving simultaneous localization and mapping problem (SLAM).

2. RELATED WORKS
There are previous works related to the implementation of probabilistic algorithms in FPGA. For instance, Cruz *et al*. [1] presented a FPGA implementation of the Extended Kalman Filter (EKF) applied to the problem of localization in mobile robots. In that work was described a sequential approach of the EKF, update the algorithm for localization problem using multi-sensors such as ultrasonic (Sonar) and Laser Range Finder (LRF). In that paper the algorithm of Extended Kalman Filter was implemented using arrays in the order of 3x3, 3x2, 3x1. For our case we will implement the Kalman Filter Algorithm in a particular way oriented for scalars (arrays of 1x1).

On the other hand, a hardware implementation (of the static case of sensor fusion) was developed in an FPGA (see reference [2]). In that case, the estimate of the distance was achieved by using a sensor fusion method involving both an ultrasonic and infrared sensors. That work had as result a speed up in the hardware solution in comparison with the embedded software implementation. That paper has the similarity with our work taking into account that the sensor fusion algorithm was implemented using scalars, such as it was developed in our work. The difference is that we will work using a KF Algorithm.

3. KALMAN FILTER ALGORITHM

The Kalman Filter is a recursive probabilistic algorithm that it considers the random state variable x_k, noise process w_k and measurement noise v_k as Gaussian distribution [4]. The Kalman filter can be divided in 2 processes (or stages): *Prediction* and *Estimation*.

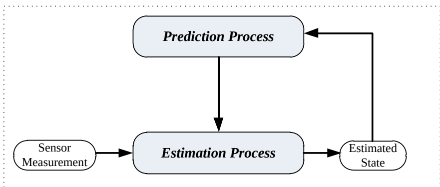

Figure 1. Diagram of KF.

In this case, the algorithm of Kalman Filter (for discrete time) is laid out in the following equations:

Predict state: $$x_k^{--} = A.\, x_{k-1}^{+} + B.u_{k-1} \quad (1)$$

Predict variance: $$P_k^{--} = A.\, P_{k-1}^{+} .\, A^T + Q \quad (2)$$

Kalman gain: $$K_k = P_k^{--} .\, H^T .(H.\, P_k^{--} .\, H^T + R)^{-1} \quad (3)$$

Estimate state: $$x_k^{+} = x_k^{--} + K_k .\, (z_k - H.\, x_k^{--}) \quad (4)$$

Estimate variance: $$P_k^{+} = P_k^{--} - K_k .\, H.\, P_k^{--} \quad (5)$$

In some variables, the superscript '--' means predicted value and the superscript '+' means estimate value. All the symbol representation is show in Table 1.

Table 1. The symbol description for the KF algorithm

Symbols	Description
X	State variable
U	Input variable
A	System matrix (1x1)
B	Input matrix (1x1)
P	Error covariance
Q	Variance of permanent process noise
K	KF Gain
H	Measurement matrix (1x1)
R	Variance of permanent measurement noise
Z	Sensor measurement
K	Actual Time

The *Prediction Process* is defined for the Equation (2), used to *Predict state* and Equation (3), used to *Predict variance* in an instant *k*.

On the other hand the *Estimation Process* is laid out for the Equation (3), where the *Kalman Gain* is calculated, the Equation (4), used to *Estimate state* and also the Equation (5), used to *Estimate variance* in an instant k.

3.1 System Model

Now, let us see the topic of the system model. In this case, we have to consider that our platform is a differential drive robot. A differential drive is a type of mobile platform that has the left and right motors driving separated. This configuration allows a simple and common mechanism of movement easy to control [6]. In the Figure 2, we show our mobile robot in a Cartesian system, where V_R and V_L are the linear velocities of the right and left wheels respectively and L is the path length of the mobile robot.

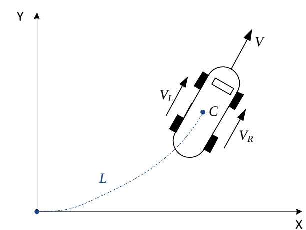

Figure 2. Path Length by the mobile robot.

Through of odometry [7], we can use the information of the velocities V_R and V_L provided by the encoders. Then the Equation (1) shows how to calculate the path length (ΔL) by the mobile robot in a small period of time (Δt).

$$\Delta L = (V_R + V_L) .\, \Delta t /2 \quad (6)$$

Due to our system works in discrete time, we define the path traveled in the instant k as L_k with the following equation:

$$L_k = L_{k-1} + \Delta L \quad (7)$$

Let us consider a linear system [8] that is represented by the Equation (3).

$$x_k = A.\, x_{k-1} + B.u_{k-1} + w_{k-1} \quad (8)$$

Now, if we consider that our system model has the form of the Equation (3), then $x_k = L_k$, $A=1$ and $B.\, u_{k-1} = \Delta L$.

3.2 Measurement System

In this case, the internal robot system will provide us information about the path length (z_k). Then let us consider a linear measurement system represented in Equation (9):

$$z_k = H.\, x_k + v_k \quad (9)$$

where $H=1$ *and* v_k *(0, R)* is the measurement noise with mean 0.

4. FPGA IMPLEMENTATION

The system implementation was performed in an Altera Cyclone II FPGA including a NIOS II processor. Basically, the system was developed as follows: 1st, 2nd, 3rd, 4th and 5th KF equations were implemented in hardware to compute the predicted state, predicted error variance, Kalman gain, estimated state and estimated error variance, respectively. The NIOS II processor was used as a controller to acquire the data from sensors and send them (after treated) to the hardware.

4.1 Integrated Hardware/Software Co-Design

This project consisted of an Integrated Hardware/Software Co-Design, which in the area of software used the NIOS II processor to acquire the information given by the encoders and the internal system of the Robot, using the RS232 Interface. These data will be processed and prepared to be sent to the hardware area through the AVALON Bus, where our KF algorithm is being executed. After the algorithm execution in the hardware area, the computed results, as the case of state predicted (x_k^{-}), would be sent back through the AVALON bus to NIOS II processor to print and show the result on the computer console. The Figure 3 shows the diagram of the general architecture of all Integrated Hardware/Software Co-Design, considering the units, modules and interfaces that were used.

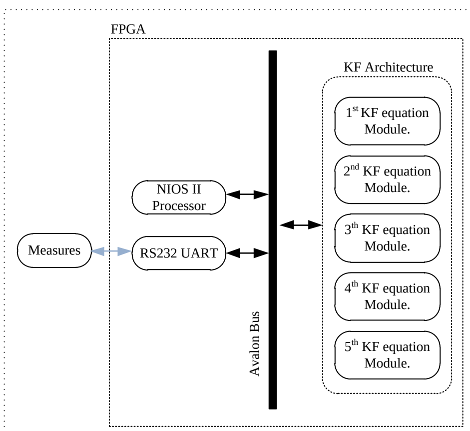

Figure 3. Diagram of the general architecture.

4.2 Hardware Architecture

The hardware development was based in previously developed floating-point units [5] such as Addition, Subtraction, Multiplication and Division units, operating in a single-precision format accuracy. The internal diagram of the KF Hardware Architecture is shown in the Figure 4.

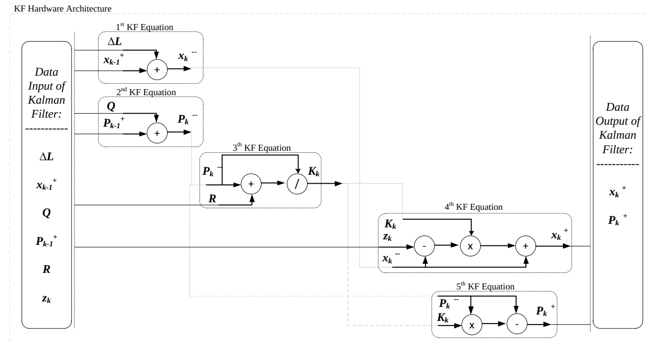

Figure 4. Kalman Filter Hardware Architecture

In Figure 4 we present the architecture of the hardware module for the 1st KF equation. In this hardware module, we use an Addition unit which receives of x_{k-1}^{+} and ΔL. The result is the *Predicted State* (x_k^{-}). Also it appears the module architecture of the 2nd KF equation, which also uses an Addition unit which receives signals of P_{k-1}^{+} and Q and the result is the *Predicted Variance* (P_k^{-}). Then we have the module architecture of the 3rd equation corresponding to the Gain Kalman Filter, where the signals variance R and P_k^{-} enter to addition unit, finally giving the *Kalman Gain* (K_k). The following module corresponds to 4th KF equation, there we use three arithmetic units: Subtraction, Addition and Multiplication, having as a result the *Estimated State* (x_k^{+}*)*. Finally there is the module architecture of the 5th KF equation, which gives as a result the *Estimated Variance* (P_k^{+}).

5. PRELIMINARY RESULTS

5.1 Hardware Implementation

After the implementation of KF algorithm in the Altera DE2-115 board we proceeded to do some tests using the Pioneer 3-AT mobile platform.

One of the tests consisted in that our robot had to perform the path shown in the Figure 5.

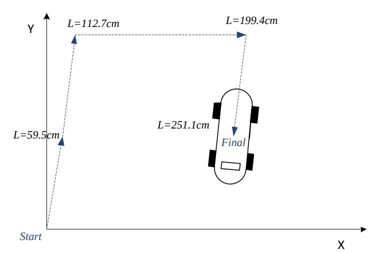

Figure 5. Robot trajectory for the test.

The results obtained of the test are shown in Table 2 where the 1st column relates to the real path length, the 2nd and 3rd column correspond to the estimated state (x_k^{+}) and estimated variance (P_k^{+}) respectively of the KF using hardware modules discussed above.

Table 2. Results of the test

Real path length *(cm)*	Hardware Solution	
	x_k^+ *(cm)*	P_k^+ *(cm)*
0	0.000042	0.291684
59.5	60.962372	0.337386
112.7	113.285233	0.337386
199.4	200.439194	0.337386
251.1	252.781815	0.337386

5.2 Software Implementation

Moreover entire KF was implemented in software using NIOS II processor, without using hardware modules. The values of the data obtained by this software solution were the same as in Table 2.

5.3 Hardware Synthesis

The synthesis results are shown in Table 3, which is related to the consumption of hardware resources for EP4CE115F29C7 FPGA Device of Family Cyclone IV E [9], where you can see the few use of FPGA resources.

Table 3. Hardware synthesis results

FPGA Family Cyclone IV E	Device EP4CE115F29C7	
	Value	*FPGA resources*
Total logic elements	17152	15%
Total combinational functions	16258	14%
Dedicated logic registers	5781	5%
Embedded Multiplier 9-bit elements	60	11%

Using the ModelSim Simulation tool, it was achieved to determine that our KF general module (hardware implementation in the FPGA) executes the algorithm in 19 clock cycles. Considering the clock of the Altera DE2-115 board operates to 50MHz so the time required for our algorithm is 0.38 us.

Furthermore, we seek to find the performance of the implementation entirely in software, so the algorithm was programmed and run in NIOS II soft processor, resulting that the time required for the implementation of the algorithm in software was 34us; this means that the hardware solution was 89 times faster than the software solution.

6. CONCLUSIONS

In this paper, we have presented the calculation of the path length by a mobile robot based in a Hardware/Software Co-design approach. For this purpose, the Kalman Filter algorithm was used and our hardware architecture was implemented on Altera Cyclone II FPGA. The results showed an acceptable estimate of path length, however in the tests it was seen that our algorithm was affected by the error propagation due to odometry model [6].

In the case of the hardware implementation in the FPGA, was obtained a speeding up of 89 times compared to the solution purely in software (using the NIOS II soft processor).

This architecture can be used in mobile robotics applications where as from the information provided by sensors, we use the KF and get to have a better estimate of our variables of interest.

The architecture could also be used in other applications of different purposes but should be considered that having a new model system this entails modifying the 1st and 2nd equation of the KF, which would mean developing new hardware modules for these equations.

7. ACKNOWLEDGMENTS

The authors would like to thank the Brazilian institution CAPES (Coordination for Higher Education Staff Development) for its financial support and to the Altera University Program.

8. REFERENCES

[1] Cruz, S.; Munoz, D.M.; Conde, M.; Llanos, C.H.; Borges, G.A., "FPGA implementation of a sequential Extended Kalman Filter algorithm applied to mobile robotics localization problem," Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on , vol., no., pp.1,4, Feb. 27 2013-March 1 2013.

[2] Conde, M.E.; Cruz, S.; Munoz, D.M.; Llanos, C.H.; Fortaleza, E.L.F., "An efficient data fusion architecture for infrared and ultrasonic sensors, using FPGA," Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on , vol., no., pp.1,4, Feb. 27 2013-March 1 2013.

[3] AdeptMobilerobots. (2012) ”P3-AT datasheet”. [Online]. Available: http://www.mobilerobots.com/Libraries/Downloads

[4] 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.

[5] 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.

[6] D. Clark and O. Michael, Building Robot Drive Trains: McGraw Hills, 2008.

[7] Kok Seng Chong; Kleeman, L., "Accurate odometry and error modelling for a mobile robot," Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on , vol.4, no., pp.2783,2788 vol.4, 20-25 Apr 1997

[8] L. A. Aguirre, Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais, 3rd ed. Belo Horizonte, Brazil: Editora UFMG, 2007, (in Portuguese).

[9] Terasic. (2012) "Altera DE-115 development and education board". Available: http://www. terasic. com. tw