A PID Controller Applied to the Gain Control of a CMOS Camera Using Reconfigurable Computing

Drausio Linardi Rossi, Vanderlei Bonato, Eduardo Marques
Institute of Mathematical and Computing Sciences
The University of São Paulo
São Carlos, Brazil
e-mail: drausior@icmc.usp.br; vbonato@icmc.usp.br; emarques@icmc.usp.br

João Miguel Gago P. de Brito Lima
Department of Electronic Engineering and Informatics
The University of Algarve
Faro, Portugal
e-mail: jlima@ualg.pt

Abstract —**This paper shows a PID (Proportional, Integral, Derivative) controller, implemented in a reconfigurable hardware, to control the gain of a CMOS image sensor. The main functions of the proposed system are: image acquisition, histogram building, histogram analysis, and PID gain control based on the histogram analysis. The system has several functional modules working in parallel in order to achieve high performance. As a result, the system is able to compute the whole PID functions in 4.8*μs*, allowing its application in real time systems.**

Keywords: *PID Controller; Reconfigurable Computing; FPGA; CMOS camera.*

I. INTRODUCTION

Modern image systems should have the ability to adapt itself to ambient light in order to ensure robust and reliable performance in all kinds of environmental conditions [1]. Parameters like sunlight, rain, fog, car lights and sudden changes of these conditions are common in most embedded applications.

In these vision systems, the first task to be accomplished is the perception of the environment. For this task a right image sensor gain tuning is essential. The performance of the sensor will affect the performance of the system where it is included, and the precision with which the sensor parameters are known is necessary to compute the precision of the final results of the system using it[2].

Most gain control for image sensors are patented, or are intellectual property from sensor providers. Others [3] use histogram bin analysis to create a gain control. The method presented in this paper shows a complete histogram based analysis and a PID controller for analog sensor control.

PID controllers are simple devices widely used in several applications. In the robotic field they are usually used to control the position of cameras, wheels, manipulators etc[4]. The popularity of these controllers comes from the robustness of the response obtained with only 3 terms to tuning. For an accurate tuning a real time implementation is required, so, the present work give us a platform over it several experiments and research could be performed. In the present case, the PID controller deal with a process that experiments a load disturbance.

In this context, the main contributions of this paper are:

- shows a PID gain controller for image sensor implemented in hardware;
- presents an histogram analysis method for the computation of the PID error variable.

This paper has the following layout: next section presents a brief resume of theoretical aspects of digital PID controller, image histogram, and reconfigurable computing. In section 3, the system implementation is described. In section 4 the main results are demonstrated. Finally, the paper ends with some conclusions and suggestions for future work.

II. THEORY FUNDAMENTALS

A. Digital PID Controller

A typical closed loop continuous parallel PID controller [5] is show in the Figure 1.

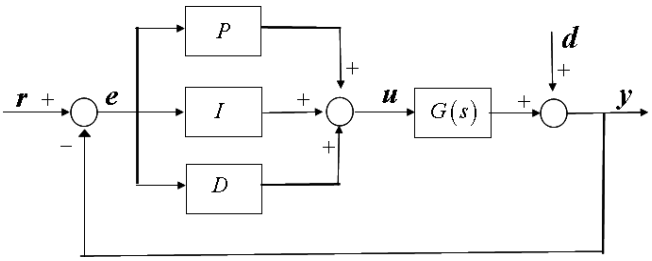

Figure 1. Close loop continuous parallel PID controller.

The signals ***r, e, u, d,*** and ***y*** represent the reference, error, control signal, disturbance, and output system, respectively. The $G(s)$ block represents the system that needs to be controlled, that in this case is the image sensor gain.

In time domain, the PID controller can be described by the relationship between the control and the error signals (1).

978-0-7695-4551-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ReConFig.2011.2

IEEE computer society

$$u(t) = k_c e(t) + \frac{1}{t_i}\int^{t} e(\tau)\,d\tau + t_d \frac{d}{dt} e(t) \qquad (1)$$

As expected, the PID controller law is defined in terms of 3 parameters, proportional gain $\boldsymbol{k_c}$, integral time $\boldsymbol{t_i}$, and derivative time t_d. The methods of evaluation of these parameters are called PID tuning methods [8]. There are several classes of PID tuning methods; the real time implementations of some of theses methods will be the next step of our research. For the purpose of this work, the tuning is empirically made.

Regarding an FPGA (Field-Programmable Gate Array) implementation a digital version of (1) should be accomplished. Thus, taking into account that signals are sampled with sampling period T_o, the integral and derivative approximations for the corresponding terms, allow us establish (2).

$$u(k) = k_c e(k) + \frac{T_o}{t_i}\sum_{i=0}^{k-2} e(i) + \frac{t_d}{T_o}\left(e(k) - e(k-1)\right) \qquad (2)$$

The incremental form of PID algorithm is obtained by computing de difference between the control signal u at sampling time k, and $k-1$, so, the incremental form is given by (3).

$$\Delta u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2), \qquad (3)$$

where the coefficients q_0, q_1, and q_2 are evaluated by the expressions (4):

$$q_0 = K_C\left(1 + \frac{T_d}{T_S}\right),\ q_1 = -K_C\left(1 + 2\frac{T_d}{T_S} - \frac{T_S}{T_i}\right) \text{ and}$$

$$q_2 = K_C \frac{T_d}{T_S} \qquad (4)$$

B. Histograms

An image histogram is a distribution of the amount of pixels for the different levels of grey [9]. This distribution is normally represented by a bar graphic that shows, for each level of grey, the percentage (5) of pixels p_r that are present in the image:

$$p_r(r_k) = \frac{n_k}{n} \qquad (5)$$

where $0 \le r_k \le 1$, k= 0,1, … , L-1; r_k represents the grey level $\boldsymbol{k}$, and $\boldsymbol{L}$ is the total number of grey level for a given image. The total image pixel number and the number of pixels that have the same $\boldsymbol{k}$ grey level are $\boldsymbol{n}$ and $\boldsymbol{n_k}$.

For a given image, the shape of its histogram tells us about its quality regarding its contrast and sharpness. We can also deduce if an image is predominantly dark (with a predominant percentage of pixels in lower levels), or clear (with a predominant percentage of pixels in higher levels).

C. Reconfigurable Computing and FPGAs

Reconfigurable computing can be defined as the process of exploit the best potential of a reconfigurable hardware [5], whose structure is modified to use the best computing approach to speed up that application [6].

A complete reconfigurable system includes also software, compilers and applications. Reconfigurable computing can be understood at chip level, during the reconfiguration process of the structure of a reconfigurable device that can be done at start-up time or at run-time. Progress in reconfiguration computing has been great in the last years. This is mostly due to the wide acceptance of FPGAs that now are established as the most widely used reconfigurable devices.

An FPGA is an integrated circuit designed to be configured by the programmer or designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL). The ability to update the functionality after shipping, partial re-configuration of the portion of the design and the low non-recurring engineering costs, offer advantages for many applications.

In many cases, applications solved with FPGAs supplant the equivalent of tens or hundreds of contemporary microprocessors or digital signal processors (DSPs). High performance is achieved by (dynamically) building custom computational operators, pathways, and pipelines suited to specific properties of the task at hand. With this approach, characteristics of a particular application, such as parallelism, locality, and data resolution can be fully exploited.

III. System Implementation

The PID controller showed in Figure 1. was implemented and validated using the Altera DE2-70 Board and the TRDB_5DM CMOS camera from Terasic. This board contains a Cyclone II 2C70 family FPGA. This development platform is equipped with hardware and software tools which accelerates the time development of SoC (System On a Chip) embedded systems.

The TRDB_5DM 5 Mega pixels CMOS camera is prepared to be connected to the development board, which also accelerates the development time. Figure 2. shows the block diagram of the digital camera design. The sensor data and status are read through the DATA, FVAL and LVAL lines. The output of PID controller (signal $\boldsymbol{u}$, Figure 1.), which is gain analog control, is send to the sensor through the I2C lines, so, the results were acquired through this line using a I2C analyzer tool, that captured the communication between FPGA and CMOS image sensor.

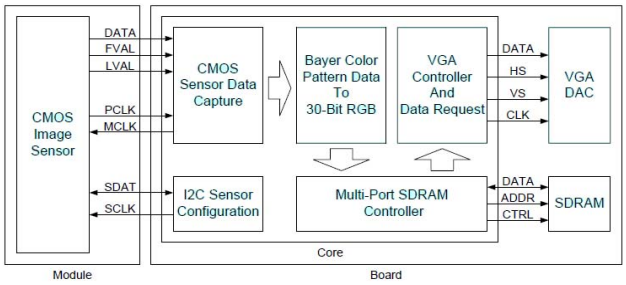

Figure 2. Block diagram of the digital camera design [10].

The hardware configuration was developed in Verilog hardware description language and was based on a previous project supplied by Terasic. This base project was able to read data from sensor and send commands to write data to the camera control registers [10].

A. Error Calculation

The error signal (***e***, Figure 1.) is the difference between the reference ***r*** and the output ***y***. For each frame the evaluation of ***y*** is based in its corresponding histogram analysis. Figure 3. shows the parameters that are combined for the total error calculation. The histogram zone shows the grey levels. The levels from left (1 to 5) indicate the considered dark zones. The levels from right (6 to 10) are considered light zones. The average value is the average value from the total number of pixels read from the sensor. In this case the average value is 625000. The sensor has 5 mega pixels, the read out mode is set to read a 4 pixel average. This total number divided by 2 results 625000.

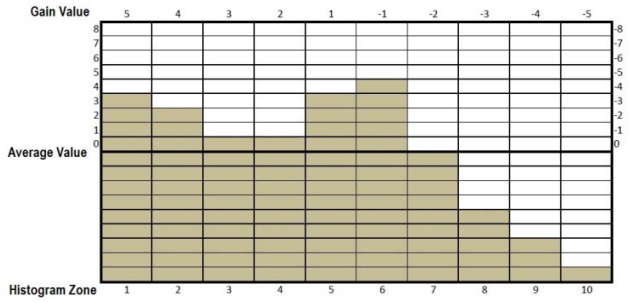

Figure 3. Histogram analysis for error calculation.

First, portion errors are calculated for each histogram zone. The value that exceeds the average value, is multiplied by a constant defined for each histogram zone. Then the total error is the sum of all values. In the example, zone 1 has a value 5. The histogram of the acquired image exceeds 3 units from the average. So the error contribution from this zone is $3\times5=15$. For zones 1 to 5, the error signal is positive, which indicates that the gain factor contributes for a gain increase. If the major part of the pixels is located in the dark zones, the gain should increase for the next image, trying to have more balance among histogram zones. The zones from 6 to 10 indicate more light. If an image has more pixels in those histogram zones, the gain should be decreased. The gain multiplier for each zone increases proportionally to the distance that the zone is far from the middle (zones 5 an 6). This is because the far zones indicate very dark pixels (zone 1) which should be compensate by a gain increase, or very light pixels (zone 10) which should be compensate by a gain decrease. In this case the total error is $3\times5+2\times4+3\times1-4\times1=22$. This value indicates that the error is 22 positive, which increases the gain for the next frame.

B. Sensor gain control

The THDB-D5M CMOS digital camera has a Global Gain register that sets all four individual gain registers, green 1, green2, red and blue (RGB format). Legal values are: [8, 127].Figure 4. shows the signal path through the voltage pixel to the data bus. The control signal ***u*** delivered by PID controller actuates in the analog gain, that is the voltage multiplier from the pixel voltage.

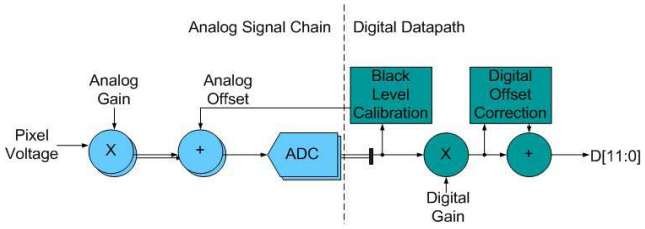

Figure 4. Signal Path[10].

The histogram construction, error calculation and PID controller modules were implemented also in Verilog HDL. These modules can be see in Figure 5. The histogram is mounted collecting data from the lines DATA and FVAL. The DATA path is a 12 bit parallel bus that transmits the pixel voltage from the sensor to the FPGA. If the frame is valid, which is indicated by FVAL signal, the histogram analysis is valid and the PID control module can be feed with the resulted error value. After the PID control calculation, a new gain value is send to the CMOS Image sensor via I2C communication line. The clock frequency was 50MHz and the I2C communication line was set to 100 kbits/s.

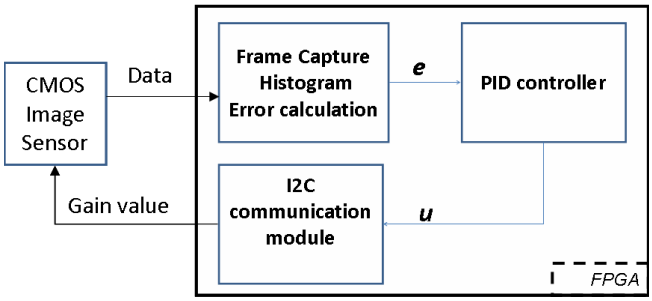

Figure 5. FPGA system implementation.

The total amount of FPGA resources used in the implementation is showed in TABLE I.

TABLE I. PROJECT COMPILATION REPORT

FPGA: Cyclone IIEP2C70F896C6 from Altera			
Resource	*Used*	*Available*	*Percentage*
Total logic elements	11,031	68,416	16%
Total memory bits	916,920	1,152,000	80%
Total PLLs	3	4	75%

C. *PID implementation*

Figure 6. shows the implemented PID architecture. It uses 3 registers, 2 two adders, three multipliers and one subtractor. The arithmetic multiplications were done based on shift instructions, so it was possible to multiply or divide the values by exponentiation on base 2.

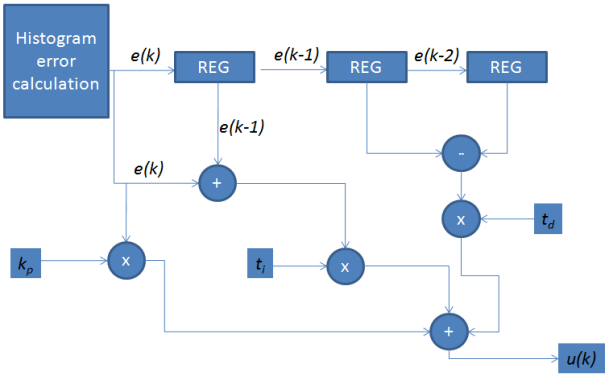

Figure 6. PID Architeture (simplified version).

Figure 7. shows the histogram calculation and error calculation implementation. It uses 20 registers, 1 adder, and 10 multipliers. The histogram and the error calculation are made parallel to the sensor data reading. After a frame is read, the final error calculation takes one clock cycle to sum all the error registers to calculate the final error. This allows this hardware implementation to be done inside a sensor. The time limitation for this implementation is the I2C communication line, that takes about 160 times the time of the hardware final calculation.

IV. EXPERIMENTAL RESULTS

The system was exhaustively tested and some results were selected to be presented in this section. The controller showed a good behavior in different types of environment light. The system behavior was tested with an abrupt light change: inside a dark room, a light was turned on. This operation represents the disturbance illustrated by the ***d*** signal from Figure 1.

The PID controller was empirically tuned and the control signal is graphically represented in this section.

In this test phase of the controller, only a discrete set of the PID parameters are needed, and for computational and system implementation reasons, these parameters are given by (6).

$$k = \frac{1}{2^n} \tag{6}$$

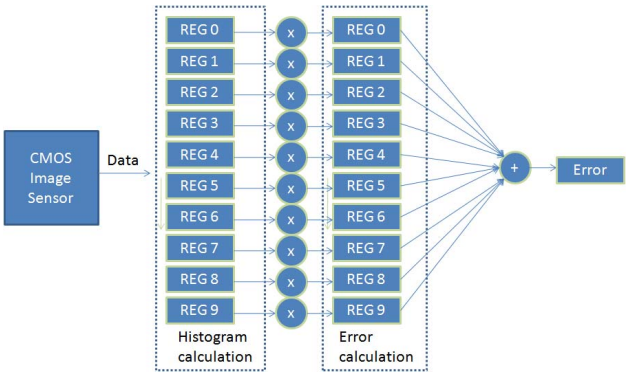

Figure 7. Histogram error calculation.

The PID parameters k_c, t_i and t_d values were set by shift operations, so these values could vary from 1 to 2.33E-10, when n varies from 0 to 32.

The system response (control signal, ***u***) can only vary between values 8 and 127 that, according to the manual [10], are the legal limits for the gain register. The controller gain output was acquired with an I2C analyzer.

The testes were done using a 50MHz clock for the FPGA and a 100Kbps I2C speed.

First only the proportional part of the controller was tested. Figure 8. shows the best result obtained for the analog gain, with a proportional controller set with k_c = 1.53E-05. The tests were made changing the values of n from 1 to 32. The system was stable with values from 2.33E-10 to 6.1E-05, which means ***n*** varying from 14 to 32. For k_c values greater than 1.2E-03 the system showed an unstable behavior.

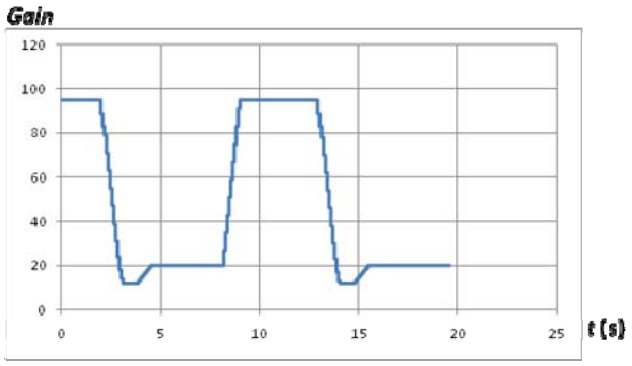

Figure 8. Control signal obtained when the system is submited to Proportional control whith k_c = 1.53E-5 .

For a quicker response, the integral component was tested with the proportional using the tested value (k_c = 1.53E-5). The limits of stability for the integral gain t_i were

from 2.32E-10 to 1.25E-1. The result was a better response to the final gain value. In Figure 9. we can see also that the threshold values (127 and 8) were reached by the PI controller.

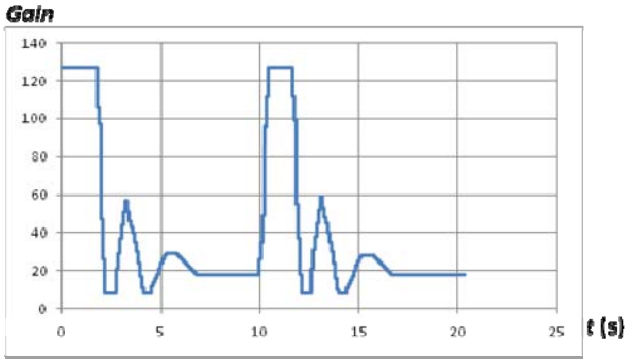

Figure 9. Control signal obtained when the system is submited to *PI* control, k_c = 1.53E-5, t_i = 6.25E-2.

Finally the derivative part was also taken into account The result is showed in Figure 10. The PID has a settling time lower than the *PI* controller. This time goes from approximately 5 to 3 seconds and a lower overshot.The limits of stability for the derivative constant t_d vary from 0.25 to 1.25E-1.

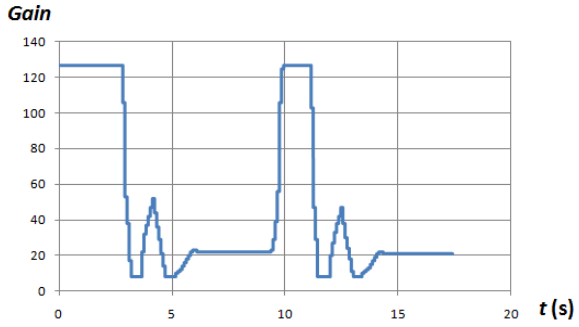

Figure 10. PID control, k_c = 1.52E-5, t_i = 6.25E-2, t_d = 0.25

Figure 11. shows the time for frame reading t_f, register value programming via I2C line t_p, and time calculation of the new gain based on histogram analysis t_c. Theses times are $t_f = 66ms$, $t_p = 800\mu s$ and $t_c = 4.8\mu s$. These times consider maximum possible hardware clock speed calculated by the Quartus II timing analyzer.

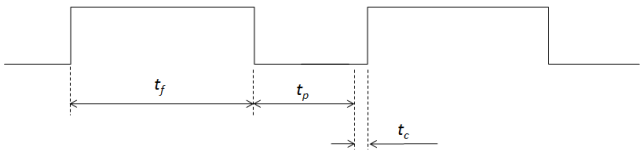

Figure 11. . Times for frame reading, register writing and gain calculation

V. Conclusions and Future Works

In this paper we proposed and analyzed an error calculation method based on histogram analysis and PID controller to control a CMOS camera analog gain. The results demonstrated the usability of the controller along with its performancefor a manual tuning.The system has been completely developed in hardware, which demonstrates its feasibility of being embedded in an image sensor. The time limitation for this implementation was the I2C communication line, which takes about 160 times of the total time of the PID controller. The proposed system is able to be adjusted to a different frame rate and image resolution automatically. As a future work, a real time automatic tuning could be developed in order to avoid the necessity of a manual adjustment of the PID parameters.

Acknowledgment

The authors thank the ICMC colleagues for support and suggestions.

References

[1] G. Tarak, T. Mohan Manubhai,Pedestrian Protection Systems: Issues,Survey, and Challenges,IEEE trans. intel. trans. sys., VOL. 8, NO. 3, SEPTEMBER 2007

[2] F. Corrêa Alegria, A. Cruz Serra, Standard Histogram Test Precision of ADC Gain and Offset Error Estimation, IEEE,trans. intrum. meas. VOL. 56, NO. 5, OCTOBER 2007 1527

[3] N. Itani,C. Wang, D. Welland, Histogram-based Automatic Gain Control Method and System for Video Applications, US Patent US00752219B2

[4] F. Michaud, and D. Letourneau, "Mobile robot that can read symbols," in Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 29 July-1 Aug. 2001, pp. 338-343.

[5] S. Hauck, "Reconfigurable Computing the Theory and Practice of FPGA Based Computation", Elsevier, 2008.

[6] C. Bobda, Introduction to Reconfigurable Computing, 2007, Springer.

[7] M. A. Johnson and M. H. Moradi,PID Control: New Identification and Design Methods, Springer, New York, 2005.

[8] K. J. Åström and T. Hägglund, Advanced PID Control,ISA – Instrumentation, Systems and Automation Society, NC, USA, 2006

[9] O. Marques Filho, H. Vieira Neto, Processamento Digital de Imagens, 1st edition, Rio de Janeiro, 1999.

[10] TRDB D5M User Guide, Terasic, 2008, www.d5m.terasic.com.