



**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL**

**INTERDISCIPLINARIA DE INGENIERÍA**

**CAMPUS ZACATECAS**

**UPIIZ**

Ingeniería Mecatrónica

Trabajo Terminal II

**“Diseño, Construcción e Implementación de un  
Control PID Digital a una Plataforma Stewart-Gough”**

Presentan:

José Ronaldo Díaz Paredes

Miguel Abraham Gallardo Carreón

Asesores:

M. en C. Fernando Olivera Domingo.

M. en C. Flabio Darío Mirelez Delgado.

M. en I. E. Adán Orenday Delgado.



Zacatecas, Zac. Viernes 8 de enero de 2021.



# INSTITUTO POLITÉCNICO NACIONAL

## UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA CAMPUS ZACATECAS

UPIIZ

Ingeniería Mecatrónica

Trabajo Terminal I

José Ronaldo Díaz Paredes

Alumno 1

Miguel Abraham Gallardo Carreón

Alumno 2

Mtr. Fernando Olivera Domingo

Asesor 1

M. en I. E. Adán Orenday Delgado

Asesor 2

M. en C. Flávio Darío Mirelez Delgado

Asesor 3



Ing. Rafael Reveles Martínez

Presidente del Jurado

Dr. Sergio Domínguez Sánchez

Profesor Titular

Zacatecas, Zac. Viernes 8 de enero de 2021.

# *José Ronaldo Díaz Paredes*

## **Dedicatoria:**

*Para mi padrino Porfirio Barraza y mi abuelo Vicente Paredes que a pesar de no tener algún grado de ingeniería me siguen enseñando algo nuevo cada día, una manera diferente de hacer las cosas y una solución que siempre existe, sin ellos no sé si esto sería posible.*

## **Agradecimientos**

*A mi familia, que a pesar de las limitantes y dificultades que pudieran presentarse en el camino nunca me dejaron sin un lugar al cual llamar hogar. A mi madre Leticia Berenice Paredes Martínez que con su esfuerzo y carácter me enseñó a no conformarme y a saber que “quizás” si puedo, que “quizás” si me esfuerzo puedo hacer algo diferente, que no tengo porque compararme ni sentirme menos que nadie. Te prometí algún día que tu hijo no sería un vago más y créeme, me estoy esforzando cada día. Padre, que no siempre tuvimos lo que quisimos, pero si lo que necesitamos, te agradezco por cada cosa que me diste y que no, ya que por eso estoy donde hoy estoy y se valorar las cosas sin dejar de pensar que solo se vive una vez.*

*A mis abuelos María Luisa Martínez y Vicente Paredes que han sido los padres más amorosos que pude tener, cada uno con su forma de ser me hicieron sentir amado y protegido. Mi abuela y sus deliciosos platillos, sus pláticas tan cálidas y tantos recuerdos de cuando niño y cuanto amor me dieron. Mi abuelo y sus enseñanzas tan firmes que sin ellas no sé si estaría aquí, es de quien aprendí tanto y de quien nació en mí la curiosidad por este mundo. A mis abuelos Francisca Gallegos y José Díaz que me protegieron siempre, más aún cuando salía en chanclas corriendo porque mi mamá me quería pegar y me escondía en el regazo de mi abuela, vaya sensación tan cálida. Y a mi abuelo José que siempre tenía buen café e historias para contar.*

*A mis tíos Blanca Estela Paredes y Porfirio Barraza que me han enseñado tanto. Mi tía siempre tenía una mesa que, con más de diez personas en ella nunca faltó un plato de comida, que siempre nos hizo ver a cada uno de sus sobrinos que teníamos donde llegar en cualquier momento. Mi padrino Porfirio, no tengo palabras para expresar lo agradecido que estoy pues si él no sé si estaría aquí ya que hasta la fecha me ayudó a resolver cosas en que no lograba ver una solución. También quiero agradecer a cada una de mis tíos de la familia Díaz que a pesar de ser yo un estudiante más me hicieron sentir a cada paso que era especial, que siempre destaque a pesar de no ser así.*

*A mi actual y más grande confidente Gabriela Isiordia Bautista por estar en mis mejores y peores momentos, por convertirse en la persona que más me conoce y con quien puedo sentir que si me quebró mis pedazos no se los llevará el viento. Por ser el regazo donde puedo descansar del mundo, un lugar cálido donde me siento seguro y que me ha visto tan frágil como un copo de nieve, y a pesar de eso sigue creyendo en mí como si fuera el guerrero más solidó parado al frente de la batalla. Porque, aunque no siempre hay un sol al final de la tormenta siempre es agradable sentir que tienes con quien compartir la lluvia.*

*A mi compañero, amigo y casi hermano Miguel Abraham que literalmente sin él esto no sería posible ya que ocupamos su firma. Pero sobre todo porque a pesar de no tener lazos de sangre que nos unan simplemente me dio la bienvenida en su hogar, con su familia de la cual me hizo sentir parte. Muchas gracias a toda la familia Carreón porque a pesar de no conocerme ni saber de dónde vengo nunca me hicieron sentir un desconocido y me recibieron a puertas abiertas dentro de su familia. A ver, llora.*

*A mis amigos y compañeros de esta tragicomedia llamada Ing. Mecatrónica que sin apoyarnos unos a otros (porque yo también apoyé, agradézcanme) las cosas quizás no habrían llegado hasta hoy. Por ahí el señor Claudio Iván que suba al podio de este párrafo, fue quien me contagió la pasión por la investigación y el desarrollo de proyectos; Eduardo Chairez nos mostró habilidades más allá de calcular circuitos y programar calculadoras de circuitos; Carlos Martínez, que ante las noches de júbilo siempre nos brindó asilo y una conversación de esas que quedan en el recuerdo. Y una dedicatoria especial para Juan Alberto Guzmán, donde quiera que estés muchas gracias por enseñarme que en la vida hay cosas mucho más importantes que una aprobación, un examen, un título o un reconocimiento.*

*Finalmente quiero agradecer a cada uno de los docentes y técnicos que estuvieron presente y formaron parte tanto de este proyecto, nuestro desarrollo académico y nuestras vidas. A Chuy y Samir que nos ayudaron con cada manufactura y pusieron empeño en ello como si fuese su proyecto; a Tenorio, que se quedaba hasta noche (a pesar de ser su horario laboral) para orientarnos y terminar cada parte del prototipo electrónico. Y un agradecimiento especial para nuestros asesores Flabio, Adán y Fernando por acompañarnos y apoyarnos de sobremanera en cada etapa de este proyecto, que siempre estuvieron al pendiente aún con la situación tan difícil que actualmente nos afecta.*

# *Miguel Abraham Gallardo Carreón*

## **Dedicatoria:**

*Dedico este trabajo a los que estuvieron, a los que están y a los que estarán.*

## **Agradecimientos**

### **A Dios**

*Por darme la fuerza y el apoyo que necesité en todo momento a lo largo de la carrera, porque siempre estuvo presente en todas las adversidades, me cuidó y me llenó de bendiciones.*

### **A mis padres.**

*Mamá has hecho un excelente trabajo, gracias a ti puedo decir con toda la certeza que soy una persona de bien, pues me has enseñado a ser justo, firme y noble; debo decirte que te amo y admiro por tanta valentía y responsabilidad que siempre demuestras; que a pesar de que estés triste, cansada o harta de las situaciones que se atraviesan siempre sigues luchando y tomando las riendas de todo lo que te toca. Aún recuerdo cuando me despedía de ti con lágrimas cuando me dejabas en el kínder y a ti también se te enjugaban los ojos, ahora sabemos que esas lágrimas no fueron en vano y que, efectivamente, fueron por mi bien pues. Gracias a ti terminé la carrera. Me da tanta que seas mi mamá, siempre supiste lo que fue bueno para mí y te doy las gracias por tanto que me has dado y enseñado. Eres luz en mi vida y te amo por la persona que eres. Te amo.*

*Papá, desde que tengo memoria siempre recuerdo haberte contado las cosas que me apasionan, lo que me gusta e incluso lo que quiero hacer, y no hubo ni una sola vez en la que me dijeras un “no”, me hicieras sentir mal o me desalentaras. Al contrario, siempre me apoyaste y confiaste en todo lo que deseé y soñé. Y bueno, son varias las razones por las que te admiro, las primeras que mencionaré son la nobleza y humildad que imprimes ante cualquier situación; las segundas son la astucia, paciencia y tu enorme inteligencia para resolver un problema; la tercera y no menos importante es el amor con el que haces las cosas pues no veo nada de maldad en ti, ni creo que tengas malos deseos hacia los demás; te agradezco por enseñarme tanto de todo. Pero con lo que más me quedo es que me dejas muy claro con tu existencia que sí existe la gente bondadosa y con un corazón enorme dispuesto a dar todo por las personas a las que quieras.*

*Yo sé que ambos hicieron muchas cosas a un lado porque yo siempre tuviera un plato de comida en la mesa, un regalo de Navidad, mucho amor y una infinidad de cosas más. Mi*

carrera no hubiera sido posible sin ustedes ¡Lo logramos! Ojalá algún día pueda devolverles todo lo que hicieron y aún hacen por mí. Por último, quiero decirles GRACIAS INFINITAS. Los quiere, su hijo.

### **A mi hermana**

Ximenita, debo confesarte que estoy muy sorprendido con la relación que tenemos. Jamás había tenido una conexión tan fuerte con alguien como contigo y si a veces soy enojón es porque no sé cómo canalizar tanto amor que siento por ti hermanita, pero tienes que saber que te quiero mucho. Sé que soy tu ejemplo a seguir (porque eres una copia mía exactita) y que piensas que yo te enseño muchas cosas, pero la verdad es que tú me enseñas mucho más de lo que yo a ti, aunque no lo creas, tú eres mi mejor maestra. Puedes reparar todo malestar emocional que yo pueda tener con un simple abrazo. Gracias por todo chatita, cuenta conmigo siempre, te amo Xiri.

### **A mi familia**

Que siempre me apoyó en todo momento y que a pesar de que hubiera dificultades en casa hicieron honor a la palabra “familia” pues siempre han estado al tanto de mí, queriéndome como a su propio hijo y demostrando su amor incondicional.

### **A mi compañero de TT**

Mi compañero, hermano, confidente e ingeniero Ronaldo Díaz ¿Recuerdas el día que hablamos quienes eran nuestros ídolos como mecatrónicos? Tú mencionaste al “Sombrerete”, yo mencioné a Eduardo y conforme fluyó la conversación ya teníamos una larga lista de compañeros; lo curioso es que en ese momento me faltó mencionarte a ti amigo y ahora me da una inmensa alegría el tener la oportunidad de decírtelo ahora.

Existen un millón de razones por las que una persona puede enfrentar una crisis y tropezar: emocionales, económicas, familiares, amorosas, lavar un baño en vez de estudiar, sentirse traicionado por quien menos te lo esperas... y qué flojera mencionártelas todas; pero tú, caída tras caída te has sabido levantar hermano y has demostrado lo fuerte que eres y activas tu modo cholo. Eres una persona excepcional y solo me queda agradecerte todo el apoyo y aprendizaje que me has brindado amigo, te estimo demasiado y deseo mucho éxito en esta nueva etapa de la vida que sigue. Un fuerte abrazo, te estimo carnal, eres el mejor. Siga siendo curioso siempre ingeniero 😊.

“Aquí somos lo que hacemos”.

### **A mis amigos**

*Un fuerte abrazo a todas las personas que conocí y que me alentaron a seguir adelante en esta etapa universitaria, son tantos que no acabaría de mencionarlos. En especial agradezco a Chairez y Charly, mis hermanos no biológicos que son responsables de tantos momentos que quedarán en mi memoria, así como de su honestidad y sincera amistad.*

### **A mis profesores**

*Quienes siempre mostraron dedicación y esfuerzo. A los técnicos de laboratorio pesados 1. Agradecimientos especiales a Toño Tenorio, Fernando Olivera, Adán Orenday, Flabio Mirelez y Uma Hernández a quienes considero mis amigos y les agradezco su apoyo en todo momento y confianza.*

### **A mi alma máter**

*A la UPIIZ, por ofrecer siempre el apoyo a cualquier estudiante que lo solicite. Me da mucho orgullo decir que soy politécnico y siempre lo llevaré en el corazón donde quiera que vaya.*

*Gracias a todos los que hicieron esto posible.*

**“LA TÉCNICA AL SERVICIO DE LA PATRIA. HUÉLUM GLORIA”**

**“APRENDER Y COMPARTIR”**

## INDICE

1	RESUMEN .....	9
2	PALABRAS CLAVE.....	9
3	ABSTRACT .....	10
4	KEYWORDS .....	10
5	INTRODUCCIÓN .....	11
<b>5.1</b>	<b>Objetivos .....</b>	<b>12</b>
5.1.1	Objetivo General: .....	12
5.1.2	Objetivos específicos:.....	12
6	JUSTIFICACIÓN.....	12
7	MARCO TEÓRICO.....	14
<b>7.1</b>	<b>Historia de los robots paralelos .....</b>	<b>14</b>
<b>7.2</b>	<b>Cinemática Inversa .....</b>	<b>16</b>
7.2.1	Fundamentos matemáticos.....	16
<b>7.3</b>	<b>Función de transferencia .....</b>	<b>17</b>
7.3.1	Sistema de primer orden.....	17
7.3.2	Tiempo de respuesta.....	17
<b>7.4</b>	<b>Control PID (Proporcional-Integral-Derivativo).....</b>	<b>17</b>
7.4.1	Transformada Z .....	19
7.4.2	La Transformada de Laplace y la Transformada Z.....	19
7.4.3	Retenedor de Orden Cero .....	20
7.4.4	PID Digital por Retenedor de Orden Cero (ZOH).....	20
<b>7.5</b>	<b>Sensores de Posición y Orientación .....</b>	<b>21</b>
7.5.1	Sensor Panel Resistivo.....	21
7.5.2	Sensor Giroscopio MPU-6050 .....	22
8	ANTECEDENTES .....	23
9	ESTADO DEL ARTE. .....	24
10	PLANTEAMIENTO DEL PROBLEMA .....	26
11	DESARROLLO DEL TRABAJO.....	28
<b>11.1</b>	<b>Selección del Microcontrolador.....</b>	<b>28</b>
11.1.1	Definición del tiempo de muestreo.....	28

<b>11.2</b>	<b>Modelo matemático .....</b>	<b>32</b>
11.2.1	Grados de libertad de la Plataforma Stewart-Gough.....	34
11.2.2	Espacio de trabajo de la <i>Plataforma Stewart-Gough</i> .....	34
11.2.3	Definición de medidas para un espacio de trabajo deseado. ....	36
11.2.4	Cinemática inversa. ....	39
<b>11.3</b>	<b>Simulaciones .....</b>	<b>45</b>
11.3.1	Simulación del microcontrolador .....	45
11.3.2	Simulación de movimientos y articulaciones .....	47
11.3.3	Simulación de fuerzas externas.....	52
<b>11.4</b>	<b>Selección de componentes.....</b>	<b>55</b>
11.4.1	Microcontrolador .....	55
11.4.2	Sensores .....	56
11.4.3	Actuadores .....	58
11.4.4	Piernas y rotulas .....	59
<b>11.5</b>	<b>Definición de características .....</b>	<b>60</b>
<b>11.6</b>	<b>Diseño Electrónico .....</b>	<b>61</b>
<b>11.7</b>	<b>Construcción del Prototipo.....</b>	<b>63</b>
11.7.1	Plan de Manufactura .....	63
11.7.2	Manufactura de Piezas de Acrílico .....	64
11.7.3	Modificación de Juntas Esféricas.....	65
11.7.4	Ensamble final .....	65
<b>11.8</b>	<b>Implementación de Sensores .....</b>	<b>66</b>
11.8.1	Panel Resistivo.....	66
11.8.2	Giroscopio MPU-6050 .....	67
<b>11.9</b>	<b>Diseño conceptual del Control PID .....</b>	<b>68</b>
11.9.1	Diseño Conceptual .....	68
11.9.2	Aproximación de la Función de Transferencia .....	70
11.9.3	Simulación del Controlador PID .....	73
12	IMPLEMENTACION, PRUEBAS Y RESULTADOS .....	75
<b>12.1</b>	<b>Control de posición en lazo abierto y trayectorias programadas .....</b>	<b>75</b>
<b>12.2</b>	<b>Control PID digital a servomotores .....</b>	<b>76</b>
<b>12.3</b>	<b>Movimientos efectuados sobre el plano inclinado.....</b>	<b>77</b>

<b>12.4 Objetos deslizantes sobre el panel resistivo.....</b>	<b>78</b>
13 CONCLUSIONES .....	79
14 CRONOGRAMA TT2 .....	81
15 BIBLIOGRAFÍA.....	82
16 ANEXOS .....	85
<b>16.1 ANEXO A: GESTION DE CAMBIOS Y MODIFICACIONES .....</b>	<b>85</b>
16.1.1 A1: Proceso de Modificación de Juntas.....	85
16.1.2 A2: Selección Del Microcontrolador.....	86
16.1.3 A3: Diseño del Circuito Impreso (PCB) .....	87
16.1.1 A5: Corrección del efector final.....	90
16.1.2 A5: Cinemática Inversa.....	91
16.1.3 A7: Cinemática Directa.....	96
16.1.4 A7: Sintonización Del Controlador PID.....	100
<b>16.2 ANEXO B: CARACTERISTICAS DE COMPONENTES ELECTRONICOS.....</b>	<b>101</b>
16.2.1 B1: ATMega328P .....	101
16.2.2 B2: ATMega2560 .....	102
16.2.3 B3: Panel Resistivo – Osatych.....	104
16.2.4 B4: Giroscopio MPU-6050 .....	105
16.2.5 B5: Servomotores DSSERVO DS3225.....	106
<b>16.3 ANEXO C: CALCULOS .....</b>	<b>108</b>
16.3.1 C1: Fusibles para el Sistema Mínimo.....	108
16.3.2 C2: Función de Transferencia de Servomotores .....	109
<b>16.4 ANEXO D: FUNCIONAMIENTO DEL PANEL RESISTIVO 5 HILOS .....</b>	<b>111</b>
<b>16.5 ANEXO E: PLANOS DE MANUFACTURA Y COMPONENTES .....</b>	<b>114</b>
16.5.1 E1: Medidas estándar de rotulas SI-TK.....	114
16.5.2 E2: Planos de la base inferior .....	115
16.5.3 E3: Planos de la base superior.....	116
16.5.4 E4: Planos de la plataforma del efector final. ....	117
16.5.5 E5: Planos del soporte para los sensores. ....	118
<b>16.6 ANEXO F: TABLAS DE TRANSFORMADA Z.....</b>	<b>119</b>
<b>16.7 ANEXO G: CODIGOS DE SIMULACIÓN.....</b>	<b>120</b>
16.7.1 G1: ADC a 18kHz.....	120

16.7.2	G2: Cinemática Inversa (Simulación de Articulaciones) .....	122
16.7.3	G3: Cinemática Inversa (Calculo Numérico) .....	130
16.7.4	G4: PID Servomotores .....	136
16.7.5	G5: PID Panel Resistivo.....	137
17	ANEXO H: CÓDIGOS DE IMPLEMENTACION.....	137
17.1.1	H1: Librerías para el Usuario .....	137
17.1.2	H2: PID servomotores .....	150
17.1.3	H3: PID Panel Resistivo.....	153
17.1.4	H4: Error marginal del PID de servomotores .....	156

## INDICE DE FIGURAS

FIGURA 1 - PATENTE DE JAMES E GWINNETT (PLATAFORMA INDUSTRIAL DE ENTRETENIMIENTO) [2] .....	14
FIGURA 2 - PRIMER ROBOT PARALELO DISEÑADO POR W. POLLARD [3].....	15
FIGURA 3 - SIMULADOR DE VUELO DESARROLLADO POR REDIFON PARA EL MODELO BOEING 747 [4].....	16
FIGURA 4 DIAGRAMA DE BLOQUES SISTEMA SISO [9].....	18
FIGURA 5 - DIAGRAMA DE BLOQUES DE CONTROLADOR PID [10]. .....	19
FIGURA 6 CONTROLADOR PID DIGITAL (ZOH). .....	20
FIGURA 7 - CONFIGURACIÓN DE LAS CAPAS DE UN PANEL RESISTIVO.....	21
FIGURA 8 - SENSOR GIROSCOPIO Y ACCELERÓMETRO MPU-6050. ....	22
FIGURA 9 - GIROSCOPIO MPU-6050 SOBRE EL PLANO INCLINADO.....	23
FIGURA 10 - ROBOT PARALELO DE 3-DOF DESARROLLADO POR REXROTH [14]. .....	24
FIGURA 11 - MÁQUINA DE MEDICIÓN DE COORDENADAS (MMC) DESARROLLADA POR LAPIC [15].....	25
FIGURA 12 - ADVANCED CONCEPT FLIGHT SIMULATOR (ACFS) DESARROLLADO POR CVSFR [16]. .....	25
FIGURA 13 - TRASLAPE DE SEÑALES (ALIASING). .....	29
FIGURA 14 – SEÑAL MUESTREADA MEDIANTE TEOREMA DE NYQUIST.....	30
FIGURA 15 - DISTINTAS CONFIGURACIONES DE LA PLATAFORMA STEWART-GOUGH [20].....	32
FIGURA 16 - PLATAFORMA STEWART 6-SPS [20].....	33
FIGURA 17 - PLATAFORMA STEWART 6-SPS CON ACTUADORES ANGULARES [21].....	33
FIGURA 18 - DESCRIPCIÓN DE TRAYECTORIA DE PLATAFORMA STEWART PARA CIRUGÍA CABG DE [25]....	35
FIGURA 19 - ESPACIO DE TRABAJO DESCrito POR LA PLATAFORMA STEWART-GOUGH EMPLEANDO LA MATRIZ JACOBIANA [21]. .....	36
FIGURA 20 - ESPACIO DE TRABAJO DE LA PLATAFORMA STEWART-GOUGH EN VISTA ISOMÉTRICA [24].....	36
FIGURA 21 - DIMENSIONES DE LA PLATAFORMA STEWART-GOUGH SEGÚN ESPACIO DE TRABAJO. (A) PLATAFORMA EN VISTA FRONTAL CON ACTUADORES PARALELOS ENTRE SÍ. (B) PIerna CON JUNTAS ESFÉRICAS EN VISTA LATERAL.....	37
FIGURA 22 PARTES DE LA BASE. RÓTULAS (ÓVALOS ROJOS), ESLABONES (LÍNEAS AMARILLAS), BRAZOS DE ACTUADOR O CUERNOS (LÍNEA AZUL) Y SERVOMOTORES (FLECHAS VERDES). .....	39
FIGURA 23 VISTA SUPERIOR CONFIGURACIÓN PUNTOS $B_i$ .....	40
FIGURA 24 VISTA SUPERIOR CONFIGURACIÓN PUNTOS $P_i$ .....	41
FIGURA 25 ALTURA INICIAL H Y MARCOS DE REFERENCIA $\{B_5\}$ Y $\{B_6\}$ . .....	42
FIGURA 26 - PLANO UIVI SERVOMOTORES 1, 3 Y 5.....	44
FIGURA 27 - PLANO UIWI SERVOMOTORES 1, 3 Y 5.....	44
FIGURA 28 - SIMULACIÓN DE ADC DE MICROCONTROLADOR ATMEGA328P.....	47
FIGURA 29 - INTERFAZ GRÁFICA DE SIMULACIÓN DE ARTICULACIONES (MODIFICADA).....	48
FIGURA 30 - VALOR MÁXIMO APROXIMADO DE ÁNGULO DE “ALABEO” .....	48
FIGURA 31 - VALOR DE ÁNGULO CRÍTICO PARA ALABEO .....	49
FIGURA 32 - VALORES MÁXIMO Y CRÍTICO DE ÁNGULO DE CABECEO. .....	50
FIGURA 33 - ÁNGULO MÁXIMO DE GUIÑADA. .....	50
FIGURA 34 - LONGITUD MÁXIMA DE LA PLATAFORMA SOBRE EJE X.....	51
FIGURA 35 - LONGITUD MÁXIMA DE LA PLATAFORMA SOBRE EJE Y.....	51
FIGURA 36 - LONGITUD MÁXIMA DE LA PLATAFORMA SOBRE EJE Z .....	51
FIGURA 37 - ESTUDIO DE ESTRÉS VON-MISES.....	53
FIGURA 38 - ESTUDIO DE DEFORMACIÓN. .....	54
FIGURA 39 - ESTUDIO DEL FACTOR DE SEGURIDAD. .....	54

FIGURA 40 - ESTUDIO DE DEFORMACIÓN Y ESFUERZOS DE LA PLATAFORMA STEWART-GOUGH APLICADO AL PANEL RESISTIVO .....	55
FIGURA 41 - MICROCONTROLADOR ATMEGA328P EN ENCAPSULADO THT [31]. .....	56
FIGURA 42 - PINES DE CONEXIÓN DEL SENSOR MPU-6050. ....	57
FIGURA 43 - PANEL RESISTIVO PRODUCTO DE OSATYCH. ....	58
FIGURA 44 SERVOMOTOR DIGITAL DS3225 25 KG/CM [34] CON CUERNO DE ALUMINIO INCORPORADO.....	59
FIGURA 45 - ESLABÓN PARA PIERNAS CON JUNTAS ESFÉRICAS EN LOS EXTREMOS.....	59
FIGURA 46 DISEÑO CIRCUITO ELECTRÓNICO. ....	62
FIGURA 47 - DISEÑO PCB PARA PLATAFORMA STEWART-GOUGH.....	63
FIGURA 48 - ENSAMBLE DEL CENTRO DE EFECTOR FINAL CON GIROSCOPIO .....	64
FIGURA 49 - ENSAMBLE DE ROTULA INFERIOR CON CUERNO DEL SERVOMOTOR.....	65
FIGURA 50 ENSAMBLE FINAL PLATAFORMA STEWART-GOUGH.....	65
FIGURA 51 - PINES DE CONEXIÓN DEL SENSOR PANEL RESISTIVO.....	66
FIGURA 52 - CONTROL CINEMÁTICO PARA LA PLATAFORMA STEWART-GOUGH. ....	68
FIGURA 53 - DIAGRAMA DEL BLOQUE DEL CONTROLADOR PID .....	69
FIGURA 54 - DIAGRAMA COMPLETO DEL CONTROLADOR DISEÑADO PARA LA PLATAFORMA STEWART-GOUGH .....	70
FIGURA 55 - CONTROLADOR PID PARA PLATAFORMA AUTOESTABILIZADA [37] .....	71
FIGURA 56 DIAGRAMA DE BLOQUES PID BOLA Y PLATO [38]. .....	72
FIGURA 57 RESPUESTA AL ESCALÓN UNITARIO. ....	73
FIGURA 58 - FIGURA 57 SIMULACIÓN CONTROLADOR PID DIGITAL CON BLOQUE PID DE SIMULINK.....	74
FIGURA 59 SIMULACIÓN CONTROLADOR PID DIGITAL CON BLOQUE PID DE SIMULINK. ....	74
FIGURA 60 PLATAFORMA STEWART DESPLAZADA 4 CM EN EJE X.....	75
FIGURA 61 - MECANISMO DE SUJECIÓN DE POTENCIÓMETRO PARA MEDIR POSICIÓN DEL SERVOMOTOR .....	76
FIGURA 62 - ERROR MARGINAL DEL CONTROLADOR PID DE LOS SERVOMOTORES .....	77
FIGURA 63 GIRO DE PLATAFORMA EN ROLL DE 15°.....	77
FIGURA 64 POSICIÓN DE OBJETO ESFÉRICO DESLIZANTE SOBRE PANEL RESISTIVO.....	79
FIGURA 65 - ADAPTACIÓN DE ROTULAS M8 A PERFORACIÓN M3 .....	85
FIGURA 66 - ROTULA ENSAMBLADA.....	86
FIGURA 67 - ADAPTADOR SMD A THT PARA EMPAQUETADO TQFP100 .....	87
FIGURA 68 - DIAGRAMA DEL SISTEMA MÍNIMO ATMEGA2560.....	88
FIGURA 69 - DISEÑO DE PCB PARA SISTEMA MÍNIMO ATMEGA2560.....	89
FIGURA 70 - VISIÓN 3D DEL SISTEMA MÍNIMO ATMEGA2560 .....	89
FIGURA 71 - ENSAMBLE DE LA PCB SISTEMA MÍNIMO DEL ATMEGA2560 .....	90
FIGURA 72 - MODIFICACIÓN DEL EFECTOR FINAL.....	90
FIGURA 73 DIAGRAMA ESQUEMÁTICO DE SISTEMAS DE REFERENCIA NUEVOS. ....	91
FIGURA 74 DIAGRAMA DE POSICIONAMIENTO DE JUNTAS.....	92
FIGURA 75 DIAGRAMA DEL EFECTOR FINAL ROTADO Y TRASLADADO.....	95
FIGURA 76 DISEÑO CAD PLATAFORMA STEWART.....	95
FIGURA 77 MODELADO DE CINEMÁTICA DIRECTA POR METODOLOGÍA DENAVIT-HARTENBERG. ....	96
FIGURA 78 VISTA SUPERIOR DE EJES Z DE ARTICULACIONES.....	97
FIGURA 79 DIAGRAMA DE BLOQUES CONTROLADOR PID SERVOMOTORES. BLOQUE CFT(s).....	100
FIGURA 80 GRÁFICA GANANCIAS KP, KI Y KD.....	101
FIGURA 81 PINOUT ATMEGA328P.....	102
FIGURA 82 POTENCIÓMETRO ACOPLADO A SERVOMOTOR POR PIEZA DE IMPRESIÓN 3D. ....	109
FIGURA 83 MEDICIÓN <i>tm0</i> .....	110

FIGURA 84 MEDICIÓN DE TIEMPO $t_1$ .	110
FIGURA 85 MEDICIÓN DE TIEMPO $t_2$ .	110
FIGURA 86 MEDICIÓN DE TIEMPO $t$ .	111
FIGURA 87 - GRADIENTE DE VOLTAJE GENERADO PARA POSICIÓN X.	112
FIGURA 88 – GRADIENTE DE VOLTAJE GENERADO PARA POSICIÓN Y.	112
FIGURA 89- DIAGRAMA ELÉCTRICO PARA SIMULACIÓN DEL ADC.	121
FIGURA 90 DIAGRAMA DE BLOQUES DE CONTROLADOR PID DIGITAL (SIMULINK).	136
FIGURA 91 DIAGRAMA DE BLOQUES CONTROLADOR PID DIGITAL PANEL RESISTIVO.	137
FIGURA 92 GANANCIAS CONTROLADOR PID PANEL RESISTIVO (SIMULINK).	137

## **INDICE DE TABLAS**

TABLA 1 - RANGO DE ESCALAS SENSOR MPU-6050.....	22
TABLA 2 - COMPARACIÓN DE MICROCONTROLADORES.....	31
TABLA 3 COMPARACIÓN ENTRE ROBOTS SERIALES Y PARALELOS [27] .....	38
TABLA 4 - SELECCIÓN DEL PRE-ESCALADOR PARA FRECUENCIA Ws=36kHz.....	46
TABLA 5 PARÁMETROS DENAVIT-HARTENBERG CALCULADOS DE PLATAFORMA STEWART.....	98
TABLA 6 TABLA MODIFICADA DE PARÁMETROS DENAVIT-HARTENBERG. ....	99

## 1 RESUMEN

En este documento se presenta el diseño de un control PID digital para una plataforma Stewart-Gough, delimitada por seis grados de libertad (GDL) que permiten al efecto final tener desplazamiento en los ejes XYZ y rotación con restricciones de alabeo, cabeceo y guiñada.

Comprende el estudio y resolución de las cinemáticas directa e inversa de la plataforma, así como el espacio de trabajo descrito por el efecto final y su correspondiente simulación de movimientos y articulaciones para estudiar puntos extremos y posibles singularidades.

A partir de la definición de características, el diseño CAD generado a partir del modelo matemático generalizado de dominio público y la selección general de materiales para la construcción del prototipo funcional, se genera un estudio de fuerzas aplicadas para observar los puntos con concentradores de esfuerzos, el factor de seguridad y posibles deformaciones.

También se considera la estimación del periodo de muestreo para la selección del microcontrolador y una definición aproximada del tiempo de respuesta.

El desarrollo de este prototipo y su documentación se plantean como material didáctico para el estudio, diseño y control de mecanismos paralelos.

## 2 PALABRAS CLAVE

Plataforma Stewart-Gough, Control PID Digital, Seis Grados de Libertad, Efecto Final, Cinemáticas Directa e Inversa, Diseño CAD.

### 3 ABSTRACT

This document presents the design of a digital PID control implemented on a Stewart-Gough platform, delimited by six degrees of freedom (DOF), i.e., displacement on the XYZ axis and rotation with constraints of roll, pitch, and yaw.

It covers the study and solution of kinematics, forward and inverse, correspondent to angular actuators, as well as the workspace described by the end-effector and its respective simulation for articulation motions to study extreme points and possible singularities.

A forces analysis is generated based on applied forces to observe stress concentrators, security factor and possible deformations. The definition of features and CAD design are generated from a math model taken from public domain and the general selection of materials to construct a functional prototype are presented.

An estimation of sampling period is also considered to select the microcontroller and an approached definition for time response.

The development of this prototype and its documentation are proposed as educational material for the study, design and control of parallel mechanisms and digital control.

### 4 KEYWORDS

Stewart-Gough Platform, PID Digital Control, Six Degrees of Freedom, End-Effector, Forward and Inverse Kinematics, CAD Design.

## 5 INTRODUCCIÓN

Cuando se habla de robótica industrial uno de los primeros pensamientos asociados es a los robots seriales que son altamente aplicables a este ámbito, sin embargo, los robots paralelos también tienen una gran importancia y gama de aplicaciones en este entorno. Entre sus características más importantes se distingue su alta relación carga/potencia. Para el caso de la Plataforma Stewart-Gough esta característica reside en su posibilidad de distribuir la carga aproximadamente en seis partes iguales por lo cual la capacidad de carga total del robot paralelo se aproxima a seis veces la capacidad de carga de cada actuador de manera individual.

La *Plataforma Stewart-Gough* es un manipulador de tipo paralelo que implica una configuración de seis grados de libertad, y a cada grado de libertad le corresponde un actuador. Los seis actuadores pueden ser lineales o angulares y unen mediante un enlace a la parte inferior del robot la cual no tiene movilidad y se le llamada *base*, con la parte superior que sí tiene movilidad y se le denomina *efector final*. Debido a esta configuración la *Plataforma Stewart-Gough* presenta una alta rigidez, lo cual se puede traducir en mayor precisión al momento de transmitir movimientos.

El *efector final* es la parte de mayor interés del robot, ya que su posición y orientación son las características que determinan si el robot es capaz de cumplir con la precisión necesaria para implementarse y satisfacer una necesidad; ya sea industrial, educacional, de investigación, etc. Puede tener distintas aplicaciones, por ejemplo, orientación de satélites, simuladores de vuelo, manufactura por CNC, máquina de medición de coordenadas, *shakers* (también llamados *agitadores* que son parte del instrumental de laboratorio de química para homogeneizar disoluciones), entre otros.

Este documento explica el diseño, construcción e implementación de un control PID discreto a una plataforma *Stewart-Gough*, para ello se pasa por un proceso de desarrollo descrito en 11 donde se contemplan procesos empleados para la selección del microcontrolador, el modelo matemático para la determinación de las medidas de la plataforma, simulaciones para comprobar el movimiento, articulaciones, factor de seguridad, esfuerzos y muestreo del microcontrolador. Se describe la selección de componentes, el diseño electrónico y la propuesta de características y diagrama conceptual del controlador PID, luego de ello se explican los procesos de ensamble, pruebas, correcciones mecánicas, y finalmente se presenta el método para la obtención de la FT, diseño del control PID, simulaciones, implementación y resultados.

## 5.1 Objetivos

### 5.1.1 Objetivo General:

Implementar un sistema de control PID digital a una plataforma *Stewart-Gough*, que detecte y corrija el error producido por perturbaciones, aplicación de fuerzas externas y movimientos sobre el plano inclinado, con la finalidad de mantener la posición horizontal del efecto final.

### 5.1.2 Objetivos específicos:

- Calcular la cinemática inversa y realizar la simulación del mecanismo.
- Construir un robot de configuración paralela tipo hexápodo de seis grados de libertad.
- Implementar el sistema PID en lazo abierto para el control de posición y velocidad del efecto final.
- Establecer el tiempo de respuesta y el error de posición necesarios para mantener la posición horizontal del efecto.
- Determinar el método para implementar un control PID digital adecuado al mecanismo, como sistema de múltiples entradas y múltiples salidas (MIMO).
- Diseñar un controlador PID digital que cumpla con el error de posición y tiempo de respuesta establecidos.
- Seleccionar e implementar el controlador para el sistema PID.
- Diseñar el circuito electrónico que permita el control de la plataforma, así como el algoritmo del controlador discreto.
- El prototipo trabajará tanto en lazo abierto como en lazo cerrado, para realizar de igual manera su función principal y el seguimiento de trayectorias preprogramadas.

## 6 JUSTIFICACIÓN

Actualmente la Unidad Profesional Interdisciplinaria de Ingeniería Zacatecas (UPIIZ) IPN no cuenta con procesos o mecanismos para prácticas y/o demostración de control. Es por ello que en el siguiente documento se plantea como propuesta el diseño y construcción de un control PID digital aplicado a una plataforma *Stewart-Gough*. Dicha plataforma pertenece a la familia de los robots paralelos, los cuales se presentan con diversas funciones en la industria, tales como el movimiento de objetos, sostén de sustancias, simulación, entre otros.

El diseño, construcción y control de este mecanismo se realiza con la finalidad de emplearlo como herramienta didáctica para centros educativos de nivel superior, más específicamente como demostración para unidades de aprendizaje de control clásico, control de sistemas mecatrónicos, sistemas neurodifusos, análisis y síntesis de mecanismos, modelado,

autómatas industriales, entre otros que impliquen sistemas con retroalimentación. De esta forma se podría conseguir un mayor aporte de las unidades de aprendizaje, siendo que el trabajo teórico se estaría complementado con prácticas y/o demostraciones físicas del mecanismo y sus variantes, usos, y características.

Anteriormente en la unidad académica se ha diseñado y construido una plataforma *Stewart-Gough* con control de posicionamiento en lazo abierto, sin embargo, la principal diferencia en el actual trabajo será definida por el diseño, construcción e implementación del control en lazo cerrado; este dará al sistema una retroalimentación que permitirá controlar su posición de manera automática con lo cual se puede obtener un aspecto más amplio de usos y aplicaciones dentro de la institución y unidades de aprendizaje.

Para el diseño del control PID digital en lazo cerrado primero se debe implementar como un sistema de lazo abierto que permitirá el control de posición y velocidad del *efector final*, por lo tanto, se ha considerado la posibilidad de enfocar el uso la plataforma *Stewart-Gough* (además de herramienta didáctica) como instrumento de laboratorio químico, con la finalidad de automatizar algunos procesos tales como la homogenización de disoluciones mediante alguna señal armónica o una trayectoria previamente programada. Así mismo, como un sistema de lazo cerrado, puede emplearse para el transporte de reactivos colocándolos en la superficie del *efector final* con el objetivo de mantenerlos lo más estables posibles durante el recorrido.

## 7 MARCO TEÓRICO.

### 7.1 Historia de los robots paralelos

El término de *robot paralelo* también conocido como *robot de cadena cerrada* o *manipulador paralelo*, consiste en una base fija unida a un *efector final* o *base móvil* la cual puede realizar su movimiento basado en el movimiento de los actuadores que están situados de tal forma que cierran la cadena cinemática entre ambas bases [1]. El principio de funcionamiento inicialmente surgió con James E. Gwinnett, quien diseño una plataforma para la industria del entretenimiento en la patente solicitada de 1928 [2]. Sin embargo, no fue sino hasta 1940 cuando W. Pollard diseña el primer robot paralelo industrial destinado a procesos de pintura con aerosol [3].

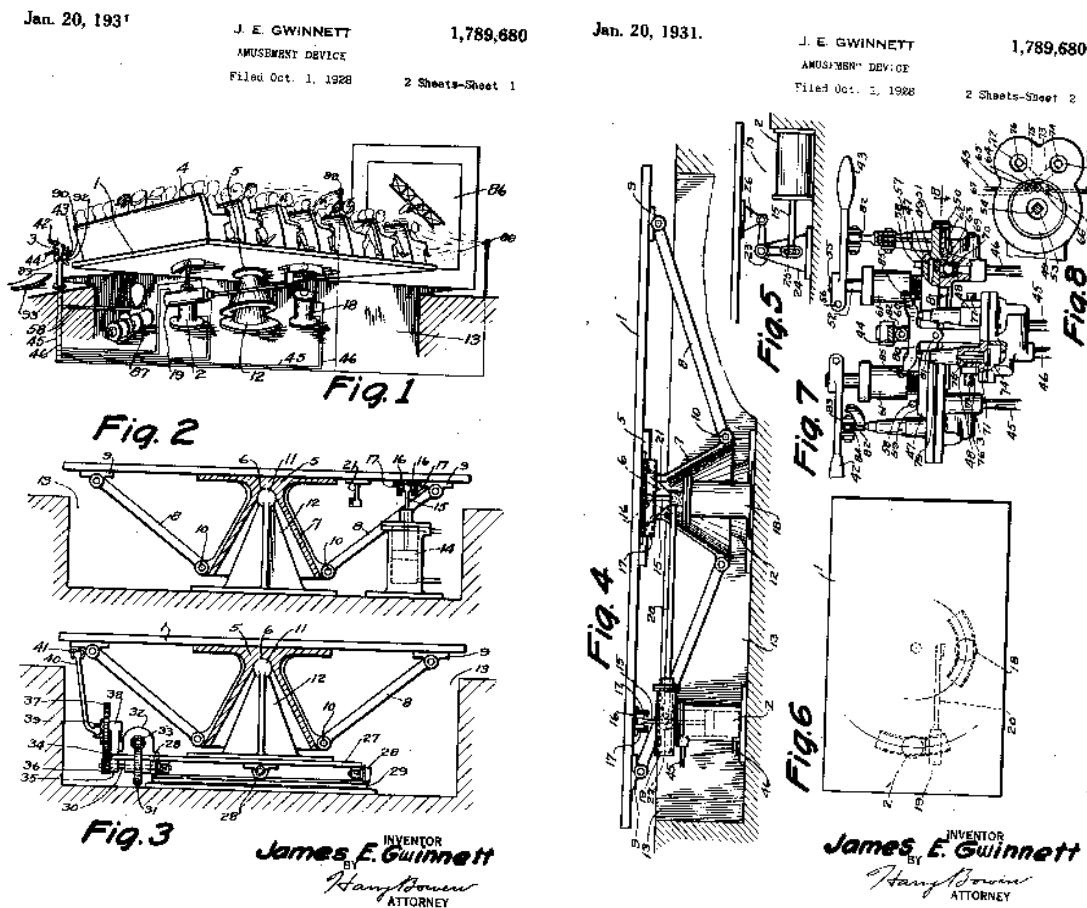


Figura 1 - Patente de James E Gwinnett (plataforma industrial de entretenimiento) [2].

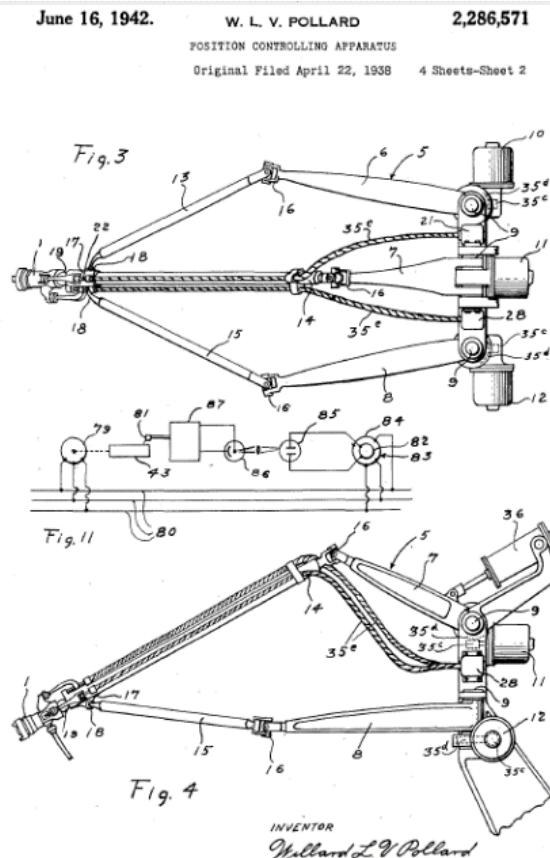


Figura 2 - Primer robot paralelo diseñado por W. Pollard [3].

Luego de los primeros avances de las plataformas Stewart aplicadas a procesos industriales, algunas empresas comenzaron a realizar variantes de estas plataformas, ejemplo de ello es *Redifon*, a quien se le solicitó por parte de *Lufthansa German Airlines* la producción de un simulador de vuelo, con el modelo inicial diseñado para su entonces nueva flota Boeing 727. Este modelo de simulador incluyó tres ejes, que brindaban al mecanismo la movilidad necesaria para recrear el comportamiento del avión. Actualmente *Redifon* continúa en servicio desde su puesta en operación en 1949, cuando comenzó con la producción de simuladores de vuelo, entrenadores y desarrollo de nuevas técnicas [4].

La producción actual de *Redifon* incluye equipo de entrenamiento en tierra para todos los aviones modernos mencionados a continuación:

- BOEING 707. DC 8 CARAVELLE.
  - BOEING 727 - VC 10 COMET.
  - FOKKER F 27 • CL 44 VISCOUNT.
  - C 130 A • CV 990 VANGUARD.



Figura 3 - Simulador de vuelo desarrollado por Redifon para el modelo Boeing 747 [4].

## 7.2 Cinemática Inversa

La cinemática inversa es el modelo matemático de un manipulador que calcula la posición de los actuadores respecto a un marco de coordenadas XYZ, por medio de la orientación y/o posición del efecto final como parámetros de entrada [5].

### 7.2.1 Fundamentos matemáticos

#### 7.2.1.1 Matrices transformación homogénea

Una matriz de transformación homogénea es una matriz de dimensión de 4x4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro [5].

$$T = \begin{bmatrix} R_{3x3} & T_{3x1} \\ f_{1x3} & w_{1x1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} \quad (1)$$

#### 7.2.1.2 Matriz de transformación homogénea de posición

Esta matriz denota la posición en el espacio tridimensional con el vector columna de posición  $P(p_x, p_y, p_z)$  [5].

$$TP = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

#### 7.2.1.3 Matriz de transformación homogénea de rotación

Se define como la serie de composiciones de rotación respecto a los ejes X, Y y Z [5]. Las letras **s** y **c** corresponden a las funciones trigonométricas seno y coseno, respectivamente.

$$R = R(z, g)R(y, b)R(x, g) = \begin{bmatrix} cbcg & -sgca + cgsbsa & sgsa + cgsbca & 0 \\ sgcb & cgca + sgsbsa & -cgsa + sgsbca & 0 \\ -sb & cbsa & cacb & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

## 7.3 Función de transferencia

### 7.3.1 Sistema de primer orden

Un sistema de primer orden posee dos parámetros a considerar para su modelado: la ganancia estática  $K_m$  y el tiempo de respuesta  $t_m$  [6].

$$S(s) = \frac{\sigma(s)}{V(s)} = \frac{K_m}{\tau_m s + 1} \quad (4)$$

Considerando  $V(s)$  una función escalón de amplitud  $A$ , la función temporal  $s(t)$  evaluada en infinito y despejando  $K_m$  resulta (6)

$$\sigma(t) = AK_m(1 - e^{-\frac{t}{\tau_m}}) \quad (5)$$

$$K_m = \frac{\omega_{final}}{A \tau_m} \quad (6)$$

### 7.3.2 Tiempo de respuesta

#### 7.3.2.1 Aproximación de Sundaresan-Krishnaswamy (SK)

La aproximación **S-K** [7] considera dos parámetros de tiempo, denominados como  $t_1$  y  $t_2$ . Estos tiempos son medidos cuando la amplitud de la función escalón alcanza respectivamente el 85.3% y 35.3% (7).

$$\tau_{mSK} = 0.674803(t_2 - t_1) + t_1 \quad (7)$$

#### 7.3.2.2 Aproximación de Ogata

La segunda opción de aproximación del servomotor es la de Ogata [6]. Evaluando (5) en el tiempo de respuesta  $\tau_{mSK}$  corresponde al 63.2% del producto entre la ganancia estática del sistema y la amplitud del escalón (8).

$$\sigma(\tau_{mSK}) = 0.632AK_m \quad (8)$$

## 7.4 Control PID (Proporcional-Integral-Derivativo)

El controlador PID ha sido una herramienta modelo o estándar cuando el control de procesos surgió en los años 40 [8]. Considérese el PID como un lazo cerrado de control de una entrada y una salida [9], su diagrama de bloques mostrado en la Figura 4:

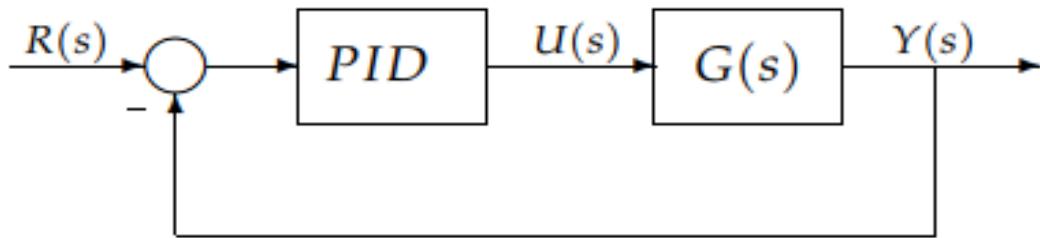


Figura 4 Diagrama de Bloques Sistema SISO [9].

De la familia de control PID existen tres acciones de control: *proporcional (P)*, *integral (I)* y *derivativa (D)* [9].

Se le denomina **proporcional** a la primera acción de control de este tipo control, y produce una salida, como su nombre lo dice, proporcional al error [9]. Es denominado como  $K_p$ .

$$u(t) = K_p \cdot e(t) \quad (9)$$

Esta ganancia o controlador proporcional puede controlar cualquier planta estable, sin embargo, presenta un error en estado estable (off-set).

La segunda acción de control es la **integral**, su trabajo es generar una salida del controlador proporcional al error acumulado, lo que implica que es un modo de controlar lento [9]. Su función de transferencia es:

$$C_i(s) = \frac{K_i}{s} \quad (10)$$

Por último, se describe que el comportamiento de la acción de control **derivativa** es responder a la velocidad del cambio del error y producir una corrección significativa, antes de que la magnitud del error se vuelva demasiado grande [6].

$$C_d(s) = K_D s \quad (11)$$

La señal de control  $u(t)$  tiene un valor diferente de cero cuando la señal de error  $e(t)$  es cero. En otras palabras, si se da como referencia una constante, o perturbaciones, el error en régimen permanente es cero [9].

Al combinar las acciones de control (proporcional, integral y derivativo) se conservan las características y ventajas de las tres acciones de control. La *Figura 5* representa el diagrama de bloques de un sistema de control realimentado, donde el controlador PID es delimitado por el bloque  $G_c(s)$  [10]:

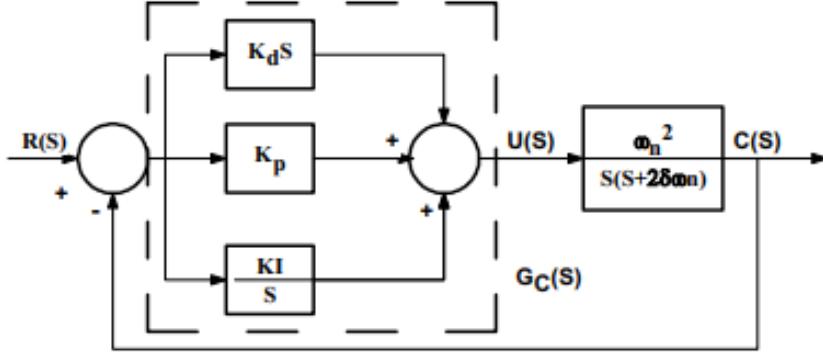


Figura 5 - Diagrama de bloques de controlador PID [10].

La función de transferencia [6]:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (12)$$

#### 7.4.1 Transformada Z

La transformada Z es un método operacional muy útil si se requiere trabajar con sistemas en tiempo discreto. Su aplicación a una función del tiempo  $x(t)$  (donde  $t$  es positivo), o a la secuencia de valores  $x(kT)$  (donde  $k$  adopta valores de cero o de enteros positivos y  $T$  es el periodo de muestreo), se define mediante la ecuación (13) [11]:

$$X(z) = Z \{ x(t) \} = Z \{ x(kT) \} = \sum_{k=0}^{\infty} x(kT) z^{-k} \quad (13)$$

Semejante a esta expresión la transformada Z aplicada a una secuencia de números  $x(k)$  será [11]:

$$X(z) = Z \{ x(k) \} = \sum_{k=0}^{\infty} x(k) z^{-k} \quad (14)$$

#### 7.4.2 La Transformada de Laplace y la Transformada Z

Así como la transformada de Laplace está definida por señales de tiempo continuo, la transformada Z es definida por señales en tiempo discreto [12].

Supóngase una función  $g^*(kTs)$ , donde  $k$  es un entero positivo mayor o igual a cero y  $Ts$  es un número positivo que denota el tiempo de muestreo (15).

$$g^*(kTs) = \sum_{k=0}^{\infty} g(kTs) \delta(t - kTs) \sum_{k=0}^{\infty} g(kTs) \delta(t - kTs) \quad (15)$$

Donde  $\delta(t - kTs)$  es la función impulso.

A continuación, se presenta la (16) como la transformada de Laplace de (15). Definiendo  $z = e^{-T_s*s}$  resulta la ecuación (17), es decir, la transformada Z de (15) [11].

$$G^*(s) = L \{g^*(kT_s)\} = \sum_{k=0}^{\infty} g(kT_s)e^{-kT_s} \quad (16)$$

$$G^*(s)|_{z=e^{-T_s*s}} = \sum_{k=0}^{\infty} g(kT_s)z^{-k} \quad (17)$$

Cabe mencionar que, así como existe una transformada Z, existe una transformada Z inversa. Para esto, todas las variables  $z$  deben ser expresadas en exponente negativo para obtener la ecuación de diferencias (anexo F 16.6).

#### 7.4.3 Retenedor de Orden Cero

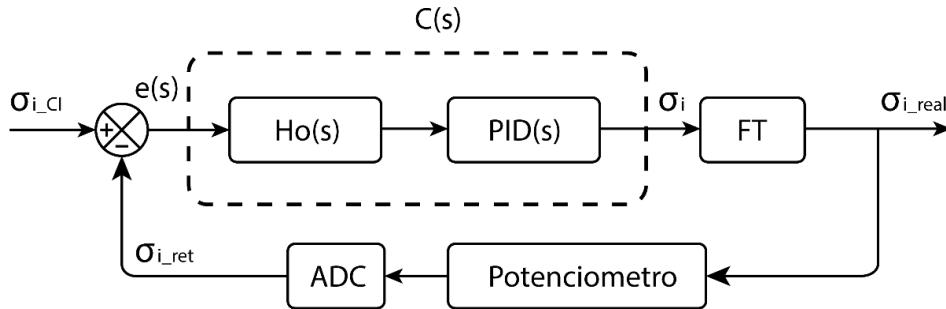


Figura 6 Controlador PID digital (ZOH).

#### 7.4.4 PID Digital por Retenedor de Orden Cero (ZOH)

La Figura 6 muestra el diagrama de bloques del controlador PID digital aplicado a los servomotores de la plataforma Stewart-Gough de este documento.

Aplicando la transformada Z al bloque  $C(s)$ :

$$C(z) = \frac{s_i(z)}{e(z)} = Z \left\{ \left( \frac{1 - e^{-T_s*s}}{s} \right) \left( K_P + \frac{K_I}{s} + K_D s \right) \right\} \quad (18)$$

Una vez reducida y expresada en términos de  $z^{-n}$  se obtiene la expresión siguiente:

$$s_i(z) = C_0 e(z) + C_1 e(z)z^{-1} + C_2 e(z)z^{-2} + s_i(z)z^{-1} \quad (19)$$

Donde los términos  $C_0$ ,  $C_1$  y  $C_2$  son términos que contienen las ganancias de control PID:

$$C_0 = K_P + K_D \quad (20)$$

$$C_1 = K_I T_s - K_P - 2K_D \quad (21)$$

$$C_2 = K_D \quad (22)$$

Por último, al aplicar transformada inversa Z en la (19) se obtiene la ecuación de diferencias del controlador PID digital (23).

$$s_i[n] = C_0 e[n] + C_1 e[n - 1] + C_2 e[n - 2] + s_i[n - 1] \quad (23)$$

## 7.5 Sensores de Posición y Orientación

### 7.5.1 Sensor Panel Resistivo

El sensor colocado en la parte superior del *efector final* encargado de medir los movimientos de objetos desplazándose sobre la superficie de éste, será un panel resistivo de 5 hilos (se puede encontrar en 4, 5 u 8 hilos), el cual consiste en una capa parcialmente conductiva (resistiva) que se aplica de forma uniforme al panel. Los paneles rígidos y flexibles se montan con las barras de bus perpendiculares entre sí como se muestra en la Figura 7.

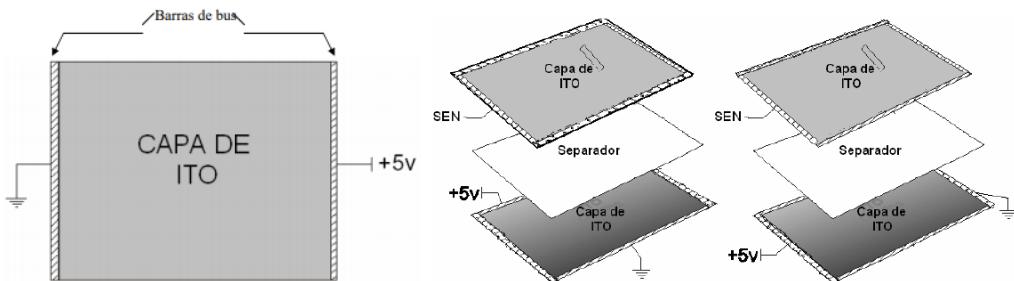


Figura 7 - Configuración de las capas de un panel resistivo.

La medición del sensor se hace mediante la aplicación de una fuerza sobre la *capa de ito* la cual genera un gradiente de voltaje a través de una de las capas y midiendo la tensión en la otra capa. Este gradiente se produce normalmente poniendo a tierra una barra de bus y aplicando +5V en la otra barra, con lo cual se produce un gradiente de tensión en los ejes que cruzan el panel, este se aplica sólo a una capa rígida, mientras que la otra capa es la sensible en ambas mediciones.

La medición se hace de manera secuencial en la cual los sucesos se presentan con el gradiente de voltaje en el eje X y se mide la tensión en la capa sensora, seguido de esto se conmuta el gradiente de voltaje al eje Y y se toma otra medición de la misma capa sensora. Para determinar la velocidad de muestreo del panel resistivo se considera usarse como interfaz humana donde el contacto será un dedo y la resolución necesaria no es muy alta. Siendo un panel de 6 in de largo y la resolución de contacto es 2 in (tamaño promedio de un dedo) entonces el convertidor análogo-digital (ADC) debe representar 30 puntos. Con 5 bits se pueden representar 32 puntos, por lo que es suficiente emplear un convertidor ADC de 8 a 12 bits y una velocidad de muestreo de 10 muestras por segundo.

### 7.5.2 Sensor Giroscopio MPU-6050

Un giroscopio es un sensor que mide velocidades angulares, y dependiendo de su disposición, puede medir velocidades sobre cualquier eje de rotación. El MPU-6050 (Figura 1) es un dispositivo *MotionTracking* (rastrea y/o describe el movimiento de un objeto en el espacio), su presentación en circuito integrado de seis ejes de libertad combina un giroscopio de tres ejes, un acelerómetro de tres ejes, y un *Procesador Digital de Movimiento* (DMP).

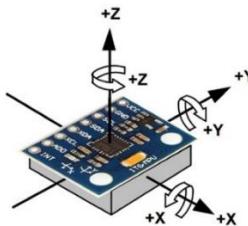


Figura 8 - Sensor giroscopio y acelerómetro MPU-6050.

Se comunica mediante el bus de I2C y se pueden introducir otros tres ejes de un magnetómetro externo para proporcionar una salida completa de nueve ejes de libertad. Cuenta con tres ADC's de 16 bits para la digitalización de las salidas giroscopio y tres ADC's de 16 bits para la digitalización de las salidas del acelerómetro. Para mayor versatilidad puede ser configurado para movimientos rápidos y lentos ajustando la sensibilidad en un rango de escalas que se muestra en la tabla

Tabla 1 - Rango de Escalas Sensor MPU-6050

Rango de Escalas				
Acelerómetro	±2g	±4g	±8g	±16g
Giroscopio	±250°/seg	±500°/seg	±1000/seg	±2000/seg

Si se desea emplear el sensor MPU-6050 para la detección de posición angular se deberá realizar un acondicionamiento de señales, esto debido principalmente a que las mediciones obtenidas mediante este son de aceleración lineal (X, Y, Z) y velocidad angular ( $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ ). Por lo tanto existen diversos métodos para obtener la posición angular a partir de estas magnitudes.

Considerando la posición del sensor como si estuviera colocado en un plano inclinado como se muestra en la figura se puede observar que la aceleración de la gravedad afectará directamente a los ejes X y Y ya que esta es una aceleración constante y paralela al eje Z, por lo tanto al realizar un giro de roll y/o pitch se podrán observar cambios en la aceleración de dichas componentes hasta llegar a un valor máximo cuando estas sean paralelas al plano vertical que traza el eje Z.

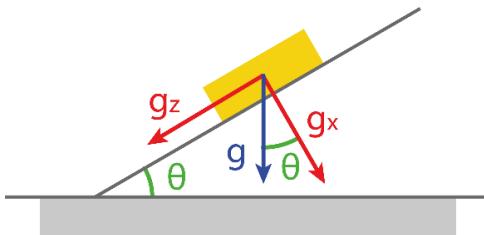


Figura 9 - Giroscopio MPU-6050 sobre el plano inclinado

A partir de esta consideración se aplica la siguiente función trigonométrica:

$$\theta_x = \left( \frac{A_x * g}{\sqrt{(A_y * g)^2 + (A_z * g)^2}} \right) \quad (24)$$

$$\theta_y = \left( \frac{A_y * g}{\sqrt{(A_x * g)^2 + (A_z * g)^2}} \right) \quad (25)$$

Para medir en ángulo de giro o posición angular implementando el giroscopio se tiene que aplicar una integración del valor obtenido puesto que este es la velocidad y en teoría debería obtenerse la posición al integrarse, para ello se implementa un diferencial de tiempo pequeño y se agrega a la ecuación quedando como se muestra en la ecuación.

$$\theta_x = \theta_{x0} + \omega_x \Delta t \quad (26)$$

$$\theta_y = \theta_{y0} + \omega_y \Delta t \quad (27)$$

## 8 ANTECEDENTES

En pasadas generaciones, se realizó un trabajo semejante al propuesto en este documento, el proyecto realizado por los alumnos Abel Villanueva Esquivel, Eduardo Hernández de Haro y Sergio Ramírez Durán, el cual consistía en un proyecto de trabajo terminal de la Unidad Profesional Interdisciplinaria de Ingeniería Zacatecas, donde se documentó una plataforma *Stewart-Gough* en configuración de hexápedo, la cual cuenta con seis grados de libertad al igual que la propuesta en este documento, sin embargo, el principal propósito de dicha plataforma era, la manipulación de la posición y velocidad del *efector final* en los planos XYZ y giro restringido por alfa, beta y gamma [13].

Esta plataforma, teóricamente fue diseñada para soportar cargas de 200 lb por actuador, por lo cual, la configuración de robot paralelo con el *efector final* completamente horizontal debería ser capaz de soportar cargas máximas hasta 1200 lb. Por lo tanto, dado que la documentación se encuentra disponible en la unidad académica, se revisará el mecanismo y el control implementado, así como los cálculos de la cinemática inversa que asemejará al de este trabajo propuesto [13].

## 9 ESTADO DEL ARTE.

Para la investigación del estado del arte, se ha basado tanto en proyectos realizados por universidades, centros de investigación, instituciones privadas, así como productos actualmente disponibles en el mercado, es por ello por lo que algunas de las referencias de los siguientes proyectos mencionados pueden pertenecer a libros, tesis, artículos de congresos, artículos revistas científicas, informes o directamente de los sitios web para la adquisición de dichos productos.

Recientemente, surgió en el mercado E-Motion 6000, la cual es una plataforma de tres grados de libertad (3-DOF) desarrollada por la empresa *Bosch Rexroth*. Este simulador de movimiento es un mecanismo que permite el desplazamiento vertical y la rotación en torno a los ejes horizontales, por lo tanto, cuenta con una torsión cero, lo cual permite realizar movimientos precisos con grandes cargas. Está diseñado principalmente para un rango de carga media de 6000Kg y el rango de movimiento del actuador alcanza los 935mm [14].



Figura 10 - Robot paralelo de 3-DOF desarrollado por Rexroth [14].

Una de las principales características o ventajas de este tipo robots paralelos es su gran capacidad de carga, sin embargo, otra gran ventaja es su gran precisión y un mínimo error de posición, por lo cual, la empresa rusa *Lapic* ha diseñado una serie máquinas de medición por coordenadas (MMC) hexápodos de alta precisión. Esto implica un nuevo nivel en el proceso de control de calidad, dado que su configuración le permite medir las dimensiones de formas tanto estándar, como basados en spines irregulares, en piezas con cavidades internas y superficies "sombreadas". Incluso se puede realizar mediciones sobre ruedas dentadas, roscas (internas y externas) en incrementos de 0.1 [15].



Figura 11 - Máquina de medición de coordenadas (MMC) desarrollada por Lapic [15].

El *advanced concept flight simulator* (ACFS por sus siglas en inglés) desarrollado dentro del *centro de investigación de sistemas de vehículos tripulados* (CVSRF) es un simulador de vuelo que recrea el ambiente de cabina de un avión de transporte de pasajeros comercial, más específicamente la del Boeing 747-400. Dentro de la cabina se encuentran los mandos de dicha aeronave y estos están ligados a un entorno de movimiento virtual. Con la visualización del entorno en pantallas, y el movimiento generado por el mecanismo de la plataforma de seis grados de libertad, recrean el ambiente de vuelo de tal modo que permitan la recopilación de datos como un gran beneficio para los investigadores que realizan simulaciones en tiempo real [16].



Figura 12 - Advanced Concept Flight Simulator (ACFS) desarrollado por CVSRF [16].

## 10 PLANTEAMIENTO DEL PROBLEMA

Al ser estudiantes de la Unidad Profesional Interdisciplinaria de Ingeniería Zacatecas (UPIIZ) IPN se observó el escaso equipo de prácticas para la implementación de sistemas de control digital. Así mismo, se observa descuido en un sector importante de índole industrial que implica procesos y/o mecanismos de robots paralelos. Es por ello que, como aporte a la unidad académica, se describe el siguiente trabajo en el cual se plantea desarrollar un controlador PID digital para una *plataforma Stewart-Gough*, así como el prototipo físico y funcional de dicha plataforma (de relativo bajo costo) que permita formar parte del material didáctico y posibles trabajos futuros dentro de la unidad.

Se realizó una búsqueda abundante de libros, artículos, conferencias, documentación y bibliografías en general de los tipos de control de sistemas, su funcionamiento, características, aplicaciones y alcances, más específicamente de proyectos de esta naturaleza con lo cual se observó que los ya existentes se han desarrollado en entornos de simuladores y extensiones de programas ya preestablecidos, en los cuales solo se ingresan parámetros de la cinemática que le permitirán trabajar dependiendo de las dimensiones y características de la plataforma.

Así mismo, se observó una escasa documentación en bibliografía consultada, en los cuales solo se mencionan los elementos empleados y algunas metodologías, sin embargo, no se recurre al uso de software de programación como lo es el caso de C, Python, Java o algún otro entorno que permita realizar modificaciones directas sobre el código en función de los nuevos objetivos que se le puedan plantear, así como sensores, extensiones o mejoras que se le puedan realizar a este.

Una vez que se ha realizado la evaluación de las características en cada uno de ellos, con la finalidad de detectar semejanzas y diferencias respecto al sistema que se propone en el actual documento, se propondrá un sistema cuya respuesta deberá ser lo suficientemente rápida para estabilizar el *efector final* respecto a la posición horizontal, ante la aplicación de fuerzas externas producidas por objetos deslizantes sobre su superficie y movimientos sobre el plano inclinado. Cabe mencionar que actualmente se han encontrado trabajos realizados en base a sensores sobre la plataforma del *efector final* o *giroscopios* que permitan corregir el error en movimientos de inclinación, sin embargo, el actual trabajo a desarrollar pretende homogeneizar ambos componentes.

Dentro de este planteamiento de posibles soluciones, como ya mencionado, se considera la implementación de dos sensores para la señal de entrada al sistema, los cuales serán un conjunto entre un panel resistivo que permita sensar los movimientos efectuados de objetos deslizantes que ejerzan fuerza aplicadas sobre el *efector final*, así como un sensor giroscopio que permita sensar los movimientos sobre el plano inclinado. En principio se plantea realizar un sistema que incorpore ambos sensores de manera simultánea, sin embargo, al no tener más información sobre las posibilidades para realizar esta operación de manera adecuada, se tiene como opción el incorporar un mecanismo de interrupción que permita pasar de un tipo de sensor a otro.

Una vez simulado el mecanismo de la plataforma *Stewart-Gough* se procede a buscar los componentes necesarios para realizar la construcción y el ensamblaje del Mecanismo. En este caso se han considerado el uso de actuadores angulares, que, aunque en principio pueden llegar a complicar los cálculos de las cinemáticas tanto directa como inversa, son en general más económicos y rápidos que los actuadores lineales de costo semejante. Se debe considerar, que este tipo de actuadores no está diseñado para el sostén de objetos demasiado pesados, sin embargo, es donde entra en juego una de las mayores ventajas de un robot paralelo, debido a que la configuración en el acomodo de los actuadores aumenta la capacidad de carga en el sistema debido a la distribución de la carga en cada actuador dentro de la cadena cinemática.

Las dimensiones y características del mecanismo estarán delimitadas por el diseño tomado de dominio público, así como los actuadores y componentes seleccionados posteriormente. Actualmente se han comparado diversos modelos disponibles en el mercado, a partir de los cuales se consideraron viables las siguientes características [17]:

- Carga: 100gr – 5Kg.
- Dimensiones:
  - Altura: 15 – 30 cm.
  - Laterales: 25 – 30 cm.
- Ángulos:
  - Roll:  $\pm 20^\circ$  –  $45^\circ$ .
  - Pitch:  $\pm 20^\circ$  –  $45^\circ$ .
  - Yaw:  $\pm 10^\circ$  –  $20^\circ$ .
- Cartesiano:
  - X:  $\pm 5$  – 15 cm.
  - Y:  $\pm 5$  – 15 cm.
  - Z: 2 – 5 cm.
- Resolución:  $0.12^\circ$  –  $3^\circ$ .

## 11 DESARROLLO DEL TRABAJO

### 11.1 Selección del Microcontrolador

#### 11.1.1 Definición del tiempo de muestreo

Uno de los principales parámetros a definir dentro de la implementación sistema de control es el periodo de muestreo  $T$ , el cual es un elemento de diseño que permite seleccionar distintos componentes entre los más importantes como lo es el microcontrolador. Existen diversos métodos para su obtención, sin embargo, para este proyecto se ha considerado la selección mediante el método descrito en [18] el cual selecciona el periodo en función del compromiso entre los siguientes factores:

1. El tiempo de cálculo del procesador.
2. Precisión numérica en la implementación.
3. Pérdida de información en el muestreo.
4. Respuesta a perturbaciones.

En general, el periodo de muestreo se debe seleccionar en compromiso entre un rango de tiempo que evite el deterioro de la calidad del control que puede producir un alto valor de  $T$  y la cantidad de cálculos necesaria para ejecutar el algoritmo de control con valores pequeños que puedan producir perdidas de información y traslape de frecuencias (aliasing). En estas se consideran 3 casos a los cuales se puede dar seguimiento con base en los elementos disponibles, la facilidad de cálculos y la índole del proyecto.

- Tomar el ancho de banda del sistema: Este considera el sistema como un sistema de lazo cerrado en el cual se plantean cada uno de los elementos y su respectiva frecuencia. A partir de esto se determina el ancho de banda del sistema que servirá como referencia para la frecuencia base del muestreo.
- Tiempo de establecimiento requerido para la respuesta transitoria: Este método se puede realizar mediante simulación o experimental, ya que considera la obtención de un tiempo de respuesta con base en el alcance del 63.3% del valor final en la respuesta transitoria. En [13] se utilizó un osciloscopio que permitiera medir la curva de respuesta y el tiempo  $T_{ao}$  mediante la aplicación de una señal impulso y un mecanismo adaptado a los actuadores.
- Seleccionar el componente de la frecuencia más alta: Este es un método que permite realizar una estimación del periodo de muestreo tomando como base el componente del sistema que requiera un muestreo con mayor frecuencia. Debido a que actualmente no se han realizado los cálculos de la planta, ni se han adquirido

los componentes para la obtención mediante la respuesta transitoria, se considera a este el método como el más acertado para la estimación del periodo de muestreo.

Ya definido el método se procede a seleccionar el componente que requiera la mayor frecuencia de trabajo. Para la selección se descartaron los actuadores debido a que estos solo requieren de un pulso de PWM con frecuencia menor a 1kHz, por lo tanto, solo se considerarán el sensor giroscopio y el panel resistivo. Como se menciona en la sección de Sensores de Posición y Orientación el MPU-6050 tiene como característica la configuración de captura de movimiento en un rango de escala de 250/500/1000/2000 grados/seg. Es por ello que al requerir de un mayor rango de muestreo que la pantalla resistiva se seleccionará como componente para determinar el tiempo de muestreo.

Para medir los movimientos del efecto final no se considera necesario trabajar a la máxima capacidad del giroscopio debido a que el máximo ángulo de acción en alabeo y cabeceo será de  $\pm 20^\circ$  según [19], ya que la configuración de los actuadores angulares limita el efecto final. Considerando que el sistema realice un movimiento extremo de compensación de  $40^\circ$  programando un muestreo mínimo de 250 los movimientos deberían ser efectuados con una velocidad de 1/250 (0.004) seg, lo cual es aproximado a la velocidad del actuador angular que varía entre 0.0028 y 0.002 seg/g (según el fabricante).

A pesar de tener suficiente tiempo de muestreo se plantea el caso en que se llegará a utilizar el sensor configurado en su máxima capacidad de  $2000^\circ/\text{seg}$  (en cuyo caso los movimientos efectuados sobre éste deberían ser de 0.0005 segundos) para tener un factor de seguridad extra en la selección del muestreo, esto conlleva al siguiente paso que menciona los casos en que se puedan presentar traslapes (aliasing) o pérdidas de información con un periodo demasiado grande.

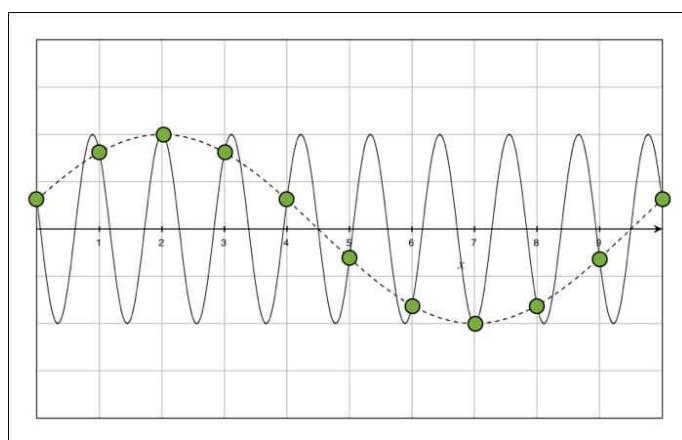


Figura 13 - Traslape de señales (Aliasing).

Para prevenir estos problemas de periodos de muestreo grandes, se aplica el teorema de Nyquist [19] el cual dicta que si la frecuencia de muestreo  $W_s$ , definida como  $2\pi/T$  en donde  $T$  es el periodo de muestreo, es mayor que  $2W_c$  (frecuencia a muestrear) es decir:

$$W_s \geq 2W_c \quad (28)$$

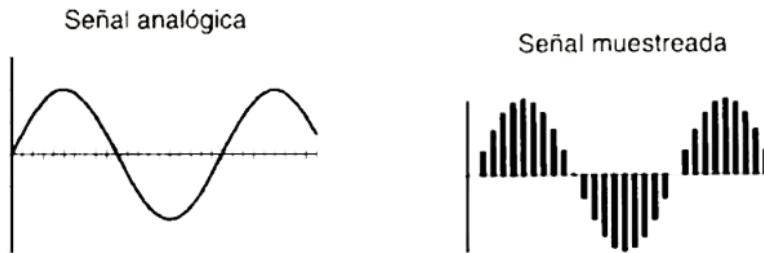


Figura 14 – Señal muestreada mediante Teorema de Nyquist.

A partir de esto, se menciona en [18] que se debe considerar como regla general que el sistema debe ser muestreado entre 8 y 12 veces durante un ciclo de la frecuencia amortiguada de la respuesta transitoria si el sistema es subamortiguado, o entre 8 y 12 veces durante el tiempo de establecimiento de la respuesta si el sistema es sobreamortiguado. En cuyo caso se tomará el mínimo de este factor debido a que se desea obtener un sistema con una respuesta subamortiguado o sobreamortiguado.

Para estimar el periodo de muestreo se aplica el siguiente criterio criterios:

- Si  $W_c$  es el ancho de banda del sistema en lazo cerrado, la frecuencia de muestreo ( $W_s$ ) se puede estimar dentro del intervalo:

$$8w_c \leq w_s \leq 2w_c \quad ; \quad T = \frac{2\pi}{w_s} \quad (29)$$

Ya seleccionado el giroscopio como el componente de mayor frecuencia dentro del sistema, se toma como referencia el mayor rango de medición que provee este dispositivo, y como ya se ha mencionado, el sensor tiene una capacidad de medir de 2000 °/seg y entregarla mediante una señal digital por protocolo I2C se considera como la frecuencia base de muestreo debido a que se está contemplando el caso crítico en el cual se deseé obtener la mayor frecuencia requerida para que el procesador cumpla con dicha tarea.

Al considerar  $W_c = 2000^{\circ}/\text{seg}$  como ancho de banda del sistema en lazo cerrado y al ser un muestreo de ciclo por segundo se puede interpretar directamente proporcional a 2000Hz con lo cual se puede determinar la frecuencia de muestreo sustituyendo en (29) se obtiene el siguiente rango para el tiempo de muestreo:

$$16000 \leq W_s \leq 4000 \quad (30)$$

Es importante señalar que  $W_s = 16$  kHz será la frecuencia de trabajo seleccionada que requiera tener como mínimo el procesador para operar de manera óptima en el muestreo de la señal del giroscopio, esto debido a que más adelante se realizarán consideraciones como la incorporación del muestreo del panel resistivo en el ADC, la frecuencia de trabajo de los actuadores de manera individual y en la cadena cinemática, así como también pueda afectar, o no, el tamaño del programa por las líneas de código debido a que al considerar la implementación un microcontrolador convencional la programación se deberá realizar en serie y no en paralelo.

Una vez que se define un tiempo estimado para el periodo de muestreo, se procede a buscar las opciones de microcontrolador que cumpla dichos requerimientos. Así mismo se pone en comparativa algunas de las características principales que estos presentan y sean requeridas por otros componentes del sistema, tal como lo serían los periféricos de PWM para los 6 actuadores, los puertos de comunicación I2C para el giroscopio, los canales de ADC para el panel resistivo así como su resolución y una característica extra que sería el tipo de montaje debido a la finalidad didáctica de la plataforma, ya que al pretender ser manipulada por alumnos de la institución, frente a cualquier problema que pueda presentarse, los componentes sean de fácil reemplazo.

A continuación, se presenta la siguiente tabla donde se compara dichas características de algunos de los microcontroladores disponibles en el mercado.

*Tabla 2 - Comparación de microcontroladores.*

Comparación de Microcontroladores								
Modelo	Velocidad de Reloj	Puertos I/O	Comunicación I2C	Canales PWM	Canales ADC	Resolución ADC	Voltaje de Operación	Tipo de Montaje
ATMega328P	16MHZ	23	Si	6	6	10 bit	3.3V - 5V	THT
ATMega2560	16MHZ	86	Si	20	16	10 bit	3.3V - 5V	SMD

Actualmente se considera como el mejor candidato al microcontrolador ATMega328P ya que al ser de montaje tipo THT es de fácil reemplazo, además de ser económico y accesible. El microcontrolador incorpora 6 canales de ADC, de los cuales 1 será destinados al panel resistivo y los otros 2 coinciden directamente con los canales SCL y SDA que realizan la comunicación I2C. Además, para el control de los actuadores se requiere de 6 canales de PWM (un canal para cada actuador) que se encuentran distribuidos en los puertos B y D.

## 11.2 Modelo matemático

El mecanismo de la plataforma Stewart, hasta mediados de la década de los 80, mantuvo en su mayoría el diseño de la plataforma triangular unida mediante articulaciones esféricas a tres actuadores lineales ajustables en longitud. Esta configuración unía la base (cuyo diseño puede variar entre una forma triangular o hexagonal) con el efecto final (que mantiene su forma triangular) mediante los actuadores coincidiendo dos a dos en los vértices del efecto final con un total de seis puntos distintos contenidos en el plano de la base. Estas configuraciones se denotan como 3-3 plataforma de Stewart y 6-3 plataformas de Stewart [20], respectivamente como se puede observar en la Figura 15.

Considerando un punto de coincidencia en articulaciones esféricas en la plataforma triangular, restringía la movilidad en el manipulador produciendo sobreesfuerzos y un espacio de trabajo reducido, a partir de ello se obtiene la configuración cuya base conserva la forma hexagonal regular o semirregular y el efecto final se genera a partir de un triángulo equilátero con los vértices recortados conocida como 6-6 *plataforma de Stewart*. Dicho diseño permite variar algunos parámetros y componentes de acuerdo a las necesidades y/o especificaciones, tales como los actuadores que pueden pasar de ser lineales a angulares mediante un mecanismo de acondicionamiento de tipo de biela-manivela. Así mismo, se puede modificar el tipo de unión que para este proyecto se seleccionaron esféricas y no universales.

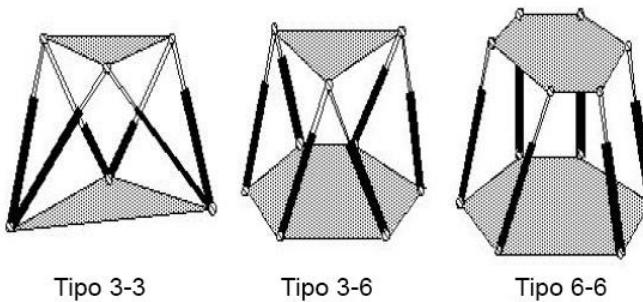


Figura 15 - Distintas configuraciones de la plataforma Stewart-Gough [20].

Una vez visto las distintas configuraciones que puede presentar la estructura de la *Plataforma Stewart-Gough* se seleccionó la configuración 6-6 *Plataforma de Stewart* que permite una mejor manipulación del efecto final y reduce al máximo posible los esfuerzos por singularidades, así mismo, para la aplicación que se requiere no es necesario un amplio espacio de trabajo ya que se plantea realizar movimientos poco agresivos para la estabilización de objetos que se desplacen sobre la superficie, aplicación de fuerzas externas o movimientos sobre el plano inclinado.

A partir de esto, se observa que existe una gran variedad de configuraciones mecánicas para este tipo de manipulador con la cadena cinemática 6-6 *Plataforma de Stewart* según la selección previa. Esta estructura cinemática considera una variante conocida como 6-SPS que se observa en la *Figura 16*, donde 6 se refiere al número de grados de libertad del robot y SPS viene del inglés Spherical-Prismatic-Spherical haciendo referencia a la pierna de punto a punto, donde las articulaciones en los extremos son esféricas y el enlace entre estas es una articulación prismática [20].

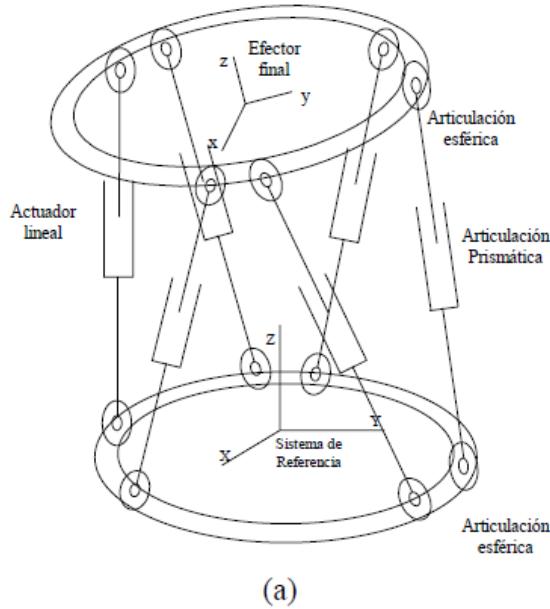


Figura 16 - Plataforma Stewart 6-SPS [20].

Una vez seleccionada esta configuración que permite tener articulaciones esféricas a ambos lados de cada actuador, se realiza una modificación en los actuadores de tal manera que en lugar de una unión prismática entre las articulaciones referente al actuador lineal, se reemplaza por un elemento de enlace entre ambas uniones y el movimiento de extensión pasa a ser generado por una configuración con actuadores angulares como se verá más adelante en el cálculo de la *cinemática inversa*. Quedando así una configuración como la presentada en la *Figura 17*.

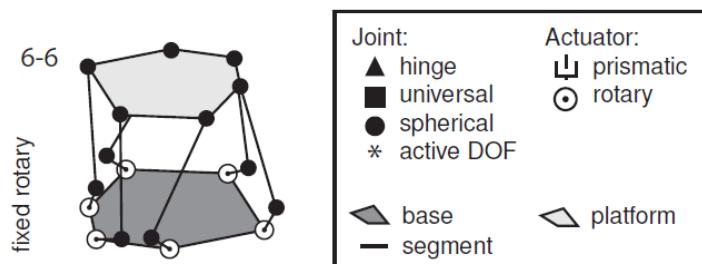


Figura 17 - Plataforma Stewart 6-SPS con actuadores angulares [21].

### 11.2.1 Grados de libertad de la Plataforma Stewart-Gough

Utilizando el criterio de Grübler [22], se puede determinar el número de grados de libertad en una plataforma tipo 6-SPS mediante la

$$m = \lambda(l - j - 1) + \sum_{i=1}^j f_i - I_d \quad (31)$$

Donde:

- $m$  = Grados de libertad del sistema
- $\lambda$  = Grados de libertad del espacio donde el mecanismo está.  $\lambda = 3$  bidimensional y  $\lambda = 6$  para el caso espacial
- $l$  = Número de eslabones fijos del mecanismo, incluyendo la base y el efecto final
- $j$  = Número de articulaciones en el mecanismo
- $f_i$  = Grados de movimiento relativos por junta.
- $I_d$  = Número de grados de libertad pasivos del mecanismo

Sustituyendo en la ecuación (31) se tiene:

$$m = 6(8 - 12 - 1) + \sum_{i=1}^{12} 3 = 6 \quad (32)$$

### 11.2.2 Espacio de trabajo de la Plataforma Stewart-Gough

El espacio de trabajo, también conocido como campo de acción, es el área o volumen espacial que puede describir el robot al llegar a puntos extremos. Este volumen descrito está determinado por las dimensiones, la forma y el movimiento de las articulaciones que lo conforma, así como los grados de libertad (dependiendo de la configuración y tipo de robot) y en algunas ocasiones también puede influir el sistema de control aplicado [23].

A pesar de que el robot tenga un espacio de trabajo definido, no es confirmado que se pueda describir dicho espacio en cualquier orientación. Existe una serie de puntos, generalmente los más extremos y los más próximos al origen que se pueden acceder solo con configuraciones determinadas y algunos otros pueden ser alcanzados en cualquier orientación. Esto como ya se menciona depende mucho del tipo de configuración del robot.

Una de las desventajas que tienen los robots paralelos frente a los seriales es la limitada descripción del espacio de trabajo, esto debido a las restricciones que tiene una articulación sobre otra en la cadena cinemática cerrada. Para el caso de la plataforma Stewart-Gough, se puede realizar la propuesta de un espacio de trabajo descrito por el *efector final* a partir

de los cuales se obtienen las dimensiones de los diferentes elementos la componen: base, juntas, plataforma, piernas, actuadores y cuerno. Por otro lado, se puede obtener un espacio de trabajo supuesto en base a la selección de componentes para evaluar en cada punto de las articulaciones, los actuadores y eslabones, las posibles interferencias que se pueda ocasionar entre sí, así como las posibles restricciones propias de cada junta [24].

Existen diversos métodos para obtener las medidas de cada elemento de la plataforma *Stewart-Gough* ya sea mediante la descripción de trayectorias como lo realizan en [25] generando una propuesta a partir de un radio de acción. Esto propuesta surge debido a que dicha plataforma tiene un diseño enfocado a la cirugía de bypass coronario mínimamente invasiva. También se puede hacer la propuesta de medidas en base a la generación de algoritmos mediante un método observado en [26].

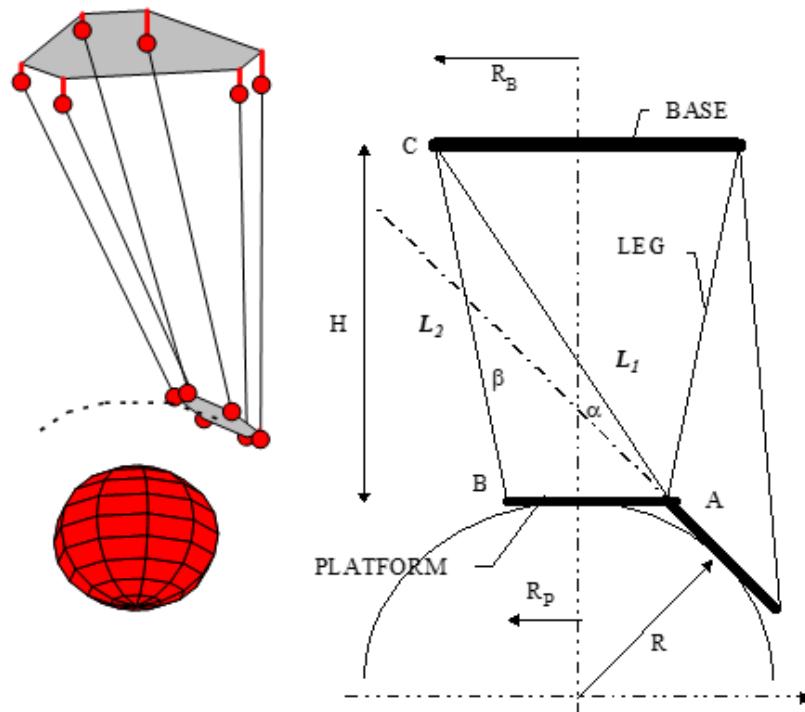


Figura 18 - Descripción de trayectoria de Plataforma Stewart para cirugía CABG de [25].

Varios de estos métodos para describir el espacio de trabajo de una *Plataforma Stewart-Gough* en forma gráfica se basan en la discretización del espacio Cartesiano para luego evaluar la longitud de los eslabones de acuerdo con los actuadores y así determinar que estén dentro del rango, detectar interferencias entre actuadores y/o juntas y las posibles restricciones propias de cada junta. Este método también puede incluir simulaciones mediante la implementación y evaluación de la matriz Jacobiana para llegar a definir una capacidad de movimiento y articulaciones [21].

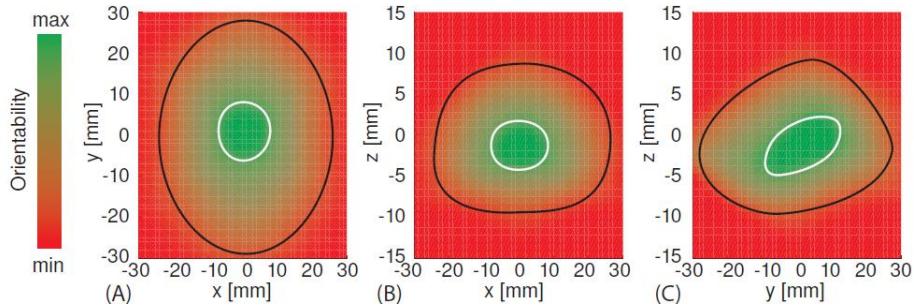


Figura 19 - Espacio de trabajo descrito por la plataforma Stewart-Gough empleando la matriz Jacobiana [21].

A pesar de las variaciones existentes entre modelos de la *Plataforma Stewart-Gough* la cinemática inversa y el espacio de trabajo en general tienen el mismo comportamiento y la misma descripción de trayectorias, como se menciona en el cálculo de la , se puede utilizar la misma para todas sin importar el tipo de actuadores. Esto claro puede variar ligeramente de acuerdo a las dimensiones, la configuración y/o elementos empleados para su elaboración, sin embargo, la sombra descrita por dicho espacio de trabajo se proyecta con gran similitud entre estas.

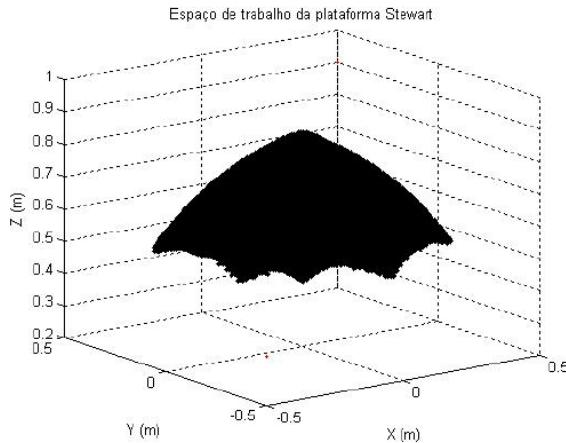


Figura 20 - Espacio de trabajo de la Plataforma Stewart-Gough en vista isométrica [24].

### 11.2.3 Definición de medidas para un espacio de trabajo deseado.

Empleando la metodología planteada en [25] se genera la siguiente *Figura 21* que corresponde a la definición de dimensiones de la *Plataforma Stewart-Gough* haciendo la consideración de que los servomotores están situados paralelos entre sí. Para ello se parte de la definición de la base, que al ser propuesta previamente la *Plataforma Stewart-Gough* con dimensiones aproximadas a un volumen que la encierra de 30-35 cm se establece un diámetro interno coincidente con el eje de los servos, este tiene una medida de 25 cm debido a que los servomotores digitales de medida *estándar* tienen una profundidad de aproximadamente 5 cm.

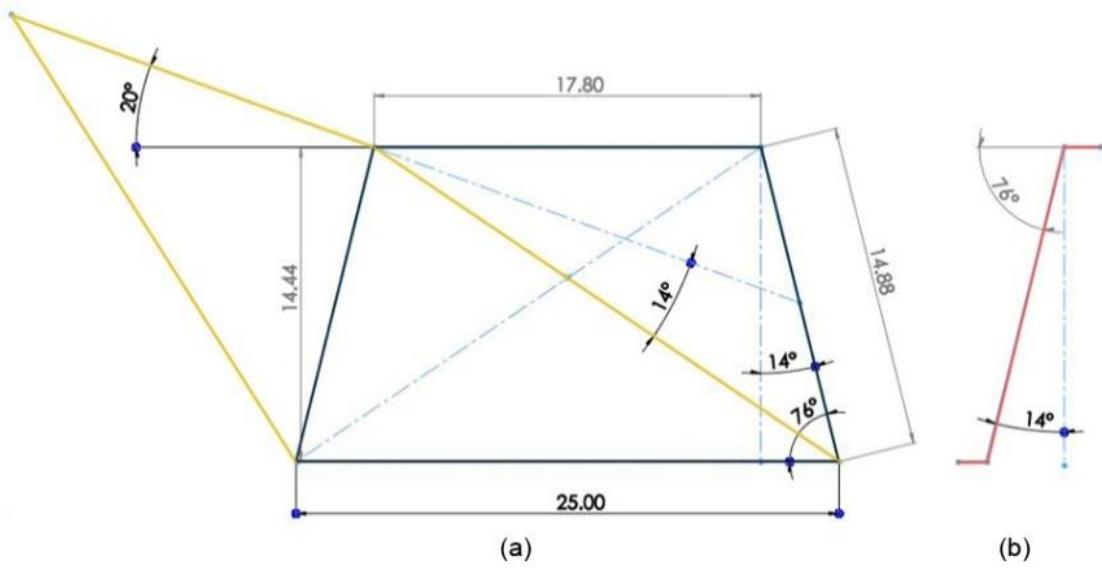


Figura 21 - Dimensiones de la Plataforma Stewart-Gough según espacio de trabajo. (a) Plataforma en vista frontal con actuadores paralelos entre sí. (b) Pierna con juntas esféricas en vista lateral.

Una vez que se define la base se plantea el espacio de trabajo requerido. En principio, el diseño de la plataforma no se requiere para describir trayectorias con demasiada profundidad o alrededor de un radio como se realiza en [25], en este caso los datos importantes a considerar serían los ángulos de inclinación alabeo y cabeceo, ya que serán destinados a la estabilización de objetos deslizantes sobre la superficie de la plataforma.

Para esta situación se propone una de las condiciones del espacio de estados que sería descrita por el trazo amarillo de la Figura 21, donde el ángulo de inclinación se propone de 20° presentes en la mayoría las *Plataformas Stewart* con actuadores angulares documentadas y que se considera suficiente inclinación para la estabilización de objetos deslizándose sobre la superficie como se puede observar en [23].

Finalmente se agregan las posibles restricciones generadas por la articulación de la junta esférica (Figura 21) ya que según la tabla del Anexo F el ángulo de trabajo (dependiendo del tamaño de la junta) ronda entre los  $\pm 13$  y  $\pm 18$ , por lo tanto, dejando un valor de 14° se observa que el ángulo contrario corresponde a 76° que se asignan en la zona interna del trapecio generado con la vista frontal que simularían las restricciones generadas por los demás eslabones perpendiculares a estos.

Es así como finalmente se obtienen los valores del eslabón L=15, el diámetro de la plataforma Dp=18, diámetro de la base Db=25 y una altura en posición de origen de 14.5.

Tabla 3 Comparación entre robots seriales y paralelos [27].

Descripción	Manipuladores paralelos	Manipuladores seriales
Cinemática del manipulador	Lazo cerrado	Lazo abierto
Efectores finales	Plataforma	Gripper
Posición de los actuadores	Cerca de la base fija	En los eslabones
Fuerzas de Inercia/Rigidez	Menores/Alta	Alta/Menores respectivamente
Consideraciones de diseño	Estructura, espacio de trabajo, singularidades, interferencia de los eslabones	Fuerza, rigidez, características de vibración
Propiedad preferida	Rigidez	Destreza
Uso de cinemática directa	Difícil y compleja	Única y sencilla
Uso de cinemática inversa	Única y sencilla	Compleja
Singularidad	Estática	Cinemática
Transformación de fuerza directa	Bien definida y única	No bien definida. Puede ser inexistente, única o infinita
Aplicaciones preferidas	Posicionamiento preciso	Movimiento robusto

La plataforma Stewart es considerada un manipulador paralelo gracias a que la cadena cinemática se cierra con las “articulaciones inferiores” o *rótulas*.

Las partes que mueven en sí a la plataforma se les llama “piernas”, y están compuestas de una varilla, roscada en cada extremo, para que se le coloquen dos rótulas (superior e inferior). Las rótulas inferiores conectan cada una de las varillas con su respectivo actuador (en este caso es el servomotor) por medio de lo que se denominará como cuerno o *brazo de actuador* y los servomotores son considerados parte de la “base fija” o simplemente llamada *base* [28]. Para facilitar la comprensión del lector se presentan en la Figura 22 las partes recién explicadas sobre el modelo CAD diseñado para este proyecto.

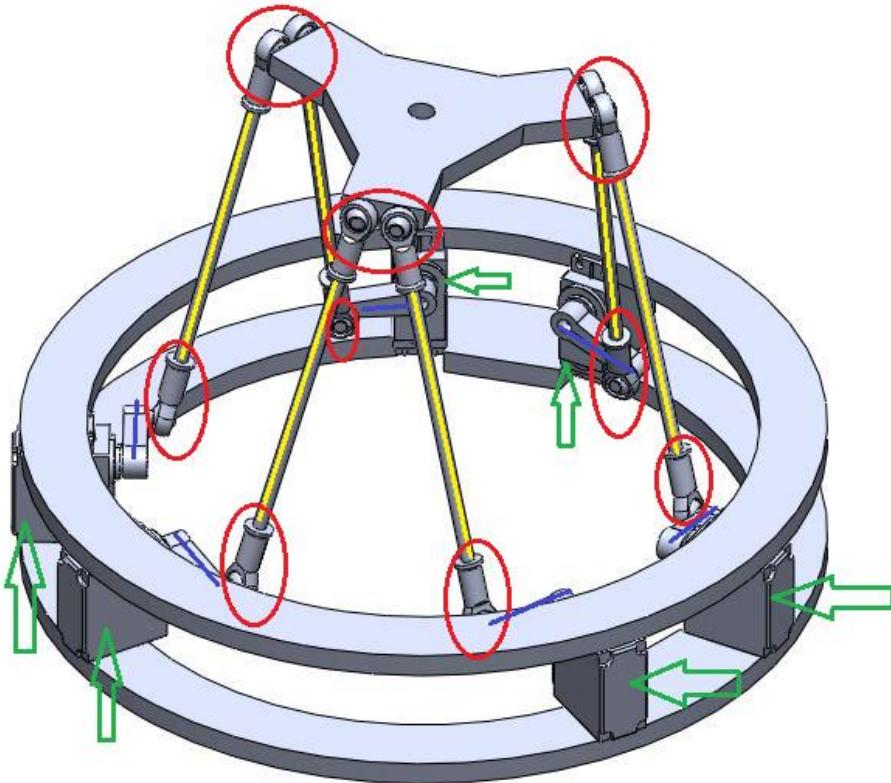


Figura 22 Partes de la base. Rótulas (óvalos rojos), eslabones (líneas amarillas), brazos de actuador o cuernos (línea azul) y servomotores (flechas verdes).

Los brazos tienen un orificio en su centro geométrico utilizado para juntar cada brazo con el eje de su respectivo servomotor. En el extremo del cuerno se acoplan las rótulas inferiores para crear una articulación y unir el cuerno con la varilla (también llamada eslabón), de manera que permita transmitir el movimiento. Tal movimiento es una conversión de movimiento angular a movimiento lineal, esto es una característica diferenciadora de este proyecto a la configuración que tiene la *Plataforma Stewart* clásica.

En el extremo superior de cada una de las varillas existe una unión a otra rótula, de manera similar que en la parte inferior del eslabón. Por lo tanto, según la configuración seleccionada (6-6 SPS) existen otras seis rótulas en la parte superior de la plataforma que mediante la cadena cinemática cerrada conecta la base fija a la plataforma, (también llamado efecto final) siendo este último la parte de mayor interés de cualquier robot serial o paralelo.

#### 11.2.4 Cinemática inversa.

La cinemática planteada anteriormente se cambió al apartado del anexo A5 (16.1.2) debido a que no cumplía con las características necesarias de la plataforma Stewart-Gough en este documento. La cinemática propuesta en este documento se describe a continuación.

La Figura 23 expone una vista superior de la configuración de los puntos  $B_i$  y  $R_{\text{inf}_i}$ . Dichos puntos son proyecciones del eje del servomotor y la punta del cuerno de servomotor. Además, forman parte del círculo verde con centro en el origen del sistema de coordenadas {B}.

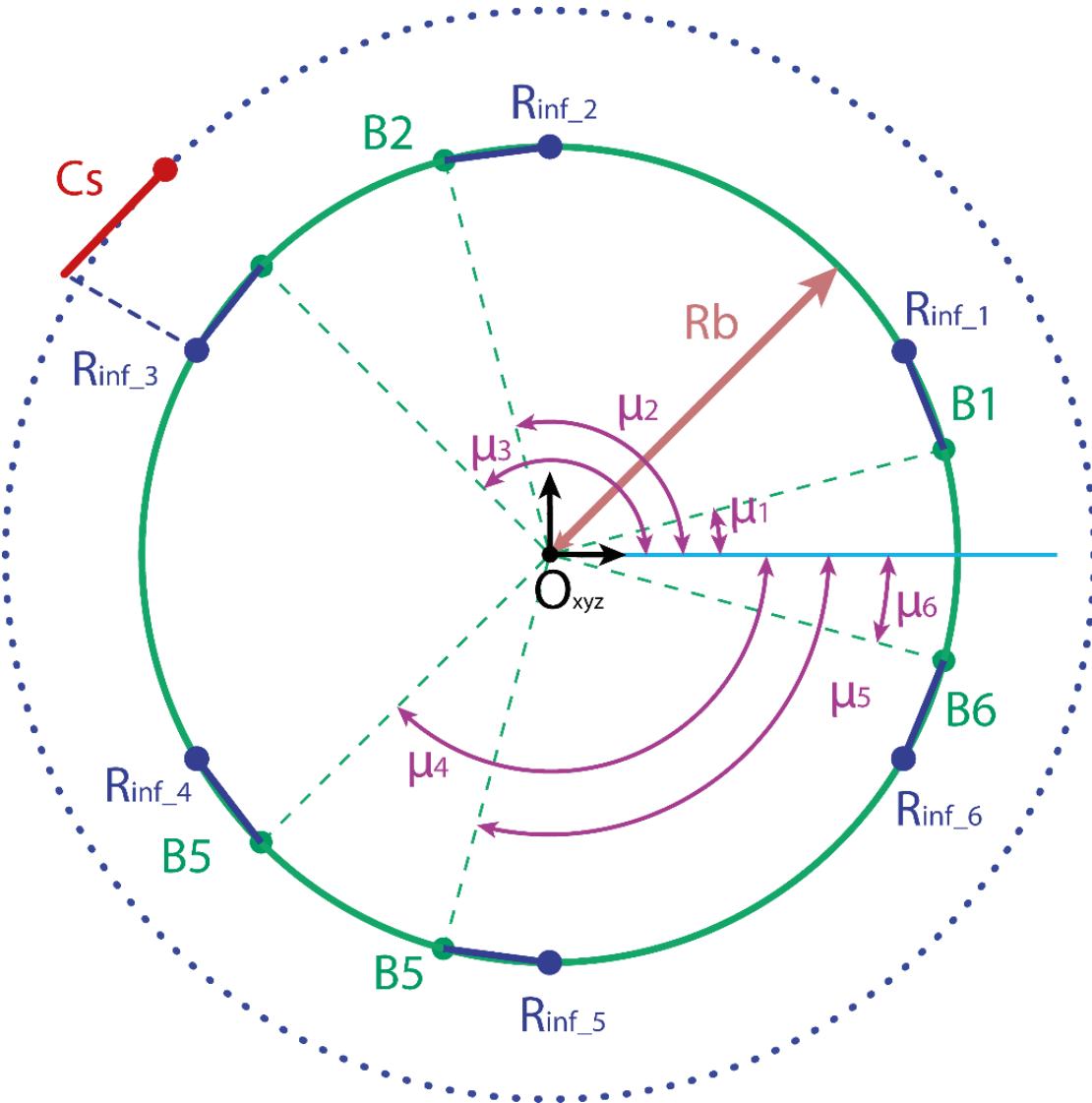


Figura 23 Vista superior configuración puntos  $B_i$ .

Nótese que la línea roja  $Cs$  es la longitud del servomotor.

Cada punto  $B_i$  forma un plano con su respectiva rótula inferior (i.e. los puntos  $B_i$  están posicionados geométricamente a la misma altura de su respectivo punto  $R_{\text{inf}_i}$ ).

De manera análoga, se forma un círculo punteado de color rojo por los puntos  $P_i$ , es decir, las rótulas superiores (Figura 24).

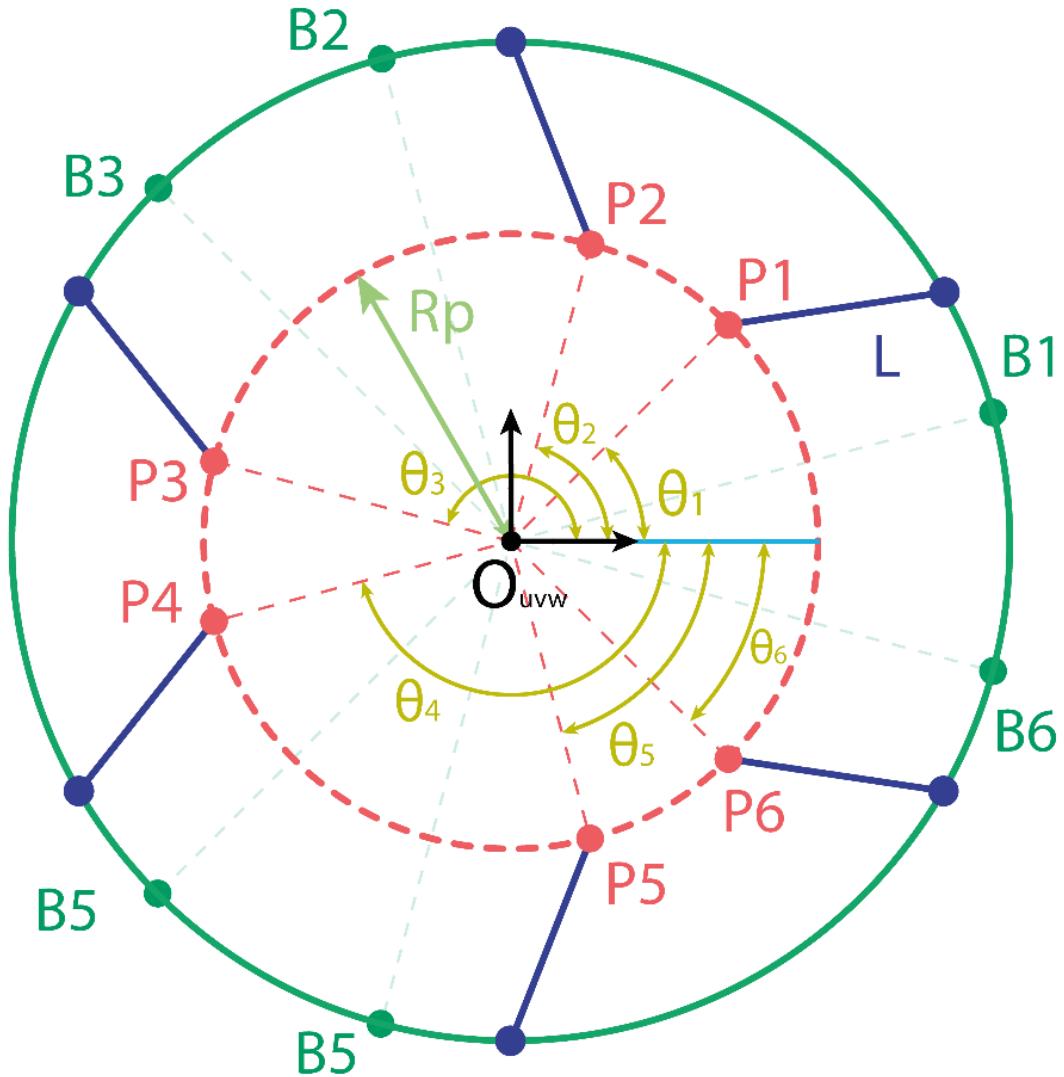


Figura 24 Vista superior configuración puntos  $P_i$ .

#### 11.2.4.1 Matrices de transformación homogénea de posición de marcos de referencia $\{B_i\}$

Se denomina  $H$  como la distancia en el eje Z, cuando la **pose** (posición-orientación) está en cero, es decir,  $X = 0, Y = 0, Z = 0, a = 0, b = 0, g = 0$  (Figura).

Ahora, para los marcos de referencia locales  $\{B_i\}$  se posicionará su origen en los puntos  $B_i$ . Los ejes de dichos marcos serán  $u_i v_i w_i$  haciendo notar que los ejes  $w_i$  son paralelos a su respectivo eje de servomotor. Entonces, se puede deducir que el ángulo de giro  $\square$  de los cuernos serán originados en los planos  $u_i v_i$  del correspondiente marco de referencia  $\{B_i\}$  (Figura 25).

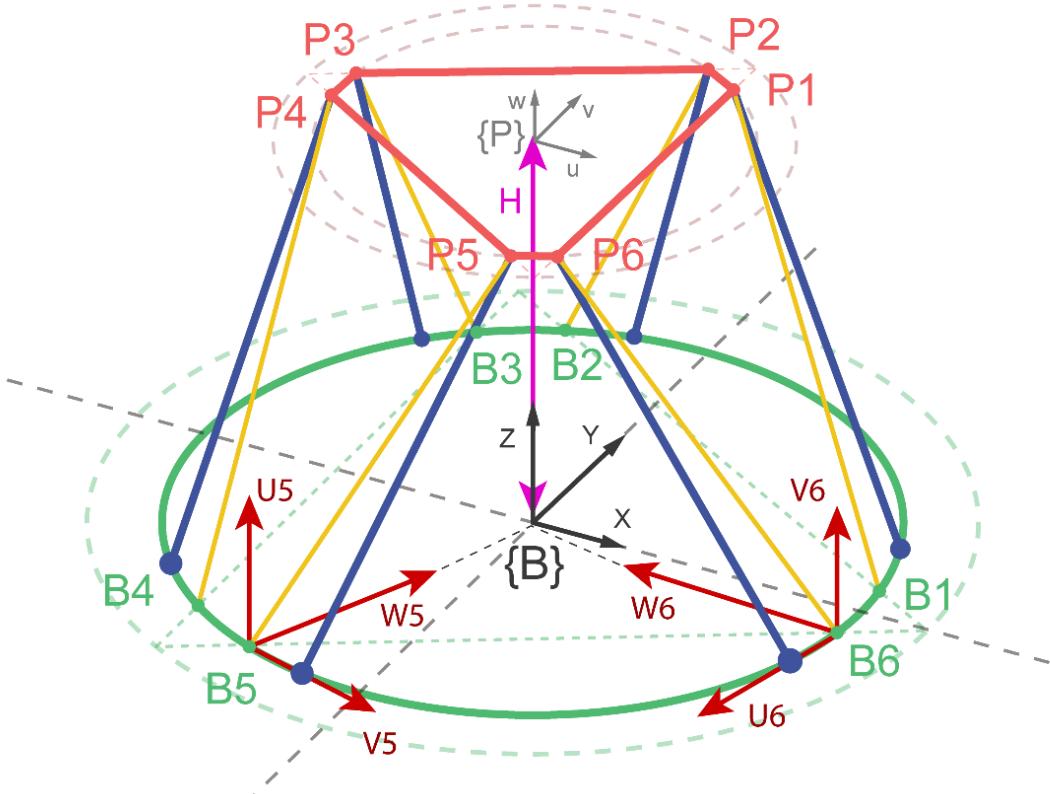


Figura 25 Altura inicial  $H$  y marcos de referencia  $\{B_5\}$  y  $\{B_6\}$ .

La posición de  $\{B_5\}$  está denotada por la composición de matrices expresada en la (figura B1).

$$B_5 = T_{B_5} * R_{(Z, \mu_5)} * R_{(Y, -90^\circ)} \quad (33)$$

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & R_b \cos(\mu_5) \\ 0 & 1 & 0 & R_b \sin(\mu_5) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\mu_5) & -\sin(\mu_5) & 0 & 0 \\ \sin(\mu_5) & \cos(\mu_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(-\frac{p}{2}) & 0 & \sin(-\frac{p}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\frac{p}{2}) & 0 & \cos(-\frac{p}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (34)$$

Por geometría, para  $\{B_2\}$  será como se muestra en la (figura B2):

$$B_6 = T_{B_6} * R_{(Z, \mu_6)} * R_{(X, \frac{p}{2})} \quad (35)$$

$$B_2 = \begin{pmatrix} 1 & 0 & 0 & R_b \cos(\mu_6) \\ 0 & 1 & 0 & R_b \sin(\mu_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\mu_6 - \frac{p}{2}) & -\sin(\mu_6 - \frac{p}{2}) & 0 & 0 \\ \sin(\mu_6 - \frac{p}{2}) & \cos(\mu_6 - \frac{p}{2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ \cos(\frac{p}{2}) & 1 & -\sin(\frac{p}{2}) & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\frac{p}{2}) & 0 & \cos(\frac{p}{2}) & 1 \end{pmatrix} \quad (36)$$

Los marcos de referencia  $\{B_1\}$  y  $\{B_2\}$  son calculados de manera análoga a  $\{B_5\}$ . Asimismo,  $\{B_2\}$  y  $\{B_4\}$  serán calculados de manera similar a  $\{B_6\}$ .

#### 11.2.4.2 Matrices de transformación homogénea de posición de marcos de referencia $\{P_i\}$

En la (Figura 25) se aprecia que los marcos de referencia  $\{P_i\}$  tendrán valor 0 en el eje  $W$ , perteneciente al marco de coordenadas  $\{P\}$ . Entonces, la posición angular de cada posición  $\{P_i\}$  poseerá únicamente coordenadas en el plano  $UV$  de  $\{P\}$ .

Con el fin de obtener la posición de los marcos de referencia  $\{P_i\}$  es necesario trasladar un vector de posición con coordenadas  $(0, 0, H)$ , así como rotar por la matriz homogénea de rotación (que proporcionará la orientación del efecto final por medio de la (ecuación R) ) y, finalmente, la posición del punto  $BP_i$  respecto a  $\{P\}$ :

$$BP_i = BT_P * R * PT_P = BT_P * R * \begin{pmatrix} 1 & 0 & 0 & R_p \cos(\theta_i) \\ 0 & 1 & 0 & R_p \sin(\theta_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (37)$$

#### 11.2.4.3 Posición de rótulas superiores respecto marcos de coordenadas $\{B_i\}$

Se determina la posición de las rótulas respecto a los marcos de referencia  $\{B_i\}$  superiores por medio de la (ecuación) para servomotores noes y la (ecuación) para servomotores pares [29].

$$\begin{pmatrix} V_{1,3,5} \\ U_{1,3,5} \\ W_{1,3,5} \\ 1 \end{pmatrix} = B_{1,3,5}^{-1} * BP_{1,3,5} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (38)$$

$$\begin{pmatrix} U_{2,4,6} \\ V_{2,4,6} \\ W_{2,4,6} \\ 1 \end{pmatrix} = B_{2,4,6}^{-1} * BP_{2,4,6} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (39)$$

Tomando como ejemplo la (figura) que representa los sistemas de coordenadas locales  $\{B_1\}$ ,  $\{B_3\}$  y  $\{B_5\}$ , se define el ángulo  $\sigma$  como el ángulo del servomotor respecto al eje horizontal en el plano  $u_i v_i$ . Para calcularlo, es necesario obtener 4 valores  $L_p$ ,  $L_i$   $S$  y  $d$ .

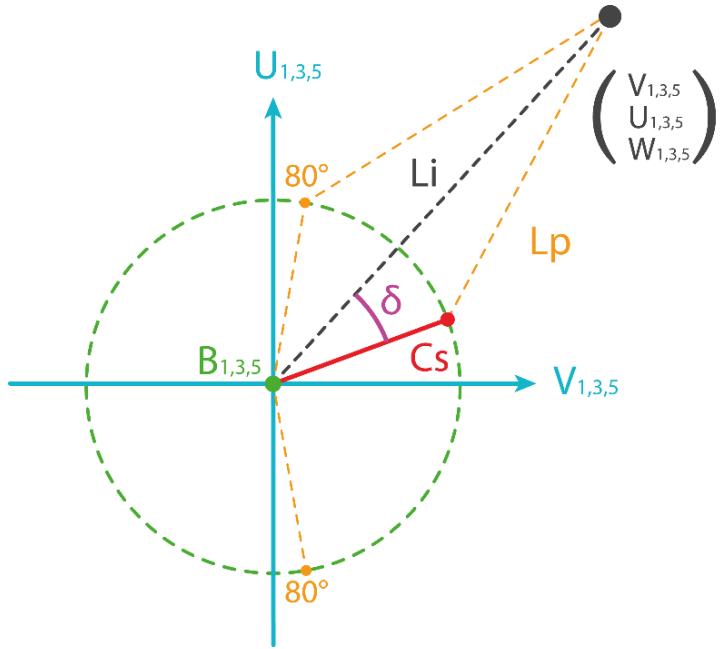


Figura 26 - Plano  $UiVi$  servomotores 1, 3 y 5.

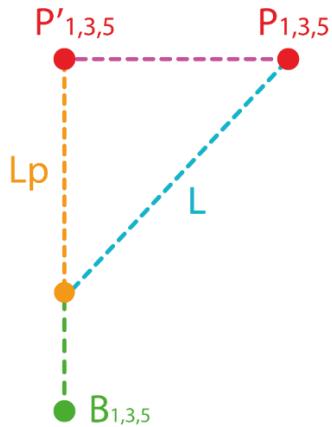


Figura 27 - Plano  $UiWi$  servomotores 1, 3 y 5.

$$L_p = \sqrt{L^2 - L_{1,3,5}^2} \quad (40)$$

$$L_i = \sqrt{u_{1,3,5}^2 + v_{1,3,5}^2} \quad (41)$$

$$S = \tan^{-1} \left( \frac{u_{1,3,5}}{v_{1,3,5}} \right) \quad (42)$$

$$d = \cos^{-1} \left( \frac{C_s^2 + L_i^2 + L_p^2}{2 C_s L_i} \right) \quad (43)$$

#### 11.2.4.4 Datos técnicos de Plataforma Stewart-Gough

Radio de la base:  $R_b = 9.109 \text{ cm}$

Radio de la plataforma:  $R_p = 8 \text{ cm}$

Longitud de cuerno:  $C_s = 8 \text{ cm}$

Longitud de eslabón:  $L = 16.5 \text{ cm}$

Altura inicial en el eje Z:  $H = 16.2 \text{ cm}$

Ángulos  $\mu$ :

$\mu_1 = 19.2^\circ ; \mu_2 = 100.8^\circ ; \mu_3 = 139.2^\circ ; \mu_4 = -139.2^\circ ; \mu_5 = -100.8^\circ ; \mu_6 = -19.2^\circ$

Ángulos  $\theta$ :

$\theta_1 = 51.24^\circ ; \theta_2 = 68.77^\circ ; \theta_3 = 171.24^\circ ; \theta_4 = -171.24^\circ ; \theta_5 = -68.77^\circ ; \theta_6 = -51.24^\circ$

### 11.3 Simulaciones

#### 11.3.1 Simulación del microcontrolador

En la Selección del Microcontrolador se consideraron diversos factores para la determinación del periodo de muestreo, empleando como fundamento diversos métodos tales como el teorema de Nyquist, el teorema de Nyquist modificado y el antialiasing. A partir de ello se obtuvo una frecuencia de muestreo ( $W_s$ ), que como ya se ha mencionado antes, a pesar de ser una frecuencia de trabajo relativamente alta para los requerimientos de los sensores y actuadores del sistema, se decidió tomarla como factor de seguridad debido a que este deberá incorporar otros elementos que puedan retrasar el proceso de cálculos por parte del microcontrolador

Partiendo de una frecuencia de muestreo  $W_s = 16\text{kHz}$  que para la simulación se implementará como frecuencia base ( $W_c$ ) debido a que se debe configurar el ADC para trabajar a dicha frecuencia, se recurre al uso del software *Arduino* empleando el código del anexo G1 (16.7.1) [30], en el cual se configura el ADC para muestrear una señal de 4kHz mediante los siguientes tres métodos:

- Configurar el ADC como Free Running.
- Configurar el ADC para muestrear dependiente de la frecuencia del reloj interno.
- Configurar el ADC para muestrear con determinadas interrupciones generadas por el Timer.

Debido a que el microcontrolador realizará otras funciones además de solo muestrear, y a su vez muestreará distintos sensores de manera simultánea, se consideró adecuado emplear el método de muestreo mediante interrupciones por el Timer, esto debido a que el microcontrolador ATMega328P (candidato seleccionado actualmente) cuenta con dos timer's de los cuales uno puede ser destinado al ADC y otro a operaciones para el control de los Servos mediante PWM.

Siguiendo con el proceso de configuración, se modifican los registros del Timer0 a razón de generar pulsos en el modo *Clear Time Compare (CTC)* como se muestra en la tabla del anexo G1(16.7.1), debido a que en este modo el contador se borra a cero cuando el contador (TCNT0) coincide con el OCR0A, esto permite tener un mejor control de los pulsos y evitar errores acumulativos. Con la intención de dar un margen o factor de seguridad, se decide tomar una frecuencia de muestreo de 18kHz a partir de la cual se emplea la ecuación (44) de [31] para la selección del OCR0:

$$f_{OCnx} = \frac{C_{clk\_I/0}}{2.N.(1 + OCRnx)} \quad (44)$$

La ecuación anterior, corresponde a la frecuencia del OCnx, donde  $C_{clk\_I/0}$  corresponde a la frecuencia del reloj interno (16MHz), N es el prescalador seleccionado (8, 64, 256 o 1024) y  $OCRnx$  es el valor configurado para la interrupción. Sin embargo, para este caso se desea obtener el valor de  $OCRnx$  a partir de la frecuencia dada, por lo tanto, se puede replantear la como se muestra en la ecuación (45):

$$OCRnx = \frac{C_{clk\_I/0}}{N.2.OCRnx} - 1 \quad (45)$$

Al sustituir los valores, para los distintos prescaladores se obtiene la siguiente tabla:

Tabla 4 - Selección del Pre-escalador para frecuencia Ws=36kHz.

Pre-escalador	OCRnx	OCRnx Entero
1	221.22	222
8	26.77	27
64	2.47	2
256	-0.1319	0
1024	-0.07829	0

Debido a que se recomienda mantener un OCRnx relativamente alto, se puede utilizar un prescalador de 8 o 1 bit. Por la resolución y el margen mínimo de error que se obtiene, se determinó utilizar el prescalador de 8 bits, tomando como referencia el OCRnx en valor entero 27. Seguido de esto, se configura el ADC para muestrear a máxima resolución del microcontrolador, tomando la interrupción del timer configurado en modo CTC para generar las interrupciones por desbordamiento con el registro TIFR0 y el comparador OCF0A.

Como resultado de la simulación se muestra la siguiente figura donde se compara mediante el uso del osciloscopio la señal sinodal del generador contra las generadas por medio del Timer (en color azul), y el ADC muestreando las interrupciones del Timer (en color morado). Se puede observar que la señal de los flancos del timer coincide directamente en los puntos de la cresta y el valle de la señal sinodal, por lo tanto, podría existir algún aliasing, sin embargo, esta frecuencia de muestreo ya tiene un amplio factor de seguridad. Seguido de esto se observan los flancos del ADC ligeramente desfasados del timer, esto se comprende por el ligero retraso que pueda haber entre instrucciones.

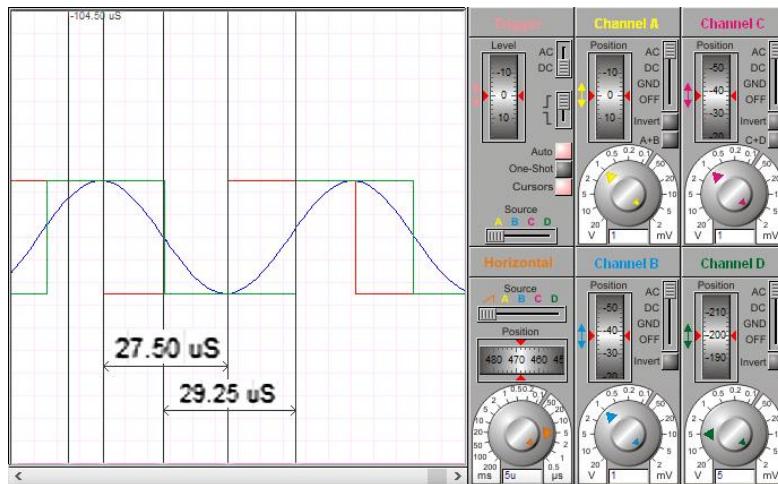


Figura 28 - Simulación de ADC de microcontrolador ATMEGA328P.

### 11.3.2 Simulación de movimientos y articulaciones

Con la finalidad de corroborar que las medidas propuestas cumplan con las condiciones propuestas para las articulaciones y movimientos de la *Plataforma Stewart-Gough* y se asemejen a las características descritas en la sección de *Planteamiento del Problema* se realizó la simulación mediante un software de programación matricial. Para las simulaciones de movimientos y articulaciones se utilizó un programa descargado del foro oficial de MathWorks, cuyo autor de alias Hannes Bartle [32] creó un código que simula la cinemática inversa para la obtención de la posición de los ángulos de los servomotores. Tal código se modificó para agregar la cinemática directa como se muestra en la Figura 29:

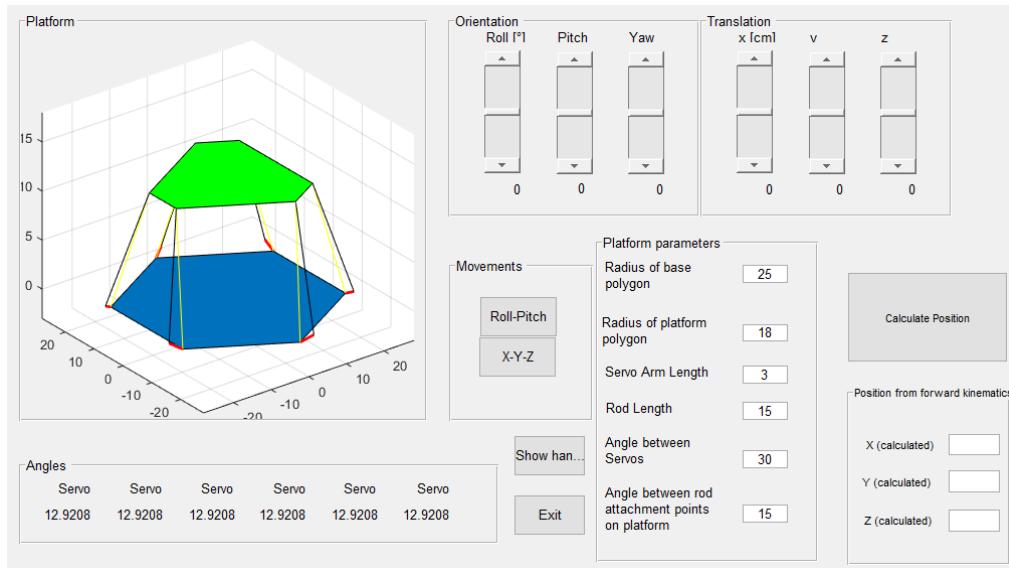


Figura 29 - Interfaz gráfica de simulación de articulaciones (modificada).

El código de esta simulación se muestra en el anexo G2 (16.7.2).

Se considera que la pantalla resistiva a seleccionar es parte de la superficie de la plataforma sobre la que deslizará el objeto a controlar su posición. Para controlar la posición, se necesita que los ángulos alabeo y cabeceo sean aproximadamente de  $20^\circ$  como mínimo para cada uno [19].

Obsérvese en la *Figura 30* el valor máximo aproximado para el giro respecto al eje *U* (perteneciente al sistema móvil de coordenadas  $\{P\}$ , al igual que los ejes *V* y *W*) es de  $21.25^\circ$ . Este giro es llamado **alabeo** o **roll** como se conoce en inglés.

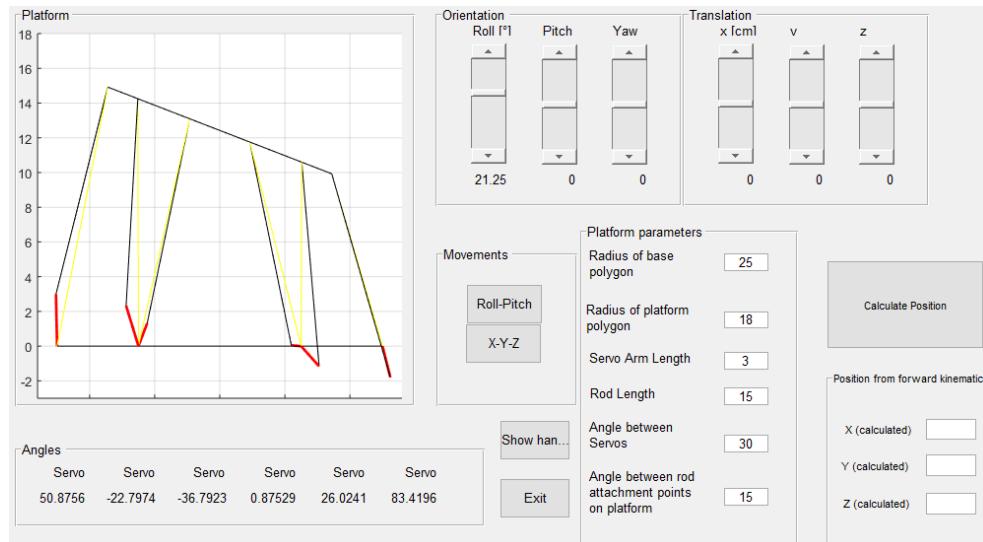


Figura 30 - Valor máximo aproximado de ángulo de ‘alabeo’.

Si se analiza visualmente la gráfica de la figura anterior, en la parte de la interfaz descrita como *angles*, se observan los valores angulares de cada servomotor. Para el sexto actuador (contando de izquierda a derecha) se calculó un valor aproximado de  $83.4196^\circ$  y, debido a los otros ángulos con valores cercanos a cero (el segundo, tercero y cuarto servomotor) la simulación grafica la plataforma con esta configuración de *alabeo* como valor aproximado máximo. Cabe destacar, que este ángulo máximo obtenido solo es válido si se modifica únicamente el *alabeo*.

Ahora, si se aumenta el ángulo máximo de *alabeo*, ya no es posible por restricciones geométricas de la plataforma conocidas como singularidades. Esto se aprecia en el valor del servomotor que tiene fondo color rojo, e indica que es físicamente imposible que el servomotor alcance ese valor. Por lo tanto, la plataforma no podrá obtener el valor de  $21.56^\circ$  de *alabeo* (véase Figura 31).

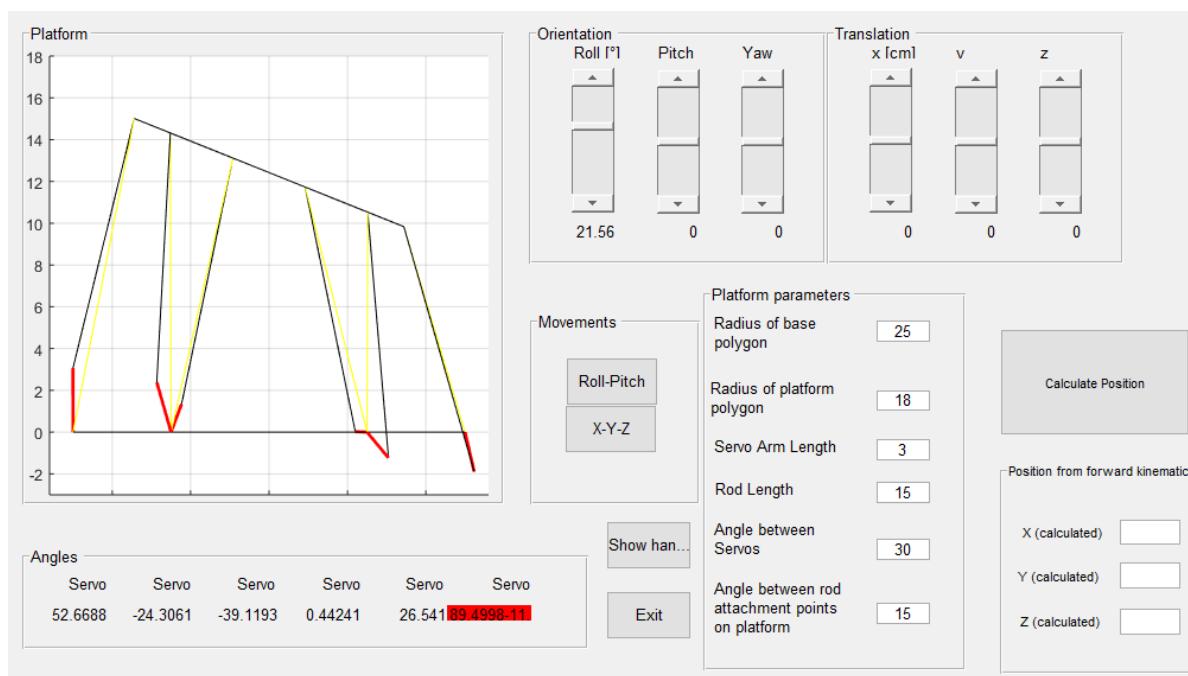


Figura 31 - Valor de ángulo crítico para alabeo.

De manera análoga nótense que la *Figura 32* presenta valor máximo y crítico para el *cabeceo* de la plataforma (ángulo correspondiente al giro respecto al eje *V*, también llamado *pitch*).

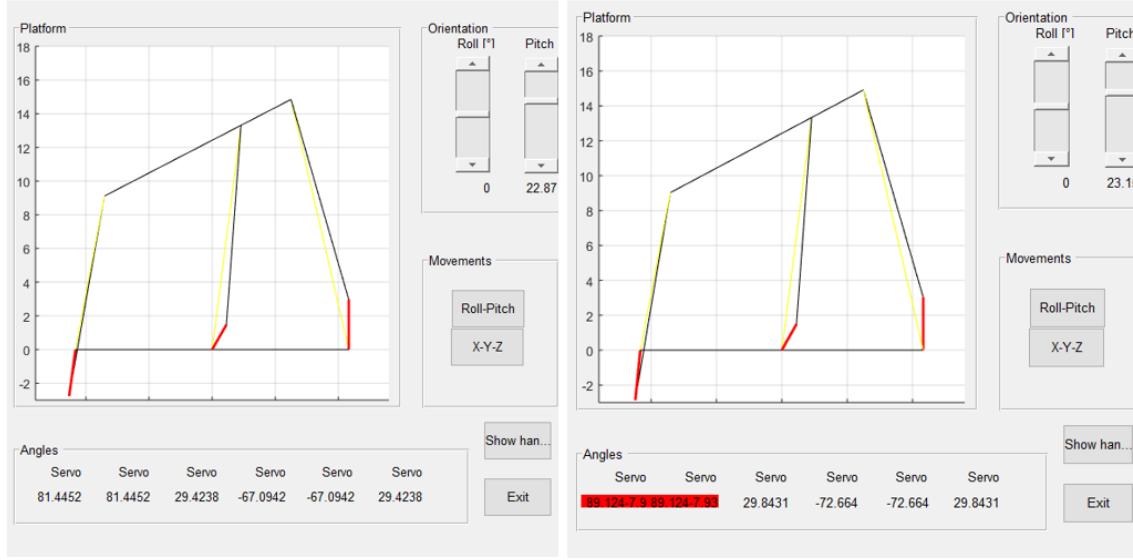


Figura 32 - Valores máximo y crítico de ángulo de cabeceo.

Como se puede observar los valores de *alabeo* y *cabeceo* son los mínimos necesarios para efectuar el movimiento de la plataforma con el fin de llevar el objeto deslizante a una posición deseada sobre el panel resistivo. El ángulo exhibido como el giro respecto al eje *W* del sistema  $\{P\}$  (mejor conocido como *guiñada* o *yaw*) alcanza un valor máximo de  $10.8^\circ$  (Figura 33).

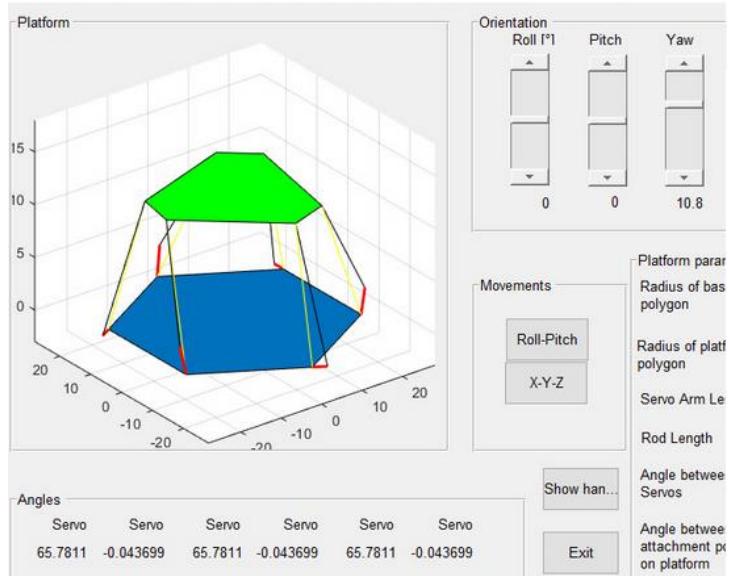


Figura 33 - Ángulo máximo de guiñada.

Las figuras 46, 47 y 48 se señalan las distancias máximas positivas que alcanza el robot sobre cada eje del sistema de coordenadas  $\{B\}$ , es decir, los ejes X, Y y Z.

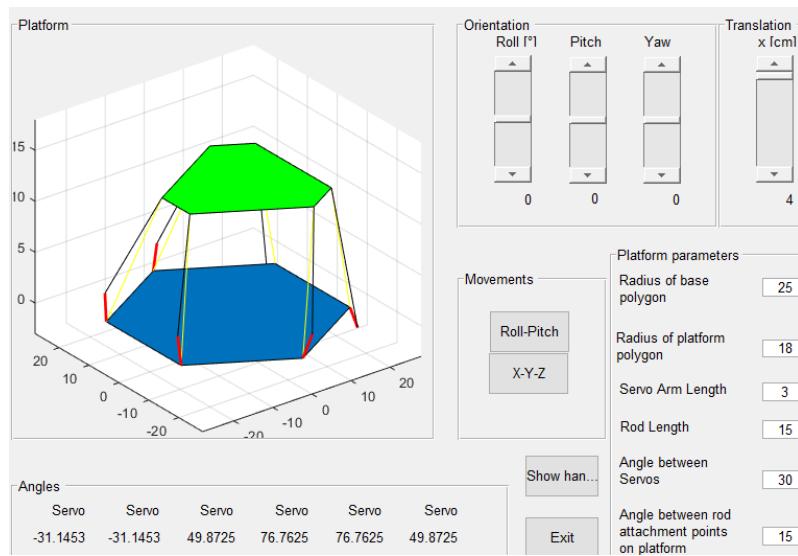


Figura 34 - Longitud máxima de la plataforma sobre eje X.

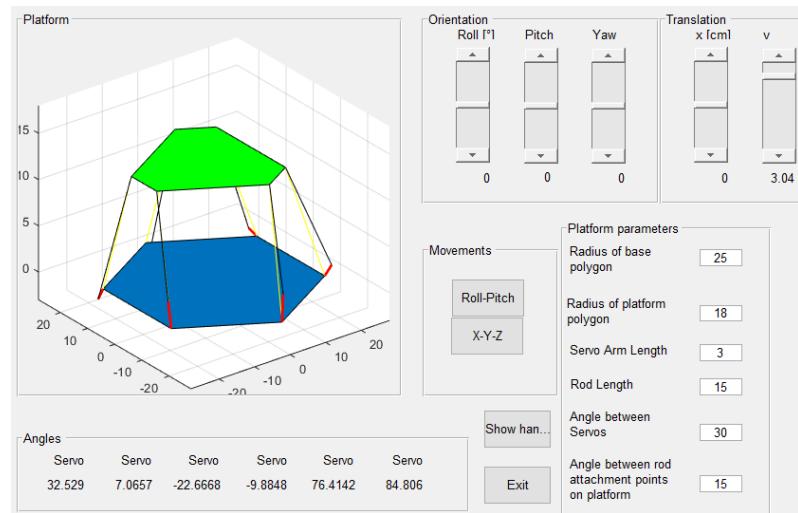


Figura 35 - Longitud máxima de la plataforma sobre eje Y.

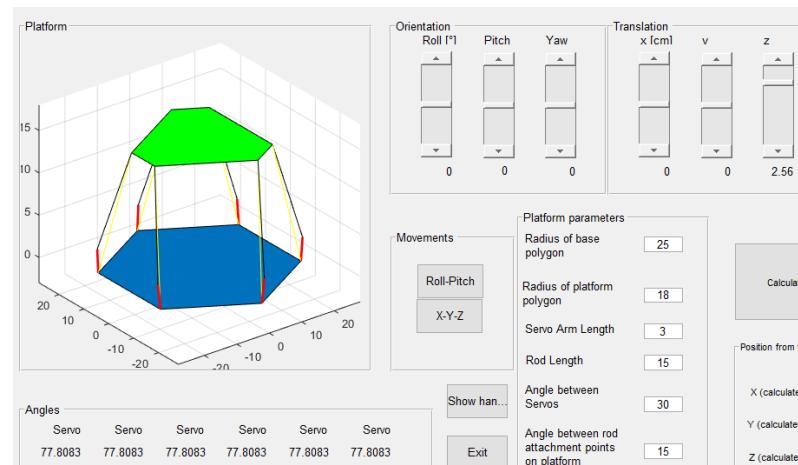


Figura 36 - Longitud máxima de la plataforma sobre eje Z.

### 11.3.3 Simulación de fuerzas externas

Para el proceso de simulación de fuerzas externas aplicadas a la plataforma Stewart-Gough se empleó el diseño CAD generado a partir de las medidas obtenidas en la sección 11.2.3 a partir del cual se hizo una estimación de las medidas para la base, plataforma, eslabones y cuerno del servo. Mientras que para los servomotores y las juntas esféricas se utilizó un modelo estándar tomando un diseño CAD del dominio público.

Las siguientes simulaciones se generaron a partir de una estimación de materiales aproximados a los reales, esto debido a que los proveedores de dichos productos no especifican el material empleado. Para obtener resultados más aproximados a la realidad, se le asignó a cada elemento del ensamble general su respectivo material, como ya mencionado, estimando el polímero o la aleación específica de cada metal.

Para obtener resultados más aproximados a la realidad, se le asignó a cada elemento del ensamble general su respectivo material, como ya mencionado, estimando el polímero o la aleación específica de cada metal. Comenzando con las bases y el efecto final, en cuyo caso se le asignó un plástico ABS dado que, por practicidad y rapidez, se consideran actualmente las opciones de corte láser o impresión 3D, así mismo, los servomotores también se consideran en ABS. Para las juntas y eslabones, la mayoría de proveedores maneja productos metálicos para carros RC, por lo tanto, se asignó un acero 1020 por su relativo bajo costo y facilidad de maquinar. Finalmente, para el cuerno del servomotor se utiliza en su mayoría aluminio y por ello mismo se le asigna aluminio de aleación 1060.

Una vez determinado cada material, se procede a definir el resto de las condiciones de simulación comenzando con cada restricción del mecanismo, que una vez definidas todas sus relaciones de posición, la única restricción geométrica será la base en su cara inferior simulando que dicha plataforma está empotrada sobre una superficie regular horizontal. Para la aplicación de fuerzas aplicadas se realiza una estimación de carga distribuida sobre el *efector final* considerando que en promedio cada actuador (dependiendo del fabricante) puede soportar de manera nominal de 15kg/cm de torque, y al ser los cuernos del servo de 3 cm, se considera que cada actuador soporta en su extremo un total de 5kg, por lo cual se aplica una fuerza total de 30kg

Nombre del modelo:Plataforma\_Stewart  
 Nombre de estudio:Estudio\_Fuerzas\_Aplicadas-(Predeterminado)-  
 Tipo de resultado: Análisis estático tensión nodal Tensiones1  
 Escala de deformación: 333.9

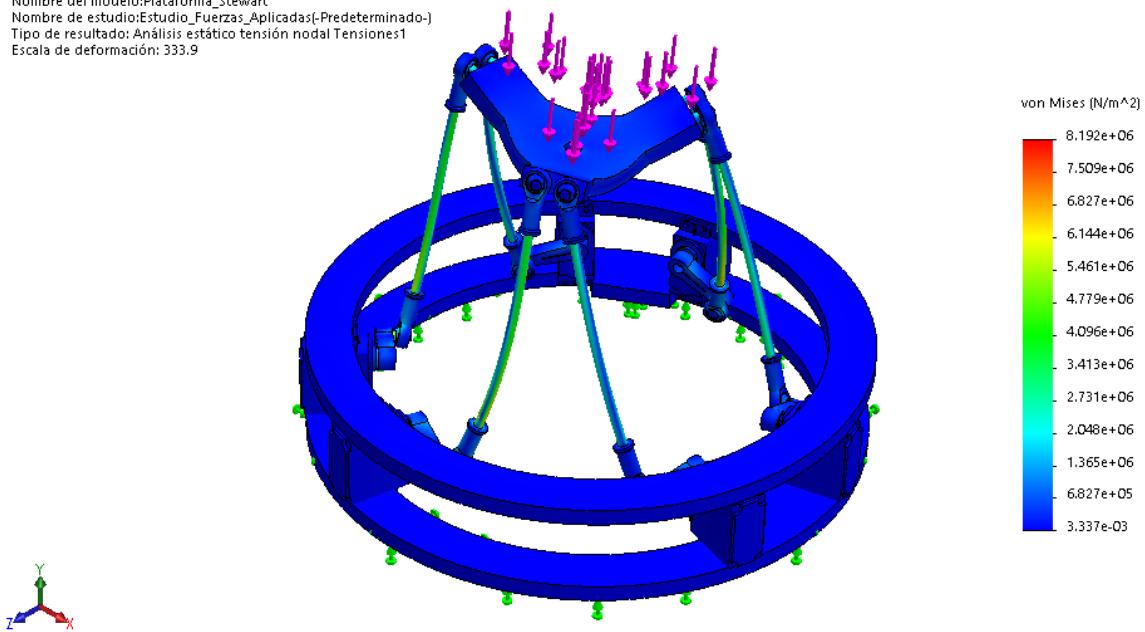


Figura 37 - Estudio de estrés von-Mises.

En la anterior figura se puede observar que dentro del mecanismo en general la mayor parte del esfuerzo se genera en las piernas de la plataforma, mientras que el mayor concentrador casi imperceptible está en la unión entre la pierna y el cuerno del servomotor, justo donde se aplica el mayor torque debido al uso de actuadores angulares. A pesar de esto, se puede observar dentro del gráfico que no existe un esfuerzo crítico que pueda representar un riesgo de ruptura.

Para el siguiente estudio, se contempla la deformación en cada parte de la plataforma, nótese que tanto en el estudio de estrés como el de deformación los elementos correspondientes a la base no sufren cambios significativos, sin embargo, la plataforma y los eslabones se deforman de manera considerable especialmente la parte del *efector final* que puede llegar a deformarse casi en su totalidad. Sin embargo, cabe mencionar que el software de diseño CAD en su mayoría exagera los resultados obtenidos de la simulación, esto para apreciar de manera clara los efectos de las condiciones aplicadas, ya que en un caso real los materiales tenderían más a fracturarse que a deformarse debido a su baja ductilidad. En cuyo caso, la mayor deformación presente es de apenas 0.096mm.

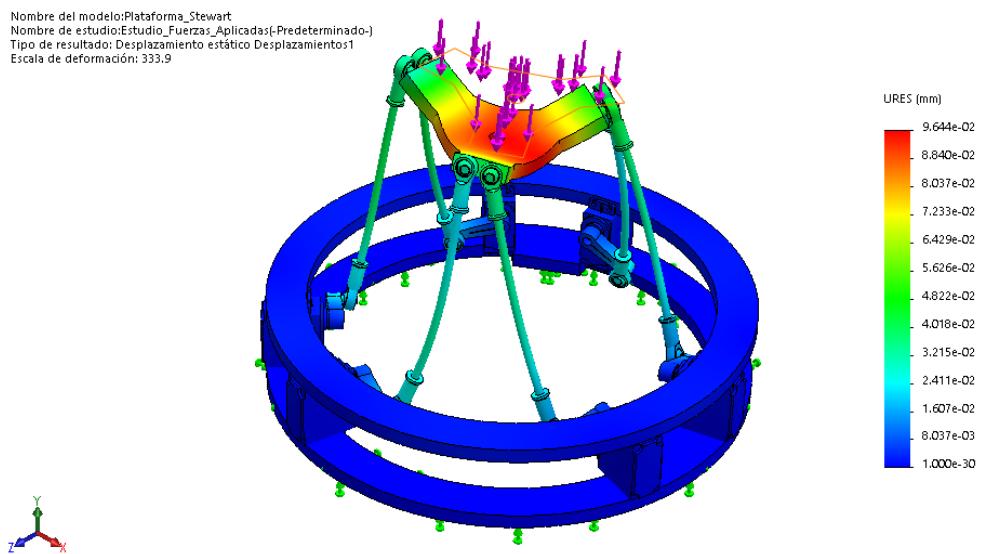


Figura 38 - Estudio de deformación.

A partir de los puntos anteriores se estudia el factor de seguridad, donde se puede apreciar de manera clara los elementos del mecanismo donde se aplica sin contemplar los elementos de la base, que como se observó en los estudios anteriores, no tiene tendencia a ruptura debido a que se contempla empotrada a una superficie fija y reforzada entre sí. Por lo tanto, en las piernas, que son importantes para el desempeño del mecanismo se presenta un factor de seguridad de 3.4, que para la aplicación que se le dará a la plataforma *Stewart-Gough* es un valor aceptable.

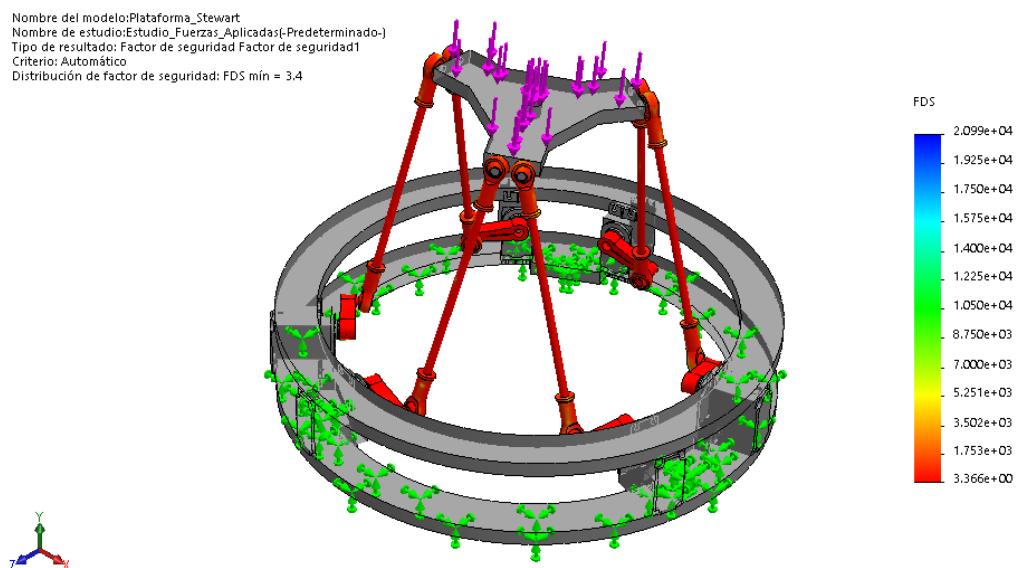


Figura 39 - Estudio del factor de seguridad.

Finalmente se realiza el estudio de esfuerzos aplicados en la situación en que la plataforma tenga un máximo de carga aplicado sobre la pantalla resistiva en uno de sus vértices, esto principalmente se contempla (adelantándose un poco a la selección de componentes) para saber que tanto puede resistir la pantalla hasta el punto de ruptura que en cuyo caso sería el elemento en la plataforma con mayor fragilidad. Como se observa en la Figura 40 el factor de seguridad se queda bajo en 0.45 y tiene una deformación de 1.12 cm que en cuyo caso, el software CAD deforma la pieza para demostrar el desplazamiento, sin embargo, en realidad el panel tendería a fracturarse antes de tener tal deformación.

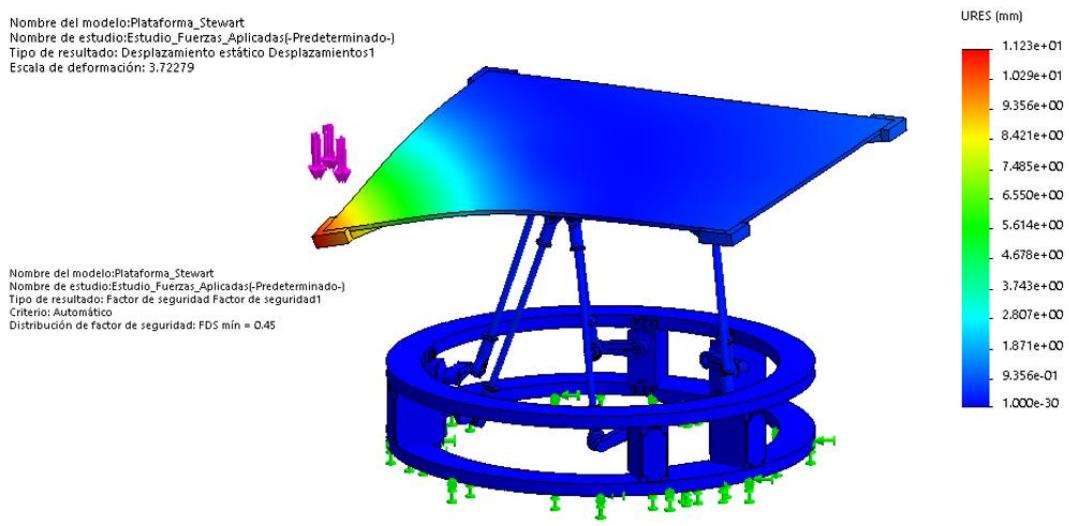


Figura 40 - Estudio de deformación y esfuerzos de la Plataforma Stewart-Gough aplicado al panel resistivo

## 11.4 Selección de componentes

La selección de componentes toma como fundamento los puntos vistos en secciones anteriores ya que se desea cumplir con dichas características tales como velocidad de cálculo, movilidad y espacio de trabajo. A su vez es un punto fundamental para la definición de características. Cabe mencionar que se tiene prevista la selección de componentes contemplando su facilidad de reemplazo siendo relativamente económicos y accesibles, esto con la intención de ser utilizados con audacia y sin limitante en su suplencia.

### 11.4.1 Microcontrolador

La labor del microcontrolador dentro del proyecto es la de procesar los cálculos requeridos para el controlador PID empleando las cinemáticas (inversa y directa) con los datos obtenidos mediante los sensores, para posteriormente proporcionar la señal correspondiente a los actuadores. Así mismo el sistema del microcontrolador tiene la función de proporcionar los periféricos necesarios para dichos sensores, actuadores y algún posible aditamento que se le incorpore al sistema general.

Como se menciona en la *Selección del Microcontrolador*, se considera emplear el ATMega328P como alma de este proyecto. En el anexo B1 (16.2.1) se presentan las características de este microcontrolador.

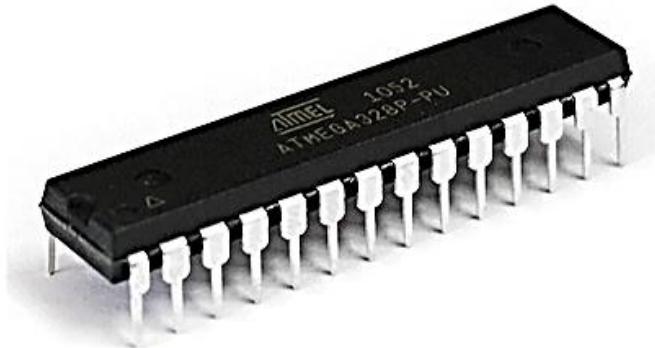


Figura 41 - Microcontrolador ATMega328P en encapsulado THT [31].

Debido al manejo personalizado de los pines y funciones del microcontrolador se tiene previsto implementarlo dentro de un sistema mínimo, es por ello que el microcontrolador se implementará en la versión THT que además de su fácil reemplazo en caso de daño, proporciona la posibilidad de alcance a características limitadas en placas de desarrollo prefabricadas. Dicho sistema mínimo tendrá la función de proporcionar los periféricos correspondientes a las entradas y salidas del sistema, así como fusibles de protección y posibles aditamentos que sean requeridos conforme al desarrollo del proyecto.

## 11.4.2 Sensores

### 11.4.2.1 Giroscopio-Acelerómetro MPU-6050

Con la intención de obtener la posición real sobre el plano inclinado se tiene previsto implementar el sensor MPU-6050 compuesto por 3 acelerómetros y 3 giroscopios, cada uno con su respectivo ADC de 16 bits. Este sensor tiene la posibilidad de modificar su configuración para obtener una mejor precisión respecto a movimientos lentos o rápidos [33] como se muestra en la Tabla 1.

El sensor emplea el protocolo I2C para la comunicación con el microcontrolador esto es una característica definida por el fabricante y se realiza mediante los pines de SCL y SDA que se encuentran presentes en los puestos PC4 y PC5. Puede alimentarse con fuente externa o directamente desde el microcontrolador ya su voltaje de operación es 3.3~5V, cuenta con regulador de voltaje en la placa, tiene seis grados de libertad correspondientes al acelerómetro y el giroscopio con una sensibilidad de 131 LSBs/dps.

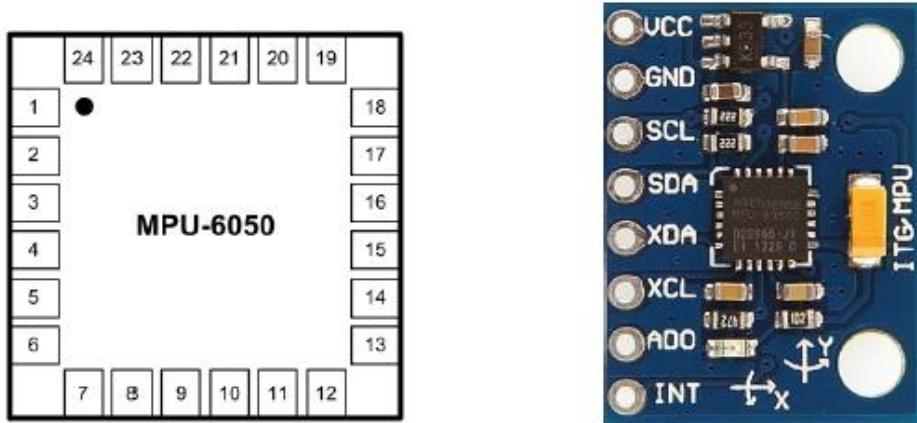


Figura 42 - Pines de conexión del sensor MPU-6050.

#### 11.4.2.2 Panel Resistivo

Para sensar el movimiento de objetos deslizantes sobre la superficie del efecto final se utilizará un panel resistivo en cuyo caso, al ser un componente comercial con gran versatilidad de usos como *panel touch* de dispositivos móviles, pantallas digitales y otros dispositivos electrónicos que empleen interfaz gráfica, existe una gran variedad para su selección. Entre sus características para selección se encuentran las dimensiones y el número de hilos (cuatro, cinco u ocho hilos) de los cuales varía su configuración como sensor.

Para la comunicación con el microcontrolador, en el caso de la pantalla de cinco hilos se utilizarán cinco puertos, de los cuales cuatro serán pines digitales (distribuidos en el puerto PB y PD) para energizar los buses laterales que generan el gradiente de voltaje. El pin restante (generalmente situado en el centro) es el pin sensor encargado de determinar la posición del objeto sobre el panel de acuerdo con el valor de voltaje entre 0-5V comutado por los ejes X y Y.

Frente a la gran diversidad de productos para este elemento, el parámetro para su selección se basó principalmente en la configuración de 5 hilos y posteriormente en su proximidad a las medidas a las de la base, esto con la intención de que la plataforma en general no pierda estabilidad, proporcione un rango amplio de sensado y la *Plataforma Stewart-Gough* abarque un volumen aproximado de 30cm<sup>2</sup>. Se encontró un producto con dichas características en la plataforma web *Aliexpress* con el proveedor *Osatych*.

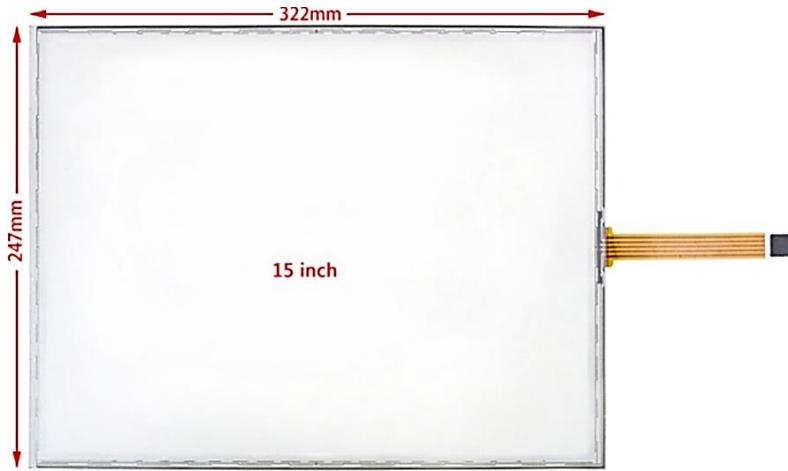


Figura 43 - Panel resistivo producto de Osatych.

#### 11.4.3 Actuadores

Encargados de la movilidad de las articulaciones y descritos a lo largo del documento los actuadores angulares son parte fundamental del trabajo a desarrollar. Para la selección de estos se tiene como parámetros definidos la medida de servo *estándar* y engranes metálicos, esto para evitar el desgaste temprano de actuador. La velocidad de estos es uno de los parámetros más importantes como ya se mencionó anteriormente, sin embargo, se observó que la mayoría de los servos trabaja a una velocidad de 12seg/60°-17seg/60° dependiendo del voltaje generalmente entre 4V-7V teniendo una variación tan solo de 0.002seg/°-0.0028seg/°.

##### 11.4.3.1 Servomotor y cuerno

Para este componente se encontró en la plataforma web *Aliexpress* con el proveedor *ShenZhen* un servo digital de 25Kg/cm de torque modelo DS3225 resistente al agua engranaje de metal. Este servomotor es empleado generalmente para autos a escala RC, sin embargo (al igual que cualquier servo comercial), puede ser utilizado de manera independiente al controlador utilizando el microcontrolador como sistema de control mediante pulsos que pueden variar dependiendo del fabricante, más específicamente este modelo maneja las siguientes características:

- Voltaje de operación: 5 ~ 6 DC Volts.  
5 ~ 7.4 DC Volts.
- Marca muerta: 3µs / 2 µs.
- Peso: 60 g (2.12 oz).
- Frecuencia de trabajo: 50-333Hz.
- Torque: 21kg/cm ~ 5V.  
24kg/cm ~ 6V.  
24kg/cm ~ 6V.

- Velocidad: 0.15seg/60° ~ 5V.  
0.13seg/60° ~ 6.8V.



Figura 44 Servomotor digital DS3225 25 kg/cm [34] con cuerno de aluminio incorporado.

Este modelo, cuenta con un cuerno de aluminio como aditamento en el producto, tiene una medida entre centros de 2.4cm y orificios M3, muy semejante al utilizado para las simulaciones que era de 3cm. Dado que las variaciones no son demasiado significativas se conserva este complemento para la ejecución del movimiento del servo.

#### 11.4.4 Piernas y rotulas

Para el ensamblaje de la pierna se contempló utilizar rótulas esféricas y enlaces con rosca, a partir de esto y siguiendo la definición de medidas del *Capítulo II* en el que las piernas tienen que tener una medida de 15 cm aproximadamente y las rótulas mantengan un ángulo de trabajo de  $\pm 14^\circ$  aproximadamente. Para ello se canalizó un proveedor en la plataforma web Aliexpress denominado *Pit Pro Racing* el cual presenta un artículo para autos a escala RC que contiene dos juntas esféricas y una barra con los extremos roscados como los presentes en la figura.



Figura 45 - Eslabón para piernas con juntas esféricas en los extremos.

Algo importante a contemplar para el ensamblaje final es la modificación o aditamiento de algún coplee en uno de los extremos roscados de las juntas esféricas, esto debido a que la junta tiene una rosca de tipo M8 y los cuernos de los servomotores mantienen el modelo con orificio M3.

### 11.5 Definición de características

- Carga: 100gr – 30Kg.
- Dimensiones:
  - Altura: 15 cm.
  - Laterales: 30 – 35 cm.
- Ángulos:
  - Roll:  $\pm 21.25^\circ$ .
  - Pitch:  $\pm 22.87^\circ$ .
  - Yaw:  $\pm 10.2^\circ$ .
- Cartesiano:
  - X:  $\pm 4.8$ .
  - Y:  $\pm 3.04$ .
  - Z:  $\pm 3.36$ cm.
- Resolución:  $0.12^\circ$  –  $3^\circ$ .
- Tiempo de asentamiento: Siendo uno de los puntos más importantes a definir, entiéndase como *tiempo de asentamiento* al tiempo correspondiente que tarda el sistema en estabilizar un objeto deslizante sobre la superficie del panel resistivo. Para este punto al no tener mayor detalle del sistema de control no se puede establecer de manera definitiva, sin embargo, se puede hacer una estimación; esto tomando como un mínimo de tiempo de 0.00216seg correspondiente a la velocidad de acción de los actuadores seleccionados, ya que a pesar de tener un controlador óptimo que actúe más rápido de lo esperado, no podrá sobrepasar las capacidades del actuador.

Se observó el proceso realizado en [35] donde comparan distintos controladores aplicados a una plataforma *Stewart-Gough* entre los cuales utilizan un *control PID* con un tiempo de asentamiento (*ts*) de 0.82seg a 3seg a partir de lo cual, siendo un sistema semejante, se puede hacer mención del *tiempo de asentamiento* que puede lograr el sistema con el controlador PID aplicado, quedando como propuesta un rango de 0.8seg – 5 seg ligeramente mayor al rango planteado en [35] con la intención de dar margen a mejorar el controlador, a pesar de no ser la *optimización de controladores* uno de los objetivos del presente proyecto.

## 11.6 Diseño Electrónico

En esta sección se detalla una parte esencial del proyecto en desarrollo, las conexiones y el proceso diseño del microcontrolador para una configuración de sistema mínimo tomando en consideración las características de los componentes observados previamente en la *Selección de componentes*. Para ello se verán dos puntos esenciales; los periféricos empleados para cada sensor y actuador, y la fuente de alimentación de acuerdo con la corriente requerida para cada componente a partir de lo cual se determinarán las pistas de la placa PCB

El primer punto a definir es el diseño del circuito electrónico. Empezando de izquierda a derecha, en la *Figura 46* se puede observar los puertos en la parte superior correspondientes a la alimentación positiva y tierra para los seis actuadores. Esta fuente denominada *F\_Externa* se implementó independiente a la fuente principal debido a que estarán encargada de la etapa de potencia para los actuadores, esto con la intención de prevenir señales parasitas dentro de la señal del procesador, así como prevenir de posibles cortocircuitos para los cuales se introduce un fusible con un valor del 15% más de la corriente de operación con carga de los servomotores.

Los puertos debajo de esta alimentación se destinan a la señal de PWM de salida para los servomotores, y justo debajo están los puertos para el panel resistivo que un cuyo caso cuatro pines son digitales distribuidos en el puerto D y uno analógico conectado al ADC del puerto. Continuando con la parte superior se encuentran los pines correspondientes al programador que en este caso se contempla el uso del programador *NoMADA ISP+* [19]. Siguiendo hacia la derecha se encuentra la fuente del microcontrolador que al igual que con la fuente externa se contempla un fusible de protección.

En la parte inferior derecha se presenta el cristal externo para el microcontrolador, así como los elementos requeridos para la implementación del controlador en sistema mínimo. Justo debajo están los puertos destinados al giroscopio en los cuales dos irán directo al ADC en el puerto C y los otros dos son correspondientes a la fuente de alimentación. Finalmente, en el centro se encuentra el microcontrolador en su encapsulado THT.

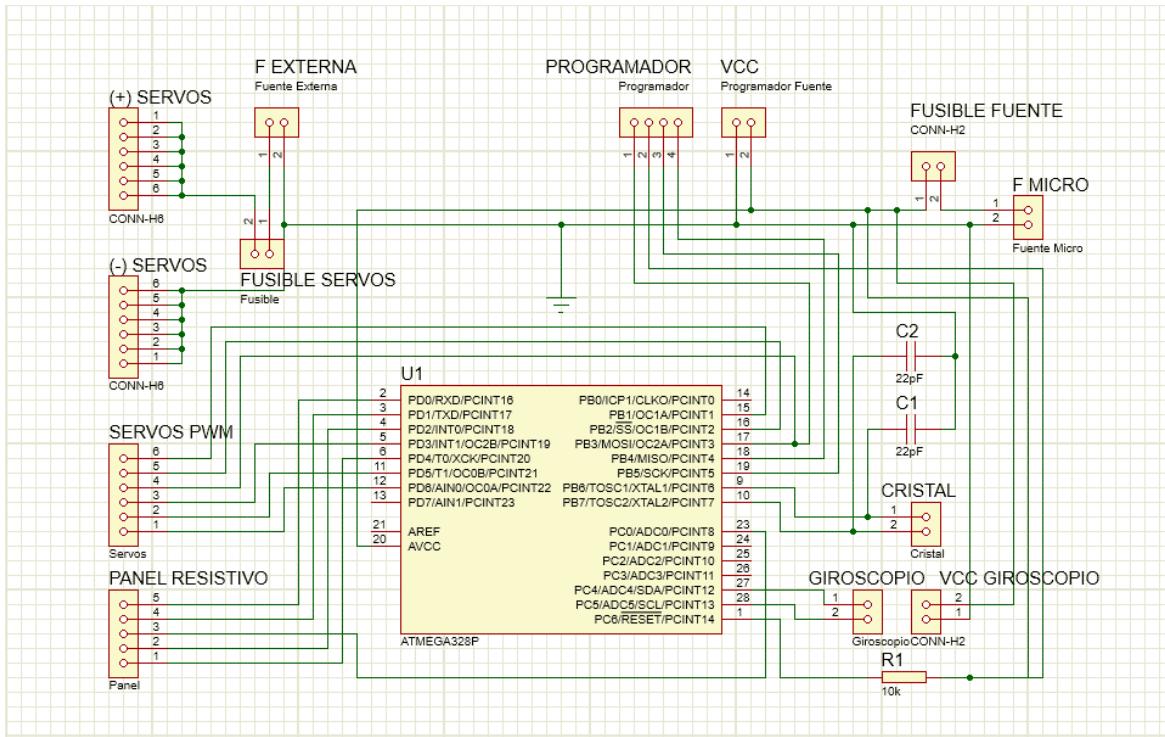


Figura 46 Diseño circuito electrónico.

Como se observa en el diagrama electrónico, de manera básica se implementan dos fuentes independientes entre sí, solo compartiendo tierra común. Al momento de generar el diseño de un *sistema de control* es una de las técnicas más comunes empleadas principalmente como prevención para la protección del microcontrolador y evitar ruidos en la señal de control. Estas se emplean de tal manera que una alimenta los periféricos de sensores y al propio microcontrolador, y otra a la etapa de potencia para la alimentación de los servomotores que en principio no es para este proyecto demasiado compleja ya que no requiere componentes de amplificación, esto debido a que no se requiere la amplificación de la señal de control y la corriente máxima que se requiere en el circuito no supera 10A.

Además de separar las fuentes de alimentación para el controlador en general, se planea incorporar un fusible de protección para ambas partes del circuito, es por eso por lo que se tiene que estimar la corriente total del controlador en cada parte del circuito. Para el cálculo de las corrientes se toma la división de las fuentes en las cuales se suma la corriente total de los componentes dependientes de cada una como se muestra en el anexo C1 16.3.1C [36].

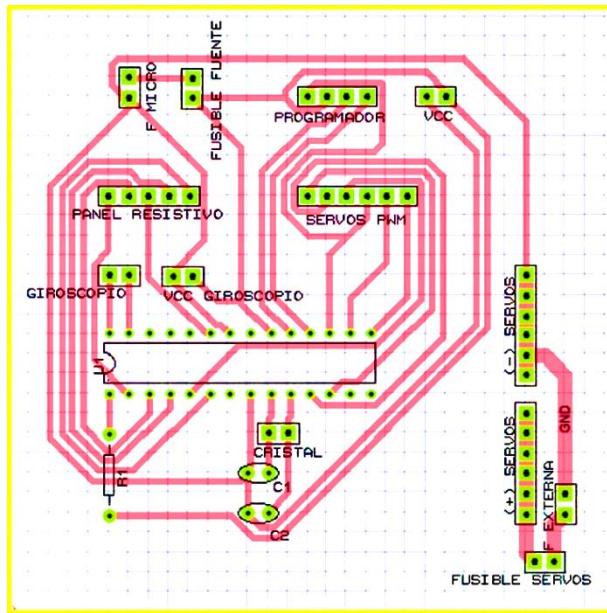


Figura 47 - Diseño PCB para Plataforma Stewart-Gough.

A partir del diseño de la PCB genera una vista 3D de cómo se vería el diseño preliminar de la PCB.

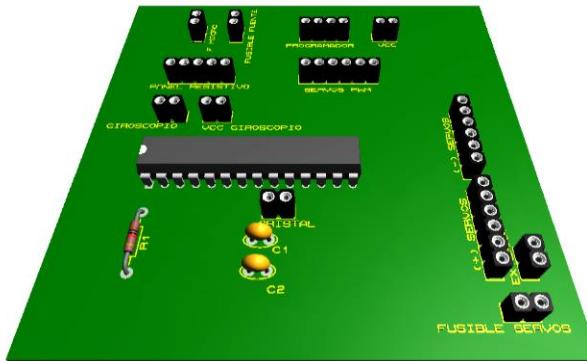


Figura 53 - Vista 3D de PCB del circuito electrónico.

## 11.7 Construcción del Prototipo

### 11.7.1 Plan de Manufactura

Respecto al plan de manufactura se presentan las piezas a maquinar en el anexo E 16.5.

Ya que el proyecto a desarrollar esta principalmente enfocado al diseño del controlador (siendo la construcción de la plataforma un medio para su implementación), se tienen contempladas pocas piezas para el proceso de manufactura siendo en su mayoría operaciones de ensamble. Es importante mencionar que actualmente se tiene como primera opción la implementación de corte láser para la obtención de dichas piezas, se contempla de esta manera debido a que el proceso de maquinado podría significar un aumento de costo y un retraso en el desarrollo del proyecto.

Ya que la plataforma no está diseñada para la manipulación de gran carga como lo es el caso de [13], sino que se implementará en la estabilización del objeto deslizante sobre el efecto final, se considera suficiente rigidez para la estructura el uso de material acrílico que como se observa en la sección de anexos se manejan en medidas estándar de 6 y 9 mm dependiendo de la pieza. Así mismo, se presenta el apartado de *Simulación de fuerzas externas* donde se aplicó ABS (material con propiedades semejantes) y presentó resultados favorables al tener un factor de seguridad de 3.4.

Es así como en los anexos E2 y E3 se presentan los planos correspondientes a la base inferior y superior respectivamente. En estas se colocarán los servos de manera vertical distribuidos alrededor de ambas plataformas, en el centro de la plataforma se colocará la PCB en la base inferior sobre el recuadro del centro. Los anexos E4 y E5 corresponden al diseño de la plataforma del efecto final y el soporte para sensores respectivamente, en donde se colocarán tanto el panel resistivo como el giroscopio referenciado en el mismo origen para evitar dificultades de compensación de coordenadas.

### 11.7.2 Manufactura de Piezas de Acrílico

Debido a las limitaciones de los proveedores locales se realizó una ligera modificación a las piezas de acrílico, ya que las piezas de 9 mm se reemplazaron por dos placas de 6 mm que posteriormente se unieron con pegamento para acrílico. A partir de ello se modificaron algunas de las piezas: las bases de los servos, para fijar estos mismos a la plataforma; el centro del efecto final, que sostiene y alinea el sensor MPU-6050, y el soporte del panel, que se ensambla y alinea al centro de la plataforma. Estas operaciones se realizaron con una perforación 2.5 mm seguido de un roscado M3. En la Figura 48 se puede observar la modificación y montura de la pieza central del efecto con el sensor MPU-6050

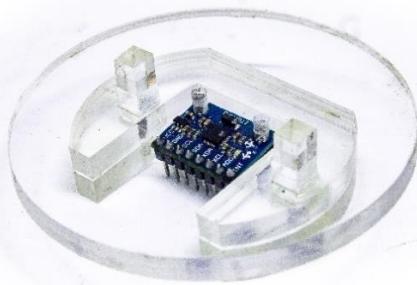


Figura 48 - Ensamble del centro de efecto final con giroscopio

Una última modificación a las piezas de ensamble se realizó sobre el efecto final, el cual, al tener juntas superiores paralelas entre sí, tendían a interferir al momento del ensamble. Esto se explica más a detalle en el anexo A5 (16.1.1).

### 11.7.3 Modificación de Juntas Esféricas

Debido a que el cuerno del servomotor tiene una entrada de tipo M3 es necesario realizar una modificación en la junta M8 (como se muestra en la Figura 49) de tal modo que quede concéntrica a la unión del cuerno. Esta operación requiere de precisión para evitar el mínimo de esfuerzos y desviaciones que este proceso pueda ocasionar al mecanismo final y con lo cual aumente el error general del sistema y se vuelva más difícil de controlar. Semejante al proceso en acrílico es necesario realizar una perforación al centro con broca M2.5 para después realizar el roscado a M3 como se describe en el anexo A1 (16.1.1).



Figura 49 - Ensamble de rotula inferior con cuerno del servomotor

### 11.7.4 Ensamble final

Una vez que se realizaron las respectivas modificaciones y uniones mecánicas se procede a realizar el ensamble final para detectar y corregir las posibles interferencias que se puedan producir debido a un error de diseño o mal proceso de manufactura. Así mismo se incluye en el ensamble las conexiones a la PCB diseñada en el anexo A3 (16.1.3) y se comprueba que éstas no estén limitadas o tensas ya que la plataforma tendrá movimientos de traslación y rotación que, aunque es en un espacio de trabajo muy limitado puede comprometer algunos componentes. El ensamble final se observa en la Figura 50.

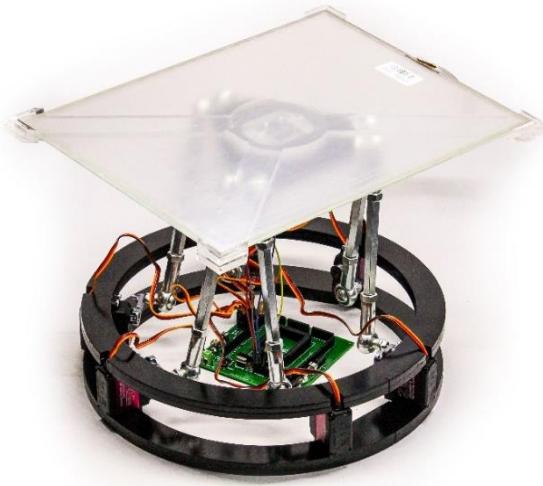


Figura 50 Ensamble final Plataforma Stewart-Gough.

## 11.8 Implementación de Sensores

### 11.8.1 Panel Resistivo

En la sección del Sensor Panel Resistivo se explica el funcionamiento básico de un panel resistivo de 5 hilos, a partir ello se programa mediante estados secuenciales como se muestra en el (anexo) utilizando como referencia los pines indicados en la figura donde se distinguen por colores los cuatro puntos de resistencia distribuidos en las esquinas y un pin central correspondiente al sensor. Estos se pueden configurar de cualquier manera considerando siempre que deben pasar por tres estados lógicos mejor descritos en el (anexo).

Por practicidad, se le denominará a cada esquina como SI (Superior Izquierda), SD (Superior Derecha), II (Inferior Izquierda) e ID (Inferior derecha) como se muestra en la Figura 52. Cada una de estas esquinas se conmutará en pareja para generar el gradiente en X y Y censando de manera secuencial. De esta forma al implementar el programa se obtuvieron los valores mínimos y máximos que para ambos ejes fueron 250 mínimo y 750 máximo.

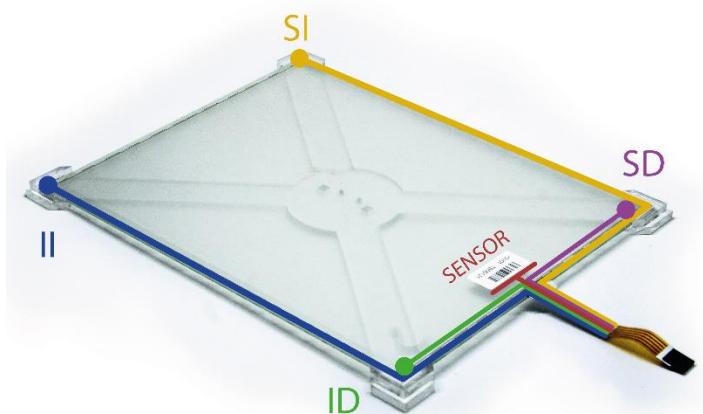


Figura 51 - Pines de conexión del Sensor Panel Resistivo

A partir de los datos obtenidos mediante esta implementación se realizó una interpolación lineal empleando la ecuación (46) para pasar de estos valores a las medidas reales del panel en mm, de tal modo que X adoptará valores de  $\pm 161$  correspondiente a los 322mm de longitud en X. Por su lado, Y tendrá valores de  $\pm 124$  que será equivalente a los 248mm de altura. De este modo se determina que la resolución del panel resistivo será de 1mm

$$Y = Y_0 + (Y_1 - Y_0) * (X - X_0) / (X_1 - X_0) \quad (46)$$

### 11.8.2 Giroscopio MPU-6050

Partiendo del principio para la obtención del ángulo de giro (posición angular) mediante las medidas obtenidas del MPU-6050 se programaron de tal modo que primero se manda a llamar una función de inicio donde se configuran todos los parámetros para la obtención correcta de los datos, esto va desde indicarle la velocidad de transferencia hasta configurar los grados/segundo con los que se desea trabajar. Realizando pruebas se observó que se obtenían mejores resultados empleando un rango máximo de 2000°/seg.

Además de lo visto en Sensor Giroscopio MPU-6050 se tenía que filtrar dicha señal debido a que cuando se usaba el método de obtención por el acelerómetro, este tendía a tener mucho error ante movimientos bruscos ya en cualquiera de los ejes ya que trataba de mandar la señal de aceleración para lo cual fue diseñado. Así mismo, cuando se utilizaba el método por giroscopio se puede observar un error muy común en los sensores tipo giroscopio: el valor inicial no es el mismo luego de realizar una serie de movimientos. Al volver a la posición de partida se observará que existe una diferencia de varios grados y se acumula conforme se mueve el giroscopio, a este error se le denomina *drift*.

Existen varios métodos para solucionar esto, pueden ser filtros promediadores, de media, pasa-bajas, pasa-altas, etc. El método seleccionado por su practicidad y poco sacrificio de procesamiento es el del *filtro complementario*, este filtro básicamente se denomina así porque implementa una parte de ambos filtros logrando obtener de manera experimental las características deseadas para el sistema, ya que; el acelerómetro, proporciona rapidez de lectura y reinicia automáticamente al alinear los ejes X y Y perpendiculares a Z (ya que es dependiente de la gravedad), y por otro lado el giroscopio, proporciona precisión y elimina el error generado por el acelerómetro.

En la ecuación se observa el funcionamiento básico del filtro complementario, donde se tiene como parámetros el valor anterior del giro, la lectura actual y su respectiva conversión a grados, y la lectura del giro mediante el acelerómetro. Antes de este filtro iría la ecuación que relaciona la aceleración el giro, y después del filtro se tiene que indicar que el ángulo anterior es igual al ángulo actual, así cada vez que pase se estará actualizando el valor. El programa de este sensor se creó como librería en el anexo H1 (17.1.1.7)

$$\text{Ángulo } X = 0.8 * (\text{Ángulo } X_{\text{previo}} + (\text{Lectura } X_{\text{giro}} / 131)) + 0.2 * (\text{Lectura } X_{\text{acel}}) \quad (47)$$

## 11.9 Diseño conceptual del Control PID

### 11.9.1 Diseño Conceptual

En este apartado se explica el diagrama general de bloques del controlador cinemático, mostrado en la *Figura 52* correspondiente al *controlador de alto nivel*:

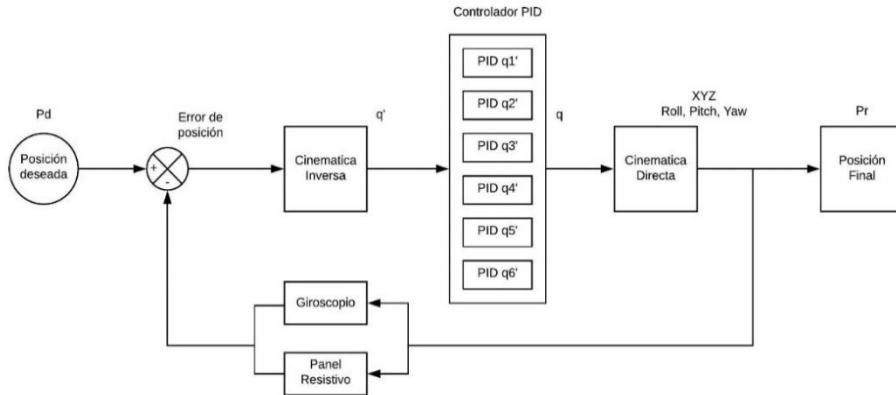


Figura 52 - Control cinemático para la plataforma Stewart-Gough.

Se establece la posición deseada como entrada del diagrama, el siguiente bloque es un sumador en el cual, dado que es el primer ciclo y no existe referencia de error, procede a la entrada del bloque de cinemática inversa. Aquí en el nodo de entrada se encontrarían las variables determinadas por X, Y, Z y roll, pitch y yaw en base a los cuales se le determina un valor de salida  $q'$  para cada actuador.

El siguiente bloque es uno de los presentes más importantes, esto debido a que corresponde al bloque del controlador PID. Este tiene la función de tomar el valor de la variable  $q'$  y aplicar las ganancias correspondientes para observar el valor final deseado con base en la cinemática inversa. En caso de no existir error, este controlador no realizaría mayor trabajo más que un ligero acondicionamiento y a partir de esto se obtiene la salida de  $q$  correspondiente a los ángulos reales definidos para la posición del efecto final y que en cuyo caso, procede a la entrada del bloque de cinemática directa que, de forma contraria a la cinemática inversa, ésta generará como salida los valores correspondientes a X, Y, Z y alabeo, cabeceo y guiñada.

Para el segundo ciclo, se tiene el valor de la salida como referencia, por lo tanto siguiendo el nodo de retroalimentación se llega a los bloques correspondientes a los sensores tanto del giroscopio como el panel resistivo colocados en paralelo, se realiza esta configuración con la intención de sensar los 6 grados de libertad de la Plataforma Stewart-Gough de tal modo que la pantalla resistiva tendrá la función de observar movimientos de objetos sobre la superficie de ésta y que al estar sobre una superficie fija horizontal el giroscopio tendría

un valor aproximado a 0 sin contemplar señales de ruido. Para sensar los valores de *roll*, *pitch* y *yaw* las perturbaciones se aplicarían a la plataforma mediante movimientos de todo el mecanismo sobre el plano inclinado, en cuyo caso, al no tener objetos sobre el panel resistivo, éste estaría mostrando un valor aproximado a 0.

Si la plataforma detecta alguna variación en sus parámetros de entrada con valores distintos a la señal de salida, los sensores estarán encargados de reconocer dicha variación y volviendo al bloque de *suma/resta* correspondiente al *error de posición* justo después de la entrada existirá una diferencia que entrarán nuevamente al bloque de cinemática inversa y posteriormente al controlador PID que se aplicará de manera individual a cada actuador con la intención de reducir el orden del sistema.

Incursionando más a detalle en el controlador PID se tiene que en la entrada estaría el valor  $q'$  obtenido por la cinemática inversa y a partir de ello se realizaría el procesamiento como se puede observar en la Figura 53 comenzando de izquierda a derecha donde enseguida de  $q'$  se tiene un bloque de suma para observar el valor del error de posición en el actuador. El siguiente bloque corresponde como tal a la acción de compensación de las ganancias *PID* (*Proporcional, Integral y Derivativo*) para después tener un bloque de convertidor analógico que tendrá la función de interpretar el valor de posición para generar a partir de ello la señal de control para el actuador, no sin antes pasar por un bloque de saturación que permita proteger al servo de señales parasitas o picos que puedan quemarlo. Así se llega finalmente al bloque del actuador para generar la acción del mecanismo y obtener la posición deseada.

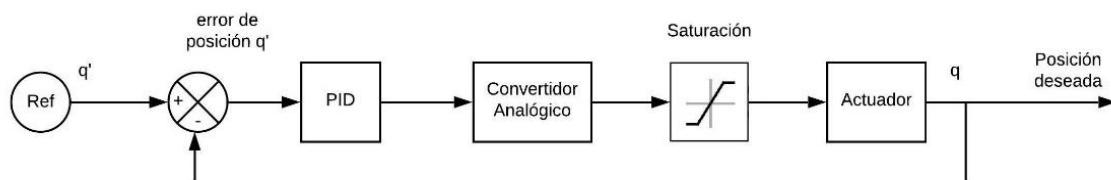


Figura 53 - Diagrama del bloque del controlador PID

Incorporando el diagrama de control PID a cada una de las salidas de la cinemática inversa se obtendría el diagrama del controlador completo.

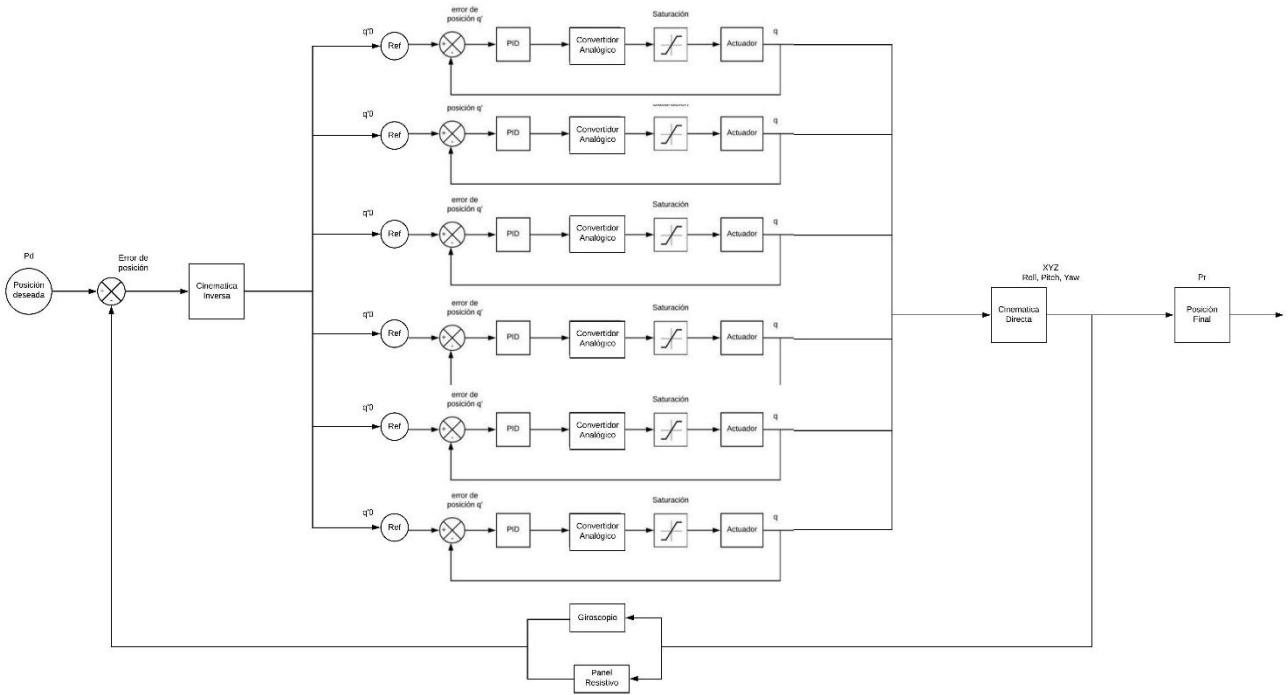


Figura 54 - Diagrama completo del controlador diseñado para la Plataforma Stewart-Gough

### 11.9.2 Aproximación de la Función de Transferencia

#### 11.9.2.1 Función de transferencia de servomotores

La función de transferencia se modeló respecto al servomotor 1 para la deducción de tres funciones de transferencia (anexo C2 16.3.2)

$$S_{1DS}(s) = \frac{179.3078716}{0.39s + 1} \quad S_{1O}(s) = \frac{163.8278295}{0.287021s + 1} \quad S_{1SK}(s) = \frac{163.8278295}{0.3147s + 1}$$

Nótese que  $S_{1SK}(s)$  es más preciso respecto a los valores de  $S_{1DS}(s)$ . Por esta razón, se selecciona la aproximación SK para los demás servomotores.

$$\begin{aligned} S_1(s) &= \frac{163.8278295}{0.3147s + 1} & S_2(s) &= \frac{163.8278295}{0.3138s + 1} & S_3(s) &= \frac{163.8278295}{0.3264s + 1} \\ S_4(s) &= \frac{163.8278295}{0.3485s + 1} & S_5(s) &= \frac{163.8278295}{0.3154s + 1} & S_6(s) &= \frac{163.8278295}{0.3176s + 1} \end{aligned}$$

#### 11.9.2.2 Función de transferencia giroscopio

Uno de los objetivos específicos consiste en realizar movimientos de la plataforma sobre el plano inclinado a partir de lo cual el efecto final deberá mantener su posición sobre el plano horizontal, siendo la base fija quien tenga movimiento respecto a este. Para obtener el diseño de la planta se parte del funcionamiento básico del acelerómetro como herramienta para medir el giro.

Se propone una masa sin deslizamiento sobre el efecto final (alineada al eje de coordenadas en XY a partir de ello se plantea el diseño de la planta que será semejante al diseño de un sistema de biga-bola omitiendo el objeto deslizante y considerando solamente los efectos de la gravedad sobre el sensor giroscopio. Algo importante de resaltar de este tipo de sistema es que el controlador se puede trabajar solo para uno de los ejes y después replicar este sistema en el eje ortogonal horizontal ya que son linealmente semejantes ya que actúan las mismas fuerzas y en la misma proporción sobre estos.

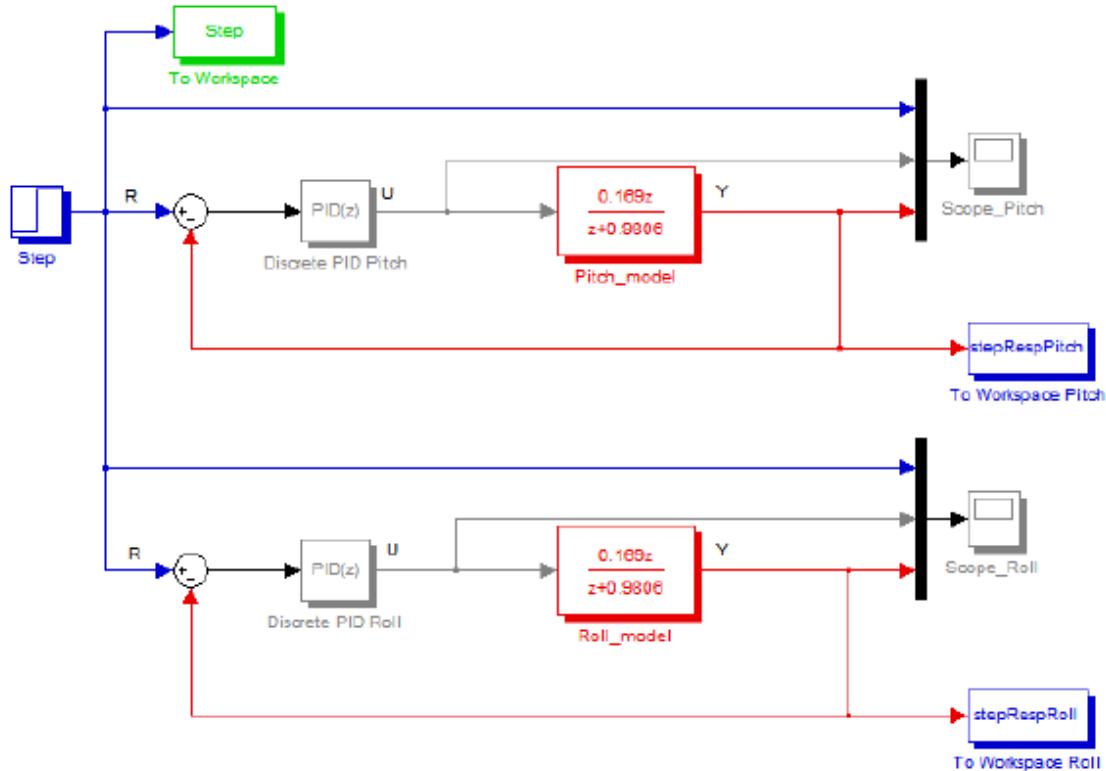


Figura 55 - Controlador PID para plataforma autoestabilizada [37]

Como se observa en la Figura 55 se propone un controlador semejante para cada uno de los ejes de rotación (roll y pitch) que presentan en [37] a partir de lo cual se propone la planta del sistema como:

$$\frac{0.169z}{z + 0.9806} \quad (48)$$

#### 11.9.2.3 Función de transferencia panel resistivo

El modelado del panel resistivo se puede observar de manera semejante a un sistema de biga-bola como dos sistemas de control, uno para el eje X y otro para el eje Y. Por lo tanto, se debe aplicar un controlador PID de posición en X, y otro en Y y así controlar la posición del objeto deslizante sobre el plano XY del panel resistivo [38].

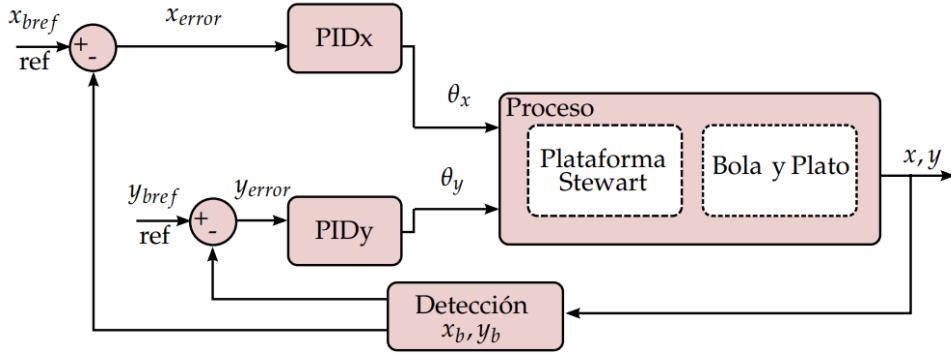


Figura 56 Diagrama de bloques PID Bola y Plato [38].

El modelado obtenido fue hecho por el método de Euler-Lagrange considerando que la esfera es un objeto homogéneo, no desliza, no se despega del plato (panel), no hay fricción y tampoco existe deslizamiento. El Lagrangiano queda en función de las energías de rotación y traslación de la esfera, así como las energías de rotación y traslación del plato (ecuación (50)), su expresión final como lo muestra la ecuación (49) [39]:

$$L(t) = E_c(t) - E_p(t) \quad (49)$$

$$L(t) = E_{rot\_bola}(t) + E_{tras\_bola}(t) + E_{rot\_plato}(t) + E_{tras\_plato}(t) \quad (50)$$

Se destaca que, puesto que son dos variables, el resultado de aplicar la ecuación de Euler-Lagrange serán dos ecuaciones que describen la aceleración de la esfera en cada uno de los ejes (X y Y):

$$\frac{d^2x(t)}{dt^2} = \frac{5}{7}g\theta_x \quad (51)$$

$$\frac{d^2y(t)}{dt^2} = \frac{5}{7}g\theta_y \quad (52)$$

Pasando al dominio de Laplace:

$$G_x(s) = \frac{\theta_x(s)}{X(s)} = \frac{7}{s^2} \quad (53)$$

$$G_y(s) = \frac{\theta_y(s)}{Y(s)} = \frac{7}{s^2} \quad (54)$$

El siguiente paso es aplicar transformada Z utilizando retenedor de orden cero (ZOH) y como consecuencia, se obtiene la ecuación (55) (aplicable para (53) y (54)):

$$G(z) = 3.5 \frac{T^2(z+1)}{z^2 - 2z + 1} \quad (55)$$

Donde  $T$  es el tiempo de muestreo.

Considérese la relación de Tustin como se muestra en la ecuación (56):

$$z = \left(\frac{2}{T}\right) \frac{(z - 1)}{(z + 1)} \quad (56)$$

Se aplica la transformada Z a la ecuación (12), correspondiente al PID en dominio de Laplace, y se sustituye (56) para obtener el controlador PID discreto  $C(z)$  (ecuación (57)):

$$C(z) = \frac{(k_p + k_I + k_D) - (k_p + 2k_D)z^{-1} + (k_D)z^{-2}}{1 - z^{-1}} \quad (57)$$

$$C(z) = \frac{Y(z)}{U(z)} = \frac{q_2 z^{-2} - q_1 z^{-1} + q_0}{1 - z^{-1}} \quad (58)$$

Donde  $Y(z)$  es la señal de control y  $U(z)$  el error producido por la posición deseada y la señal obtenida del panel resistivo. Aplicando transformada Z inversa se obtiene la ecuación de diferencias que se implementará en el microcontrolador:

$$y[n] = y[n - 1] + q_2 u[n - 2] + q_1 u[n - 1] + q_0 u[n] \quad (59)$$

### 11.9.3 Simulación del Controlador PID

#### 11.9.3.1 Controlador PID digital servomotores

El **scope** de la simulación (Anexo 16.7.4) muestra tres gráficas: la respuesta al escalón unitario (amarilla), la respuesta del controlador PID digital (azul) y la respuesta de la función de transferencia sin controlador (roja), donde el tiempo de respuesta es menor si se aplica el controlador. Ganancias  $K_p=1$ ,  $K_i=0.1$  y  $K_d=1$ .

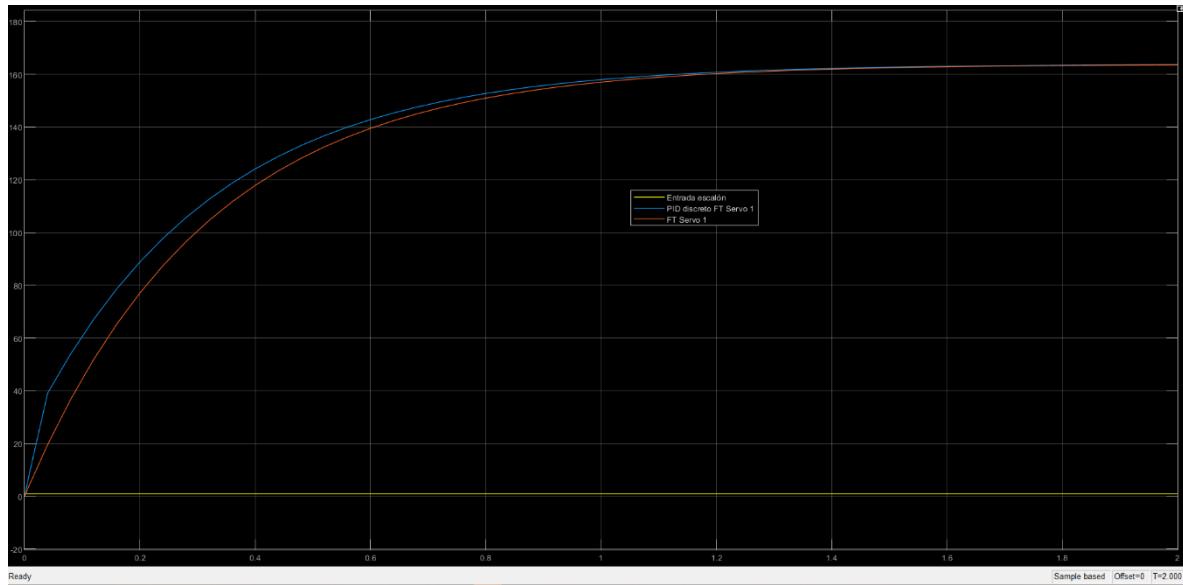


Figura 57 Respuesta al escalón unitario.

### 11.9.3.1 Controlador PID digital plano inclinado

Se puede observar que la planta alcanza el estado estable en aproximadamente 8 seg, sin embargo, mantiene un error marginal no deseado. Ganancias:  $K_p=0.0586$ ,  $K_i=11.719$  y  $K_d=0$ .

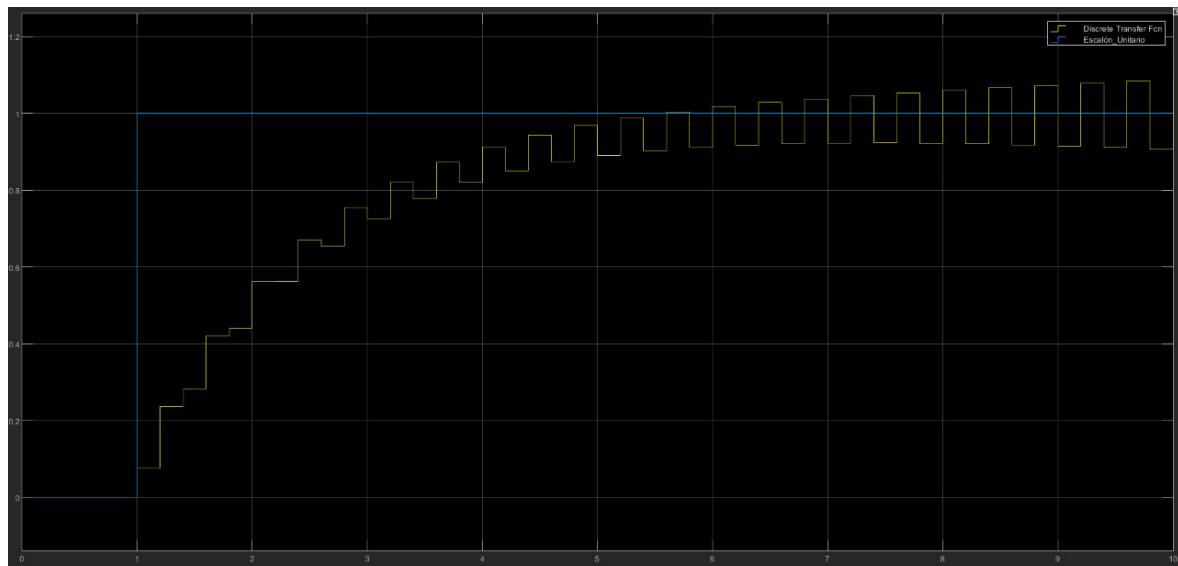


Figura 58 - Figura 57 Simulación controlador PID digital con bloque PID de Simulink.

### 11.9.3.2 Controlador PID digital panel resistivo

La simulación para el controlador del panel resistivo indica un tiempo de establecimiento aproximadamente de 8 segundos (*G5: PID Panel Resistivo*). Ganancias  $K_p=0.065$ ,  $K_i=0.001$  y  $K_d=0.85$ .

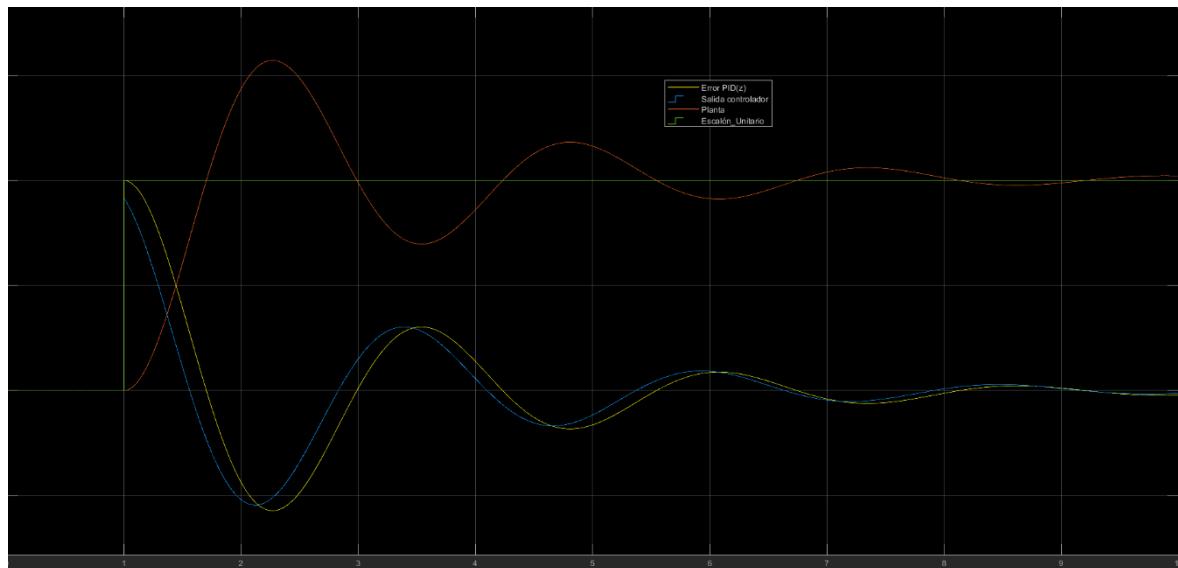


Figura 59 Simulación controlador PID digital con bloque PID de Simulink.

## 12 IMPLEMENTACION, PRUEBAS Y RESULTADOS

### 12.1 Control de posición en lazo abierto y trayectorias programadas

Este tipo de control permite desplazar y orientar el efecto final. Se propuso realizar este control en lazo abierto para realizar trayectorias previamente programadas por el usuario y que la plataforma realice estos movimientos cílicos. Esto comúnmente se emplea como material de laboratorio para hacer reaccionar disoluciones de sustancias. Las características finales obtenidas experimentalmente fueron las siguientes:

Característica	Propuesta	Resultado
Carga	100gr – 30kg	0gr – 30kg
Dimensiones	Altura: 15cm Laterales: 30-35 cm	Altura: 21 cm Laterales: 30 cm
Ángulos	Roll: $\pm 21.25^\circ$ Pitch: $\pm 21.25^\circ$ Yaw: $\pm 21.25^\circ$	Roll: -14° a 14.7° Pitch: -16° a 13.75° Yaw: -32° a 31.66°
Cartesiano	X: $\pm 4.8$ . Y: $\pm 3.04$ . Z: $\pm 3.36$ cm.	X: -4.3 a 4.3 cm Y: -4.2 a 4.4 cm Z: -3.1 a 1.87 cm

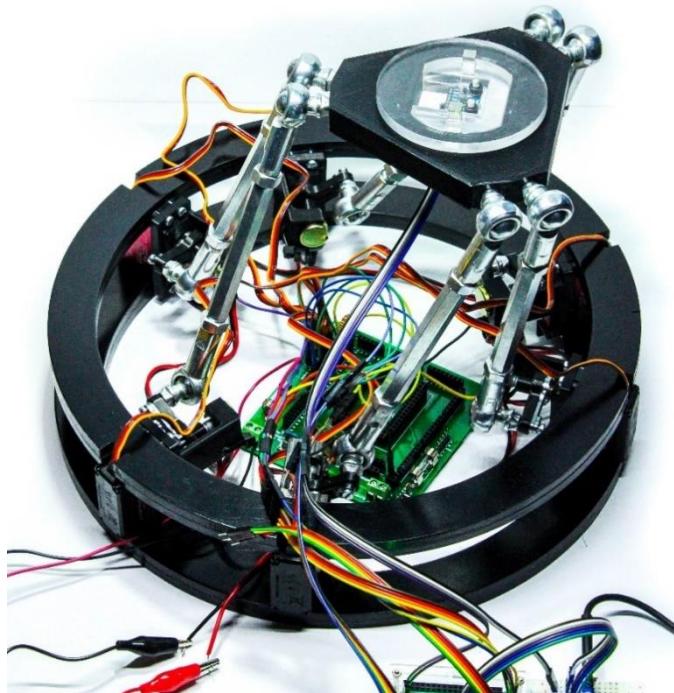


Figura 60 Plataforma Stewart desplazada 4 cm en eje X.

## 12.2 Control PID digital a servomotores

El PID de los servomotores se implementó con el código del anexo H2: PID servomotores en el cual se utilizan las librerías creadas y descritas en secciones anteriores, tales como el control de los servomotores, el ADC, UART, cinemática inversa y millis. Se empleó el Timer 1 para realizar el tiempo de muestreo del controlador PID que se seleccionó de 10 ms. A partir de ello se implementó la cinemática inversa como entrada del PID y la salida de éste aplicada a cada servomotor. Para la retroalimentación de los servomotores se empleó un mecanismo de sujeción que posiciona un potenciómetro concéntrico al servomotor como se muestra en la Figura 61.

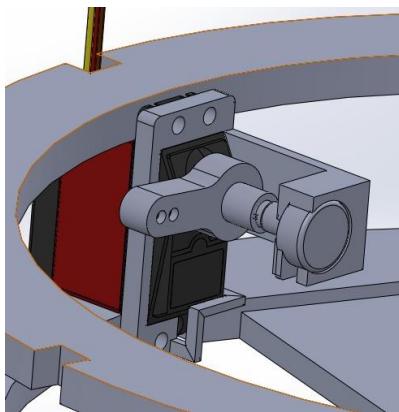


Figura 61 - Mecanismo de sujeción de potenciómetro para medir posición del servomotor.

El controlador tiene una velocidad de reacción de apenas unos milisegundos, y el error marginal en estado estable oscila entre  $\pm 1^\circ$ . Al realizar trayectorias programadas el efecto final las describe tal cual lo haría sin control, sin embargo, se observa que el controlador puede limitar los movimientos de los servomotores según el antiwindUP que se implemente, esto se hace principalmente para proteger al propio mecanismo y evitar que los servos se disloquen o genere alguna fractura en las bases de los potenciómetros.

Una de las desventajas que se hace presente es el uso de potenciómetros convencionales como fuente de retroalimentación. A pesar de que se calibraron cada uno de los potenciómetros tanto individual como grupalmente se presentaba error en las lecturas debido a que estos no suelen ser lineales. Esto hizo evidente la acción del controlador sobre el sistema, puesto que al realizar las trayectorias programadas con el controlador se observó que se tendían a deformar un poco, y/o no alcanzaban el valor de referencia que generaba la cinemática inversa. Al analizar este error, se observó que el controlador realmente estaba manteniendo ese error marginal a la respuesta, sin embargo, los potenciómetros mandaban lecturas erróneas de posición.

$$K_p = 1 ; \quad K_i = 0.1 ; \quad K_d = 1$$

Para corroborar este último punto se puede observar en la Figura 62 que estudia el servomotor 1 (Los demás servomotores se muestran en el anexo H4 17.1.4) donde; CI\_1, corresponde al valor de *setpoint* indicado por la cinemática inversa; S1, es el valor de posición del servomotor mediante el potenciómetro, y PID corresponde al valor de acción del controlador impuesta sobre el servomotor

CI_1: -0.37	S1: -0.34	PID: 1.47
CI_1: -0.37	S1: -0.34	PID: 1.47
CI_1: -0.37	S1: -0.34	PID: 1.47

Figura 62 - Error marginal del controlador PID de los servomotores

### 12.3 Movimientos efectuados sobre el plano inclinado

Para este objetivo se empleó el controlador propuesto por [37] planteado en la sección 11.9.2.2 donde se proponen las siguientes ganancias:

$$K_p = 0.0586 ; \quad K_i = 11.719 ; \quad K_d = 0$$

Al emplearlo directamente sobre la plataforma, se observó que esta tiende a oscilar con estas ganancias y subsecuente a divergir. Por lo tanto se hicieron diversas pruebas a partir de las cuales se obtuvieron las siguientes ganancias con las cuales se logró un error marginal en estado estable de  $\pm 4$  aproximadamente.

$$K_p = 0.5 ; \quad K_i = 8 ; \quad K_d = 0.001$$



Figura 63 Giro de plataforma en roll de 15°.

Algo importante de rescatar de este controlador PID es que para lograr estabilizar el efecto final se tiene que ser cuidadoso al modificar las ganancias, ya que si se aplica demasiada ganancia *diferencial* el sistema tiende a oscilar debido a que cualquier grado de inclinación por muy mínimo que sea tratará de compensarlo de manera brusca de tal modo que se tendrá un sobreimpulso que cause un efecto en cadena. Es por ello por lo que para aplicar un buen control en este caso se debe priorizar las ganancias *proporcional e integral*.

## 12.4 Objetos deslizantes sobre el panel resistivo

Al momento de implementar el control PID al sistema *bola y plato* (o de *objetos deslizantes sobre el panel resistivo*) se observó que el sistema divergía respecto a la posición deseada con los valores de ganancias obtenidas mediante cálculos (adjunto a la simulación) y por lo tanto no era capaz de alcanzar el *setpoint* y tendía a expulsar el objeto deslizante.

Realizando diversas pruebas con ajustes manuales de las ganancias, se observó algo semejante al fenómeno del PID sobre el plano inclinado ya que como el objeto deslizante tiende a acelerarse dependiendo de la inclinación del panel, el controla deberá realizar una acción de compensación lo suficientemente rápida para evitar que el sistema expulse al objeto, por lo tanto, se observa que para este sistema son más importantes las ganancias K<sub>p</sub> y K<sub>d</sub> y el controlador tiende a ser más del tipo PD.

Finalmente, mediante pruebas se encontró un error marginal en estado estable de aproximadamente ±3cm respecto al *setpoint*. Cabe mencionar que el panel resistivo está configurado para medir con una resolución de 1mm por lo tanto se debe tener cuidado al modificar las ganancias ya que al ser un sistema real hay diversos factores que influyen sobre este y cualquiera de ellos lo hace divergir. Tal es el caso de las rotulas, las cuales se observó que generaban esfuerzos no deseados por la fricción excesiva que no se logró erradicar completamente y esto generaba cierta deformación en la trayectoria real del enlace entre el servomotor y el efecto final.

$$K_p = 8.6 \quad ; \quad K_i = 0.02 \quad ; \quad K_d = 310$$

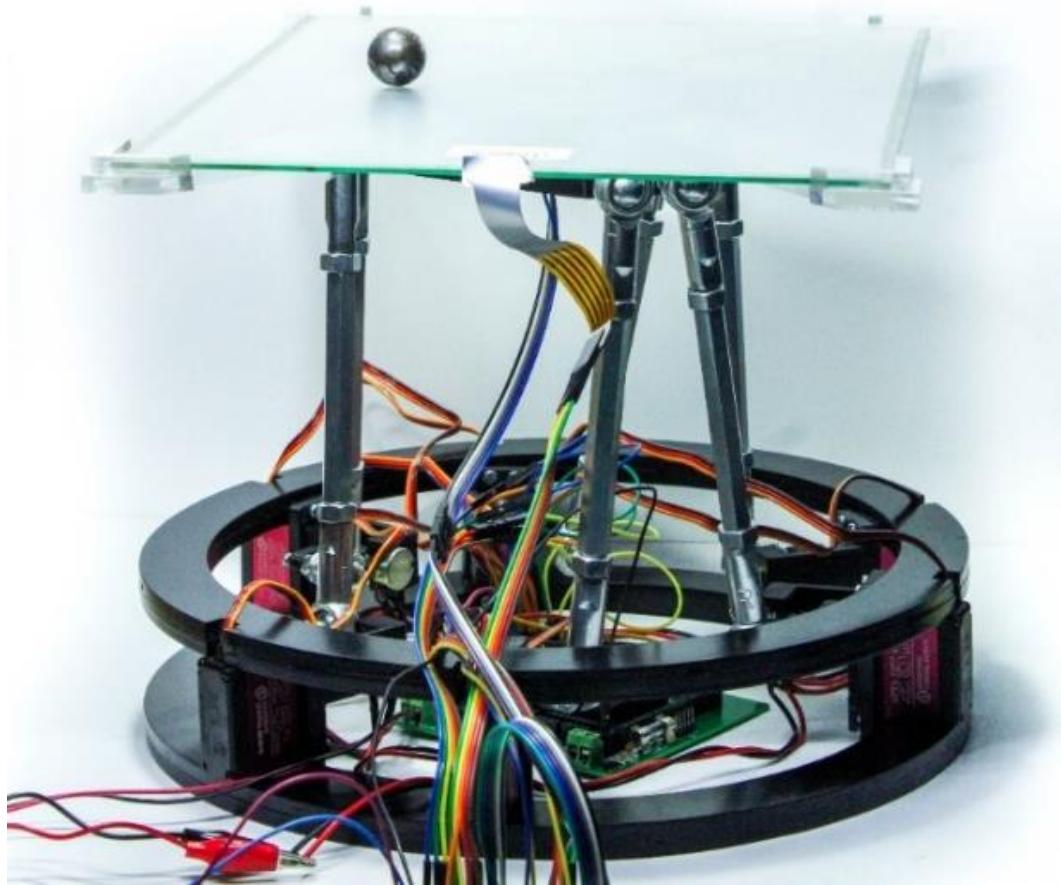


Figura 64 Posición de objeto esférico deslizante sobre panel resistivo

## 13 CONCLUSIONES

- El modelo matemático de la cinemática inversa de un robot siempre va a tener un error entre el prototipo calculado/simulado y el prototipo físico. Cabe destacar, que el cálculo de ésta lo realiza el microcontrolador en menos de 1 ms por lo tanto no produce ningún retardo significativo para el proceso del sistema.
- En la cinemática directa, se determinó que el cálculo por el método de Denavit-Hartenberg no era el adecuado, pues las simulaciones calculaban resultados con mucho error respecto a algún valor esperado.
- Al construir un robot paralelo se debe considerar el espacio que va a necesitar cada componente durante y después de su implementación debido a que generalmente los componentes se agrupan dentro del cuerpo del mismo robot. Al realizar un mayor número de consideraciones amplia la posibilidad de agregar trabajos futuros al prototipo.

- La sintonización de un controlador PID por método analítico suele ser muy distinto a la sintonización manual, debido a factores que no son medibles. Es decir, el modelo matemático suele ser muy ideal.
- Como se menciona en la implementación de los controladores PID, uno de los factores que afectó directamente al control de la *Plataforma Stewart-Gough* fue la selección de las rotulas, ya que al tener demasiada fricción que no se logró erradicar afectaba al movimiento del enlace entre el servomotor y el efecto final.
- Actuadores como los servomotores seleccionados para este proyecto presentan un control digital muy preciso, debido a esto las ganancias de control tienden a ser ganancias unitarias.

## 14 CRONOGRAMA TT2

Actividad	Inicio	Fin	Octubre				Noviembre				Diciembre				Enero			
			1 05 - 11	2 12 - 18	3 19 - 25	4 26 - 01	5 02 - 08	6 09 - 15	7 16 - 22	8 23 - 29	9 30 - 06	10 07 - 13	11 14 - 20	12 21 - 27	13 28 - 03	14 04 - 10	15 11 - 17	16 18 - 24
Construcción y modificación de submecanismos	05/10/2020	15/10/2020																
Pruebas de movimiento y límites de submecanismos	12/10/2020	22/10/2020																
Corrección de interferencias del mecanismo	19/10/2020	25/10/2020																
Programación y pruebas del sensor panel resistivo	19/10/2020	28/10/2020																
Programación y pruebas del sensor giroscopio MPU-6050	12/10/2020	28/10/2020																
Implementación de sensores para la recepción de datos	26/10/2020	05/11/2020																
Pruebas y modelado de actuadores	02/11/2020	10/11/2020																
Obtención de la FT en base al modelado de actuadores	09/11/2020	20/11/2020																
Ensamble del sistema, pruebas e inspección de interferencias	16/11/2020	22/11/2020																
Diseño y simulación del control PID	16/11/2020	25/11/2020																
Implementación del control PID	20/11/2020	05/12/2020																
Pruebas, reajustes y sintonización del control PID del sistema	01/12/2020	10/12/2020																
Pruebas finales y detallado	10/12/2020	12/12/2020																
Documentación	05/10/2020	12/12/2020																
Entrega del documento final para revisión.	14/12/2020	14/12/2020																
Cambios y correcciones imprevistas																		
Presentación final	11/12/2020	11/12/2020																

## 15 BIBLIOGRAFÍA

- [1] D. A. R. Rodríguez, Diseño de una plataforma robótica paralela de 6 DOF para asistente quirúrgico en cirugías de reconstrucción craneo-facial, 2010.
- [2] J. E. Gwinnett, Amusement device, Estados Unidos de America: Patentes, 1931.
- [3] W. Pollard, Spray painting machine, Estados Unidos de America: Patentes, 1940.
- [4] Redifon, «Another Redifon flight simulator sold,» Flight International, Crawley Sussex England, 1962.
- [5] A. Barrientos, Fundamentos de Robótica, McGraw-Hill, 2007.
- [6] K. Ogata, Ingeniería de control moderna, Madrid: Pearson, 2010.
- [7] K. &. Krishnaswamy, Estimation of Time Delay Constant Parameters in Time, Frequency, and Laplace Domains, The Canadian Journal of Chemical Engineering, pp. p. 257-262.
- [8] K. J. Aström, «PID Control,» de *Control System Design*, 2002, pp. 216-251.
- [9] V. Mazzone, «Controladores PID,» de *Automatización y Control Industrial*, Quilmes, Argentina, Universidad Nacional de Quilmes, 2002, pp. 1-11.
- [10] I. A. P. Hidalgo, «Resumen de las principales características de controladores PID,» Universidad Nacional de San Juan, San Juan, Argentina.
- [11] K. Ogata, Sistemas de control en tiempo discreto, Minnesota: Prentice Hall, 1996.
- [12] C.-T. Chen, Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods, USA: Oxford University Press, 2006.
- [13] S. R. D. E. H. d. H. Abel Villanueva Esquivel, Análisis, diseño y construcción de un robot paralelo en configuración Stewart-Gough, Zacatecas, 2017.
- [14] Redox, «eMotion-6000». E.E.U.U. 2017.
- [15] Lapik, «Maquina de coordenadas KIM-750». E.E.U.U 2017.
- [16] A. Division, «Simulador de ACFS». E.E.U.U 2017.
- [17] J. (. Co., «Alibaba,» [En línea]. Available: <https://spanish.alibaba.com/product-detail/hot-sales-6dof-stewart-platform-for-pid-controlled-ball-which-is-small-size-62265590530.html?spm=a2700.8699010.normalList.16.593a182f7T2xtD>. [Último acceso: 07 2019].

- [18] L. E. G. Jaimes, Control Digital Teoría y Práctica, Medellín, Colombia: Politécnico Colombiano Jaime Isaza Dadavid, 2009.
- [19] NoMADA, «NoMADA ISP+». México 2018.
- [20] L. Á. Silva, Control Visual de Robots Pralelos. Análisis, desarrollo y aplicación a la plataforma robotenis, Madrid: Escuela Técnica Superior de Ingenieros Industriales, 2005.
- [21] F. Szufnarowski, Stewart platform with fixed rotary actuators: a low cost design study, E.E.U.U., 2018.
- [22] J. M. R. Martínez, Las diferentes formas del criterio de Kutzbach-Grübler, Salamanca, Gto: Facultad de Ingeniería Mecánica, Eléctrica y Electrónica, 2015.
- [23] E. P. R. M. Natalia Ximena González Alvarez, Estudio, diseño y construcción de una plataforma robótica didáctica tipo stewart aplicada al estudio de controladores difusos, Cuenca, Ecuador: Universidad Politécnica Salesiana, 2011.
- [24] M. S. D. Hernan González Acuña, Cinematica inversa y análisis del espacio de trabajo de una plataforma Stewart, Bucaramanga, Colombia: Universidad Autónoma de Bucaramanga, 2009.
- [25] Z. Lazatevic, Feasibility of a Stewart Platform with Fixed Actuators as a Platform for CABG Surgery Device, E.E.U.U: Coumbia University Department of Bioengineering, 2003.
- [26] B. Chatterjee, Design of a semi-regular Stewart Platform manipulator for a desired workspace, Bangalore, India: Mechanical Engineering Department, Indian Institute of Science, 2009.
- [27] D. P. 1. P.M.G. 2, Parallel Manipulators Applications-A Survey, Modern Mechanical Engineering, 2012.
- [28] E. F. Fichter, "A Stewart Platform-Based Manipulator: General Theory and Practical Construction", Oregon State University.
- [29] A. R. Gómez, Inertial Stablization System Based on a Gough-Stewart Parallel Platform, Barcelona, España, 2017.
- [30] J. P. M. R. A. Esquivel, Conversión A/D, Bogotá, Colombia, 2019.
- [31] A. Corporation, «ATMega328P». E.E.U.U 02 2009.

- [32] H. Bartle, «MathWorks,» 2016. [En línea]. Available: [https://la.mathworks.com/matlabcentral/fileexchange/59781-hbartle-stewart\\_platform](https://la.mathworks.com/matlabcentral/fileexchange/59781-hbartle-stewart_platform). [Último acceso: 26 Septiembre 2019].
- [33] InvenSense, «MPU6050-Giroscope/Acelerometer». E.E.U.U 2011.
- [34] ShenZhen, «AliExpress,» [En línea]. Available: <https://es.aliexpress.com/item/32897006346.html?spm=a2g0s.8937460.0.0.43572e0e2E0Gem>.
- [35] A. Kassem, Comparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform, Damascus, Syria: Higher Institute of Applied Science and Technology, 2015.
- [36] J. A. Pérez, Estimador de torque electrónico para motores de inducción jaula de ardilla, Bogotá D.C.: Universidad Distrital Francisco José de Caldas, 2017.
- [37] P. T. Blanco, Diseño de controlador PID para plataforma giroestabilizada, Alcalá: Escuela Politécnica Superior, 2015.
- [38] T. P. & J. Martínez, Diseño e implementación de un Control Óptimo LQR con la tarjeta Raspberry Pi, Bogotá, Colombia: Universidad Distrital Francisco José de Caldas, 2016.
- [39] B. B. M. U. a. C. K. S. Avatar, Mechatronic design of ball on plate balancing system, Elsevier Science- Mechatronics no. 12, 2002.
- [40] D. A. Arias, Control de una plataforma Stewart, Pereira, Colombia: Universidad Tecnológica de Pereira, 2015.
- [41] NoMADA, «NoMADA Pro+». México 2015.
- [42] M. J. B. Díaz, Normas Básicas y Recomendaciones en el Diseño de PCB's, 2015.
- [43] A. Roca, Control Automático de Procesos Industriales, Diaz de Santos, 2014.
- [44] K. Yaovaja, «Ball Balancing on a Stewart Platform using,» *IEEE Xplore*, pp. 170-175, 2018.
- [45] K. M. Moudgalya, Digital Control, Wiley, 2007.
- [46] C.-C. H. B. Z. Qing-Guo Wang, «A frequency response approach to auto-tuning of multivariable PID controllers,» *Elsevier*, 1996.
- [47] R. B. C. C. Diego A. Mejía, «Técnicas de control adaptativas aplicadas a sistemas multivariados (M.I.M.O.),» Scientia et Technica, Pereira, 2007.
- [48] Adept, «Robot Paralelo Adept Quattro s650H». Saltillo México 2017.

## 16 ANEXOS

### 16.1 ANEXO A: GESTION DE CAMBIOS Y MODIFICACIONES

#### 16.1.1 A1: Proceso de Modificación de Juntas

Como se puede apreciar en la Figura 65 el proceso para la obtención del *cero pieza* (x, y) o *alineación concéntrica* de la pieza a manufacturar se realizó empleando la misma herramienta *chuck* de la fresadora CNC, haciendo uso de su propiedad de mordaza se abrazó la junta M8 de tal manera que quedaran unidas concéntricamente para posteriormente ajustar las calzas de la prensa, retirar el chuck, colocar la broca de centros, realizar la marca y finalmente realizar la perforación con la broca M2.5 a una profundidad de 1cm como se puede observar en la Figura 65.

Cabe mencionar que para la realización de esta tarea fue necesario implementar unas calzas metálicas que permitieran el sostén de dicha pieza a manufacturar, ya que la pieza es una junta móvil era practicante imposible sostenerla de la base y evitar el movimiento del enlace o brazo. Es por ello que se optó por inmovilizar directamente el brazo y realizar la perforación directamente sobre el como se muestra en la Figura 65.

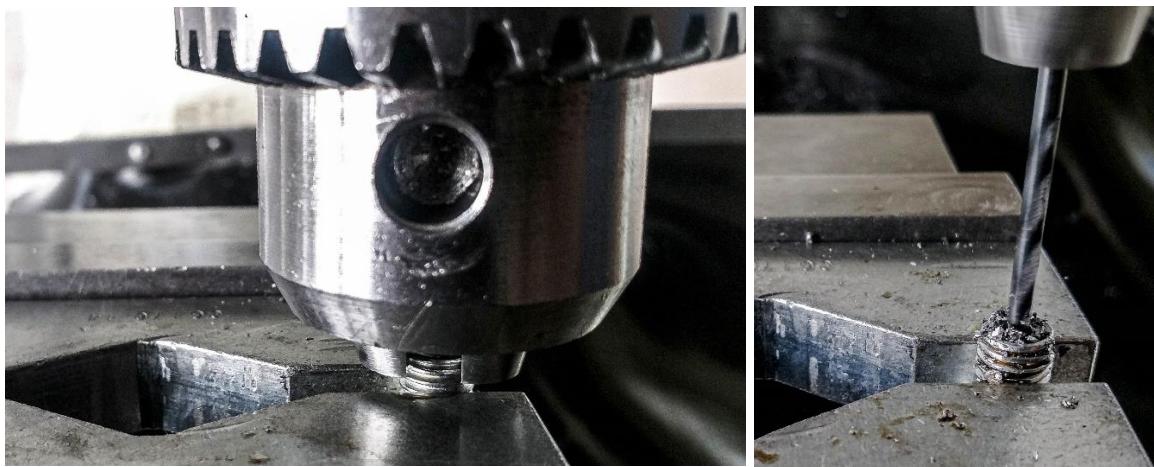


Figura 65 - Adaptación de Rotulas M8 a perforación M3

Una vez que se tienen todas las juntas roscadas se procede a realizar una inmersión en aceite (se empleó un aceite de tipo automotriz 15w40), esto se hace con la intención de lubricar y aflojar un poco la cabeza de la rótula para que tenga un movimiento más libre del que tiene actualmente. Haciendo pruebas se notó que las juntas al no contener ningún tipo de lubricante tienen bastante fricción interna y esto puede ocasionar problemas al implementar el controlador PID. Finalmente se une nuevamente todo el brazo (rotula-enlace-rotula) como se observa en la figura para implementarlo al ensamble final.



Figura 66 - Rotula ensamblada

#### 16.1.2 A2: Selección Del Microcontrolador

Conforme se fue avanzando en el proyecto y su implementación empezaron a hacerse notables ciertas limitaciones con el microcontrolador seleccionado (ATMega328P) entre las cuales involucraban los timer's, que al ser dos (de los tres con los que cuenta) de 8 bits solo se pueden generar frecuencias de aproximadamente 60Hz (16ms) y los servomotores trabajan generalmente con frecuencias de 50Hz (20ms) para controlar la posición de 0°-180°. Realizando pruebas con estos *timer's* se observó que el servomotor no alcanzaba los 180°, generando solo un giro de aproximadamente 120°.

Debido a esto se contempló la implementación mediante conteo de interrupciones por desborde del timer, sin embargo, esto implicaba mayor cálculo numérico y manejo de variables, a esto se le aúna el cálculo para la implementación de la cinemática inversa, además de las interrupciones necesarias para el tiempo de muestreo del PID y una función *millis* que cuente cada milisegundo transcurrido desde que inicia el programa del microcontrolador, esta última servirá para tener un tiempo de referencia y poder trabajar con intervalos de tiempo (dt).

Con todas estas condiciones como obstáculo para implementar el controlador PID se observó el último factor detonante de evaluar nuevamente la selección de un microcontrolador: la memoria flash del ATMega328P estaba trabajando al 80% de su capacidad total. Esto regularmente genera inestabilidad en el programa y era necesario aun implementar todo lo referente al PID. Es por ello que se tomó la decisión de cambiar al microcontrolador ATMega2560 ya que fue una opción y propuesta planteada en caso de que los recursos del ATMega328P se vieran limitados, y por lo tanto se pasó de 32KB a 256KB y 4 *timer's* de 16 bits que pueden generar sin problema señales de 50Hz (20ms).

### 16.1.3 A3: Diseño del Circuito Impreso (PCB)

Una vez que se seleccionó el nuevo microcontrolador a emplear es necesario realizar un rediseño de la PCB, ya que en principio se había seleccionado al ATMega328P por su practicidad de uso por ser un empaquetado de tipo THT y por lo tanto su reemplazo es bastante más sencillo que el de un empaquetado de tipo SMD como lo es el ATMega2560. Ante esta situación se encontró una solución práctica que consiste en adaptar un microcontrolador SMD a THT, esto es posible mediante un adaptador específico denominado TQFP100 que es una PCB sencilla que une los 100 pines del ATMega2560 con pines tipo SIL como se observa en la Figura 67.



Figura 67 - Adaptador SMD a THT para empaquetado TQFP100

Una vez que se tiene el adaptador y el pinout del ATMega2560 (anexo) se procede diseñar la PCB partiendo del diseño original del sistema mínimo propuesto en el *Diseño Electrónico*, y agregando algunas mejoras para su implementación. Lo primero en contemplarse para el sistema mínimo serán los componentes necesarios para que éste pueda funcionar: Se puede observar en la Figura 68 en la parte superior de la placa un capacitor (100nF) y un inductor (10uH) que corresponden a la configuración capacitor/inductor recomendada por el fabricante para implementar el ADC con AVCC referenciada a VCC (anexo C2 16.2.2.3).

En la parte inferior, justo debajo del socket se puede observar una resistencia en configuración PULL-UP paralela a un diodo rectificador de protección contra inversión de polaridad y a uno de los pines de esta configuración se conecta un push-button que fungirá como *reset* del sistema, y paralelo a este está un capacitor de 100nF para prevenir el efecto de rebote en los pulsadores. Esta configuración se puede ver más a detalle en la Figura 68.

Continuando por la parte inferior se coloca el oscilador (para este sistema se seleccionó de 16MHz) y sus respectivos capacitores a tierra. Justo a un lado se agregan los pines de alimentación correspondientes al programador ISP que irá colocado a un lado de la fuente de alimentación del sistema general. Estos son los elementos necesarios para implementar el sistema mínimo al cual solo falta agregar los periféricos de los servos, colocados paralelos entre sí y con alimentación compartida a 6.7V y su respectivo fusible de 10A según lo calculado en el anexo C1 (16.3.1).

Finalmente se agrega el conector de alimentación para el sistema y su respectivo fusible de 2A (según el anexo C1 (16.3.1)), los pines para cada uno de los puertos, el sensor MPU-6050, el panel resistivo y al centro el socket para el adaptador SMD a THT. Debido a limitaciones de laboratorio se optó por realizar las PCB con proveedores extranjeros logrando acabados profesionales a doble capa y con mascara anisodonte. El ensamblaje final se muestra en la figura Figura 71.

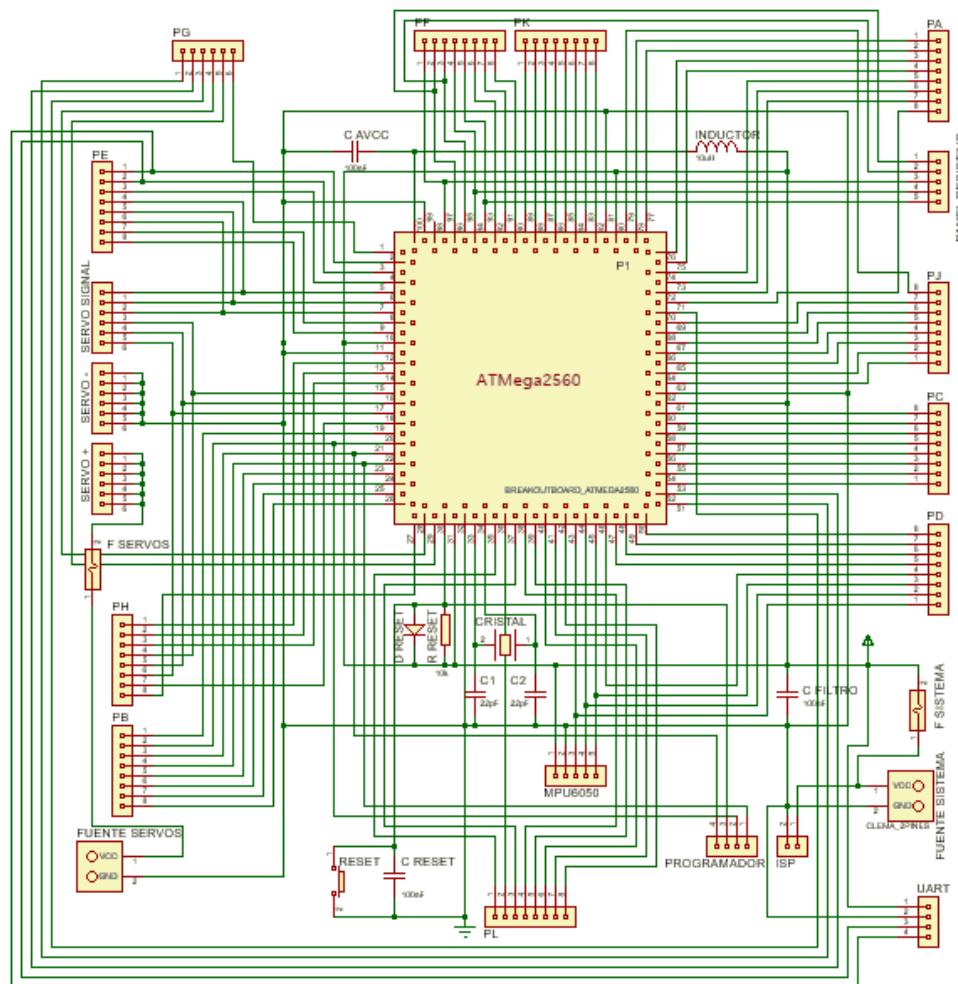


Figura 68 - Diagrama del Sistema Mínimo ATMega2560

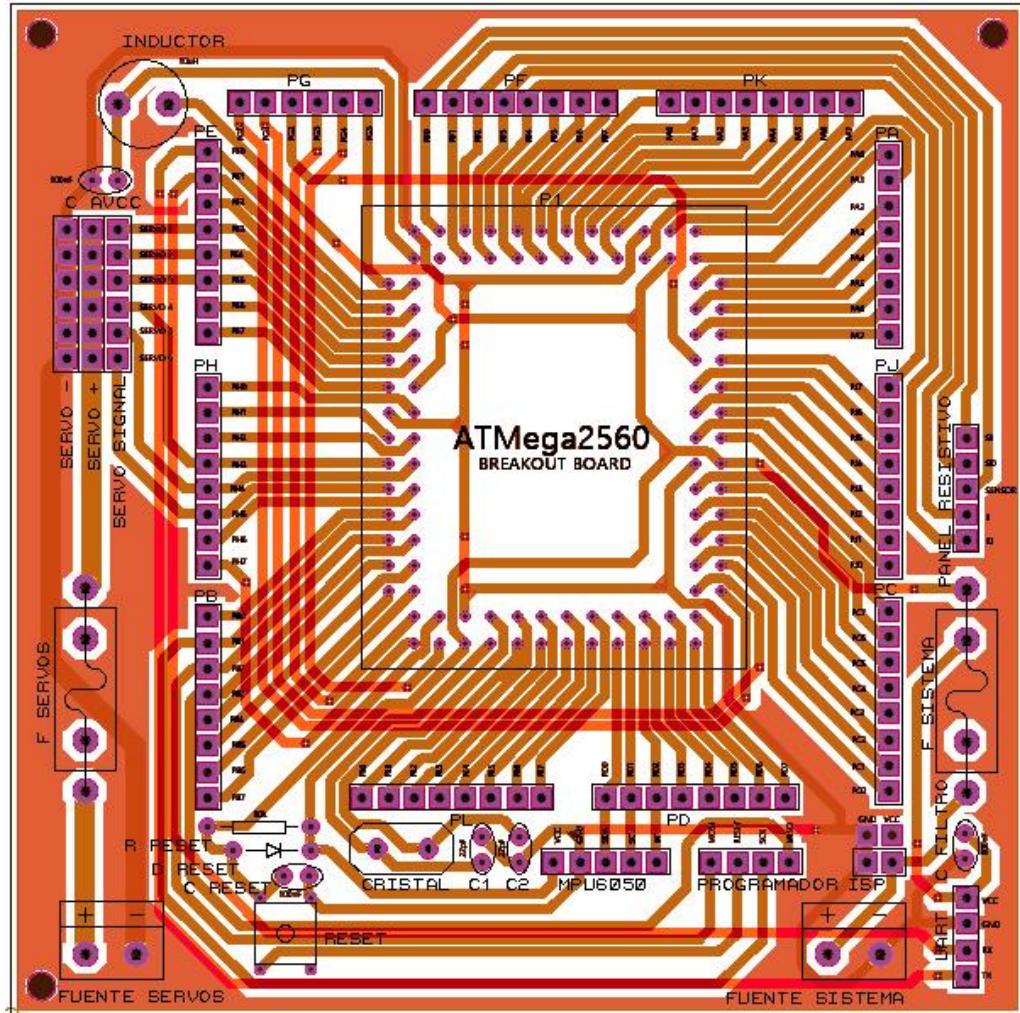


Figura 69 - Diseño de PCB para Sistema Mínimo ATMega2560

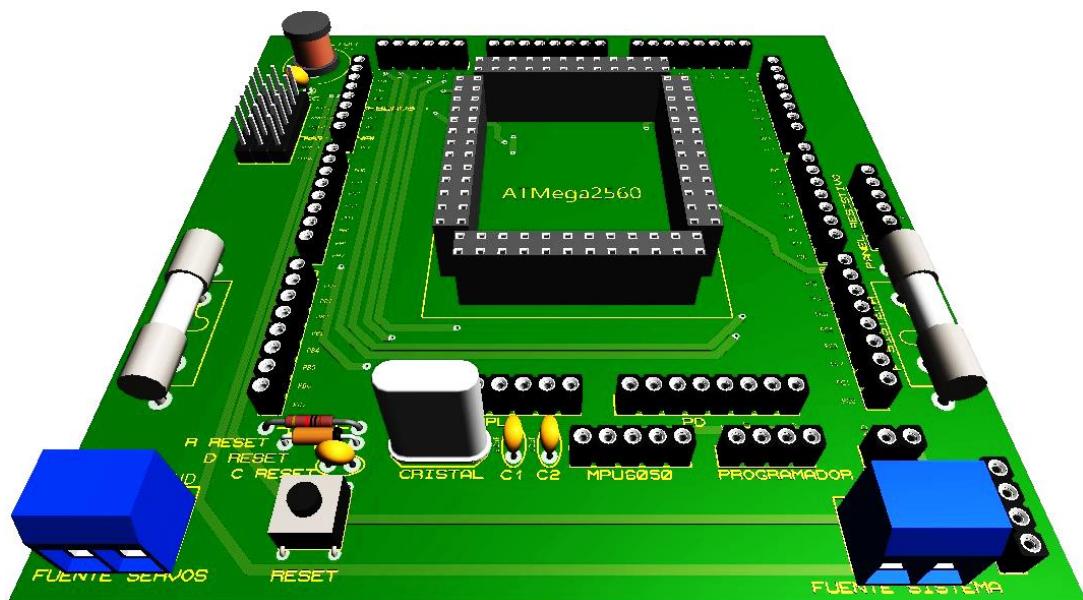


Figura 70 - Visión 3D del Sistema Mínimo ATMega2560

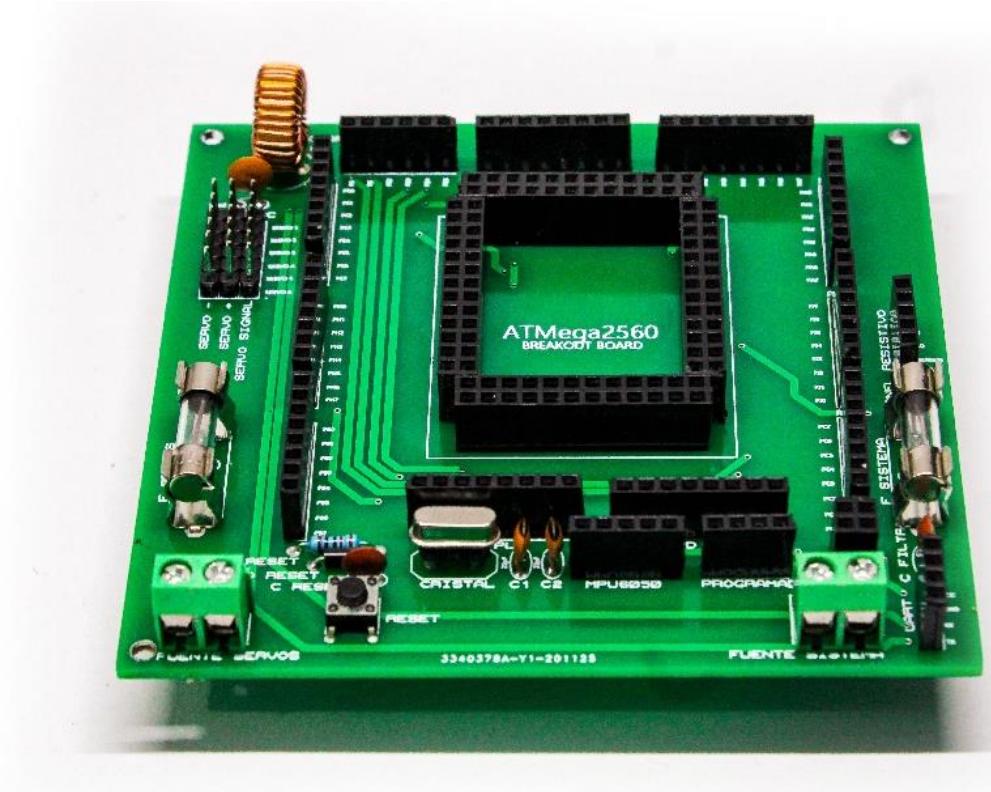


Figura 71 - Ensamble de la PCB Sistema Mínimo del ATMega2560

#### 16.1.1 A5: Corrección del efecto final

Durante el proceso de ensamblaje parcial se detectó un detalle en la pieza de acrílico que corresponde a la base del efecto final, ésta al ser de un material relativamente frágil y ser unida mediante dos placas tendía a fracturarse al realizar las perforaciones justo en medio de ambas, por lo tanto se tuvo que optar por otra opción para su manufactura (impresión 3D en petG). Así mismo, se encontró que al colocar las juntas superiores paralelas entre sí la cabeza de estas colisionaban, por lo tanto se tuvo que realizar una corrección de diseño que pasaba de tener un efecto final con juntas paralelas entre sí y apuntando al centro, a juntas individualmente apuntando al centro como se muestra en la Figura 72.

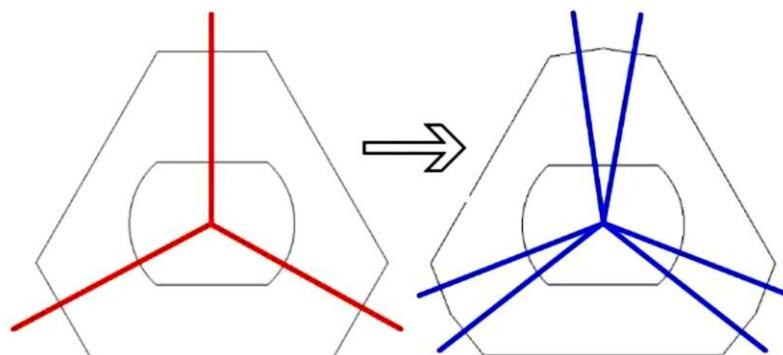


Figura 72 - Modificación del efecto final

### 16.1.2 A5: Cinemática Inversa

La posición de los puntos  $B_i$  y  $P_i$ :

$$B_i = \begin{bmatrix} B_{Xi} \\ B_{Yi} \\ B_{Zi} \end{bmatrix} = \begin{bmatrix} r_b \cos(\mu_i) \\ r_b \sin(\mu_i) \\ 0 \end{bmatrix} \quad (60)$$

$$P_i = \begin{bmatrix} P_{Ui} \\ P_{Vi} \\ P_{Wi} \end{bmatrix} = \begin{bmatrix} r_p \cos(\lambda_i) \\ r_p \sin(\lambda_i) \\ 0 \end{bmatrix} \quad (61)$$

La Figura 73 presenta el diagrama esquemático de los sistemas de referencia:

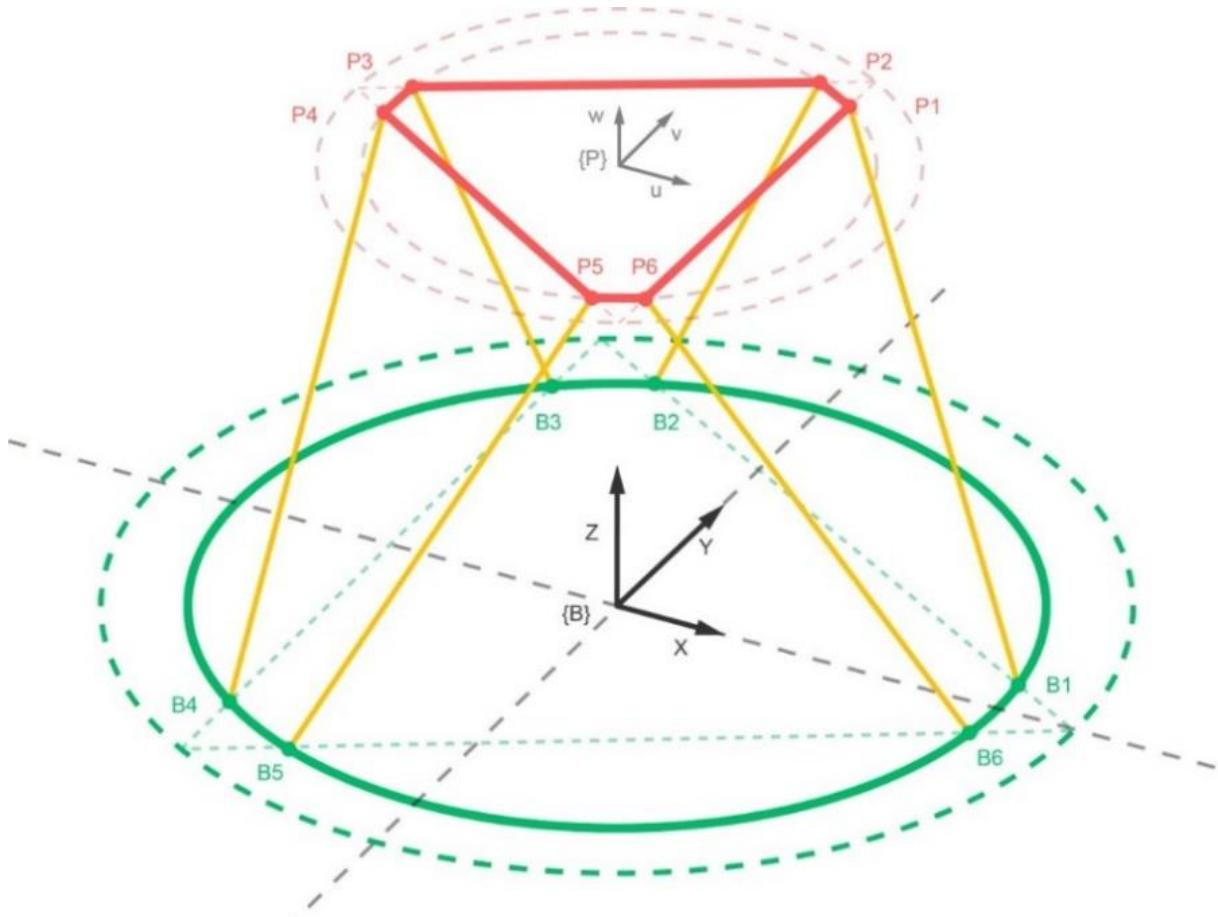


Figura 73 Diagrama esquemático de sistemas de referencia nuevos.

Obsérvese la Figura 74, esta señala el diagrama de posicionamiento de las juntas, suponiendo que el plano  $uv$  del sistema de referencia  $\{P\}$  es paralelo al plano  $xy$ . Ambos sistemas coordinados ( $\{P\}$  y  $\{B\}$ ) están centrados y separados por una distancia vertical respecto a la base, es decir, sobre el eje  $z$  (altura de la plataforma o del efecto final respecto al sistema de coordenadas  $\{B\}$ ):

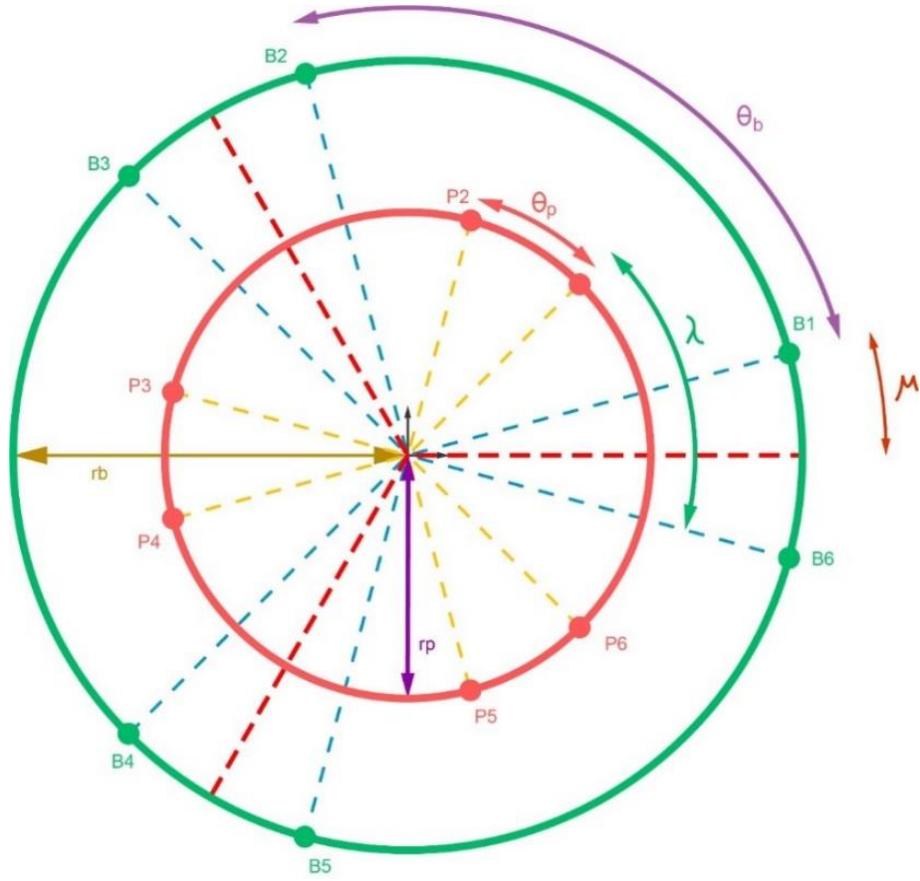


Figura 74 Diagrama de posicionamiento de juntas.

Se hace hincapié en que las uniones van colocadas por pares sobre la circunferencia, tanto en la base como de la plataforma. Para la base, el primer par son las uniones  $B_6$  y  $B_1$ , segundo par  $B_2$  y  $B_3$ , y por último  $B_4$  y  $B_5$ . Nótese que el ángulo formado por las líneas rectas punteadas (color azul cielo), que van del centro a las uniones  $B_i$ , tiene una bisectriz (líneas rectas punteadas de color rojo).

Dado que son tres pares de uniones, la base tiene tres bisectrices. De las cuales se deduce por simetría de la circunferencia que el ángulo entre dos de estas líneas es de  $120^\circ$  o también expresado como  $\frac{2}{3}\pi$ . Este valor le corresponde a  $\theta_b$ , y es el ángulo formado por las líneas rectas de los puntos  $B_i$  aledaños que estén separados por un ángulo mayor respecto a de su punto con el que se agrupó. Dicho de otra manera, este ángulo está formado por las líneas rectas de  $B_1$  y  $B_2$ ,  $B_3$  y  $B_4$ ,  $B_5$  y  $B_6$ .

Por el contrario, el ángulo  $\theta_p$  se forma con la abertura que hay entre las líneas rectas punteadas de los puntos  $P_i$  de su par agrupado (líneas punteadas color marrón) de la plataforma. Las distancias  $r_p$  y  $r_b$  son los radios de la plataforma y la base, respectivamente.

Para el posicionamiento de las uniones de la base se toma en cuenta el radio  $r_p$  y, a su vez, el ángulo entre bisectrices que es  $\frac{2}{3}\pi$ , como ya se mencionó con anterioridad. Este dato se deduce de observar la simetría del nuevo diagrama del posicionamiento de juntas de la figura 24, para definir  $B_1$  respecto al sistema  $\{B\}$  por medio del ángulo  $\mu_1$ :

$$\frac{2}{3}\pi = 2\mu_1 + \theta_p \quad (62)$$

Por lo tanto, al despejar  $\mu_1$ :

$$\mu_1 = \frac{\pi}{3} - \frac{\theta_p}{2} \quad (63)$$

Para  $B_2$ , por simetría:

$$\mu_2 = \mu_1 + \theta_p \quad (64)$$

Para  $B_3$ :

$$\mu_3 = \frac{2}{3}\pi + \mu_1 \quad (65)$$

$$\mu_3 = \pi - \frac{\theta_p}{2} \quad (66)$$

Para  $B_4$ :

$$\mu_4 = \mu_3 + \theta_p \quad (67)$$

Para  $B_5$ :

$$\mu_5 = \frac{4}{3}\pi + \mu_1 \quad (68)$$

$$\mu_5 = \frac{5}{3}\pi - \frac{\theta_p}{2} \quad (69)$$

Para  $B_6$ :

$$\mu_6 = \mu_5 + \theta_p \quad (70)$$

El primer vector columna será el de las uniones de la base:

$$B_i = \begin{bmatrix} B_{xi} \\ B_{yi} \\ B_{zi} \end{bmatrix} = \begin{bmatrix} r_b \cos(\mu_i) \\ r_b \sin(\mu_i) \\ 0 \end{bmatrix} \quad (71)$$

$$\mu_i = \frac{i\pi}{3} - \frac{\theta_p}{2}, \quad \text{para } i = 1, 3, 5 \quad (72)$$

$$\mu_i = \mu_{i-1} + \theta_p, \quad \text{para } i = 2, 4, 6 \quad (73)$$

De manera análoga, el segundo vector se determina por medio de igualdades de ángulos inferidos por el diagrama de posicionamiento de juntas, sólo que ahora en vez del ángulo  $\mu_i$  y el radio  $r_b$  se utilizan  $\lambda_i$  y  $r_p$  respectivamente:

$$\mathbf{P}_i = \begin{bmatrix} P_{Ui} \\ P_{Vi} \\ P_{Wi} \end{bmatrix} = \begin{bmatrix} r_p \cos(\lambda_i) \\ r_p \sin(\lambda_i) \\ 0 \end{bmatrix} \quad (74)$$

$$\lambda_i = \frac{i\pi}{3} - \frac{\Theta_p}{2}, \quad \text{para } i = 1, 3, 5 \quad (75)$$

$$\lambda_i = \lambda_{i-1} + \Theta_p, \quad \text{para } i = 2, 4, 6 \quad (76)$$

Obsérvese que el sistema de coordenadas ahora es  $\{P\}$ .

Una vez obtenidas las posiciones de las uniones necesitamos la postura de la plataforma (también llamada *efector final*), que se denotará como el vector  $E_F$  que determina la traslación del sistema de coordenadas móvil ( $\{P\}$ ) con respecto a la base:

$$E_F = [E_{FX} \ E_{FY} \ E_{FZ}]^T \quad (77)$$

Y una rotación  ${}^B R_P$  que está definida por los ángulos de Euler alabeo, cabeceo y guiñada (roll, pitch, yaw respectivamente). Estos ángulos corresponden a una rotación  $\alpha$  sobre el eje  $x$ ,  $\beta$  sobre el eje  $y$  y por último  $\gamma$  sobre el eje  $z$ .

$$\begin{aligned} {}^B R_P &= R_Z(\gamma)R_Y(\beta)R_X(\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\ &= \begin{bmatrix} c(\beta)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) & s(\alpha)s(\gamma) + c(\alpha)s(\beta)c(\gamma) \\ c(\beta)s(\gamma) & c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma) & c(\alpha)s(\beta)c(\gamma) - s(\alpha)c(\gamma) \\ -s(\beta) & c(\beta)s(\alpha) & c(\alpha)c(\beta) \end{bmatrix} \quad (78) \end{aligned}$$

En la matriz  $c$  significa “coseno” y  $s$  “seno”.

Después de suponer que los sistemas  $\{P\}$  y  $\{B\}$  tienen sus planos  $xy$  y  $uv$  paralelos y sus orígenes están centrados, se muestra la rotación y la traslación del efector final:

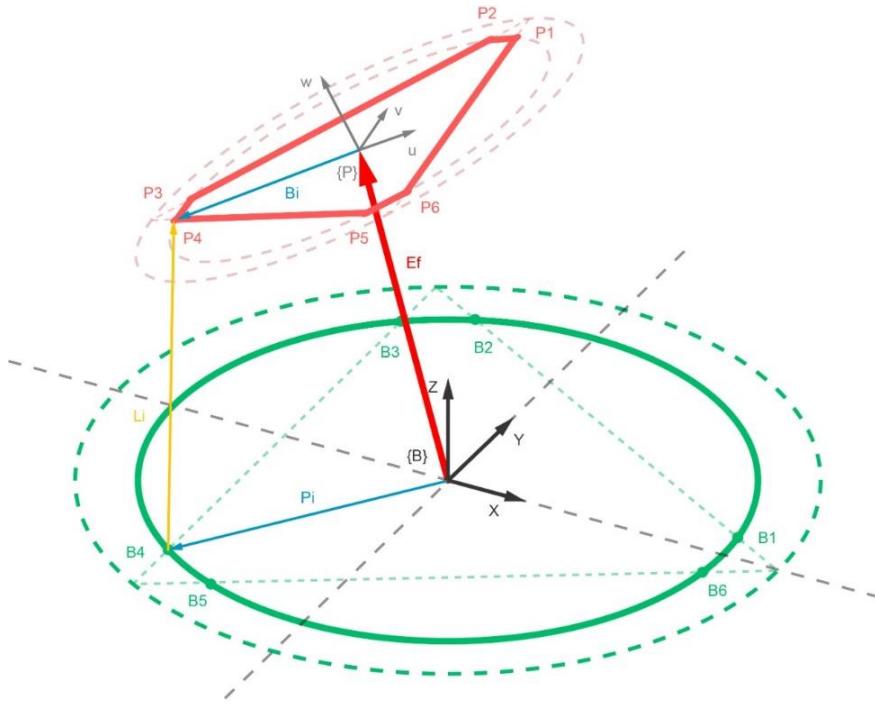


Figura 75 Diagrama del efecto final rotado y trasladado.

Donde:

$L_i$  es el vector que determina la distancia del actuador.

Hasta aquí, se ha referenciado la cinemática inversa con la Plataforma Stewart-Gough de [13], pero existe una diferencia entre los actuadores lineales que se utilizaron y los que se implementarán en este proyecto, como ya se mencionó al inicio de esta sección de cinemática inversa similar a la plataforma Stewart de Aristizabal [40].

La configuración que utilizó este autor es similar a una biela manivela, por lo tanto, se tiene el servomotor seguido del brazo que va unido a la biela con una rótula en cada extremo, la figura 26 es una captura que se encuentra en el documento y es parte del diseño CAD de este proyecto:

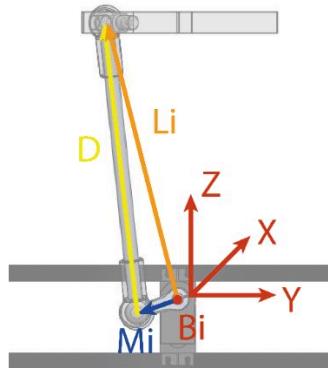


Figura 76 Diseño CAD Plataforma Stewart.

La disposición del conjunto de vectores  $L_i$  determinará altura del efecto final. Dicho de otra manera, las uniones  $B_i$  de la base estarán físicamente en el centro del eje del servomotor. La magnitud del vector  $L_i$ , que va desde el servomotor hasta el centro del eje de la rótula superior, es la *altura efectiva o neta*.

La altura que existe (sobre el eje Z del sistema  $\{P\}$ ) desde el punto más bajo de la plataforma hasta el efecto final se le conoce como *altura absoluta*. Pero en este proyecto se considerará la altura efectiva.

Entonces el vector  $L_i$  se define como:

$$\begin{bmatrix} L_{Xi} \\ L_{Yi} \\ L_{Zi} \end{bmatrix} = \begin{bmatrix} E_{FX} \\ E_{FY} \\ E_{FZ} \end{bmatrix} + \begin{bmatrix} c(\beta)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) & s(\alpha)s(\gamma) + c(\alpha)s(\beta)c(\gamma) \\ c(\beta)s(\gamma) & c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma) & c(\alpha)s(\beta)c(\gamma) - s(\alpha)c(\gamma) \\ -s(\beta) & c(\beta)s(\alpha) & c(\alpha)c(\beta) \end{bmatrix} \begin{bmatrix} P_{Ui} \\ P_{Vi} \\ P_{Wi} \end{bmatrix} - \begin{bmatrix} B_{Xi} \\ B_{Yi} \\ B_{Zi} \end{bmatrix} \quad (79)$$

### 16.1.3 A7: Cinemática Directa

Para el cálculo de la cinemática directa se utilizó la metodología clásica de Denavit-Hartenberg [5]. La Figura 77 muestra el resultado del modelo final de la plataforma Stewart:

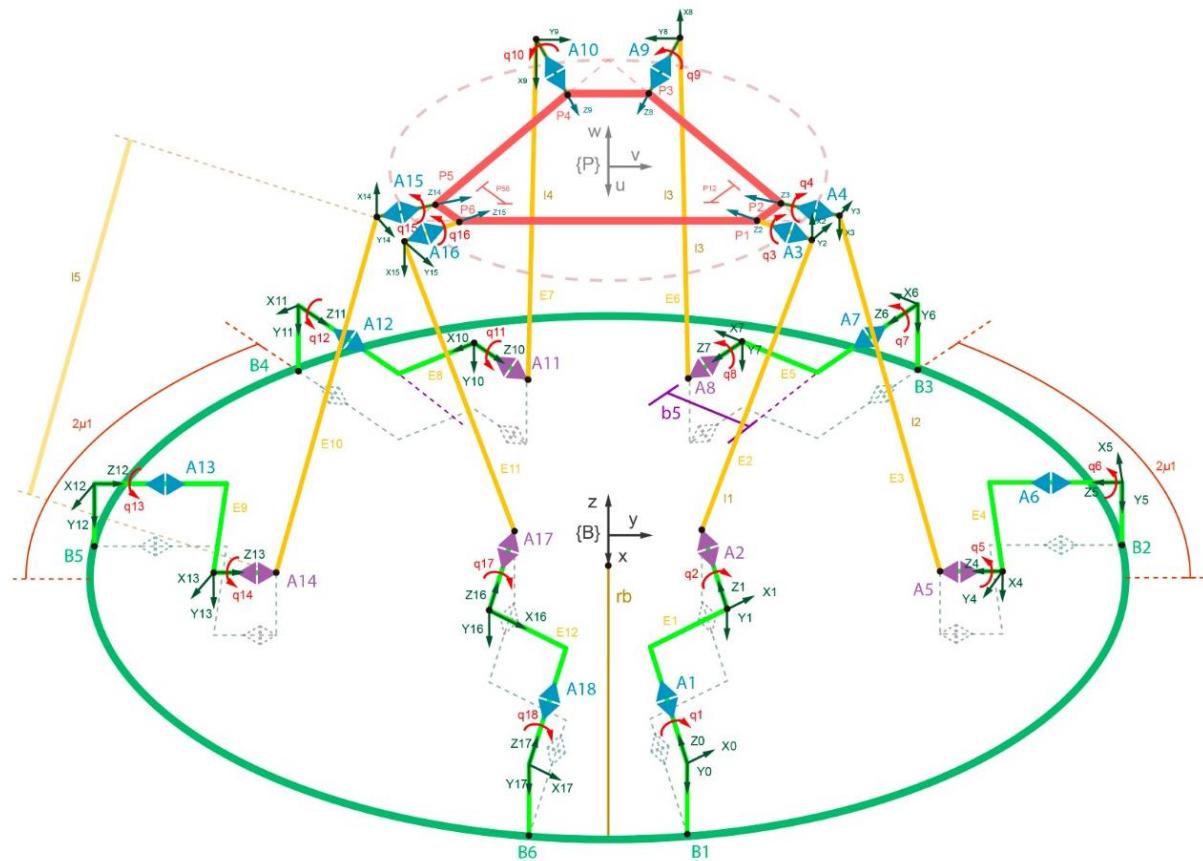


Figura 77 Modelado de cinemática directa por metodología Denavit-Hartenberg.

Nótese que la asignación de sistemas de coordenadas  $\{S_i\}$  y ángulos de giro  $q_i$  para cada una de las articulaciones (denotadas  $A_i$ , donde  $i$  va desde 1 hasta 18) están numeradas siguiendo la cadena cinemática. La articulación  $A_1$  es el primer servomotor del sistema. Las articulaciones  $A_2$  y  $A_3$  son las rótulas inferior y superior accionadas directamente por  $A_1$ .

Se observa que el orden de numeración hasta aquí es ascendente, pero en  $A_4$  cambia, pues si en la primera pierna se numeró el servomotor, después rótula inferior y por último rótula superior, para la segunda pierna se numera en orden descendente respecto a la primer pierna. Y así se continúa la numeración hasta llegar a la articulación  $A_{18}$ . Cabe destacar que entre los orígenes de los sistemas de coordenadas  $O_0, O_5, O_6, O_{11}, O_{12}$  y  $O_{17}$  (pertenecientes a los servomotores) están en el mismo plano y coordenada que su respectivo punto  $B_i$ . En otras palabras, la circunferencia color verde está posicionada de tal manera que todos y cada uno de los orígenes de los servomotores están sobre ella.

Para simplificar el modelado se supuso que las circunferencias de base y plataforma se “parten” justo por la mitad (**M1**), y así formar una línea recta con dos de las circunferencias del modelo matemático; las cuales pueden observarse en la Figura 78, una de ellas es la ya antes mencionada (mostrada de color verde) y la otra es la circunferencia punteada de mayor diámetro. La línea recta mencionada en el párrafo anterior es perpendicular a los ejes  $z_i$  de cada articulación. Además, estos ejes quedan apuntando en la misma dirección, por lo tanto, se consideran son paralelos. Esto se presenta también en la Figura 78:

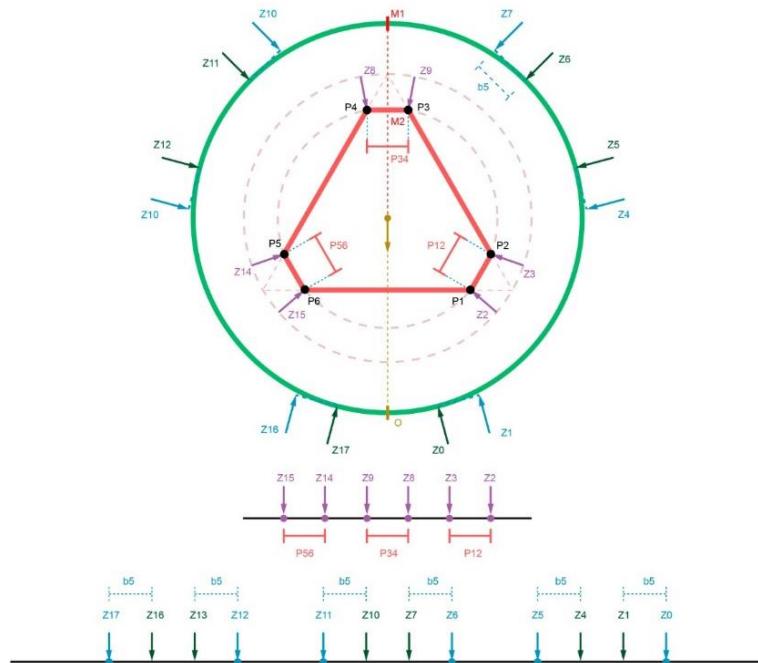


Figura 78 Vista superior de ejes z de articulaciones.

Una vez aplicados los pasos de la metodología Denavit-Hartenberg (al que se le llamará “DH”) se obtiene la *cinemática directa* con los valores de los parámetros que se desean obtener para la matriz de transformación con lo cual se obtendrá la posición del efecto final:

Tabla 5 Parámetros Denavit-Hartenberg calculados de Plataforma Stewart.

<b>Articulación</b>	<b><math>\theta</math></b>	<b><math>d</math></b>	<b><math>a</math></b>	<b><math>\alpha</math></b>
<b>0</b>	$\frac{\pi}{2}$	0	$r_b$	$-\frac{\pi}{2}$
<b>1</b>	$q_1$	0	$b_s$	$\frac{\pi}{9}$
<b>2</b>	$q_2 - \frac{\pi}{2}$	0	$l_1$	$\frac{\pi}{9}$
<b>3</b>	$q_3 + \pi$	0	$p_{12}$	$\frac{\pi}{9}$
<b>4</b>	$q_4$	0	$l_2$	$\frac{\pi}{9}$
<b>5</b>	$q_5 - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>6</b>	$q_6$	0	$2r_b \mu_1$	$\frac{\pi}{9}$
<b>7</b>	$q_7$	0	$b_s$	$\frac{\pi}{9}$
<b>8</b>	$q_8 - \frac{\pi}{2}$	0	$l_3$	$\frac{\pi}{9}$
<b>9</b>	$q_9 + \pi$	0	$p_{34}$	$\frac{\pi}{9}$
<b>10</b>	$q_{10}$	0	$l_4$	$\frac{\pi}{9}$
<b>11</b>	$q_{11} - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>12</b>	$q_{12}$	0	$2r_b \mu_1$	$\frac{\pi}{9}$
<b>13</b>	$q_{13}$	0	$b_s$	$\frac{\pi}{9}$
<b>14</b>	$q_{14} - \frac{\pi}{2}$	0	$l_5$	$\frac{\pi}{9}$
<b>15</b>	$q_{15} + \pi$	0	$p_{56}$	$\frac{\pi}{9}$
<b>16</b>	$q_{16}$	0	$l_6$	$\frac{\pi}{9}$
<b>17</b>	$q_{17} - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>18</b>	$q_{18}$	0	$2r_b \mu_1$	$\frac{\pi}{9}$

La articulación 0 de la tabla anterior se considera un matriz de transformación que, en primer lugar, implica una rotación con un ángulo  $\theta_0$  respecto al eje Z del sistema de coordenadas  $\{B\}$  (con el fin de hacer los ejes X y  $x_0$  paralelos y poder cumplir el método de DH) y, posterior a ello una traslación para hacer coincidentes los orígenes O y  $O_0$ . Hasta aquí se definieron por cada articulación los ángulos  $q_i$  desde 1 hasta 18, numerados así para seguir la cadena cinemática desde el primer motor (sistema  $\{S_1\}$  perteneciente a la articulación  $A_1$ ) hasta regresar al motor 6 (sistema  $\{S_{18}\}$  perteneciente a la articulación  $A_{18}$ ).

Pero debido a que los únicos ángulos controlados van a ser los ángulos pertenecientes a los actuadores ( $q_1, q_6, q_7, q_{12}, q_{13}$  y  $q_{18}$ ) existe dependencia entre los ángulos de las articulaciones que representan las rótulas en el modelo cinemático y su respectiva articulación de actuador (servomotor).

Entonces los ángulos  $q_2$  y  $q_3$  dependen del ángulo  $q_1$ , puesto que pertenecen al mismo enlace entre la articulación que va del cuerno (rótula inferior que le pertenece ángulo  $q_2$ ) a la rótula que une a la plataforma con la pierna (rótula superior, le corresponde el ángulo  $q_3$ ). De este modo pondremos que para el parámetro  $\theta$  los ángulos  $q_2$  y  $q_3$  serán reemplazados por  $q_1$ . Del mismo modo se renombrarán los ángulos de los actuadores  $q_i$  con valores del 1 al 6. De manera análoga, el giro de las rótulas se reemplaza por su respectivo giro de actuador y, por lo tanto, la nueva tabla de parámetros Denavit-Hartenberg quedará de la siguiente manera:

Tabla 6 Tabla modificada de parámetros Denavit-Hartenberg.

<b>Articulación</b>	<b><math>\theta</math></b>	<b><math>d</math></b>	<b><math>a</math></b>	<b><math>\alpha</math></b>
<b>0</b>	$\frac{\pi}{2}$	0	$r_b$	$-\frac{\pi}{2}$
<b>1</b>	$q_1$	0	$b_s$	$\frac{\pi}{9}$
<b>2</b>	$q_1 - \frac{\pi}{2}$	0	$l$	$\frac{\pi}{9}$
<b>3</b>	$q_1 + \pi$	0	$p$	$\frac{\pi}{9}$
<b>4</b>	$q_2$	0	$l$	$\frac{\pi}{9}$
<b>5</b>	$q_2 - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>6</b>	$q_2$	0	$2r_b \mu_1$	$\frac{\pi}{9}$

<b>7</b>	$q_3$	0	$b_s$	$\frac{\pi}{9}$
<b>8</b>	$q_3 - \frac{\pi}{2}$	0	$l$	$\frac{\pi}{9}$
<b>9</b>	$q_3 + \pi$	0	$p$	$\frac{\pi}{9}$
<b>10</b>	$q_4$	0	$l$	$\frac{\pi}{9}$
<b>11</b>	$q_4 - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>12</b>	$q_4$	0	$2r_b \mu_1$	$\frac{\pi}{9}$
<b>13</b>	$q_5$	0	$b_s$	$\frac{\pi}{9}$
<b>14</b>	$q_5 - \frac{\pi}{2}$	0	$l$	$\frac{\pi}{9}$
<b>15</b>	$q_5 + \pi$	0	$p$	$\frac{\pi}{9}$
<b>16</b>	$q_6$	0	$l$	$\frac{\pi}{9}$
<b>17</b>	$q_6 - \frac{\pi}{2}$	0	$b_s$	$\frac{\pi}{9}$
<b>18</b>	$q_6$	0	$2r_b \mu_1$	$\frac{\pi}{9}$

Nótese que las longitudes  $l_1, l_2, l_3, l_4, l_5$  y  $l_6$  (longitudes lineales entre el centro geométrico de rótula inferior su respectiva rótula superior) se cambiaron por  $l$  debido a que todas las longitudes poseen el mismo valor. Lo mismo sucedió para las distancias  $p$  (son los valores de longitud de los 3 lados más pequeños del hexágono irregular representado de color rojo en la Figura 78).

#### 16.1.4 A7: Sintonización Del Controlador PID

La sintonización del controlador se realizó en el dominio de Laplace.

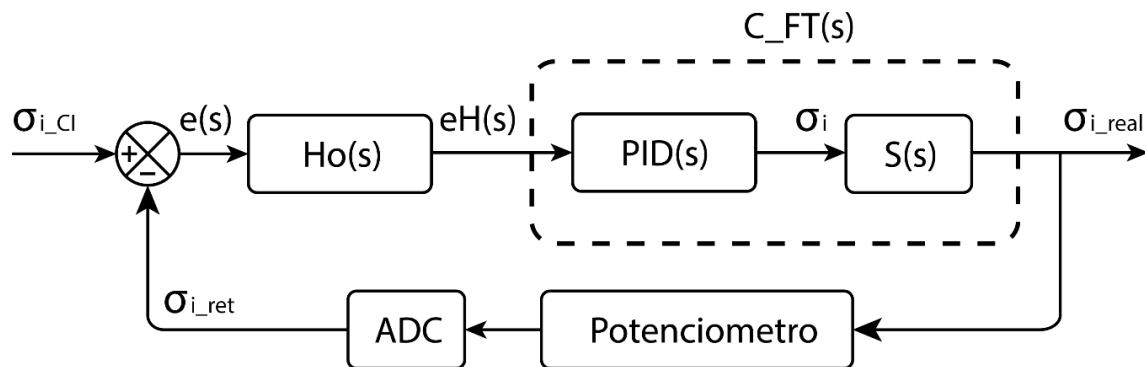


Figura 79 Diagrama de bloques controlador PID servomotores. Bloque CFT(s).

El diagrama de bloques expuesto en el diagrama es (Figura 79):

$$C_{FT}(s) = \frac{s_{i\_real}(s)}{eH(s)} = \frac{PID(s) * S(s)}{1 + PID(s) * S(s)} \quad (80)$$

Los bloques poseen el controlador PID y la función de transferencia del servomotor:

$$PID(s) = k_P + \frac{k_I}{s} + k_D s \quad (81)$$

$$S(s) = \frac{s_{i\_real}(s)}{s_i(s)} = \frac{K_m}{t_m + 1} \quad (82)$$

Y como función de transferencia total:

$$C_{FT}(s) = \frac{s_{i\_real}(s)}{s_i(s)} = \frac{K_m k_D s^2 + K_m k_P s + K_m k_I}{(t_m + K_m k_D)s^2 + (K_m k_P + 1)s} \dots (4) \quad (83)$$

Considerando que el sistema  $S(s)$  tiene el polo en " $s = \frac{1}{t_m}$ " y se propone que el controlador incremente en 5 veces la velocidad del sistema, es decir:  $s = \frac{5}{t_m}$ .

Al determinar los polos de se calcula  $k_D = 1.784988313 \times 10^{-3}$  se propone  $k_P = 1$  y se calcula  $k_I = 41.45848337$ .

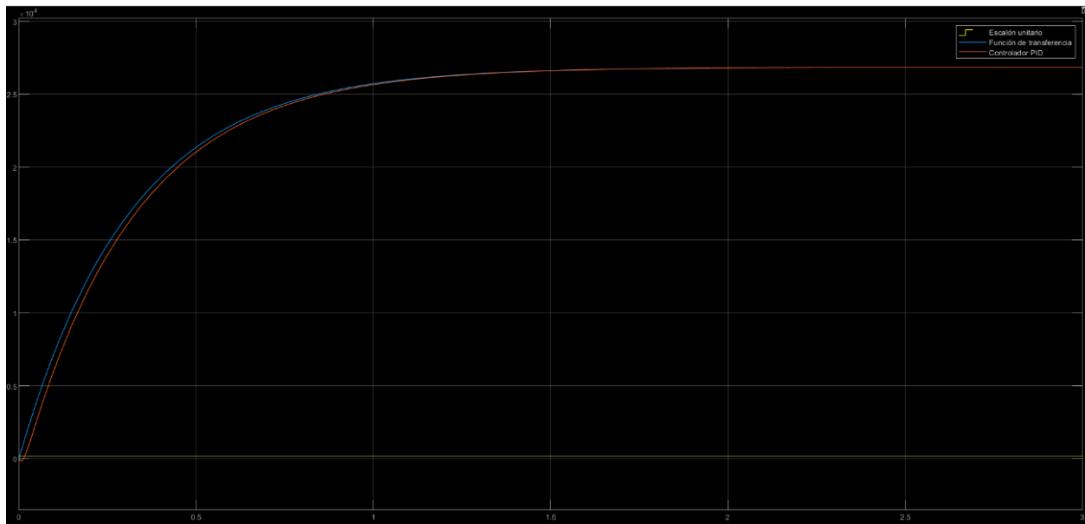


Figura 80 Gráfica ganancias  $K_p$ ,  $K_i$  y  $K_d$ .

## 16.2 ANEXO B: CARACTERISTICAS DE COMPONENTES ELECTRONICOS

### 16.2.1 B1: ATMega328P

#### 16.2.1.1 Características ATMega328P

- 32 Kbytes Memoria Flash.
- 1 Kbytes EEPROM.
- 6 canales PWM.
- 6 canales ADC de 10 bits.
- Periféricos de comunicación: 1 canal USART, I2C, SPI.
- Velocidad de Reloj Hasta 20MHz (Oscilador interno de 8MHz).

- 0 - 4MHz / 1.8 - 5.5V.
- 0 - 16MHz / 2.7 - 5.5V.
- 0 - 20MHz / 4.5 - 5.5V.
- Voltaje de Operación 3.3V - 5V.
- Pines I/O 20.
- Voltaje de alimentación 5V.
- Corriente máxima 1 A.
- Voltaje de operación: 1.8 a 5.5 VDC.
- Arquitectura de CPU: 8 bit AVR.

### 16.2.1.2 Pinout ATMega328P

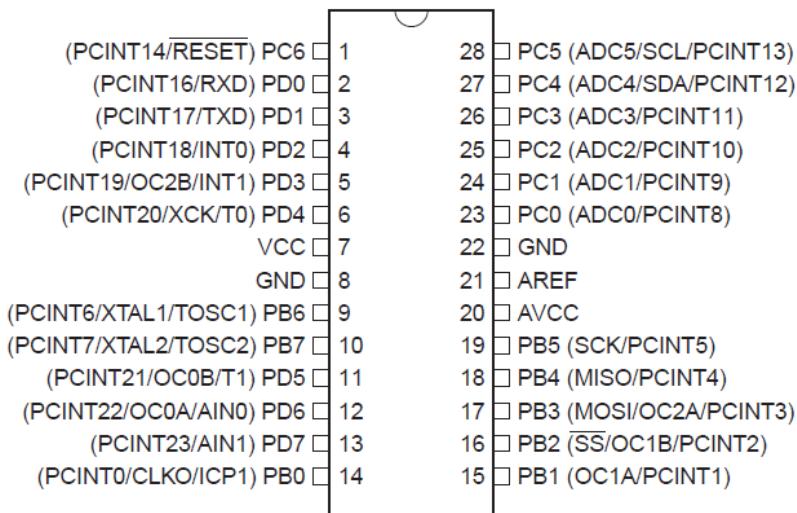


Figura 81 PinOut ATMega328P.

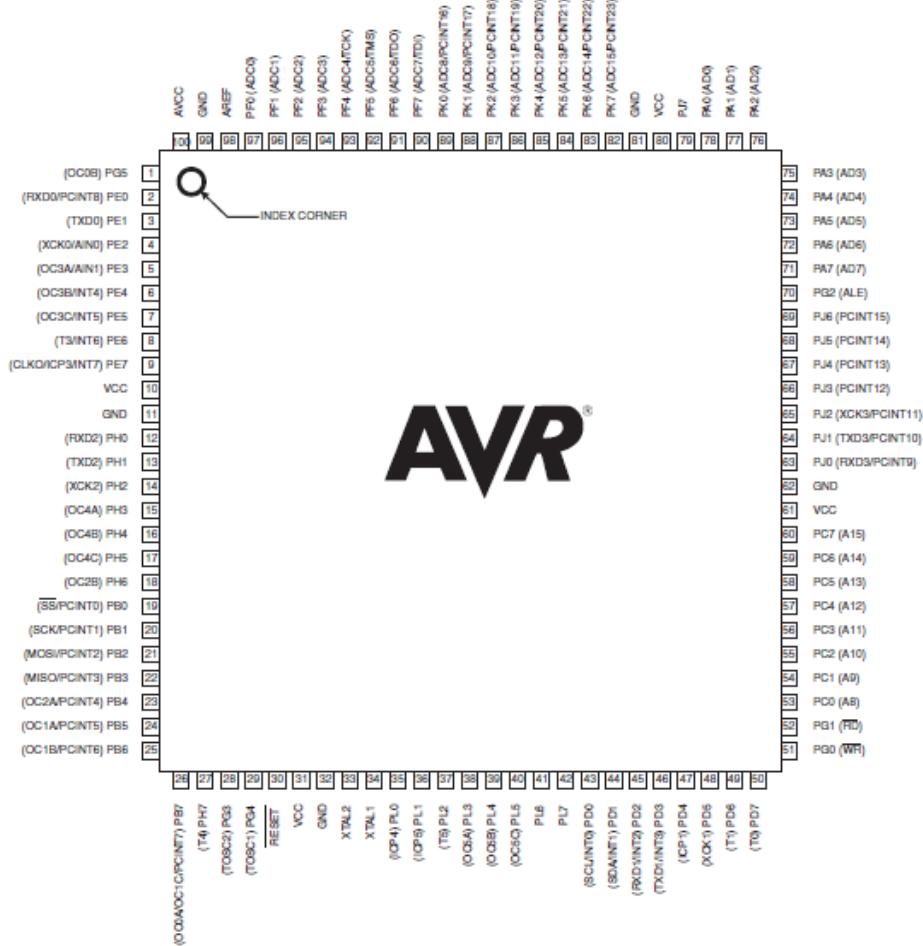
### 16.2.2 B2: ATMega2560

#### 16.2.2.1 Características ATMega2560

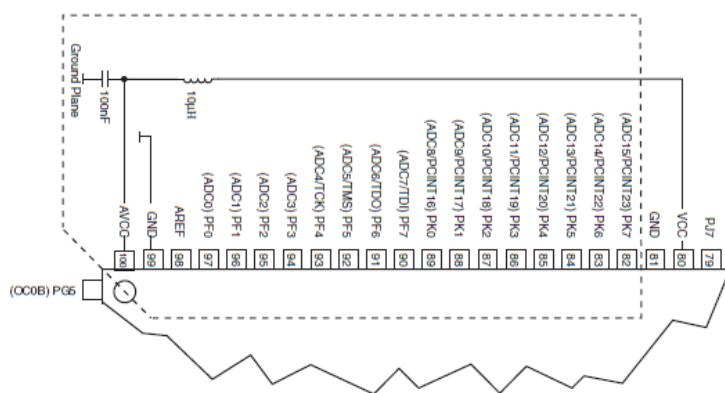
- Arquitectura de 8 bits de baja potencia
- 256KB de memoria flash para programación
- 4Kbytes EEPROM
- Tamaño de datos RAM: 8KB
- Protocolos de comunicación:
  - Two Wire
  - SPI
  - USART (4)
- Velocidad de Reloj Hasta 20MHz (Oscilador interno de 8MHz)
  - 0 - 4MHz / 1.8 - 5.5V.
  - 0 - 16MHz / 2.7 - 5.5V.
  - 0 - 20MHz / 4.5 - 5.5V.
- Puertos de entrada/salida programables: 86
- Temporizadores: 6
  - Dos de 8 bits (3 comparadores c/u)
  - Cuatro de 16 bits (3 comparadores c/u)

- Canales de ADC: 16 canales de 10 bits
- Empaquetado: TQFP-100

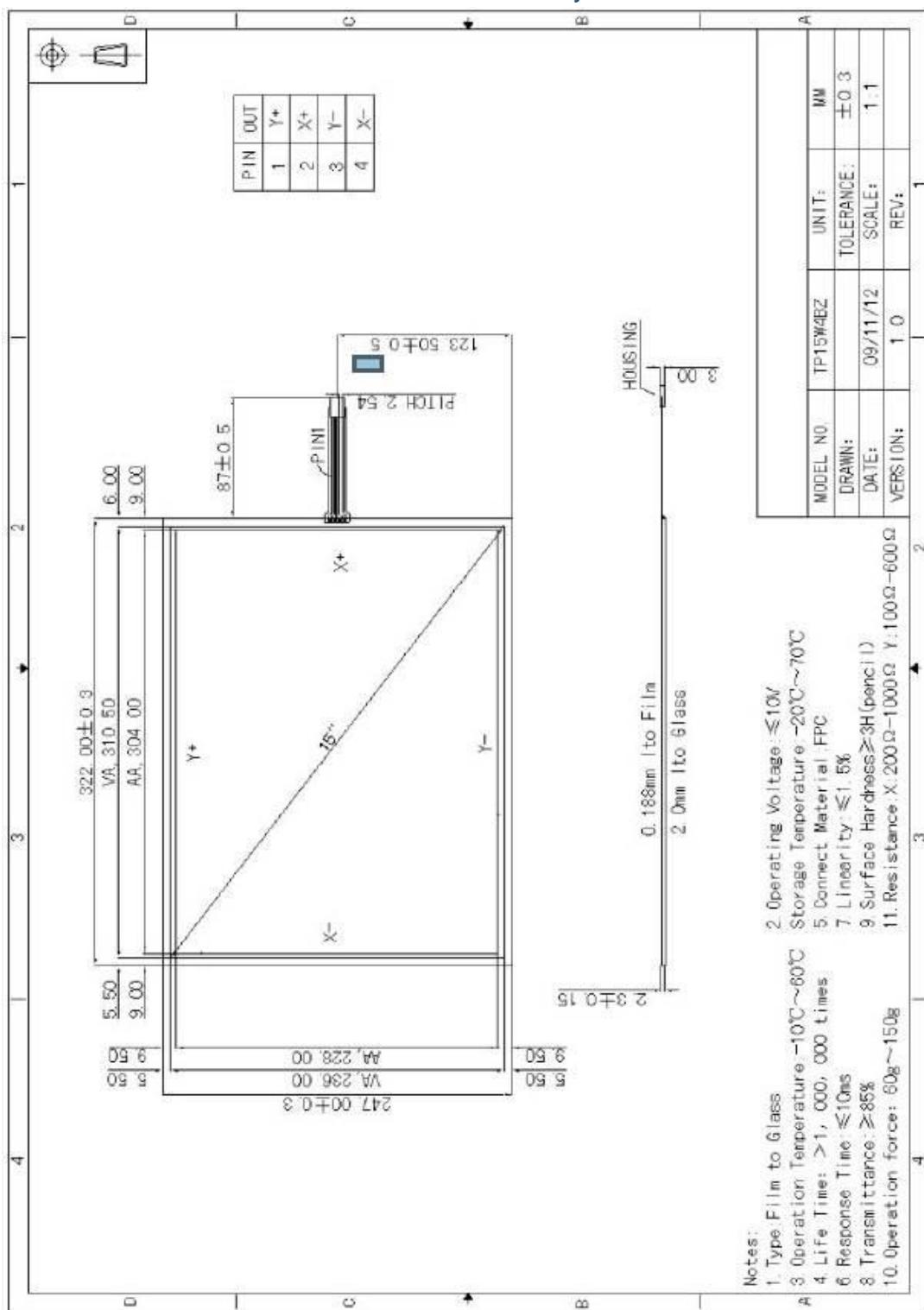
### 16.2.2.2 Pinout ATMega2560



### 16.2.2.3 Configuración Capacitor/Inductor para la implementación del ADC ATMega2560



### 16.2.3 B3: Panel Resistivo – Osatych

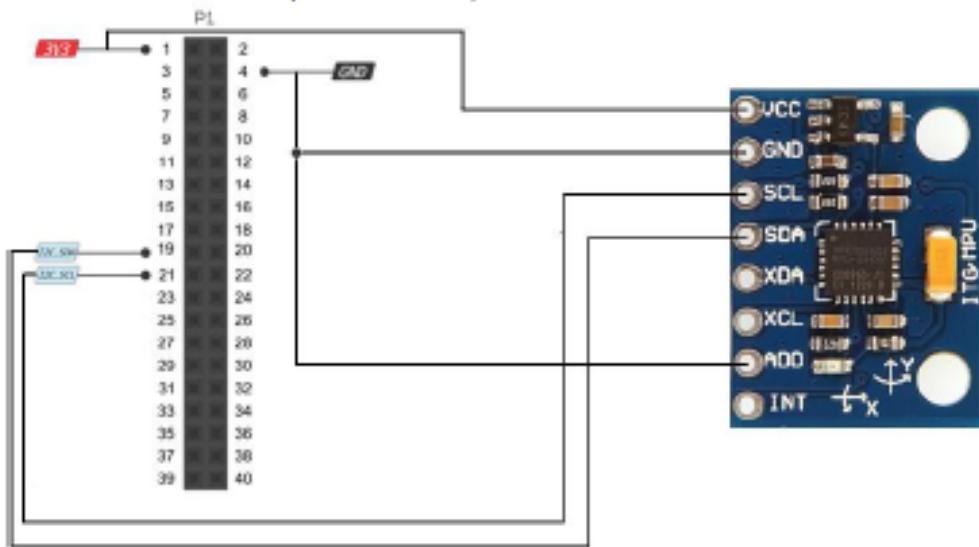


#### 16.2.4 B4: Giroscopio MPU-6050

Placa de expansión	MPU6050	Descripción
-	1 CLKIN	Entrada de referencia externa de clock (opcional)
XDA	6 AUX_DA	BUS maestro de datos I2C, para conectar un sensor externo
XCL	7 AUX_CL	BUS maestro de datos I2C, para conectar un sensor externo
Vcc	8 VLOGIC	Referencia para salida digital
ADD	9 ADD	LSB para la dirección de esclavo en el protocolo I2C
-	10 REGOUT	Conector para filtro regulador
-	11 FSYNC	Sincronización
INT	12 INT	Interrupción digital
Vcc	13 VDD	Tensión de alimentación
GND	18 GND	Tierra
-	19 RESV	Reservado. No conectar
-	20 CPOUT	Capacitor
-	21 RESV	Reservado. No conectar
-	22 RESV	Reservado. No conectar
SCL	23 SCL	Bus de entrada de clock I2C esclavo
SDA	24 SDA	Bus de datos I2C esclavo
-	2	
-	3	
-	4	
-	5	
-	14	
-	15	
-	16	
-	17	

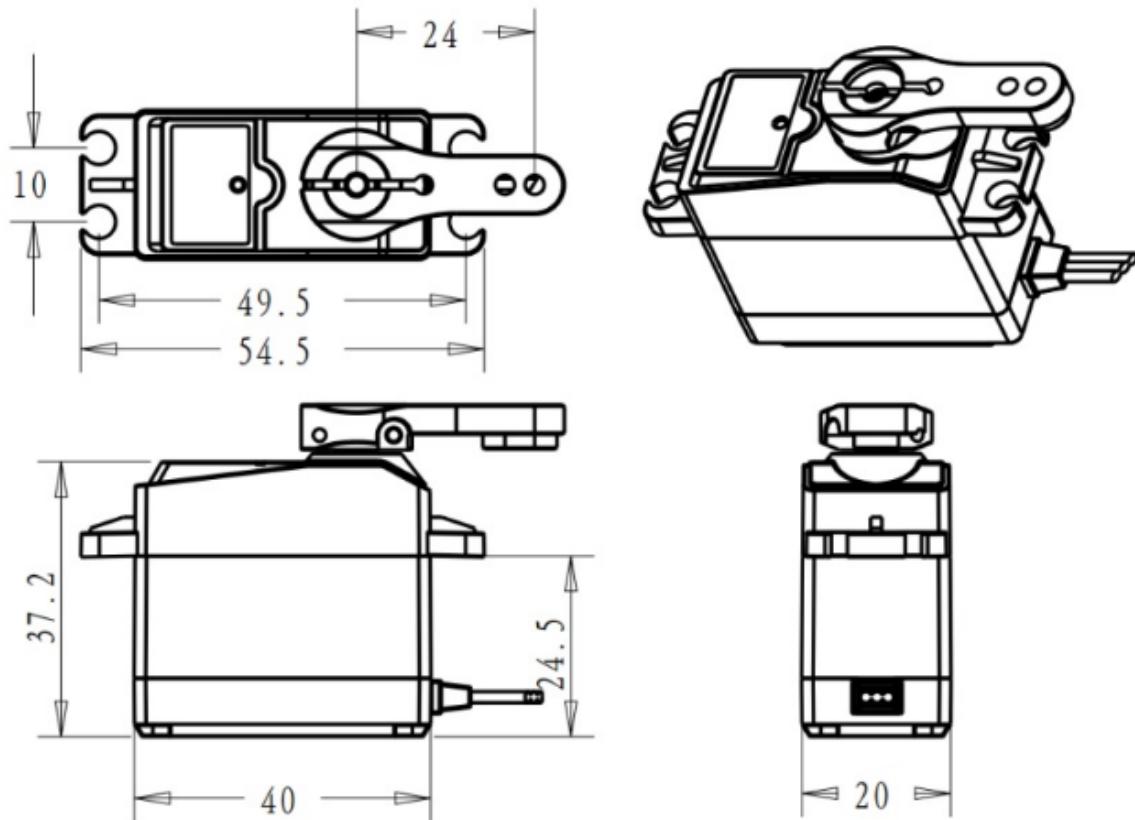
Es recomendable fijar una tensión de referencia al pin ADD, ya que el mismo configura la dirección del dispositivo en el protocolo I2C. Si se deja sin conexión se corre el riesgo que cualquier ruido externo modifique el valor de la dirección y se pierda la comunicación en medio de la operación.

A continuación se muestra un esquema de conexión para el sensor:



Donde P1 corresponde al puerto de expansión 1 de la placa EDU-CIAA.

### 16.2.5 B5: Servomotores DSSERVO DS3225



#### Specifications

Certificate	CE
Retail Packaging	Colored Box
User Manual (Language)	No User Manual
Gross Weight	0.082kg
Volume Weight	0.034kg
Length	10.000cm
Width	7.500cm
Height	2.000cm
Weight	0.073kg
With Retail Packaging	Yes

## Product Description

DSSERVO DS3225 25kg Metal Gear Waterproof Large Torque Digital Steering Servo for RC Cars

- Large torque, maximum torque can up to 25kg.cm (6.8V)
- High-precision metal gear with hard anodizing for better accuracy and longer service life
- CNC aluminium middle shell enhances heat dissipation and ensures the servo motor can work well
- Water proof rubber seals screws, made of first-grade material ensure the longevity of the products
- Good performance digital standard servo, which can be a common use for robots, mechanical arms, climbing cars and remote control toys
- Great choice for university professors and students who are interested in developing software for robot parts
- Tailored for RC hobbyist to provide different experience and delight

### Specification:

- Brand: DSSERVO
- Item: DSSERVO DS3225 Digital Servo
- Operating voltage: 4.8 ~ 6.8V
- Operating speed (5.0V): 0.15 sec/60 degree
- Operating speed (6.8V): 0.13 sec/60 degree
- Stall torque (5.0V): 21kg/cm
- Stall torque (6.8V): 25kg/cm
- Dead band: 3μs
- Working frequency: 50-333Hz
- Connector wire: About 300mm
- Gear type: Metal
- Motor Type: DC motor
- Size: 40x20x40.5mm



## 16.3 ANEXO C: CALCULOS

### 16.3.1 C1: Fusibles para el Sistema Mínimo

- Fuente del Microcontrolador y periféricos: Esta fuente será una alimentación fija de 5V ya que todos los elementos dependientes de ella utilizan este estándar de trabajo. Para los sensores se considera una corriente nominal relativamente baja ya que rara vez sobrepasan los 100mA para el giroscopio y el panel resistivo y los picos al arrancar el sistema son despreciables. Para el programador se requiere una corriente de 350 mA [19]. Finalmente, para el programador se considera una corriente total de 1A [41].
- Fuente de los servomotores: Para este periférico, su única función es la de alimentar los servomotores a un voltaje de operación de 5V a 8V. Según el fabricante, la corriente de operación de este componente va de los 350mA a 500mA sin carga, sin embargo, a diferencia de los sensores, estos componentes si suelen tener un pico de voltaje al momento de ser puestos en operación y aumenta conforme a la carga aplicada.

Para la estimación de la corriente en punto de operación a plena carga de los servomotores se realiza tomando de referencia algunos de los valores proporcionados por el fabricante, tales como el torque y la velocidad angular, así como el método empleado en [36] mediante el cual obtienen la corriente a partir de la potencia generada a la salida de dicho actuador. Considerando que el servomotor está diseñado para soportar una carga total aproximada de 25kg.cm y teniendo una distancia entre centros de 2.5cm aproximadamente, se considera que es capaz de soportar un total de 10kg.cm aproximadamente 0.9806 N.m en el extremo del cuerno donde se unirá con la junta.

Se contempla la velocidad de 0.13seg/60° a 6.8V equivalentes a 0.00216°/seg y que al hacer la conversión corresponde a 77.1604 rpm. Una vez se tiene estos valores en sistema internacional compatible con valores de voltaje y corriente, se utiliza la ecuación (84) para determinar la potencia.

$$P = T \cdot \omega \quad (84)$$

A partir de una potencia  $P = 7.92W$  se determina la corriente con la ecuación (85).

$$V * I = P \rightarrow I = \frac{P}{V} = \frac{7.97W}{6.8V} = 1.172A \quad (85)$$

Una vez que se evalúa cada una de las corrientes del circuito del controlador, se hace la sumatoria de estas respectivamente a su fuente de alimentación para determinar el valor del fusible de protección colocado justo después de los bornes de conexión. Para la *Fuente de alimentación del Micro* se obtiene una corriente total de 1.450 A al cual se le aplica un factor de seguridad de 15% obteniendo un valor total de 1.667 A que approximándose a valores estándar, se colocará un fusible de 2 A y se diseñará la pista en 2mm según [42].

De manera semejante se realiza el proceso de sumatoria de corrientes para la *Fuente de los Servos* obteniendo un total de 7.032 A, al aplicarle un factor de seguridad de 15% se obtiene una corriente total de 8.08 A. Adecuándose a fusibles de valores estándar se colocará uno de 10 A adaptando la pista al grosor de 4 mm para soportar dicha corriente. A partir de estas consideraciones se genera el circuito de la PCB que se muestra en la Figura 47.

#### 16.3.2 C2: Función de Transferencia de Servomotores

La función de transferencia se modela mediante la gráfica del voltaje con respecto al tiempo. Dicho de voltaje depende del giro del servomotor que acciona un potenciómetro acoplado.



Figura 82 Potenciómetro acoplado a servomotor por pieza de impresión 3D.

##### 16.3.2.1 Medición de Tiempo de Respuesta

Con ayuda de un osciloscopio se mide el comportamiento del giro del servomotor respecto al tiempo del servomotor 1. Los tiempos medidos son  $t_{m0}$ ,  $t_1$ ,  $t_2$  y  $t$  para las aproximaciones de Ogata, SK y la velocidad final.

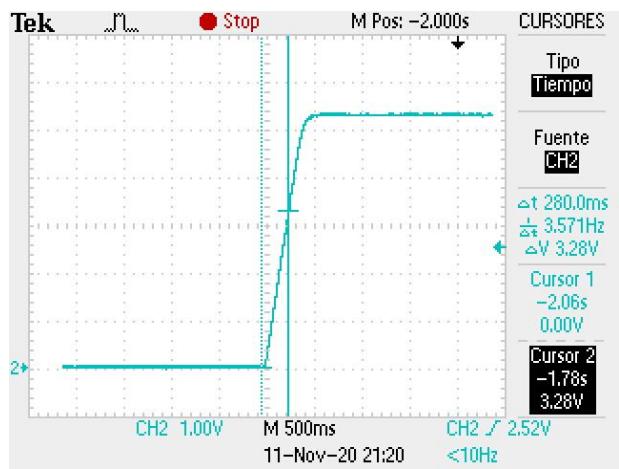


Figura 83 Medición  $\tau_{m0}$ .

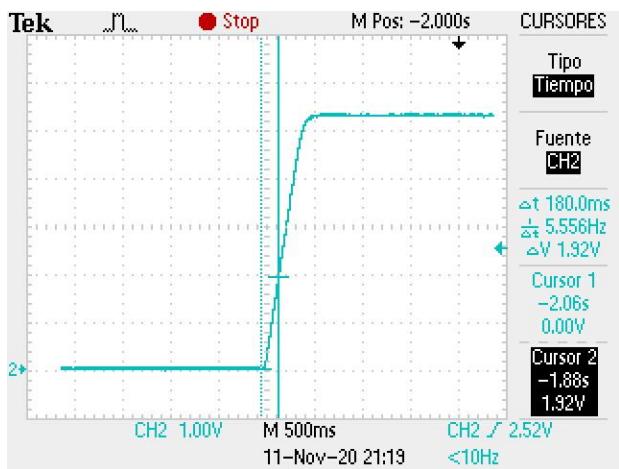


Figura 84 Medición de tiempo  $t_1$ .

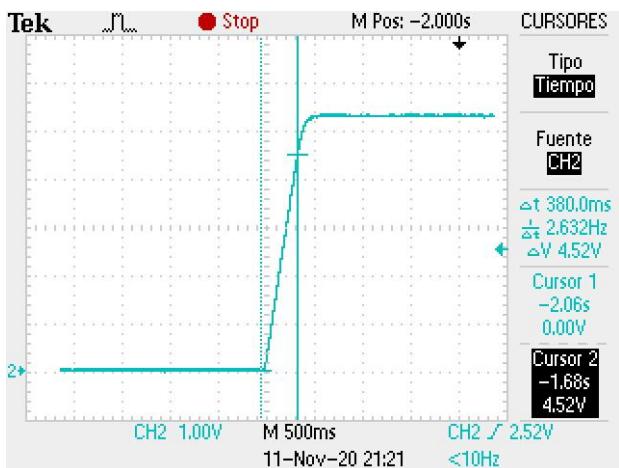


Figura 85 Medición de tiempo  $t_2$ .

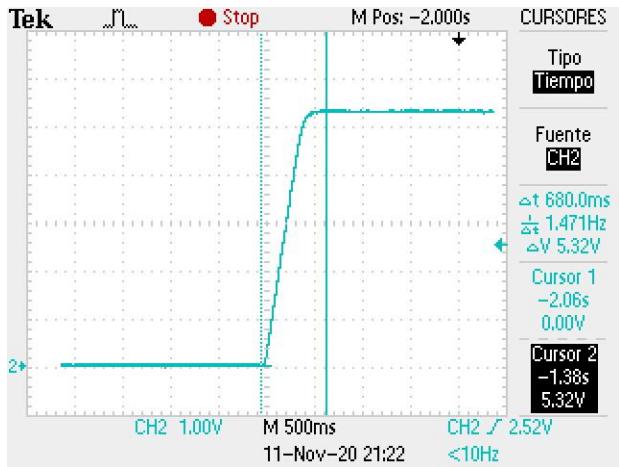


Figura 86 Medición de tiempo  $t$ .

La ecuación (7) requiere los tiempos  $t_2$  y  $t_1$ . Por lo tanto el tiempo de respuesta es:

$$\tau_{mSK} = 0.287021 \text{ seg}$$

Para el tiempo de la aproximación de Ogata (ecuación (8)):

$$\tau_{mo} = 0.31147 \text{ seg}$$

Y, por hoja de datos el tiempo de respuesta es:

$$\tau_{mo} = 0.39 \text{ seg}$$

#### 16.3.2.2 Ganancias Estáticas

Con fundamento en los datos del fabricante se sabe que el voltaje de entrada al servomotor es de 6.6V y además se deduce que  $\omega_{final} = 461.5389 \frac{\circ}{seg}$ , por lo tanto, la ganancia por hoja de datos queda como:

$$K_{mDS} = 179.3078716 \frac{\circ}{Vs^2}$$

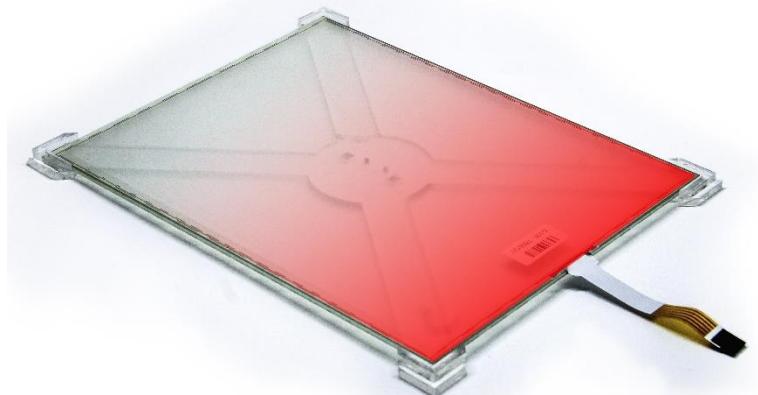
De manera semejante aplica el mismo cálculo para las ganancias de los servomotores.

$$K_{mDS} = 163.8278295 \frac{\circ}{Vs^2}$$

#### 16.4 ANEXO D: FUNCIONAMIENTO DEL PANEL RESISTIVO 5 HILOS

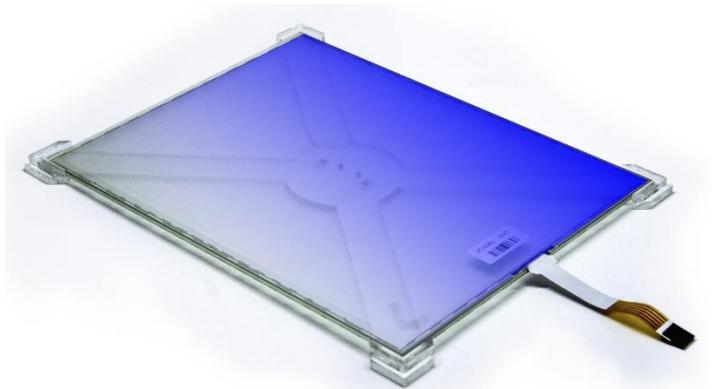
Para sensar el valor de la posición en X y Y, el microcontrolador deberá mandar los valores lógicos a los respectivos pines para determinar el estado en alto o bajo de las esquinas y generar un gradiente de voltaje. En la se observa el gradiente generado por el microcontrolador, este se representa como 0V en la zona más transparente indicando que esas dos esquinas están en valor bajo, y 5V en la zona con mayor intensidad de color indicando que está en valor alto.

En la Figura 87 se puede observar las esquinas con su respectivo pin representadas con color diferente para indicar las posiciones de las terminales que generan el gradiente de resistencia. Como se menciona anteriormente el panel debe pasar por tres estados básicos: el primer estado consta en poner en bajo las esquinas de la izquierda tanto inferior como superior, (o como a este caso se le denominará *II* y *SI*) y estado alto a las esquinas inferior derecha (*ID*) y superior derecha (*ID*). Luego de preparar la pantalla para sensar se obtiene el valor mediante el pin central que se conecta directamente al puerto ADC 0.



*Figura 87 - Gradiente de voltaje generado para posición X*

El siguiente estado se utiliza para sensar el valor de posición en Y, el microcontrolador al igual que el estado anterior deberá mandar los valores lógicos a los respectivos pines para preparar la pantalla de tal modo que genere un gradiente de *II* e *ID* en estado bajo y *SI* y *SD* en estado alto. Luego de preparar la pantalla se obtiene el valor mediante el mismo pin al puerto A0.



*Figura 88 – Gradiente de voltaje generado para posición Y*

El ultimo estado consiste en apagar todo el panel resistivo o mandar cada pin a nivel bajo, para eso ya no será necesario volver a sensar mediante el puerto A0; solamente se reinicia el ciclo con el primer estado para volver a sensar el valor en la posición X.

Se observará al momento de implementar el sensor que se tienden a traslapar las señales por lo cual será necesario implementar un método que permita separar cada señal, para esto se le da un tiempo de asentamiento entre las medidas de X y Y para registrar el valor y que no se interfieran una señal a la otra.

## 16.5 ANEXO E: PLANOS DE MANUFACTURA Y COMPONENTES

### 16.5.1 E1: Medidas estándar de rotulas SI-TK

Housing:

Free-cutting or forged carbon steel, Zinc plated

Insert: free-cutting bronze or steel with PTFE

Inner ring:

Bearing steel, hardened, ground, polished or chrome plated

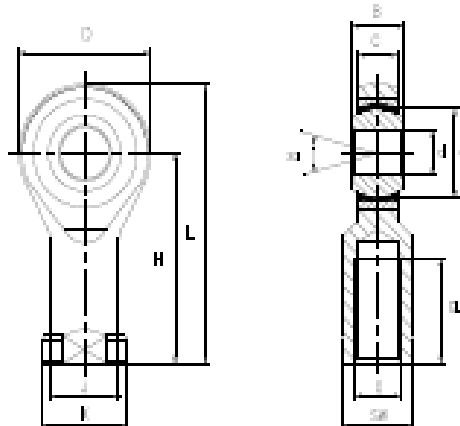
Sliding contact surface:

Bronze or steel/PTFE, maintenance free

杆体:碳钢, 半制作或精锻件, 镀白锌

衬套: 铜或钢制, PTFE自润滑

内圈: 滑承钢, 淬火, 磨加工, 抛光或镀铬

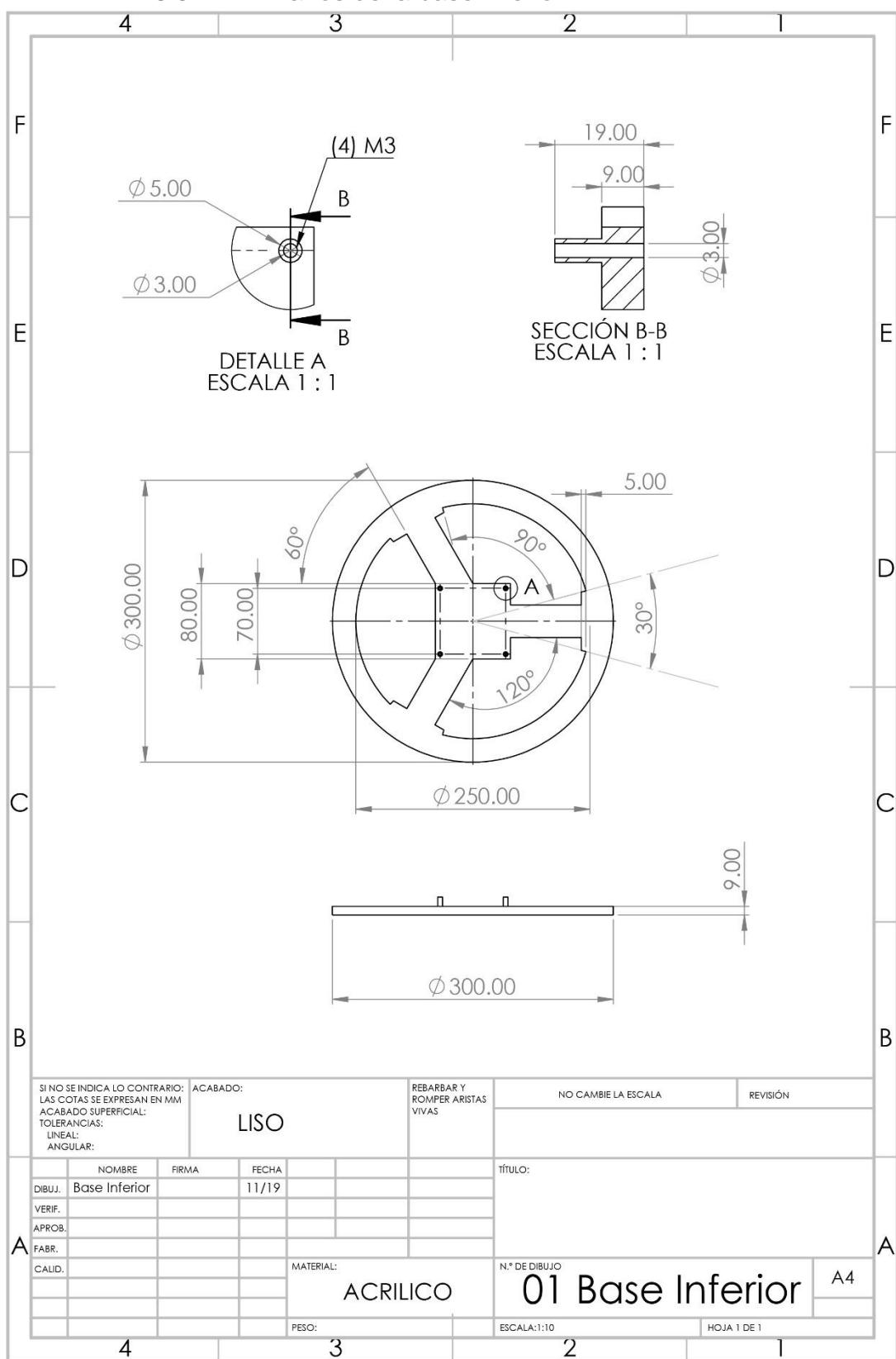


Bearing No. 型号	Dimensions millimeters 尺寸 mm													Load ratings 负荷		Weight 重量
	d	B	C	D	H	L	K	J	dk	SW	G	GL	a	dyn.	stat.	~kg
SI 4 TK	4	7	5.25	16	24	32	9.5	7.8	9.53	8	M4X0.7	10	13	2.3	4	0.011
SI 5 TK	5	8	6	18	27	36	11	8.5	11.1	9	M5X0.8	10	13	3.3	6	0.016
SI 6 TK	6	9	6.75	20	30	40	13	10	12.7	11	M6X1	12	13	4.3	7.5	0.025
SI 8 TK	8	12	9	24	36	48	16	12.5	15.9	14	M8X1.25	16	14	7.1	13	0.046
SI 10 TK	10	14	10.5	28	43	57	19	15	19	17	M10X1.5	20	13	10	18	0.076
SI 10 TKA	10	14	10.5	28	43	57	19	15	19	17	M10X1.25	20	13	10	18	0.076
SI 12 TK	12	16	12	32	50	66	22	17.5	22.2	19	M12X1.75	22	13	13.5	23.5	0.1
SI 12 TKA	12	16	12	32	50	66	22	17.5	22.2	19	M12X1.25	22	13	13.5	23.5	0.1
SI 14 TK	14	19	13.5	36	57	75	25	20	25.4	22	M14X2	25	16	17	24	0.16
SI 14 TKA	14	19	13.5	36	57	75	25	20	25.4	22	M14X1.5	25	16	17	24	0.16
SI 16 TK	16	21	15	42	64	85	27	22	28.5	22	M16X2	28	15	21.5	31	0.22
SI 16 TKA	16	21	15	42	64	85	27	22	28.5	22	M16X1.4	28	15	21.5	31	0.22
SI 18 TK	18	23	16.5	44	71	93	31	25	31.7	27	M18X1.5	32	15	26	38.5	0.32
SI 20 TK	20	25	18	50	77	102	34	27.5	34.9	30	M20X1.5	33	14	31	42.5	0.42
SI 22 TK	22	26	20	54	84	111	38	30	38.1	32	M22X1.5	37	15	42	53	0.54
SI 25 TK	25	31	22	60	94	124	42	33.5	42.9	36	M24X2	42	15	53	68	0.75
SI 28 TK	28	35	25	66	103	136	45	37	47.6	41	M27X2	48	15	59	86	0.98
SI 30 TK	30	37	25	70	110	145	50	40	50.8	41	M30X2	51	17	70	88	1.13
SI 30 TKA	30	37	25	70	110	145	50	40	50.8	41	M27X2	51	17	70	88	1.13
SI 35 TK	35	43	28	81	125	165.5	58	46	57.2	50	M36X2	56	16	80	101	1.6
SI 40 TK	40	49	35	91	142	187.5	69	57	66	60	M42X2	60	17	116	124	2.77
SI 50 TK	50	60	45	117	160	218.5	78	65	82	65	M48X2	65	12	185	308	5

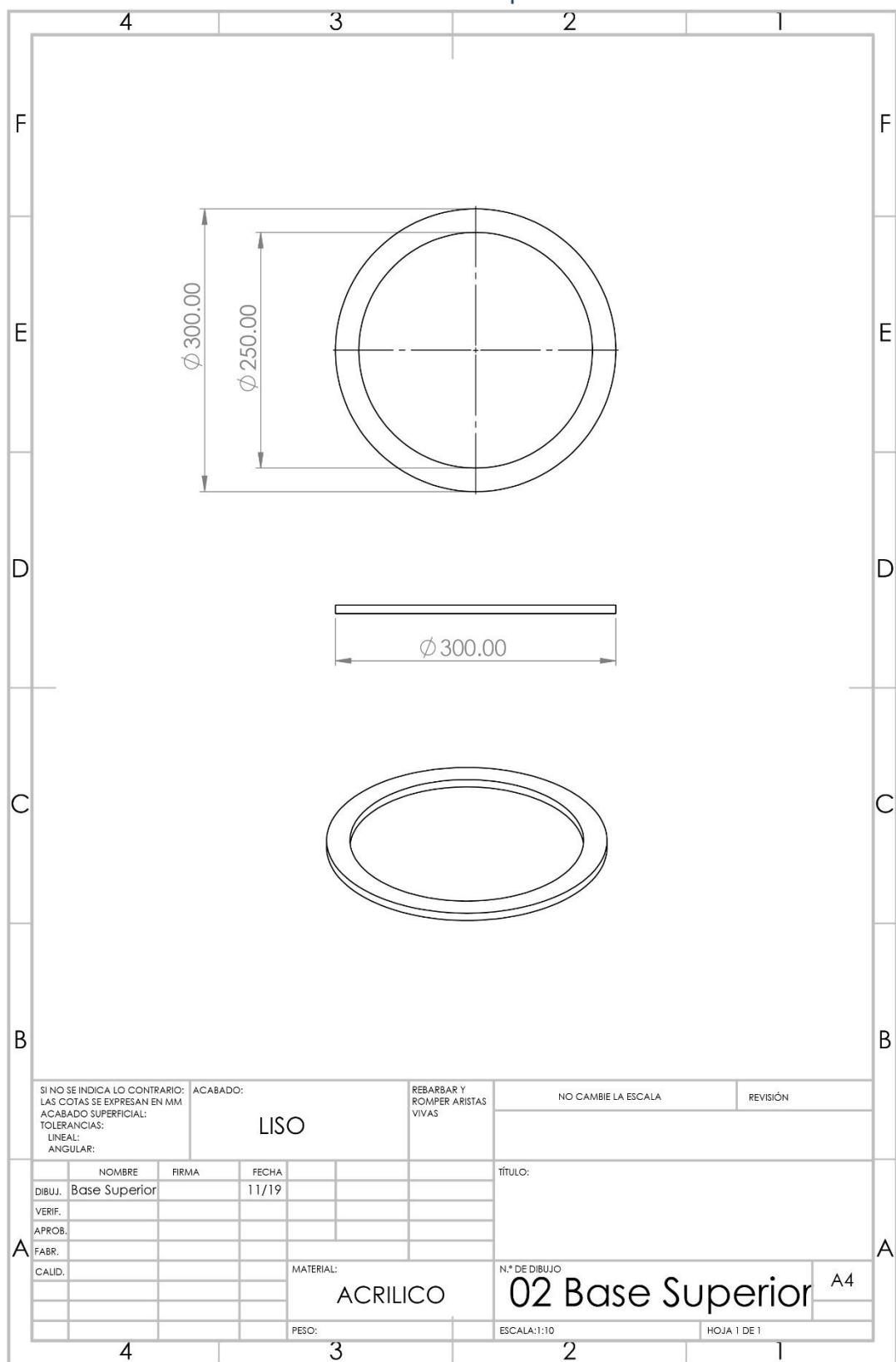
Female thread: for a left hand thread, add L (example: S1L..D)

内螺纹: 对左螺纹, 加 L (例如: S1L..D)

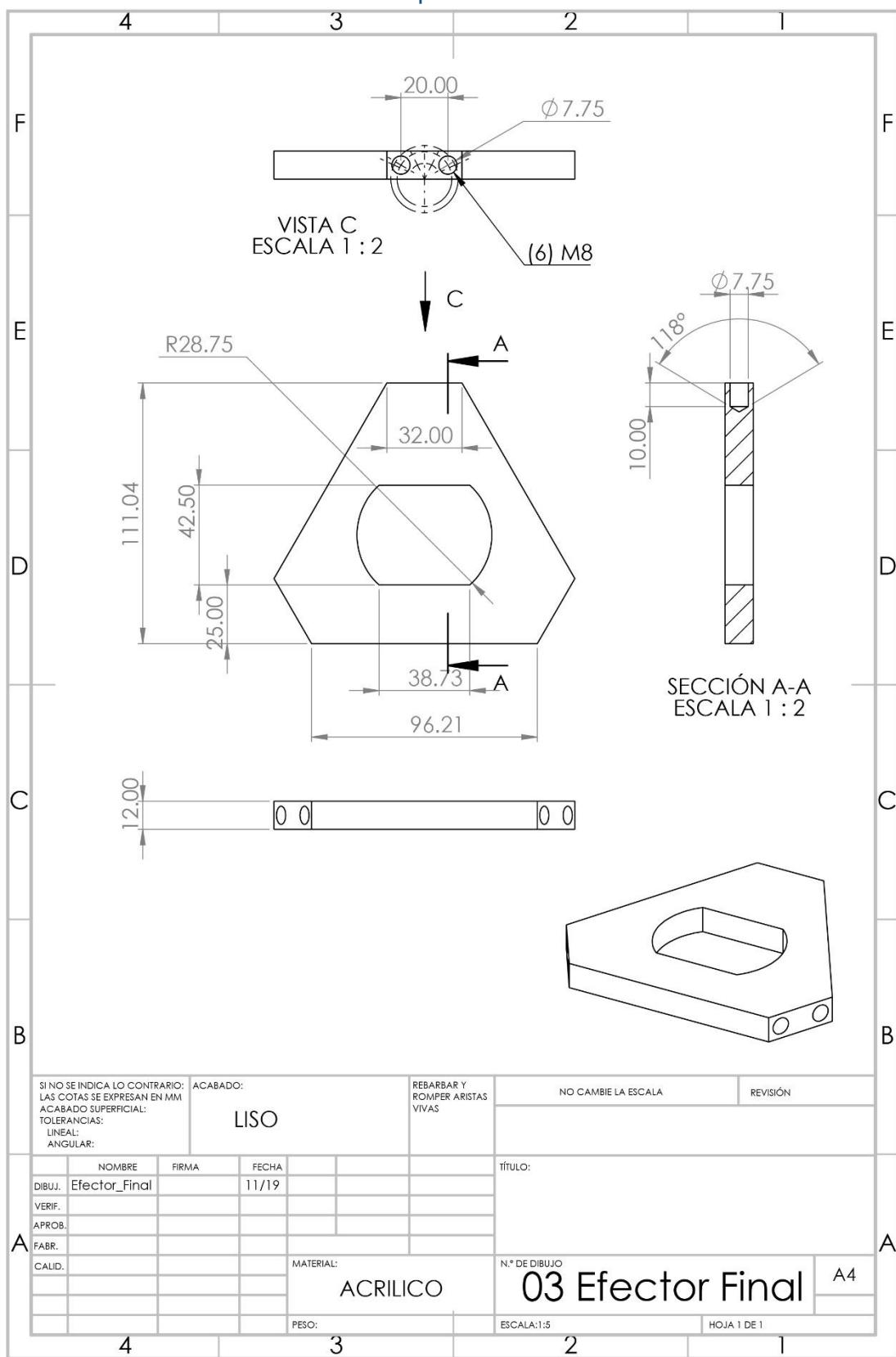
### 16.5.2 E2: Planos de la base inferior



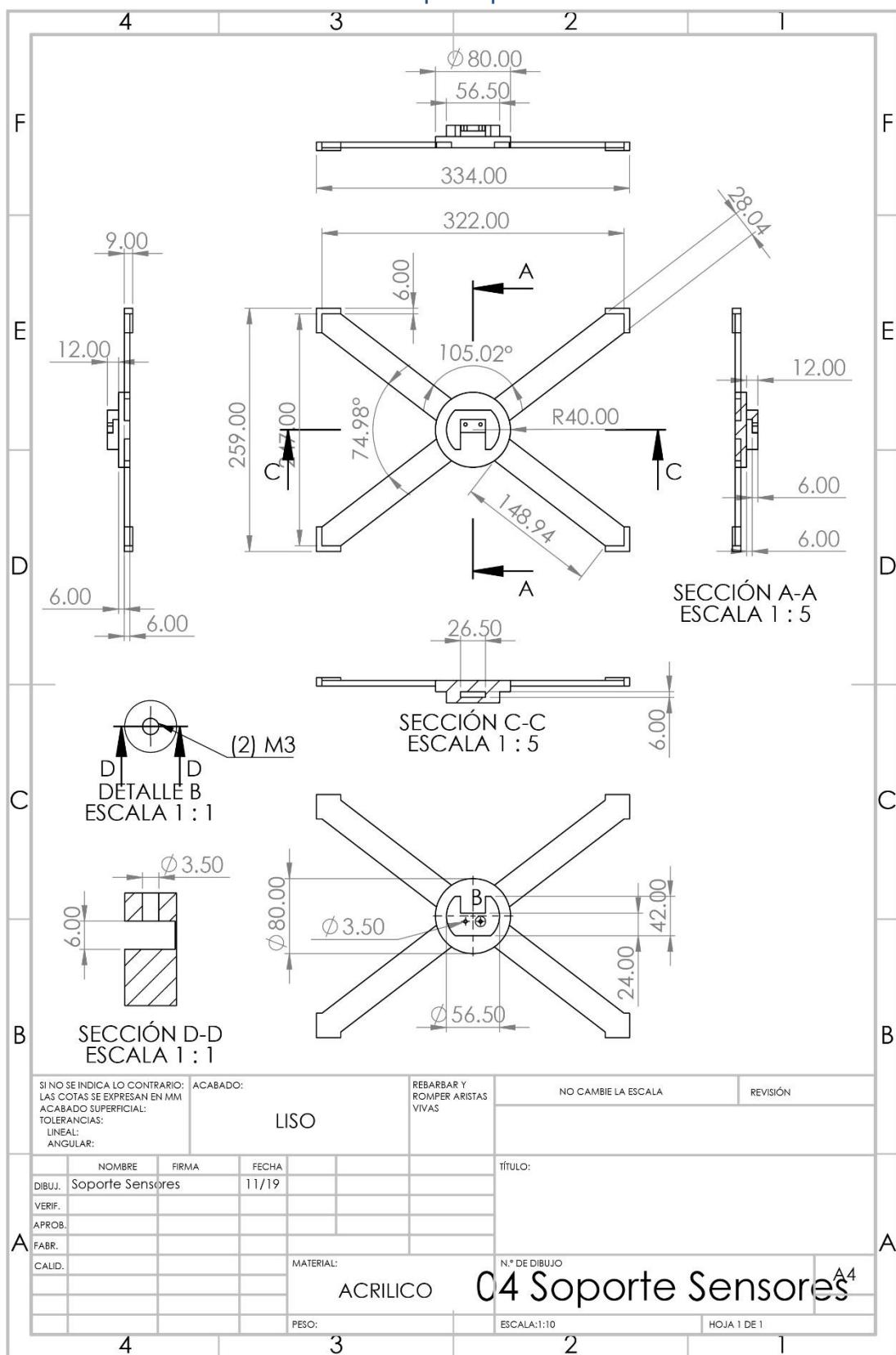
16.5.3 E3: Planos de la base superior



16.5.4 E4: Planos de la plataforma del efecto final.



16.5.5 E5: Planos del soporte para los sensores.



## 16.6 ANEXO F: TABLAS DE TRANSFORMADA Z

	$f[n]$	$F(z)$	$ROC$
	$f[n] = \frac{1}{2\pi i} \oint_{\gamma} F(z) z^{n-1} dz$	$F(z) = \sum_{n=-\infty}^{+\infty} f[n] z^{-n}$	$A(r_1, r_2)$
1	$u[n]$	$\frac{1}{1 - (1/z)} = \frac{z}{z-1}$	$ z  > 1$
2	$\delta[n]$	1	$z \in C$
3	$k \in Z$	$\delta[n-k]$	$k > 0 \quad  z  > 0$ $k < 0 \quad z \in C$
4	$k \in Z$ $k > 0$ $pulso[n,k] := u[n+k] - u[n-k]$	$\frac{1}{1 - (1/z)} (z^k - z^{-k})$	$ z  > 1$
5	$a^n u[n]$	$\frac{1}{1 - (a/z)} = \frac{z}{z-a}$	$ z  >  a $
6	$\sin[\alpha n] u[n]$	$\frac{z \sin \alpha}{(z^2 - 2z \cos \alpha + 1)}$	$ z  > 1$
7	$\cos[\alpha n] u[n]$	$\frac{z (z - \cos \alpha)}{(z^2 - 2z \cos \alpha + 1)}$	$ z  > 1$
8	$sh[\alpha n] u[n]$	$\frac{z \operatorname{sh} \alpha}{(z^2 - 2z \operatorname{ch} \alpha + 1)}$	$ z  > e^{ \alpha }$
9	$ch[\alpha n] u[n]$	$\frac{z (z - ch \alpha)}{(z^2 - 2z ch \alpha + 1)}$	$ z  > e^{ \alpha }$
10	$n = C_{n,1}$	$\frac{z}{(z-1)^2}$	$ z  > 1$
11	$\frac{1}{2!} n(n-1) = C_{n,2}$	$\frac{z}{(z-1)^3}$	$ z  > 1$
12	$\frac{1}{2!} n(n-1)\dots(n-k) = C_{n,k}$	$\frac{z}{(z-1)^{k+1}}$	$ z  > 1$

13	$n^2$	$\frac{z(z+1)}{(z-1)^3}$	$ z  > 1$
14 $a \neq 0$ $b \neq 0$ $a \neq b$	$\frac{1}{(a-b)} (a^{n+1} - b^{n+1}) u[n+1]$	$\frac{1}{1-\frac{a}{z}} \frac{1}{1-\frac{b}{z}}$	$ z  > \max( a ,  b )$
15 $a \neq 0$	$(n+1) a^n u[n+1]$	$\frac{1}{(1-\frac{a}{z})^2}$	$ z  >  a $
16	$C_{n-1,n-k} a^{n-k} u[n-1]$	$\frac{z}{(z-1)^{k+1}}$	$ z  >  a $
17	$f[n] = f[n+T]$ $f[n] = \begin{cases} 1 & n = 0 + pT \quad p \in N \\ 0 & n \neq 0 + pT \end{cases}$	$\frac{1}{1-z^{-T}}$	$ z  > 1$
18	$\frac{1}{n} u[n-1]$	$L(1 - \frac{1}{z})$	$ z  > 1$
19	$\frac{1}{n-1} u[n-2] - \frac{1}{n} u[n-1]$	$-(1 - \frac{1}{z}) L(1 - \frac{1}{z})$	$ z  > 1$

## 16.7 ANEXO G: CODIGOS DE SIMULACIÓN

### 16.7.1 G1: ADC a 18kHz

```
// Iniciar variables.
boolean toggle = false;
boolean toggleADC = false;
int pulso_timer = 13;
int pulso_ADC = 12;

void setup() {
cli(); // 1. Suspender interrupciones.
// 2. Programar temporizador 0 para producir interrupciones a la tasa de muestreo de 4 kHz.
TCCR0A &= 0x00;
TCCR0B &= 0x00;
TCCR0A |= (0 << WGM00) | (1 << WGM01);
TCCR0B |= (0 << WGM02);
TCNT0 &= 0x00;
TCCR0B |= (0 << CS00) | (1 << CS01) | (0 << CS02);
OCR0A |= 27;
TIMSK0 |= (1 << OCIE0A);
// 3. Programar conversor A/D para convertir a la máxima velocidad de reloj
CLKPR = 0x80;
CLKPR = 0x00;
ADCSRA |= (1 << ADPS0) | (1 << ADPS1) | (0 << ADPS2);
ADMUX |= (1 << REFS0) | (0 << REFS1);
ADMUX |= (0 << MUX0) | (0 << MUX1) | (0 << MUX2) | (0 << MUX3);
ADMUX |= (0 << ADLAR);
ADCSRB |= (1 << ADTS0) | (1 << ADTS1) | (0 << ADTS2);
ADCSRA |= (1 << ADEN) | (1 << ADSC) | (1 << ADATE);
```

```

ADCSRA |= (1 << ADIE);
sei();
ADCSRA |= (1 << ADSC);
pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
}
void loop() {
    TIFR0 |= (1 << OCF0A); // Genera las interrupciones del ADC a partir del Timer 0
    cambiarEstadoADC();} // Ejecuta la función testigo del ADC

ISR (ADC_vect){ } //Funcion para muestrear señales analogicas
ISR(TIMER0_COMPA_vect) {
    cambiarEstadoTimer(); } // Ejecuta la función testigo del TIMMER

void cambiarEstadoTimer(void){
toggle = !toggle; // Cambia de estado la variable toggle cada vez que se produce una
interrupción en el TIMER
digitalWrite(13, toggle); }

// Genera un pulso cambiando el valor de salida en el pin PB5
void cambiarEstadoADC(void){
toggleADC = !toggleADC; // Cambia de estado la variable toggle cada vez que se produce una
interrupción en ADC
digitalWrite(12, toggleADC); } // Genera un pulso cambiando el valor de salida en el pin PB4

```

A continuación, se presenta el diagrama eléctrico para la simulación del ADC, para esto se empleó un generador con una frecuencia sinodal de 18kHz, el microcontrolador ATmega328P con el programa cargado y finalmente los pines PB4 y PB5 como salidas al osciloscopio para compararlas con el generador.

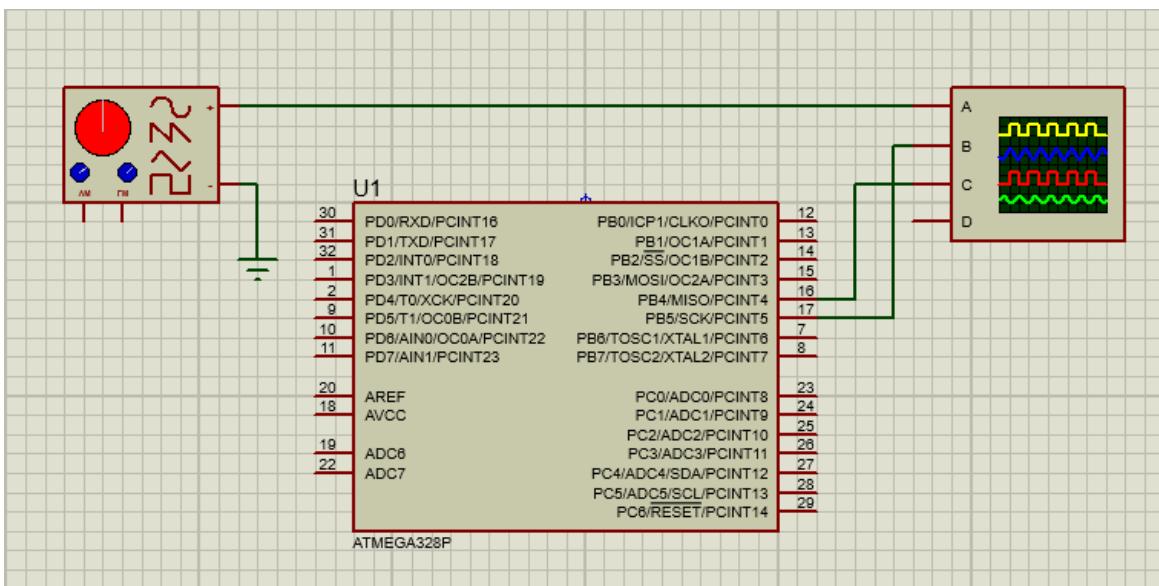


Figura 89- Diagrama eléctrico para simulación del ADC.

### 16.7.2 G2: Cinemática Inversa (Simulación de Articulaciones)

```
function varargout = stewart_platform(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',     mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @stewart_platform_OpeningFcn, ...
    'gui_OutputFcn',  @stewart_platform_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function stewart_platform_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.output = hObject;
    guidata(hObject, handles);
    handles.trans=[0 0 0];
    handles.orient=[0 0 0 ];
    handles.ardu=0;
    handles.arduino_status=0;
    handles.servos=0;
    handles.toggle=0;
    do_the_stewart(handles);
    guidata(hObject, handles);

function varargout = stewart_platform_OutputFcn(hObject, eventdata, handles)
    varargout{1} = handles.output;

function exit_pushbutton_Callback(hObject, eventdata, handles)
    close all;

function x_slider_Callback(hObject, eventdata, handles)
    max_x= 4;
    min_x= -4;
    slide= get(hObject,'Value');
    handles.x= slide * (max_x - min_x) + min_x;
    x_pos= num2str(handles.x);
    handles.trans(1)= handles.x;
    set(handles.show_x,'String',x_pos);
    guidata(hObject, handles);
    do_the_stewart(handles);

function x_slider_CreateFcn(hObject, eventdata, handles)
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
    max_x= 4;
    min_x= -4;
    slide= get(hObject,'Value');
```

```

handles.x= slide * (max_x - min_x) + min_x;
handles.trans(1)= handles.x;
guidata(hObject, handles)

function y_slider_Callback(hObject, eventdata, handles)
    max_y= 4;
    min_y= -4;
    slide= get(hObject,'Value');
    handles.y= slide * (max_y - min_y) + min_y;
    y_pos= num2str(handles.y);
    handles.trans(2)= handles.y;
    set(handles.show_y,'String',y_pos);
    guidata(hObject, handles);
    do_the_stewart(handles);

function y_slider_CreateFcn(hObject, eventdata, handles)
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
    max_y= 4;
    min_y= -4;
    slide= get(hObject,'Value');
    handles.y= slide * (max_y - min_y) + min_y;
    handles.trans(2)= handles.y;
    guidata(hObject, handles);

function z_slider_Callback(hObject, eventdata, handles)
    max_z= 4;
    min_z= -4;
    slide= get(hObject,'Value');
    handles.z= slide * (max_z - min_z) + min_z;
    z_pos= num2str(handles.z);
    handles.trans(3)= handles.z;
    set(handles.show_z,'String',z_pos);
    guidata(hObject, handles);
    do_the_stewart(handles);

function z_slider_CreateFcn(hObject, eventdata, handles)
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
    max_z= 4;
    min_z= -4;
    slide= get(hObject,'Value');
    handles.z= slide * (max_z - min_z) + min_z;
    handles.trans(3)= handles.z;
    guidata(hObject, handles);

function roll_slider_Callback(hObject, eventdata, handles)
    max_roll= 15*pi/180;
    min_roll= -15*pi/180;
    slide= get(hObject,'Value');
    handles.roll= slide * (max_roll - min_roll) + min_roll;
    roll_pos= num2str(handles.roll*180/pi);
    handles.orient(1)= handles.roll;

```

```

set(handles.show_roll,'String',roll_pos);
guidata(hObject, handles);
do_the_stewart(handles);

function roll_slider_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
max_roll= 15*pi/180;
min_roll= -15*pi/180;
slide= get(hObject,'Value');
handles.roll= slide * (max_roll - min_roll) + min_roll;
handles.orient(1)= handles.roll;
guidata(hObject, handles);

function pitch_slider_Callback(hObject, eventdata, handles)
max_pitch= 15*pi/180;
min_pitch= -15*pi/180;
slide= get(hObject,'Value');
handles.pitch= slide * (max_pitch - min_pitch) + min_pitch;
handles.orient(2)= handles.pitch;
pitch_pos= num2str(handles.pitch*180/pi);
set(handles.show_pitch,'String',pitch_pos);
guidata(hObject, handles);
do_the_stewart(handles);

function pitch_slider_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
max_pitch= 15*pi/180;
min_pitch= -15*pi/180;
slide= get(hObject,'Value');
handles.pitch= slide * (max_pitch - min_pitch) + min_pitch;
handles.orient(2)= handles.pitch;
guidata(hObject, handles);

function yaw_slider_Callback(hObject, eventdata, handles)
max_yaw= 15*pi/180;
min_yaw= -15*pi/180;
slide= get(hObject,'Value');
handles.yaw= slide * (max_yaw - min_yaw) + min_yaw;
yaw_pos= num2str(handles.yaw*180/pi);
handles.orient(3)= handles.yaw;

set(handles.show_yaw,'String',yaw_pos);
guidata(hObject, handles);
do_the_stewart(handles);

function yaw_slider_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
max_yaw= 15*pi/180;
min_yaw= -15*pi/180;

```

```

slide= get(hObject,'Value');
handles.yaw= slide * (max_yaw - min_yaw) + min_yaw;
handles.orient(3)= handles.yaw;
guidata(hObject, handles);

function show_handle_pushbutton_Callback(hObject, eventdata, handles)
    disp(handles);

function r_B_inputbox_Callback(hObject, eventdata, handles)
    handles.r_B= str2num(get(hObject,'String'));
    guidata(hObject,handles);
    do_the_stewart(handles);

function r_B_inputbox_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
    handles.r_B= str2num(get(hObject,'String'));
    guidata(hObject,handles);

function r_P_inputbox_Callback(hObject, eventdata, handles)
    handles.r_P= str2num(get(hObject,'String'));
    guidata(hObject,handles);
    do_the_stewart(handles);

function r_P_inputbox_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
    handles.r_P= str2num(get(hObject,'String'));
    guidata(hObject,handles);

function servo_arm_length_inputbox_Callback(hObject, eventdata, handles)
    handles.servo_arm_length= str2num(get(hObject,'String'));
    guidata(hObject,handles);
    do_the_stewart(handles);

function servo_arm_length_inputbox_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
    handles.servo_arm_length= str2num(get(hObject,'String'));
    guidata(hObject,handles);

function rod_length_inputbox_Callback(hObject, eventdata, handles)
    handles.rod_length= str2num(get(hObject,'String'));
    guidata(hObject,handles);
    do_the_stewart(handles);

function rod_length_inputbox_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
handles.rod_length= str2num(get(hObject,'String'));
guidata(hObject,handles);

function alpha_B_inputbox_Callback(hObject, eventdata, handles)
handles.alpha_B= str2num(get(hObject,'String'));
guidata(hObject,handles);

do_the_stewart(handles);

function alpha_B_inputbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
handles.alpha_B= str2num(get(hObject,'String'));
guidata(hObject,handles);

function alpha_P_inputbox_Callback(hObject, eventdata, handles)
handles.alpha_P= str2num(get(hObject,'String'));
guidata(hObject,handles);
do_the_stewart(handles);

function alpha_P_inputbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
handles.alpha_P= str2num(get(hObject,'String'));
guidata(hObject,handles);

function roll_pitch_sweep_pushbutton_Callback(hObject, eventdata, handles)
set(handles.roll_pitch_sweep_pushbutton,'BackgroundColor','green');
for i=0:0.1:2*pi
    handles.orient=[10*pi/180*cos(i) 10*pi/180*sin(i) 0];
    do_the_stewart(handles);
    drawnow
end
for i=2*pi:-0.1:1
    handles.orient=[10*pi/180*cos(i) 10*pi/180*sin(i) 0];
    do_the_stewart(handles);
    drawnow
end
set(handles.roll_pitch_sweep_pushbutton,'BackgroundColor','default');
handles.orient=[0 0 0];
guidata(hObject,handles);

function trans_sweep_pushbutton_Callback(hObject, eventdata, handles)
set(handles.trans_sweep_pushbutton,'BackgroundColor','green');
for i=0:0.1:2*pi
    handles.trans=[4*sin(i) 0 0];
    do_the_stewart(handles);

```

```

drawnow
end
for i=0:0.1:2*pi
    handles.trans=[0 4*sin(i) 0];
    do_the_stewart(handles);
    drawnow
end
for i=0:0.1:2*pi
    handles.trans=[0 0 2*sin(i)];
    do_the_stewart(handles);
    drawnow
end
set(handles.trans_sweep_pushbutton,'BackgroundColor','default');
handles.orient=[0 0 0];
guidata(hObject,handles);

function [ angles ] = do_the_stewart(handles)
tic;
imag_count=0;
angles= calculate_stewart_platform(handles.r_B, ...
    handles.r_P, ...
    handles.servo_arm_length, ...
    handles.rod_length, ...
    handles.alpha_B*pi/180, ...
    handles.alpha_P*pi/180, ...
    handles.trans, ...
    handles.orient);
if(imag(angles(1))~=0)
    set(handles.show_angle1,'Background','red');
else
    set(handles.show_angle1,'Background','default');
    imag_count=imag_count+1;
end
if(imag(angles(2))~=0)
    set(handles.show_angle2,'Background','red');
else
    set(handles.show_angle2,'Background','default');
    imag_count=imag_count+1;
end
if(imag(angles(3))~=0)
    set(handles.show_angle3,'Background','red');
else
    set(handles.show_angle3,'Background','default');
    imag_count=imag_count+1;
end
if(imag(angles(4))~=0)
    set(handles.show_angle4,'Background','red');
else
    set(handles.show_angle4,'Background','default');
    imag_count=imag_count+1;
end
if(imag(angles(5))~=0)
    set(handles.show_angle5,'Background','red');
else
    set(handles.show_angle5,'Background','default');
    imag_count=imag_count+1;

```

```

end
if(imag(angles(6))~=0)
    set(handles.show_angle6,'Background','red');
else
    set(handles.show_angle6,'Background','default');
    imag_count=imag_count+1;
end
set(handles.show_angle1,'String', num2str(angles(1)*180/pi));
set(handles.show_angle2,'String', num2str(angles(2)*180/pi));
set(handles.show_angle3,'String', num2str(angles(3)*180/pi));
set(handles.show_angle4,'String', num2str(angles(4)*180/pi));
set(handles.show_angle5,'String', num2str(angles(5)*180/pi));
set(handles.show_angle6,'String', num2str(angles(6)*180/pi));
if(handlesarduino_status && imag_count==6)
    angles= angles*180/pi;
    servo_angles= ([90 90 90 90 90 90]+angles)/180;
    writePosition(handles.s1, 1-servo_angles(1));
    writePosition(handles.s2, servo_angles(2));
    writePosition(handles.s3, 1-servo_angles(3));
    writePosition(handles.s4, servo_angles(4));
    writePosition(handles.s5, 1-servo_angles(5));
    writePosition(handles.s6, servo_angles(6));
end
lag=toc;
disp('Calc duration: ')
disp(lag)
function Posx_Forward_Callback(hObject, eventdata, handles)

function Posx_Forward_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function Posy_Forward_Callback(hObject, eventdata, handles)

function Posy_Forward_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Posz_Forward_Callback(hObject, eventdata, handles)

function Posz_Forward_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Calculate_Position_Callback(hObject, eventdata, handles)
do_the_stewart(handles);
L=18;
bs=2.5;
p=4.53;
rb=25;
u1=15;
error=0;

```

```

q1=str2num(get(handles.show_angle1,'String'))
q2=str2num(get(handles.show_angle2,'String'))
q3=str2num(get(handles.show_angle3,'String'))
q4=str2num(get(handles.show_angle4,'String'))
q5=str2num(get(handles.show_angle5,'String'))
q6=str2num(get(handles.show_angle6,'String'))

th1=q1;
th2=q1-pi/2;
th3=q1+pi;
th4=q2;
th5=q2-pi/2;
th6=q2;
th7=q3;
th8=q3-pi/2;
th9=q3+pi;
th10=q4;
th11=q4-pi/2;
th12=q4;
th13=q5;
th14=q5-pi/2;
th15=q5+pi;
th16=q6;
th17=q6-pi/2;
th18=q6;

% PARÁMETROS DENAVIT HARTENBERGH
th=[th1,th2,th3,th4,th5,th6,th7,th8,th9,th10,th11,th12,th13,th14,th15,th16,th17,th18];
d=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
a=[bs,L,p,L,bs,2*rb*u1,bs,L,p,L,bs,2*rb*u1,bs,L,p,L,bs,2*rb*u1];
alpha=[pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9,pi/9];

for i=1:1:length(a)
A(:,:,i)=[ cos(th(i)),-sin(th(i))*cos(alpha(i)),sin(th(i))*sin(alpha(i)),
a(i)*cos(th(i));
sin(th(i)), cos(th(i))*cos(alpha(i)), -cos(th(i))*sin(alpha(i)), a(i)*sin(th(i));
0, sin(alpha(i)), cos(alpha(i)), d(i) ;
0, 0, 0, 1 ];
end

T01=A(:,:,1);
T02=A(:,:,1)*A(:,:,2);
T03=A(:,:,1)*A(:,:,2)*A(:,:,3);
T04=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4);
T05=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5);
T06=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6);
T07=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7);
T08=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8);
T09=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9);
T10=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10);
T11=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11);
T12=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12);

```

```

T13=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13);
T14=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13)*A(:,:,14);
T15=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13)*A(:,:,14)*A(:,:,15);
T16=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13)*A(:,:,14)*A(:,:,15)*A(:,:,16);
T17=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13)*A(:,:,14)*A(:,:,15)*A(:,:,16)*A(:,:,17);
T18=A(:,:,1)*A(:,:,2)*A(:,:,3)*A(:,:,4)*A(:,:,5)*A(:,:,6)*A(:,:,7)*A(:,:,8)*A(:,:,9)*A(:,:,10)*A(:,:,11)*A(:,:,12)*A(:,:,13)*A(:,:,14)*A(:,:,15)*A(:,:,16)*A(:,:,17)*A(:,:,18);

```

MatrizHomogeneaFinal=T18

```

Position=[T18(1,4); T18(2,4);T18(3,4)]
set(handles.Posx_Forward,'String',num2str(Position(1,1)));
set(handles.Posy_Forward,'String',num2str(Position(2,1)));
set(handles.Posz_Forward,'String',num2str(Position(3,1)));
guidata(hObject,hObject);

```

### 16.7.3 G3: Cinemática Inversa (Calculo Numérico)

```

clear;
clc;

%MEDIDAS [cm]
Rb=9.109; %Radio de la base %-----AVR define-----
Rp=8; %Radio de la plataforma (delimitada por
       %rótulas superiores) %-----AVR define-----
Cs=2.4; %Longitud de cuerno %-----AVR define-----
       %de servomotor
L=16.5; %Longitud de junta %-----AVR define-----
Cs_2=2*Cs;

%Posiciones angulares de puntos Bi,
%respecto a eje X de marco {B}
Mu1= 19.2*(pi/180);
Mu2= 100.8*(pi/180);
Mu3= 139.2*(pi/180);
Mu4= -139.2*(pi/180); %220.8°
Mu5= -100.8*(pi/180); %259.2°
Mu6= -19.2*(pi/180); %340.8°

H=16.2; %Altura plataforma respecto a eje Z de marco {B} 16.271cm en cad, 0.72mm de error
Alfa=0; %Roll De -15 a +15
Beta=0; %Pitch
Gamma=25;
z=0; %Altura deseada De - 2.6 a +1.99 respecto a sistema móvil de coordenadas {P} y con roll,
pitch and yaw = (0,0,0)

%%%
%
```

```

%----- POSICIÓN SISTEMAS {Bi} RESPECTO A {B} -----
%
%
%----- Rotación de -90° respecto
RotY_135=[ cos(-pi/2),  0,  sin(-pi/2),  0 ;
            0 ,  1,  0 ,  0 ;
           -sin(-pi/2),  0,  cos(-pi/2),  0 ;
            0 ,  0,  0 ,  1 ];

RotX_246=[1 ,  0 ,  0 ,  0;
           0 , cos(pi/2), -sin(pi/2), 0;
           0 , sin(pi/2), cos(pi/2), 0;
           0 ,  0 ,  0 ,  1];

%%%
%%%%%%%%%%%%%%% Servo 1
%%%%%%%%%%%%%%% Servo 1
RotZ_Servo1=[ cos(Mu1) , -sin(Mu1) ,  0,  0 ;
              sin(Mu1) , cos(Mu1) ,  0,  0 ;
              0 ,  0 ,  1,  0 ;
              0 ,  0 ,  0,  1 ];

Tras_Servo1=[ 1,  0,  0,  Rb*cos(Mu1);
              0,  1,  0,  Rb*sin(Mu1);
              0,  0,  1,  0 ;
              0,  0,  0,  1 ];

%%%
%%%%%%%%%%%%%%% Servo 2
%%%%%%%%%%%%%%% Servo 2
RotZ_Servo2=[ cos(Mu2-pi/2) , -sin(Mu2-pi/2) ,  0,  0 ;
              sin(Mu2-pi/2) , cos(Mu2-pi/2) ,  0,  0 ;
              0 ,  0 ,  1,  0 ;
              0 ,  0 ,  0,  1 ];

Tras_Servo2=[ 1,  0,  0,  Rb*cos(Mu2);
              0,  1,  0,  Rb*sin(Mu2);
              0,  0,  1,  0 ;
              0,  0,  0,  1 ];

%%%
%%%%%%%%%%%%%%% Servo 3
%%%%%%%%%%%%%%% Servo 3
RotZ_Servo3=[ cos(Mu3) , -sin(Mu3) ,  0,  0 ;
              sin(Mu3) , cos(Mu3) ,  0,  0 ;
              0 ,  0 ,  1,  0 ;
              0 ,  0 ,  0,  1 ];

Tras_Servo3=[ 1,  0,  0,  Rb*cos(Mu3);
              0,  1,  0,  Rb*sin(Mu3)];

```

```

0, 0, 1, 0 ;
0, 0, 0, 1 ];

%%
%%%%%%%%%%%%% Servo 4
%%%%%%%%%%%%% Servo 4
RotZ_Servo4=[ cos(Mu4-pi/2) , -sin(Mu4-pi/2) , 0, 0 ;
              sin(Mu4-pi/2) , cos(Mu4-pi/2) , 0, 0 ;
              0 , 0 , 1, 0 ;
              0 , 0 , 0, 1 ];

Tras_Servo4=[ 1, 0, 0, Rb*cos(Mu4);
              0, 1, 0, Rb*sin(Mu4);
              0, 0, 1, 0 ;
              0, 0, 0, 1 ];

%%
%%%%%%%%%%%%% Servo 5
%%%%%%%%%%%%% Servo 5
RotZ_Servo5=[ cos(Mu5) , -sin(Mu5) , 0, 0 ;
              sin(Mu5) , cos(Mu5) , 0, 0 ;
              0 , 0 , 1, 0 ;
              0 , 0 , 0, 1 ];

Tras_Servo5=[ 1, 0, 0, Rb*cos(Mu5);
              0, 1, 0, Rb*sin(Mu5);
              0, 0, 1, 0 ;
              0, 0, 0, 1 ];

%%
%%%%%%%%%%%%% Servo 6
%%%%%%%%%%%%% Servo 6
RotZ_Servo6=[ cos(Mu6-pi/2) , -sin(Mu6-pi/2) , 0, 0 ;
              sin(Mu6-pi/2) , cos(Mu6-pi/2) , 0, 0 ;
              0 , 0 , 1, 0 ;
              0 , 0 , 0, 1 ];

Tras_Servo6=[ 1, 0, 0, Rb*cos(Mu6);
              0, 1, 0, Rb*sin(Mu6);
              0, 0, 1, 0 ;
              0, 0, 0, 1 ];

%%
%Matrices de transformación de posición B1, B3 y B5
%respecto {B}
B1=Tras_Servo1*RotZ_Servo1*RotY_135; %-----AVR-----
B2=Tras_Servo2*RotZ_Servo2*RotX_246; %-----AVR-----

```

```

B3=Tras_Servo3*RotZ_Servo3*RotY_135; %-----AVR-----
B4=Tras_Servo4*RotZ_Servo4*RotX_246; %-----AVR-----
B5=Tras_Servo5*RotZ_Servo5*RotY_135; %-----AVR-----
B6=Tras_Servo6*RotZ_Servo6*RotX_246; %-----AVR-----

%Nota: Servos 1,3 y 5 tienen eje ui paralelo al eje Y
%De manera análoga servos 2, 4 y 6 poseen eje vi paralelo
%al eje Y
%%%
%----- POSICIÓN DE PUNTOS {Pi} -----
%%%
%%%Matriz de traslación de origen Oxyz a Ouvw
b_T_Op=[1,0,0,0;
         0,1,0,0;
         0,0,1,H+z;
         0,0,0,1];
%%%Matrices de rotación
% Roll=[1 , 0 , 0 , 0;
%       0 , cos(Alfa), -sin(Alfa), 0;
%       0 , sin(Alfa), cos(Alfa), 0;
%       0 , 0 , 0 , 1];
%
% Pitch=[ cos(Beta), 0, sin(Beta), 0 ;
%          0 , 1, 0 , 0 ;
%          -sin(Beta), 0, cos(Beta), 0 ;
%          0 , 0 , 0 , 1 ];
%
R=[ cosd(Gamma)*cosd(Beta),
     -sind(Gamma)*cosd(Alfa)+cosd(Gamma)*sind(Beta)*sind(Alfa),
     sind(Gamma)*sind(Alfa)+cosd(Gamma)*sind(Beta)*cosd(Alfa), 0;
     sind(Gamma)*cosd(Beta), cosd(Gamma)*cosd(Alfa)+sind(Gamma)*sind(Beta)*sind(Alfa),
     -cosd(Gamma)*sind(Alfa)+sind(Gamma)*sind(Beta)*cosd(Alfa), 0;
     -sind(Beta) , cosd(Beta)*sind(Alfa) ,
     cosd(Beta)*cosd(Alfa) , 0;
     0 , 0 , 0 , 0];
1]

%%%Matriz de posición y orientación de sistema {P} respecto
%%% a {B}
% bTRp=b_T_Op*Roll*Pitch %AITOR
% bTRp=b_T_Op*Pitch*Roll; %MIKE
bTRp=b_T_Op*R

%%%Posición angular de puntos Pi (rótulas superiores)
%%%respecto marco {P} de coordenadas
Theta1= 51.24*(pi/180);
Theta2= 68.77*(pi/180);
Theta3= 171.24*(pi/180);
Theta4= -171.24*(pi/180); %188.76
Theta5= -68.77*(pi/180); %291.23
Theta6= -51.24*(pi/180); %308.75

```

```

P1=[ 1, 0, 0, Rp*cos(Theta1);
      0, 1, 0, Rp*sin(Theta1);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

P2=[ 1, 0, 0, Rp*cos(Theta2);
      0, 1, 0, Rp*sin(Theta2);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

P3=[ 1, 0, 0, Rp*cos(Theta3);
      0, 1, 0, Rp*sin(Theta3);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

P4=[ 1, 0, 0, Rp*cos(Theta4);
      0, 1, 0, Rp*sin(Theta4);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

P5=[ 1, 0, 0, Rp*cos(Theta5);
      0, 1, 0, Rp*sin(Theta5);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

P6=[ 1, 0, 0, Rp*cos(Theta6);
      0, 1, 0, Rp*sin(Theta6);
      0, 0, 1, 0 ;
      0, 0, 0, 1 ];

%Posición de las rótulas superiores
b_T_p1=bTRp*P1;
b_T_p2=bTRp*P2;
b_T_p3=bTRp*P3;
b_T_p4=bTRp*P4;
b_T_p5=bTRp*P5;
b_T_p6=bTRp*P6;

inv_B1= inv(B1);
inv_B2= inv(B2);
inv_B3= inv(B3);
inv_B4= inv(B4);
inv_B5= inv(B5);
inv_B6= inv(B6);

Pos_P1=inv_B1*b_T_p1*[0;0;0;1]
x1=Pos_P1(1,1);
y1=Pos_P1(2,1);
z1=Pos_P1(3,1);

Pos_P2=inv_B2*b_T_p2*[0;0;0;1]

```

```

x2=Pos_P2(1,1);
y2=Pos_P2(2,1);
z2=Pos_P2(3,1);

Pos_P3=inv_B3*b_T_p3*[0;0;0;1]
x3=Pos_P3(1,1);
y3=Pos_P3(2,1);
z3=Pos_P3(3,1);

Pos_P4=inv_B4*b_T_p4*[0;0;0;1]
x4=Pos_P4(1,1);
y4=Pos_P4(2,1);
z4=Pos_P4(3,1);

Pos_P5=inv_B5*b_T_p5*[0;0;0;1]
x5=Pos_P5(1,1);
y5=Pos_P5(2,1);
z5=Pos_P5(3,1);

Pos_P6=inv_B6*b_T_p6*[0;0;0;1]
x6=Pos_P6(1,1);
y6=Pos_P6(2,1);
z6=Pos_P6(3,1);

%Aitor - 80
% delta1=(atan( / ))*(180/pi)-80
% psi1=( acos( ( Cs^2 ) + ( Pos_P1(1,1)^2 ) + ( Pos_P1(2,1)^2 ) - ( L^2 ) ) / ( 2*Cs*(sqrt( ( Pos_P1(1,1)^2 ) + ( Pos_P1(2,1)^2 ) ) ) )*(180/pi)-80
% sigma1=psi1+delta1

% K= atand( Pos_P1(2,1) / Pos_P1(1,1) )

%-----sigma1-----
% SIGMA1=90-atand(y1/x1);
SIGMA1= atand(x1/y1)
delta1= acosd((-266.49+ x1^2+ y1^2+ z1^2) /(4.8*(sqrt( x1^2 + y1)) ))
sigma1=SIGMA1-delta1

%-----sigma2-----
SIGMA2=atand(y2/x2)
delta2=acosd((x2^2 + y2^2 + z2^2 - 266.49) / ( 4.8*(sqrt(x2^2+y2^2)) ) )
sigma2=SIGMA2-delta2

%-----sigma3-----
SIGMA3=atand(x3/y3);
delta3= acosd( (-266.49+ x3^2+ y3^2+ z3^2) /(4.8*(sqrt( x3^2 + y1)) ));
sigma3=SIGMA3-delta3;

%-----sigma4-----

```

```

SIGMA4=atand(y4/x4);
delta4=acosd((x4^2 + y4^2 + z4^2 - 266.49) / ( 4.8*(sqrt(x4^2+y4^2)) ));
sigma4=SIGMA4-delta4;

%-----sigma5-----
SIGMA5=atand(x5/y5);
delta5= acosd( (-266.49+ x5^2+ y5^2+ z5^2) /(4.8*(sqrt( x5^2 + y1)))); 
sigma5=SIGMA5-delta5;

%-----sigma6-----
SIGMA6=atand(y6/x6);
delta6=acosd( (x6^2 + y6^2 + z6^2 - 266.49) / ( 4.8*(sqrt(x6^2+y6^2)) ) );
sigma6=SIGMA6-delta6;

Angulos_Servomotores=[sigma1;
sigma2;
sigma3;
sigma4;
sigma5;
sigma6]

```

#### 16.7.4 G4: PID Servomotores

El diagrama de bloques de la simulación del controlador PID digital de servomotores (ecuación (23)) se muestra en la Figura 90. Donde las ganancias tienen valores de  $k_p=1$ ,  $k_I=0.1$  y  $k_D=1$ .

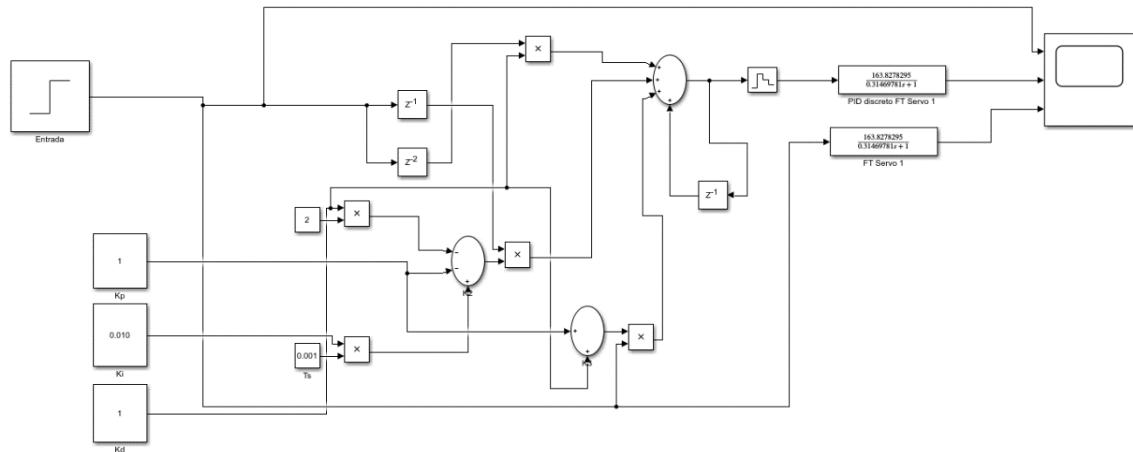


Figura 90 Diagrama de Bloques de Controlador PID digital (Simulink).

### 16.7.5 G5: PID Panel Resistivo

En la Figura 91 se muestra el PID implementado a la función de transferencia del panel resistivo:

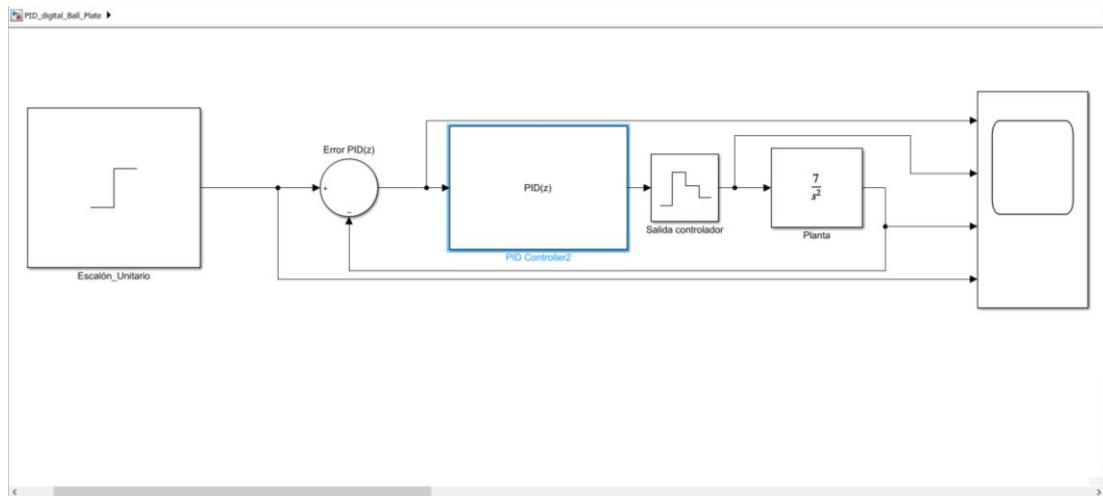


Figura 91 Diagrama de bloques controlador PID digital panel resistivo.

Las ganancias se observan en la Figura 92

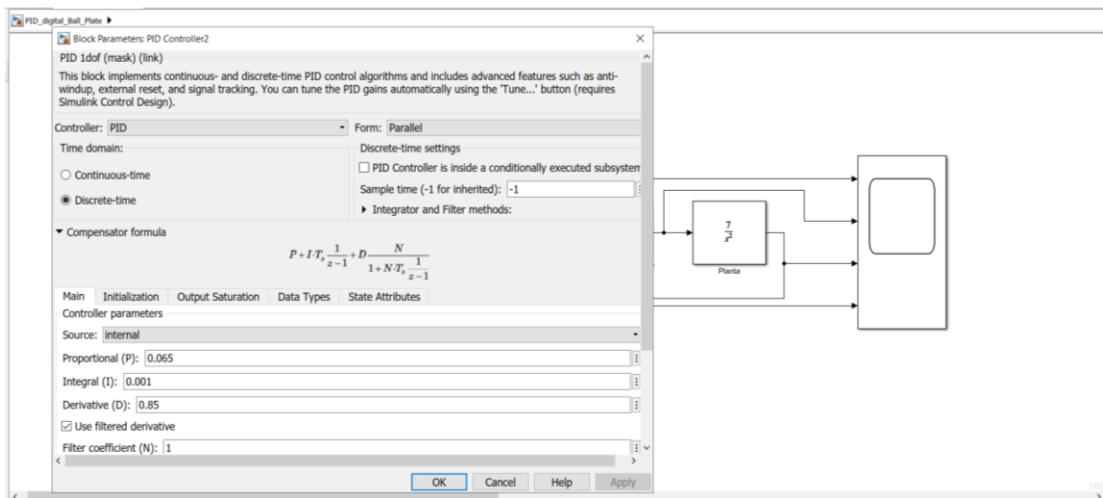


Figura 92 Ganancias controlador PID panel resistivo (simulink).

## 17 ANEXO H: CÓDIGOS DE IMPLEMENTACION

### 17.1.1 H1: Librerías para el Usuario

#### 17.1.1.1 Puertos

```
#ifndef PORTS_H_
#define PORTS_H_

void ports_init();
```

```

// ----- CONFIGURAR PUERTO A -----
//      DDRA &=~ (1<<PA0);
//      DDRA |= (1<<PA1);
//      PORTA &=~ (1<<PA0);
//      PORTA |= (1<<PA0);
//      PORTA &=~ (1<<PA1);
//      PORTA |= (1<<PA1);

// ----- CONFIGURAR PUERTO B -----
//      DDRB &=~ 1;
//      DDRB |= 0;
//      PORTB &=~ 1;
//      PORTB |= 0;

// ----- CONFIGURAR PUERTO C -----
//      DDRC &=~ 1;
//      DDRC |= 0;
//      PORTC &=~ 1;
//      PORTC |= 0;

// ----- CONFIGURAR PUERTO D -----
//      DDRD &=~ 1;
//      DDRD |= 0;
//      PORTD &=~ 1;
//      PORTD |= 0;

// ----- CONFIGURAR PUERTO E -----
//      DDRE &=~ (1<<PE0);
//      DDRE |= (1<<DDE1)|(1<<DDE3)|(1<<DDE4)|(1<<DDE5);
//      PORTE &=~ 1;
//      PORTE |= 0;

// ----- CONFIGURAR PUERTO F -----
//      DDRF &=~ (1<<DDF0);
//      DDRF |= (1<<DDF1)|(1<<DDF2)|(1<<DDF3)|(1<<DDF4);
//      PORTF &=~ (1<<PF0);
//      PORTF = (0<<PF1)|(0<<PF2)|(0<<PF3)|(0<<PF4);

// ----- CONFIGURAR PUERTO G -----
//      DDRG &=~ 1;
//      DDRG |= 0;
//      PORTG &=~ 1;
//      PORTG |= 0;

// ----- CONFIGURAR PUERTO H -----
//      DDRH &=~ 1;
//      DDRH |= (1<<DDH3)|(1<<DDH4)|(1<<DDH5);
//      PORTH &=~ 1;
//      PORTH |= 0;

// ----- CONFIGURAR PUERTO J -----
//      DDRJ &=~ 1;
//      DDRJ |= 0;
//      PORTJ &=~ 1;
//      PORTJ |= 0;

// ----- CONFIGURAR PUERTO K -----
//      DDRK &=~ 1;
//      DDRK |= 0;

```

```

//      PORTK &=~ 1;
//      PORTK |= 0;

// ----- CONFIGURAR PUERTO L -----
//      DDRL &=~ 1;
//      DDRL |= 0;
//      PORTL &=~ 1;
//      PORTL |= 0;

}

```

### 17.1.1.2 ADC

```

#ifndef ADC_H_
#define ADC_H_

void adc_init(){
    ADMUX &=~ (1<<ADLAR);
    ADMUX &=~ (1<<REFS1);
    ADMUX |= (1<<REFS0);
    ADCSRA |= (1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
    ADCSRA |= (1<<ADEN);
    ADCSRA &=~ (1<<ADATE);
    ADCSRA &=~ (1<<ADIE);
    ADCSRB &=~ (1<<ACME);
    ADCSRB &=~ (1<<ADTS2)&(1<<ADTS1)&(1<<ADTS0);
    DIDR0 |= 0x00;
    DIDR2 |= 0x00;
    ADCSRA |= (1<<ADSC);
    while(!(ADCSRA & (1<<ADIF))){}
    ADCSRA |= (1<<ADIF);
}

uint16_t read_ADC(uint8_t canal)
{
    ADMUX &= 0b11100000;
    ADCSRB &= 0b11110111;
    ADMUX &= 0b11100000;
    ADCSRB &= 0b11110111;
    ADMUX |= canal & 0x07;
    ADCSRB = canal & (1<<MUX5);
    ADCSRA |= (1<<ADSC);
    while(!(ADCSRA & (1<<ADIF))){}
    ADCSRA |= (1<<ADIF);
    return(ADC);
}

#endif /* ADC_H_ */

```

### 17.1.1.3 UART

```

#ifndef UART_H_
#define UART_H_

#define DEBUG 1
#define BAUDRATE 9600
#define BAUDRATE_UBRR0 ((F_CPU/8/BAUDRATE)-1)

```

```

void uart_init(){
    UCSR0A &= (1<<MPCM0);
    UCSR0A |= (1<<U2X0);
    UCSR0B &= (1<<TXCIE0)&(1<<UDRIE0)&(1<<UCSZ02)&(1<<RXB80)&(1<<TXB80);
    UCSR0B |= (1<<RXCIE0)|(1<<RXEN0)|(1<<TXEN0);
    UCSR0C &=-
(1<<UMSEL01)&(1<<UMSEL00)&(1<<UPM01)&(1<<UPM00)&(1<<USBS0)&(1<<UCPOL0);
    UCSR0C |= (1<<UCSZ01)|(1<<UCSZ00);
    UBRR0 = BAUDRATE_UBRR0;
}

unsigned char uart_read(){
    if(UCSR0A&(1<<RXC0)){
        return UDR0;
    }
    else
        return 0;
}

void uart_write(char *caracter){
    while (*caracter) {
        while (!(UCSR0A & (1 << UDRE0)));
        UDR0 = *caracter;
        caracter++;
    }
    while (!(UCSR0A & (1 << UDRE0)));
}

void uart writeln(char *caracter){
    while (*caracter) {
        while (!(UCSR0A & (1 << UDRE0)));
        UDR0 = *caracter;
        caracter++;
    }
    while (!(UCSR0A & (1 << UDRE0)));
    UDR0 = '\r';
    UDR0 = '\n';
}

#endif /* UART_H */

```

#### 17.1.1.4 Servomotores

```

#ifndef SERVOS_H_
#define SERVOS_H_

#define prescalador 8
#define f_servos 50
#define TOP_timer ((F_CPU/prescalador/f_servos) - 1)

float pot[6]={};
float pot_giro[6]={};
char pot1[10],pot2[10],pot3[10],pot4[10],pot5[10],pot6[10];

void servos_init()
{
    TCCR3A &= (1<<WGM30);
    TCCR3A |= (1<<WGM31);
    TCCR3B |= (1<<WGM32);
    TCCR3B |= (1<<WGM33);

```

```

TCCR3B &=~ (1<<ICNC3);
TCCR3B &=~ (1<<CS32);
TCCR3B &=~ (1<<CS31);
TCCR3B &=~ (1<<CS30);
TCCR3C = 0x00;
TIMSK3 = 0x00;
TIFR3 = 0x00;
ICR3 = TOP_timer;
TCCR4A &=~ (1<<WGM40);
TCCR4A |= (1<<WGM41);
TCCR4B |= (1<<WGM42);
TCCR4B |= (1<<WGM43);
TCCR4B &=~ (1<<ICNC4);
TCCR4B &=~ (1<<CS42);
TCCR4B &=~ (1<<CS41);
TCCR4B &=~ (1<<CS40);
TCCR4C = 0x00;
TIMSK4 = 0x00;
TIFR4 = 0x00;
ICR4 = TOP_timer; }

void servos_start()
{
    TCNT3 = 0x0000;
    TCCR3B &=~ (1<<CS32);
    TCCR3B |= (1<<CS31);
    TCCR3B &=~ (1<<CS30);
    TCNT4 = 0x0000;
    TCCR4B &=~ (1<<CS42);
    TCCR4B |= (1<<CS41);
    TCCR4B &=~ (1<<CS40); }

void servos_stop()
{
    TCCR3B &=~ (1<<CS32);
    TCCR3B &=~ (1<<CS31);
    TCCR3B &=~ (1<<CS30);
    TCNT3 = 0x0000;
    TCCR4B &=~ (1<<CS42);
    TCCR4B &=~ (1<<CS41);
    TCCR4B &=~ (1<<CS40);
    TCNT4 = 0x0000;     }

void servo1(float posicion)
{
    if (posicion>60)
    {
        posicion=60;}
    if (posicion<-60)
    {
        posicion=-60;}
    float duty;
    duty = 2 + (((13-2)*(-posicion-(-90)))/(90-(-90)));
    TCCR3A |= (1<<COM3A1);
    TCCR3A &=~ (1<<COM3A0);
    OCR3A = (TOP_timer*duty)/100;
}
void servo2(float posicion)
{
    if (posicion>60)
    {
        posicion=60;}

```

```

    if (posicion<-60)
    {
        posicion=-60;
        float duty;
        duty = 2 + (((13-2)*(posicion-(-90)))/(90-(-90)));
        TCCR3A |= (1<<COM3B1);
        TCCR3A &=~ (1<<COM3B0);
        OCR3B = (TOP_timer*duty)/100;
    }
void servo3(float posicion)
{
    if (posicion>60)
    {
        posicion=60;
        if (posicion<-60)
        {
            posicion=-60;
            float duty;
            duty = 2 + (((13-2)*(-posicion-(-90)))/(90-(-90)));
            TCCR3A |= (1<<COM3C1);
            TCCR3A &=~ (1<<COM3C0);
            OCR3C = (TOP_timer*duty)/100;
        }
void servo4(float posicion)
{
    if (posicion>60)
    {
        posicion=60;
        if (posicion<-60)
        {
            posicion=-60;
            float duty;
            duty = 2 + (((12-2)*(posicion-(-90)))/(90-(-90)));
            TCCR4A |= (1<<COM4A1);
            TCCR4A &=~ (1<<COM4A0);
            OCR4A = (TOP_timer*duty)/100;
        }
void servo5(float posicion)
{
    if (posicion>60)
    {
        posicion=60;
        if (posicion<-60)
        {
            posicion=-60;
            float duty;
            duty = 3 + (((13-3)*(-posicion-(-90)))/(90-(-90)));
            TCCR4A |= (1<<COM4B1);
            TCCR4A &=~ (1<<COM4B0);
            OCR4B = (TOP_timer*duty)/100;
        }
void servo6(float posicion)
{
    if (posicion>60)
    {
        posicion=60;
        if (posicion<-60)
        {
            posicion=-60;
            float duty;
            duty = 2 + (((12-2)*(posicion-(-90)))/(90-(-90)));
            TCCR4A |= (1<<COM4C1);
            TCCR4A &=~ (1<<COM4C0);
            OCR4C = (TOP_timer*duty)/100;
        }
#endif /* SERVOS_H_ */

```

### 17.1.1.5 Panel

```
#ifndef PANEL_H_
#define PANEL_H_

#include "adc.h"
#include "ports.h"
#include "millis.h"

unsigned long panel_time;
#define asentamiento 1
char estado;
int temp;
float panel_aX, panel_aY, pX, pY;
char set_X[5], set_Y[5];

#define U_LONG_MAX 4294967295
#define U_INT_MAX 4294967295

unsigned long loop_tiempo(unsigned long s,unsigned long e,int t)
{
    unsigned long ret;
    if (s > e)
    {
        ret = s-e;
        if (t)
        {
            ret = U_LONG_MAX - ret;
        } else
        {
            ret = U_INT_MAX - ret;
        }
    } else
        ret = e-s;
    }
    return ret;
}

void read_PR(){
    switch(estado)
    {
        case 0:
        PORTF = 0b00001100;
        panel_time = millis();
        estado = 1;
        break;

        case 1:
        temp = loop_tiempo(panel_time, millis(), true);
        if (temp >= asentamiento)
        {
            estado = 2;
        }
        break;

        case 2:
        panel_aX=read_ADC(0);
        estado = 3;
        break;

        case 3:
```

```

PORTF = 0b00000110;
panel_time = millis();
estado = 4;
break;

case 4:
temp = loop_tiempo(panel_time, millis(), true);
if (temp >= asentamiento)
{
estado = 5;
break;

case 5:
panel_aY=read_ADC(0);
estado = 6;
break;

case 6:
pX = -161 + (161-(-161))*(panel_aX-250)/(750-250);
pY = -124 + (124-(-124))*(panel_aY-250)/(750-250);
dtostrf(pX, 0, 0, set_X);
dtostrf(pY, 0, 0, set_Y);
estado = 7;
break;

case 7:
PORTF = 0b00000000;
estado = 0;
break;

default:
uart_write("\n\rAlgo anda mal bro");
break;
}
}

#endif /* PANEL_H_ */

```

#### 17.1.1.6 Millis

```

#ifndef MILLIS_H_
#define MILLIS_H_

#include <util/atomic.h>
unsigned long millis_counter;

ISR(TIMER0_OVF_vect){
    PORTJ ^= (1<<PJ5);
    millis_counter++;
    TCNT0 = 6;
}

void millis_init()
{
    DDRJ |= (1<<DDJ5);
    TCCR0B &=~ (1<<CS02);
    TCCR0B |= (1<<CS01);
}

```

```

        TCCR0B |= (1<<CS00);
        TCNT0 = 6;
        TIMSK0 |= (1<<TOIE0);
    }

    unsigned long millis(){
        unsigned long millis_return;
        ATOMIC_BLOCK(ATOMIC_FORCEON) {
            millis_return = millis_counter;
        }
        return millis_return;
    }

#endif /* MILLIS_H_ */

```

### 17.1.1.7 MPU-6050

```

#ifndef MPU6050_H_
#define MPU6050_H_
#define XG_OFFSET_TC 0x00
#define YG_OFFSET_TC 0x01
#define ZG_OFFSET_TC 0x02
#define X_FINE_GAIN 0x03
#define Y_FINE_GAIN 0x04
#define Z_FINE_GAIN 0x05
#define XA_OFFSET_H 0x06
#define XA_OFFSET_L_TC 0x07
#define YA_OFFSET_H 0x08
#define YA_OFFSET_L_TC 0x09
#define ZA_OFFSET_H 0x0A
#define ZA_OFFSET_L_TC 0x0B
#define XG_OFFSET_USRH 0x13
#define XG_OFFSET_USRL 0x14
#define YG_OFFSET_USRH 0x15
#define YG_OFFSET_USRL 0x16
#define ZG_OFFSET_USRH 0x17
#define ZG_OFFSET_USRL 0x18
#define SMPLRT_DIV 0x19
#define CONFIG 0x1A
#define GYRO_CONFIG 0x1B
#define ACCEL_CONFIG 0x1C
#define FF_THR 0x1D
#define FF_DUR 0x1E
#define MOT_THR 0x1F
#define MOT_DUR 0x20
#define ZRMOT_THR 0x21
#define ZRMOT_DUR 0x22
#define FIFO_EN 0x23
#define I2C_MST_CTRL 0x24
#define I2C_SLV0_ADDR 0x25
#define I2C_SLV0_REG 0x26
#define I2C_SLV0_CTRL 0x27
#define I2C_SLV1_ADDR 0x28
#define I2C_SLV1_REG 0x29
#define I2C_SLV1_CTRL 0x2A
#define I2C_SLV2_ADDR 0x2B
#define I2C_SLV2_REG 0x2C

```

```

#define I2C_SLV2_CTRL 0x2D
#define I2C_SLV3_ADDR 0x2E
#define I2C_SLV3_REG 0x2F
#define I2C_SLV3_CTRL 0x30
#define I2C_SLV4_ADDR 0x31
#define I2C_SLV4_REG 0x32
#define I2C_SLV4_DO 0x33
#define I2C_SLV4_CTRL 0x34
#define I2C_SLV4_DI 0x35
#define I2C_MST_STATUS 0x36
#define INT_PIN_CFG 0x37
#define INT_ENABLE 0x38
#define DMP_INT_STATUS 0x39
#define INT_STATUS 0x3A
#define ACCEL_XOUT_H 0x3B
#define ACCEL_XOUT_L 0x3C
#define ACCEL_YOUT_H 0x3D
#define ACCEL_YOUT_L 0x3E
#define ACCEL_ZOUT_H 0x3F
#define ACCEL_ZOUT_L 0x40
#define TEMP_OUT_H 0x41
#define TEMP_OUT_L 0x42
#define GYRO_XOUT_H 0x43
#define GYRO_XOUT_L 0x44
#define GYRO_YOUT_H 0x45
#define GYRO_YOUT_L 0x46
#define GYRO_ZOUT_H 0x47
#define GYRO_ZOUT_L 0x48
#define EXT_SENS_DATA_00 0x49
#define EXT_SENS_DATA_01 0x4A
#define EXT_SENS_DATA_02 0x4B
#define EXT_SENS_DATA_03 0x4C
#define EXT_SENS_DATA_04 0x4D
#define EXT_SENS_DATA_05 0x4E
#define EXT_SENS_DATA_06 0x4F
#define EXT_SENS_DATA_07 0x50
#define EXT_SENS_DATA_08 0x51
#define EXT_SENS_DATA_09 0x52
#define EXT_SENS_DATA_10 0x53
#define EXT_SENS_DATA_11 0x54
#define EXT_SENS_DATA_12 0x55
#define EXT_SENS_DATA_13 0x56
#define EXT_SENS_DATA_14 0x57
#define EXT_SENS_DATA_15 0x58
#define EXT_SENS_DATA_16 0x59
#define EXT_SENS_DATA_17 0x5A
#define EXT_SENS_DATA_18 0x5B
#define EXT_SENS_DATA_19 0x5C
#define EXT_SENS_DATA_20 0x5D
#define EXT_SENS_DATA_21 0x5E
#define EXT_SENS_DATA_22 0x5F
#define EXT_SENS_DATA_23 0x60
#define MOT_DETECT_STATUS 0x61
#define I2C_SLV0_DO 0x63
#define I2C_SLV1_DO 0x64
#define I2C_SLV2_DO 0x65
#define I2C_SLV3_DO 0x66
#define I2C_MST_DELAY_CTRL 0x67
#define SIGNAL_PATH_RESET 0x68

```

```

#define MOT_DETECT_CTRL 0x69
#define USER_CTRL 0x6A
#define PWR_MGMT_1 0x6B
#define PWR_MGMT_2 0x6C
#define BANK_SEL 0x6D
#define MEM_START_ADDR 0x6E
#define MEM_R_W 0x6F
#define DMP_CFG_1 0x70
#define DMP_CFG_2 0x71
#define FIFO_COUNTH 0x72
#define FIFO_COUNTL 0x73
#define FIFO_R_W 0x74
#define WHO_AM_I 0x75

#define SCL_CLK 100000UL
#define BITRATE(TWSR) ((F_CPU/SCL_CLK)-16)/(2*pow(4,(TWSR&((1<<TWPS0)|(1<<TWPS1)))))

void i2c_init(){
    TWBR = BITRATE(TWSR = 0x00);
}

uint8_t I2C_Start(char slave_write_address){
    uint8_t status;
    TWCR = (1<<TWSTA)|(1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    status = TWSR & 0xF8;
    if (status != 0x08)
        return 0;
    TWDR = slave_write_address;
    TWCR = (1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    status = TWSR & 0xF8;
    if (status == 0x18)
        return 1;
    if (status == 0x20)
        return 2;
    else
        return 3;
}

uint8_t I2C_Repeated_Start(char slave_read_address){
    uint8_t status;
    TWCR = (1<<TWSTA)|(1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    status = TWSR & 0xF8;
    if (status != 0x10)
        return 0;
    TWDR = slave_read_address;
    TWCR = (1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    status = TWSR & 0xF8;
    if (status == 0x40)
        return 1;
    if (status == 0x20)
        return 2;
    else
        return 3;
}

```

```

void I2C_Stop(){
    TWCR=(1<<TWSTO)|(1<<TWINT)|(1<<TWEN);
    while(TWCR & (1<<TWSTO));
}

void I2C_Start_Wait(char slave_write_address){
    uint8_t status;
    while (1)
    {
        TWCR = (1<<TWSTA)|(1<<TWEN)|(1<<TWINT);
        while (!(TWCR & (1<<TWINT)));
        status = TWSR & 0xF8;
        if (status != 0x08)
            continue;
        TWDR = slave_write_address;
        TWCR = (1<<TWEN)|(1<<TWINT);
        while (!(TWCR & (1<<TWINT)));
        status = TWSR & 0xF8;
        if (status != 0x18 )
        {
            I2C_Stop();
            continue;
        }
        break;
    }
}

uint8_t I2C_Write(char data){
    uint8_t status;
    TWDR = data;
    TWCR = (1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    status = TWSR & 0xF8;
    if (status == 0x28)
        return 0;
    if (status == 0x30)
        return 1;
    else
        return 2;
}

char I2C_Read_Ack(){
    TWCR=(1<<TWEN)|(1<<TWINT)|(1<<TWEA);
    while (!(TWCR & (1<<TWINT)));
    return TWDR;
}

char I2C_Read_Nack(){
    TWCR=(1<<TWEN)|(1<<TWINT);
    while (!(TWCR & (1<<TWINT)));
    return TWDR;
}

int get_Ax, get_Ay, get_Az, get_Gx, get_Gy, get_Gz, get_Temperature;
void mpu6050_init(){
    _delay_ms(150);
    I2C_Start_Wait(0xD0);
    I2C_Write(SMPLRT_DIV);
    I2C_Write(0x07);
    I2C_Stop();
}

```

```

I2C_Start_Wait(0xD0);
I2C_Write(PWR_MGMT_1);
I2C_Write(0x01);
I2C_Stop();

I2C_Start_Wait(0xD0);
I2C_Write(CONFIG);
I2C_Write(0x00);
I2C_Stop();

I2C_Start_Wait(0xD0);
I2C_Write(GYRO_CONFIG);
I2C_Write(0x18);
I2C_Stop();

I2C_Start_Wait(0xD0);
I2C_Write(INT_ENABLE);
I2C_Write(0x01);
I2C_Stop();
}

void MPU_Start_Loc(){
    I2C_Start_Wait(0xD0);
    I2C_Write(ACCEL_XOUT_H);
    I2C_Repeated_Start(0xD1);
}

void get_MP6050()
{
    MPU_Start_Loc();
    get_Ax = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Ay = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Az = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Temperature = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Gx = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Gy = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Ack());
    get_Gz = (((int)I2C_Read_Ack()<<8) | (int)I2C_Read_Nack());
    I2C_Stop();
}

char set_Ax[5],set_Ay[5],set_Az[5],set_Gx[5],set_Gy[5],set_Gz[5];
int ang_x, ang_y, ang_z, ang_x_prev, ang_y_prev, ang_z_prev;
float accel_ang_x, accel_ang_y;

void read_MP6050(){
    get_MP6050();
    float accel_ang_x = atan(get_Ay/sqrt(pow(get_Ax,2) + pow(get_Az,2)))*(180.0/3.14);
    float accel_ang_y = atan(get_Ax/sqrt(pow(get_Ay,2) + pow(get_Az,2)))*(180.0/3.14);
    ang_x = 0.8*(ang_x_prev + (get_Gx/131)) + 0.2*(accel_ang_x);
    ang_y = -(0.5*(ang_y_prev + (get_Gy/131)) + 0.5*(accel_ang_y));
    ang_z = ang_z_prev + (get_Gz/131);
    ang_x_prev = ang_x;
    ang_y_prev = ang_y;
    ang_z_prev = ang_z;

    dtostrf(ang_x, 2, 0, set_Gx);
    dtostrf(ang_y, 2, 0, set_Gy);
    dtostrf(ang_z, 2, 0, set_Gz);
}

```

```
#endif /* MPU6050_H_ */
```

### 17.1.2 H2: PID servomotores

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "librerias/ports.h"
#include "librerias/servos.h"
#include "librerias/cinematica.h"
#include "librerias/mpu6050.h"
#include "librerias/uart.h"
#include "librerias/adc.h"
#include "librerias/pid.h"

#define T 0.010
float kp=1;
float ki=0.1;
float kd=1;
float c_2 = kd;
float c_1 = ki*T - kp - 2*kd;
float c_0 = kp + kd;
char pid_1[8],pid_2[8],pid_3[8],pid_4[8],pid_5[8],pid_6[8];

float set_servo1, A1, retro1;
float e_1[3]={0.0,0.0,0.0};
float u_1[2]={0.0,0.0};

float set_servo2, A2, retro2;
float e_2[3]={0.0,0.0,0.0};
float u_2[2]={0.0,0.0};

float set_servo3, A3, retro3;
float e_3[3]={0.0,0.0,0.0};
float u_3[2]={0.0,0.0};

float set_servo4, A4, retro4;
float e_4[3]={0.0,0.0,0.0};
float u_4[2]={0.0,0.0};

float set_servo5, A5, retro5;
float e_5[3]={0.0,0.0,0.0};
float u_5[2]={0.0,0.0};

float set_servo6, A6, retro6;
float e_6[3]={0.0,0.0,0.0};
float u_6[2]={0.0,0.0};

ISR(TIMER1_COMPA_vect){
    PORTJ ^= (1<<PJ6);

    set_servo1 = sigma1;
    pot[0] = read_ADC(8);
    pot_giro[0] = -80 + (((80-(-80))*(pot[0]-(0)))/(470-(0)));
    e_1[0]= set_servo1-pot_giro[0];
}
```

```

u_1[0]=c_0*e_1[0]+c_1*e_1[1]+c_2*e_1[2]+u_1[1];
if (u_1[0]>70)
{
    u_1[0]=70;
}
if (u_1[0]<-70)
{
    u_1[0]=-70;
}
servo1(u_1[0]);
e_1[2]=e_1[1];
e_1[1]=e_1[0];
u_1[1]=u_1[0];

set_servo2 = sigma2;
pot[1] = read_ADC(9);
pot_giro[1] = -80 + (((80-(-80))*(pot[1]-(0)))/(610-(0)));
e_2[0]= set_servo2-pot_giro[1];
u_2[0]=c_0*e_2[0]+c_1*e_2[1]+c_2*e_2[2]+u_2[1];
if (u_2[0]>70)
{
    u_2[0]=70;
}
if (u_2[0]<-70)
{
    u_2[0]=-70;
}
servo2(u_2[0]);
e_2[2]=e_2[1];
e_2[1]=e_2[0];
u_2[1]=u_2[0];

set_servo3 = sigma3;
pot[2] = read_ADC(10);
pot_giro[2] = -80 + (((80-(-80))*(pot[2]-(0)))/(540-(0)));
e_3[0]= set_servo3-pot_giro[2];
u_3[0]=c_0*e_3[0]+c_1*e_3[1]+c_2*e_3[2]+u_3[1];
if (u_3[0]>70)
{
    u_3[0]=70;
}
if (u_3[0]<-70)
{
    u_3[0]=-70;
}
servo3(u_3[0]);
e_3[2]=e_3[1];
e_3[1]=e_3[0];
u_3[1]=u_3[0];

set_servo4 = sigma4;
pot[3] = read_ADC(11);
pot_giro[3] = -80 + (((80-(-80))*(pot[3]-(0)))/(622-(0)));
e_4[0]= set_servo4-pot_giro[3];
u_4[0]=c_0*e_4[0]+c_1*e_4[1]+c_2*e_4[2]+u_4[1];
if (u_4[0]>70)
{
    u_4[0]=70;
}
if (u_4[0]<-70)
{
    u_4[0]=-70;
}
servo4(u_4[0]);
e_4[2]=e_4[1];
e_4[1]=e_4[0];
u_4[1]=u_4[0];

set_servo5 = sigma5;
pot[4] = read_ADC(12);
pot_giro[4] = -80 + (((80-(-80))*(pot[4]-(0)))/(545-(0)));

e_5[0]= set_servo5-pot_giro[4];
u_5[0]=c_0*e_5[0]+c_1*e_5[1]+c_2*e_5[2]+u_5[1];

```

```

if (u_5[0]>70)
{
    u_5[0]=70;
}
if (u_5[0]<-70)
{
    u_5[0]=-70;
}
servo5(u_5[0]);
e_5[2]=e_5[1];
e_5[1]=e_5[0];
u_5[1]=u_5[0];

set_servo6 = sigma1;
pot[5] = read_ADC(13);
pot_giro[5] = -80 + (((80-(-80))*(pot[5]-(0)))/(575-(0)));
e_6[0]= set_servo6-pot_giro[5];
u_6[0]=c_0*e_6[0]+c_1*e_6[1]+c_2*e_6[2]+u_6[1];
if (u_6[0]>70)
{
    u_6[0]=70;
}
if (u_6[0]<-70)
{
    u_6[0]=-70;

servo6(u_6[0]);
e_6[2]=e_6[1];
e_6[1]=e_6[0];
u_6[1]=u_6[0];
}

void pid_init()
{
    DDRJ |= (1<<DDJ6);
    TCCR1B &= (1<<WGM13);
    TCCR1B |= (1<<WGM12);
    TCCR1A &= (1<<WGM11);
    TCCR1A &= (1<<WGM10);

    TCCR1B &= (1<<CS12);
    TCCR1B |= (1<<CS11);
    TCCR1B |= (1<<CS10);

    OCR1A = (F_CPU/64/(1/T)) - 1;
    TIMSK1 |= (1<<OCIE1A);
}
void pid_on(){
    TCNT1 = 0x0000;
    TCCR1B &= (1<<CS12);
    TCCR1B |= (1<<CS11);
    TCCR1B |= (1<<CS10);
}
void pid_off(){
    TCCR1B &= (1<<CS10);
    TCCR1B &= (1<<CS11);
    TCCR1B &= (1<<CS12);
    TCNT1 = 0x0000;
}
void servo_cero(){
    servo1(0);
    servo2(0);
    servo3(0);
    servo4(0);
    servo5(0);
    servo6(0);
}

```

```

}

void demo1(){
    cinematica_inversa(0,0,0,0,0,0);
    _delay_ms(1000);
    cinematica_inversa(4,0,0,0,0,0);
    _delay_ms(1000);

    cinematica_inversa(0,0,0,0,0,0);
    _delay_ms(1000);
    cinematica_inversa(0,4,0,0,0,0);
    _delay_ms(1000);

    cinematica_inversa(0,0,0,0,0,0);
    _delay_ms(1000);
    cinematica_inversa(0,0,1.5,0,0,0);
    _delay_ms(1000);
}

void demo_giro(){
    read_MPU6050();
    cinematica_inversa(0,0,0,ang_x,ang_y,0);
}

int main(void){
    cli();
    ports_init();
    uart_init();
    i2c_init();
    mpu6050_init();
    servos_init();
    servos_start();
    adc_init();
    pid_init();
    sei();

    cinematica_inversa(0,0,0,0,0,0);
    _delay_ms(1000);
    while (1)
    {
        //demo1();
        //demo_giro();
    }
}

```

### 17.1.3 H3: PID Panel Resistivo

```

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "librerias/ports.h"
#include "librerias/servos.h"
#include "librerias/cinematica.h"
#include "librerias/mpu6050.h"
#include "librerias/uart.h"
#include "librerias/adc.h"
#include "librerias/pid.h"

```

```

#include "librerias/millis.h"

#include <avr/interrupt.h>
#include "adc.h"
#include "cinematica.h"
#include "servos.h"
#include "panel.h"

#define T 0.010 //Se define el tiempo de muestreo en ms
float kp=0.065;
float ki=0.001;
float kd=0.82;
float c_2 = kd;
float c_1 = ki*T - kp - 2*kd;
float c_0 = kp + kd;
char pid_1[8],pid_2[8];

float set_panelX;
float e_9[3]={0.0,0.0,0.0};
float u_9[2]={0.0,0.0};

float set_panelY;
float e_10[3]={0.0,0.0,0.0};
float u_10[2]={0.0,0.0};

ISR(TIMER1_COMPA_vect){
    PORTJ ^= (1<<PJ6);
    read_PR();

    set_panelX = 0;
    e_9[0]= set_panelX-pX;
    u_9[0]=c_0*e_9[0]+c_1*e_9[1]+c_2*e_9[2]+u_9[1];
    if (u_9[0]>15)
    {
        u_9[0]=15;
    }
    if (u_9[0]<-15)
    {
        u_9[0]=-15;
    }
    e_9[2]=e_9[1];
    e_9[1]=e_9[0];
    u_9[1]=u_9[0];

    set_panelY = 0;
    e_10[0]= set_panelY-pY;
    u_10[0]=c_0*e_10[0]+c_1*e_10[1]+c_2*e_10[2]+u_10[1];
    if (u_10[0]>15)
    {
        u_10[0]=15;
    }
    if (u_10[0]<-15)
    {
        u_10[0]=-15;
    }
    e_10[2]=e_10[1];
    e_10[1]=e_10[0];
    u_10[1]=u_10[0];
}

void pid_init()
{
    DDRJ |= (1<<DDJ6);
    TCCR1B &=~ (1<<WGM13);
    TCCR1B |= (1<<WGM12);
    TCCR1A &=~ (1<<WGM11);
    TCCR1A &=~ (1<<WGM10);
}

```

```

    TCCR1B &=~ (1<<CS12);
    TCCR1B |= (1<<CS11);
    TCCR1B |= (1<<CS10);
    OCR1A = (F_CPU/64/(1/T)) - 1;
    TIMSK1 |= (1<<OCIE1A);
}

void pid_on(){
    TCNT1 = 0x0000;
    TCCR1B &=~ (1<<CS12);
    TCCR1B |= (1<<CS11);
    TCCR1B |= (1<<CS10);
}

void pid_off(){
    TCCR1B &=~ (1<<CS10);
    TCCR1B &=~ (1<<CS11);
    TCCR1B &=~ (1<<CS12);
    TCNT1 = 0x0000;
}

void demo_panel(){
    cinematica_inversa(0,0,0,u_9[0],u_10[0],0);
}

int main(void){
    cli();
    ports_init();
    uart_init();
    i2c_init();
    mpu6050_init();
    millis_init();
    servos_init();
    servos_start();
    adc_init();
    pid_init();
    sei();

    cinematica_inversa(0,0,0,0,0);
    _delay_ms(1000);
    while (1)
    {
        demo_panel();
        dtostrf(u_9[0],0,4,pid_1);
        dtostrf(u_10[0],0,4,pid_2);

        uart_write("Posicion X: ");
        uart_write(set_X);
        uart_write("\tPID: ");
        uart_write(pid_1);

        uart_write("\t\tPosicion Y: ");
        uart_write(set_Y);
        uart_write("\tPID: ");
        uart_write(pid_1);
        uart_write("\r\n");
    }
}

```

#### 17.1.4 H4: Error marginal del PID de servomotores

