Journal of Theoretical and Applied Information Technology
31st December 2019. Vol.97. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REDUCED SISO POSITION MODEL AND PD CONTROL OVER FPGA FOR A DIFFERENTIAL ROBOT PLATFORM THAT USES A TWIN ULTRASONIC SENSORS SYSTEM

[1]EDWAR JACINTO GÓMEZ, [2] FREDY HERNÁN MARTÍNEZ, [3]FERNANDO MARTÍNEZ SANTA

[1,3]Assistant professor, Universidad Distrital Francisco José de Caldas, Technology Faculty, Bogotá, Colombia
[2]Associated professor, Universidad Distrital Francisco José de Caldas, Technology Faculty, Bogotá, Colombia
E-mail: [1]ejacintog@udistrital.edu.co, [2]fhmartinezs@udistrital.edu.co, [3]fmartinezs@udistrital.edu.co

ABSTRACT

This paper shows the design, calculus and implementation of a digital PD (proportional derivative) position control for a differential two-wheeled robot commanded by means of a FPGA (Field Programmable Gate Array). It started with the design of the digital driver for the ultrasonic sensors (located in both sides of the robot) which gets the differential distance between walls or any other obstacle; this is followed by the modeling of the robot taking it into account as a dynamic plant and giving to both sensors and both motors the treatment of only one sensor and only one motor, which reduces the model to a simple SISO system. After that, this shows the design and optimization of the controller using the root location approach, then the digital controller is obtained by means of applying a discretization method over the designed controller. After, the design of the digital controller is implemented on a FPGA device using VHDL as hardware description language. Next, some simulations of the system response in open loop and along with the digital control are shown, as well as the block diagram of the whole digital control and the code of the PD controller are shown. Finally, this shows how the generated controller improves the behavior of the position of the robot for most of the testing cases.

Keywords: *PD Controller, Differential Robot Platform, Digital Position Control, FPGA, Ultrasonic Differential Sensor*

1. INTRODUCTION

One of the basic tasks in autonomous mobile robotics is to develop speed and position controls with high performance and acceptable response speed. For this task it is required to know the coordinates of the robot at each moment of time, therefore the position is dynamically measured to determine the speed of the mobile, in this case, an ultrasonic type position sensor is used as a distance measurement element.

In addition to this, in order to achieve a correct functioning of the digital controller, a correct modeling of the plant and design of the control system must be carried out. In this case, a Proportional and Derivative (PD) controller has been chosen, which will be responsible for performing the respective correction of the position and speed of the system so that the mobile robot navigates in an unknown but static environment [1].

In this way, the structure and operation of a two-wheel type differential robot is planned, which achieves bidirectional linear displacement, reducing the oscillations in its movement by means of a digital position controller [3], in order to stabilize the position of the robot in a central point of an environment with two side walls [2].

Within its versatility, digital PID controllers and their variations can be used in dynamic positioning control, because it provides an adequate response speed for this type of applications [4], a characteristic that is important to bear in mind, since, the time in which control decisions must be made in the face of environmental variations in the robot are critical and depend proportionally on the speed of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processing, of the information by the digital system where the control is implemented.

In addition to this, the derivative contribution of the PD, takes into account an instantaneous error provided by the input information that reduces the cumulative error in the reading time frame, which decreases the steady state error [5]. By making a detailed model that takes into account the position and speed of the mobile with which adjustment instructions are calculated that are governing the movement of the robot [6] [7], the speed control is allowed to be bidirectional, that is, the adjustment variables respond to a change in both the direction of movement and the speed of the motors, to allow the robot to go forward and backwards [8].

So far, the response to environmental factors and the behavior of the system in the face of changes in information readings has been explained. The reading processes have been facilitated thanks to the simplicity and low cost of the devices with which the readings of the physical variables are made.

Obviously it is the case of ultrasonic sensors, which are not suitable for environments with high hostility and even with large variations [9], in addition to having measurement limitations, in that their accuracy becomes lower at greater distances and not they have a large sampling rate [10], which makes them barely usable for the navigation application of a low-speed mobile robot.

The robot has ultrasonic sensors, which allow to identify objects and nearby structures, with high precision and avoid them so as not to cause collisions, or to maintain a fixed distance from an object or wall, and even to follow up. All of the above could even be carried out Omni directionally, if that were the case, with the use of several ultrasonic sensors placed in a panoramic way. [11]

There are currently several applications of these robot models, because the ultrasonic waves are emitted by the sensors and returned to them, it is possible to obtain various data from these receptions, since the sensor measures the time it takes the wave to go and return to a surface being deduced, for example, the distance or the frequency; variables that can even allow studying and determining the properties of various materials, [12] the image capture taking into account the spectral wavelengths, etc. [13]

Other applications are aimed at people with physical disabilities, using the adaptation of the sensor and control system in assistive devices for the blind [14] and for the control of wheelchairs. [15]

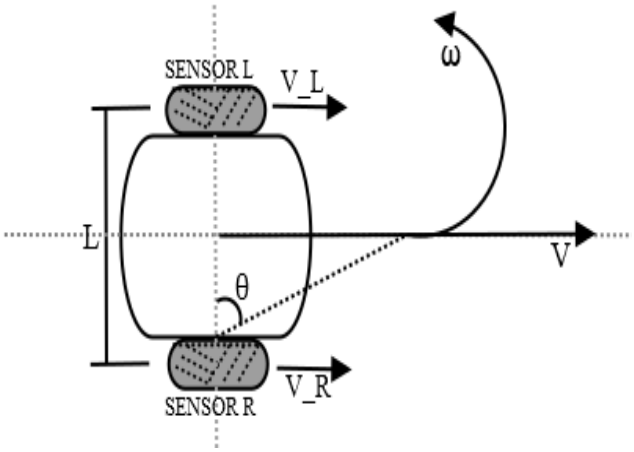

Figure 1. Structural Diagram Of The System.

2. METHODOLOGY

The implementation of a differential mobile robot that navigates in a static environment with two- side walls, that will be counted with a pair of ultrasonic sensors, of which a driver in VHDL was made; This platform mentioned complies with the form factor (standard) of an Arduino. The work shows the mathematical modeling of the displacement of the mobile robot, analyzing its geometry, electrical and physical characteristics, in order to generate a digital controller that will be implemented in an FPGA. The mathematical model, the design and its respective implementation in hardware will be shown.

2.1. Design Of The Ultrasonic Sensor Driver

To realize the ultrasonic sensor driver that will allow to measure the distance from the sides of the robot to the walls, the VHDL design of a state machine that performs the reading of the sensor in centimeters is performed. For this design, the operation of the sensor is taken into account, which is presented below; it has a pin that must be on high for a period of approximately 10 microseconds, which will allow it to emit a burst of 8 pulses; once the burst detects the closest surface, the wave is reflected obtaining pulses that are equivalent to the time it takes to travel the wave. The remote conversion is made with the product of the speed of sound (343.2 m / s) and the time read by the sensor receiver.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

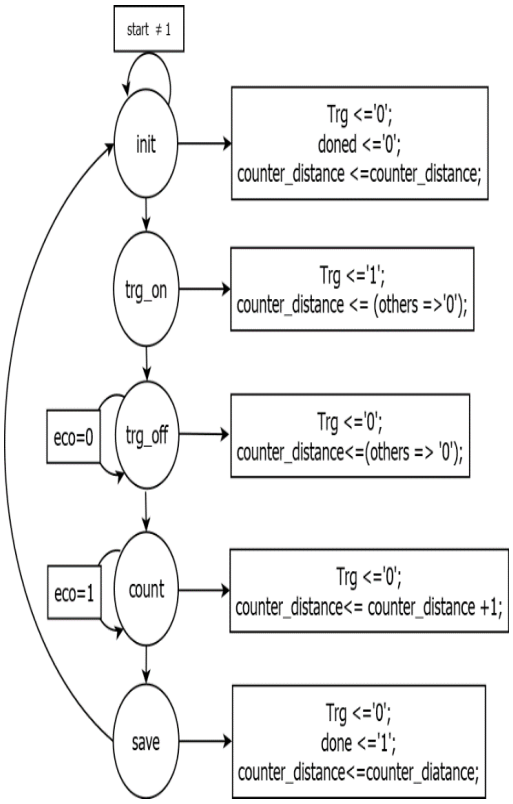

Figure 2. State Diagram Of The Ultrasonic Differential Sensor Driver.

Initially the ports and input and output signals are characterized, being "reset", "start" and "echo", input ports in charge of restarting the system, indicating the sending of the ultrasonic wave of the sensor and indicating the return of the same, respectively.

The "Trg" and "done" ports are output ports. "Trg" is the emission wave of the sensor and "done" is a signal to end the process.

```
entity HCsr_04 is
    Port
( clk, reset, eco, start :
in  STD_LOGIC;
  Trg, done: out STD_LOGIC;
 Distance: out STD_LOGIC_VECTOR
                    (7 downto 0));
end HCsr_04;
architecture Behavioral of HCsr_04 is
type state_type is
(init,trg_on,trg_off,count,save);
signal state: state_type;
signal counter_distance :
STD_LOGIC_VECTOR (7 downto 0);
begin
process(clk,reset)
```

begin

Every time that a rising edge is read in reset, the state "init" must be executed and in addition, drop flanks are sent in the ports Trg, where all the positions of the distance vector counter_distance are sent, in order to initialize the data of the ports, before starting a new reading of the sensors. If a falling edge is read in reset, which implies that the process has not been restarted, then it is sent directly to the start state "init".

```
if reset = '1' then
       state <= init;
       Trg <= '0'; done <= '0';
       counter_distance <= (others =>
'0');

elsif rising_edge(clk) then
       case state is
```

The "init" state sends a falling edge to the ports Trg and done, and preserves the value contained in the distance vector "counter_distance". Subsequently, it validates the value of the "start" port, which is responsible for starting the operation of the sensors, based on a change of state.

```
when init =>
       Trg <= '0'; done <= '0';
       counter_distance <=
counter_distance;

if start = '1' then
       state <= Trg_on;
else
       state <= init;
end if;
```

The state "trg_on" sends a rising edge to the "Trg" port, which indicates that the wave emission of the sensor was started, puts the positions of the distance vector in zeros and finally leads to the execution of the state " trg_off ".

```
when trg_on =>
       Trg <= '1'; done <= '0';
       counter_dist <= (others=> '0');
state <= trg_off;
```

The status "trg_off" changes the state of the "Trg" port, that is, turns off the sensor's emission signal, then validates the value of the "echo" port, which is responsible for checking the return of the wave emitted by the sensor. If the wave has not been

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reflected, it remains in the current state, otherwise the "count" state is achieved.

```
when trg_off =>
       Trg <= '0'; done <= '0';
       counter_dist <= (others=>'0');

              if eco = '0' then
                     state <= trg_off;
              else
                     state <= count;
              end if;
```

Due to the wave emitted by the sensor has been reflected once, the "count" state adds a unit of measurement to the distance vector, and continues counting the times this occurs, as long as the "echo" port indicates it, otherwise directs the process to the "save" state.

```
when count =>
Trg <= '0'; done <= '0';
counter_dist <= counter_dist+1;

       if eco = '1' then
              state <= count;
       else
                     state <= save;

       end if;
```

In "save", the "done" port receives a rising edge, which indicates finishing the process of sending, receiving and reading the sensors, and saves the value of the distance units traveled.

```
when save =>
       Trg <= '0'; done <= '1';
       counter_dist <= counter_dist;

       state <= init;

       end case;
end if;
end process;
```

At the end of the process, the information contained in the distance vector "counter_distance" is stored in Distance to use this information in the dessitema control.

```
Distance <= counter_dist;
end Behavioral;
```

2.2. PD CONTROLLER

Proportional Derivative Controller is a versatile control structure and highly recognized in the industry, applied in most artifacts with control systems. It is a feedback controller, which allows the error between the input signal and the output tend to disappear in time, and in turn, to predict the behavior of the system responses, through the proportional and derivative function respectively. This type of controller was chosen specifically, since it does not have a steady state error and with this simple controller an acceptable peak over time is achieved in response time appropriate for the system.

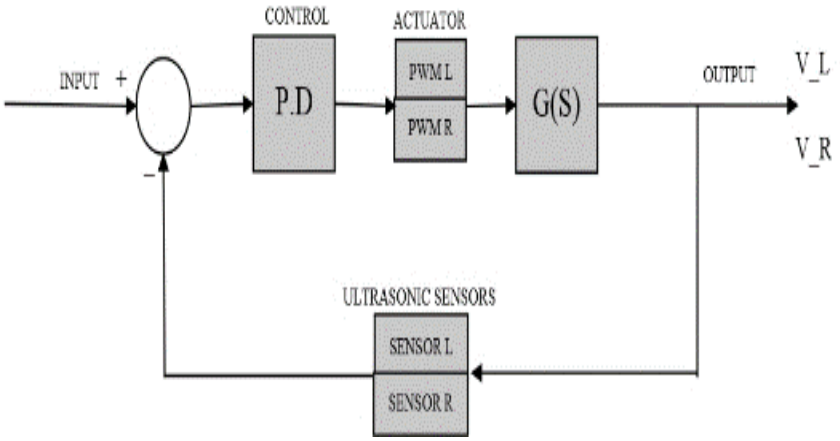

Figure 3. Block Diagram Of The System.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the previous diagram, the reference signal is a constant value of zero, which represents the objective of the system to maintain the position of the robot centered. The reference signal and the difference of the distances obtained by the sensors, which must be zero, are compared to determine if the difference between these data is negative, then the course must be corrected to the right and vice versa. The value of the subtraction of the sensors is sent to the control, where the actions of the PWM are defined by the digital control performed by the PD algorithm.

2.3. Cinematic Modeling Of The Differential Platform:

The variables of the model are described, from the structural diagram of figure 1, where L is the distance between the wheels V_R (t) and V_L (t), is the speed of the right and left wheel respectively, w (t) is the angular velocity of the robot and θ (t) is the angular position of the robot.

Initially, it is stated that the angular velocity depends on the difference between the speeds of the wheels at some instant of time, because of the distance between them.

$$w(t) = \frac{V_R(t) - V_L(t)}{L} \quad (1)$$

Deriving the angular position and equaling the angular velocity, it is obtained:

$$\frac{d\theta(t)}{dt} = \frac{1}{L}\left(V_R(t) - V_L(t)\right) \quad (2)$$

In the Laplace domain:

$$S\theta(s) = \frac{1}{L}\left(V_R(s) - V_L(s)\right) \quad (3)$$

The model for the right motor:

$$\frac{R_1(s)}{T_1(s)} = \frac{Km}{\tau s+1} \quad (4)$$

Where R_1 (s) is the angular output speed of the motor, T_1 (s) the motor voltage, Km is the motor constant, τ the motor Tao and r the wheel radius.

The speed of the right wheel is defined by:

$$V_1(t) = r * R_1(t) \quad (5)$$

Clearing the angular velocity of the motor output:

$$\frac{V_1(s)}{r} = R_1(s) \quad (6)$$

Where, $V_1(s)$ is the speed of the motor in the frequency domain.

Multiplying to 4 by the radius and replacing 4 in 1 you get:

$$\frac{V_1(s)}{T_1(s)} = \frac{Km*r}{\tau s+1} \quad (7)$$

In this way you will find the expression of the speed of the wheel as:

$$V_1(s) = T_1(s)\frac{Km*r}{\tau s+1} \quad (8)$$

Combining the two motors and assuming that they are equal in their mechanical characteristics, the following equation is proposed:

$$V_2(s) - V_1(s) = \left(T_1(s) - T_2(s)\right)\frac{Km*r}{\tau s+1} \quad (9)$$

taking into account the speed and tension of the system in the differential model:

$$\Delta V(s) = \Delta T(s)\frac{Km*r}{\tau s+1} \quad (10)$$

$$\frac{\Delta V(s)}{\Delta T(s)} = \frac{km*r}{\tau s+1} \quad (11)$$

Returning to the kinematic model, the equation that describes the movement in terms of the linear velocity and the steering angle of the robot is stated:

$$S\theta(s) = \frac{1}{L}\Delta V(s) \quad (12)$$

Finally, the transfer function is obtained in terms of the steering angle and the speed of the robot.

$$\frac{\theta(s)}{\Delta V(s)} = \frac{1}{sL} \rightarrow \frac{\theta(s)}{\Delta T(s)} = \frac{Km*r}{sL(\tau s+1)} \quad (13)$$

From the plan of movement of the robot

$$Sen(\theta) = \frac{\Delta Y}{\Delta X} \quad (14)$$

$$\Delta Y(t) = \Delta X(t) * Sen\theta(t) \quad (15)$$

If the speed in X is constant

$$\Delta Y(t) = KV * Sen\theta(t) \quad (16)$$

Approaching Senθ as $\frac{\theta*2\sqrt{2}}{\pi}$ to $0 \leq \theta \leq \frac{\pi}{4}$ that is to say angular between $0^0 y 45^0$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$Y(t) = KV * \theta(t)\left(\frac{2\sqrt{2}}{\pi}\right) \quad (17)$$

$$\frac{\Delta Y(s)}{\theta(s)} = \frac{2\sqrt{2}*KV}{\pi} \quad (18)$$

This statement can be made, since it is shown graphically that the sinusoidal function is approximately linear in this range.

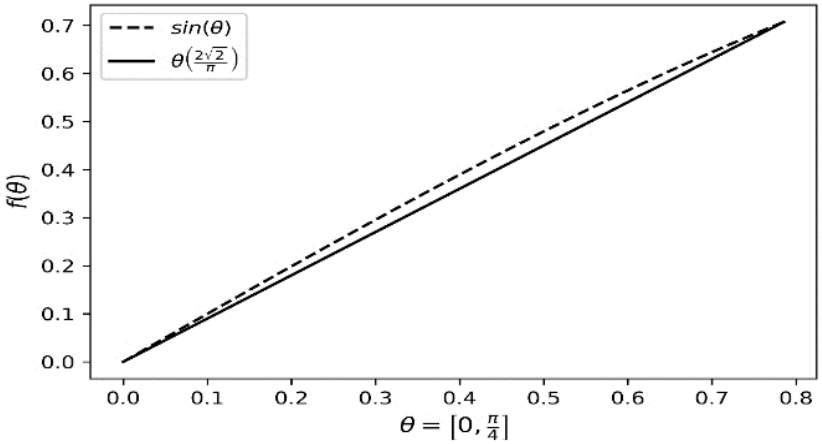

Figure 4. Linear Approximation Of The Sinus Function Between 0º And 45º.

Now multiplying the transfer functions:

$$\frac{\frac{\theta(s)}{\Delta T(s)}*\Delta Y(s)}{\theta(s)} = \frac{Km*r}{sL(\tau s+1)} \quad (19)$$

$$\frac{\Delta Y(s)}{\Delta T(s)} = \frac{2\sqrt{2}Km*KV*r}{\pi*sL*(\tau s+1)} \quad (20)$$

$$\frac{\Delta Y(s)}{\Delta T(s)} = \frac{\frac{2\sqrt{2}Km*KV*r}{\pi L}}{s(\tau s+1)} \quad (21)$$

Finally, we obtain a transfer function with constant terms in the numerator and a second order in the denominator.

$$\frac{\Delta Y(s)}{\Delta T(s)} = \frac{\frac{2\sqrt{2}Km*KV*r}{\tau\pi L}}{s\left(s+\frac{1}{\tau}\right)} \quad (22)$$

3. IMPLEMENTATION AND RESULTS

After having fully understood the functioning of the System together with the sensor driver and the complete model of the plant, we proceed to carry out a series of tests and simulations of the System in open loop, in closed loop, the design of the controller and its respective tests in closed loop; in addition to the design of the digital controller and its respective implementation in VHDL.

Response in open loop of the system (Figure 5)If the system is in open loop and starts up, the robot will be in a constant movement that has an error that tends to infinity, which is why the system is unstable. This behavior can be evidenced below:

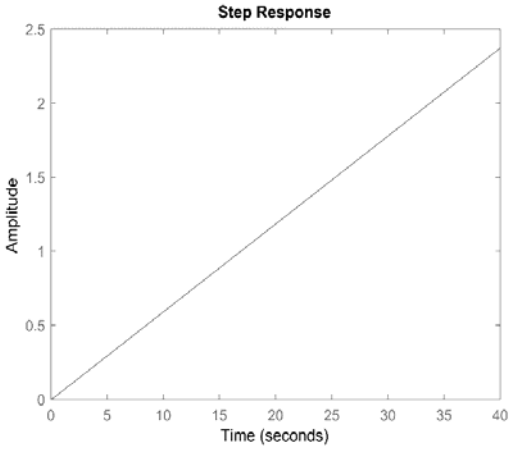

Figure 5. Open Loop Step Response Of The System.

Closed loop response only with a proportional gain

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

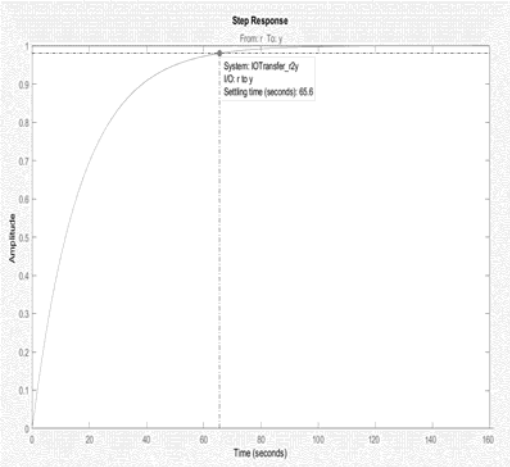

Figure 6. Closed Loop Step Response.

When only a proportional controller is used the system easily reaches the stability point, but it cannot be adequately controlled due to it presents a stabilization time of approximately 60 seconds, which makes this kind of controller non-viable for controling this kind of plant.

Design of the PD controller using the Root locus technique (figure 7)

The Root Locus technique was used as a design tool to determine the points for which the PD controller offers a response that meets the specifications of the system.

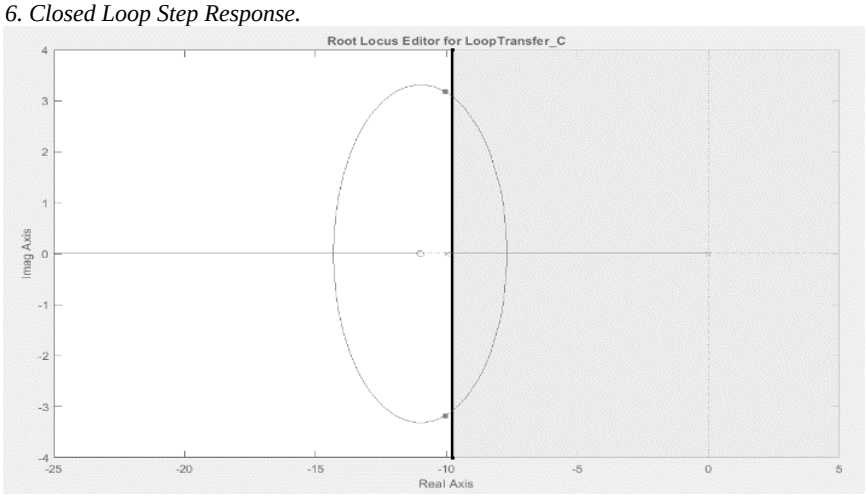

Figure 7. Root Locus.

The points are chosen for the poles of the system, as close as possible to the boundary line and within the blank area of Figure 7, which is defined by the speed of response of the system and the time needed to capture the data of distance read by the sensors, which is close to 0.4 seconds. It was determined that the values for the system are given by equation 23.

$$G(s) = 17s + 187 \qquad (23)$$

Response to the closed loop system step using the PD controller

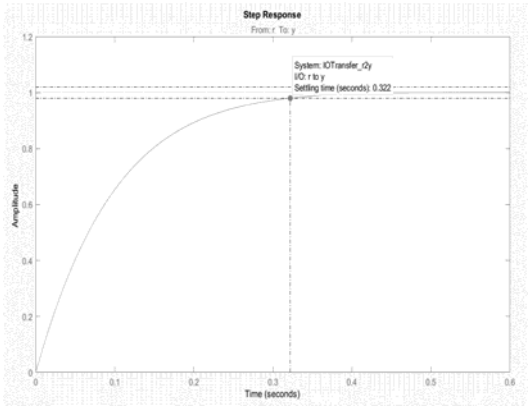

Figure 8. Closed Loop Response With PD Controller.

It is seen that the system after closing the loop with the controller, has an acceptable response time taking into account the physical conditions of the plant and the speed of response of the sensors, in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

addition to this the system does not present on peak, but has a behavior critically damped thanks to the kindness of the controller type, has no steady state error.

It is necessary to determine the time it takes the waves emitted by the sensors (time of flight), to go to the walls and back, taking into account that the maximum distance is 2 meters for each sensor. From the equation 24 the sampling time is cleared, taking into account that v is the speed of sound (342 m / s approximately), d is the total sampling distance (2 meters round trip for each sensor) and t the flight time.

$$v = \frac{d}{t} \quad (24)$$

$$t = \frac{d}{v} = \frac{8m}{342\frac{m}{s}} = 23{,}4ms \quad (25)$$

From this time the sampling time approaching 40Hz is determined, which is used to give a design parameter of the controller, with a desired stabilization time of 0.4 seconds.

$$t_m = \frac{1}{23{,}4ms} = 42{,}73Hz \quad (26)$$

Since in the real system the response to the step does not reflect a behavior that can explain the dynamics of the system, tests were performed using the delta Dirac signal or impulse response, which is more approximate to the type of signals used by the controller; the previous thing is reflected in figure 9 where it can be observed that the system in a long time has a stationary error of zero.

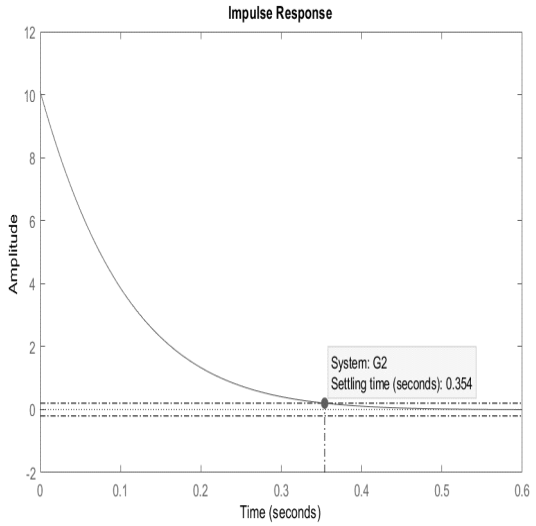

Figure 9. Impulse Response.

It can be seen how after applying the impulse signal to the controller, a satisfactory response is generated in less than 0.4 seconds, which is the response speed of the system.

Design of a digital controller in hardware:
From the constants calculated in equation 23 and using the Tustin technique, a digitalization of the control system is carried out, going from the S to Z plan. This response is evidenced in equation 27

$$C(z) = \frac{1547z+1173}{z+1} = \frac{O(z)}{I(z)} \quad (27)$$

By performing the respective clearings, the difference equation is obtained in terms of the input and output of the System.

$$O(z)z + O(z) = (1547zI(z)) + (1173I(z)) \quad (28)$$

The equation is expressed in such a way that it is implementable in a digital system, leaving the equation in terms of the previous values of the input and output, so that an equation in differences is obtained.

$$O(z)z^{-1} + O(z) = 1173I(z)z^{-1} + 1547I(z) \quad (29)$$

$$O(k-1) + O(k) = 1173i(k-1) + 1547ik \quad (30)$$
$$O(k) = 1547ik + 1173i(k-1) - O(k-1) \quad (31)$$

The Tustin method was used for the discretization of the PD which allows the discretization of improper transfer functions, that is, with a numerator order higher than the denominator, such as the PD and PID controllers.

3.1. FPGA Implementation

The description of the controller was done in a single code that performs the necessary calculations in parallel, all this done in a high-level hardware description language such as VHDL, below there are some sections of which hardware description shows how the description concurrently performs the equation in differences of the controller. Besides, this is included in the block diagram to see the block performing the reading of ultrasonic sensors along with the block that generates the output of PWM for each of the motors used for locomotion.

First of all, the source code of the hardware description of each one of the PWM's that control the motors of the autonomous mobile is shown below.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
if reset = '1' then
       sawtooth <= (others => '0');
       PWM_out <= '0';
elsif rising_edge(clk) then
       sawtooth <= sierra + 1;
       if sawtooth <  Duty then
              PWM_out <= '1';
       else
              PWM_out <= '0';
       end if;
end if;
```

The only additional condition is that the resolution of this PWM depends on two factors, the first one has to do with the number of bits delivered by the controller and the second one is the working frequency of the motors, for the latter a frequency divider is required to achieve the output frequency of the driver.

It was decided to make a description of the controller by state variable, below there are the most relevant code structures of the hardware description in the VHDL language of the PD controller.

The inputs and outputs of the system are declared taking into account the input sizes from the distance sensors and the output for the motors.

```
entity PDControler is
Port (
data_in_0    : in std_logic_vector
                    (12 downto 0);
data_out_0: out  std_logic_vector
                    (12 downto 0);
reset        : in   STD_LOGIC;
clk,ce       : in   STD_LOGIC);
end    PDControler;
```

The constants are placed in hexadecimal that are the result of the design by root locus technique.

```
architecture Behavioral ofPDControler
       is
constant A:std_logic_vector
       (15 downto 0) :=    (X"7ffc");
constant B:std_logic_vector
       (15 downto 0) :=    (X"0e61");
constant C:std_logic_vector
       (15 downto 0) :=    (X"0e61");
constant D:std_logic_vector
       (15 downto 0) :=    (X"7d71");
signal U  :std_logic_vector
       (22 downto 0);
```

Updating of some signals depending on the sampling frequency of the controller.

```
process (Xptrun,clk,reset)
begin
       if reset = '1' then
              VX <= (others => '0');
       elsif clk'event and clk = '1'
then
              VX <= Xptrun;
       end if;
end process;
```

Variables that allow the controller to be calculated concurrently.

```
process(VX,U)
variable AX : std_logic_vector
                    (38 downto 0);
variable BU : std_logic_vector
                    (38 downto 0);
variable CX : std_logic_vector
                    (38 downto 0);
variable DU : std_logic_vector
                    (38 downto 0);
variable prodA : std_logic_vector
                    (38 downto 0);
variable prodB : std_logic_vector
                    (38 downto 0);
variable prodC : std_logic_vector
                    (38 downto 0);
variable prodD : std_logic_vector
                    (38 downto 0);
```

Finally, everything can be summed up as simply calculating the controller in a parallel matrix manner by means of the status variables with which the system output was calculated.

```
begin
       AX := A*VX;
       BU := B*U;
       CX := C*VX;
       DU := D*U;
       Xp   <=  AX + BU;
       Y    <=  CX + DU;
end process;
```

Below there is a block diagram generated by the ISE Xilinx Project Navigator graphical tool:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

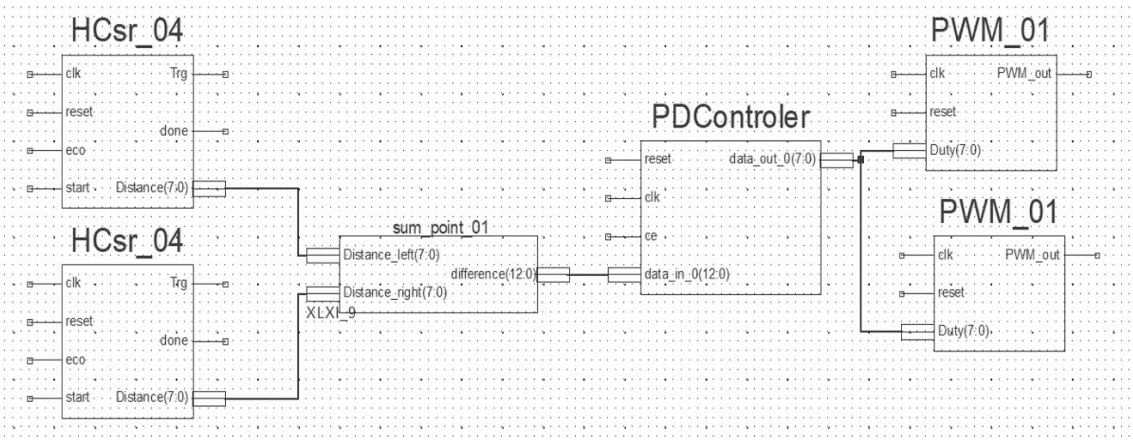

Figure 10. General Block Diagram Generated By The Graphic Tool.

4. CONCLUSIONS

A SISO model of the robot movement of only second order was obtained, which given the complexity of the system (it includes the electrical and mechanical elements of the robot, as well as its kinematic model) is quite reduced, also facilitating the calculation of the controller. This was achieved by treating both ultrasonic sensors as one, making the system input the difference in distances between the two, and the output of the system is the difference between the angular speeds of both motors instead of the angular speeds separately.

Calculating externally to the system the difference between the sensors and externally generating the speed of each motor from the calculated speed difference the system was reduced to one of SISO type.

Using Root Locus as a tuning technique for a PD controller achieves highly acceptable results quickly that can be easily exported to a digital controller easily implemented in a number of digital devices on the market, not only in PLD but any device type microcontroller or system with embedded linux.

Due to the simplicity of the SISO model obtained, it was possible to obtain a simple PD controller that satisfies the controlling requirements of the differential robot, accomplishing the task of moving through, maintaining itself in the middle line between two walls or obstacles.

The implementation of the sensor reading as the PD controller over the FPGA only uses 3% of the resources of a Spartan 3AN and to be done in a distributed and concurrent way allows the controller to have a high performance with minimal use of the hardware resource available in this type of devices, which allows this controller is the basis of a more complex application that would have the ability to perform a number of possible navigation applications in autonomous mobile robots.

5. DISCUSSION

This article shows a reduced mathematical modeling, which can be applied to any differential platform to control the speed and therefore the dynamic position of this autonomous mobile robot using different types of sensors. Although for this work the digital control was carried out with a programmable logic device type FPGA, it could be exported, modified or adapted to a microcontroller type device or a system running embedded Linux using any digital control technique that suits the requirements of the future application to be carried out.

6. ACKNOWLEDGMENT

This work was supported by the Universidad Distrital Francisco José de Caldas Technological Faculty. The views expressed in this paper are not necessarily endorsed by the University. The authors thank the research group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

[1] A. Gencer, “A new speed/position control technique for travelling wave ultrasonic motor under different load conditions,” *16th Int. Power Electron. Motion Control Conf. Expo. PEMC 2014*, pp. 65–70, 2014.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] E. K. T. Kien, M. Shanmugavel, and S. Veera Ragavan, "Motion planning of a bipedal walking robot with leg-mounted ultrasonic sensors-An experimental study," *Int. Conf. Robot. Autom. Humanit. Appl. RAHA 2016 - Conf. Proc.*, pp. 1–6, 2017.

[3] J. J. Wang, "Position and speed tracking control of inverted pendulum based on double PID controllers," *Chinese Control Conf. CCC*, vol. 2015–Septe, pp. 4197–4201, 2015.

[4] L. Xu and Z. Q. Liu, "Design of fuzzy PID controller for ship dynamic positioning," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 3130–3135, 2016.

[5] T. A. Tran, X. Yan, and Y. Yuan, "Marine engine rotational speed control automatic system based on Fuzzy PID logic controller," *2017 4th Int. Conf. Transp. Inf. Safety, ICTIS 2017 - Proc.*, pp. 1099–1104, 2017.

[6] Z. Yang, J. Du, and S. Zhou, "Position control for filter rod splint based on fuzzy-PID," *2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016*, pp. 1435–1439, 2016.

[7] T. R. D. Kumar and S. J. Mija, "Dynamic SMC control scheme with adaptively tuned PID controller for speed control of DC motor," *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2015–June, no. June, pp. 187–191, 2015.

[8] S. Pothorajoo and H. Daniyal, "PID bidirectional speed controller for BLDC with seamless speed reversal using Direct Commutation Switching Scheme," *2017 IEEE 8th Control Syst. Grad. Res. Colloquium, ICSGRC 2017 - Proc.*, no. August, pp. 7–12, 2017.

[9] R. Klambauer and A. Bergmann, "A new principle for an ultrasonic flow sensor for harsh environment," *Proc. IEEE Sensors*, vol. 2017–Decem, pp. 1–3, 2017.

[10] S. Shin, M. H. Kim, and S. B. Choi, "Improving efficiency of ultrasonic distance sensors using pulse interval modulation," *Proc. IEEE Sensors*, pp. 1–3, 2017.

[11] B. Chu, "Mobile robot position control algorithm based on multiple ultrasonic distance sensors," *ICCAS 2015 - 2015 15th Int. Conf. Control. Autom. Syst. Proc.*, no. Iccas, pp. 1238–1240, 2015.

[12] G. Rus *et al.*, "Torsion ultrasonic sensor for tissue mechanical characterization," *IEEE Int. Ultrason. Symp. IUS*, vol. 2016–Novem, pp. 11–14, 2016.

[13] H. Song, J. Popovics, and J. Park, "Contactless ultrasonic wavefield imaging of concrete elements using an automated scanning MEMS ultrasonic sensor array," *IEEE Int. Ultrason. Symp. IUS*, pp. 1–3, 2017.

[14] S. F. Toha, H. M. Yusof, M. F. Razali, and A. H. A. Halim, "Intelligent path guidance robot for blind person assistance," *2015 4th Int. Conf. Informatics, Electron. Vision, ICIEV 2015*, pp. 1–5, 2015.

[15] M. Njah and M. Jallouli, "Wheelchair obstacle avoidance based on fuzzy controller and ultrasonic sensors," *Int. Conf. Comput. Appl. Technol. ICCAT 2013*, pp. 1–5, 2013.