**Documentación Avance Proyecto 2: Seguidor de Línea**

**Presentado por: Andrés Felipe Ardila- Diego Cárdenas- Maycol Sabogal**

Un Robot seguidor de línea es aquel diseñado con una única misión. Seguir una trayectoria marcada con una línea, la cual debe diferenciarse del entorno, el microcontrolador ATMEGA-328p del Arduino Uno, será el encargado de enviarle las debidas funciones para que el robot se comporte de manera correcta.

En la implementación del seguidor de línea es primordial en su funcionamiento el uso del control PID, en este caso será análogo. Este control permite un desarrollo más óptimo del sistema.

***Control PID***

El control PID es un mecanismo de control que a través de un lazo de retroalimentación que en este caso permite regular la velocidad de los motores. Este control tiene tres parámetros fundamentales. Ganancia proporcional (P), integral (I) y derivativa (D).

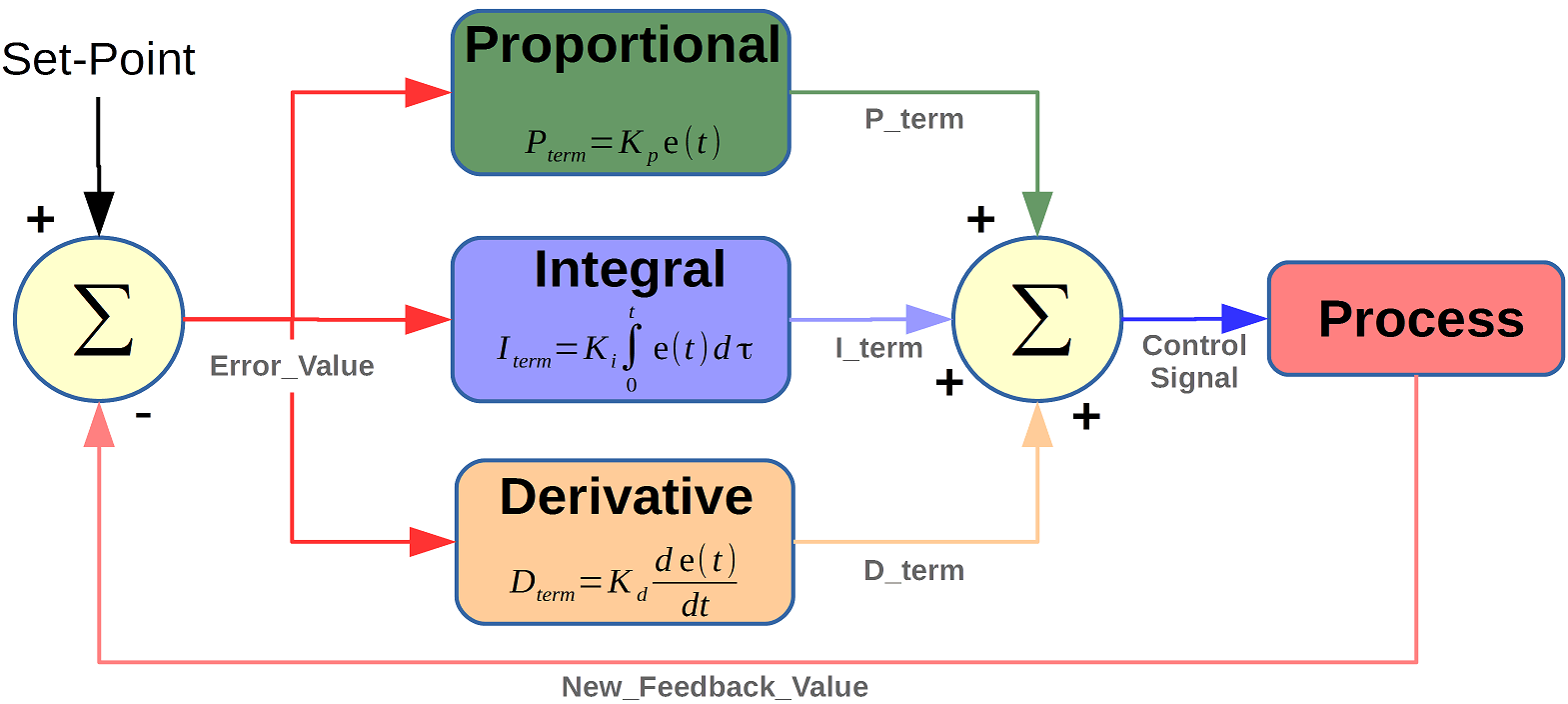


Figura 1. Diagrama del control PID

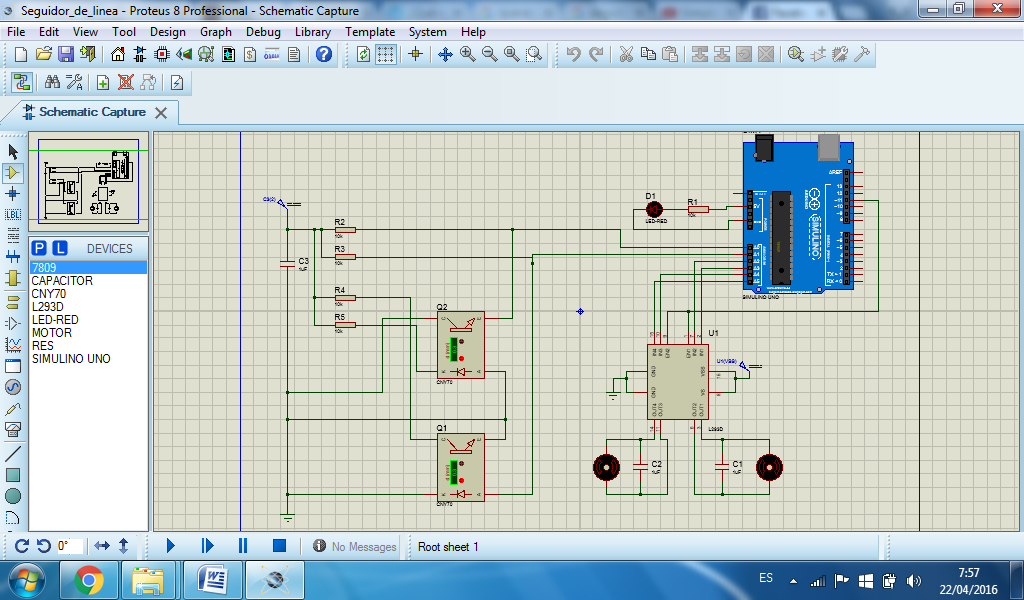
Acción proporcional: La acción proporcional mide la diferencia entre el valor actual y el set point, y aplica el cambio.

Acción integral: La función principal de la función integral es asegurar que la salida del proceso concuerde con la referencia del estado estacionario, es decir, el tiempo que se toma para llevar a cabo la acción correctiva.

Acción derivativa: El propósito de la acción derivativa es mejorar la estabilidad del lazo cerrado, es decir, emite una acción predictiva prevé el error e inicia una acción oportuna.

En el seguidor de línea se planea implementar un control PID digital, y por medio de un potenciómetro variar los parámetros de entrada de ese control PID digital

***Programación***



**Figura.2 simulación del seguidor de línea**

Por medio del uso del microcontrolador arduino 1 se planea implementar un programa para la lectura del sensor del seguidor de línea además de la dirección de los motores como todavía no se tienen los motores ni los sensores solo se ha hecho una base de dicho código ya que no se ha probado del todo y cuadrarlo

#include <QTRSensors.h> // libreria necesaria para el funcionamiento del sensor con arduino uno

float vproporcional;/variables del PID

float vderivativo;

float vintegrativo;

float posicion;

float ul\_posicion=0;

void setup()

{

pinMode(9, INPUT);

pinMode(8, INPUT);

pinMode(7,INPUT);

vproporcional = digitalRead(7);

vderivativo = digitalRead(8);

vintegrativo = digitalRead(9);

posicion = analogRead(10);

// Calculos PID

vderivativo = posicion – ul\_posicion;

integral= posicion;

ul\_posicion = posicion;

int d\_voltaje = posicion/vproporcional + integral\*0 + derivativo\*vderivativo;

if (d\_voltaje> maximum)

d\_voltaje = maximum;

if (d\_voltaje < -maximum)

d\_voltaje = -maximum;

if (d\_voltaje < 0)

OrangutanMotors::setSpeeds(maximum, maximum + power\_difference);

else

digitalWrite(HIGH);

};