

DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE PLATAFORMA STEWART  
SERVO-ACTUADA

CRISTIAN ANDRES ACOSTA ALTAHONA

JHON EDINSON NÚÑEZ LARROTA

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA

2017

DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE PLATAFORMA STEWART  
SERVO-ACTUADA

CRISTIAN ANDRÉS ACOSTA ALTAHONA

JHON EDINSON NÚÑEZ LARROTA

Trabajo de Grado para optar el título de  
Ingeniero Mecánico

Director

JORGE ENRIQUE MENESES FLÓREZ

Ingeniero Mecánico

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA

2017

***A Dios.***

*Por haberme permitido alcanzar esta meta y haberme brindado salud, perseverancia y sabiduría para lograr mis objetivos además de su infinita bondad, amor y protección.*

***A mi madre Teresa.***

*Por ser el pilar fundamental en todo lo que soy y brindarme tu incondicional apoyo perfectamente mantenido a través del tiempo, por tus consejos, tus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por tu amor. Mamá gracias, todo este trabajo ha sido posible gracias a ti que con tu voz de aliento aumentabas la fe en momentos de debilidad.*

***A mis hermanos.***

*Por ser el incentivo para seguir adelante con este objetivo y espero que vean en mí un ejemplo a seguir.*

***A mi familia.***

*Al ejemplo de vida más grande que tengo, mis abuelos Juan y Evelia por quererme y apoyarme siempre, esto también se lo debo a ustedes; a mi tía Mary, a mi tía Denix y a todos aquellos que me apoyaron directa e indirectamente en el camino hacia convertirme en profesional.*

*¡Gracias, no hay un día en el que no agradezca a Dios el haberme colocado entre ustedes!*

*Cristian Acosta*

## **AGRADECIMIENTOS**

A nuestro director de proyecto de grado, Jorge Enrique Meneses Flórez por tener siempre la disposición de guiarnos en cada paso de este proyecto y plasmar en el la excelencia que lo caracteriza.

Al ingeniero Fabián Núñez Larrota, quien siempre tuvo disposición a colaborarnos y compartir sus conocimientos en áreas de control y caracterización de sistemas automáticos.

Al ingeniero Johan Sebastian Suarez Largo por compartir sus conocimientos en sistemas electrónicos, programación y sus habilidades de soldadura.

A los docentes Florelba Rozo García y Jaime Freddy Ricardo Barrera que nos ayudaron en asesorías y dudas presentadas durante la elaboración del proyecto de grado y a todos los docentes que marcaron cada etapa de nuestro camino universitario.

Al compañero y colega Carlos Augusto Ortiz Ortiz con quien siempre hubo un apoyo mutuo en nuestra formación profesional y aún ahora, seguimos siendo amigos, y a todos aquellos amigos que de una u otra manera aportaron a la realización de este proyecto de grado y con los cuales valió la pena luchar juntos por una meta en común.

Finalmente agradecer a todas las personas que siempre creyeron en la capacidad de nosotros, en especial a Nattanan Kultrailak, Kathrin Cisakowski y Jose Carlos Gómez Armenta por sus palabras de apoyo en todo momento.

Autores

## TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN.....	22
1. PLANTEAMIENTO DEL PROBLEMA .....	24
1.1 JUSTIFICACIÓN DEL PROBLEMA .....	25
1.2 OBJETIVO .....	26
1.2.1 Objetivo General.....	26
1.2.2 Objetivos Específicos. ....	26
1.3 ALTERNATIVAS DE DISEÑO .....	27
1.3.1 Según el tipo de grados de libertad .....	27
1.3.2 Según el tipo de actuadores .....	28
1.3.3 Según el tipo de unión .....	30
1.3.4 Según la adquisición de datos .....	31
1.3.5 Según el Software de aplicación .....	33
1.3.6 Según la señal de mando .....	34
1.3.7 Según la sensorica .....	37
1.4 SELECCIÓN DE ALTERNATIVAS .....	39
1.4.1 Prototipo .....	39
1.4.2 Actuación.....	39
1.4.3 Sistema de Articulación .....	39
1.4.4 Adquisición de Datos.....	40
1.4.5 Software de Aplicación .....	40
1.4.6 Señal de Mando .....	40
1.4.7 Sensorica.....	40
2. PROTOTIPO DE PLATAFORMA STEWART SERVOACTUADA.....	41
2.1 PLANTA.....	42
2.1.1 Prototipo Plataforma Stewart.....	43

2.1.2	Tarjeta de Adquisición de Datos.....	46
2.1.3	Sensor IMU.....	46
2.2	CONTROL DEL PROTOTIPO .....	46
2.3	INTERFAZ DE USUARIO .....	49
2.3.1	Aplicación Móvil.....	49
2.3.2	Emulador .....	50
3.	DISEÑO MECÁNICO Y CONSTRUCCIÓN DEL PROTOTIPO DE PLATAFORMA STEWART .....	51
3.1	PROTOTIPO PLATAFORMA STEWART .....	51
3.1.1	COMPONENTES DEL PROTOTIPO .....	55
3.1.1.1	Actuación .....	55
3.1.1.2	Elementos de Unión.....	56
3.1.1.3	Plataforma Base.....	57
3.1.1.4	Plataforma Móvil O Efector .....	58
3.1.2	Modelado CAD del Prototipo .....	60
3.1.3	Cinemática Inversa.....	61
3.1.3.1	Coordenadas de la Base.....	61
3.1.3.2	Coordenadas del Efector.....	67
3.1.3.3	Solución Cinemática Inversa.....	69
3.2	TARJETA DE ADQUISICIÓN DE DATOS .....	71
3.3	SENSOR IMU .....	72
4.	SOFTWARE DE CONTROL DEL PROTOTIPO DE PLATAFORMA STEWART .....	74
4.1	BLOQUE COMUNICACIÓN CON LA APLICACIÓN MÓVIL .....	75
4.2	BLOQUE SELECTOR MODO DE ENVÍO .....	77
4.2.1	Modo Automático ( <i>Caso 1</i> ).....	79
4.2.2	Modo Manual ( <i>Caso 2</i> ) .....	80
4.3	BLOQUE CINEMÁTICA INVERSA I y II .....	80
4.3.1	Coordenadas Junta "X" .....	81
4.3.2	Cálculo ángulo Servo "1".....	83

4.4	BLOQUE DE CONTROL .....	84
4.4.1	Controladores PID .....	87
4.4.1.1	Caracterización del Sistema.....	88
4.4.1.2	Diseño del Controlador.....	95
4.6	BLOQUE EMULADOR.....	99
4.7	BLOQUE MPU6050 .....	99
5.	INTERFAZ DE USUARIO .....	101
5.1	APLICACIÓN MÓVIL.....	101
5.1.1	Diseño de la Interfaz de la Aplicación.....	102
5.1.2	Programación de la Aplicación .....	103
5.1.2.1	Modo Automático .....	104
5.1.2.2	Modo Manual .....	105
5.1.3	Comunicación Bluetooth.....	107
5.1.4	Funcionamiento de la Aplicación .....	108
5.2	EMULADOR .....	109
6.	RESULTADOS.....	113
6.1	ESPACIO DE TRABAJO .....	113
6.1.1	Espacio de Trabajo de Traslación .....	115
6.1.1.1	Desplazamiento lineal X.....	116
6.1.1.2	Desplazamiento lineal Y .....	117
6.1.1.3	Desplazamiento lineal Z.....	118
6.1.2	Espacio de Trabajo De Rotación .....	119
6.1.2.1	Desplazamiento angular X o Pitch .....	119
6.1.2.2	Desplazamiento angular Y o Roll .....	120
6.1.2.3	Desplazamiento angular Z o Yaw.....	121
6.1.3	Espacio de Trabajo Diestro. ....	122
6.1.4	Desplazamientos Y Rotaciones Límites .....	122
6.2	COMPARACIÓN CON UN AVIÓN .....	123
6.3	TIPOLOGÍA DE LAS LECTURAS DEL SENSOR IMU .....	123

6.4	CONTROL EN LAZO ABIERTO .....	125
6.5	CONTROL EN LAZO CERRADO .....	130
7.	CONCLUSIONES .....	137
8.	RECOMENDACIONES .....	145
	BIBLIOGRAFÍA.....	147
	ANEXOS.....	149

## LISTAS DE FIGURAS

	pág.
Figura 1. Robot de estructura paralela, Plataforma Stewart. ....	25
Figura 2. Robot delta marca FESTO.....	27
Figura 3. Plataforma Stewart .....	28
Figura 4. Sistemas hidráulicos y neumáticos .....	29
Figura 5. Servomotor HITEC.....	30
Figura 6. Junta Esférica .....	30
Figura 7. Junta Universal Cardánica.....	31
Figura 8. Tarjeta Arduino UNO .....	32
Figura 9. Dispositivo DAQ Multifunción.....	32
Figura 10. IMU 9DOF V2 (Acelerómetro, Magnetómetro y Giroscopio).....	35
Figura 11. Simulación de despegue Aeronáutico.....	36
Figura 12. Ejes coordenados de los sensores de un móvil.....	37
Figura 13. Acelerómetro de tres ejes .....	38
Figura 14. Giroscopio de tres ejes digital.....	38
Figura 15. Concepto general del proyecto plataforma Stewart. ....	41
Figura 16. Comunicación entre la planta y el software de control.....	43
Figura 17. Modelo CAD del prototipo diseñado y partes principales.....	44
Figura 18. Esquema Software de control.....	48
Figura 19. Desarrollo de la App Android en App Inventor.....	49
Figura 20. Emulador de movimiento de la plataforma.....	50
Figura 21. Configuración de la plataforma hexápoda vista en Planta .....	52
Figura 22. Modelo de plataforma Stewart Servoactuada diseñada.....	53
Figura 23. Servomotor Analógico HK15178.....	55
Figura 24. Elementos de unión tipo rótulas.....	56
Figura 25. Dimensiones de la Plataforma Base.....	57
Figura 26. Mecanizado de la plataforma base. ....	58

Figura 27. Dimensiones de la Plataforma Móvil.....	59
Figura 28. Taladrado de la plataforma móvil.....	59
Figura 29. Modelo CAD de la plataforma Stewart.....	60
Figura 30. Ubicación ideal de las juntas de la Plataforma Base. ....	62
Figura 31. Representación vista frontal de los puntos B1 y B2 .....	63
Figura 32. Ubicación de los puntos B3 y B4.....	64
Figura 33. Ubicación de los puntos B5 y B6.....	65
Figura 34. Ubicación ideal de las juntas del Efecto. ....	67
Figura 35. Representación de las variables de influencia en la plataforma Stewart. .....	71
Figura 36. Sensor IMU MPU 6050.....	72
Figura 37. Programa de comunicación y control del prototipo de plataforma Stewart .....	74
Figura 38. Bloque de comunicación del programa de control .....	76
Figura 39. Configuración de los bloques de comunicación serial. ....	76
Figura 40. Programación del bloque selector modo de envío.....	78
Figura 41. Programación del bloque Modo Pitch. ....	78
Figura 42. Programación para la adquisición de datos en Modo Automático. ....	79
Figura 43. Programación para la adquisición de datos en Modo Manual. ....	80
Figura 44. Programación del bloque cinemática I y II. ....	81
Figura 45. Líneas de programación en el bloque Coordenadas Junta 1.....	82
Figura 46. Línea de programación del bloque Cálculo ángulo Servo "X". ....	83
Figura 47. Representación de la distancia $v$ y ángulo servomotor para punto <b>P2</b> . ....	84
Figura 48. Programación del bloque de control .....	85
Figura 49. Configuración básica del bloque <i>PID Controller</i> .....	86
Figura 50. Configuración avanzada del bloque <i>PID Controller</i> .....	87
Figura 51. Lectura de los ángulos Pitch y Roll del sensor instalado en el prototipo. .....	88
Figura 52. Posición del servomotor 1 sin controlador .....	89
Figura 53. Posición del servomotor 2 sin controlador .....	90

Figura 54. Posición del servomotor 3 sin controlador .....	90
Figura 55. Posición del servomotor 4 sin controlador .....	91
Figura 56. Posición del servomotor 5 sin controlador .....	91
Figura 57. Posición del servomotor 6 sin controlador .....	92
Figura 58. Respuesta de los modelos BJ de cada servomotor a un paso .....	94
Figura 59. Respuesta del sistema en lazo cerrado y lazo abierto.....	95
Figura 60. Ubicación del polo del sistema .....	96
Figura 61. Respuesta de los modelos a una señal escalón en el ángulo Pitch .....	97
Figura 62. Respuesta de los modelos a una señal escalón en el ángulo Roll .....	97
Figura 63. Programación del bloque de comunicación con el prototipo.....	99
Figura 64. Programación del bloque MPU6050 .....	100
Figura 65. Diseño de la interfaz de la aplicación móvil .....	103
Figura 66. Programación por bloques de la aplicación móvil.....	104
Figura 67. Bloques de programación del modo automático de la aplicación móvil .....	105
Figura 68. Bloques de programación del modo manual de la aplicación móvil....	106
Figura 69. Envío de las señales Modo Manual .....	107
Figura 70. Bloques de la conexión Bluetooth.....	108
Figura 71. App Android Plataforma Stewart.....	108
Figura 72. Emulación del prototipo de plataforma Stewart en el ambiente SIMMECHANICS.....	110
Figura 73. Programa general de la plataforma Stewart. ....	111
Figura 74. Diagrama de bloques del subsistema Simmechanics.....	111
Figura 75. Diagrama de bloques del subsistema Simmechanics.....	112
Figura 76. Diagrama de bloques del programa utilizado para determinar el espacio de trabajo del prototipo de plataforma diseñado. ....	114
Figura 77. Efecto en posición inicial o reposo.....	115
Figura 78. Espacio de trabajo del prototipo para desplazamiento en X .....	116
Figura 79. Espacio de trabajo del prototipo para desplazamiento en Y .....	117
Figura 80. Espacio de trabajo del prototipo para desplazamiento en Z .....	118

Figura 81. Espacio de trabajo del prototipo para desplazamiento angular en X. .	119
Figura 82. Espacio de trabajo del prototipo para desplazamiento angular en Y. .	120
Figura 83. Espacio de trabajo del prototipo para desplazamiento angular en Z. .	121
Figura 84. Espacio de trabajo de la Plataforma Stewart .....	122
Figura 85. Picos de lectura en los ángulos Pitch y Roll del sensor acelerómetro	124
Figura 86. Lecturas fusionadas de posición angular del efector móvil.....	125
Figura 87. Comportamiento prototipo en control lazo abierto .....	126
Figura 88. Posicionamiento del servomotor 1 en control con lazo abierto .....	127
Figura 89. Posicionamiento del servomotor 2 en control con lazo abierto .....	127
Figura 90. Posicionamiento del servomotor 3 en control con lazo abierto .....	128
Figura 91. Posicionamiento del servomotor 4 en control con lazo abierto .....	128
Figura 92. Posicionamiento del servomotor 5 en control con lazo abierto .....	129
Figura 93. Posicionamiento del servomotor 6 en control con lazo abierto .....	129
Figura 94. Resultado de control para variaciones en el ángulo Roll .....	132
Figura 95. Control de posición en lazo cerrado del servomotor 1 .....	133
Figura 96. Control de posición en lazo cerrado del servomotor 2.....	134
Figura 97. Control de posición en lazo cerrado del servomotor 3.....	134
Figura 98. Control de posición en lazo cerrado del servomotor 4.....	135
Figura 99. Control de posición en lazo cerrado del servomotor 5.....	135
Figura 100. Control de posición en lazo cerrado del servomotor 6.....	136
Figura 101. Picos de lectura en ángulos Pitch y Roll del sensor acelerómetro....	138
Figura 102. Cadenas cinemáticas 1 y 2 del prototipo diseñado.....	139
Figura 103. Gráfica ejemplo de saltos en la posición de los servomotores. ....	140
Figura 104. Sobre impulso en la señal de control del servomotor 2. ....	141
Figura 105. Control de posición angular del efector del prototipo.....	141
Figura 106. Respuesta del controlador PID inicial .....	142
Figura 107. Respuesta del controlador PID con cambio de temperatura.....	143
Figura 108. Picos presentes al inicio de cambio de posición del efector móvil. ...	144
Figura 109. Ejes coordenados del prototipo de plataforma Stewart.....	145

## LISTA DE TABLAS

pág.

Tabla 1. Características generales de la plataforma.....	51
Tabla 2. Características de los miniservos HK15178.....	56
Tabla 3. Modelos BJ de cada servomotor.....	93
Tabla 4. Variables $k$ y $T$ de los modelos BJ de cada servomotor .....	94
Tabla 5. Espacio de trabajo desplazamiento en X.....	116
Tabla 6. Espacio de trabajo desplazamiento en Y .....	117
Tabla 7. Espacio de trabajo desplazamiento en Z. ....	118
Tabla 8. Espacio de trabajo desplazamiento angular en X. ....	119
Tabla 9. Espacio de trabajo desplazamiento angular en Y .....	120
Tabla 10. Espacio de trabajo desplazamiento angular en Z .....	121
Tabla 11. Desplazamientos y rotaciones limites reales en la plataforma Stewart diseñada. ....	122
Tabla 12. Comparación de ángulos Pitch ejecutados en vuelo y alcanzados por prototipo desarrollado. ....	123
Tabla 13. Constantes del control PID de cada sistema .....	131
Tabla 14. Características del control de posición angular del efector móvil del prototipo.....	133
Tabla 15. Características de los controladores de cada servomotor .....	136
Tabla 16. Ángulos ejecutados en vuelo y en el prototipo diseñado. ....	137

## TABLA DE ANEXOS

	pág.
ANEXO A. PLANOS DE DISEÑO MODELO CAD PROTOTIPO PLATAFORMA STEWART .....	149
ANEXO B. DESARROLLO DE LA CINEMÁTICA INVERSA .....	155
ANEXO C. CÓDIGO TARJETA ARDUINO UNO .....	160
ANEXO D. PLANOS DE DISEÑO DE LA TARJETA PCB.....	170
ANEXO E. CREACIÓN DE UN DISPOSITIVO BLUETOOTH .....	172
ANEXO F. INSTALACIÓN DE LA LIBRERÍA ARDUINO IO EN SIMULINK .....	174
ANEXO G. INSTALACIÓN DEL COMPLEMENTO SIMMECHANICS.....	175

## RESUMEN

**TITULO:** DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE PLATAFORMA STEWART SERVO-ACTUADA\*

**AUTORES:** CRISTIAN ANDRES ACOSTA ALTAHONA, JHON EDINSON NÚÑEZ LARROTA \*\*

**PALABRAS CLAVE:** CONTROL, ROBÓTICA PARALELA, PLATAFORMA STEWART, SIMULINK, MATLAB, SIMECHANICS, MECATRÓNICA.

### DESCRIPCIÓN:

El diseño del prototipo de plataforma Stewart tratado en este proyecto cuenta con sistemas tecnológicos económicos y prácticos conocidos por los estudiantes de ingeniería, dispositivos actuales con los cuales es posible manipular, comunicar y controlar el prototipo, entre los que tenemos mini servomotores, dispositivos IMU's, *smartphones*, comunicación *bluetooth* y uso de fuentes computacionales que hacen de este proyecto una iniciativa al estudio, diseño e innovaciones de robots paralelos.

El primer capítulo abarca el diseño y construcción de la planta constituida por el prototipo de plataforma Stewart, los servomotores usados como actuadores, la tarjeta de adquisición de datos Arduino UNO y el sensor de retroalimentación MPU 6050. El segundo capítulo abarca el *software* de control del prototipo ejecutado en el *software* MatLab/Simulink, el cual es el encargado de comunicar el prototipo con la aplicación de mando, recibir y tratar las señales de posición angular de mando, efectuar los cálculos cinemáticos que resuelven la posición angular que debe configurarse en cada servomotor para cumplir la posición del efector móvil

---

\* Trabajo de Investigación

\*\* Facultad de Ingenierías Fisicomecánicas, Ingeniería Mecánica. Jorge Enrique Meneses Flórez

configurada por el usuario, comunicar con la tarjeta de adquisición y realizar el control PID a cada una de las señales de los servomotores. Por último, el capítulo de diseño y creación de la interfaz de usuario del proyecto, constituida por la aplicación móvil utilizada como mando del prototipo desarrollada en el compilador App Inventor y el emulador de movimiento utilizado para visualizar el comportamiento cinemático del prototipo de forma virtual mediante el uso de la librería SimMechanics en Simulink.

Por otro lado se analiza el espacio de trabajo resultado del diseño considerado en este proyecto y los resultados de aplicar control PID a las señales de cada uno de los servomotores que actúan el prototipo.

## ABSTRACT

**TITLE:** DESIGN AND CONSTRUCTION OF A SERVO-ACTUATED STEWART PLATFORM PROTOTYPE.\*

**AUTHORS:** CRISTIAN ANDRES ACOSTA ALTAHONA, JHON EDINSON NÚÑEZ LARROTA\*\*

**KEYWORDS:** CONTROL, PARALELL ROBOTICS, STEWART PLATFORM, SIMULINK, MATLAB, SIMMECHANICS, MECHATRONICS.

### **DESCRIPTION:**

The design of the prototype Stewart platform worked in this project has economic and practical technological systems known by engineering students, current devices with which it is possible to manipulate, communicate and control the prototype, among which we have mini servomotors, IMU devices, Smartphones, bluetooth communication and use of computational sources that make this project an initiative for the study, design and innovations of parallel robots.

The first chapter covers the design and construction of the prototype Stewart platform, the servo motors used as actuators, the Arduino UNO data acquisition board and the MPU 6050 feedback sensor. The second chapter covers the control software of the Prototype executed in the software MatLab / Simulink, which is responsible for communicating the prototype with the application of command, receive and treat the angular command position signals, perform kinematic calculations that solve the angular position to be configured in each servomotor To meet the position of the mobile effector configured by the user, communicate with

---

\* Research work

\*\* Faculty of Physicomechanical Engineering, Mechanical Engineering, Jorge Enrique Meneses Flórez

the acquisition card and perform the PID control to each of the signals of the servomotors. Finally, the chapter on design and creation of the user interface of the project, consisting of the mobile application used as a prototype command developed in the App Inventor compiler and the motion emulator used to visualize the kinematic behavior of the prototype in a virtual way through The use of the SimMechanics library in Simulink.

On the other hand we analyze the workspace resulting from the design considered in this project and the results of applying PID control to the signals of each of the servomotors that act the prototype.

## INTRODUCCIÓN

La simulación de vuelo es un notable avance tecnológico utilizado en la actualidad por millones de personas alrededor del mundo, ya sea presentado en videojuegos como entretenimiento o en situaciones más serias como el entrenamiento de pilotos. Los simuladores de vuelo actuales no solo tratan de recrear ambientes donde se experimenta el control de movimiento de un aeromodelo sino también poner a prueba la capacidad del usuario en situaciones aleatorias y complejas, es por esto que en la actualidad este es un campo de estudio que busca reproducir experiencias cada vez más reales y precisas.

El mecanismo que acciona una cabina de simulación de vuelo es un mecanismo de estructura paralela denominado Plataforma Stewart, este tipo de robot cuenta con dos plataformas, una fija y una móvil, actuadores responsables del movimiento de la plataforma móvil y una serie de cadenas cinemáticas independientes que unen dichas plataformas para formar en su conjunto el robot paralelo.

La iniciativa concebida en el laboratorio de Automatización Industrial de la Universidad Industrial de Santander que consiste en desarrollar un simulador de vuelo que funcione como plataforma para el estudio de control en lazo cerrado de robots paralelos por parte de la comunidad UIS, planteó como primera fase la necesidad de crear un prototipo a escala reducida de plataforma Stewart que respondiera de manera real a las solicitudes realizadas desde un mando y controlara el movimiento del efector móvil en lazo cerrado.

El diseño del prototipo de plataforma Stewart planteado en este proyecto cuenta con sistemas tecnológicos económicos y prácticos conocidos por los estudiantes de ingeniería, dispositivos actuales con los cuales es posible manipular, comunicar y controlar el prototipo. El proyecto se centró en diseñar la estructura y analizar las

respuestas de movimiento del prototipo de plataforma Stewart hexápoda actuada con mini servomotores, recibir las señales de posición configurada por el usuario en dispositivos IMU's, generar y controlar las señales de actuación del prototipo y emular el movimiento del mismo con el uso de fuentes computacionales como MatLab/Simulink que convierten este proyecto en una iniciativa al estudio, diseño e innovaciones de robots paralelos en la Universidad Industrial de Santander.

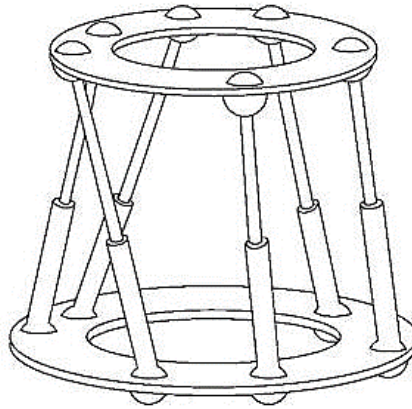
## 1. PLANTEAMIENTO DEL PROBLEMA

La robótica se encuentra en la vida cotidiana para brindar solución a una variada lista de dificultades presentes en el trabajo del hombre, brindando alta precisión, seguridad, eficiencia, alta calidad de vida y productividad. La robótica en la actualidad hace parte de la gran mayoría de las actividades del hombre, desde entornos peligrosos, entornos difíciles de alcanzar por el ser humano o donde se necesita extrema precisión.

El estudiante de ingeniería mecánica de la Universidad Industrial de Santander necesita tener siempre una transición de lo teórico a lo pragmático, construyendo y aplicando conocimiento, aumentando la calidad técnica y científica del ingeniero y la capacidad de uso adecuado de los recursos, en pro de plantear soluciones óptimas a las necesidades de la sociedad. Es por eso que la escuela de ingeniería mecánica debe enfocarse en proveer conocimientos prácticos que sean dominantes en los ambientes industriales actuales. Nace de esta situación en el laboratorio de Automatización Industrial la necesidad de construir un prototipo de plataforma Stewart como banco de estudio en robótica paralela, que se establezca como plataforma de aprendizaje en la línea de sistemas mecatrónicos y control de sistemas, aplicando el manejo, control y simulación de robots paralelos utilizados en gran variedad de procesos de ambientes industriales actuales.

Actualmente los robots se dividen en dos grandes grupos, los robots en serie y los robots paralelos, estos últimos hoy en día no se han desarrollado en la misma magnitud que los primeros, por ello son aún tema de investigación. Los robots paralelos son estructuras cerradas que permiten levantar cargas considerables debido a que sus actuadores están distribuidos en forma paralela y permiten distribuir la carga en cada uno de ellos tal como se ilustra en la Figura 1, además de lo anterior, logran alcanzar altas velocidades y aceleraciones.

**Figura 1.** Robot de estructura paralela, Plataforma Stewart.



Fuente: <http://sade.disam.etsii.upm.es/trepa/es/project.asp>

### **1.1 JUSTIFICACIÓN DEL PROBLEMA**

Planteada la necesidad de desarrollar un banco de laboratorio, que lleve al estudiante de ingeniería mecánica a practicar e implementar control de alto nivel sobre un prototipo de plataforma Stewart, se inició como proyecto de grado, el diseño y construcción de la misma por parte de estudiantes de ingeniería mecánica, dado que se posee la capacidad de diseñarla y los conocimientos teóricos de control en sistemas de lazo cerrado.

El diseño y construcción de la plataforma redujo en gran manera los costos del prototipo relativo a la compra externa del mismo. Para ello se tuvieron en cuenta métodos ingenieriles aplicados a diseño de sistemas de control, buscando confiabilidad y precisión en pro de optimizar el proceso.

Este proyecto llevó a los estudiantes de ingeniería mecánica de lo teórico a lo pragmático, construyendo y aplicando conocimiento, aumentando la calidad técnica y científica del ingeniero y la capacidad de uso adecuado de los recursos en pro de plantear soluciones óptimas a las necesidades de la sociedad. Lo anterior se logrará

por medio del uso del prototipo como banco de estudio en robótica paralela, estableciendo el prototipo como plataforma de aprendizaje en la línea de sistemas mecatrónicos. Por otro lado, este proyecto hace parte de la visión de construir un simulador de vuelo y aplicaciones afines por parte del laboratorio de Automatización Industrial.

## **1.2 OBJETIVO**

**1.2.1 Objetivo General.** Contribuir a la formación integral de los Ingenieros Mecánicos de la Universidad Industrial de Santander, aportándoles alta calidad técnica y científica en áreas de robótica, control de sistemas y automatización de ambientes industriales, brindando al estudiante la posibilidad de experimentar teorías de control de alto nivel a través del manejo de un prototipo de robot paralelo.

### **1.2.2 Objetivos Específicos.**

- Diseñar y construir un prototipo de plataforma Stewart hexápoda, actuada por servomotores, capaz de realizar traslaciones en los ejes X, Y y Z menores a 3 [cm], con un volumen de trabajo máximo de 30x30x25 [cm<sup>3</sup>].
- Desarrollar una interfaz en el ambiente de programación gráfico Simulink perteneciente a MatLab, a través de la librería SimMechanics de segunda generación, capaz de simular y generar la señal para la actuación de los servomotores proporcionando el control a la plataforma móvil del prototipo en lazo cerrado haciendo uso de Arduino como tarjeta de adquisición de datos.
- Implementar el control físico y mando del prototipo por medio de la señal generada por un dispositivo IMU (acelerómetro y giroscopio).

## 1.3 ALTERNATIVAS DE DISEÑO

### 1.3.1 Según el tipo de grados de libertad

- **Alternativa 1: Robot Delta (3 GDL)**

Es un tipo de robot paralelo destacado por su velocidad y precisión, muy utilizado en envasadoras. Por la cantidad y disposición de sus brazos permite únicamente tres grados de libertad limitando su funcionalidad en procesos industriales en los que se necesita flexibilidad en sus tareas.

**Figura 2.** Robot delta marca FESTO



Fuente: [http://www.festo.com/cat/es-co\\_co/products\\_EXPT](http://www.festo.com/cat/es-co_co/products_EXPT)

- **Alternativa 2: Plataforma Stewart**

Es un robot de tipo paralelo con 6 grados de libertad, utilizado para simular movimientos de sistemas mecánicos tales como autos, aviones, motocicletas entre otros. Su actuación puede darse con actuadores lineales o servomotores, los cuales van montados sobre una plataforma base y transmiten su movimiento a través de cuerpos de dos fuerzas a la plataforma móvil que posee seis grados de libertad tal como se ilustra en la Figura 3.

**Figura 3.** Plataforma Stewart



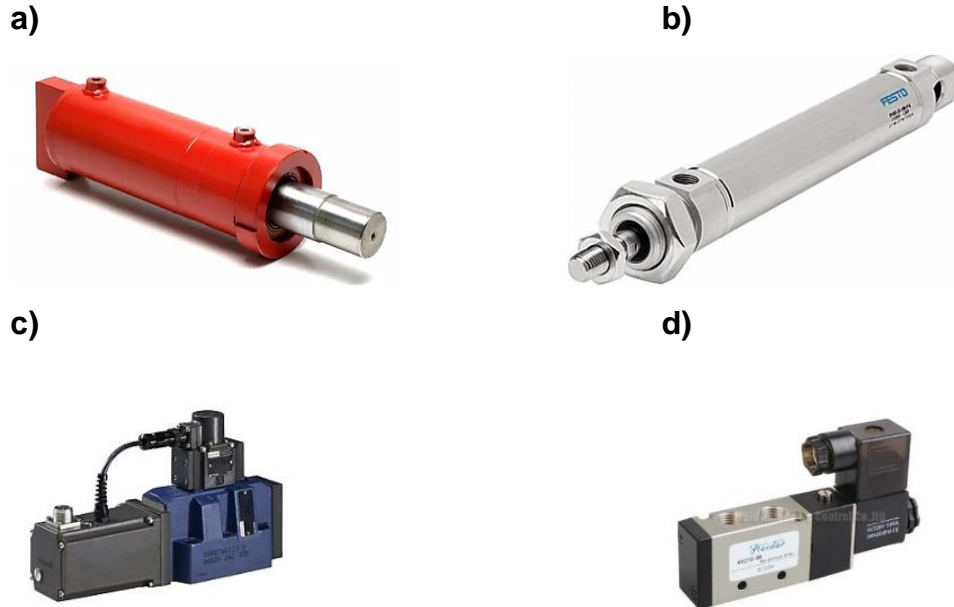
Fuente: <http://es.slideshare.net/htrmoreno/robots-parallel>

### 1.3.2 Según el tipo de actuadores

- **Alternativa 1: Actuadores Lineales Hidráulicos o Neumáticos.**

A pesar de ser más precisos que los servomotores debido al modo de actuación y tener la posibilidad de trabajar con mayores cargas, estos actuadores son más costosos, más robustos y por consiguiente ocupan mayor espacio. Además al poseer un sistema de potencia fluida se hace necesaria la instalación de una bomba o compresor que impulse el fluido según sea el caso, aumentando los costos no solo de construcción sino mantenimiento del sistema donde se encuentren instalados.

**Figura 4.** Sistemas hidráulicos y neumáticos: a) c) cilindro y servoválvula hidráulica  
b) d) Cilindro y válvula direccional neumática.



Fuente: [http://www.unimil.com.br/u/es/?page\\_id=13](http://www.unimil.com.br/u/es/?page_id=13)

#### ▪ Alternativa 2: Servomotores

Dispositivo actuador compuesto por un motor de corriente continua, una caja reductora y un circuito de control. El servomotor permite el desplazamiento de los brazos a través de la rotación de su eje de 0 a 180° con la capacidad de ubicarse en cualquier punto y con alto par. El funcionamiento de estos dispositivos es sencillo pues hacen uso de la modulación por ancho de pulsos PWM para controlar la dirección o posición de los motores. Son económicos y de fácil adquisición.

**Figura 5.** Servomotor HITEC



Fuente:

[www.dynamoelectronics.com/components/com\\_virtuemart/shop\\_image/product/Servo\\_HS\\_755HB\\_H\\_4e1f66b3b74ef.png](http://www.dynamoelectronics.com/components/com_virtuemart/shop_image/product/Servo_HS_755HB_H_4e1f66b3b74ef.png)

### 1.3.3 Según el tipo de unión

- **Alternativa 1: Juntas Esféricas**

Son el componente que brinda al sistema la movilidad fluida de un robot, permitiendo la rotación de los brazos sin restricciones con respecto a la plataforma y la transmisión de movimientos de traslación. Deben ser capaces de soportar el peso de la plataforma y la carga actuando como puntos de pivote.

**Figura 6.** Junta Esférica

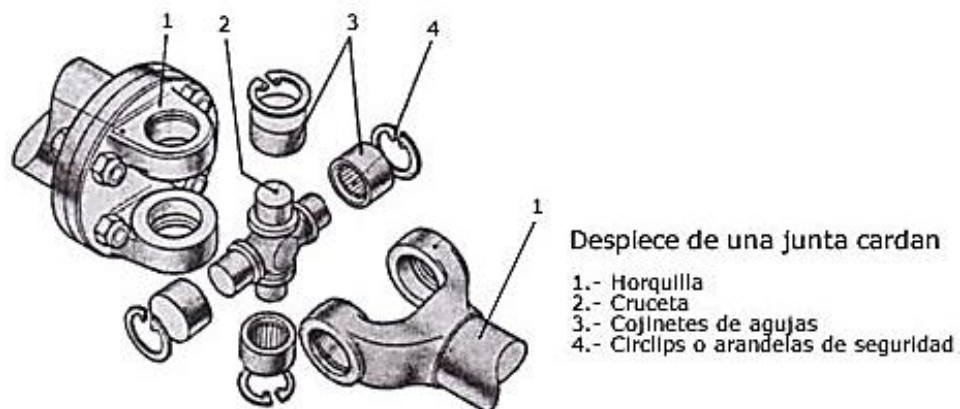


Fuente: <http://stidexkart.com/30-varillas-y-rotulas-de-direccion>

- **Alternativa 2: Juntas Universales o Cardánicas**

Son un par cinemático con dos grados de libertad (o 3 según el montaje). El objetivo de las juntas universales es transmitir el movimiento de rotación de un eje al otro a pesar de la no colinealidad. Las juntas universales pueden transmitir un gran par motor y permiten desplazamientos angulares de hasta 15° en las de construcción normal, llegando a un límite de hasta los 25° en las de construcción especial.

**Figura 7.** Junta Universal Cardánica



Fuente: <http://www.aficionadosalamecanica.com/transmisiones.htm>

### 1.3.4 Según la adquisición de datos

- **Alternativa 1: Tarjeta Arduino UNO**

Es una plataforma electrónica abierta que permite adquirir datos a través de sus pines de entrada de toda una gama de sensores y generar señales controladas para la actuación de motores, relés, etc. La tarjeta cuenta con 14 pines digitales que pueden ser utilizados como entradas o salidas, 6 entradas analógicas, 6 pines PWM y con una alimentación de 5V.

**Figura 8.** Tarjeta Arduino UNO



Fuente: [http://arduino.cc/en/uploads/Main/ArduinoUno\\_R3\\_Front.jpg](http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg)

▪ **Alternativa 2: Field Point (National Instruments)**

Es accesible para uso de estudiantes, pero lo suficientemente poderoso para aplicaciones de medida más sofisticadas<sup>1</sup>. Dispone de 8 entradas analógicas, 2 salidas analógica, 12 E/S digitales.

**Figura 9.** Dispositivo DAQ Multifunción



Fuente: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/201986>

---

<sup>1</sup> Tomado de: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/201986>

### 1.3.5 Según el Software de aplicación

- **Alternativa 1: MatLab/Simulink**

MATLAB<sup>®</sup> es un lenguaje de alto nivel, un entorno interactivo para el cálculo numérico, visualización y programación. Usando MATLAB se puede analizar datos, desarrollar algoritmos, crear modelos, aplicaciones que incluyan el procesamiento de señales y comunicaciones, procesamiento de imágenes y de vídeo, sistemas de control, prueba y medición.

Además de las bondades de MatLab antes mencionadas, también posee un entorno de programación visual llamado Simulink<sup>2</sup>, que funciona sobre el entorno de programación MatLab, Simulink es una herramienta de modelos o sistemas, con cierto grado de abstracción de los fenómenos físicos involucrados en los mismos y es empleado en procesamiento digital de señales, ingeniería de control y robótica debido a su facilidad de comunicación con hardware y software interno, sin contar la posibilidad que el usuario posee de crear sus propias librerías que se ajusten a las necesidades particulares del mismo.

- **Alternativa 2: LabVIEW<sup>3</sup>**

Es una plataforma de programación gráfica que ayuda a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. El software es ideal para cualquier sistema de medidas y control., integra todas las herramientas que los ingenieros y científicos necesitan para construir una amplia variedad de aplicaciones.

Esta plataforma de programación es bastante versátil a la hora de comunicar, procesar y analizar datos, permitiendo interfazarla con gran cantidad de tarjetas de

---

<sup>2</sup> Simulink, Wikipedia. [En línea] Disponible en: <http://es.wikipedia.org/wiki/Simulink>

<sup>3</sup> LabVIEW. *National Instruments*. [En línea] Disponible en: <http://www.ni.com/labview/esa>

adquisición de datos haciendo uso de librerías específicas para cada tipo de sensor y tarjeta.

- **Alternativa 3: Arduino**

Además de las características expuestas anteriormente, la tarjeta Arduino es programada en lenguaje basado en C++, es decir, programación por código e instrucciones que permite hacer uso las características de la tarjeta y además incluir librerías específicas para miles de dispositivos de medición y actuación, entre otros. Este lenguaje de programación hace muy tentadora la opción de utilizar la tarjeta Arduino también como tarjeta para control de los dispositivos acoplados a la misma.

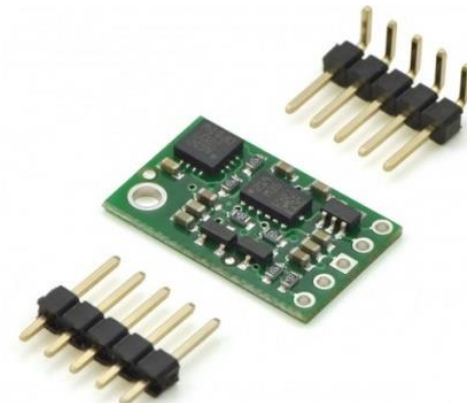
**1.3.6 Según la señal de mando.** Ejercer control sobre la plataforma Stewart requiere de un mando que el usuario pueda utilizar a su antojo, por ello se analizan a continuación diferentes dispositivos y posibilidades de mando para la plataforma.

- **Alternativa 1: Dispositivo IMU**

Unidad de medición inercial, estos dispositivos son conocidos como IMU por sus siglas en inglés, este dispositivo electrónico es capaz de informar acerca de la velocidad angular y fuerzas gravitacionales relativas del lugar donde se encuentre instalado, valiéndose de dispositivos como acelerómetros y giróscopos. Los datos recolectados por el dispositivo IMU permiten calcular la aceleración, velocidad y posición del dispositivo relativos a la posición inicial por medio del histórico de datos recolectados y comunicar esta información por medio de voltajes a un controlador o tarjeta de adquisición de datos. Básicamente el dispositivo IMU funciona detectando la actual tasa de aceleración usando el acelerómetro y los cambios rotacionales

usando el gir6scopo, convirti6ndose en un dispositivo ambiguo que puede ser utilizado como sensor o como mando seg6n las necesidades.

**Figura 10.** IMU 9DOF V2 (Aceler6metro, Magnet6metro y Giroscopio)



Fuente: Dynamo Electronics

- **Alternativa 2: Simulador**

Los simuladores son sistemas que intentan replicar o simular la experiencia de manejar un sistema mec6nico tal como automotores o aeronaves, entre otros. El objetivo principal de un simulador es emular caracter6sticas espec6ficas de los veh6culos, de la manera m6s realista y precisa posible. El alcance en la simulaci6n llega hasta rayar en la realidad, con sistemas complejos cuyas simulaciones van hasta los 6 grados de libertad del veh6culo, pasando desde 6nicamente simulaci6n de imagen hasta simulaci6n f6sica del movimiento montando cabinas a escala real sobre plataformas Stewart cuyo mando es generado por el simulador.

**Figura 11.** Simulación de despegue Aeronáutico

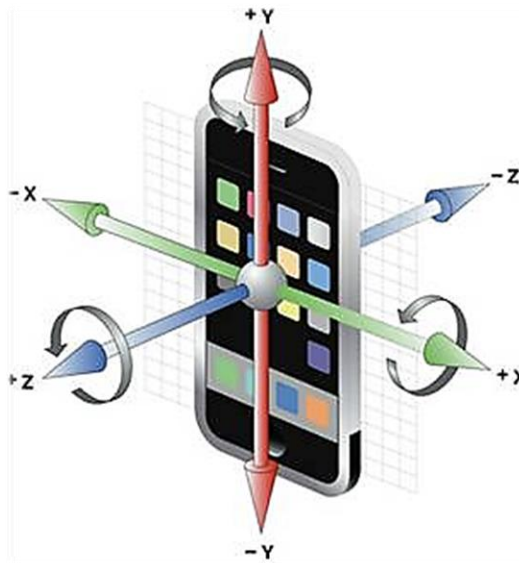


Fuente: <http://simuladordevuelos.com/simulador-de-vuelo-online/>

▪ **Alternativa 3: Aplicación Móvil**

El auge de los teléfonos inteligentes o *Smartphones* ha marcado el uso de estos dispositivos en la cotidianidad del ser humano. Existen sistemas operativos como Android que permiten a cualquier usuario diseñar y crear aplicaciones que utilizan la gran mayoría de hardware que se encuentra en el dispositivo, dando libertad a que novatos se dediquen a producir aplicaciones capaces de enviar señales vía inalámbrica a otro dispositivo cercano. Debido que gran porcentaje de las personas poseen un Smartphone y estos poseen en su gran mayoría acelerómetros se hace atractivo crear una aplicación para dispositivos Android que recolecte los datos del acelerómetro y giroscopio del teléfono y los envíe vía inalámbrica a un receptor que haga uso de ellos para controlar los movimientos de la plataforma.

**Figura 12.** Ejes coordenados de los sensores de un móvil.



Fuente: <http://img.genbetadev.com/2012/01/Ejes-acelerometro-iPhone.jpg>

**1.3.7 Según la sensorica.** El sensor ubicado en la plataforma móvil tiene como función medir las posiciones, aceleraciones y rotaciones en tiempo real para garantizar el control en lazo cerrado de la plataforma, a continuación se expondrán algunos de los sensores más utilizados en la medición de estos parámetros.

- **Alternativa 1: Acelerómetro**

Realizan una medida de aceleración o vibración, proporcionando una señal eléctrica según la variación física, en este caso la variación física es la aceleración o la vibración. Los acelerómetros en el mercado varían de acuerdo al número de ejes de trabajo que van desde uno hasta tres ejes, al voltaje de alimentación y a la sensibilidad.

**Figura 13.** Acelerómetro de tres ejes



Fuente: <http://www.dynamoelectronics.com/mma7260.jpg>

- **Alternativa 2: Giroscopio**

Permite determinar el sentido de giro de la plataforma por medio de la medición de la velocidad angular de rotación de los tres ejes. Estos sensores por lo general son muy pequeños y manejan un voltaje de alimentación de 5V.

**Figura 14.** Giroscopio de tres ejes digital.



Fuente: [http://www.dynamoelectronics.com/Giroscopio\\_de\\_tr\\_50857fc6af52c.png](http://www.dynamoelectronics.com/Giroscopio_de_tr_50857fc6af52c.png)

**Alternativa 3: IMU**

Este dispositivo electrónico antes mencionado también puede ser utilizado como sensor de posiciones y aceleraciones, llegando hasta la medición de 6 grados de libertad, con el beneficio de combinar las mediciones de acelerómetros y giróscopos.

## 1.4 SELECCIÓN DE ALTERNATIVAS

Teniendo en cuenta las alternativas planteadas anteriormente se tomó como criterio para su selección las que reproduzcan óptima operación de la plataforma y bajos costos de construcción y tamaño. Las alternativas más convenientes a criterio de los diseñadores son:

**1.4.1 Prototipo.** La plataforma Stewart es un dispositivo completo ya que se pueden manipular los seis grados de libertad en la plataforma móvil contrastados con los tres grados de libertad del robot Delta. El poco desarrollo, las posibles aplicaciones de las plataformas Stewart, el control complejo del movimiento y seguimiento de su plataforma móvil hicieron más llamativa la investigación de estos robots paralelos.

**1.4.2 Actuación.** Debido a las características del prototipo, su reducido tamaño, su bajo costo, la rápida velocidad de respuesta de hasta  $0.16 [s/60^\circ]$  y la fácil manipulación por medio de tren de pulsos PWM, los servomotores se escogieron por encima de los actuadores lineales tanto neumáticos como hidráulicos.

**1.4.3 Sistema de Articulación.** Dado que las juntas esféricas se pueden montar directamente sobre la plataforma móvil sin necesidad de elementos adicionales como balineras en el caso de las juntas cardánicas, necesarias para lograr los tres grados de libertad, las primeras fueron seleccionadas para la construcción del prototipo.

**1.4.4 Adquisición de Datos.** Se implementó una tarjeta Arduino UNO como sistema DAQ pues es un dispositivo ampliamente conocido por los estudiantes de ingeniería debido a su manejable entorno de programación basado en lenguaje C, de bajo costo, tamaño reducido respecto a otras tarjetas de su misma clase, de código abierto con amplia capacidad de comunicación serial con MatLab/Simulink y dispositivos electrónicos IMU's por medio del bus I2C, además cuenta con los puertos PWM necesarios para la actuación de los servomotores.

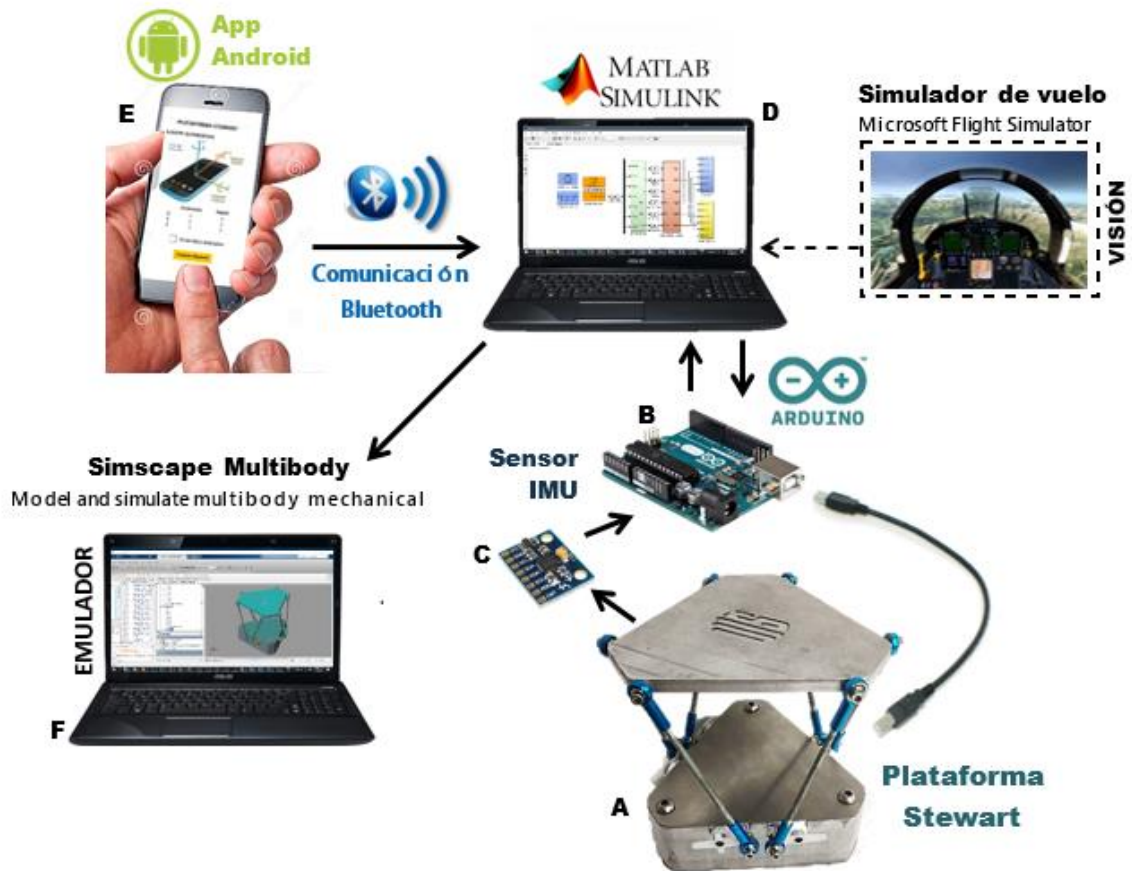
**1.4.5 Software de Aplicación.** Debido a que se esperaba que el software de aplicación y control de la plataforma fuese extenso se descartó la tarjeta Arduino para la aplicación del software debido a su baja capacidad de memoria y se usa únicamente como tarjeta de adquisición de datos. Dado que Simulink y LabVIEW poseen características similares en cuanto a la ejecución y control de sistemas reales se decidió usar Simulink ya que es un software utilizado con mayor frecuencia por estudiantes de ingeniería y además cuenta con el soporte para utilizar la tarjeta Arduino como tarjeta de adquisición.

**1.4.6 Señal de Mando.** Aunque las tres posibilidades de señal de mando son bastante prometedoras, optamos por la señal de mando captada por el dispositivo IMU del celular y enviada a través de una aplicación móvil, garantizando un mando de manipulación física por parte del usuario.

**1.4.7 Sensorica.** El dispositivo IMU ejerce una óptima medición de los fenómenos físicos del sistema dado que integra las funciones de un acelerómetro y un giroscopio, dotando la toma de datos de una precisión superior a la que se puede lograr con cada uno de estos dispositivos por separado.

## 2. PROTOTIPO DE PLATAFORMA STEWART SERVOACTUADA

Figura 15. Concepto general del proyecto plataforma Stewart.



El prototipo de plataforma Stewart diseñado en este proyecto e identificado en la Figura 15 con la letra (A) es actuado con seis servomotores<sup>4</sup> que le brindan libertad de movimiento a la plataforma móvil del robot en el espacio tridimensional. La plataforma móvil actúa de acuerdo a los requerimientos del usuario por medio de

<sup>4</sup> Los servomotores también conocidos como servos son dispositivos constituidos por motores de corriente continua con la capacidad de ubicarse en cualquier posición de un rango de acción (usualmente es de 0 a 180°) y mantenerse en ella.

una aplicación móvil ejecutada en un *smartphone* (E) que usa el dispositivo IMU<sup>5</sup> del celular como mando y envía las señales de posición angular del celular vía *bluetooth* a un entorno de programación en el *software* MatLab/Simulink (D) ejecutado en un computador personal; Simulink es el encargado de procesar las señales del dispositivo IMU, analizarlas y emitir las señales de control de posición de los servomotores a través de la tarjeta de adquisición de datos Arduino (B) para que la plataforma móvil del prototipo reproduzca la posición requerida en la aplicación móvil. El prototipo cuenta con un lazo cerrado de control de posición retroalimentado por un sensor IMU (C) que mide la posición angular de la plataforma móvil y envía las señales a través de Arduino a Simulink para ser comparada con la posición establecida en la aplicación móvil y controlar la posición de cada servomotor. Además Simulink ejecuta un emulador (F) del prototipo que permite visualizar en tiempo real el comportamiento del mismo de forma virtual.

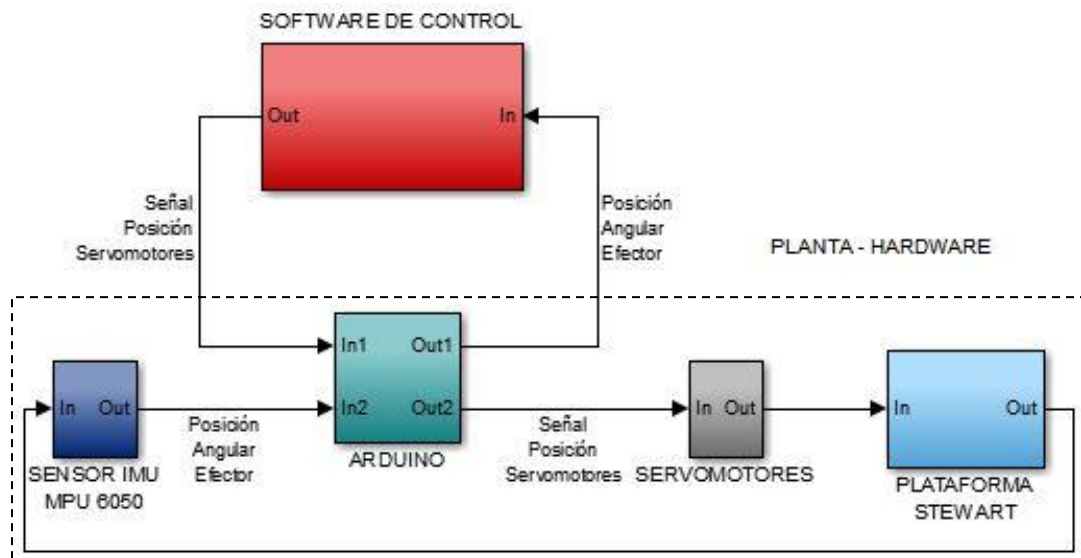
## 2.1 PLANTA

La planta está conformada por el *hardware* mecánico que incluye el prototipo de plataforma Stewart y los servomotores, y el *hardware* electrónico conformado por la tarjeta de adquisición de datos Arduino y el sensor IMU. La planta (*hardware* de este proyecto) interactúa con el *software* de control recibiendo las señales de posición de los servomotores y enviando las señales de posición angular del efector móvil de la plataforma mediante el enlace de comunicación serial con la tarjeta de adquisición Arduino, tal como se ilustra en la Figura 16.

---

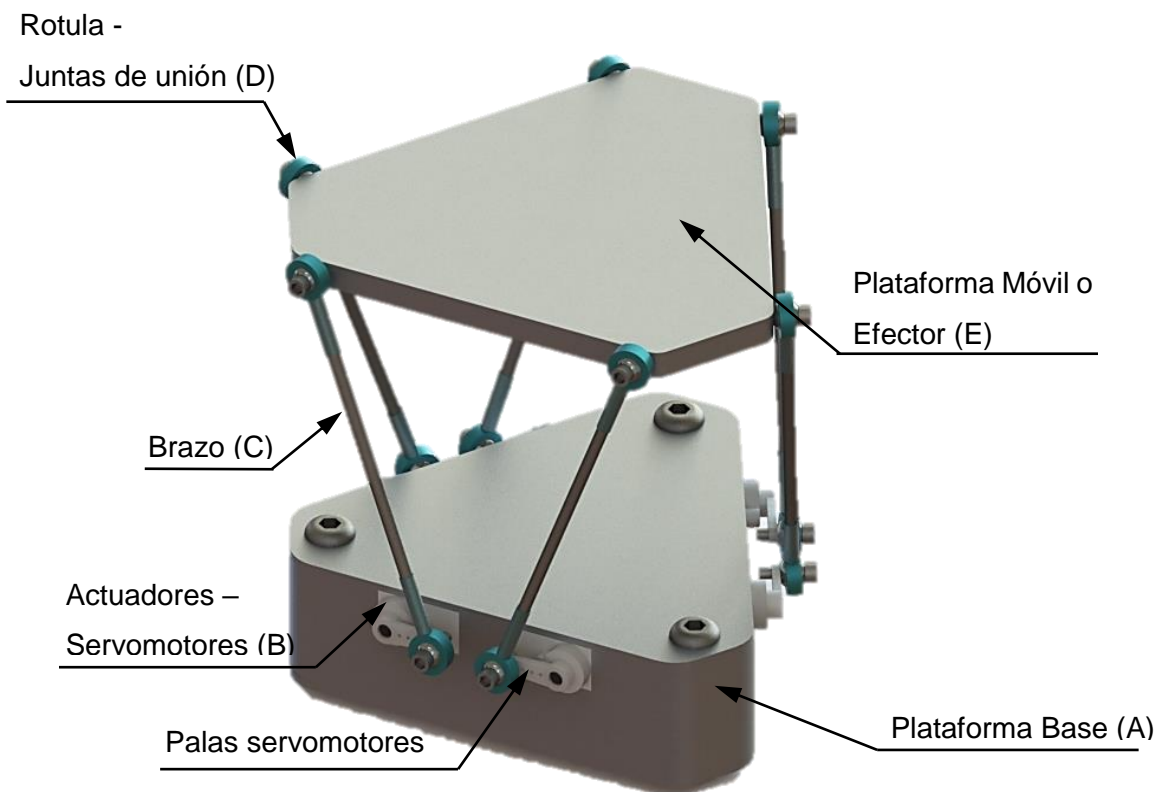
<sup>5</sup> Unidad de medición inercial, más conocida como IMU por sus siglas en Inglés, son sensores comúnmente constituidos por un acelerómetro y giroscopio que determinan la aceleración y la velocidad angular del dispositivo u objeto donde se encuentre instalado.

**Figura 16.** Comunicación entre la planta y el software de control



**2.1.1 Prototipo Plataforma Stewart.** El prototipo de plataforma Stewart brinda capacidad al usuario de analizar el comportamiento de los robots paralelos con seis grados de libertad en el espacio. El prototipo se ilustra en la Figura 17 y está compuesto básicamente por la plataforma base (A), los servomotores (B), los brazos del prototipo (C), las rotulas (D) y el efector móvil (E).

**Figura 17.** Modelo CAD<sup>6</sup> del prototipo diseñado y partes principales.



El diseño de la plataforma Stewart se basó en las recomendaciones del libro *Principles of Flight Simulation* escrito por Allerton<sup>7</sup>, en el cual el autor explica de manera específica el proceso de diseño para un robot paralelo de seis grados de libertad en el espacio que posea la capacidad de simular el movimiento de vuelo. Allerton brinda recomendaciones acerca de relaciones de dimensión entre componentes principales de la plataforma Stewart tales como la plataforma base, la plataforma móvil y los brazos actuadores, posición óptima de los actuadores,

---

<sup>6</sup> Diseño Asistido por Computadora, también conocido CAD por sus siglas en ingles. Consiste en el uso de software de diseño para la creación de representaciones gráficas de objetos físicos.

<sup>7</sup> ALLERTON D, *Principles of Flight Simulation*. 1st ED, Editorial Wiley & Sons, Ltd, United Kingdom, Chichester 2009.

además de brindar un análisis comparativo de las juntas de unión que pueden ser utilizadas en el robot, entre otras recomendaciones.

Debido a las dimensiones y el alcance de la plataforma se decidió instalar 6 mini servomotores analógicos en el prototipo, los cuales permiten realizar movimientos precisos de manera rápida y levantar altas cargas en comparación a su tamaño reducido, características que definen la calidad y fluidez del movimiento del prototipo, por otro lado, los servomotores también cuentan con un sistema de control interno que garantiza el posicionamiento de los mismos.

El diseño de la plataforma Stewart se optimizó haciendo uso del modelo CAD de la misma (Figura 1Figura 17) desarrollado en el programa SolidWorks<sup>8</sup>, en el cual se variaron características geométricas de la plataforma tales como la longitud de las palas de los servomotores, las posiciones de las juntas de unión de los brazos actuadores y las dimensiones de las plataformas base y móvil.

Teniendo definidos los componentes físicos del prototipo, se realizó el análisis matemático de la misma haciendo uso de los ángulos de Euler<sup>9</sup> utilizados en el cálculo de este tipo de robots y del análisis de movimiento de objetos en el espacio tridimensional mediante el uso de matrices de desplazamiento y rotación. Para la creación del modelo cinemático se definió el origen de coordenadas en el centroide de la plataforma base y se crearon las matrices de posición relativa de la plataforma en su estado inicial con respecto al origen antes mencionado, habiendo definido estas matrices por medio del uso de la cinemática inversa del robot se determinó las variaciones de posición de los componentes del prototipo únicamente con la posición final deseada de la plataforma móvil, determinando así la posición que

---

<sup>8</sup> Solidworks es un *software* de diseño gráfico computacional en tercera dimensión que permite crear, simular y manejar modelos.

<sup>9</sup> Los ángulos de Euler son tres ángulos introducidos por Leonhard Euler para describir la orientación de un objeto en relación a un origen de coordenadas establecidas.

debe tener cada servomotor en la plataforma con el fin de reproducir una posición del efector móvil.

**2.1.2 Tarjeta de Adquisición de Datos.** Como tarjeta de adquisición de datos y comunicación entre el *software* de control y el *hardware* mecánico del prototipo diseñado se utilizó una tarjeta Arduino UNO. La tarjeta Arduino se encarga de recibir por comunicación serial las señales de posición angular de los servomotores que provienen de Simulink y enviarlas a los servomotores del prototipo conectados a sus puertos PWM, por otro lado, también se encarga de leer las señales de aceleración y velocidad angular del IMU instalado en el efector móvil de la plataforma y enviarlas a Simulink.

**2.1.3 Sensor IMU.** Se utilizó un sensor MPU 6050 del fabricante Invensense que cuenta con un acelerómetro y giroscopio capaces de leer las aceleraciones y velocidades de giro en las tres dimensiones a las que está siendo sometido el sensor. El sensor fue instalado en el centroide del efector móvil de la plataforma con el fin de determinar la posición angular del mismo y ser el dispositivo de retroalimentación que cierra el lazo de control del prototipo enviando las señales a Simulink por medio de la tarjeta Arduino.

## 2.2 CONTROL DEL PROTOTIPO

Debido a que el comportamiento dinámico de los robots paralelos se analiza organizando matrices<sup>10</sup> de posición, el *software* de control de la plataforma se

---

<sup>10</sup> Matriz es un arreglo de expresiones de posición en este caso, organizadas en filas y columnas según un orden específico.

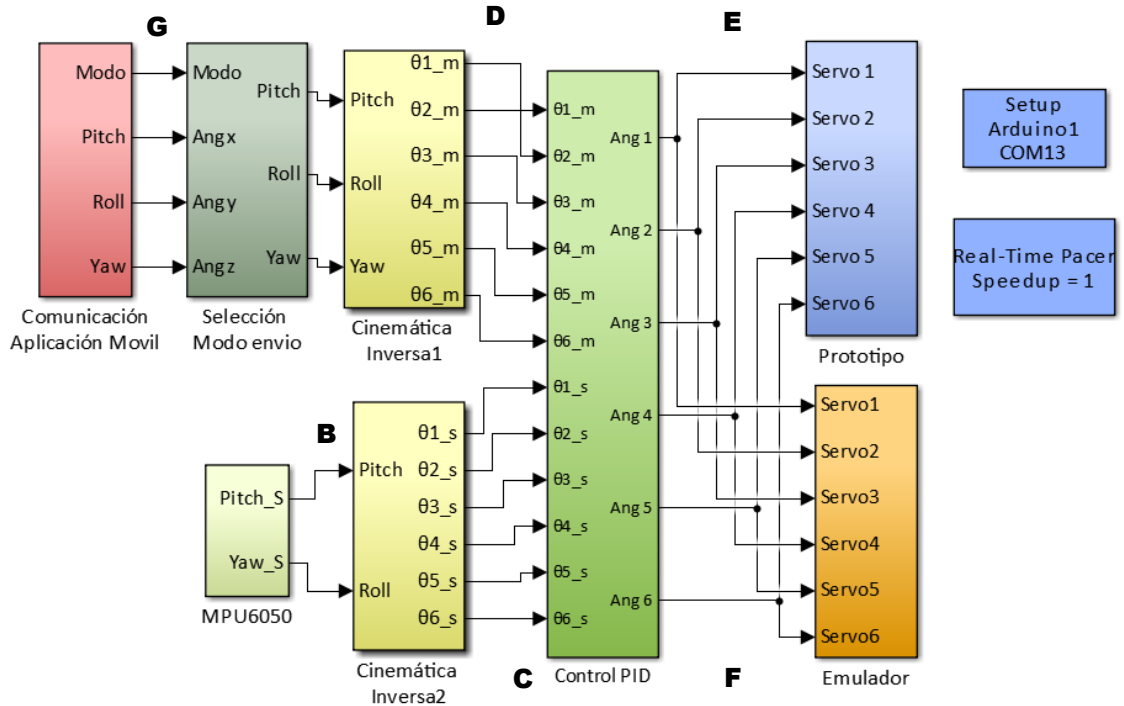
desarrolló en MatLab/Simulink<sup>11</sup> con el fin de aprovechar la potencia y capacidad de análisis de este software en la solución de matrices de grado complejo producto de la cantidad de información de posición de la plataforma Stewart definida en sus seis grados de libertad. Además Simulink hace posible la comunicación directa con dispositivos *bluetooth* y tarjetas de adquisición como Arduino mediante el uso de programas desarrollados para múltiples fines. Por todo lo anterior, se determinó realizar el *software* de control del prototipo en Matlab utilizando el complemento Simulink debido a que brinda un lenguaje gráfico e intuitivo para los estudiantes de pregrado, en comparación a la programación codificada del ambiente Matlab.

El programa de control desarrollado en Simulink e ilustrado en la Figura 18, se encarga de recibir las señales de posición angular enviadas por la aplicación móvil, determinar el modo de operación de la plataforma (automático o manual), determinar la posición que debe configurarse en cada uno de los seis servomotores de la plataforma para reproducir la señal de mando, enviar las señales de posición de los servomotores a Arduino, recibir las señales de aceleración y velocidad de giro enviadas por Arduino leídas del IMU y con ellas hacer la retroalimentación para controlar la posición de los servomotores, enviar la señal de control de posición de los servomotores a Arduino y por último alimentar el emulador de movimiento de la plataforma.

---

<sup>11</sup> Simulink es un ambiente de programación gráfico para modelar, simular y analizar sistemas dinámicos.

**Figura 18.** Esquema Software de control



En la Figura 18 el programa de control encargado de asegurar que el prototipo adquiera la posición que se desea utiliza las señales (A) del IMU instalado sobre la plataforma móvil en la planta y calcula la posición angular (B) de la misma, con dicha posición genera la posición que poseen los servomotores (C) y realiza realimentación negativa con la posición de los servomotores determinada con la señal de mando de la aplicación móvil (D), la señal resultado de la realimentación negativa entra a bloques de control de posición PID de cada servomotor en el bloque de control y generan la señal controlada de posición de los servomotores que se alimenta a la planta (E) y al emulador (F), garantizando exactitud en el posicionamiento de la plataforma móvil.

## 2.3 INTERFAZ DE USUARIO

La Interfaz de usuario de este proyecto está conformada por la aplicación móvil utilizada como mando instalada en el *smartphone* que envía las señales de posición angular del celular al software de control Simulink (Ver Figura 15 (E)) y la interfaz visual del emulador de movimientos de la plataforma que recibe las señales de posición de cada servomotor como se ilustró en la Figura 15 (F).

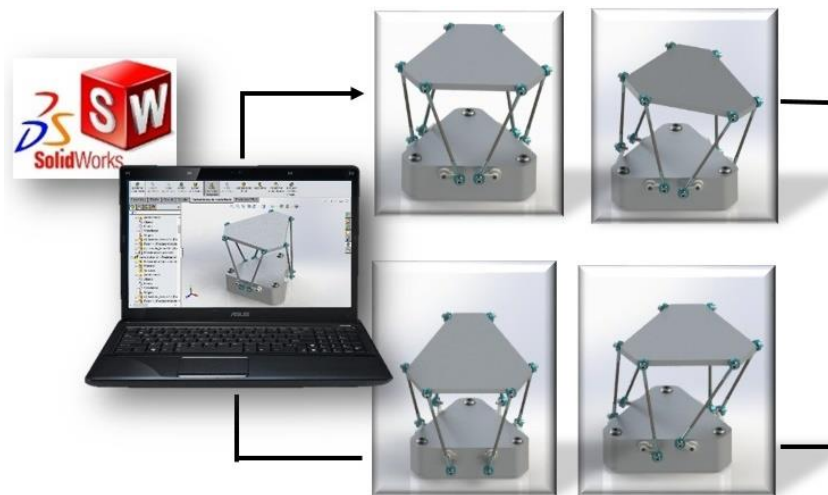
**2.3.1 Aplicación Móvil.** Aprovechando la facilidad de programación de aplicaciones móviles en el sistema operativo Android y toda la utilidad de los *Smartphone*, se creó una aplicación llamada Plataforma Stewart (Ver Figura 19) en el entorno online App Inventor, por medio de la cual se establece una comunicación vía *Bluetooth* con Matlab/Simulink que transmite la información mediante dos diferentes modos, un modo automático que toma la posición angular de los sensores de inclinación del celular en cada uno de sus ejes y un modo manual que envía rotaciones angulares específicas de la plataforma móvil ingresadas de forma manual.

**Figura 19.** Desarrollo de la App Android en App Inventor.



**2.3.2 Emulador.** El emulador de movimiento de la plataforma Stewart es una interfaz virtual que permite visualizar las respuestas dinámicas del prototipo como se ilustra en la Figura 20. Haciendo uso del complemento Simmechanics<sup>12</sup> en Simulink se alimentó el modelo CAD antes mencionado con las señales que alimentan el prototipo (Ver Figura 18 (F)), teniendo como resultado el movimiento virtual en tiempo real de la plataforma Stewart en el ambiente MatLab.

**Figura 20.** Emulador de movimiento de la plataforma



Un resultado de notar en este proyecto es la investigación de los principios de actuación de una plataforma Stewart, es decir, el análisis cinemático del movimiento y respuestas a señales continuas e intermitentes. Para finalizar, este proyecto es la fase inicial para la construcción de un simulador de vuelo, aportando para el mismo el estudio cinemático de una plataforma con seis grados de libertad y una alternativa a la comunicación entre un ambiente gráfico virtual como lo es el emulador y la plataforma Stewart.

<sup>12</sup> Simmechanics es un complemento de Simulink que proporciona un ambiente de simulación de movimiento de múltiples cuerpos en tercera dimensión pertenecientes a sistemas mecánicos.

### 3. DISEÑO MECÁNICO Y CONSTRUCCIÓN DEL PROTOTIPO DE PLATAFORMA STEWART

La planta está compuesta por *hardware* mecánico que es cada uno de los elementos físicos del sistema, es decir, cada una de las partes tangibles que en conjunto permiten el movimiento del prototipo de plataforma Stewart y el *hardware* electrónico que es la tarjeta de adquisición de datos y el sensor IMU.

#### 3.1 PROTOTIPO PLATAFORMA STEWART

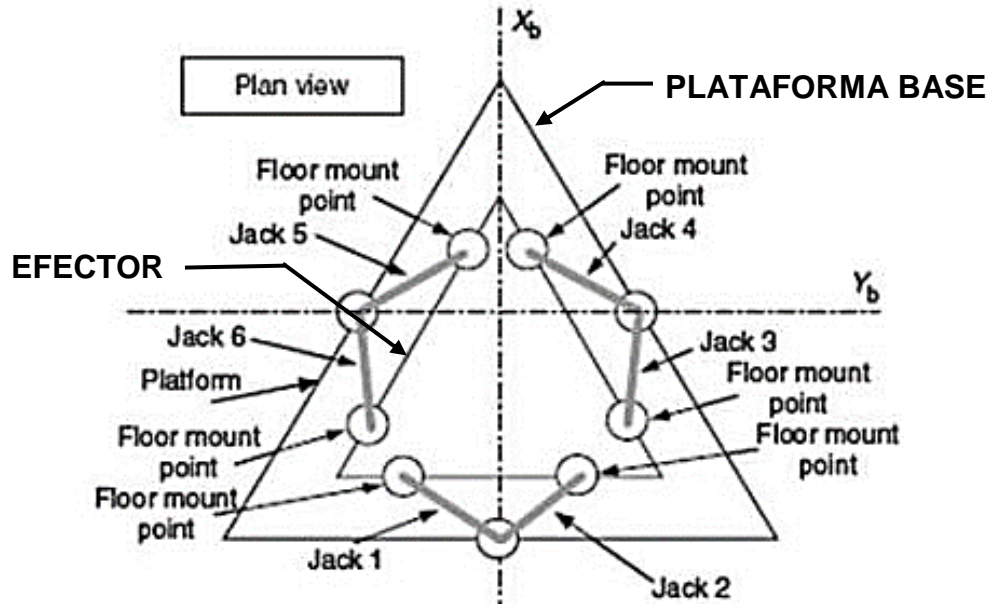
Para el diseño mecánico del prototipo de plataforma Stewart servoactuada se tomó en cuenta en primer lugar los aspectos relacionados en la Tabla 1.

**Tabla 1.** Características generales de la plataforma.

CARACTERÍSTICAS	DATO
Número de grados de libertad	6
Espacio de trabajo	30x30x25 [cm <sup>3</sup> ]
Tipo de uniones	Rótula
Tipo de actuadores	Servomotores
Carga máxima que soportará	4 [kg]

Una vez definidas las características de la plataforma, se eligió el tipo de configuración. Dado que en la actualidad existen gran variedad de plataformas Stewart se seleccionó una de las configuraciones más eficientes en cuanto a funcionalidad, dicha configuración es hexápoda (ver Figura 21) y consta básicamente de 2 triángulos equiláteros, uno denominado base y el otro efector móvil.

**Figura 21.** Configuración de la plataforma hexápoda vista en Planta

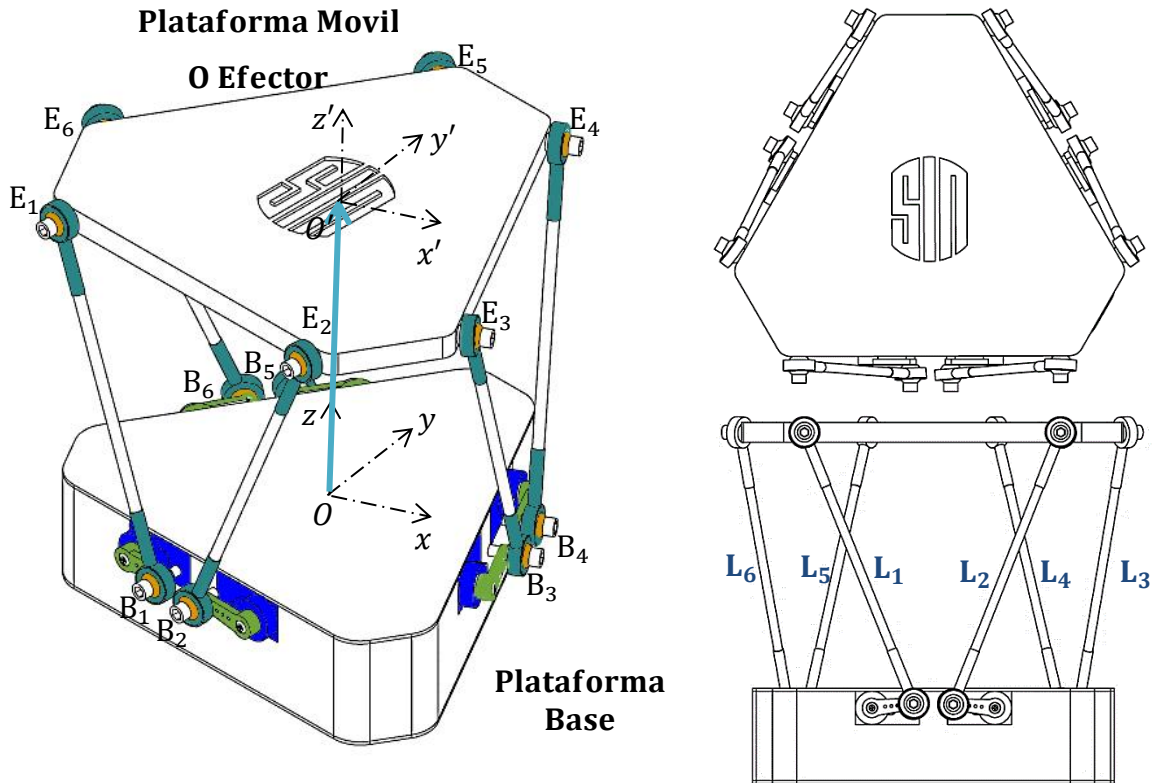


Fuente: D. Allerton, Principles of Flight Simulation<sup>13</sup>.

En la Figura 21 cada par de brazos inician en el mismo punto del triángulo base y terminan en vértices opuestos del efector sobre la misma cara superficial, formando en total 3 pares de puntos próximos a los vértices del mismo. Los puntos de unión entre los brazos del prototipo diseñado con la plataforma base y el efector se realizaron utilizando rótulas roscadas sobre los brazos y fijando la bola (identificada en color naranja) de la rótula al efector y a las palas de los servomotores tal como se ilustra en la Figura 22.

<sup>13</sup> *Op.cit.* ALLERTON D., Principles of Flight Simulation. 1<sup>st</sup> ed, Editorial Wiley & Sons, Ltd, United Kingdom, Chichester 2009

**Figura 22.** Modelo de plataforma Stewart Servoactuada diseñada.



En la Figura 22, cada una de las articulaciones rotuladas de la base están denominadas por la letra B, por ejemplo:  $B_1$  y las juntas del efector por la letra E, estos puntos se representan con el color naranjado. Los puntos de unión en la base, es decir, los puntos B serán muy cercanos entre sí, pero nunca partirán del mismo punto en el espacio, su posición está determinada por la rotación de la pala (elemento verde) del servomotor (elemento color azul) respectivo; para este caso de diseño todas las longitudes denominadas con la letra L, serán iguales. Por último se define el origen de cada plataforma con la letra O ubicado exactamente en el centroide de las mismas.

Antes de abarcar el diseño de la plataforma Stewart es importante verificar si es posible obtener los 6 grados de libertad en la plataforma según el modelo diseñado

con uniones esféricas o juntas de rótulas, para ello se utilizó el criterio de Grübler basado en la siguiente ecuación:

$$GDL = \lambda(n - j - 1) + \sum_{i=1}^j G_i - I_f$$

Donde:

- $\lambda$  = Grados de libertad del espacio donde operará el mecanismo.
- $n$  = Número de eslabones fijos del mecanismo (incluye las plataformas)
- $j$  = Número de juntas en el mecanismo.
- $G_i$  = Número de grados de libertad de la junta  $i$ .
- $I_f$  = Número de grados de libertad pasivos del mecanismo

Los valores definidos para el diseño del prototipo de plataforma Stewart en este proyecto son los siguientes:

$$\lambda = 6 \text{ (Espacio tridimensional)}$$

$$n = 6 \text{ Brazos} + 6 \text{ Palas} + 1 \text{ base} + 1 \text{ efector} = \mathbf{14}$$

$$j = 12 \text{ uniones esféricas} + 6 \text{ uniones de revolución} = \mathbf{18}$$

$$G_i = \mathbf{3} \text{ (juntas esféricas), } \mathbf{1} \text{ (uniones de revolución)}$$

$$I_f = \mathbf{6}$$

Sustituyendo los valores en la ecuación de Grübler tenemos:

$$GDL = 6(14 - 18 - 1) + \sum_{i=1}^{18} G_i - 6 = -30 + 12(3) + 6(1) - 6$$

$$\mathbf{GDL = 6}$$

Según los resultados de la ecuación de Grübler con juntas esféricas tanto en la plataforma base como en la plataforma móvil es posible obtener los 6 grados de libertad en el prototipo.

### 3.1.1 COMPONENTES DEL PROTOTIPO

El prototipo de plataforma Stewart está compuesto principalmente por la plataforma base, la plataforma móvil, los elementos de unión entre las dos plataformas y los actuadores.

**3.1.1.1 Actuación.** La actuación del prototipo de plataforma Stewart se realiza con 6 mini servomotores analógicos de referencia HK15178 como el que se muestra en la Figura 23. Estos servomotores fueron escogidos debido su excelente relación de carga y velocidad con respecto al tamaño reducido disponible en el mercado. Cada servomotor posee las características consignadas en la Tabla 2.

**Figura 23.** Servomotor Analógico HK15178.



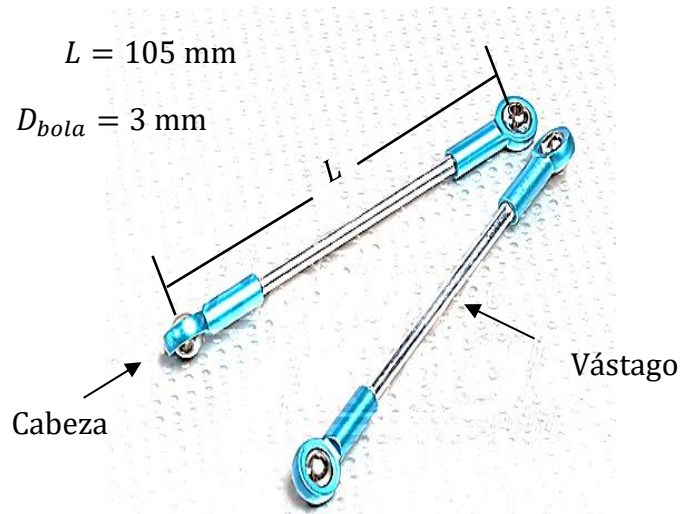
Fuente: [https://hobbyking.com/de\\_de/hobbykingtm-hk15178-analog-servo-1-4kg-0-09sec-10g.html](https://hobbyking.com/de_de/hobbykingtm-hk15178-analog-servo-1-4kg-0-09sec-10g.html)

**Tabla 2.** Características de los miniservos HK15178.

Largo:	23 [mm]	Ancho:	12 [mm]
Alto:	30 [mm]	Peso:	10 [gr]
Velocidad:	0.09 [sec/60deg]	Torque:	1.4 [kg/cm]

**3.1.1.2 Elementos de Unión.** Como elementos de unión entre la plataforma móvil o efector y la plataforma base se utilizaron barras en cuyos extremos están roscadas rótulas tal como se representa en la Figura 24. A dichos elementos se les llama brazos de la plataforma.

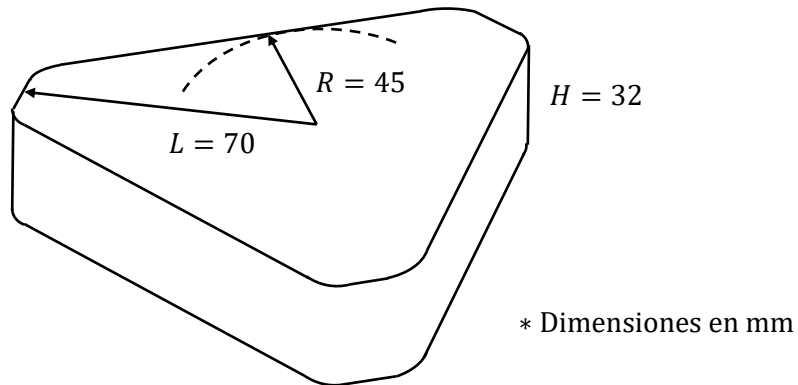
**Figura 24.** Elementos de unión tipo rótulas.



Fuente: [https://hobbyking.com/en\\_us/heavy-duty-push-rods-with-ball-link-ends-m4x83mm-2pcs-bag.html](https://hobbyking.com/en_us/heavy-duty-push-rods-with-ball-link-ends-m4x83mm-2pcs-bag.html)

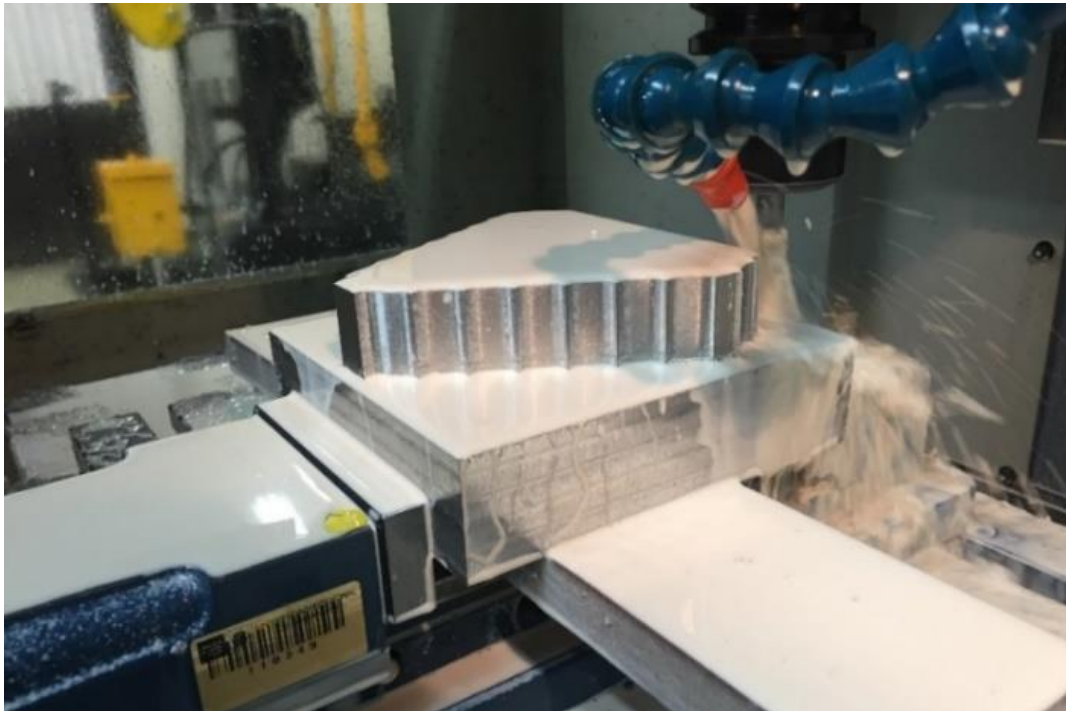
**3.1.1.3 Plataforma Base.** La plataforma base de acuerdo a la configuración hexápoda del prototipo es una base triangular con cada uno de sus vértices a 120 grados entre ellos. El diseño de la plataforma base se realizó creando un modelo en SolidWorks de la misma que guardase congruencia dimensional con los servomotores y los elementos de unión de la plataforma, resultando una plataforma base con las dimensiones expuestas en la Figura 25.

**Figura 25.** Dimensiones de la Plataforma Base.



La plataforma base fue fabricada a partir del mecanizado (ver Figura 26) de un bloque de aluminio en la máquina fresadora del Laboratorio CNC de la Universidad Industrial de Santander.

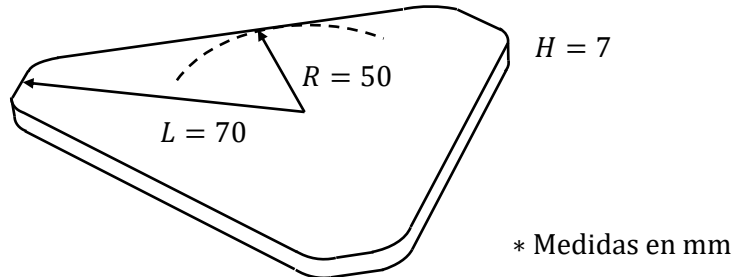
**Figura 26.** Mecanizado de la plataforma base.



En la fabricación de las tapas de la base se usaron dos chapas de acero inoxidable recortadas a la medida.

**3.1.1.4 Plataforma Móvil O Efecto.** La plataforma móvil del prototipo debe conservar características similares a la base con respecto al espaciamiento entre sus vértices y poseer los puntos de rotula en el mismo plano vertical que los poseen las palas de los servomotores, para cumplir dicho parámetro se utilizó el modelo CAD con el fin de determinar las dimensiones respectivas. Las dimensiones del efecto son las expuestas en la Figura 27.

**Figura 27.** Dimensiones de la Plataforma Móvil.



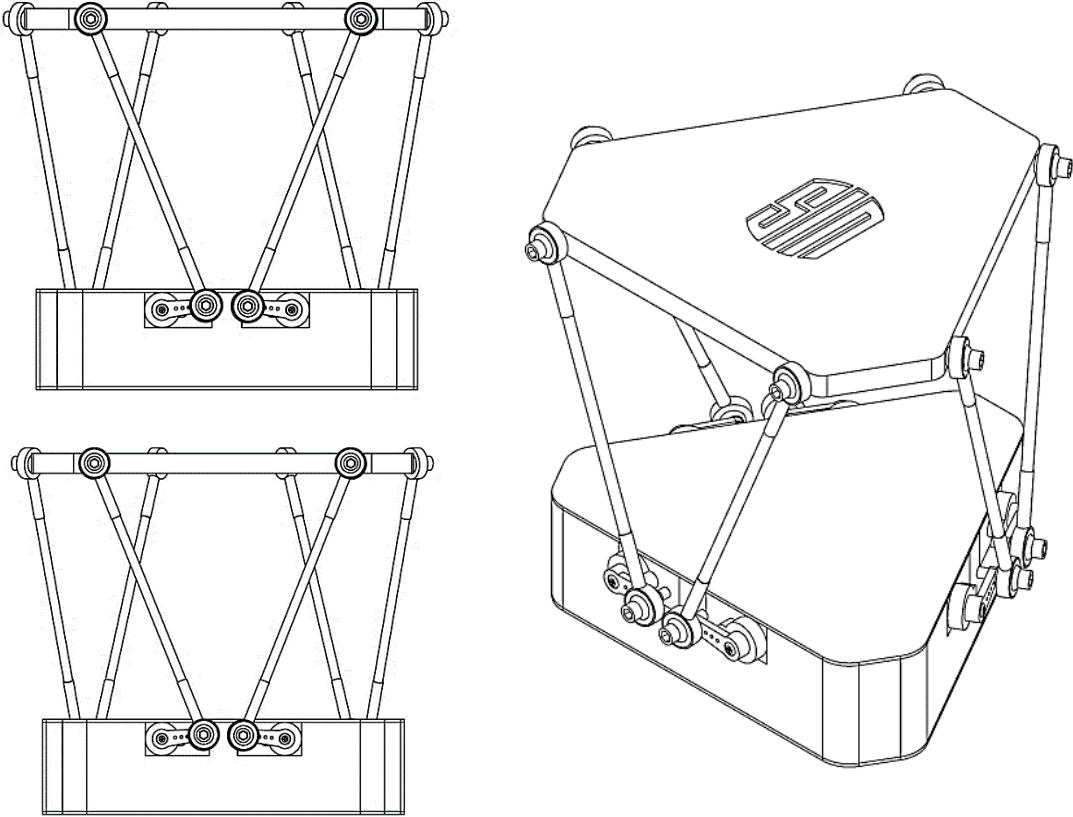
La plataforma móvil también fue fabricada a partir del mecanizado de un bloque de aluminio en la máquina fresadora del Laboratorio CNC y las operaciones de manufactura (ver Figura 28) en los talleres de la escuela de Ingeniería Mecánica de la Universidad Industrial de Santander.

**Figura 28.** Taladrado de la plataforma móvil.



**3.1.2 Modelado CAD del Prototipo.** Se creó un modelo del prototipo de plataforma Stewart a escala real en el programa SolidWorks con el propósito de optimizar el diseño de la misma. En la Figura 29 se aprecian las vistas frontal, superior e isométrica del modelo CAD. Los planos de diseño del modelo CAD se encuentran consignados en el Anexo B.

**Figura 29.** Modelo CAD de la plataforma Stewart.



### 3.1.3 Cinemática Inversa.

En comparación con los robots seriales, en los robots paralelos la cinemática inversa es menos compleja que la cinemática directa, por ende la solución de posicionamiento de esta plataforma se realizó por medio de cinemática inversa. La cinemática inversa de la plataforma Stewart actuada por cilindros hidráulicos o neumáticos consiste en determinar las longitudes de los actuadores para satisfacer una posición y orientación conocida del efector. Sin embargo para la plataforma Stewart servoactuada, la longitud de los actuadores es constante por ende se deben generar rotaciones en los servomotores que varíen las distancias entre la plataforma base y la plataforma móvil, es decir, la cinemática inversa para este modelo de plataforma determina los ángulos de los servomotores que satisfacen una posición y orientación del efector. El análisis de cinemática inversa se alimenta con dos variables de entrada configuradas por el usuario, correspondientes a los dos cambios de rotación (Pitch y Roll) y genera 6 salidas correspondientes a los ángulos de cada servomotor.

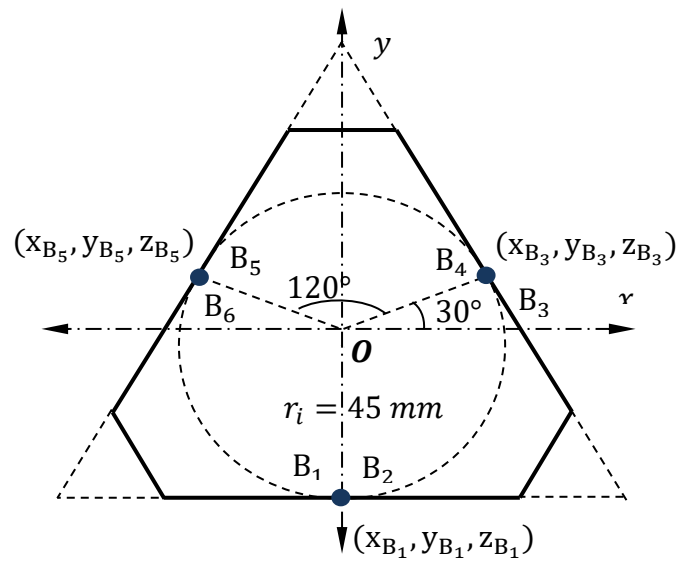
Para determinar la cinemática inversa de la plataforma móvil se deben conocer las coordenadas X, Y, Z iniciales de los puntos de unión tanto del efector como de la base, por tanto se determinaron dichos puntos en la plataforma de la siguiente manera:

**3.1.3.1 Coordenadas de la Base.** Según Allerton<sup>14</sup>, las uniones en la plataforma base deben encontrarse en el punto medio de los laterales de la misma, lo que significa que estarán espaciadas 120° las unas de las otras, tal como se ilustra en la **Figura 30**. Ubicación ideal de las juntas de la Plataforma Base. Figura 30.

---

<sup>14</sup> *Op. Cit.* ALLERTON D, Principles of Flight Simulation. 1st ED, Editorial Wiley & Sons, Ltd, United Kingdom, Chichester 2009.

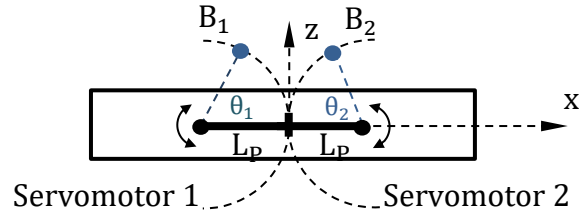
**Figura 30.** Ubicación ideal de las juntas de la Plataforma Base.



En la Figura 30 el origen cartesiano se encuentra en el centroide de la plataforma base, por ende cada punto de junta se expresa en coordenadas cartesianas a partir de dicho origen, a continuación se definirán las coordenadas cartesianas de todos los puntos de unión de la plataforma Stewart diseñada:

- **Puntos  $B_1$  y  $B_2$ .** En la Figura 31 se representa en vista frontal los servomotores 1 y 2 donde se encuentran los puntos  $B_1$  y  $B_2$  respectivamente, y las relaciones trigonométricas utilizadas para la definición de coordenadas de los mismos.

**Figura 31.** Representación vista frontal de los puntos  $B_1$  y  $B_2$



$$x_{B_1} = -16 + 16 \cos \theta_1 \text{ [mm]} \quad x_{B_2} = 16 - 16 \cos \theta_2 \text{ [mm]}$$

$$y_{B_1} = -45 \text{ [mm]} \quad y_{B_2} = -45 \text{ [mm]}$$

$$z_{B_1} = -7 + 16 \sin \theta_1 \text{ [mm]} \quad z_{B_2} = -7 + 16 \sin \theta_2 \text{ [mm]}$$

Sin embargo, las coordenadas reales de las juntas del prototipo fueron definidas a partir de los resultados de la simulación de la plataforma Stewart en el *software* Simulink haciendo uso de la librería SimMechanics sobre el modelo CAD, la creación de la simulación se explicará en el siguiente capítulo. El objetivo de la simulación del prototipo de plataforma Stewart fue modificar las coordenadas de cada una de las juntas tanto en la plataforma base y el efector con el fin de reproducir coordenadas cercanas a las ideales y obtener un máximo rango de movimiento en el espacio del efector móvil. Las coordenadas reales producto de la simulación antes mencionada son las siguientes:

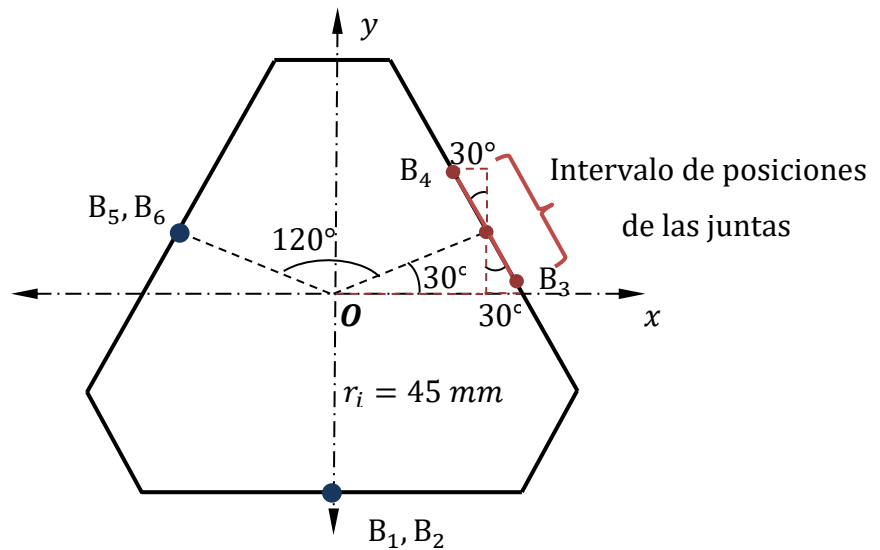
$$x_{B_1}' = -24.3 + 16 \cos \theta_1 \text{ [mm]} \quad x_{B_2}' = 24.3 - 16 \cos \theta_2 \text{ [mm]}$$

$$y_{B_1}' = -53 \text{ [mm]} \quad y_{B_2}' = -53 \text{ [mm]}$$

$$z_{B_1}' = -7 + 16 \sin \theta_1 \text{ [mm]} \quad z_{B_2}' = -7 + 16 \sin \theta_2 \text{ [mm]}$$

- **Puntos  $B_3$  y  $B_4$ .** En la Figura 32 se representa la ubicación de los puntos  $B_3$  y  $B_4$  ideales en la plataforma base y las relaciones trigonométricas utilizadas para la definición de sus coordenadas.

**Figura 32.** Ubicación de los puntos  $B_3$  y  $B_4$



$$x_{B_3} = 45 \cos 30^\circ + ((16 - 16 \cos \theta_3) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_3} = 45 \sin 30^\circ - ((16 - 16 \cos \theta_3) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_3} = -7 + 16 \sin \theta_3 \text{ [mm]}$$

$$x_{B_4} = 45 \cos 30^\circ - ((16 - 16 \cos \theta_4) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_4} = 45 \sin 30^\circ + ((16 - 16 \cos \theta_4) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_4} = -7 + 16 \sin \theta_4 \text{ [mm]}$$

Las coordenadas reales de los puntos  $B_3$  y  $B_4$  en el prototipo de plataforma Stewart son:

$$x_{B_3}' = 53 \cos 30^\circ + ((24.3 - 16 \cos \theta_3) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_3}' = 53 \sin 30^\circ - ((24.3 - 16 \cos \theta_3) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_3}' = -7 + 16 \sin \theta_3 \text{ [mm]}$$

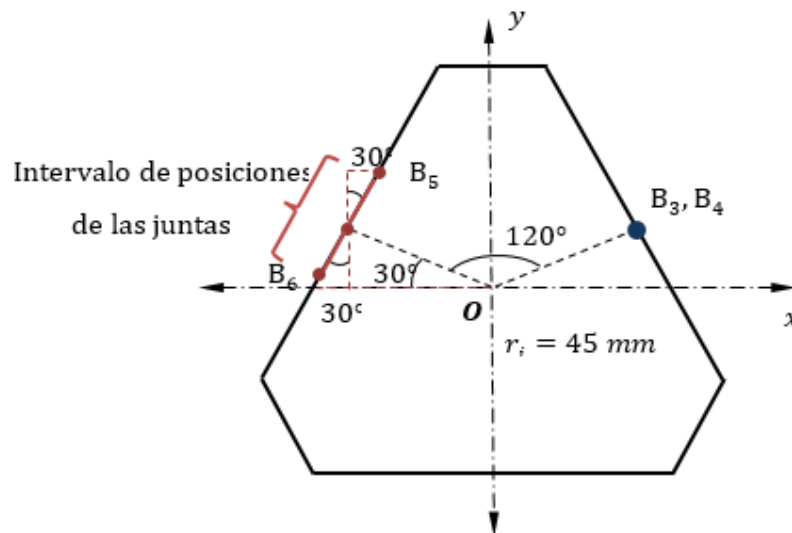
$$x_{B_4}' = 53 \cos 30^\circ - ((24.3 - 16 \cos \theta_4) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_4}' = 53 \sin 30^\circ + ((24.3 - 16 \cos \theta_4) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_4}' = -7 + 16 \sin \theta_4 \text{ [mm]}$$

➤ **Puntos  $B_5$  y  $B_6$ .** En la Figura 33 se representa la ubicación de los puntos  $B_5$  y  $B_6$  ideales en la plataforma base y las relaciones trigonométricas utilizadas para la definición de sus coordenadas.

**Figura 33.** Ubicación de los puntos  $B_5$  y  $B_6$ .



$$x_{B_5} = -45 \cos 30^\circ + ((16 - 16 \cos \theta_5) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_5} = 45 \sin 30^\circ + ((16 - 16 \cos \theta_5) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_5} = -7 + 16 \sin \theta_5 \text{ [mm]}$$

$$x_{B_6} = -45 \cos 30^\circ - ((16 - 16 \cos \theta_6) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_6} = 45 \sin 30^\circ - ((16 - 16 \cos \theta_6) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_6} = -7 + 16 \sin \theta_6 \text{ [mm]}$$

Las coordenadas reales de los puntos  $B_3$  y  $B_4$  en el prototipo de plataforma Stewart son:

$$x_{B_5}' = -53 \cos 30^\circ + ((24.3 - 16 \cos \theta_5) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_5}' = 53 \sin 30^\circ + ((24.3 - 16 \cos \theta_5) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_5}' = -7 + 16 \sin \theta_5 \text{ [mm]}$$

$$x_{B_6}' = -53 \cos 30^\circ - ((24.3 - 16 \cos \theta_6) \sin 30^\circ) \text{ [mm]}$$

$$y_{B_6}' = 53 \sin 30^\circ - ((24.3 - 16 \cos \theta_6) \cos 30^\circ) \text{ [mm]}$$

$$z_{B_6}' = -7 + 16 \sin \theta_6 \text{ [mm]}$$

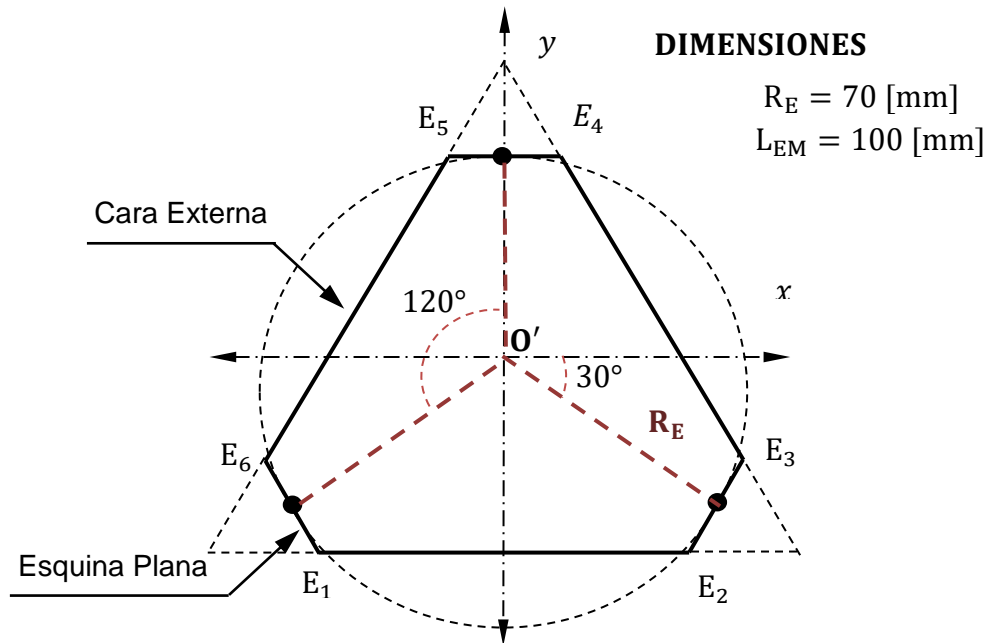
**3.1.3.2 Coordenadas del Efecto.** Teóricamente las uniones en el efector deben estar ubicadas en los vértices de la plataforma triangular, pero el diseño estrictamente triangular es impráctico, por ello se ubicaron los puntos de unión en el corte entre las esquinas planas y la línea que va al centroide del efector tal como se ilustra en la Figura 34. El origen de coordenadas  $O'$  en el efector posee las siguientes coordenadas relativas al origen  $O$  de la plataforma base:

$$X = 0 \text{ [mm]}$$

$$Y = 0 \text{ [mm]}$$

$$Z = 78,15 \text{ [mm]}$$

**Figura 34.** Ubicación ideal de las juntas del Efecto.



➤ **Puntos  $E_2$  y  $E_3$ .** A continuación se presentan las definiciones de las coordenadas ideales de los puntos de unión  $E_2$  y  $E_3$  sobre el efector.

$$x = 70 \cos 30^\circ = 35\sqrt{3} \text{ [mm]}$$

$$y = -70 \sin 30^\circ = -35 \text{ [mm]}$$

$$z = 78,15 \text{ [mm]}$$

A pesar que las ubicaciones de los puntos E sobre el efector en la Figura 34 son aproximaciones a las ubicaciones ideales de los puntos, se observó en la emulación del movimiento del prototipo que se obtenían mayores rangos de movimiento si dichos puntos se encontraban en las caras externas del efector y no en las esquinas planas, es decir, ubicando los puntos del efector en el mismo plano vertical donde se encuentran los puntos de unión sobre la plataforma base. Con el propósito que los puntos de unión sobre el efector fuesen cercanos a las coordenadas ideales, estos se ubicaron lo más cerca posible a los vértices de las caras externas en el efector, resultando las siguientes coordenadas reales de los puntos  $E_2$  y  $E_3$  sobre el prototipo de plataforma Stewart:

$$E_2x' = 45.96 \text{ [mm]}$$

$$E_3x' = 66.28 \text{ [mm]}$$

$$E_2y' = -50 \text{ [mm]}$$

$$E_3y' = -14.8 \text{ [mm]}$$

$$E_2z' = 90,87 \text{ [mm]}$$

$$E_3z' = 90,87 \text{ [mm]}$$

➤ **Puntos  $E_4$  y  $E_5$ .** A continuación se presentan las definiciones de las coordenadas ideales de los puntos de unión  $E_4$  y  $E_5$  sobre el efector.

$$x = 0 \text{ [mm]}$$

$$y = 70 \text{ [mm]}$$

$$z = 78,15 \text{ [mm]}$$

Las siguientes son coordenadas reales de los puntos  $E_4$  y  $E_5$  sobre el prototipo de plataforma Stewart:

$$\begin{aligned} E_4x' &= 20.32 \text{ [mm]} & E_5x' &= -20.32 \text{ [mm]} \\ E_4y' &= 64.80 \text{ [mm]} & E_5y' &= 64.80 \text{ [mm]} \\ E_4z' &= 90,87 \text{ [mm]} & E_5z' &= 90,87 \text{ [mm]} \end{aligned}$$

➤ **Puntos  $E_6$  y  $E_1$ .** A continuación se presentan las definiciones de las coordenadas ideales de los puntos de unión  $E_6$  y  $E_1$  sobre el efector.

$$\begin{aligned} x &= -70 \text{ Cos } 30^\circ = -35 \sqrt{3} \text{ [mm]} \\ y &= -70 \text{ Sen } 30^\circ = -35 \text{ [mm]} \\ z &= 78,15 \text{ [mm]} \end{aligned}$$

Las siguientes son coordenadas reales de los puntos  $E_6$  y  $E_1$  sobre el prototipo de plataforma Stewart:

$$\begin{aligned} E_6x' &= -66.28 \text{ [mm]} & E_1x' &= -45,96 \text{ [mm]} \\ E_6y' &= -14.8 \text{ [mm]} & E_1y' &= -50 \text{ [mm]} \\ E_6z' &= 90,87 \text{ [mm]} & E_1z' &= 90,87 \text{ [mm]} \end{aligned}$$

**3.1.3.3 Solución Cinemática Inversa.** Una vez conocidas las coordenadas de cada junta en la plataforma, es importante definir las distancias entre los servomotores y el efector móvil, pues estarán en función del cambio de la posición y de la orientación de la plataforma móvil con respecto al sistema de coordenadas inicial ( $X = 0$ ,  $Y = 0$ ,  $Z = 0$ ) ubicado en el centroide de la plataforma base. La distancia entre cada elemento está dada por la siguiente ecuación:

$$L_i^2 = (x_{E_i} - x_{B_i})^2 + (y_{E_i} - y_{B_i})^2 + (z_{E_i} - z_{B_i})^2$$

Para este caso, las longitudes de cada uno de los brazos son constantes, pero las coordenadas de las juntas de la base varían con respecto al ángulo de los actuadores (Servomotores). Para cada caso, la longitud  $L_i = 105$  [mm], por tanto resulta el siguiente arreglo de ecuaciones:

$$(1) 105^2 = \left(x_{E_1}' - (-24,3 + 16 \cos \theta_1)\right)^2 + \left(y_{E_1}' - (-53)\right)^2 + \left(z_{E_1}' - (-7 + 16 \sin \theta_1)\right)^2$$

$$(2) 105^2 = \left(x_{E_2}' - (24,3 - 16 \cos \theta_2)\right)^2 + \left(y_{E_2}' - (-53)\right)^2 + \left(z_{E_2}' - (-7 + 16 \sin \theta_2)\right)^2$$

$$(3) 105^2 = \left(x_{E_3}' - (53 \cos 30^\circ + (24,3 - 16 \cos \theta_3 \sin 30^\circ))\right)^2 \\ + \left(y_{E_3}' - (53 \sin 30^\circ - (24,3 - 16 \cos \theta_3 \cos 30^\circ))\right)^2 + \left(z_{E_3}' - (-7 + 16 \sin \theta_3)\right)^2$$

$$(4) 105^2 = \left(x_{E_4}' - (53 \cos 30^\circ - (24,3 - 16 \cos \theta_4 \sin 30^\circ))\right)^2 \\ + \left(y_{E_4}' - (53 \sin 30^\circ + (24,3 - 16 \cos \theta_4 \cos 30^\circ))\right)^2 + \left(z_{E_4}' - (-7 + 16 \sin \theta_4)\right)^2$$

$$(5) 105^2 = \left(x_{E_5}' - (-53 \cos 30^\circ + (24,3 - 16 \cos \theta_5 \sin 30^\circ))\right)^2 \\ + \left(y_{E_5}' - (53 \sin 30^\circ + (24,3 - 16 \cos \theta_5 \cos 30^\circ))\right)^2 + \left(z_{E_5}' - (-7 + 16 \sin \theta_5)\right)^2$$

$$(6) 105^2 = \left(x_{E_6}' - (-53 \cos 30^\circ + (24,3 - 16 \cos \theta_6 \sin 30^\circ))\right)^2 \\ + \left(y_{E_6}' - (53 \sin 30^\circ - (24,3 - 16 \cos \theta_6 \cos 30^\circ))\right)^2 + \left(z_{E_6}' - (-7 + 16 \sin \theta_6)\right)^2$$

Para la solución de cada ecuación, es decir, para calcular el ángulo de cada servomotor, se deben conocer las nuevas coordenadas del efector. Para calcular cada una de estas coordenadas de las 6 juntas del efector  $(x_{E_i}', y_{E_i}', z_{E_i}')$ , es necesario conocer la coordenada de origen y el cambio de posición y giro dado por el usuario. Esto se logra a través de la multiplicación de la coordenada original y la matriz de transformación, es decir:

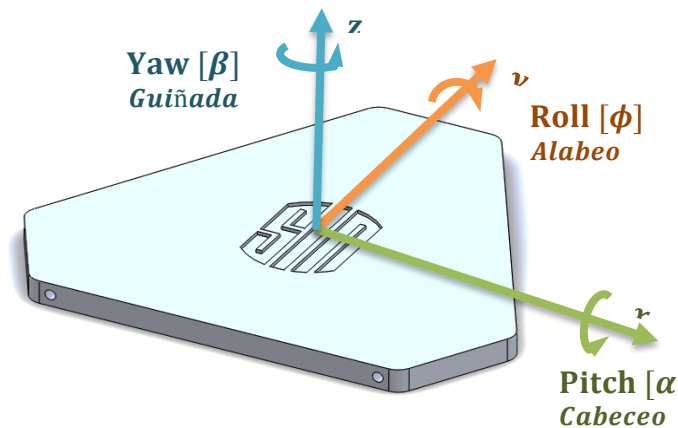
$$\begin{bmatrix} x'_{E_i} \\ y'_{E_i} \\ z'_{E_i} \\ 1 \end{bmatrix} = T * \begin{bmatrix} x_{E_i} \\ y_{E_i} \\ z_{E_i} \\ 1 \end{bmatrix}$$

La matriz de transformación representada por  $T$  es igual a:

$$T = \begin{bmatrix} \cos\varphi\cos\beta + \sin\varphi\sin\alpha\sin\beta & -\cos\varphi\sin\beta + \sin\varphi\sin\alpha\cos\beta & \sin\varphi\cos\alpha & X \\ \cos\alpha\sin\beta & \cos\alpha\cos\beta & -\sin\alpha & Y \\ -\sin\varphi\cos\beta + \cos\varphi\sin\alpha\sin\beta & \sin\varphi\sin\beta + \cos\varphi\sin\alpha\cos\beta & \cos\varphi\cos\alpha & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde las variables  $\alpha, \varphi, \beta, X, Y$  y  $Z$ , representan los 3 cambios de rotación y los 3 cambios de posición del Efecto ilustrado en la Figura 35.

**Figura 35.** Representación de las variables de influencia en la plataforma Stewart.



### 3.2 TARJETA DE ADQUISICIÓN DE DATOS

Se empleó una tarjeta Arduino UNO como tarjeta de adquisición de datos que fuese el elemento de comunicación entre el *software* de control en Simulink y el *hardware* mecánico del prototipo conformado por el prototipo de plataforma Stewart, los servomotores y el sensor de retroalimentación. Básicamente la tarjeta Arduino

recibe por comunicación serial las señales de posición de cada uno de los servomotores desde Simulink y las envía a los servomotores a través de seis de sus puertos PWM, realiza la lecturas de las señales de aceleración y velocidad angular en el sensor IMU utilizando protocolo I2C, las filtra y procesa convirtiéndolas en los ángulos Pitch y Roll para posteriormente enviarlas como señal de retroalimentación a Simulink. Las acciones antes mencionadas son llevadas a cabo por la tarjeta Arduino UNO con el programa que se adjunta en el Anexo C y la instalación de la tarjeta PCB que se presenta en el Anexo D.

### 3.3 SENSOR IMU

Se seleccionó el dispositivo IMU de referencia MPU 6050 (ver Figura 36) debido a la facilidad de adquisición en el mercado local, su reducido tamaño, bajo costo, su alta resolución de medida (16 bits) y la posibilidad de fusionar las mediciones del acelerómetro y giroscopio que lo conforman con el fin de determinar ángulos de posición más precisos que los que se pueden conseguir con cada uno de sus componentes por separado.

**Figura 36.** Sensor IMU MPU 6050.



Fuente: [https://robu.in/wp-content/uploads/2014/12/mpu\\_-\\_6050\\_gyro\\_sensor.png](https://robu.in/wp-content/uploads/2014/12/mpu_-_6050_gyro_sensor.png)

El dispositivo IMU fue instalado en el centroide del efector móvil de la plataforma y paralelo al mismo con el fin de obtener los cambios de posición angular al que está siendo sometido. El sensor tiene la capacidad de medir los cambios de aceleración

y velocidad angular en las tres dimensiones espaciales, es decir, en coordenadas X, Y y Z y comunicarlos vía protocolo de comunicación I2C<sup>15</sup> a otros dispositivos que cuenten con dicho protocolo de comunicación como lo son las tarjetas Arduino.

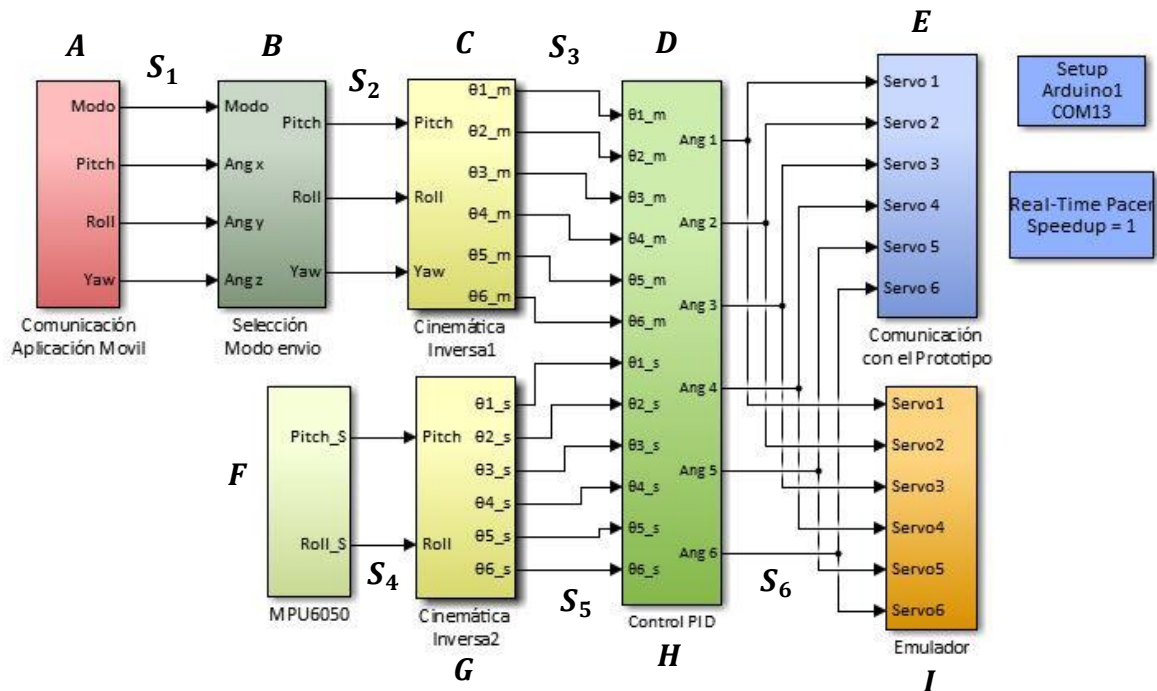
---

<sup>15</sup> IIC o I2C por sus siglas en ingles *Inter-Integrated Circuit*, es un protocolo de comunicación entre dispositivos maestros y esclavos que se realiza por medio de dos líneas, una línea de datos por la cual se transmite la información y una línea de señal de reloj para la sincronización entre los dos dispositivos.

## 4. SOFTWARE DE CONTROL DEL PROTOTIPO DE PLATAFORMA STEWART

Este capítulo aborda el *software* del controlador del prototipo de plataforma Stewart, es decir, el programa encargado de la comunicación entre la aplicación móvil, el cálculo de cinemática inversa y el cálculo y generación de las señales de control hacia la planta. El programa de control fue desarrollado en el *software* Simulink y se representa en la Figura 37 donde se muestra cada uno de los componentes que hacen posible la comunicación, el control y la manipulación del prototipo.

**Figura 37.** Programa de comunicación y control del prototipo de plataforma Stewart



El programa de control de la plataforma Stewart ilustrado en la Figura 37, se encarga de recibir las señales de posición angular ( $S_1$ ) enviadas vía *bluetooth* por la

aplicación móvil a través del bloque de comunicación (A), determinar el modo de envío (automático o manual) de señales que se realizó por la misma en el bloque (B), alimentar las matrices de rotación en el bloque (C) con las señales de posición angular ( $S_2$ ) indicadas por la aplicación móvil, determinar la señal de posición ( $S_3$ ) que debe configurarse en cada uno de los seis servomotores de la plataforma para reproducir la posición angular indicada por el mando, enviar las señales de posición de los servomotores a Arduino a través del bloque (E), recibir las señales de rotación angular ( $S_4$ ) enviadas por Arduino leídas del IMU instalado en el prototipo por medio del bloque (F) y con ellas determinar la posición ( $S_5$ ) que tiene cada servomotor en el prototipo de la plataforma en el bloque (G) y retroalimentarlas al bloque de control (H) para generar la señal de control de posición de los servomotores ( $S_6$ ) que es enviada a Arduino (E) y al emulador de movimiento de la plataforma en el bloque (I).

#### 4.1 BLOQUE COMUNICACIÓN CON LA APLICACIÓN MÓVIL

El bloque (A) del programa de control ilustrado en la Figura 37 recibe las señales de posición angular desde el dispositivo móvil a través de la comunicación serial de Simulink con un puerto COM configurado para recibir información vía *bluetooth*; para configurar el puerto COM ver el Anexo E.

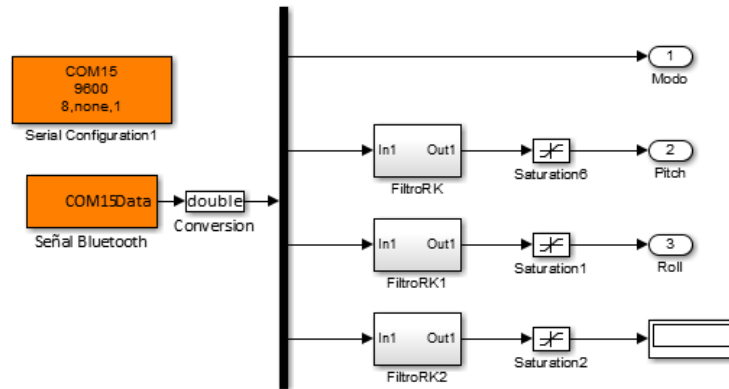
La programación dentro del bloque A se ilustra en Figura 38, consiste en el bloque *Serial Configuration* en el cual se configuran parámetros de la comunicación serial tales como el puerto por el cual se recibirá información para nuestro caso COM15, la velocidad de comunicación o *baud rate*<sup>16</sup> de 9600, la cantidad de bits de cada dato recibido son 8, el tipo de paridad y el *bit* que indica la transmisión de cada dato; el bloque *Serial Receive*, nombrado Señal Bluetooth en el programa de control, en el cual se debe configurar el puerto por el cual se va a recibir información, la cantidad

---

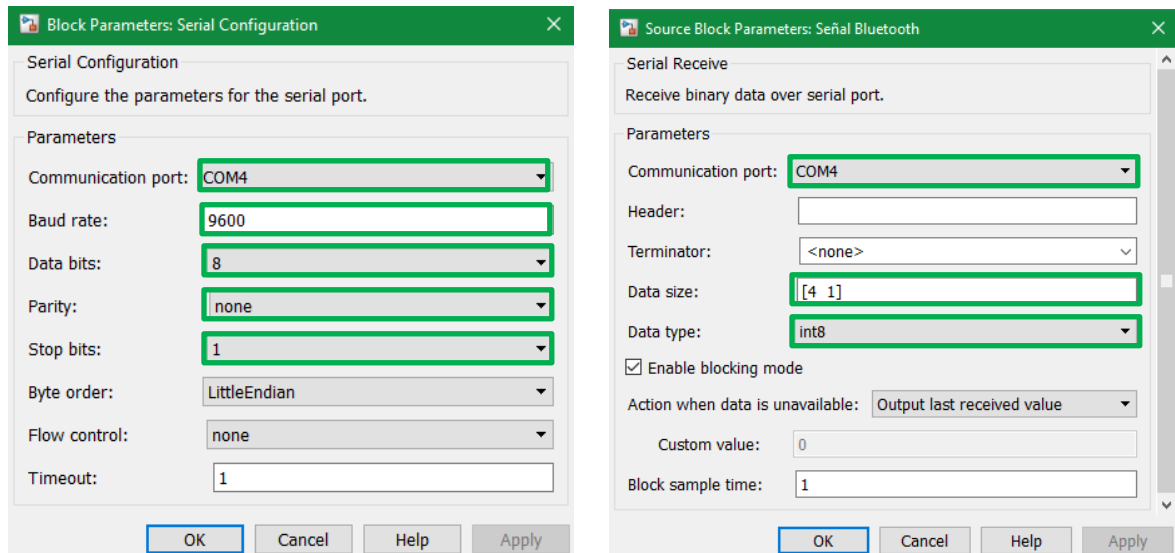
<sup>16</sup> La tasa de baudios o baudaje es el número de unidades de señal por segundo en comunicaciones. Tomado de: [https://es.wikipedia.org/wiki/Tasa\\_de\\_baudios](https://es.wikipedia.org/wiki/Tasa_de_baudios)

de datos a recibir que para nuestro caso son 4 señales unidimensionales, es decir, un arreglo [4 1] y el tipo de dato que se recibirá que es formato int8, la configuración de los bloques antes explicados se ilustra en la Figura 39.

**Figura 38.** Bloque de comunicación del programa de control



**Figura 39.** Configuración de los bloques de comunicación serial.



Además en el bloque de comunicación con la aplicación móvil se convierten los datos recibidos a un formato *double* que facilita el procesamiento de los mismos en Simulink, para luego des multiplexar la señal y obtener cada uno de los 4 datos

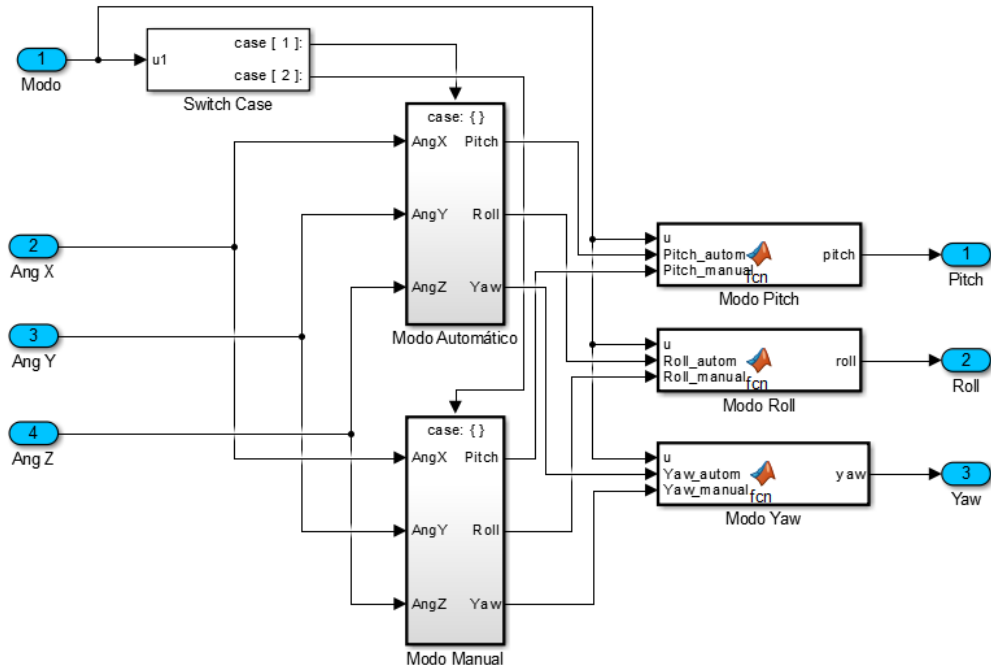
enviados desde la aplicación móvil, el primero determina el modo de envío y los 3 restantes son las señales Pitch, Roll y Yaw respectivamente. El bloque filtroRK realiza el filtrado de la señal por medio de una batería FIFO y convierte el valor filtrado a un valor entero que luego es pasado por un saturador que limita la señal a un valor máximo y mínimo que puede ser configurado en el prototipo.

## **4.2 BLOQUE SELECTOR MODO DE ENVÍO**

En el bloque selector modo de envío se separan las señales recibidas del bloque de comunicación con la aplicación móvil de acuerdo al modo en que se ejecuta el envío de información, seleccionando entre la opción de modo automático o manual. En el modo automático, las señales enviadas son las generadas por el sensor de inclinación del celular y el magnetómetro, mientras que para el modo manual, las señales son escalares que el usuario digita manualmente en la aplicación móvil.

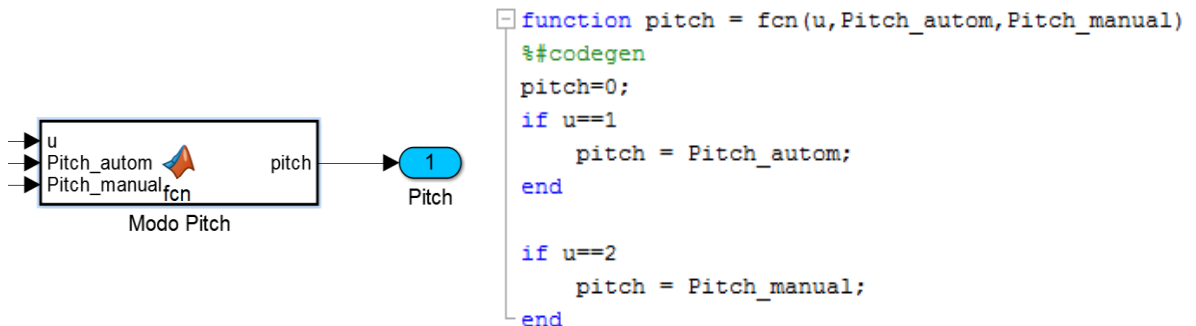
En la Figura 40 se muestra la programación utilizada en el bloque para la selección del modo de envío por parte de la aplicación móvil. El bloque selector modo de envío recibe 4 señales y verifica a través de un bloque *switch case* el valor del primer dato recibido, que puede ser 1 para el modo automático o 2 para el modo manual, este valor activa la respectiva función *case* donde se asignará el valor recibido a una variable con el sufijo automático o manual según sea el caso.

**Figura 40.** Programación del bloque selector modo de envío.



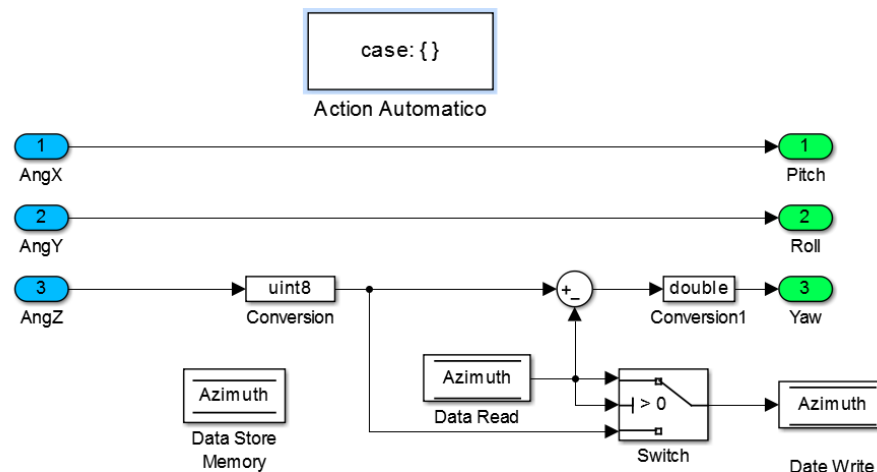
Una vez son definidas las variables automático y manual son alimentadas a los bloques de Modo, donde en cada uno de los bloques se asigna el valor de la señal a utilizar en la variable que será enviada al siguiente bloque del programa de control del prototipo, cada bloque de Modo tiene líneas de programación como las expuestas en la Figura 41.

**Figura 41.** Programación del bloque Modo Pitch.



**4.2.1 Modo Automático (Caso 1).** La captura de los ángulos Pitch y Roll no requiere tratamiento, sin embargo dado que el ángulo Yaw se mide mediante el magnetómetro del celular, el valor que registra inicialmente es una magnitud relacionada a la posición del mismo con respecto al norte magnético de la tierra, por ello se debe capturar la primera lectura y guardar la como valor inicial en la variable Azimuth para luego restarla a las variaciones del ángulo y así tener una medida del ángulo yaw del celular con respecto a la posición inicial del mismo. La adquisición de información en el bloque de modo automático se representa en la Figura 42.

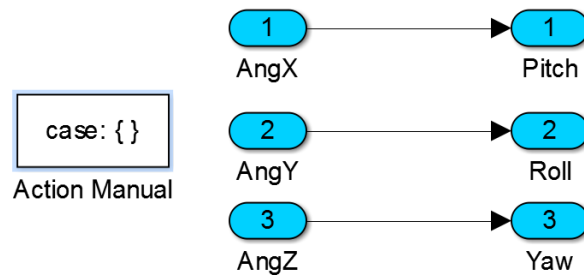
**Figura 42.** Programación para la adquisición de datos en Modo Automático.



Las magnitudes de las señales enviadas por el celular tienen las unidades de lectura del sensor de orientación, es decir, ángulos en unidades de grados sexagesimales [°].

**4.2.2 Modo Manual (Caso 2).** En este modo los valores de rotación que se desean en la plataforma son ingresados manualmente en la aplicación móvil en las unidades especificadas en la interfaz gráfica de la aplicación, es decir, grados [°] que son las unidades necesarias para alimentar el siguiente bloque en el programa de control, por ello, únicamente se realiza la adquisición de los valores deseados y se envían a los bloques siguientes, tal como se representa en la Figura 43.

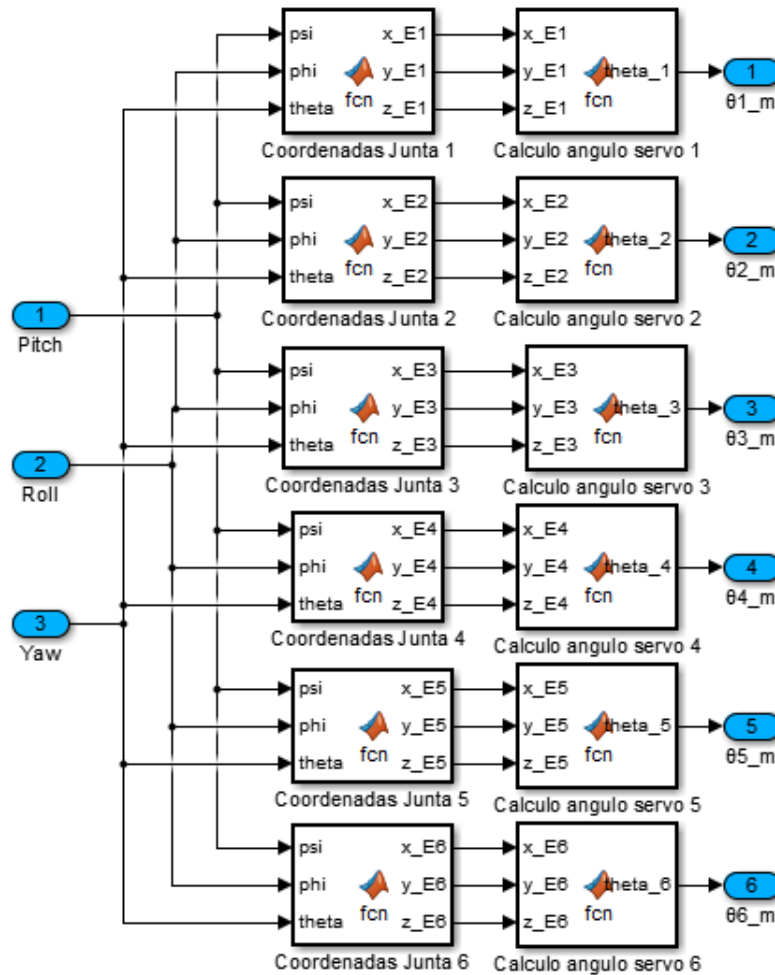
**Figura 43.** Programación para la adquisición de datos en Modo Manual.



### 4.3 BLOQUE CINEMÁTICA INVERSA I y II

Los bloques de cinemática inversa I y II reciben las señales de posición angular (Pitch y Roll) deseada o leída en la plataforma móvil según sea el caso y se encargan de calcular las coordenadas X, Y y Z de cada junta del efector por medio de los bloques “Coordenada Junta X” y determinar el ángulo que debe establecerse en cada servomotor para cumplir la posición deseada por el usuario en la aplicación móvil. Los bloques de programación de la cinemática inversa se encuentran representados en la Figura 44.

Figura 44. Programación del bloque cinemática I y II.



**4.3.1 Coordenadas Junta "X".** En la Figura 45 se ilustra el código interno del bloque Coordenadas Junta 1, mediante el cual se determinan las coordenadas de las juntas del efector. El bloque Coordenadas Junta 1 es el primero de los bloques en la Figura 44 y se toma como ejemplo para explicar la programación de los bloques de coordenadas de junta.

**Figura 45.** Líneas de programación en el bloque Coordenadas Junta 1.

```
function [x_E1, y_E1, z_E1] = fcn(psi,phi,theta)
    %#codegen
    x0_E1=-45.96;
    y0_E1=-50;
    z0_E1=90.87;
    x=0;
    y=0;
    z=0;
    T_E1=[cosd(phi)*cosd(theta)+sind(phi)*sind(psi)*sind(theta), -cosd(phi)*sind(theta)+
        cosd(psi)*sind(theta), cosd(psi)*cosd(theta), -sind(psi), y;
        -sind(phi)*cosd(theta)+cosd(phi)*sind(psi)*sind(theta), sind(phi)*sind(theta)+
        0, 0, 0, 1];
    E1=[x0_E1;
        y0_E1;
        z0_E1;
        1];
    M_E1= T_E1*E1;
    x_E1=M_E1(1,1);
    y_E1=M_E1(2,1);
    z_E1=M_E1(3,1);
```

En la parte Inicial del código de la Figura 45 se especifican las coordenadas iniciales de cada junta del efector definidas con el identificador 0 (por ejemplo x0\_E1) y la matriz de transformación  $T$ , luego se calculan cada una de las nuevas coordenadas ( $X_{E1}$ ,  $Y_{E1}$ ,  $Z_{E1}$ ) para las juntas a partir de la multiplicación de la matriz de transformación y la matriz de coordenadas iniciales.

$$\begin{bmatrix} x'_{E_i} \\ y'_{E_i} \\ z'_{E_i} \\ 1 \end{bmatrix} = T * \begin{bmatrix} x_{E_i} \\ y_{E_i} \\ z_{E_i} \\ 1 \end{bmatrix}$$

La matriz de transformación representada por  $T$  es igual a:

$$T = \begin{bmatrix} \cos\varphi\cos\beta + \sin\varphi\sin\alpha\sin\beta & -\cos\varphi\sin\beta + \sin\varphi\sin\alpha\cos\beta & \sin\varphi\cos\alpha & X \\ \cos\alpha\sin\beta & \cos\alpha\cos\beta & -\sin\alpha & Y \\ -\sin\varphi\cos\beta + \cos\varphi\sin\alpha\sin\beta & \sin\varphi\sin\beta + \cos\varphi\sin\alpha\cos\beta & \cos\varphi\cos\alpha & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde las variables  $\alpha, \varphi, \beta, X, Y$  y  $Z$ , representan los 3 cambios de rotación y los 3 cambios de posición del efector.

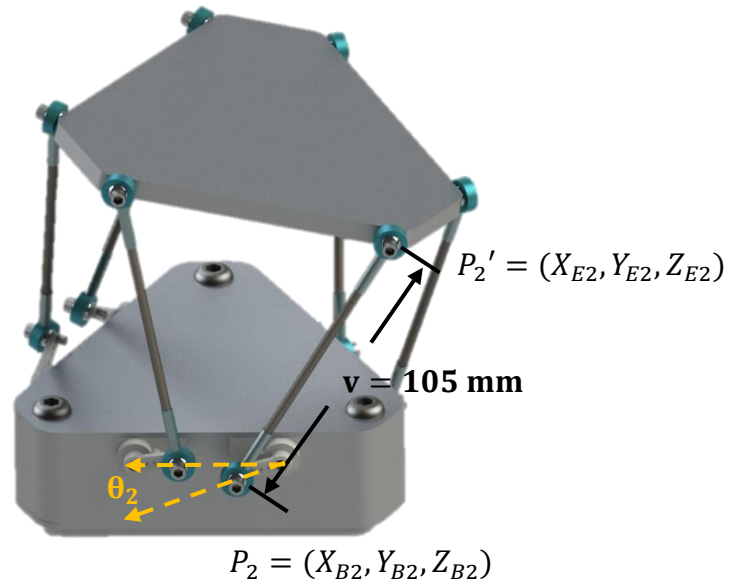
**4.3.2 Cálculo ángulo Servo “1”.** En la Figura 46 se ilustra el código interno del bloque de programación Cálculo ángulo Servo 1, tomado a manera de ejemplo para explicar el código mediante el cual se determina el ángulo que debe cumplir cada uno de los servomotores para satisfacer la posición deseada en la aplicación móvil. El proceso de solución del ángulo de cada servomotor se determina a partir de iteraciones entre las posiciones mínima ( $-90^\circ$ ) y máxima ( $90^\circ$ ), comparando los valores para cada ángulo y seleccionando el valor que tenga un error inferior a 0,1.

**Figura 46.** Línea de programación del bloque Cálculo ángulo Servo “X”.

```
1 function [theta_1,P1,P1_B]= fcn(x_E1,y_E1,z_E1)
2
3     P1=[x_E1,y_E1,z_E1];
4     theta_1=0;
5     for theta_E1=-90:0.1:90
6         v=((x_E1-(-24.3+16*cosd(theta_E1)))^2+(y_E1-(-53))^2+(z_E1-(-7+16*sind(theta_E1)))^2)^0.5;
7         error=abs(105-v);
8         if error<0.1
9             theta_1=theta_E1;
10        end
11    end
```

En la Figura 46 la variable  $v$  representa la longitud entre los puntos de unión de la plataforma base y móvil, para el prototipo diseñado en este proyecto la longitud es constante e igual a 105 [mm]. En la

**Figura 47.** Representación de la distancia  $v$  y ángulo del servomotor para un punto  $P_2$ .



$$v^2 = (X_{E2} - X_{B2})^2 + (Y_{E2} - Y_{B2})^2 + (Z_{E2} - Z_{B2})^2$$

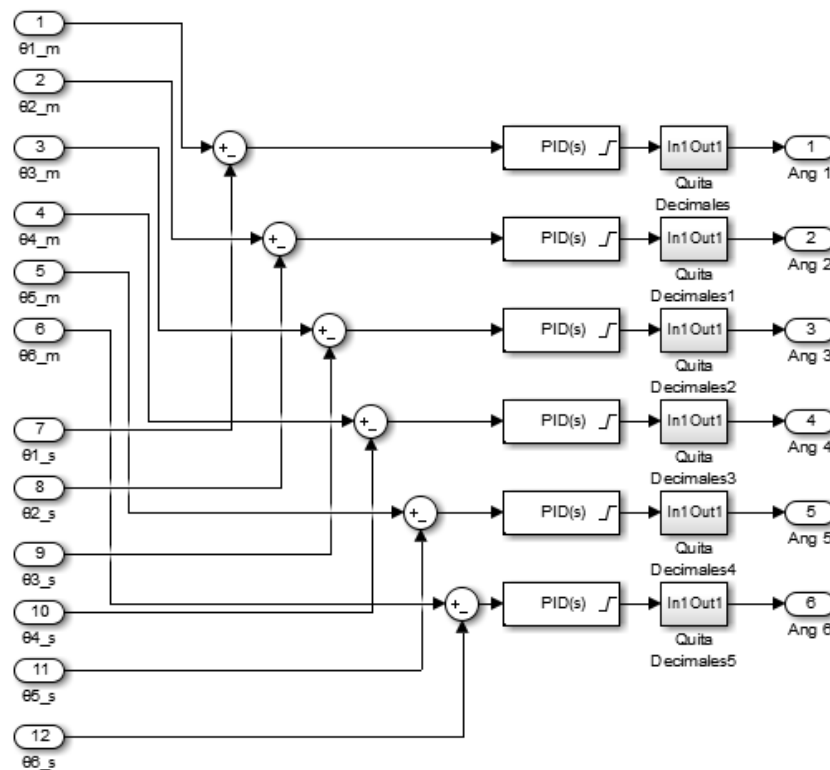
Para profundizar en la teoría del cálculo de la cinemática inversa ver Anexo B.

#### 4.4 BLOQUE DE CONTROL

La programación del bloque de control ilustrada en la Figura 48 recibe las señales de posición de los servomotores que emite el bloque de cinemática inversa I, estas señales son las posiciones angulares de los seis servomotores del prototipo que deben configurarse para que el efector móvil alcance la posición angular deseada o *set point*, también recibe las señales emitidas por el bloque cinemática inversa II que representan las posiciones de los servomotores que realmente están establecidas en el prototipo conocidas como señales de realimentación, con las

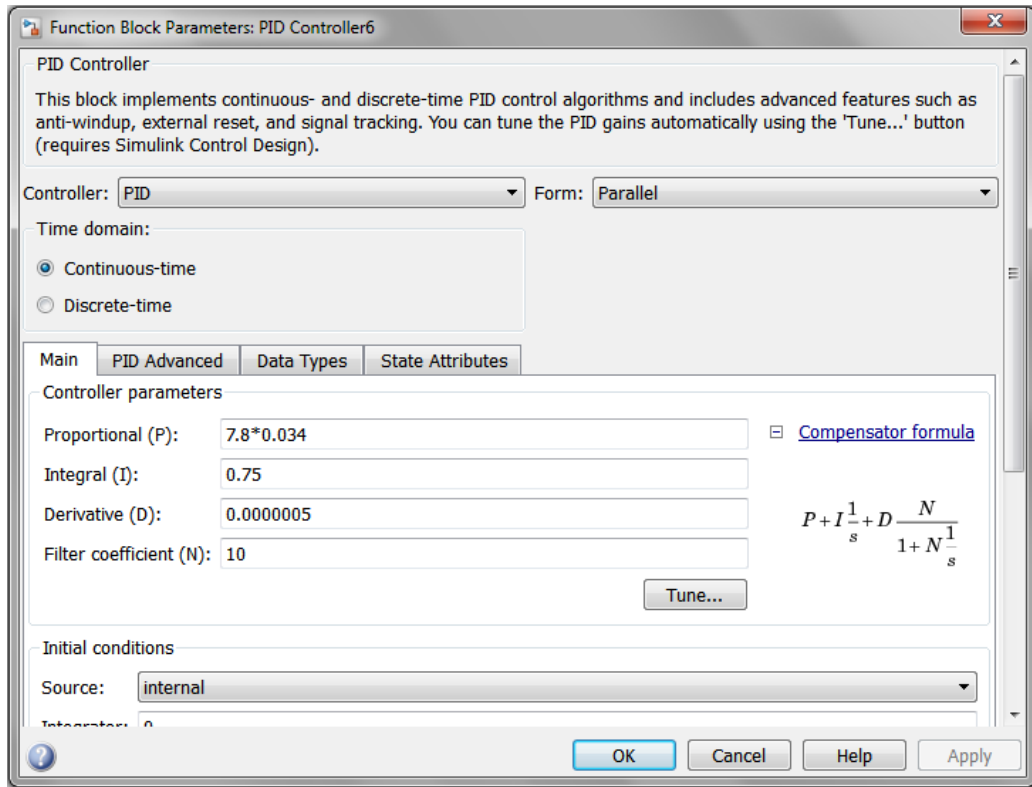
señales anteriores se determina el error por realimentación negativa de la señal de cada servomotor y se alimenta a los bloques de control PID de los mismos, luego la señal de control de posición de cada servo es redondeada a una magnitud con un valor decimal y enviada al siguiente bloque en el programa de control.

**Figura 48.** Programación del bloque de control



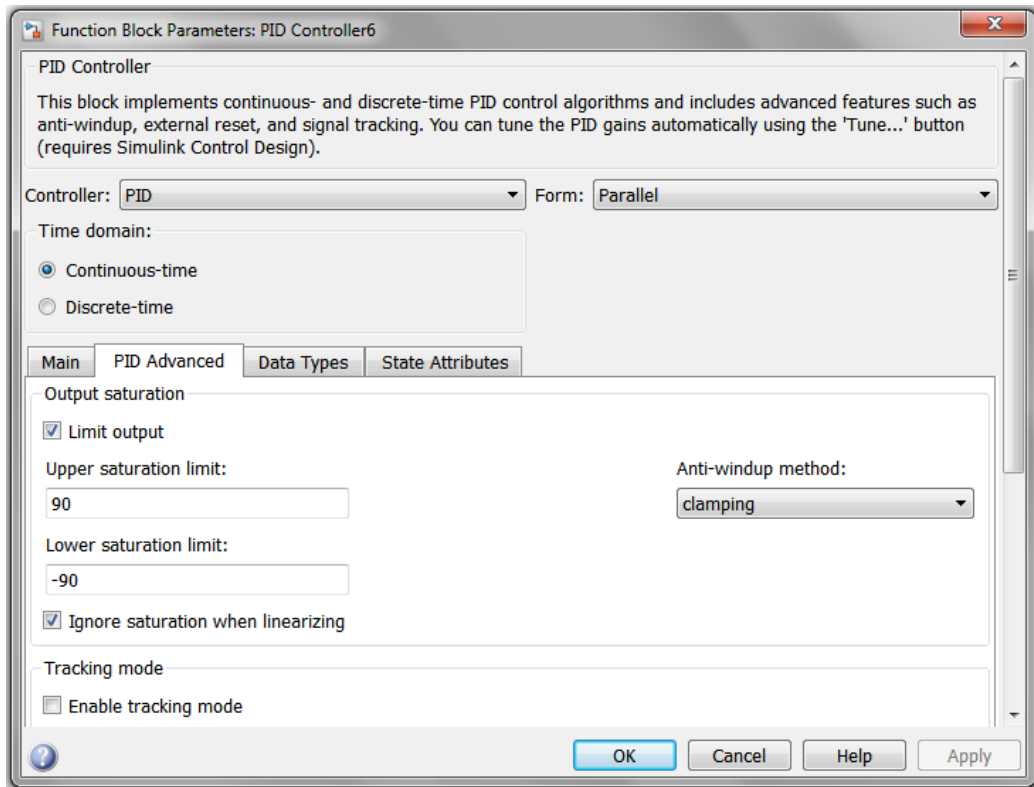
Para implementar los controladores PID en el software Simulink se utilizó la herramienta instalada por defecto *Continuous*, la cual tiene un bloque llamado *PID Controller* apreciado en la Figura 48, este bloque debe configurarse con el tipo de controlador que se desea, para nuestro caso PID, el tipo de control ideal o paralelo, el dominio del tiempo y las constantes de control que se vayan a implementar, la configuración básica del bloque se representa en la Figura 49.

**Figura 49.** Configuración básica del bloque *PID Controller*



Adicionalmente se realizaron configuraciones avanzadas del bloque *PID Controller* con el fin de limitar las respuestas de las señales de control a los límites mínimo y máximo de señal que puede recibir el servomotor. En la Figura 50 se aprecia la configuración del límite mínimo (-90) y máximo (90) de la señal de control y la configuración *clamping* como método de rebose, la cual evita que la acción integral continúe aumentando o disminuyendo cuando la señal de control se encuentra en el límite inferior o superior.

**Figura 50.** Configuración avanzada del bloque *PID Controller*



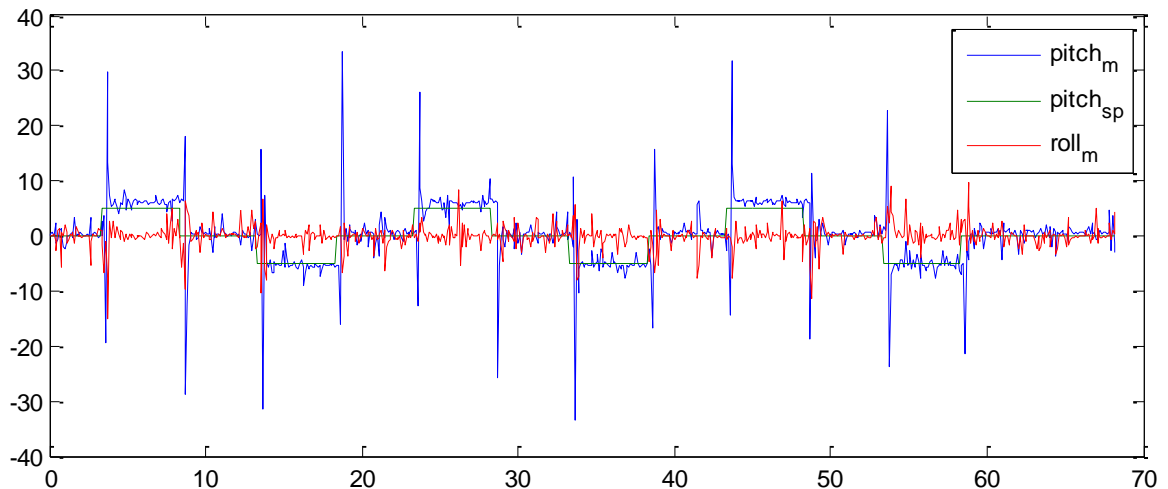
**4.4.1 Controladores PID.** Para la generación de las señales de control del prototipo de plataforma Stewart se utilizaron controladores PID en cada una de las señales de posición de los servomotores, de manera que los bloques PID de cada servomotor fueron alimentados con el valor de error resultado de restar la posición angular del servomotor en el prototipo al valor de la posición del servomotor que debe configurarse.

Para diseñar los controladores PID del prototipo de plataforma Stewart inicialmente se caracterizó el sistema para encontrar la función de transferencia que define el comportamiento en lazo abierto del mismo, dado que el prototipo cuenta con seis actuadores, es decir, con seis sistemas, se obtuvieron las funciones de transferencia

respectivas de cada uno de ellos y se promediaron con el fin de determinar un controlador que funcione para todos los sistemas.

**4.4.1.1 Caracterización del Sistema.** Para caracterizar el sistema se tomaron inicialmente las lecturas Pitch y Roll del dispositivo IMU instalado en el prototipo para inclinaciones del ángulo Pitch entre -5 y 5 grados emitidos por el mando. Para obtener buenas lecturas se realizaron ajustes a los datos que consistieron en restar el valor inicial de lectura del sensor a cada uno de los ángulos medidos y así ajustar las mediciones al cero del prototipo. La respuesta del sensor IMU instalado en el prototipo a las acciones antes mencionadas se ilustra en la Figura 51.

**Figura 51.** Lectura de los ángulos Pitch y Roll del sensor instalado en el prototipo.

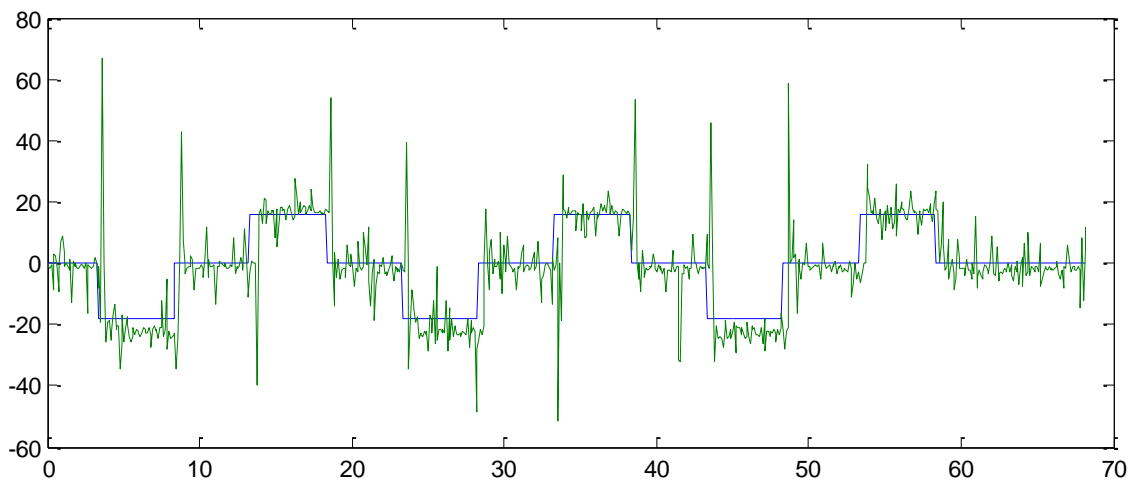


En la Figura 51 se observa que los datos inician en un valor aproximado a cero grados, sin embargo se eliminaron aproximadamente los primeros dos segundos de lectura de valores para dar tiempo a que las lecturas del IMU se estabilizaran. A

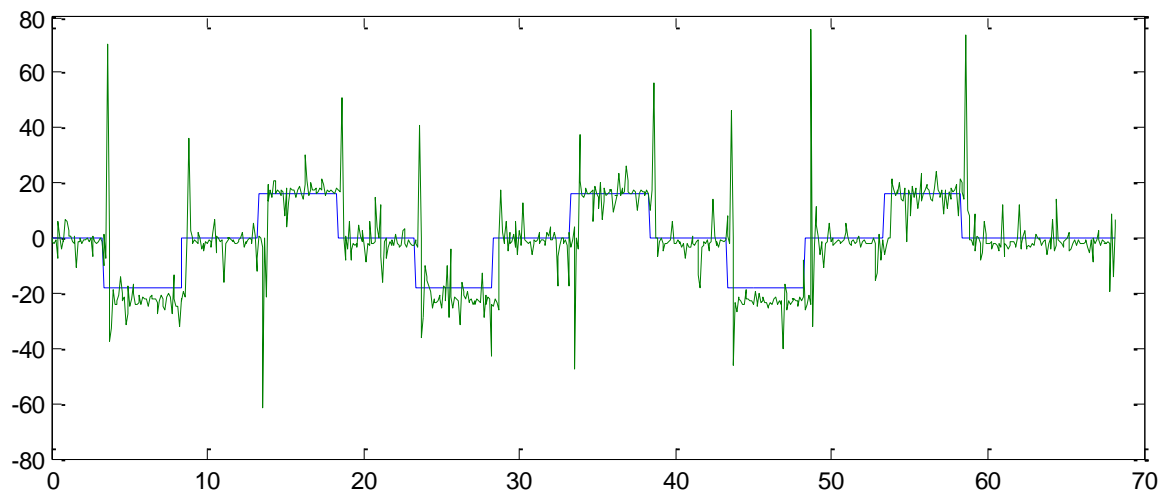
pesar que en los resultados de la gráfica el sistema posee valores cercanos al *set point*, lo ideal es que los valores de subida del ángulo Pitch logren llegar al *set point* y no sea notable el valor de *offset* que se observa en la gráfica. Por otro lado, la implementación del control verifica si los picos que se presentan en cada cambio se deben a características propias del sensor o a movimientos bruscos de la plataforma.

Con información presentada en la Figura 51 y usando la cinemática inversa, se determina el valor aproximado de los ángulos de cada servomotor en el prototipo, resultando las siguientes gráficas de posición de cada servomotor (Ángulo del respectivo servomotor con respecto al tiempo).

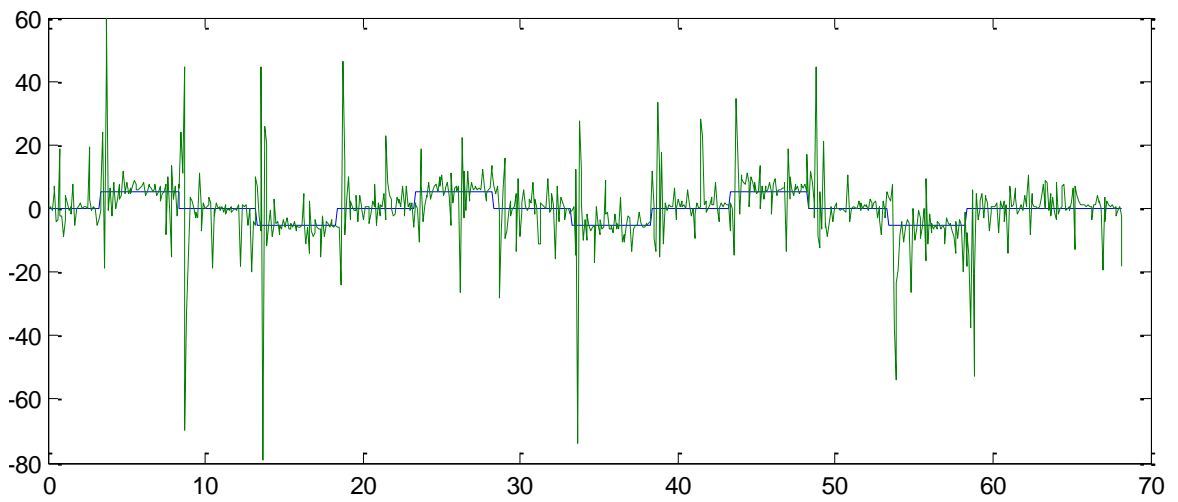
**Figura 52.** Posición del servomotor 1 sin controlador



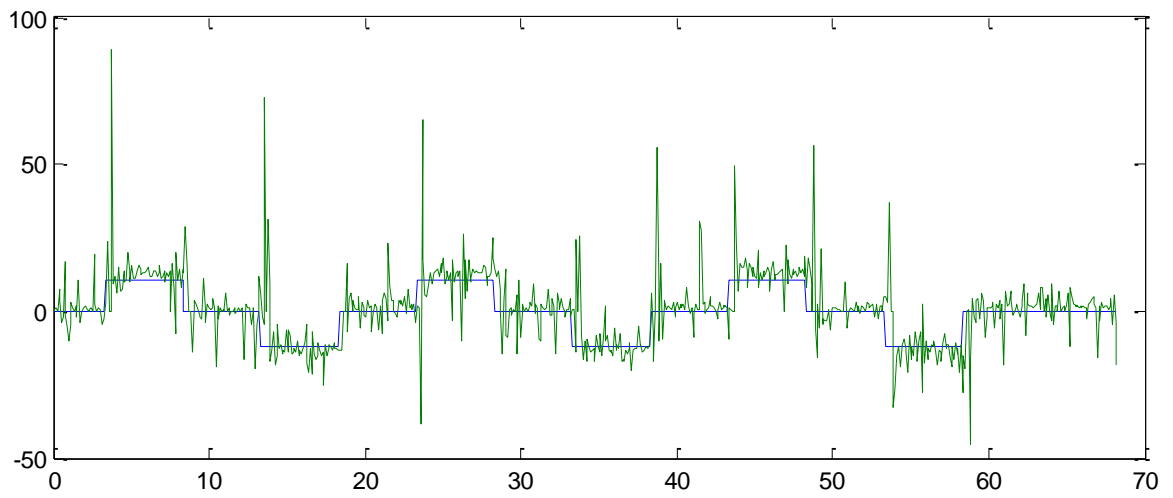
**Figura 53.** Posición del servomotor 2 sin controlador



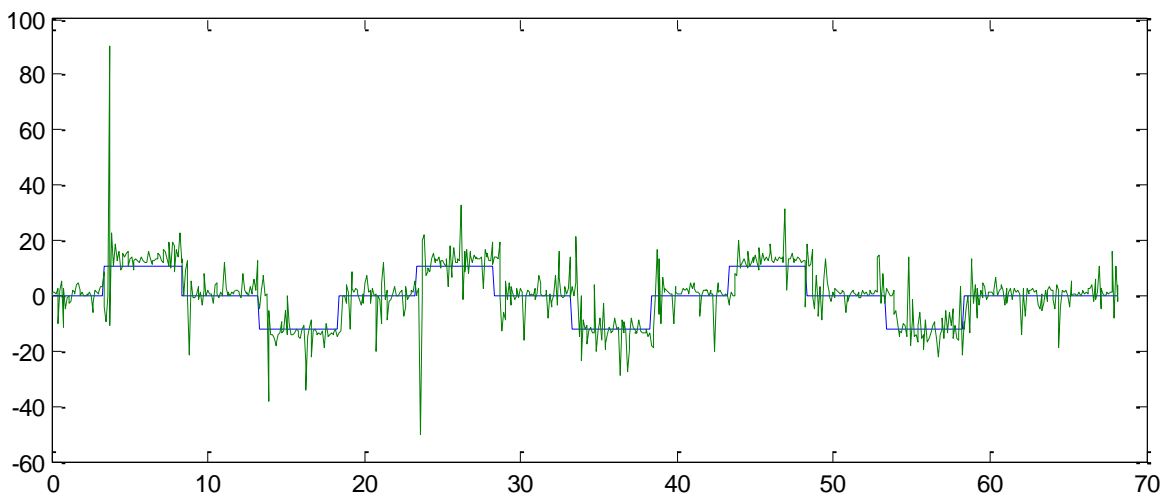
**Figura 54.** Posición del servomotor 3 sin controlador



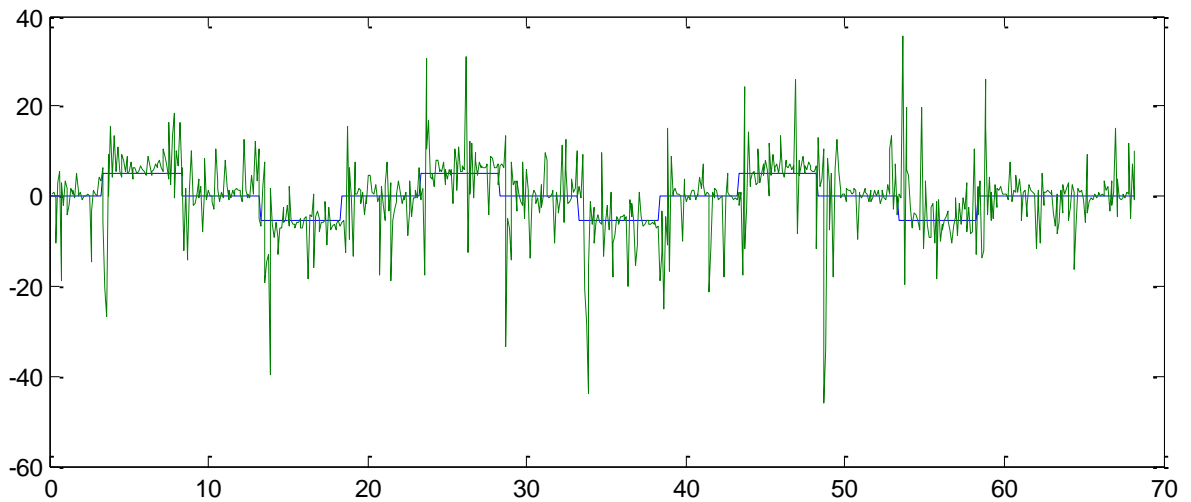
**Figura 55.** Posición del servomotor 4 sin controlador



**Figura 56.** Posición del servomotor 5 sin controlador



**Figura 57.** Posición del servomotor 6 sin controlador



Con la información de las gráficas anteriores se utilizó la teoría de identificación de sistemas en la cual se realiza una regresión lineal que contrasta la entrada del sistema con la salida del mismo y se obtiene un modelo matemático discreto. En este proyecto se usó una estructura Box-Jenkins más conocida como BJ para la identificación de los sistemas, la cual se realiza en Matlab usando el siguiente comando:

$$\text{modelo BJ} = \text{bj}(\text{datos de entrada y salida}, [\text{nb nc nd nf nk}]);$$

Para alimentar la herramienta BJ es necesario organizar los datos de entrada y salida del sistema en un mismo arreglo, para ello se utilizó el comando *iddata* de Matlab que organiza las señales de entrada y salida del sistema en un solo paquete de información, el comando *iddata* debe ejecutarse con la siguiente sintaxis:

$$\text{iddata}(\text{salida}, \text{entrada}, Ts)$$

Los valores de *nb*, *nc*, *nd* y *nf* en la estructura BJ son las órdenes de los polinomios del modelo a identificar y *nk* es el retraso de la salida respecto a la entrada.

Entonces para la identificación de los sistemas del prototipo diseñado en este proyecto se utilizaron comandos BJ como el que se presenta a continuación:

$$bj1_{s1} = bj(data\_ide\_s1, [1 \ 1 \ 1 \ 1 \ 2]);$$

Los modelos BJ se encuentran definidos por la siguiente ecuación:

$$G(z) = \frac{B(z)}{F(z)}u(t - k) + \frac{C(z)}{D(z)}e(t)$$

Donde  $u(t - k)$  representa la entrada atrasada  $k$  intervalos de tiempo y  $e(t)$  representa una entrada de ruido gaussiano. Por lo tanto,  $B(z)/F(z)$  es el modelo de la planta, es decir, el modelo de cada servomotor y  $C(z)/D(z)$  el modelo de ruido del sistema. Para efectos prácticos, fue utilizado el modelo de la planta y se descartó el modelo de ruido.

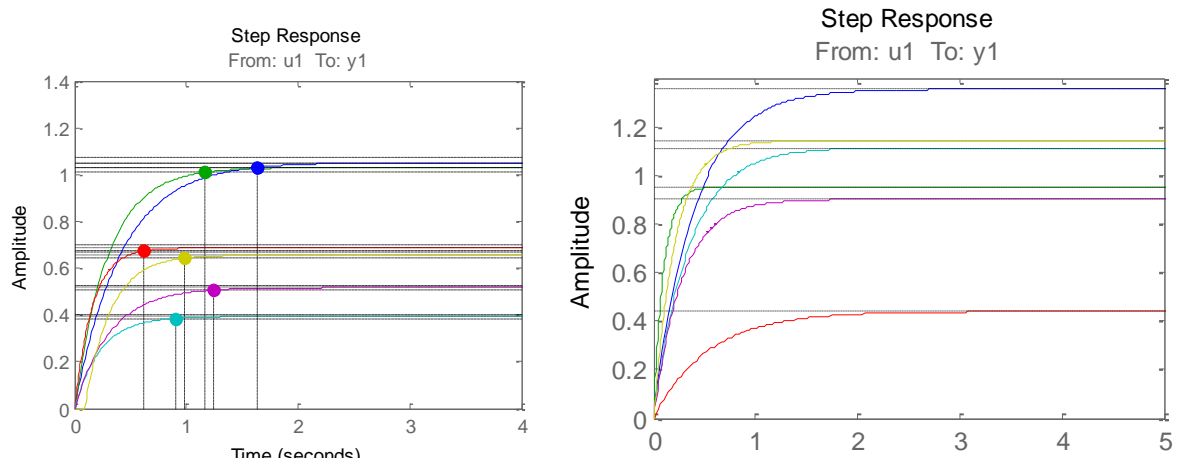
En la Tabla 3. **Modelos BJ de cada servomotor** Tabla 3 se presentan los modelos BJ para cada uno de los servomotores discriminados por el eje rotacional del prototipo e identificados con el sufijo S y el número del servomotor.

**Tabla 3.** Modelos BJ de cada servomotor

Modelo	Pitch	Roll
$G_{s1}$	$\frac{2.518}{s + 2.395}$	$\frac{3.409}{s + 2.518}$
$G_{s2}$	$\frac{3.473}{s + 3.373}$	$\frac{9.752}{s + 10.24}$
$G_{s3}$	$\frac{4.359}{s + 6.352}$	$\frac{0.8215}{s + 1.872}$
$G_{s4}$	$\frac{1.703}{s + 4.329}$	$\frac{3.207}{s + 2.881}$
$G_{s5}$	$\frac{1.624}{s + 3.142}$	$\frac{3.228}{s + 3.583}$
$G_{s6}$	$\frac{2.859}{s + 4.356}$	$\frac{5.453}{s + 4.775}$

En la Figura 58 se presentan las respuestas de los modelos BJ de cada servomotor a la entrada de señal de un paso.

**Figura 58.** Respuesta de los modelos BJ de cada servomotor a un paso



La Tabla 4 presenta las variables  $k$  y  $T$  en cada uno de los ejes de rotación del prototipo.

**Tabla 4.** Variables  $k$  y  $T$  de los modelos BJ de cada servomotor

Variable	Pitch		Roll	
	$k$	$T$	$k$	$T$
<b>Servo1</b>	2.518	2.395	3.409	2.518
<b>Servo2</b>	3.473	3.373	9.752	10.24
<b>Servo3</b>	4.359	6.352	0.8215	1.872
<b>Servo4</b>	1.703	4.329	3.207	2.881
<b>Servo5</b>	1.624	3.142	3.228	3.583
<b>Servo6</b>	2.859	4.356	5.453	4.775
<b>Promedio</b>	2.756	3.991167	4.31175	4.3115

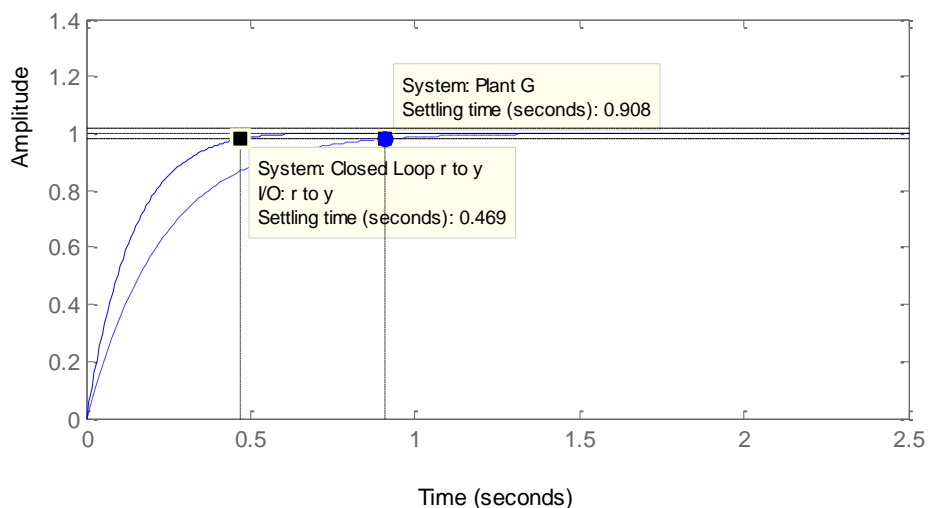
Los valores promedio de las variables  $k$  y  $T$  se utilizaron para generar una función de transferencia de primer orden con la que se diseñó el controlador PID que actúa sobre los servomotores.

$$G_{\text{prom}_x} = \frac{2.756}{s + 3.99}$$

$$G_{\text{prom}_y} = \frac{4.31}{s + 4.31}$$

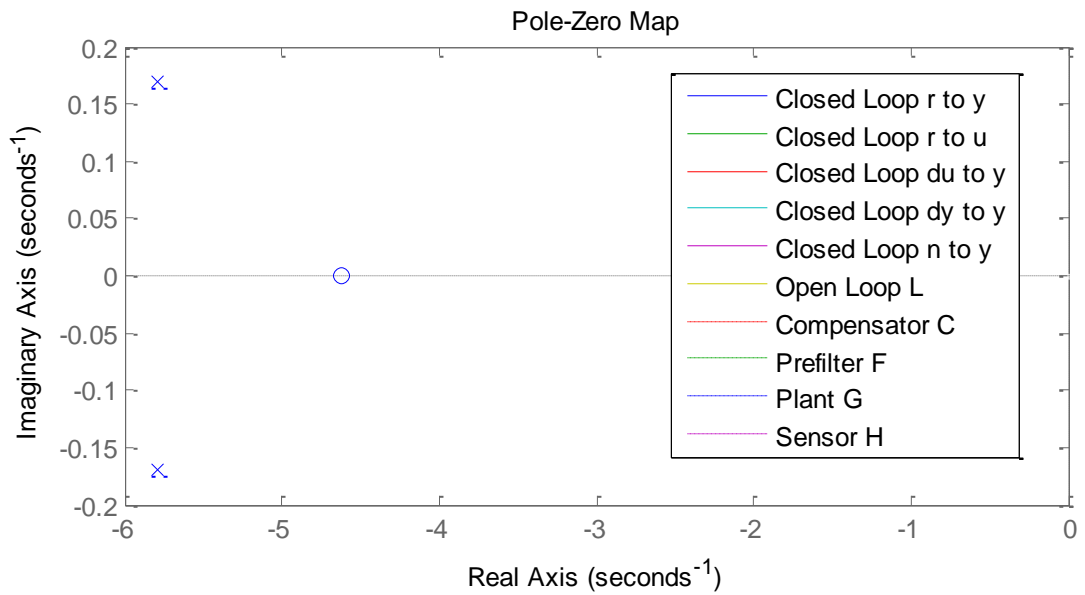
**4.4.1.2 Diseño del Controlador.** Para diseñar los controladores PID se utilizó la herramienta gráfica *Sisotool*<sup>17</sup> de MatLab en la cual es posible modificar los polos y ceros (ver Figura 60) del sistema en lazo cerrado y verificar la respuesta del mismo a una señal de entrada *step* o paso. En este proyecto se ajustaron los polos y ceros para obtener una velocidad de respuesta aproximadamente 2 veces mayor por parte del sistema, tal como se muestra la Figura 59.

**Figura 59.** Respuesta del sistema en lazo cerrado y lazo abierto



<sup>17</sup> Sisotool es una herramienta gráfica que permite el análisis de sistemas lineales. Sisotool muestra la ubicación de las raíces del sistema, los polos y los ceros en cadena abierta.

**Figura 60.** Ubicación del polo del sistema

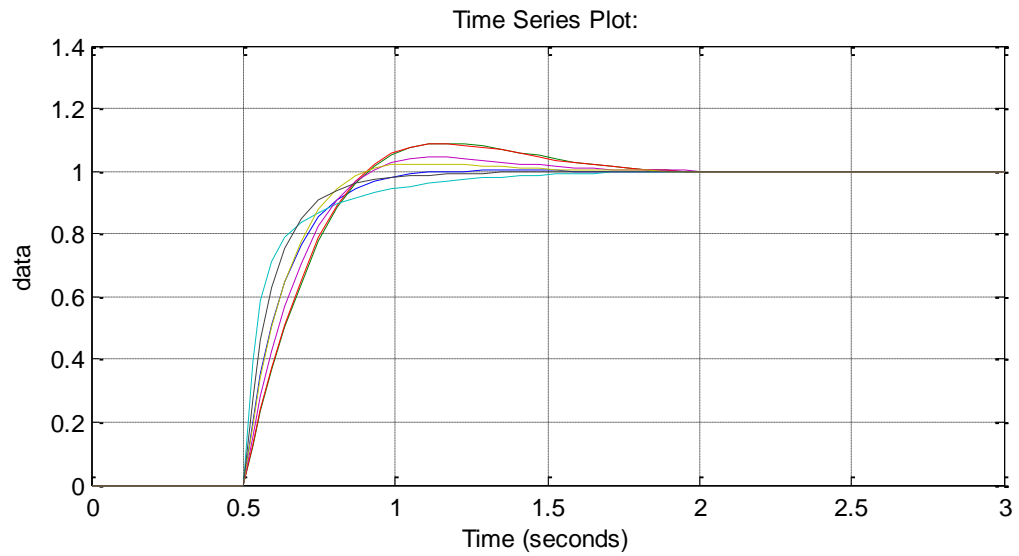


La ecuación del controlador PI calculado con las características de respuesta del sistema en lazo cerrado presentado en la Figura 59 se presenta en la siguiente ecuación:

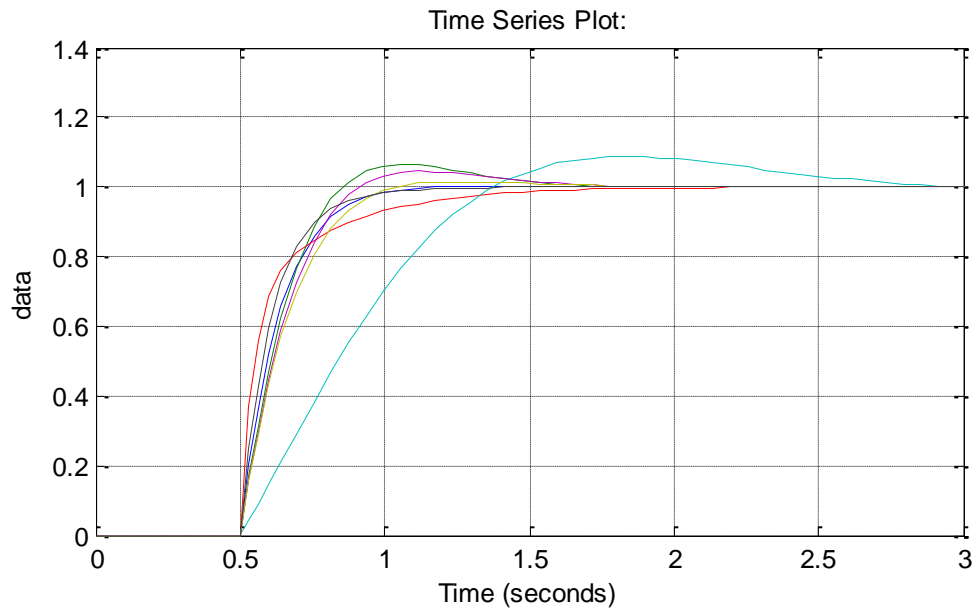
$$PI = 1.716 + \frac{7.8}{s}$$

El controlador diseñado se utilizó en todos los modelos identificados con el fin de verificar la acción de control sobre los mismos, en las Figuras Figura 61 y Figura 62 se muestra la respuesta de todos los modelos a una señal escalón como entrada, utilizando el controlador diseñado.

**Figura 61.** Respuesta de los modelos a una señal escalón en el ángulo Pitch



**Figura 62.** Respuesta de los modelos a una señal escalón en el ángulo Roll



En todos casos los expuestos en las dos figuras anteriores se logra el valor de referencia ( $r=1$ ) y no se presentan valores de sobre pico mayores a 10%, entonces el controlador diseñado presenta una respuesta rápida y con un sobre-pico aceptable para controlar el prototipo diseñado.

#### **4.5 BLOQUE COMUNICACIÓN CON EL PROTOTIPO**

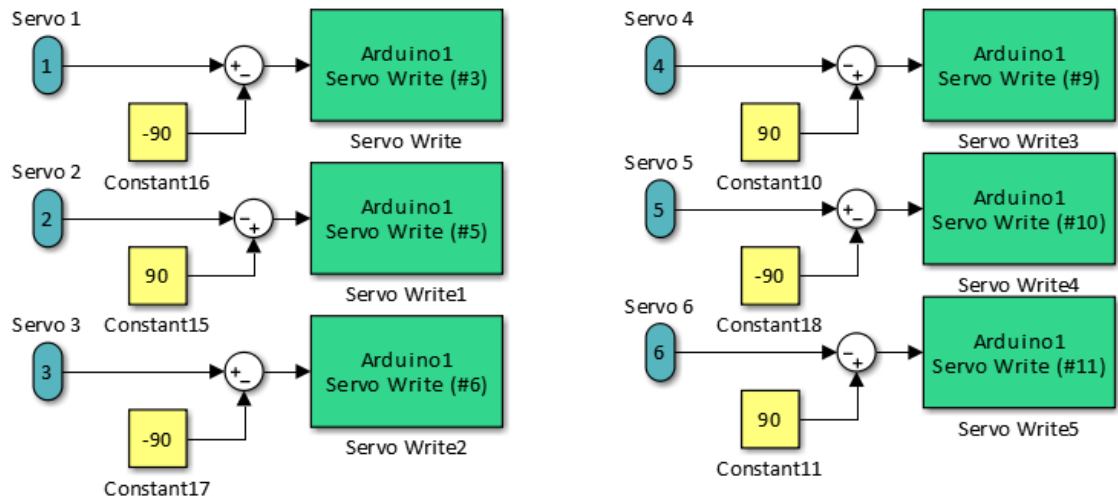
En el bloque de comunicación con el prototipo ingresan las señales controladas de la posición de los servomotores que se deben configurar para cumplir la posición angular deseada en el efector móvil. En este bloque se encuentran los bloques necesarios de la librería ArduinoIO<sup>18</sup> de Simulink que permiten la comunicación entre el programa de control y el prototipo de plataforma Stewart, estos bloques son el bloque *Arduino IO Setup* en el cual se configura el Arduino a utilizar y el puerto COM al cual se encuentra conectado y el bloque *Real- Time Pacer* en el cual se configura la relación deseada entre el tiempo de simulación y el tiempo real. Para información detallada del proceso de instalación de la librería ArduinoIO en MatLab ver el Anexo F.

En la Figura 63 se ilustra la programación del bloque de comunicación con el prototipo donde las señales de entrada son los ángulos respectivos a cada servomotor, estos ángulos deben ser desfasados  $90^\circ$  o  $-90^\circ$  debido a la orientación instalada de los servomotores en el prototipo y finalmente alimentar los bloques *Servo Write* de la librería ArduinoIO que controlan los puertos PWM del Arduino UNO conectados a los servomotores del prototipo.

---

<sup>18</sup> Librería creada por Giampiero Campa para la comunicación por puerto serial entre MatLab/Simulink y Arduino.

**Figura 63.** Programación del bloque de comunicación con el prototipo



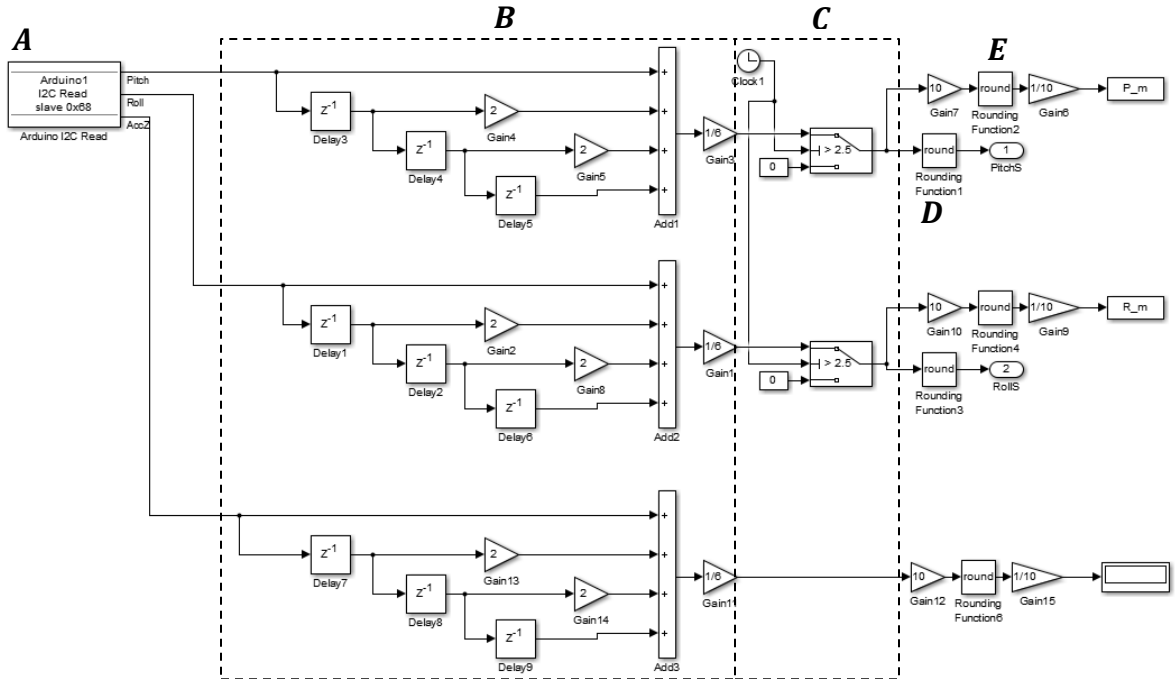
#### 4.6 BLOQUE EMULADOR

La programación del bloque que alimenta la animación del modelo CAD del prototipo de plataforma diseñado en este proyecto se explicará... en la sección 5.2... correspondiente al emulador de movimiento.

#### 4.7 BLOQUE MPU6050

La configuración de este bloque ilustrada en la Figura 64 tiene como función principal leer las señales de posición angular (Pitch y Roll) que envía Arduino haciendo uso de los bloques de lectura I2C de la librería ArduinoIO (A), además filtra las señales aplicando un filtro basado en una batería FIFO (B), elimina las mediciones del sensor de los primeros 2.5 segundos reemplazándolas por el valor de cero en (C), el bloque (D) *Rounding Function* convierte los valores en números enteros y la rama representada en (E) convierte la señal tomada del sensor en un número con un valor decimal y lo almacena en una variable del *Workspace*.

Figura 64. Programación del bloque MPU6050



## 5. INTERFAZ DE USUARIO

La interfaz de usuario está compuesta por la aplicación móvil Android utilizada como mando, donde el usuario configura el *set point* del prototipo plataforma Stewart y el emulador de movimiento que permite visualizar en tiempo real la animación de un diseño CAD con idénticas propiedades y características del prototipo.

### 5.1 APLICACIÓN MÓVIL

El sistema de mando del prototipo diseñado se realiza por medio de un dispositivo IMU, pero se decidió utilizar el IMU de un *Smartphone* teniendo en cuenta los siguientes aspectos:

- Dinamismo entre la comunicación del usuario con la plataforma.
- Uso de las tecnologías de la Información y Comunicación (TIC's).
- Facilidad en el desarrollo de aplicaciones móviles ejecutadas en *smartphones*.

La aplicación móvil que se diseñó realiza las lecturas de los sensores de aceleración e inclinación del celular en el cual es ejecutada y envía dichas medidas a través de comunicación bluetooth al computador personal donde se ejecuta el programa de control del prototipo de plataforma Stewart en Simulink. Utilizando la aplicación móvil también se visualiza en tiempo real los cambios de aceleración y de ángulo del celular respecto a su sistema coordinado. La aplicación móvil para Android se diseñó y programó en una innovadora aplicación online desarrollada por el Instituto Tecnológico de Massachusetts llamada *App Inventor*<sup>19</sup> que hace posible crear

---

<sup>19</sup> <http://ai2.appinventor.mit.edu/>

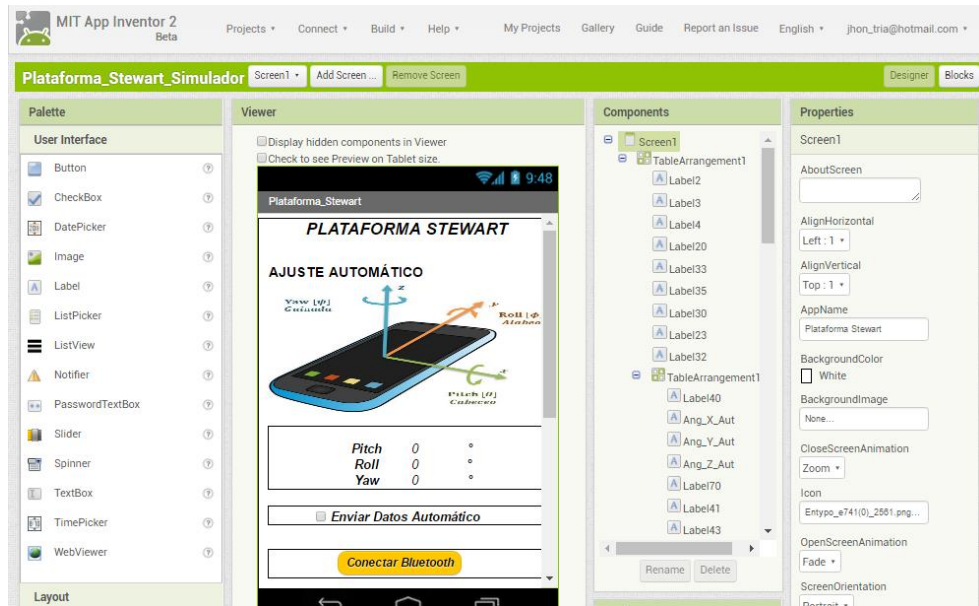
aplicaciones Android en un lenguaje visual que consiste básicamente en arrastrar y soltar bloques de construcción.

La interfaz de *App Inventor* consiste en 2 entornos:

- *Designer* que es la Interfaz de diseño, en la cual se determinan todas las características visuales de la aplicación.
- *Blocks* que es la interfaz de programación a través de bloques, en la cual se especifican las características y recursos del *smartphone* que utilizará la aplicación móvil que se creará.

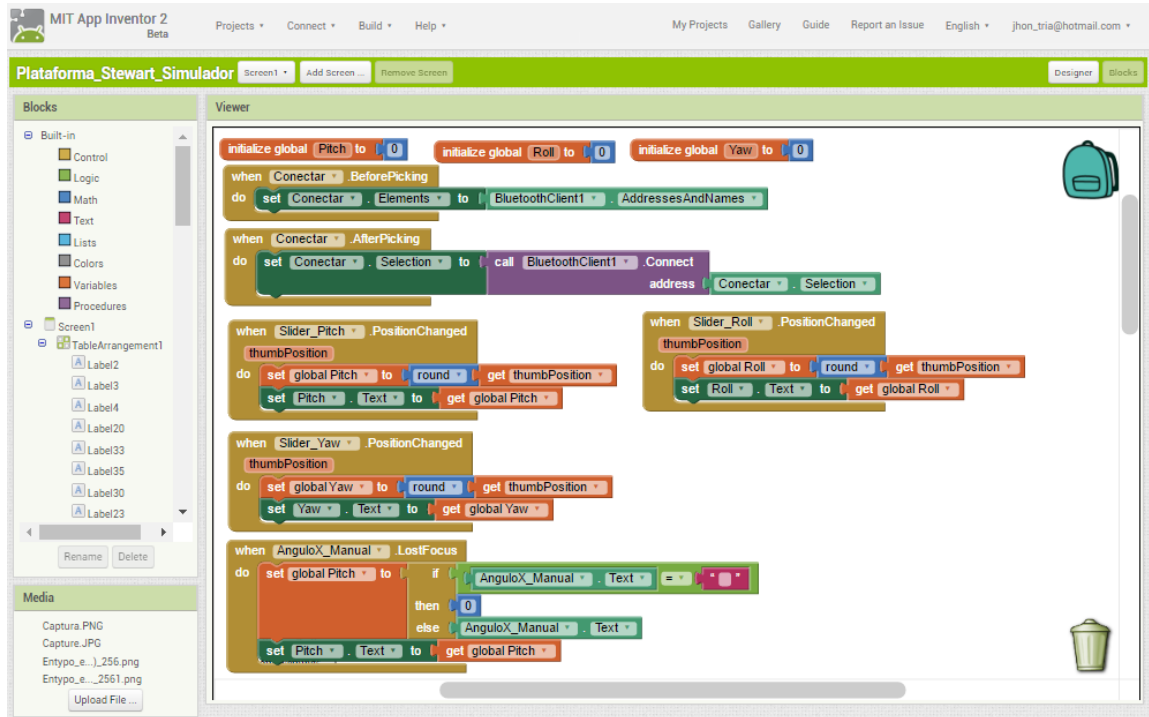
**5.1.1 Diseño de la Interfaz de la Aplicación.** La aplicación móvil nombrada Plataforma Stewart contiene una serie de botones, textos, imagen, *sliders* y cuadros de chequeo de los cuales algunos se pueden apreciar en la Figura 65. El diseño visual de la aplicación es sencillo y consiste básicamente en presentar la información que será enviada al programa de control en Simulink.

**Figura 65.** Diseño de la interfaz de la aplicación móvil



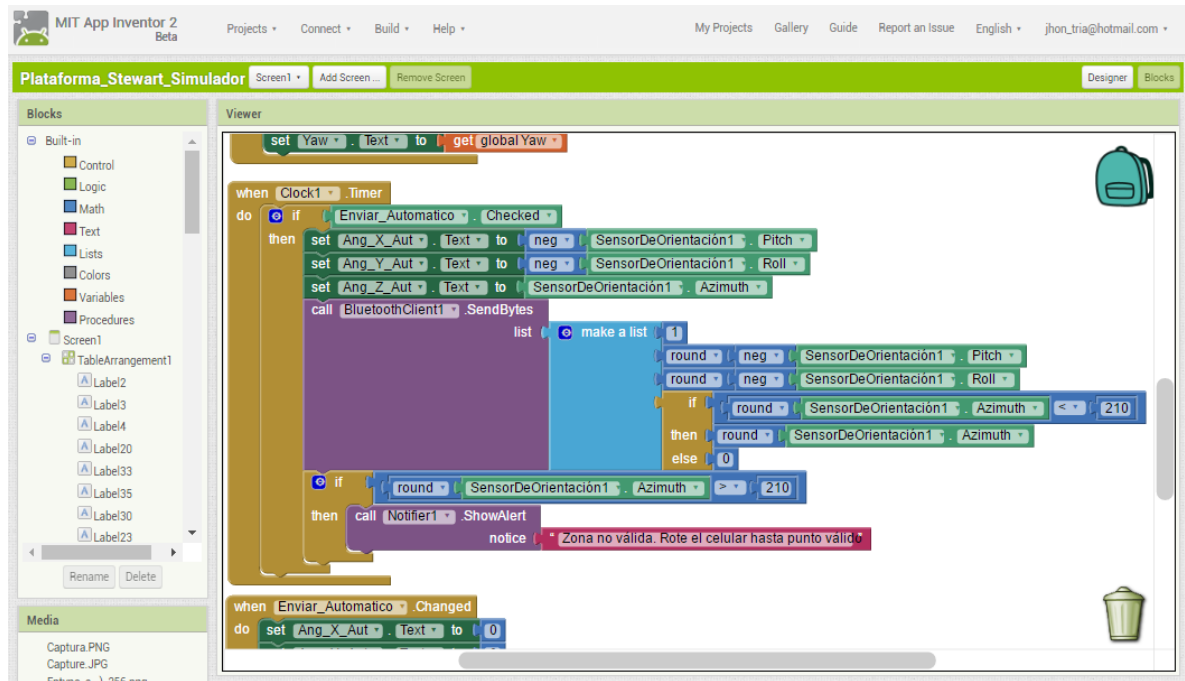
**5.1.2 Programación de la Aplicación.** El lenguaje de programación de App Inventor, es un lenguaje gráfico (Ver Figura 66) basado en bloques y orientado a eventos, característica fundamental que se tuvo en cuenta para seleccionar este compilador para la creación la aplicación móvil Plataforma Stewart. La aplicación móvil está programada para que el usuario interactúe con el prototipo por medio de dos modos: en ajuste automático y ajuste manual.

Figura 66. Programación por bloques de la aplicación móvil



**5.1.2.1 Modo Automático.** En la programación del modo automático ilustrada en la Figura 67 se realiza la lectura de los valores de aceleración y orientación del celular con respecto a su sistema coordinado y se envían estos datos por medio del módulo *bluetooth*.

Figura 67. Bloques de programación del modo automático de la aplicación móvil



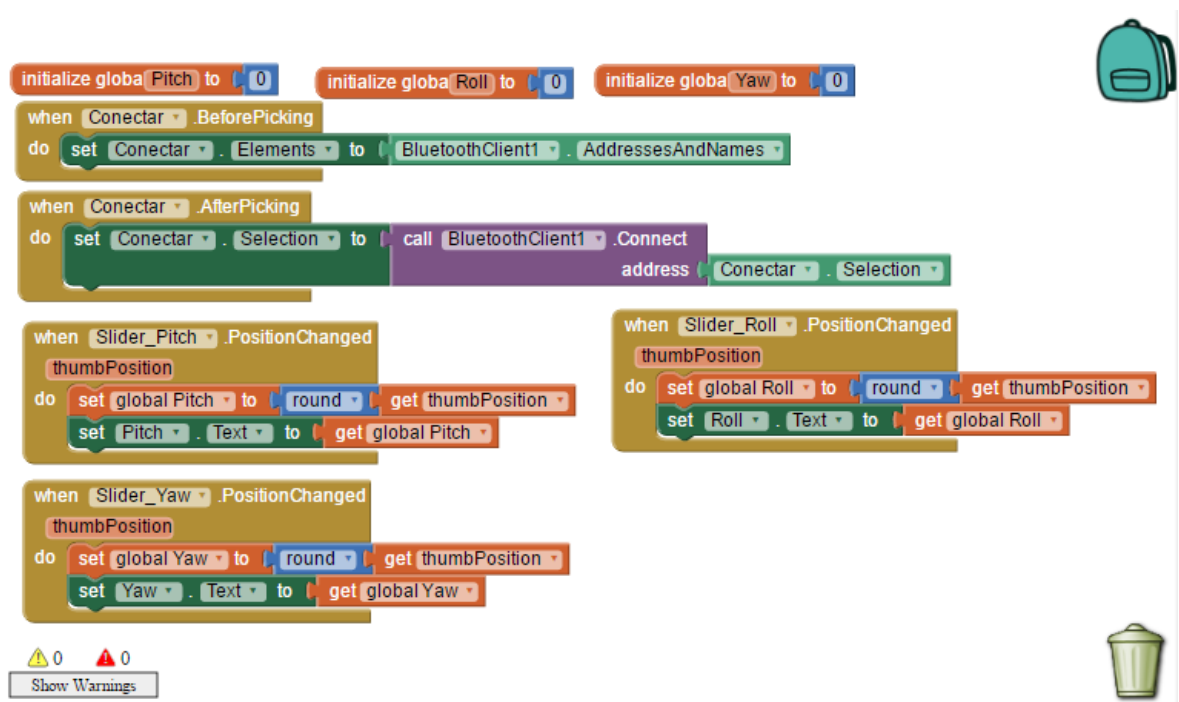
Los bloques *set Acel\_X\_Aut* ilustrados en la **¡Error! No se encuentra el origen de a referencia.** permiten visualizar la aceleración en la pantalla del celular dentro de la aplicación móvil y el bloque *BluetoothClient1* envía las señales vía bluetooth. El primer valor enviado en el bloque *BluetoothClient1* hace referencia al modo:

- Modo automático: 1
- Modo manual: 2

**5.1.2.2 Modo Manual.** En la programación del modo manual se obtienen los valores especificados por el usuario de cada variable a través de una barra deslizante o *slider* cuyos datos son leídos y enviados vía bluetooth. Inicialmente en la aplicación cada una de las variables se encuentra definida como 0, luego para cualquier cambio en la barra deslizante, se lee el dato y se guarda en las variables

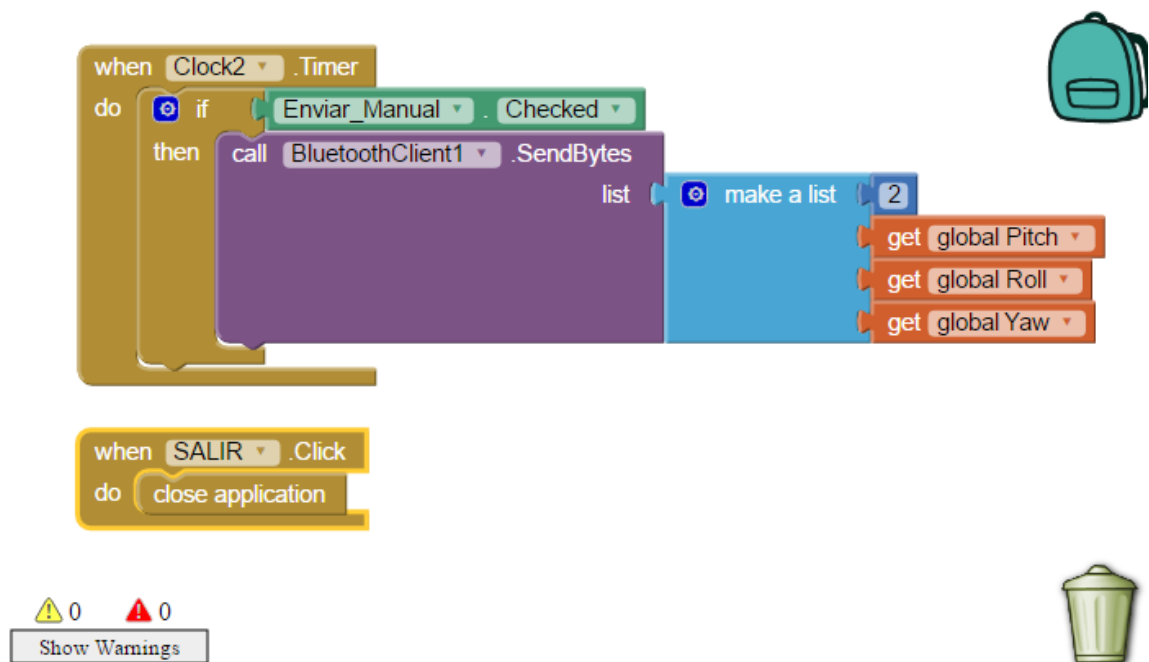
antes mencionadas tal como se ilustra en la Figura 68. La información enviada al programa de control es impresa en la pantalla de la aplicación para que el usuario tenga presente el valor que configuro como deseado. Las unidades de cada variable establecida por el usuario son leídas como grados (°) para los cambios de ángulo de la plataforma.

**Figura 68.** Bloques de programación del modo manual de la aplicación móvil



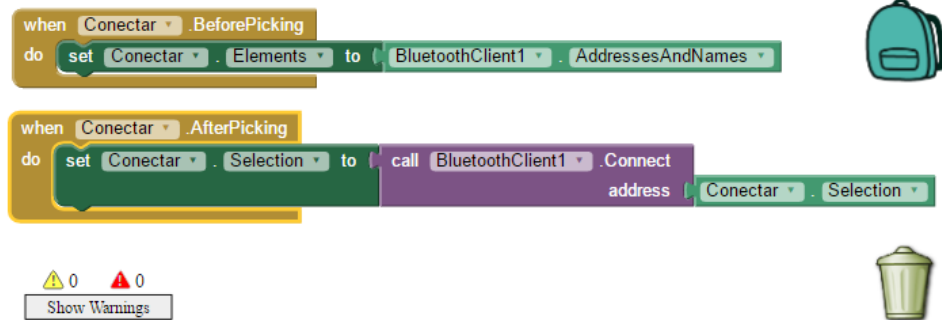
Finalmente, se envían los datos vía bluetooth (Ver Figura 69) a través de un segundo contador denominado *Clock 2*. El primer valor de la lista hace referencia al modo, variable definida en la sección anterior... Véase sección 5.1.2.1..., es decir, 2 para modo manual.

Figura 69. Envío de las señales Modo Manual



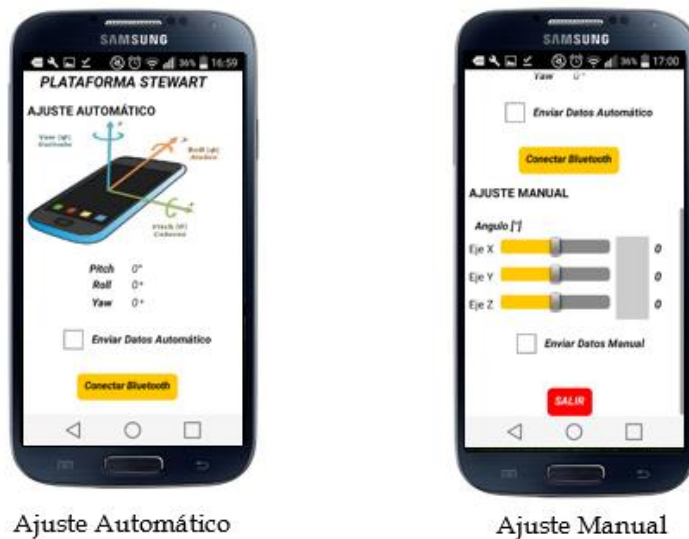
**5.1.3 Comunicación Bluetooth.** En el compilador de aplicaciones móviles el dispositivo *bluetooth* instalado en el celular es llamado *BluetoothClient*, este dispositivo se inicia al hacer clic en la aplicación móvil sobre el botón “Conectar Bluetooth” buscando las direcciones y nombres de otros dispositivos disponibles que cuenten con comunicación *bluetooth* y se conecta una vez se selecciona la dirección del computador; los bloques de programación para esta conexión se ilustran en la Figura 70. Para ver el *bluetooth* del computador en la aplicación es necesario agregarlo como un dispositivo en el celular antes de ingresar a la misma.

Figura 70. Bloques de la conexión Bluetooth



**5.1.4 Funcionamiento de la Aplicación.** Una vez es ejecutado el programa de control en Simulink se debe iniciar la transmisión de datos desde la aplicación móvil, para ello se conecta el *smartphone* vía *bluetooth* con el computador y una vez establecida la comunicación entre ambos dispositivos, se realiza la transmisión de datos chequeando o cliqueando en el cuadro Enviar Datos Automático o Enviar Datos Manual (Ver Figura 71), según sea el caso.

Figura 71. App Android Plataforma Stewart



## 5.2 EMULADOR

Haciendo uso del modelo CAD del prototipo de plataforma Stewart expuesto en... la sección 3.1.2... se creó un emulador de movimiento en tiempo real con el fin de crear una interfaz HDMI que permitiera visualizar virtualmente el comportamiento cinemático del prototipo. El emulador se creó haciendo uso de la interfaz SIMMECHANICS<sup>20</sup>, la cual es una librería de MATLAB / SIMULINK.

El emulador permitió analizar características de diseño, características propias del movimiento y límites dimensionales del prototipo de plataforma Stewart diseñado en este proyecto, permitiendo realizar mejoras en:

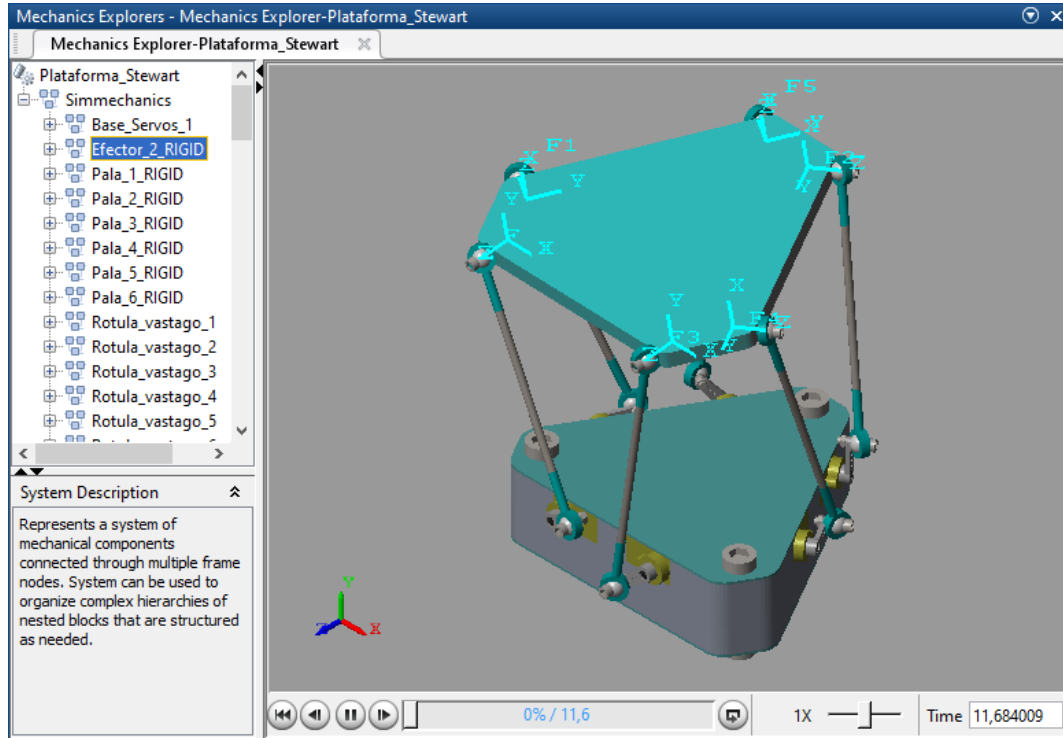
- Las dimensiones de la plataforma base y móvil.
- Las posiciones de los servomotores en la plataforma base.
- La ubicación de los puntos de unión de la base, dicha posición se optimizó definiendo los pares de puntos lo más cercano entre ellos.
- La ubicación de los puntos de unión del efector, optimizado dejando los puntos lo más cercano al borde de cada cara externa del efector.
- Análisis del espacio de trabajo del prototipo.

Producto de las optimizaciones de diseño logradas con el emulador de movimiento (ver Figura 72) se obtuvo una mayor movilidad de las rotulas lo que se traduce en mayor movilidad del efector del prototipo.

---

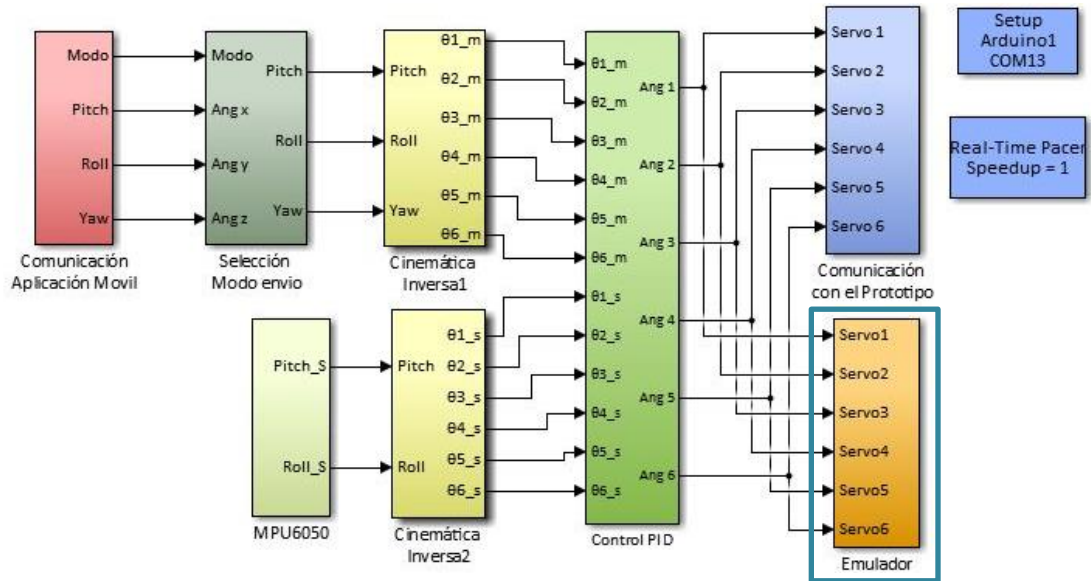
<sup>20</sup> SIMMECHANICS es un complemento de MatLab/Simulink que permite usar archivos CAD de un modelo y alimentarlo con información adquirida o generada en el ambiente de trabajo de MatLab. Para información más detallada Ver Anexo G.

**Figura 72.** Emulación del prototipo de plataforma Stewart en el ambiente SIMMECHANICS.



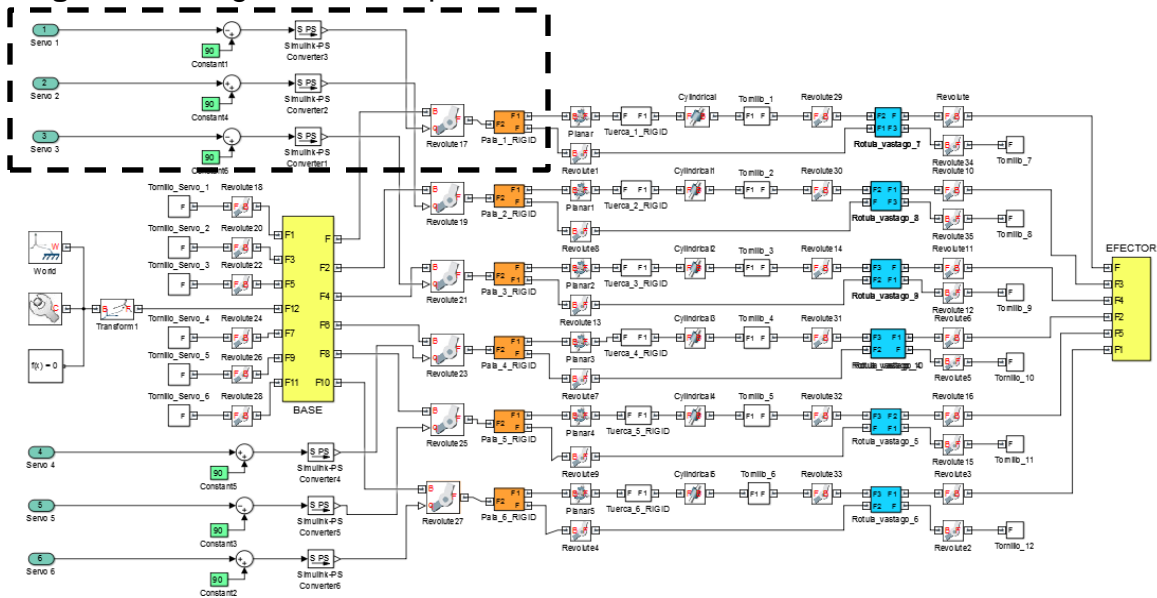
El subsistema de programación del emulador, identificado en la Figura 73 con el color naranjado, representa el diseño del prototipo de plataforma Stewart en SolidWorks mediante representación en bloques de la librería SimMechanics, y hace posible manipular cada elemento del modelo CAD que esté vinculado a algún movimiento. Para el caso particular del prototipo de este proyecto la señal de entrada al bloque de programación del emulador es la posición de la pala de cada servomotor que alimenta el prototipo real.

**Figura 73.** Programa general de la plataforma Stewart.



En la Figura 74 se representa la programación del bloque Emulador del programa de control del prototipo de plataforma Stewart representado en la Figura 73. En la Figura 74 se representa la sección de recepción de señales de posición angular mediante una línea a trazos y se explicará más adelante.

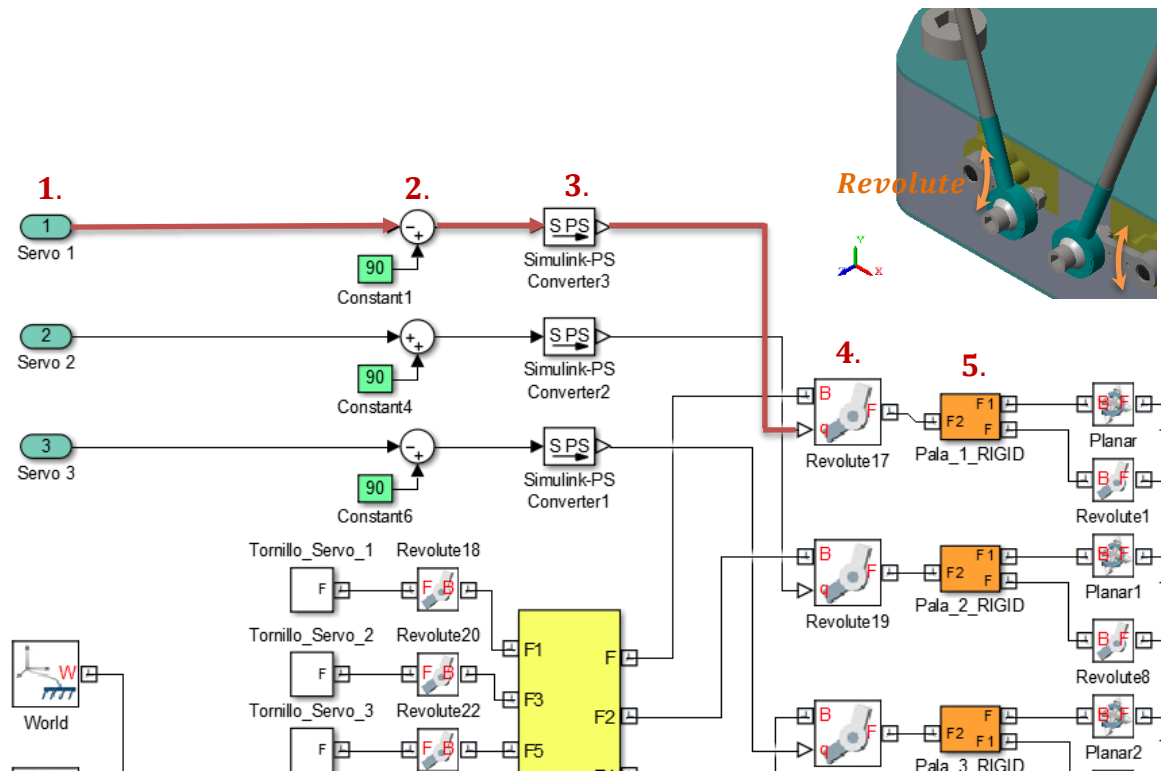
**Figura 74.** Diagrama de bloques del subsistema Simmechanics.



La alimentación del bloque de programación del emulador se da por los siguientes pasos representados en la Figura 75:

1. Señal de entrada real del ángulo correspondiente al servomotor.
2. Se desfasa la señal de entrada 90 grados para ajustar la posición del servomotor al cero del prototipo.
3. La señal desfasada pasa a través de un convertidor de señales que hace posible manipular la con los bloques de programación de SimMechanics.
4. La señal transformada actúa en el bloque de revolución *Revolute* de la pala respectiva. Para ello es importante configurar en el bloque *Revolute* que el movimiento será una señal de entrada externa.
5. La señal de revolución actúa en la pala del modelo CAD a la que se conecte el flujo de programación.

**Figura 75.** Diagrama de bloques del subsistema Simmechanics



## 6. RESULTADOS

### 6.1 ESPACIO DE TRABAJO

En robótica el Espacio de Trabajo definido por Cao<sup>21</sup> es el grupo de puntos que pueden ser alcanzados por el efector final del robot, o dicho de otra manera, es el espacio en el cual la plataforma puede trabajar. El estudio del espacio de trabajo de un robot ha sido estudiado por varios autores y su definición puede llegar a ser mucho más extensa.

Se realizó un análisis de espacio de trabajo del prototipo con el fin de determinar el entorno donde el mecanismo trabajará y el alcance del efector. Para determinar el espacio de trabajo se utilizó el método de discretización del espacio que consiste en encontrar los puntos límites del efector a través de un proceso iterativo. Para ello se realizó el trazado del espacio de trabajo de traslación, espacio de trabajo de orientación y finalmente el espacio de trabajo diestro que está compuesto por todos los puntos en el espacio que pueden ser alcanzados por la plataforma móvil. En el espacio de trabajo diestro se analizan cada una de las variables X, Y y Z por separado y finalmente en conjunto.

Aprovechando la facilidad de programación en el entorno de MatLab/Simulink se realizó el trazado del espacio de trabajo del prototipo en gráficos 3D a partir de los siguientes comandos digitados en el *command window* de MatLab:

```
plat=[P1;P2;P3;P4;P5;P6;P1];  
plot3(plat(:,1),plat(:,2),plat(:,3),'k','LineWidth',2);  
grid on  
hold on  
xlabel('X [mm]');
```

---

<sup>21</sup> **Yi Cao, Ke Lu, Xiujuan Li and Zang** (2011). Accurate Numerical Methods for Computing 2D and 3D Robot Workspace [Journal] International Journal of Advanced Robotic Systems: INTECH, August 2011. – 6: Vol. VIII – pp. 1 – 13.

```

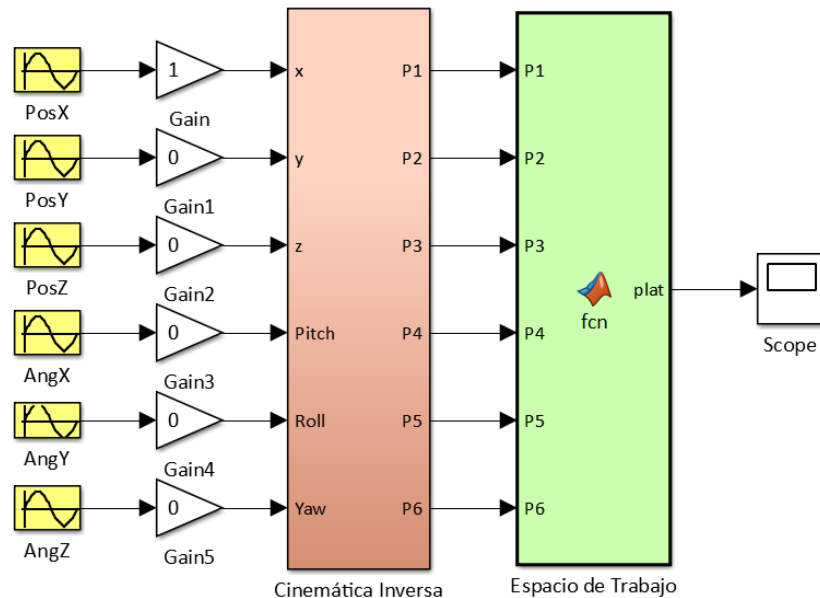
ylabel('Y [mm]');
zlabel('Z [mm]');
title('Espacio de Trabajo de la Plataforma Stewart')

```

Donde el arreglo *plat* contiene las coordenadas del efector obtenidas a partir de la cinemática inversa y el comando *plot3* permite realizar un gráfico en 3D a través del trazado de líneas entre cada uno de los puntos del arreglo *plat*.

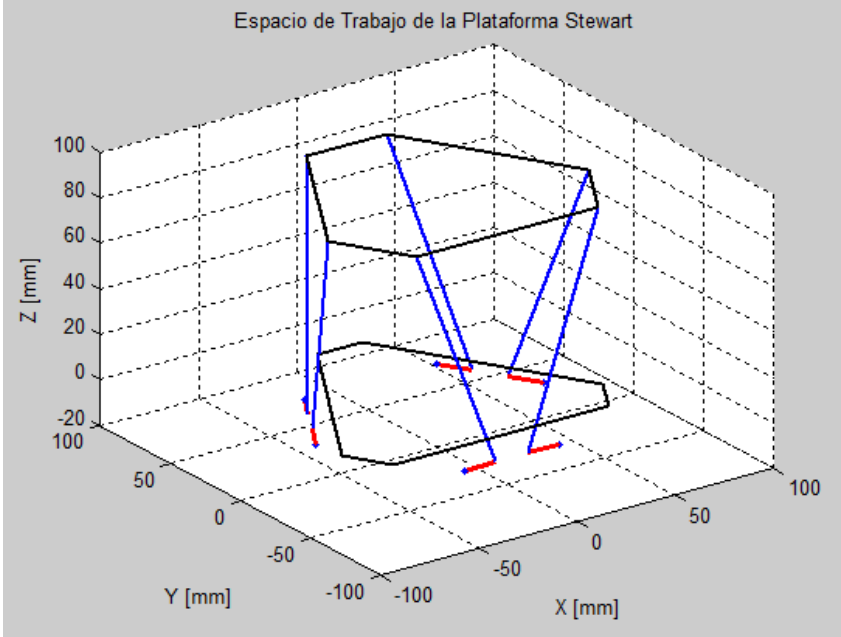
La Figura 76 muestra el diagrama de bloques que crea el trazado de las posiciones del efector para una entrada sinusoidal con amplitud máxima de 50 (dado que tanto los desplazamientos como las rotaciones no superan los 50 [mm] y 50° respectivamente) y una frecuencia de 0.07 rad/s, magnitud que permite obtener un trazado óptimo. Los bloques de ganancia en el grafico permiten habilitar o deshabilitar las señales de entrada para cada variable.

**Figura 76.** Diagrama de bloques del programa utilizado para determinar el espacio de trabajo del prototipo de plataforma diseñado.



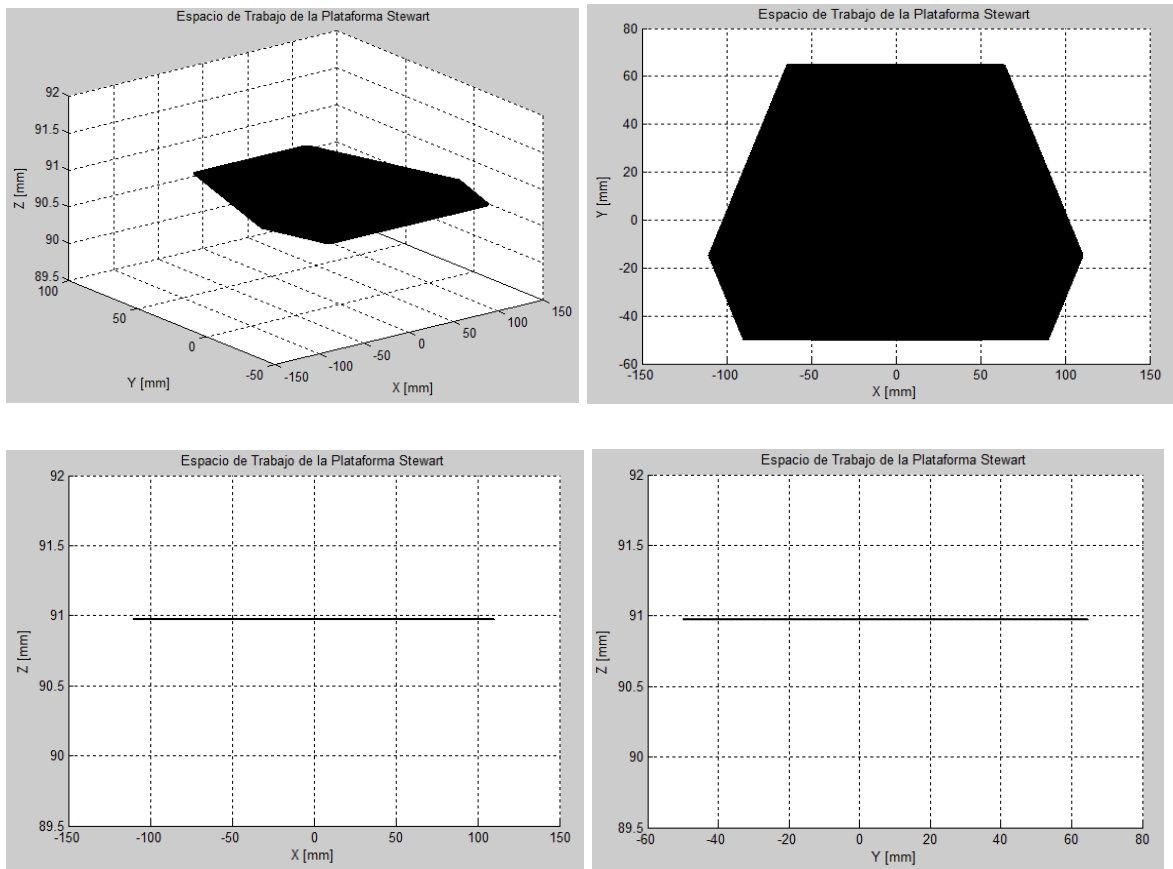
**6.1.1 Espacio de Trabajo de Traslación.** El espacio de trabajo de traslación está compuesto por todos aquellos puntos en el espacio que son alcanzados por el efector. A partir del trazado de cada coordenada de los desplazamientos del modelo simplificado de la plataforma móvil (ver Figura 77) usando el programa de la Figura 76, se obtienen las coordenadas límites del movimiento. A continuación se determina el espacio de trabajo para cada eje en el espacio dimensional.

**Figura 77.** Efector en posición inicial o reposo



**6.1.1.1 Desplazamiento lineal X.** A continuación se presentan los resultados gráficos del desplazamiento lineal en X de la plataforma en la Figura 78, resultando los valores consignados en la Tabla 5.

**Figura 78.** Espacio de trabajo del prototipo para desplazamiento en X

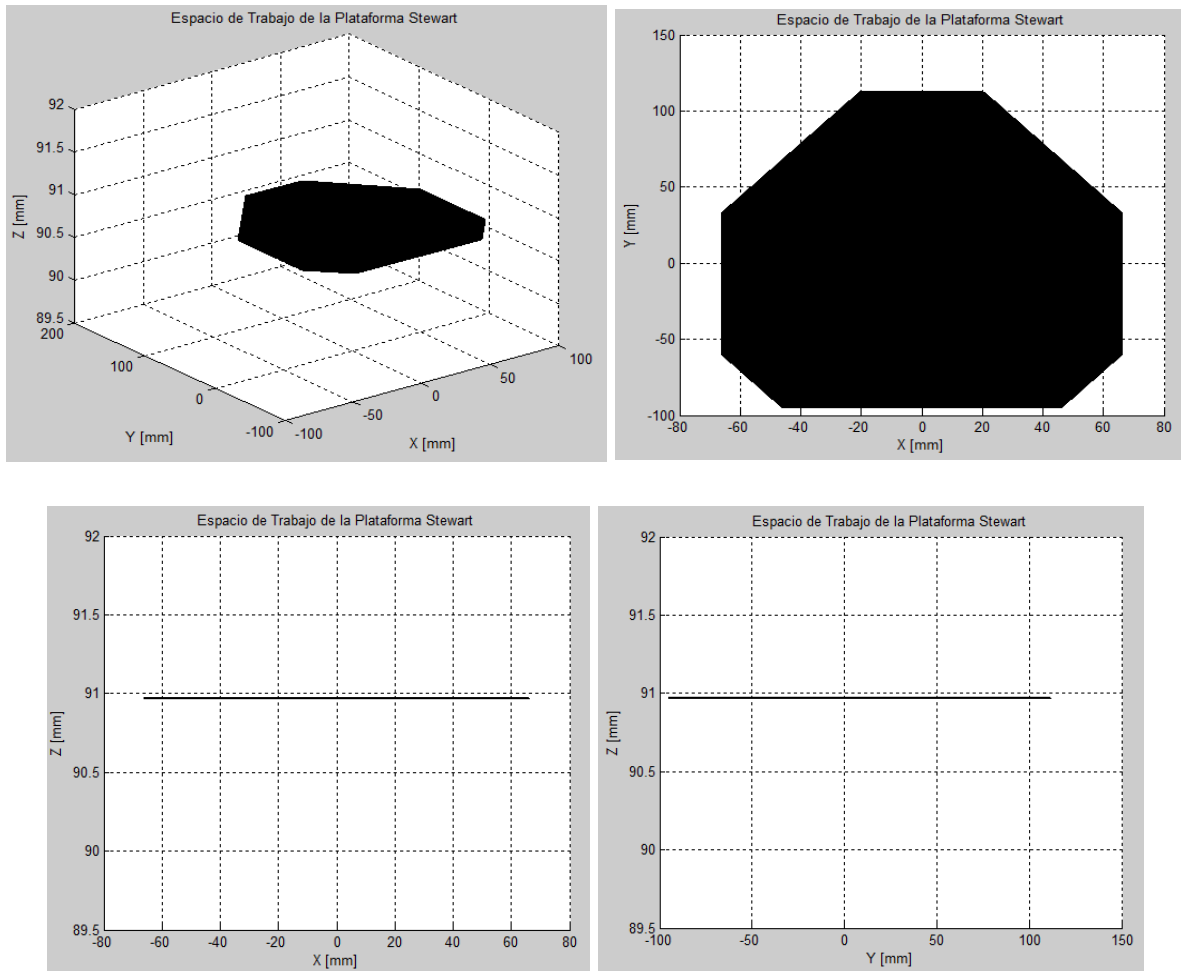


**Tabla 5.** Espacio de trabajo desplazamiento en X

Variable	Máximo [mm]	Mínimo [mm]
x	109,6	-109,1
y	64,8	-50
z	90,97	

**6.1.1.2 Desplazamiento lineal Y.** A continuación se presentan los resultados gráficos del desplazamiento lineal en Y de la plataforma en la Figura 79, resultando los valores consignados en la Tabla 6.

**Figura 79.** Espacio de trabajo del prototipo para desplazamiento en Y

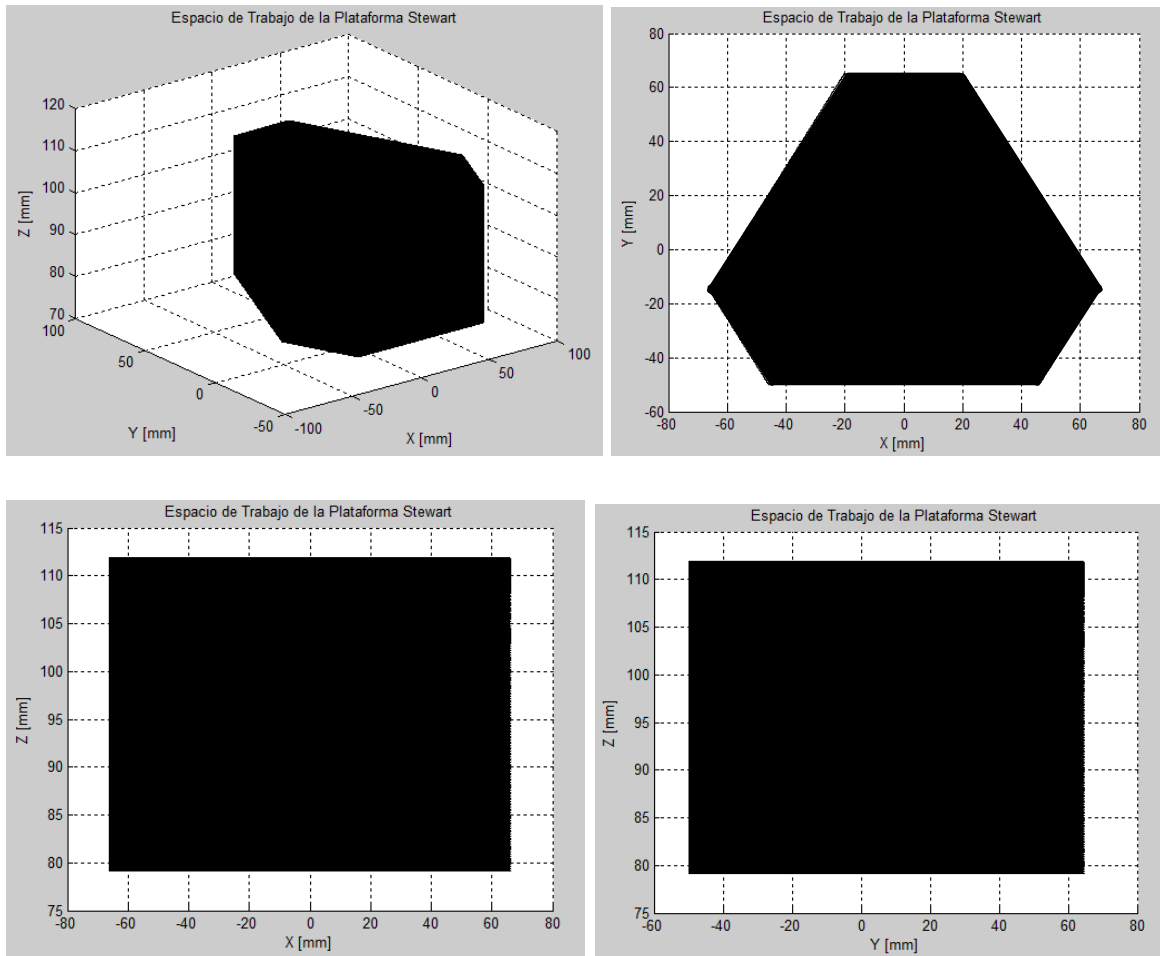


**Tabla 6.** Espacio de trabajo desplazamiento en Y

Variable	Máximo [mm]	Mínimo [mm]
x	66,28	-66,28
y	110,8	-93,44
z	90,97	

**6.1.1.3 Desplazamiento lineal Z.** A continuación se presentan los resultados gráficos del desplazamiento lineal en Z de la plataforma en la Figura 80, resultando los valores consignados en la Tabla 7.

**Figura 80.** Espacio de trabajo del prototipo para desplazamiento en Z



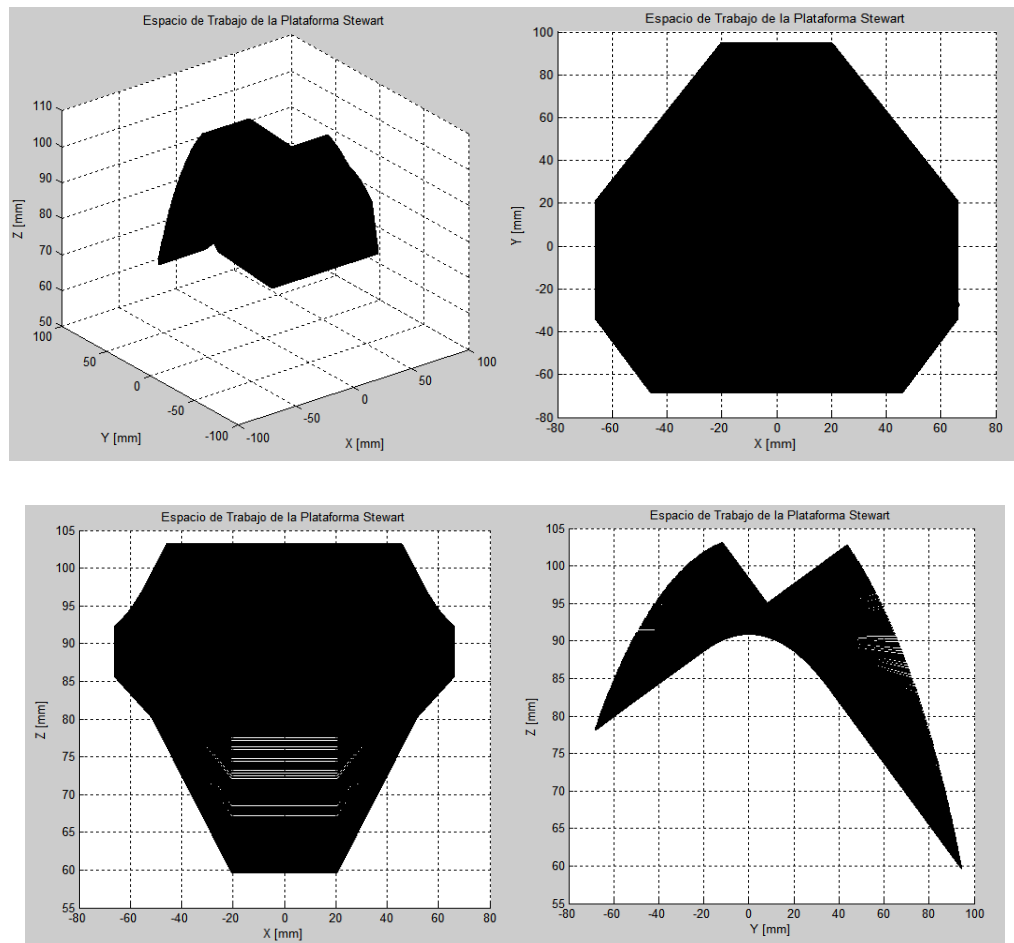
**Tabla 7.** Espacio de trabajo desplazamiento en Z.

Variable	Máximo [mm]	Mínimo [mm]
x	66,28	-66,28
y	64,8	-50
z	111,8	79,18

## 6.1.2 Espacio de Trabajo De Rotación

**6.1.2.1 Desplazamiento angular X o Pitch.** A continuación se presentan los resultados gráficos del desplazamiento angular en X de la plataforma en la Figura 81, resultado los valores consignados en la Tabla 8.

**Figura 81.** Espacio de trabajo del prototipo para desplazamiento angular en X.

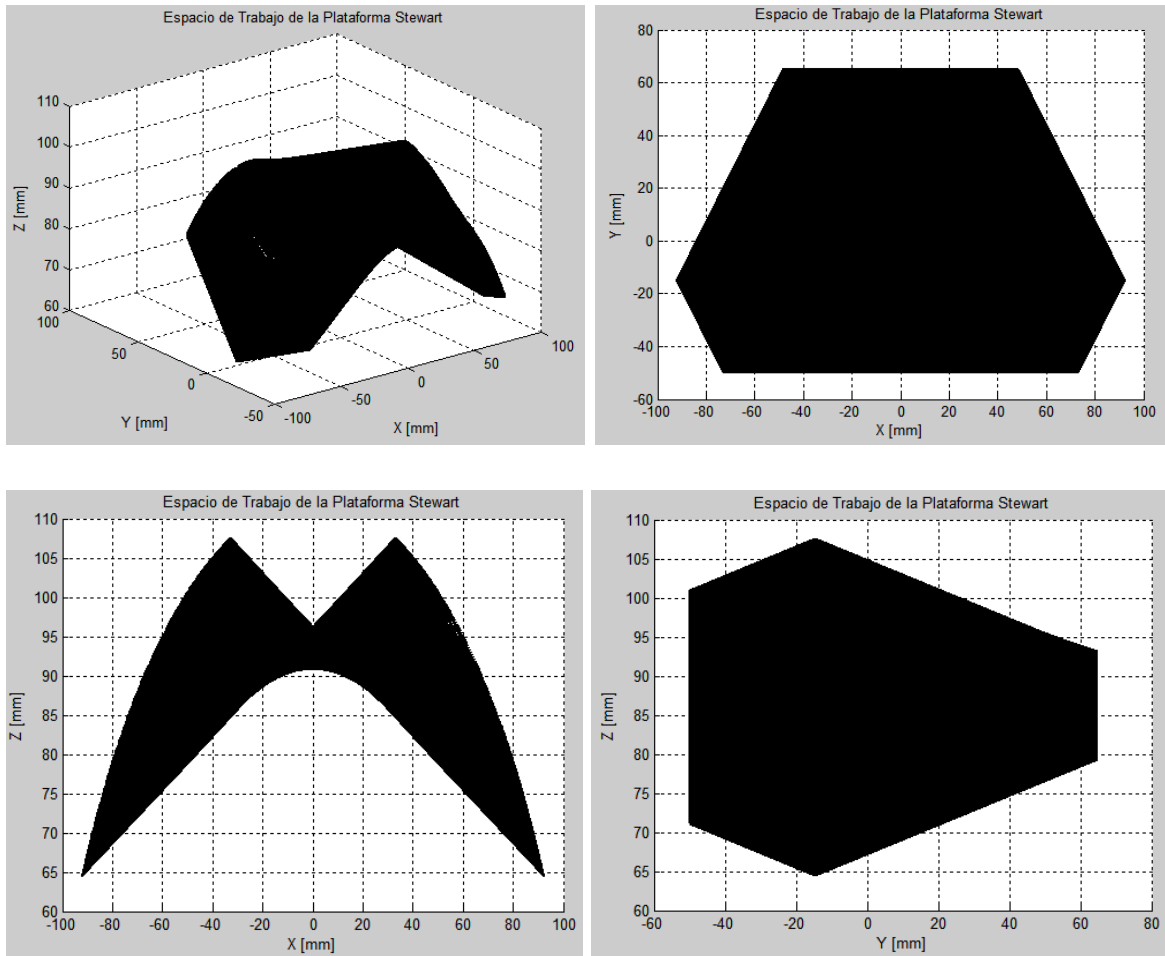


**Tabla 8.** Espacio de trabajo desplazamiento angular en X.

Variable	Máximo [mm]	Mínimo [mm]
x	66.28	-66.28
y	93,9	-67,74
z	103	59,81

**6.1.2.2 Desplazamiento angular Y o Roll.** A continuación se presentan los resultados gráficos del desplazamiento angular en Y de la plataforma en la Figura 82, resultando los valores consignados en la Tabla 9.

**Figura 82.** Espacio de trabajo del prototipo para desplazamiento angular en Y.

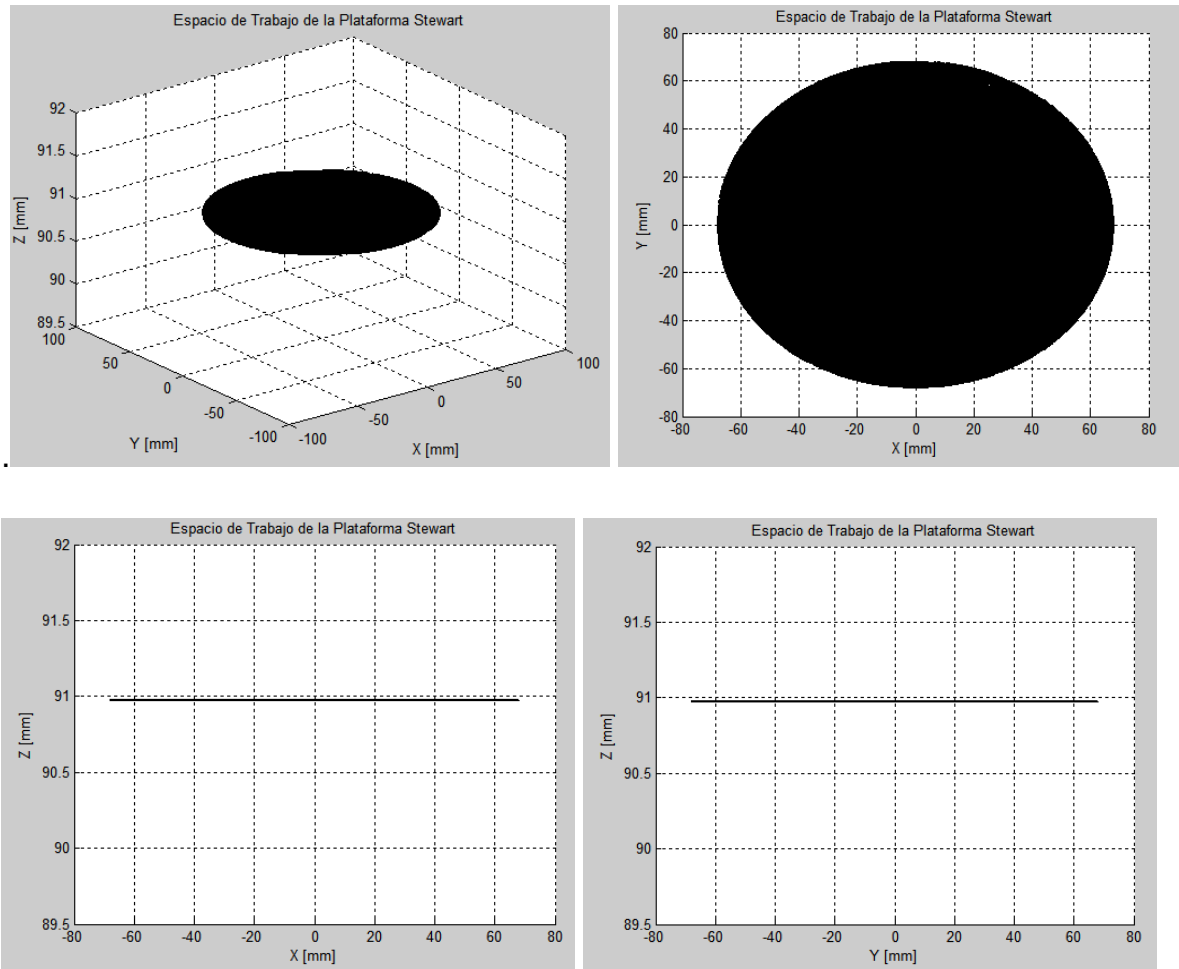


**Tabla 9.** Espacio de trabajo desplazamiento angular en Y

Variable	Máximo [mm]	Mínimo [mm]
x	92,05	-92,05
y	64,8	-50
z	107,5	64,7

**6.1.2.3 Desplazamiento angular Z o Yaw.** A continuación se presentan los resultados gráficos del desplazamiento angular en Z de la plataforma en la Figura 83, resultando los valores consignados en la Tabla 10.

**Figura 83.** Espacio de trabajo del prototipo para desplazamiento angular en Z

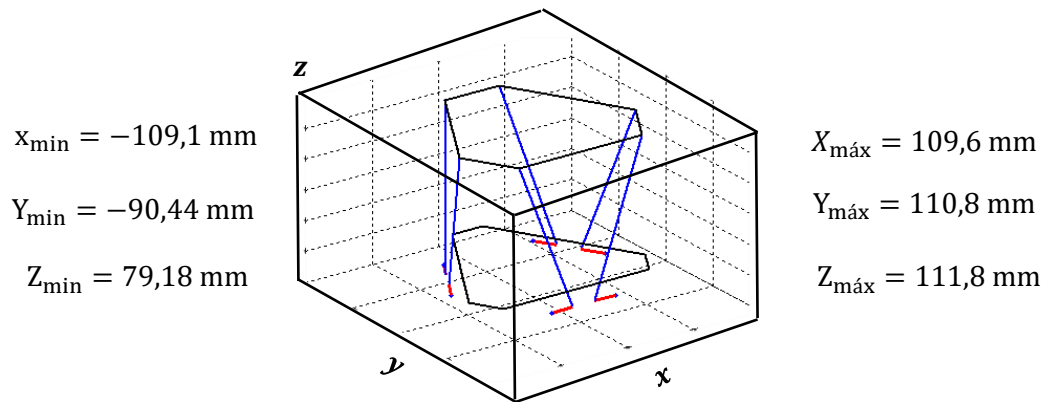


**Tabla 10.** Espacio de trabajo desplazamiento angular en Z

Variable	Máximo [mm]	Mínimo [mm]
x	67.86	-67.86
y	67,86	-67,86
z	90,97	

**6.1.3 Espacio de Trabajo Diestro.** Grosso modo el espacio de trabajo del prototipo de plataforma Stewart servoactuada sería un cubo con las coordenadas límites expuestas en la Figura 84.

**Figura 84.** Espacio de trabajo de la Plataforma Stewart



$$V = (X_{\max} - X_{\min}) * (Y_{\max} - Y_{\min}) * (Z_{\max} - Z_{\min})$$

$$V = 1435,64 \text{ cm}^2$$

**6.1.4 Desplazamientos Y Rotaciones Límites.** A partir del espacio de trabajo tanto de traslación como de rotación se obtuvieron los desplazamientos y los ángulos máximos del prototipo y se muestran en la Tabla 11.

**Tabla 11.** Desplazamientos y rotaciones límites reales en la plataforma Stewart diseñada.

EJE	MÁXIMO	MÍNIMO
X	44 [mm]	-44 [mm]
Y	47,1 [mm]	-45 [mm]
Z	20,8 [mm]	-11,8 [mm]
$\angle$ X	12,4 °	-22,3 °
$\angle$ Y	19°	-19°
$\angle$ Z	43,3°	-43,3°

## 6.2 COMPARACIÓN CON UN AVIÓN

Para el vuelo de aeronaves los ángulos de despegue y aterrizaje, es decir, ángulos Pitch, dependen entre otros factores externos de la velocidad de despegue, el peso de la aeronave y la velocidad del viento circundante, por lo anterior es difícil estimar un ángulo general para todas las aeronaves. Sin embargo, según fuentes no oficiales en la web<sup>22</sup>, el ángulo de despegue no debe superar los 15° (Pitch positivo) y el ángulo de aterrizaje se realiza máximo a 3,2° (Pitch negativo). La información anterior se compara en la Tabla 12 con los rangos del ángulo Pitch obtenidos con el prototipo y se concluye que el diseño del mismo es aceptable para la simulación de vuelo.

**Tabla 12.** Comparación de ángulos Pitch ejecutados en vuelo y alcanzados por prototipo desarrollado.

<i>Ángulo</i>	<i>± AERONAVE</i>	<i>± PROTOTIPO</i>
<i>Pitch</i>	-3,2° a 20°	-22,3° a 12.4°
<i>Roll</i>	±15° a ± 20°	±19°

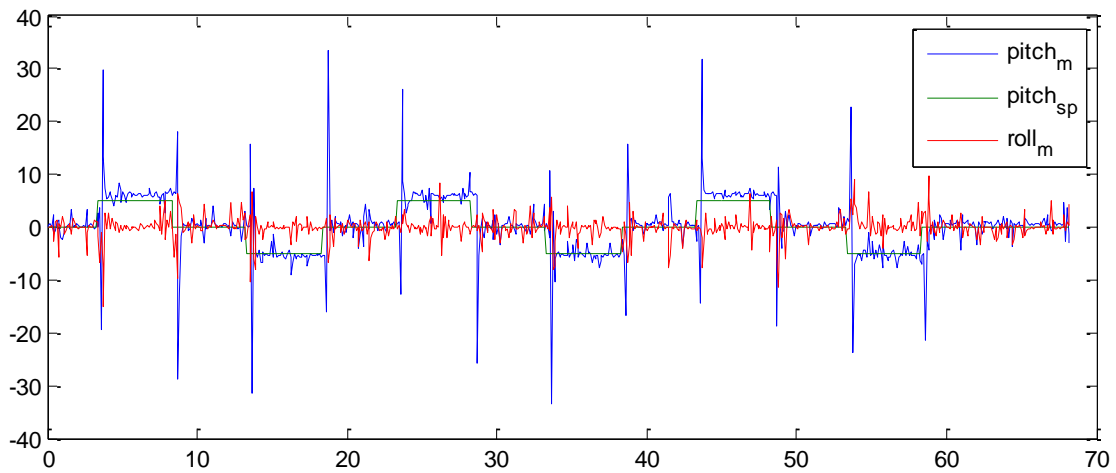
## 6.3 TIPOLOGÍA DE LAS LECTURAS DEL SENSOR IMU

Los picos y ruido (ver Figura 85) presentados en... la sección 4.4.1.1... en las figuras de lectura de ángulos del efector móvil y la posición de los servomotores eran producidos por el ruido excesivo de las lecturas de los ángulos de posición (Pitch y Roll) realizadas únicamente con la información suministrada por el acelerómetro del dispositivo IMU.

---

<sup>22</sup> <https://www.xataka.com/seguridad/variacion-la-inclinacion-de-los-aviones-en-el-aterrizaje-la-idea-que-se-prueba-en-londres-para-reducir-el-ruido>

**Figura 85.** Picos de lectura en los ángulos Pitch y Roll del sensor acelerómetro

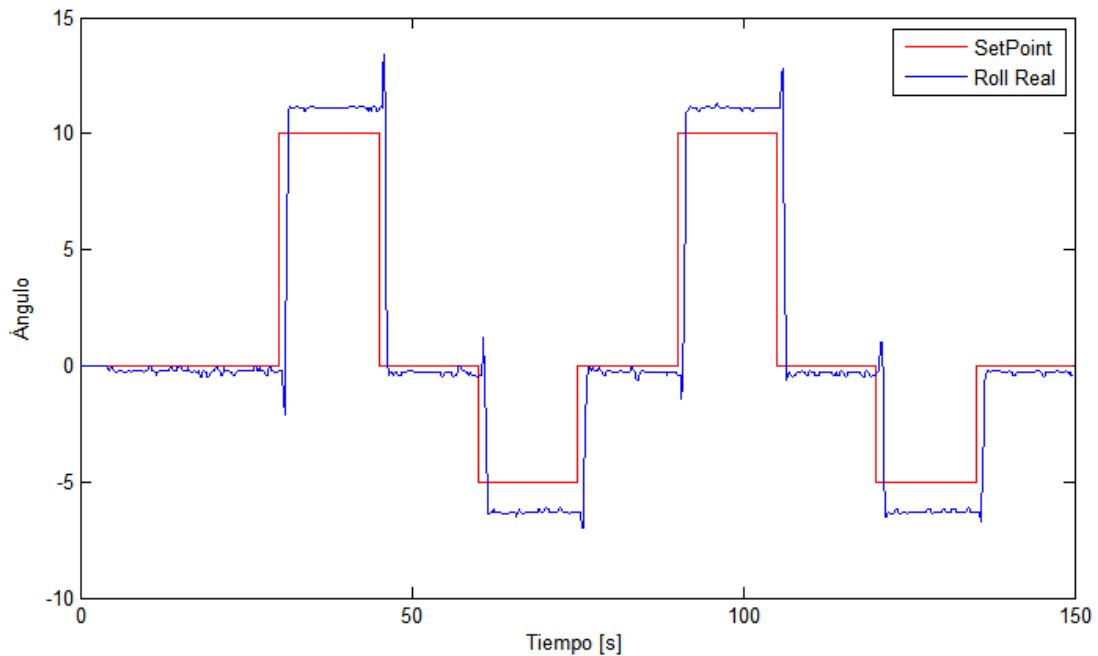


Los ruidos presentados en la Figura 85 fueron eliminados con la implementación de un filtro complementario entre las señales del acelerómetro y las del giroscopio, cuya fusión se ilustra en la Figura 86 y se implementó utilizando la siguiente ecuación:

$$\text{Ángulo} = \alpha * (\text{Ángulo}_{\text{Giroscopio}}) + (1 - \alpha) * (\text{Ángulo}_{\text{Acelerómetro}})$$

Donde el valor de la constante  $\alpha$  es de 0.94 y representa el nivel de credibilidad que se tiene a las lecturas realizadas por el giroscopio.

**Figura 86.** Lecturas fusionadas de posición angular del efector móvil.

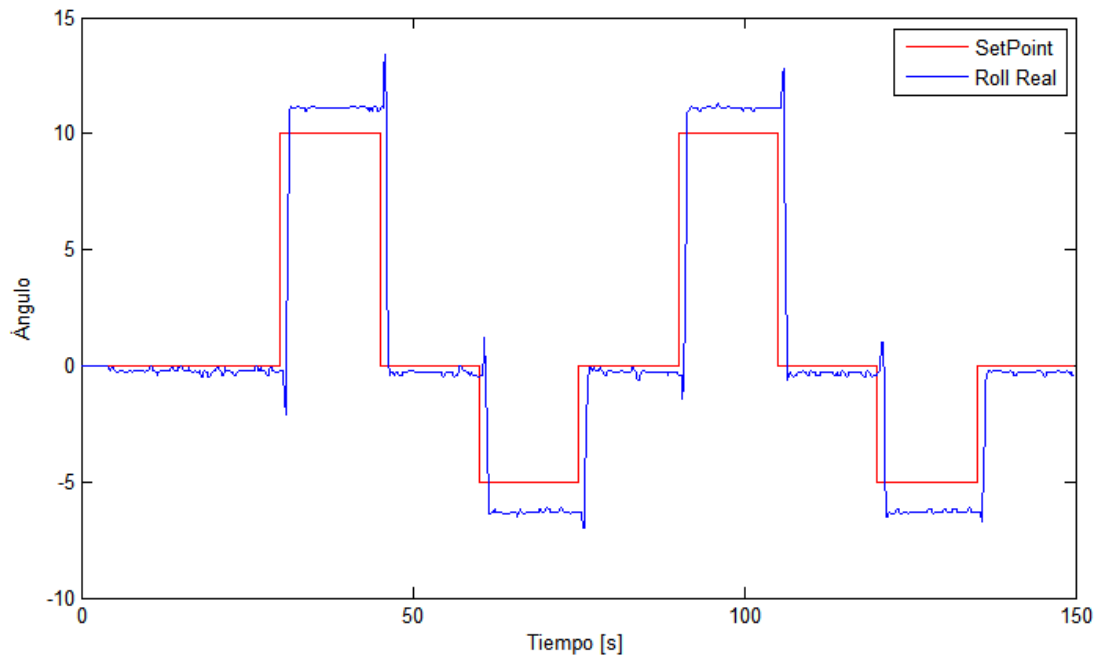


El filtro complementario como su nombre lo indica complementa las medidas de ángulos del acelerómetro y el giroscopio eliminando el error a corto plazo en el acelerómetro generado por vibraciones y el *drift* o deriva que se presenta en los giroscopios después de un tiempo determinado.

#### 6.4 CONTROL EN LAZO ABIERTO

Inicialmente se testeó el prototipo de plataforma Stewart en control de lazo abierto con el fin de observar el comportamiento, para dicho test se generó una señal de posición angular en Roll y el valor del Pitch se mantuvo constante en cero resultando la gráfica de la Figura 87.

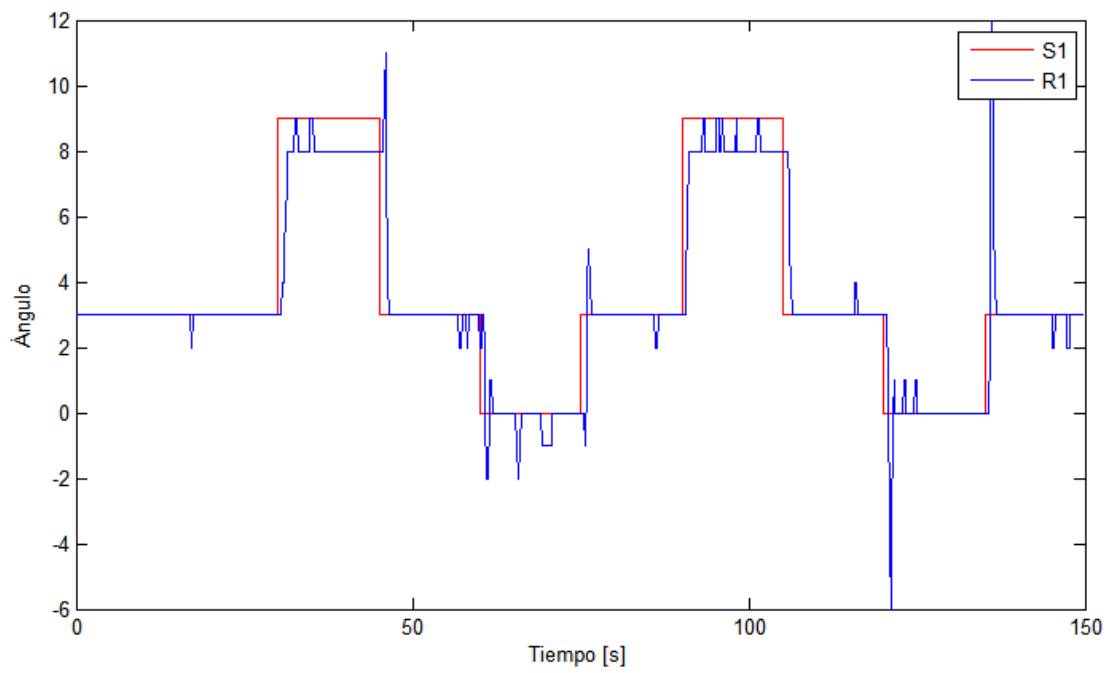
**Figura 87.** Comportamiento prototipo en control lazo abierto



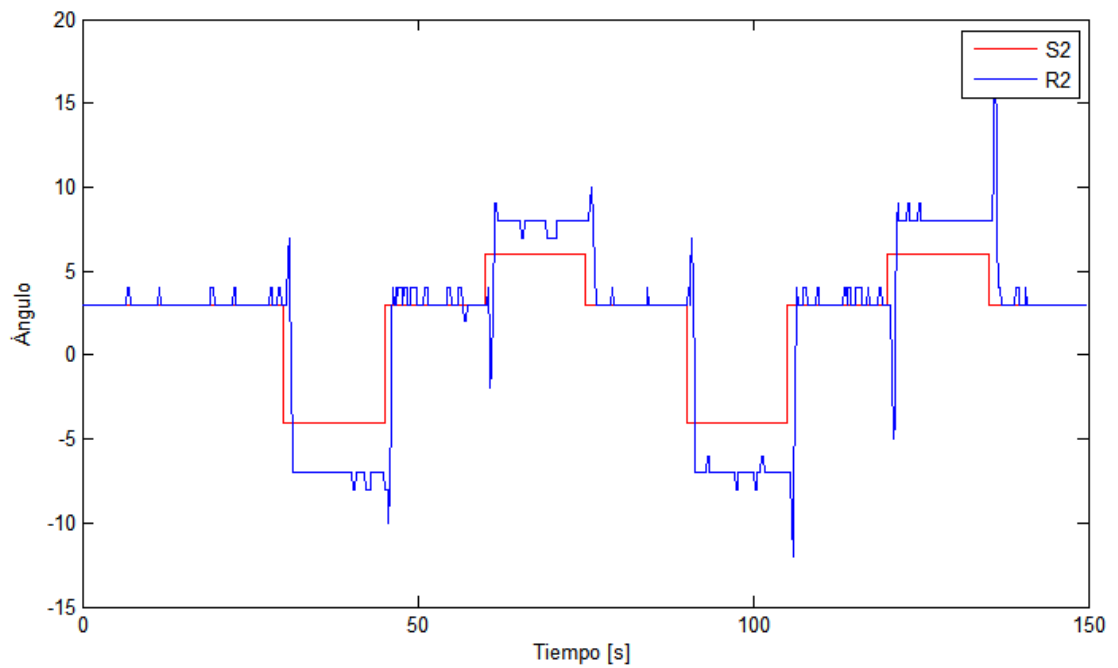
Como se observa en la gráfica de la Figura 87, el prototipo intenta llegar al valor del *set point* configurado en el ángulo Roll pero no lo logra, sin importar que los ángulos sean pequeños o grandes respecto al rango de movimiento del prototipo. El comportamiento anterior sugiere la utilización de un método de control como por ejemplo PI que maximice las señales y garantice un error mínimo en el prototipo.

A continuación se presentan las gráficas de los valores de *set point* de posición de cada servomotor, generados por el bloque cinemática inversa I y los valores de posición real de los servomotores en el prototipo establecidos en la prueba de control en lazo abierto:

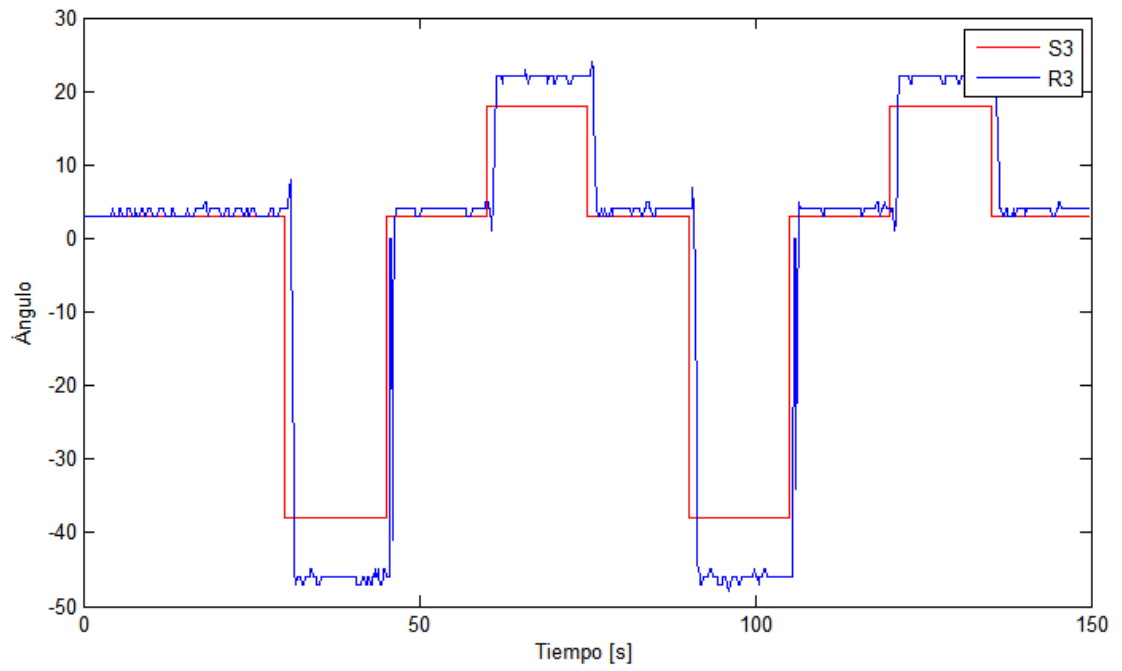
**Figura 88.** Posicionamiento del servomotor 1 en control con lazo abierto



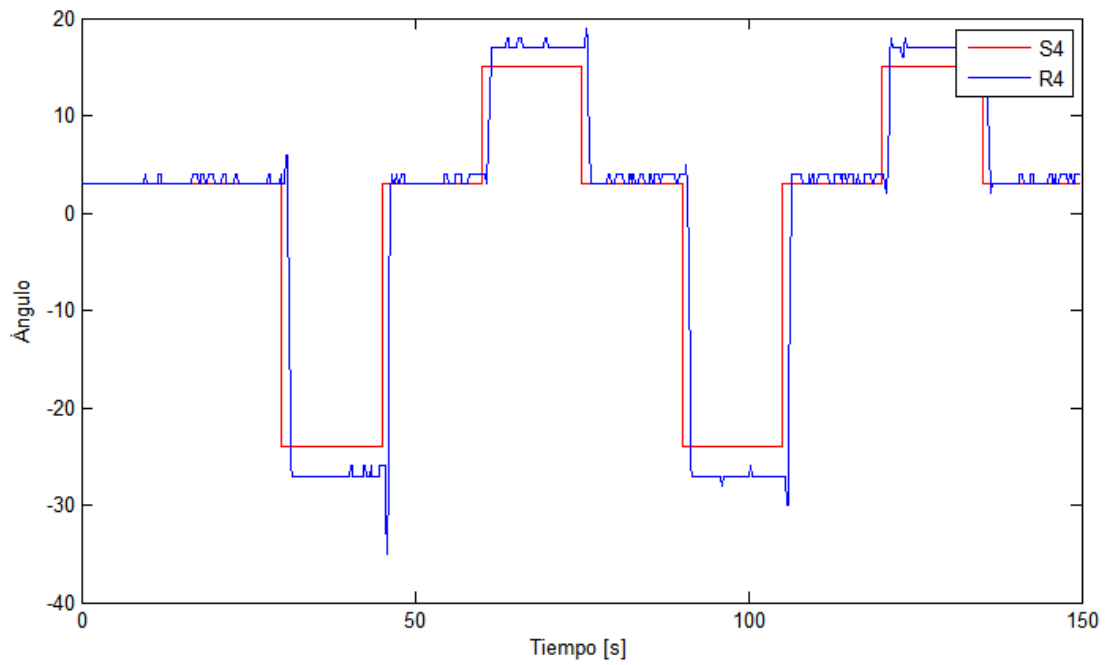
**Figura 89.** Posicionamiento del servomotor 2 en control con lazo abierto



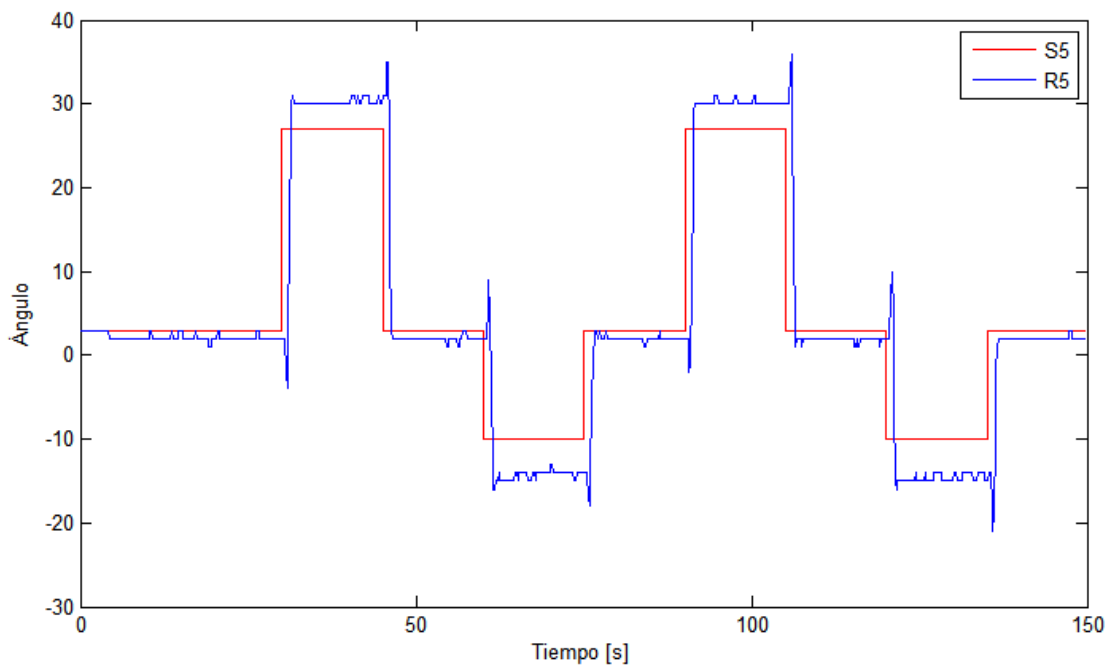
**Figura 90.** Posicionamiento del servomotor 3 en control con lazo abierto



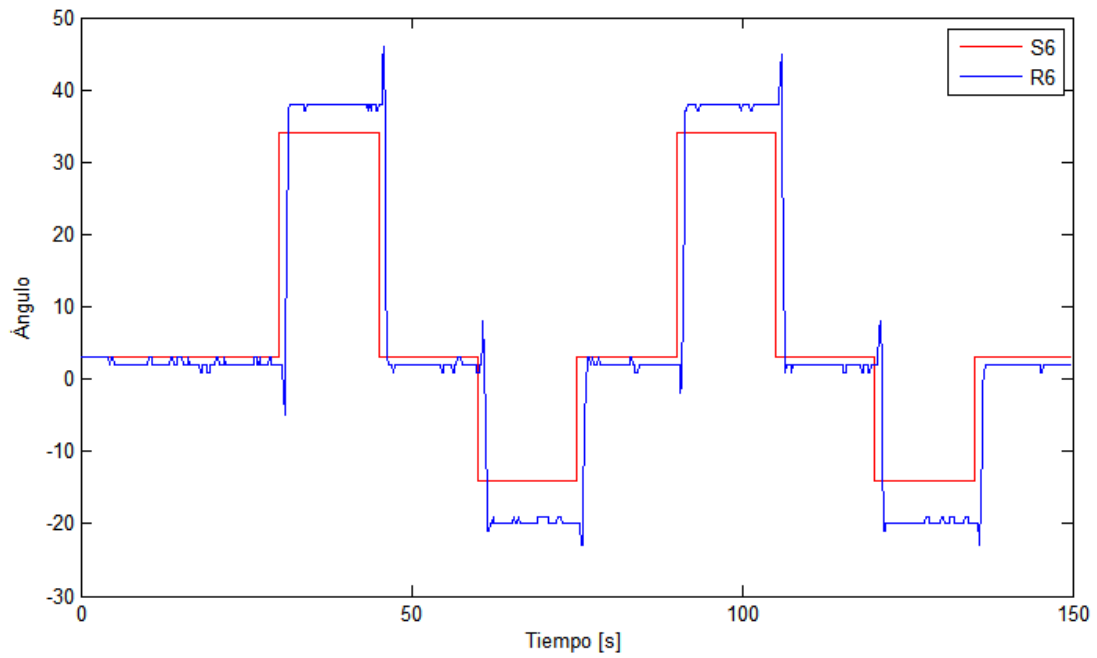
**Figura 91.** Posicionamiento del servomotor 4 en control con lazo abierto



**Figura 92.** Posicionamiento del servomotor 5 en control con lazo abierto



**Figura 93.** Posicionamiento del servomotor 6 en control con lazo abierto



En las seis gráficas anteriores las señales S (rojas) representan el valor de *set point* de posición de los servomotores y las señales R (azules) representan las señales de posición real que poseen los servomotores en el prototipo, se observa que la posición real de los servomotores en la prueba de control en lazo abierto no es la que dicta el bloque de cinemática inversa, por ello es necesario un controlador de posición en cada servomotor para así asegurar la reproducción de la posición angular del efector móvil especificada en la aplicación móvil.

## **6.5 CONTROL EN LAZO CERRADO**

El control del prototipo en lazo cerrado se realizó creando una señal de error por realimentación negativa entre las señales generadas en el bloque de cinemática inversa I, es decir, las señales de las posiciones que deben configurarse en cada servomotor para que el efector móvil tenga la posición angular deseada por la aplicación móvil y las señales generadas por el bloque cinemática inversa II que representan las señales de posición de cada servomotor establecidas realmente en el prototipo, tal como se explicó en... la sección 4.4.

Inicialmente se utilizaron las constantes proporcional e integral resultado de la metodología expuesta en... la sección 4.4.1.2... pero dichas constantes producían que el sistema oscilara drásticamente, por ello se sintonizaron los controladores PID del prototipo hasta encontrar las constantes que realmente controlaban la posición del efector móvil.

El proceso de sintonización del control PID del prototipo consistió en iniciar todos los controladores únicamente con la acción proporcional determinada en... la sección 4.4.1.2... y reducirla hasta que la planta no oscilara y tuviese un comportamiento controlado, luego activar la acción integral determinada en la misma sección y sintonizarla hasta una respuesta controlada. Observando las respuestas del controlador se determinó que se necesitaba acción derivativa con el

fin de reducir el tiempo de respuesta del controlador y evitar pequeñas oscilaciones. Las constantes que sintonizaron los controladores PID del prototipo son las siguientes:

$$P = 0.26$$

$$I = 0.75$$

$$D = 0.0000035$$

Dado que la planta del prototipo de plataforma Stewart está formada por 6 sistemas diferentes representados por cada servomotor, se sintonizó cada uno de los controladores de los servomotores de manera individual iniciando con las constantes anteriores. Resultado de la sintonización individual de cada servomotor se observó que los servomotores instalados en cada cara de la plataforma base deben poseer las mismas constantes de control, dado que si son ligeramente diferentes, los controladores de dicho par de servomotores oscilan mucho más que si las constantes son exactamente iguales.

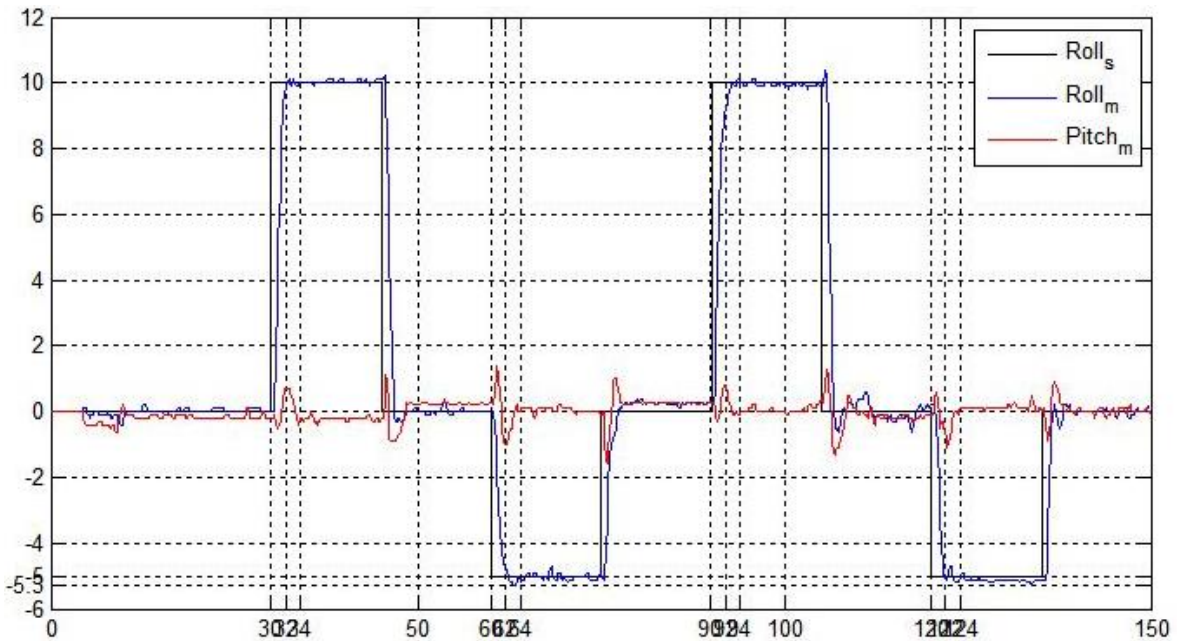
Para finalizar las constantes sintonizadas en la planta y utilizadas para el control final del prototipo de plataforma Stewart se muestran en la Tabla 13.

**Tabla 13.** Constantes del control PID de cada sistema

CONSTANTES	SERVO 1 y 2	SERVO 3 y 4	SERVO 5 y 6
PROPORCIONAL	0.23	0.26	0.24
INTEGRAL	0.75	0.75	0.75
DERIVATIVA	0.0000035	0.0000035	0.0000035

La Figura 94 es el resultado de control del ángulo Pitch constante y variaciones de Roll para una señal de *set point* cuadrada.

**Figura 94.** Resultado de control para variaciones en el ángulo Roll



Se determinaron las características sobre impulso, tiempo de respuesta y porcentaje de error máximas del control de posición angular del efector móvil del prototipo con las siguientes ecuaciones y se consignaron los resultados en la Tabla 14.

$$\text{Sobre Impulso} = \frac{\text{Magnitud pico más alto} - \text{Magnitud estado estable}}{\text{Magnitud estado estable}} \times 100$$

$t_r$  = tiempo en que la señal controlada alcanza la estabilidad

$$\%Error = \frac{\text{Magnitud Teórica} - \text{Magnitud Experimental}}{\text{Magnitud Teórica}} \times 100$$

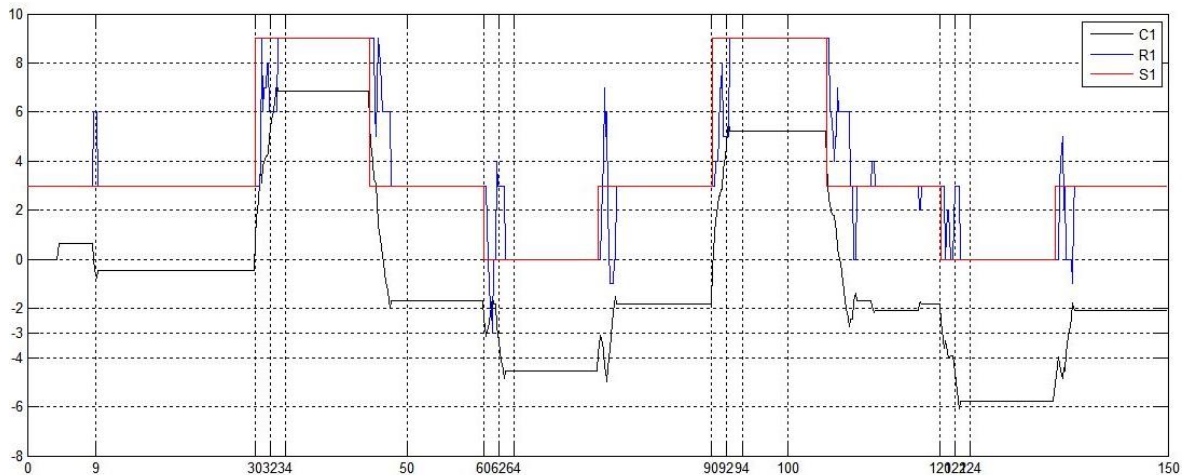
**Tabla 14.** Características del control de posición angular del efector móvil del prototipo.

<b>Sobre Impulso</b>	6%
<b>Tiempo de respuesta</b>	4 [s]
<b>% Error</b>	4% ( $\pm 0.2^\circ$ )

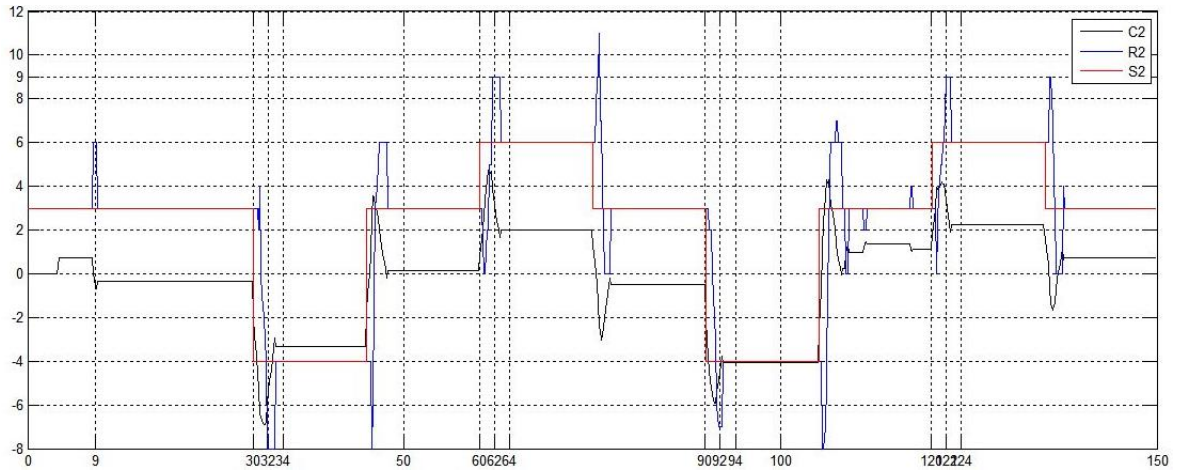
Se observa que las características de control de posición del efector móvil del prototipo diseñado en este proyecto son aceptables dado que el sobre impulso es inferior a un 10%, el tiempo de respuesta es el mínimo logrado por el prototipo y el porcentaje de error en el valor de estado estable es muy bajo e imperceptible al ojo humano a pesar que no es inexistente.

A continuación se presentan las gráficas de control de posición de los servomotores instalados en el prototipo:

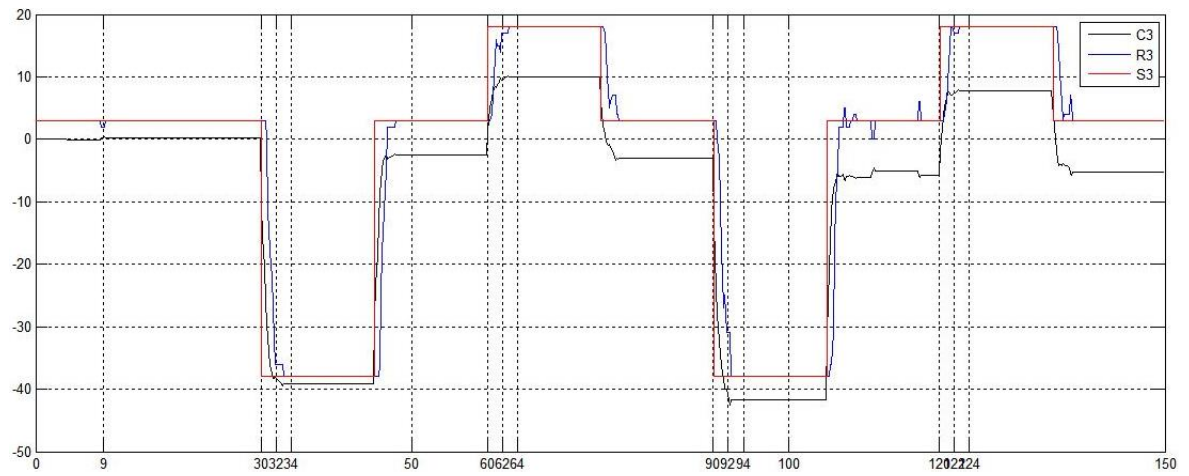
**Figura 95.** Control de posición en lazo cerrado del servomotor 1



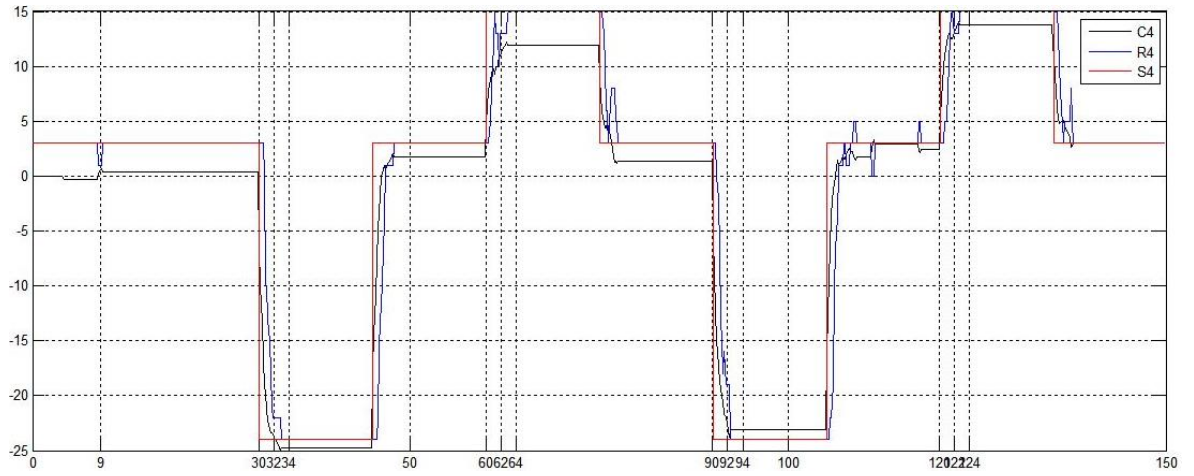
**Figura 96.** Control de posición en lazo cerrado del servomotor 2



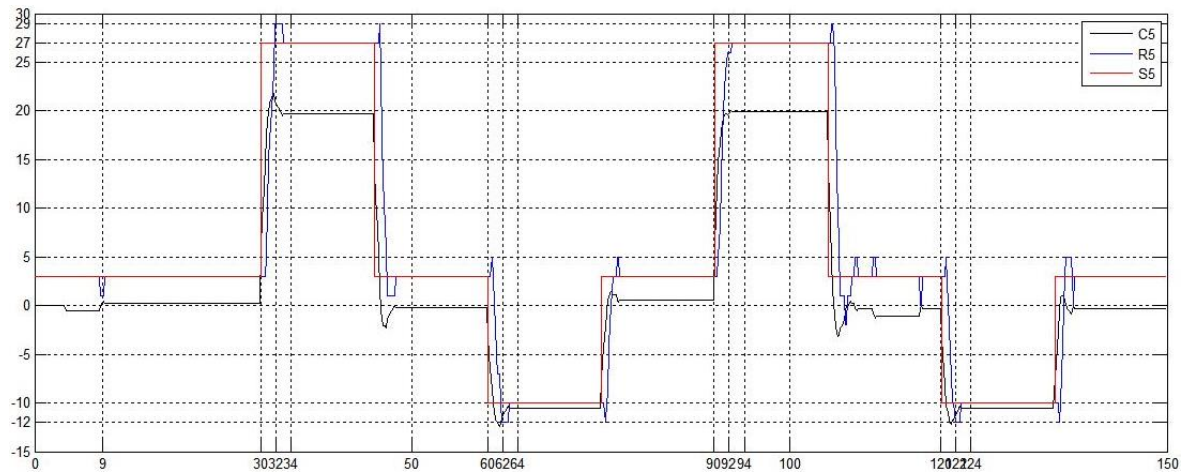
**Figura 97.** Control de posición en lazo cerrado del servomotor 3



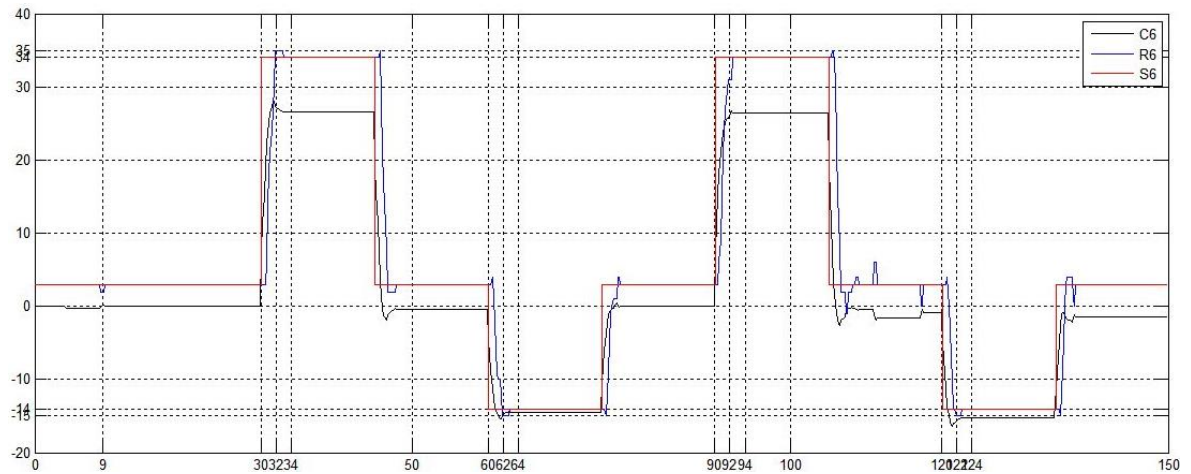
**Figura 98.** Control de posición en lazo cerrado del servomotor 4



**Figura 99.** Control de posición en lazo cerrado del servomotor 5



**Figura 100.** Control de posición en lazo cerrado del servomotor 6



En la Tabla 15 se presentan las características máximas presentadas en los controladores PID aplicados a cada una de las señales de posición de los servomotores. La creación de la tabla tuvo únicamente en cuenta los resultados de control presentados para *set points* diferentes de cero.

**Tabla 15.** Características de los controladores de cada servomotor

	<b>Servo 1</b>	<b>Servo 2</b>	<b>Servo 3</b>	<b>Servo 4</b>	<b>Servo 5</b>	<b>Servo 6</b>
<b>Sobre Impulso</b>	No Existe	50%	No Existe	No Existe	20%	7.2%
<b>Tiempo de respuesta</b>	3 [s]	3 [s]	3 [s]	3 [s]	3 [s]	3 [s]
<b>% Error Estacionario</b>	No Existe	No Existe	No Existe	No Existe	No Existe	No Existe

## 7. CONCLUSIONES

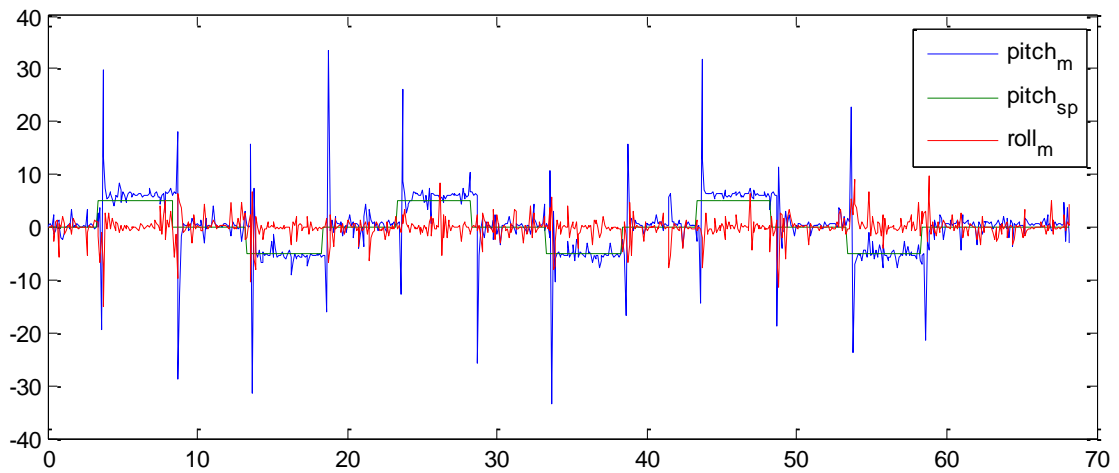
- La instalación de servomotores como actuadores del prototipo permiten una actuación rápida del mismo, sin embargo, el uso de este tipo de actuadores hace mucho más complejo el análisis de cinemática inversa en relación a actuadores lineales, debido a que se deben crear relaciones geométricas y trigonométricas que definan el cambio de posición de la junta instalada en el efector móvil utilizando el ángulo de posición del servomotor, de manera que encontrar dicho ángulo de solución requiere de un proceso iterativo que genera un gasto computacional y un pequeño retraso en el programa de control del prototipo.
- Aunque con el diseño hexápodo del prototipo de plataforma Stewart los ángulos de rotación en los ejes X y Y no superan los  $23^\circ$ , este rango es suficiente para simular el vuelo de aeronaves en condiciones normales tal como se compara en la Tabla 16.

**Tabla 16.** Ángulos ejecutados en vuelo y en el prototipo diseñado.

<i>Ángulo</i>	<i>∠ AERONAVE</i>	<i>∠ PROTOTIPO</i>
<i>Pitch</i>	$-3,2^\circ$ a $20^\circ$	$22,3^\circ$
<i>Roll</i>	$\pm 15^\circ$ a $\pm 20^\circ$	$19^\circ$

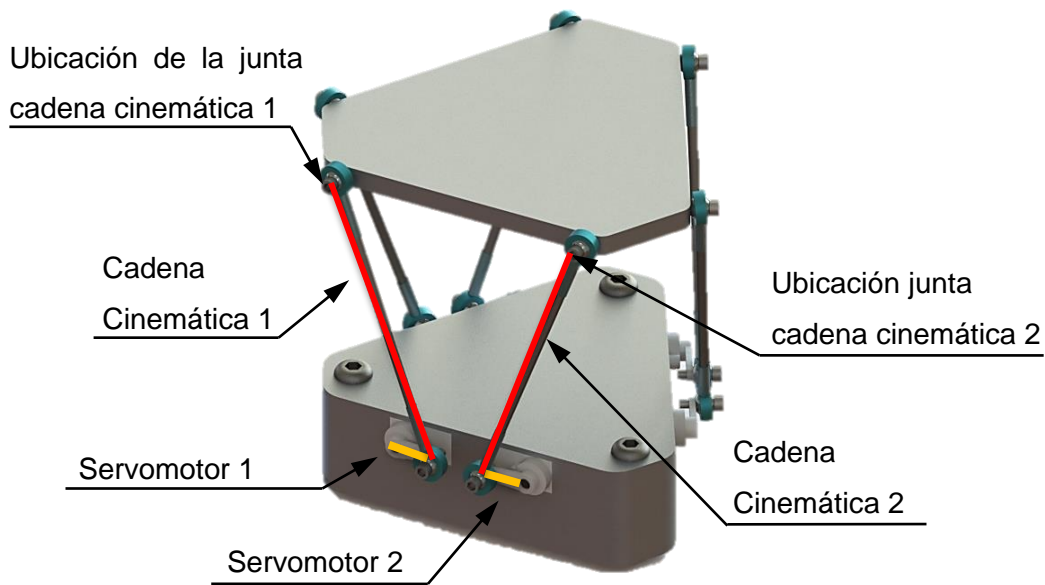
- Los picos y ruido (ver Figura 101) presentados en... la sección 4.4.1.1... en las figuras de lectura de ángulos del efector móvil y la posición de los servomotores no son producto de un movimiento brusco del prototipo sino un ruido generado en las lecturas del sensor de posición angular tal como se demostró en... la sección 6.3.

**Figura 101.** Picos de lectura en ángulos Pitch y Roll del sensor acelerómetro



➤ Ejecutar un control PID a la posición de cada uno de los servomotores, identificando cada cadena cinemática del robot como un sistema diferente es una decisión acertada, sin embargo, se resalta que cada una de las cadenas cinemáticas del robot poseen los mismos elementos que las demás y actúan sobre las mismas características (la posición del efector móvil), por ello sus controladores deben tener un nivel de congruencia en la sintonización, diferenciados únicamente por las variaciones del efecto que generan cada una de las acciones de las cadenas cinemáticas sobre la posición angular del efector móvil debidas a la ubicación de las juntas sobre el mismo.

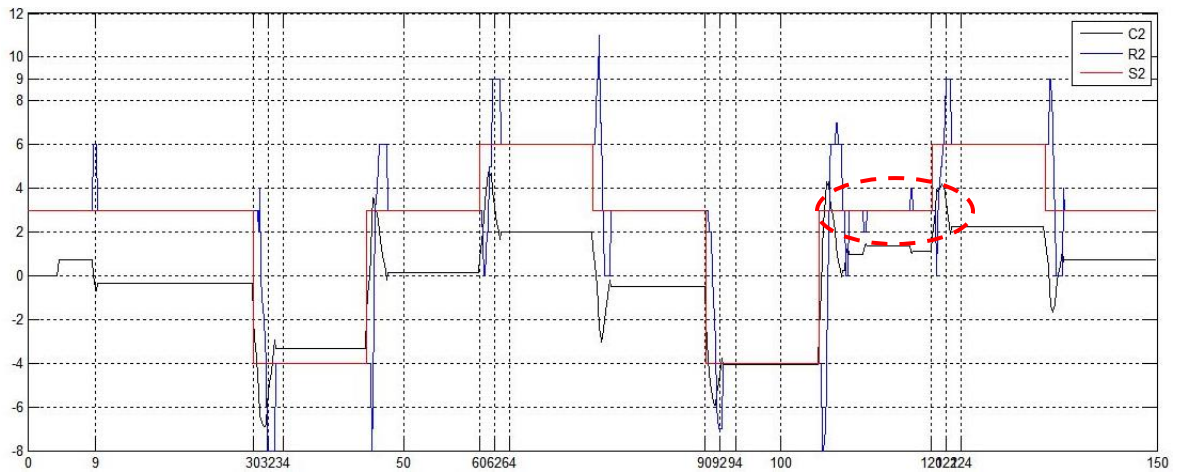
**Figura 102.** Cadenas cinemáticas 1 y 2 del prototipo diseñado



Por otro lado, los controladores de los servomotores cercanos entre sí, es decir, por ejemplo los controladores de los servomotores uno y dos instalados sobre la misma cara de la plataforma base deben tener controladores idénticos, dado que de otra manera se presentan grandes oscilaciones en las respuestas al control de los mismos para pequeñas variaciones de las constantes de control PID.

- La existencia de error en estado estacionario en el control de posición angular del efector móvil del prototipo se presenta debido a pequeñas vibraciones de la plataforma producto de pequeños saltos de posición de los servomotores como los que se presentan en la Figura 103.

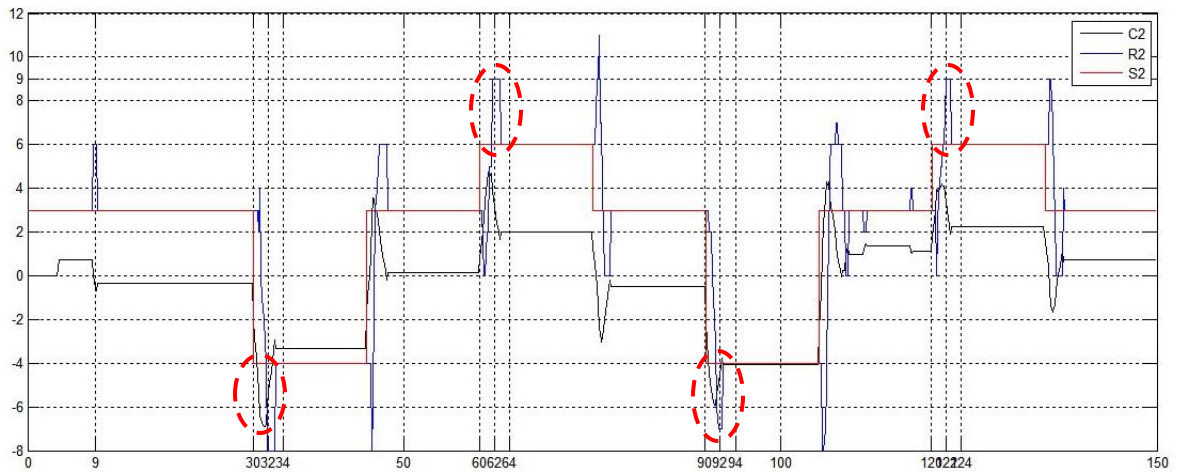
**Figura 103.** Gráfica ejemplo de saltos en la posición de los servomotores.



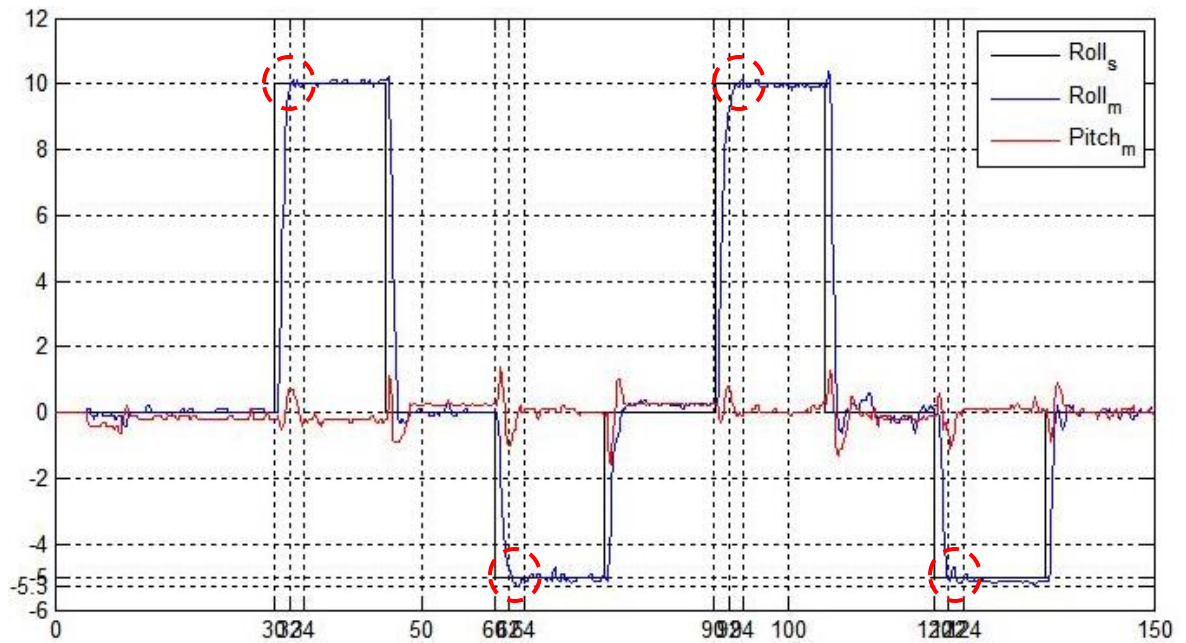
Los saltos de posición en un servomotor afectan la posición angular del efector móvil del prototipo generando una reacción en cadena que ocasiona la oscilación de cada uno de los servomotores, dado que los sistemas de control (cadenas cinemáticas) reciben efectos mecánicos de los demás sistemas.

- Los sobre impulsos que se presentan en los controles individuales de posición de los servomotores (ver Figura 104) no afectan en gran manera el resultado general del control de posición del efector móvil del prototipo (ver Figura 105).

**Figura 104.** Sobre impulso en la señal de control del servomotor 2.



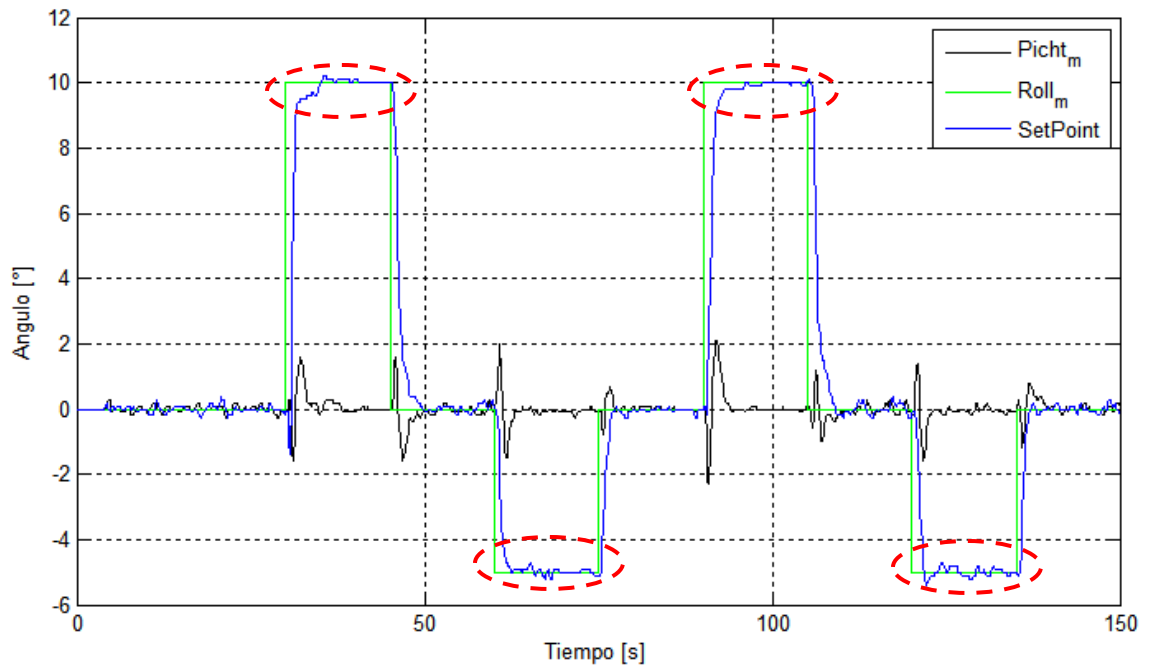
**Figura 105.** Control de posición angular del efector del prototipo



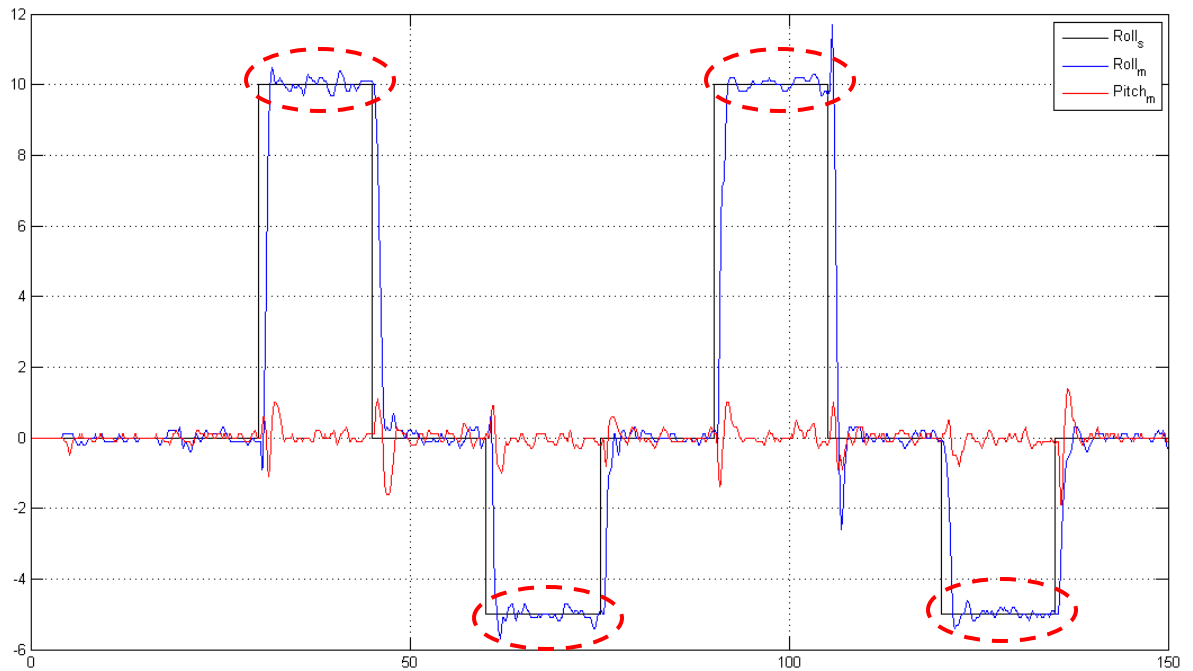
➤ Las respuestas del controlador se ven afectadas notablemente por cambios bruscos de temperatura, principalmente en el sobre impulso de la señal controlada y en el error en estado estacionario. La temperatura influncia las respuestas

eléctricas del *hardware* electrónico y también características dimensionales como la longitud de las barras de los brazos de la plataforma. Este cambio notable en la respuesta del controlador se aprecia en las Figuras Figura 106 y Figura 107 tomadas con un delta de temperatura aproximado de 10°C.

**Figura 106.** Respuesta del controlador PID inicial

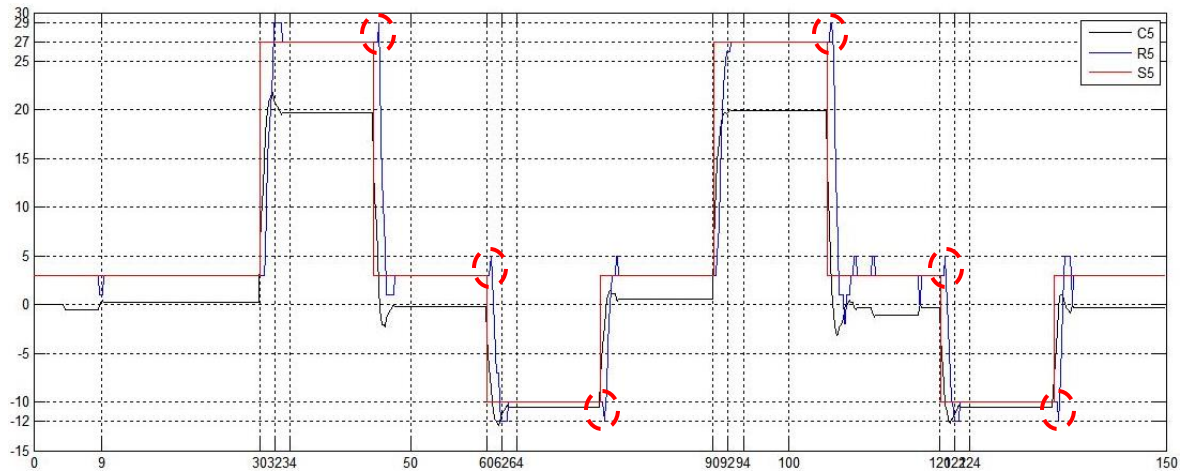


**Figura 107.** Respuesta del controlador PID con cambio de temperatura



- Los picos presentados (ver Figura 108) cuando el prototipo inicia el movimiento, son producto del efecto cinemático del cambio de posición del efector sobre las lecturas de los sensores, que al sentir un movimiento repentino generan lecturas altas que afectan por un lapso de tiempo corto (aproximadamente 1 segundo) el cálculo del ángulo configurado en el efector del prototipo.

**Figura 108.** Picos presentes al inicio de cambio de posición del efector móvil.



➤ El sensor MPU 6050 hace posible determinar los ángulos Pitch y Roll con un nivel de precisión bastante alto dado que el acelerómetro usa la gravedad terrestre para definir la posición angular con respecto a la superficie de la tierra y las mediciones de cambio de giro del giroscopio eliminan el ruido generado por vibraciones en el acelerómetro mediante la fusión de las lecturas de los dos sensores, sin embargo, no es posible obtener una señal del ángulo Yaw estable en el tiempo ya que este ángulo debe determinarse únicamente con las mediciones del giroscopio, las cuales sufren de *drift*<sup>23</sup> después de un intervalo de tiempo de medio minuto aproximadamente. Entonces el sensor IMU MPU 6050 es insuficiente para definir la posición angular del prototipo mediante los ángulos de Euler.

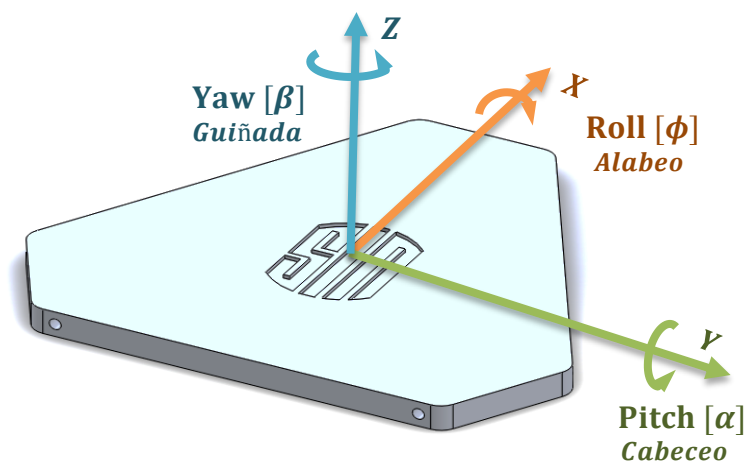
---

<sup>23</sup> El *drift* o deriva es una característica negativa de ciertos sensores que consiste en el cambio de la señal del sensor ya sea en aumento o decremento aún sin que el sensor o la característica medida tengan ningún cambio.

## 8. RECOMENDACIONES

- El gasto de recurso computacional o de procesamiento es evitable en el desarrollo de futuras plataforma Stewart dedicando hardware específico como módulo de resolución (por ejemplo Raspberry PI<sup>24</sup> o un PLC<sup>25</sup>) encargado de resolver únicamente la cinemática inversa de la plataforma.
- El prototipo de plataforma Stewart de configuración hexápoda diseñado en este proyecto es simétrico con respecto al eje X pero no con respecto al eje Y (ver Figura 109), por ende los límites del ángulo de inclinación Pitch son diferentes para inclinaciones positivas y negativas. Con la orientación del ángulo Pitch establecida en el prototipo diseñado el rango de inclinación positivo de dicho ángulo es inferior al rango negativo.

**Figura 109.** Ejes coordenados del prototipo de plataforma Stewart



<sup>24</sup> Raspberry PI es un controlador de uso libre, definido como un computador de placa única de bajo costo fabricado y desarrollado por Fundación Raspberry PI.

<sup>25</sup> PLC conocido así por sus siglas en inglés *Programmable Logic Controller*, es un dispositivo de control programable generalmente utilizado en ambientes industriales automatizados.

Por razones de seguridad en vuelo una aeronave aterriza con ángulo de inclinación inferior a  $3,2^\circ$ . Por ende se recomienda cambiar en futuras plataformas la orientación del eje Y de la misma con el propósito de establecer el rango mínimo de inclinación de dicho eje coincidente con el ángulo de aterrizaje de una aeronave.

➤ La interfaz del *software* Simulink permite el tratamiento de flujo de señales de forma sencilla y el uso de una amplia gama de librerías programadas en un lenguaje gráfico e intuitivo, sin embargo, algunos bloques de programación deben configurarse con un valor llamado *sample time* que representa el tiempo de “muestreo” o el periodo en que dicho bloque realiza su acción, por ende al crear un programa voluminoso y complejo se crea un fraccionamiento del tiempo de simulación que debe sincronizarse para que todos los bloques con esta característica de configuración trabajen de manera adecuada con los dispositivos de *hardware* externos. Siendo así, con el fin de evitar inconvenientes en la sincronización del *software* Simulink y el *hardware* externo se recomienda el uso de otro *software* de control como por ejemplo LabVIEW.

➤ Dado que el control de los servomotores, pero en especial el control de los servomotores 1 y 2 presentan oscilaciones en estado estable y esta no fue corregida en totalidad con controlador PID individual de cada servomotor, se recomienda utilizar un control multi-variables que garantice el control de posición de un sistema teniendo en cuenta los efectos que ejercen sobre él las acciones de los demás sistemas.

➤ Determinar los ángulos de Euler de una plataforma Stewart, crea la necesidad de fusionar las lecturas de un sensor IMU (aceleración y velocidad angular) con una brújula o magnetómetro, dado que es posible calcular los ángulos Pitch y Roll con las lecturas del IMU y fusionar con el uso del filtro complementario las señales del ángulo de Yaw del giroscopio y el *heading* del magnetómetro que brinda una referencia del polo magnético de la tierra y corrige el *drift* del giroscopio.

## BIBLIOGRAFÍA

ALLERTON D, Principles of Flight Simulation. 1st ED, Editorial Wiley & Sons, Ltd, United Kingdom, Chichester 2009.

ARIAS GUZMÁN, Roberto Patricio. Diseño, construcción y control de una plataforma Stewart con 6 grados de libertad que funcione como un simulador de vuelo. Escuela Politécnica Nacional: Quito, 2013.

BOLTON, W. Mecatrónica: Sistemas de control electrónico en ingeniería mecánica y eléctrica. 2 ed: Alfaomega.

CRAIG, Jhon J. Robótica. Traducido por el Ing. Alfonso Vidal. Pearson Educación, México, 2006. 3 ed.

DONELAN, Peter. Kinematic Singularities of Robot Manipulators. Victoria University of Wellington: New Zealand.

DONELAN, Peter. Singularities of Robot Manipulators. Victoria University of Wellington: New Zealand.

GÓMEZ P, Jhon Andrés y GUACANEME G, José Vicente. Diseño e Implementación de una Plataforma de Stewart.

GONZÁLEZ ACUÑA, Hernán y SUELL DUTRA, Max. Cinemática inversa y análisis del espacio de trabajo de una plataforma Stewart. 2009.

GONZÁLEZ SUÁREZ, Gerardo y VARGAS VARGAS, Andrea Liliana. Modelado matemático y simulación de una Plataforma de 6 grados de libertad (Plataforma Stewart – Gough). Universidad Industrial de Santander: 2013.

MALPARTIDA V, Sergio Ernesto. Diseño mecánico de una cabina para un simulador de entrenamiento de vuelo. Pontificia Universidad Católica del Perú: 2014.

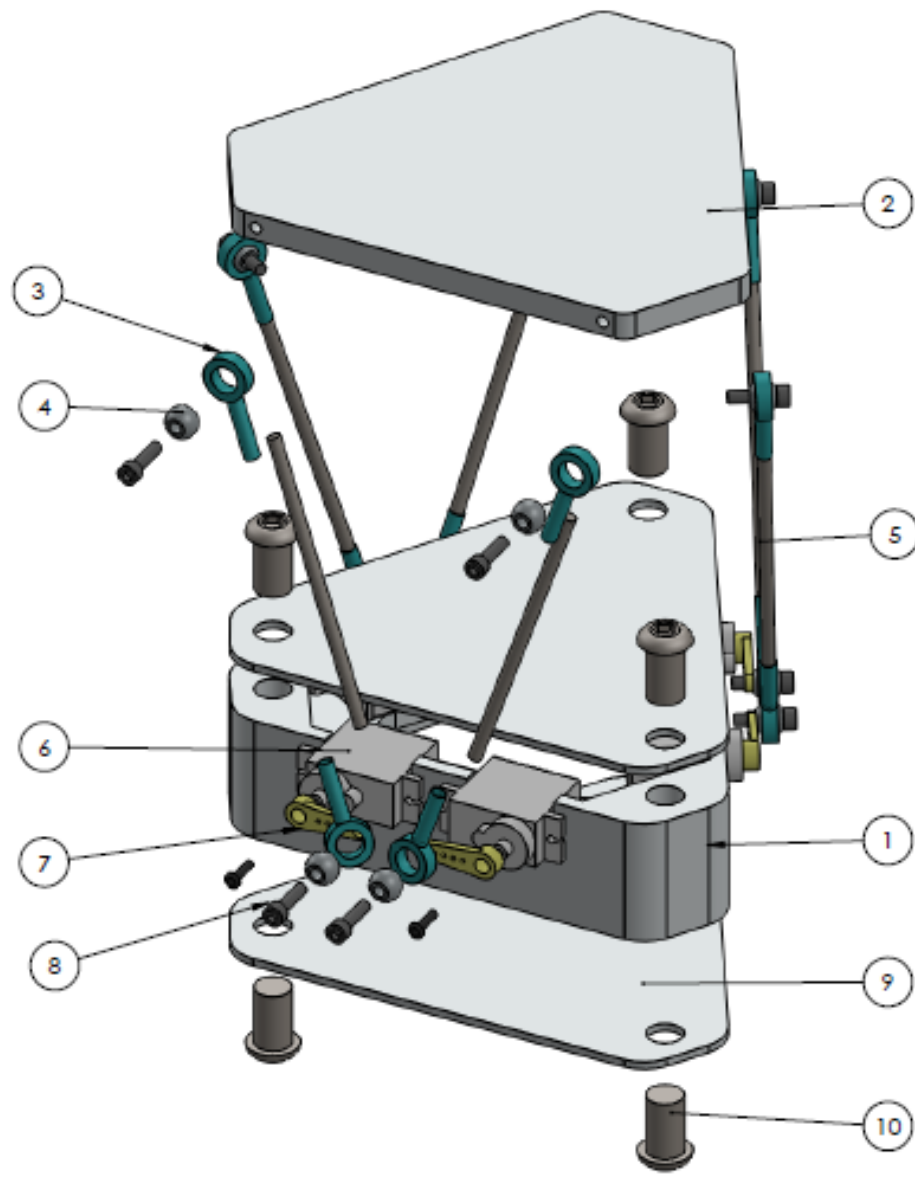
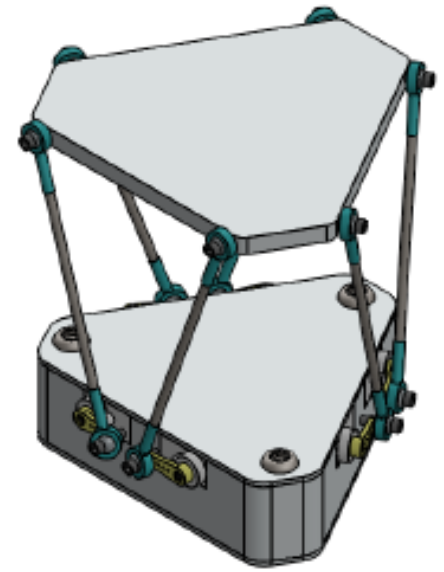
MATHWORKS. Legacy MATLAB and Simulink Support for Arduino. [sitio web]. Disponible en: <https://de.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino>

MORENO AVALOS, Héctor A. Robots Paralelos: Conceptos básicos y aplicaciones. Centro de Automática y Robótica CSIC.

RAMÍREZ R, D.A y MÉNDEZ M, L.M. Modelamiento dinámico de mecanismos paralelos con actuadores prismáticos utilizando SimMechanics de Matlab. Intekhnia: 2011.

## **ANEXO A. PLANOS DE DISEÑO MODELO CAD PROTOTIPO PLATAFORMA STEWART**

A continuación se adjuntan los planos de diseño de del modelo CAD desarrollado, el cual incluye un plano explosionado del prototipo de plataforma Stewart y los planos de los elementos principales, tales como la plataforma base, la plataforma móvil y las rotulas utilizadas en los brazos actuadores del prototipo.



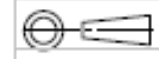
Nº	NOMBRE	CANT	MATERIAL
1	PLATAFORMA FIJA (BASE)	1	ALUMINIO FUNDIDO
2	PLATAFORMA MOVIL (EFECTOR)	1	ALUMINIO FUNDIDO
3	RÓTULA	12	ACERO C45 TEMPLADO
4	ELEMENTO RODANTE	12	ACERO INOXIDABLE
5	VÁSTAGO	6	ACERO INOXIDABLE AL Cr
6	SERVOMOTOR	6	VARIOS
7	PALA	6	PLÁSTICO POM
8	TORNILLO PALA	12	ACERO CARBURIZADO
9	TAPA	1	LÁMINA ACERO
10	TORNILLO TAPA	6	ACERO CARBURIZADO



**ESCUELA DE INGENIERIA MECANICA**

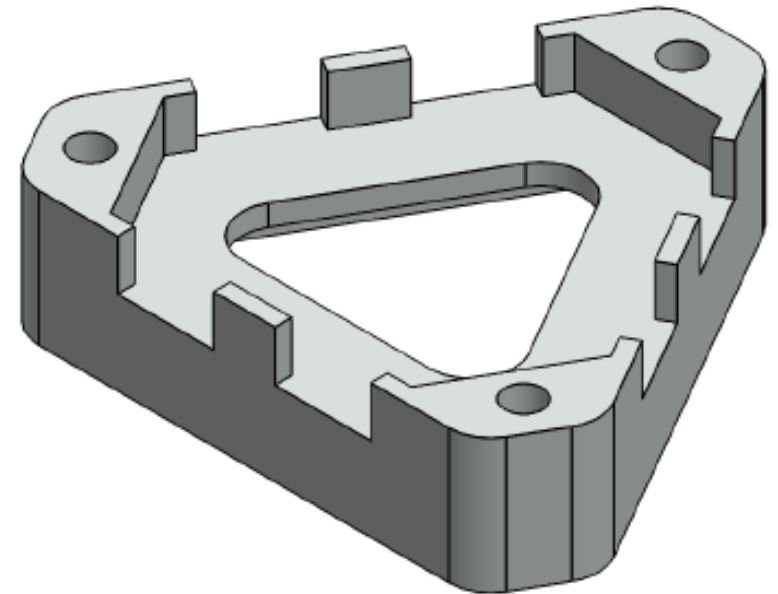
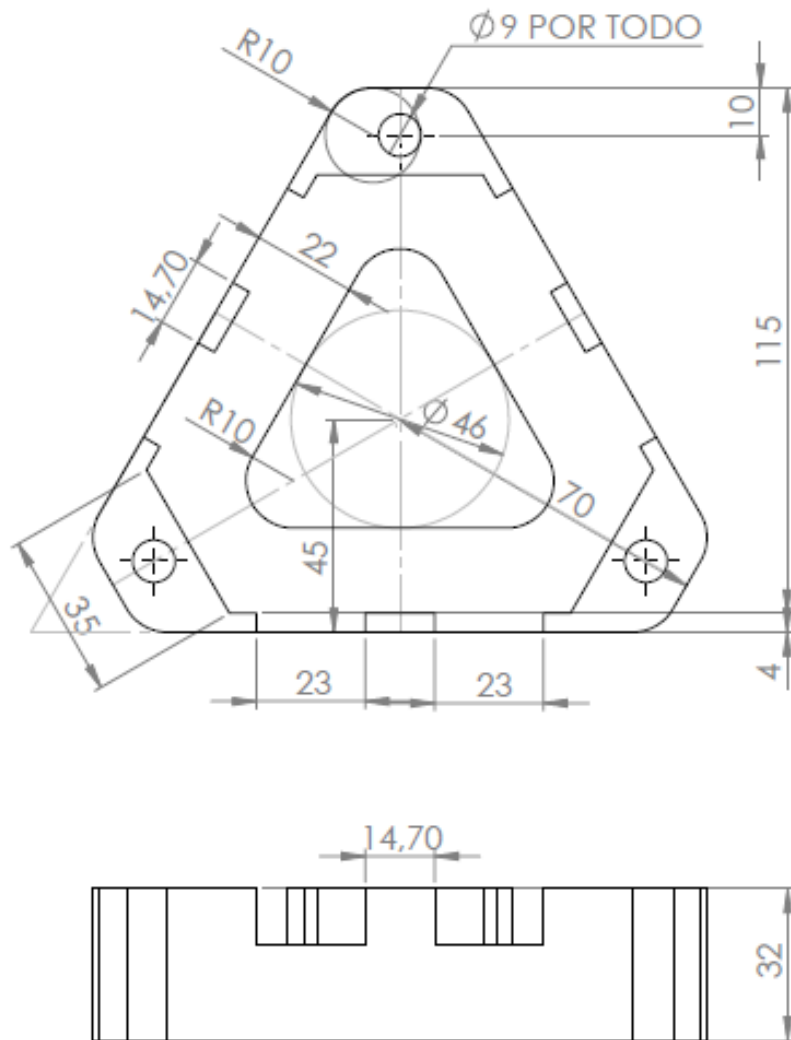
ESCALA: 1:3

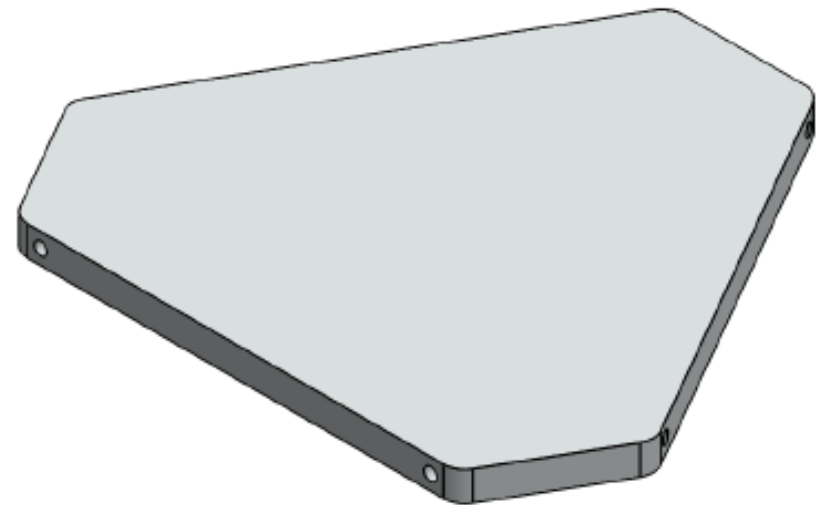
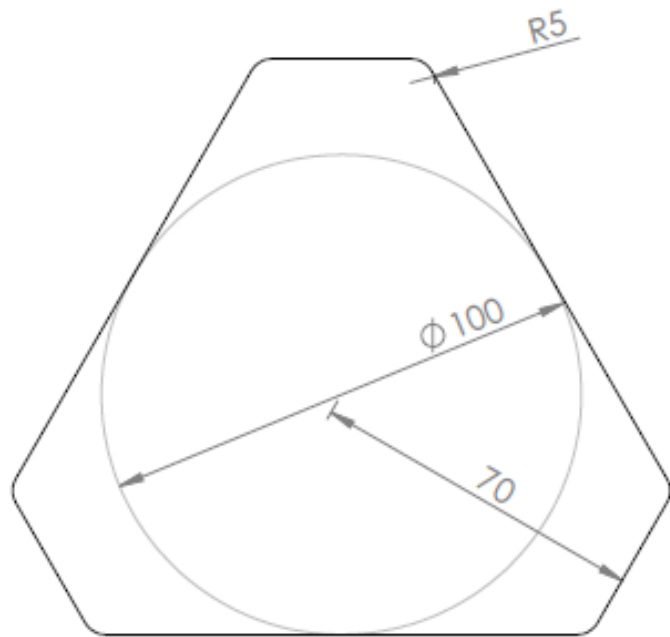
**PLATAFORMA STEWART**

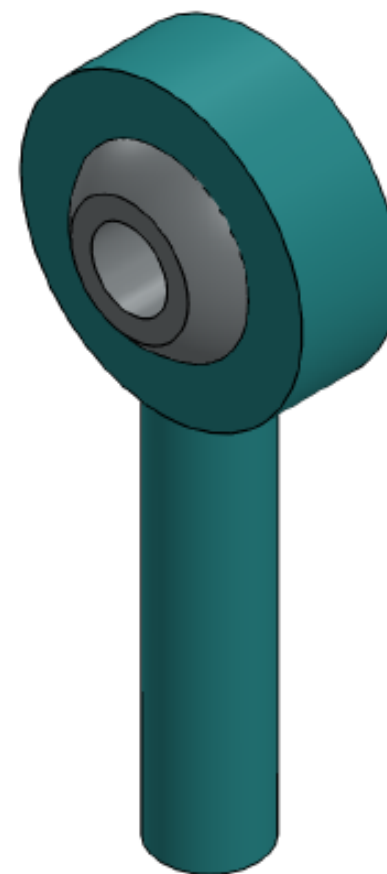
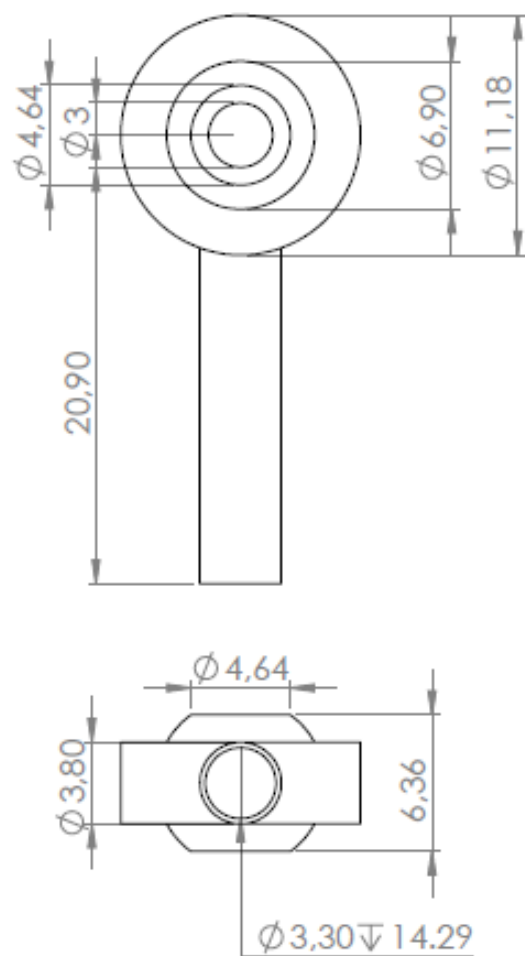


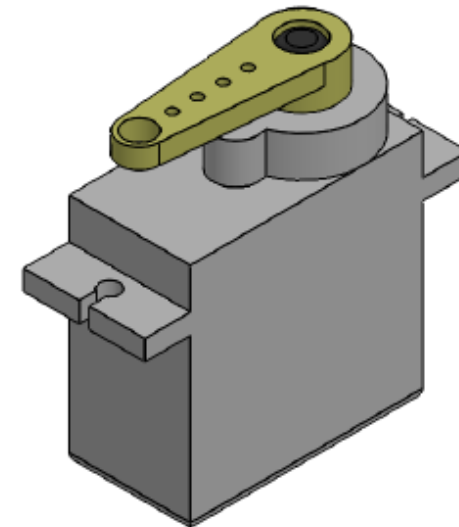
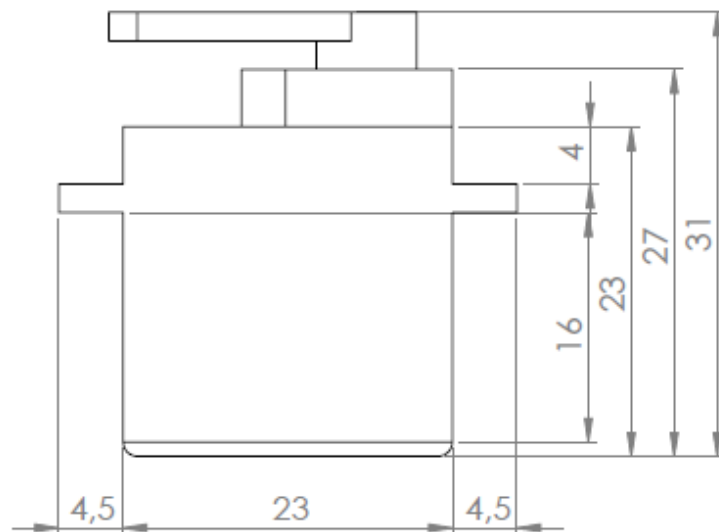
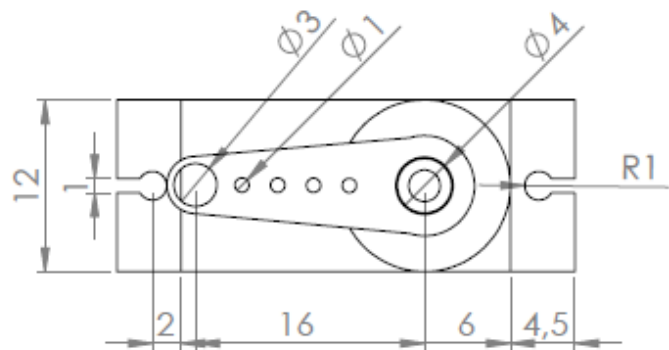
FORMATO A3      2017-01-19      HOJA 1/5

MEDIDAS EN: mm      DIBUJADO POR: JHON NUÑEZ - CRISTIAN ACOSTA      PL N° 01-03-00









## ANEXO B. DESARROLLO DE LA CINEMÁTICA INVERSA

Este anexo se ha incluido con el fin de brindar una explicación detallada de la solución a la cinemática inversa de la plataforma Stewart.

Para comenzar se define la orientación y posición de la plataforma mediante el vector:

$$P = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

Donde,  $X, Y$  y  $Z$  definen las coordenadas del nuevo punto de la plataforma y  $\alpha, \beta$  y  $\gamma$  definen los ángulos de rotación. En otras palabras, la plataforma gira con respecto al eje  $X$ ,  $\alpha$  grados, sobre el eje  $Y$ ,  $\beta$  grados y sobre el eje  $Z$ ,  $\gamma$  grados.

Un objeto ubicado en el espacio puede cambiar su posición y/o tamaño respecto a su sistema de referencia mediante transformaciones lineales, es decir, si un objeto cambia su escala, se traslada, rota o se refleja es gracias a la transformación lineal que sufre.

Un punto expresado mediante un vector  $(x, y)$  puede ser trasladado, rotado, etc. Si a este vector se le aplica una matriz de transformación, supóngase que se desea trasladar el punto  $(x, y)$  hasta el punto  $(x', y')$ , solo se debe sumar el vector  $(x, y)$  más un vector de traslación  $T$ .

$$(x', y') = (x, y) + \mathbf{T}$$

Si se desea trasladar no solo un punto sino un objeto completo, se debe someter a dicho valor de traslación a todos los puntos significativos de dicho objeto. Por otro lado si se desea rotar o cambiar de escala a un objeto, se debe multiplicar el vector  $(x, y)$  por las matrices de transformación.

$$(x', y') = (x, y) * \mathbf{R}$$

$$(x', y') = (x, y) * \mathbf{S}$$

Por tanto, para trasladar un objeto, se debe sumar el vector de traslación y para rotarlo o cambiar su escala es necesario multiplicarlo por las matrices de transformación. Para generalizar el proceso se debe homogeneizar el sistema de referencia (sistema de coordenadas homogéneo), esto se logra agregando una dimensión más al sistema de referencia dado, con lo que se obtiene la siguiente ecuación.

$$(x', y', \omega') = (x, y, \omega) + \mathbf{T}$$

Con el objetivo de facilitar el cálculo  $\omega'$  y  $\omega$  se hacen iguales a 1. Por otro lado si se quiere trasladar un objeto y luego rotarlo, las operaciones necesarias son las siguientes:

$$(x', y', z', 1) = (x, y, z, 1) + \mathbf{T}$$

$$(x'', y'', z'', 1) = (x', y', z', 1) * \mathbf{R}$$

Como se observa se necesitan 2 operaciones por cada punto. Pero existe una manera simplificada de realizar esta operación, y esta se obtiene multiplicando las matrices  $\mathbf{T} * \mathbf{R}$  (necesariamente en este orden, ya que la propiedad conmutativa no se cumple en estos casos) para obtener una matriz  $\mathbf{X}$ . Por lo tanto, se puede obtener el mismo resultado usando la siguiente ecuación:

$$(x'', y'', z'', 1) = (x', y', z', 1) * \mathbf{X}$$

Este tipo de matriz es llamada matriz neta. La forma general de una matriz neta es la indicada:

$$\mathbf{X} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ d_x & d_y & d_z & 1 \end{bmatrix}$$

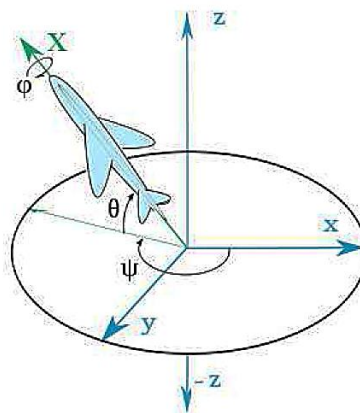
Donde  $a_{ij}$  representa el cambio de escala y/o rotación, mientras  $d_i$  representa el cambio de traslación.

Cuando se analiza una matriz de traslación se puede observar que es bastante simple, únicamente se ubican los valores de  $d_x$ ,  $d_y$ , y  $d_z$ . Mientras que para la matriz rotacional no resulta tan simple. En primer lugar se debe elegir en qué sentido se quiere girar y para hacer esto se debe elegir uno de los ejes de referencia para hacer girar el objeto en ese sentido.

### A.1 ÁNGULOS DE EULER Y DE TAIT-BRYAN

Los ángulos de Tait-Bryan, nombrados así en honor al matemático escocés Peter Guthrie Tait, son tres ángulos que son usados para describir una rotación en el espacio euclidiano tridimensional mediante 3 rotaciones distintas entre sí. En la aeronáutica se utiliza el nombre de ángulos de Euler para hablar de los ángulos, que en geometría son conocidos como los ángulos de Tait-Bryan. Como habitualmente en la aviación se les conoce como ángulos de Euler, de aquí en adelante se considerara estos ángulos también como ángulos de Euler.

**Figura 71.** Ángulos de Euler.



Fuente: [https://es.wikipedia.org/wiki/%C3%81ngulos\\_de\\_navegaci%C3%B3n](https://es.wikipedia.org/wiki/%C3%81ngulos_de_navegaci%C3%B3n)

Como se muestra en la Figura 71, el ángulo  $\theta$  corresponde al ángulo de pitch, el ángulo  $\varphi$  corresponde al ángulo de roll y el ángulo  $\psi$  corresponde al ángulo de yaw.

Es posible definir las matrices del cuerpo rígido rotando (eliminando la coordenada extra de homogeneidad solo para ilustrar el funcionamiento de esta matriz):

$$R(x, \varphi) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\text{sen}\varphi \\ 0 & \text{sen}\varphi & \cos\varphi \end{vmatrix}$$

$$R(y, \theta) = \begin{vmatrix} \cos\theta & 0 & \text{sen}\theta \\ 0 & 1 & 0 \\ -\text{sen}\theta & 0 & \cos\theta \end{vmatrix}$$

$$R(z, \psi) = \begin{vmatrix} \cos\psi & -\text{sen}\psi & 0 \\ \text{sen}\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Multiplicando estas tres matrices, se obtiene una matriz completa, llamada matriz coseno director, que está definida por la siguiente expresión:

$$R = R(z, \psi) * R(y, \theta) * R(x, \varphi)$$

$$R = \begin{vmatrix} \cos\psi\cos\theta & \cos\psi\text{sen}\theta\text{sen}\varphi - \text{sen}\psi\cos\varphi & \cos\psi\text{sen}\theta\cos\varphi + \text{sen}\psi\text{sen}\varphi \\ \text{sen}\psi\cos\theta & \text{sen}\psi\text{sen}\theta\text{sen}\varphi + \cos\psi\cos\varphi & \text{sen}\psi\text{sen}\theta\cos\varphi - \cos\psi\text{sen}\varphi \\ -\text{sen}\theta & \cos\theta\text{sen}\varphi & \cos\theta\cos\varphi \end{vmatrix}$$

## A.2 RESOLUCIÓN DEL PROBLEMA DE CINEMÁTICA INVERSA

Para resolver la cinemática inversa de la plataforma Stewart construida, es necesario conocer la geometría de la plataforma en su totalidad, es por eso que en el numeral 2.2 se estableció las medidas de cada actuador y de las plataformas base y móvil. Una vez determinadas las características geométricas se puede iniciar la resolución del problema de cinemática inversa de un robot.

Para calcular la cinemática inversa de este tipo de robot, debe obtenerse la longitud del actuador que es función del ángulo del servomotor, este valor debe estar en

función de la orientación que tenga el efector en el sistema coordenado X, Y y Z y los ángulos de Euler, pitch, yaw y roll. Esto se obtiene fácilmente usando la matriz de rotación vista anteriormente. Aunque la matemática puede parecer complicada, si se realiza un análisis geométrico completo de la estructura, el análisis cinemático inverso resulta ser bastante simple.

## ANEXO C. CÓDIGO TARJETA ARDUINO UNO

El siguiente código es el resultado de la modificación del código adioesrv.pde diseñado por Giampiero Campa para el control de la tarjeta Arduino UNO desde Simulink por medio de la librería ArduinoIO. Las modificaciones realizadas para el proyecto de plataforma Stewart fueron la lectura por comunicación I2C del sensor IMU 6050, el filtrado de señales provenientes del sensor mediante el uso de filtros pasa bajas, el cálculo de los ángulos Pitch y Roll utilizando trigonometría y un filtro complementario con las mediciones del acelerómetro y el giroscopio y el envío de los ángulos antes mencionados vía comunicación serial hacia Simulink.

```
/* Analog and Digital Input and Output Server for MATLAB */
/* Giampiero Campa, Copyright 2012 The MathWorks, Inc */
/* Modified by Zhuo Li, 4/2013, to enable the I2C capability */

/* This file is meant to be used with the MATLAB arduino IO
   package, however, it can be used from the IDE environment
   (or any other serial terminal) by typing commands like:*/
#include <Servo.h>
#include <Wire.h>

/* define internal for the MEGA as I.IV (as as for the 328) */
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
#define INTERNAL INTERNALIV
#endif

//Ratios de conversion
#define A_R 16384.0
#define G_R 131.0

float AcX, AcY, AcZ, AcZw, AcXf, AcYf, AcZf;
float GyX, GyY, GyZ, GyXw, GyYw, GyZw, GyXf, GyYf, GyZf;

//Definición Vectores para filtro
float AcXv[5];
float AcYv[5];
float AcZv[5];
float GyXv[5];
float GyYv[5];
float GyZv[5];
```

```

//Angulos
float Acc[3];
float Gy[3];
float Angle[3];

//Offsets Definition
float AcZ_offset= 1803;
float GyX_offset= -19;
float GyY_offset= 90;
float GyZ_offset= 41;

float filterRK(float RawValue, float *Vector)
{
    float FilteredValue;
    //Se guardan los valores en los vectores
    Vector[3]= Vector[2];
    Vector[2]= Vector[1];
    Vector[1]= Vector[0];
    Vector[0]= RawValue;

    //Se calcula el valor filtrado
    FilteredValue = (1./6)*(Vector[0]+2*Vector[1]+2*Vector[2]+Vector[3]); //0.16666667
    return FilteredValue;
}

/* create servo vector */
Servo servo[70];

void setup() {
    /* Make sure all pins are put in high impedance state and
       that their registers are set as low before doing anything.
       This puts the board in a known (and harmless) state */
    int i;
    for (i=0;i<20;i++) {
        pinMode(i,INPUT);
        digitalWrite(i,0);
    }

    Wire.begin();
    Wire.beginTransmission(0x68);
    Wire.write(0x6B); //*Accede al registro 6B - Energiza el sensor
    Wire.write(0b00000000); //*Configura el modo Sleep a 0
    Wire.endTransmission(true);
    Wire.write(0x1B); //*Accede al registro 1B - Gyroscope Configuration (Sec. 4.4)
    Wire.write(0x00000000); //*Setting the gyro to full scale +/- 250deg./s
    Wire.beginTransmission(0b1101000); //*I2C address of the MPU

```

```

Wire.write(0x1C);      /*Accessing the register 1C - Acccelerometer Configuration (Sec. 4.5)
Wire.write(0b00000000); /*Setting the accel to +/- 2g
Wire.endTransmission();

/* initialize serial          */
Serial.begin(115200);
}

void loop() {

/* variables declaration and initialization      */

static int s = -1; /* state          */
static int pin = 13; /* generic pin number      */
static int enc = 0; /* generic encoder number      */

int val = 0; /* generic value read from serial */
int agv = 0; /* generic analog value      */
int dgv = 0; /* generic digital value      */

// for I2C communication
static byte addr = 0x00;
static byte dev = 0x00;
static byte reg = 0x00;
byte num = 2;
float valueI2C = 0x0000; // zero out the data

//Leer los valores del Acelerometro de la IMU
Wire.beginTransmission(0x68);
Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
Wire.endTransmission(false);
Wire.requestFrom(0x68,6,true); //A partir del 0x3B, se piden 6 registros
AcXf=(float) (Wire.read()<<8|Wire.read()); //Cada valor ocupa 2 registros
AcYf=(float) (Wire.read()<<8|Wire.read());
AcZf=(float) (Wire.read()<<8|Wire.read());

AcX = filterRK(AcXf,AcXv);
AcY = filterRK(AcYf,AcYv);
AcZw = filterRK(AcZf,AcZv);
AcZ=AcZw-AcZ_offset;

//A partir de los valores del acelerometro, se calculan los angulos Y, X
//respectivamente, con la formula de la tangente.
Acc[0] = (atan2((AcX/A_R), sqrt((AcY/A_R)*(AcY/A_R) + (AcZ/A_R)*(AcZ/A_R)))*180)/M_PI;
Acc[1] = (atan2((AcY/A_R), (AcZ/A_R))*180)/M_PI;

//Leer los valores del Giroscopio

```

```

Wire.beginTransmission(0x68);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(0x68,6,true); //A diferencia del Acelerometro, solo se piden 4 registros
GyXw=Wire.read()<<8|Wire.read();
GyYw=Wire.read()<<8|Wire.read();
GyZw=Wire.read()<<8|Wire.read();
GyXf=(float) (Wire.read()<<8|Wire.read());
GyYf=(float) (Wire.read()<<8|Wire.read());
GyZf=(float) (Wire.read()<<8|Wire.read());

GyXw = filterRK(GyXf,GyXv);
GyX=GyXw-GyX_offset;
GyYw = filterRK(GyYf,GyYv);
GyY=GyYw-GyY_offset;
GyZw = filterRK(GyZf,GyZv);
GyZ=GyZw-GyZ_offset;

//Calculo del angulo del Giroscopio
Gy[0] = GyX/G_R;
Gy[1] = GyY/G_R;
Gy[2] = GyZ/G_R;

//Aplicar el Filtro Complementario
Angle[0] = 0.96*(Angle[0]+Gy[0]*0.001) + 0.04*Acc[0];
Angle[1] = 0.96*(Angle[1]+Gy[1]*0.001) + 0.04*Acc[1];

delay(1); //Nuestra dt sera, pues, 0.010, que es el intervalo de tiempo en cada medida

/* The following instruction constantly checks if anything
   is available on the serial port. Nothing gets executed in
   the loop if nothing is available to be read, but as soon
   as anything becomes available, then the part coded after
   the if statement (that is the real stuff) gets executed */

if (Serial.available() >0) {

  /* whatever is available from the serial is read here */
  val = Serial.read();

  /* This part basically implements a state machine that
     reads the serial port and makes just one transition
     to a new state, depending on both the previous state
     and the command that is read from the serial port.
     Some commands need additional inputs from the serial
     port, so they need 2 or 3 state transitions (each one
     happening as soon as anything new is available from

```

the serial port) to be fully executed. After a command is fully executed the state returns to its initial value s=-1

```

switch (s) {

/* s=-1 means NOTHING RECEIVED YET ***** */
case -1:
/* calculate next state */
if (val>47 && val<90) {
/* the first received value indicates the mode
49 is ascii for I, ... 90 is ascii for Z
From this sketch was deleted s = 30,40,41,210,220,230,240,250.
s=0 is change-pin mode;
s=10 is D1; s=20 is D0; s=30 is A1; s=40 is A0;
s=50 is servo status; s=60 is aervo attach/detach;
s=70 is servo read; s=80 is servo write;
s=90 is query script type (1 basic, 2 motor);
s=210 is encoder attach; s=220 is encoder detach;
s=230 is get encoder position; s=240 is encoder reset;
s=250 is set encoder debounce delay;
s=340 is change analog reference;
s=400 example echo returning the input argument;
*/
s=10*(val-48);
}

/* the following statements are needed to handle
unexpected first values coming from the serial (if
the value is unrecognized then it defaults to s=-1) */
if ((s>90 && s<210) || (s>250 && s!=340 && s!=400)) {
s=-1;
}

/* the break statements gets out of the switch-case, so
/* we go back to line 97 and wait for new serial data */
break; /* s=-1 (initial state) taken care of */

/* s=0 or 1 means CHANGE PIN MODE */
case 0:
/* the second received value indicates the pin
from abs('c')=99, pin 2, to abs('i')=166, pin 69 */
if (val>98 && val<167) {
pin=val-97; /* calculate pin */
s=1; /* next we will need to get 0 or 1 from serial */
}
}

```

```

}
else {
  s=-1; /* if value is not a pin then return to -1 */
}
break; /* s=0 taken care of */

case 1:
/* the third received value indicates the value 0 or 1 */
if (val>47 && val<50) {
  /* set pin mode */
  if (val==48) {
    pinMode(pin,INPUT);
  }
  else {
    pinMode(pin,OUTPUT);
  }
}
s=-1; /* we are done with CHANGE PIN so go to -1 */
break; /* s=1 taken care of */

/* s=10 means DIGITAL INPUT ***** */
case 10:
/* the second received value indicates the pin
   from abs('c')=99, pin 2, to abs('i')=166, pin 69 */
if (val>98 && val<167) {
  pin=val-97; /* calculate pin */
  dgv=digitalRead(pin); /* perform Digital Input */
  Serial.println(dgv); /* send value via serial */
}
s=-1; /* we are done with DI so next state is -1 */
break; /* s=10 taken care of */

/* s=20 or 21 means DIGITAL OUTPUT ***** */
case 20:
/* the second received value indicates the pin
   from abs('c')=99, pin 2, to abs('i')=166, pin 69 */
if (val>98 && val<167) {
  pin=val-97; /* calculate pin */
  s=21; /* next we will need to get 0 or 1 from serial */
}
else {
  s=-1; /* if value is not a pin then return to -1 */
}
break; /* s=20 taken care of */

```

```

case 21:
/* the third received value indicates the value 0 or 1 */
if (val>47 && val<50) {
    dgv=val-48; /* calculate value */
    digitalWrite(pin,dgv); /* perform Digital Output */
}
s=-1; /* we are done with 00 so next state is -1 */
break; /* s=21 taken care of */

/* s=50 means SERVO STATUS (ATTACHED/DETACHED) ***** */
case 50:
/* the second value indicates the servo attachment pin
from abs('c')=99, pin 2, to abs('l')=166, pin 69 */
if (val>98 && val<167) {
    pin=val-97; /* calculate pin */
    dgv=servo[pin].attached(); /* read status */
    Serial.println(dgv); /* send value via serial */
}
s=-1; /* we are done with servo status so return to -1*/
break; /* s=50 taken care of */

/* s=60 or 61 means SERVO ATTACH/DETACH ***** */
case 60:
/* the second value indicates the servo attachment pin
from abs('c')=99, pin 2, to abs('l')=166, pin 69 */
if (val>98 && val<167) {
    pin=val-97; /* calculate pin */
    s=61; /* next we will need to get 0 or 1 from serial */
}
else {
    s=-1; /* if value is not a servo then return to -1 */
}
break; /* s=60 taken care of */

case 61:
/* the third received value indicates the value 0 or 1
0 for detach and 1 for attach */
if (val>47 && val<50) {
    dgv=val-48; /* calculate value */
    if (dgv) servo[pin].attach(pin); /* attach servo */
    else servo[pin].detach(); /* detach servo */
}
s=-1; /* we are done with servo attach/detach so -1 */
break; /* s=61 taken care of */

```

```

/* s=70 means SERVO READ ***** */
case 70:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('l')=166, pin 69 */
if (val>98 && val<167) {
    pin=val-97; /* calculate pin */
    agv=servo[pin].read(); /* read value */
    Serial.println(agv); /* send value via serial */
}
s=-1; /* we are done with servo read so go to -1 next */
break; /* s=70 taken care of */

```

```

/* s=80 or 81 means SERVO WRITE ***** */
case 80:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('l')=166, pin 69 */
if (val>98 && val<167) {
    pin=val-97; /* calculate pin */
    s=81; /* next we will need to get value from serial */
}
else {
    s=-1; /* if value is not a servo then return to -1 */
}
break; /* s=80 taken care of */

```

```

case 81:
/* the third received value indicates the servo angle */
servo[pin].write(val); /* write value */
s=-1; /* we are done with servo write so go to -1 next */
break; /* s=81 taken care of */

```

```

/* s=90 means Query Script Type (1 basic, 2 motor) *****/
case 90:
if (val==57) {
    /* if string sent is 99 send script type via serial */
    Serial.println(l);
}
s=-1; /* we are done with this so next state is -1 */
break; /* s=90 taken care of */

```

```

/* s=340 or 341 means ANALOG REFERENCE ***** */
case 340:
/* the second received value indicates the reference.

```

```

    which is encoded as is 0,1,2 for DEFAULT, INTERNAL
    and EXTERNAL, respectively          */
switch (val) {

    case 48:
        analogReference(DEFAULT);
        break;

    case 49:
        analogReference(INTERNAL);
        break;

    case 50:
        analogReference(EXTERNAL);
        break;

    default:          /* unrecognized, no action */
        break;
}
s--; /* we are done with this so next state is -1 */
break; /* s=34 taken care of          */

/* s=400 I2C read from slave ***** */
case 400: /* read slave address in string and convert to hex */
    addr = 0x00;
    addr = val;
    s=401; /* continue to 401 to get the register addr */
    break; /* s=400 taken care of          */

case 401: /* read register address          */
    reg = 0x00;
    reg = val;

    switch (reg){
    case 59:
        valueI2C = Angle[0]; //val = valueI2C;
        Serial.println(valueI2C);
        break;

    case 61:
        valueI2C = Angle[1]; //val = valueI2C;
        Serial.println(valueI2C);
        break;

    case 63:
        valueI2C = AcZ; //val = valueI2C;

```

```

Serial.println(value2C);
break;

default:
value2C = 0; //val = value2C;
Serial.println(value2C);
break;
}

s=-1; /* we are done with the aux function so -1 */
break; /* s=401 taken care of */

/* ***** UNRECOGNIZED STATE, go back to s=-1 ***** */
default:
/* we should never get here but if we do it means we
are in an unexpected state so whatever is the second
received value we get out of here and back to s=-1 */
s=-1; /* go back to the initial state, break unneeded */

} /* end switch on state s */

} /* end if serial available */

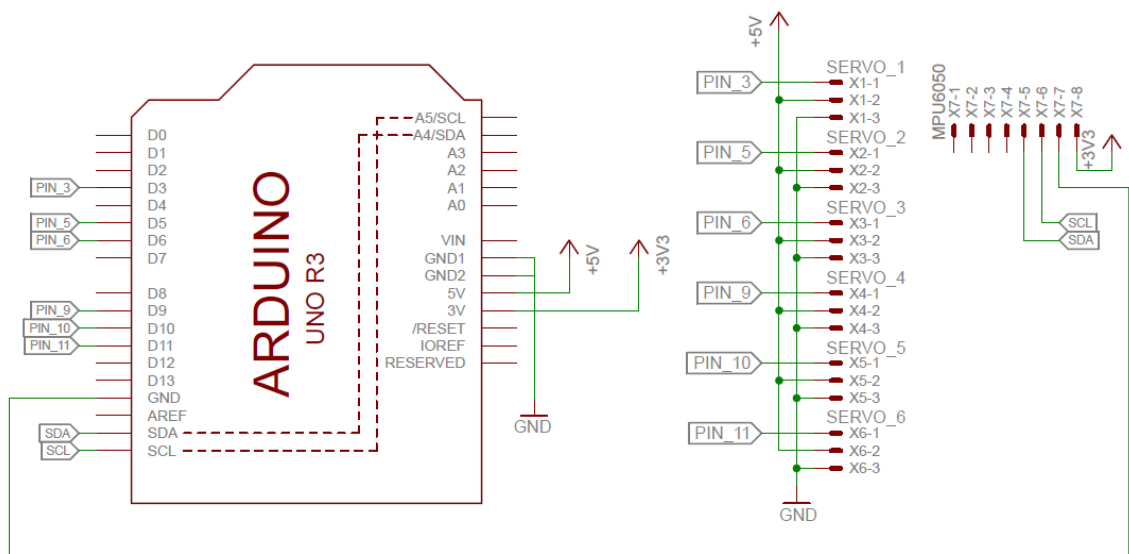
} /* end loop statement */

```

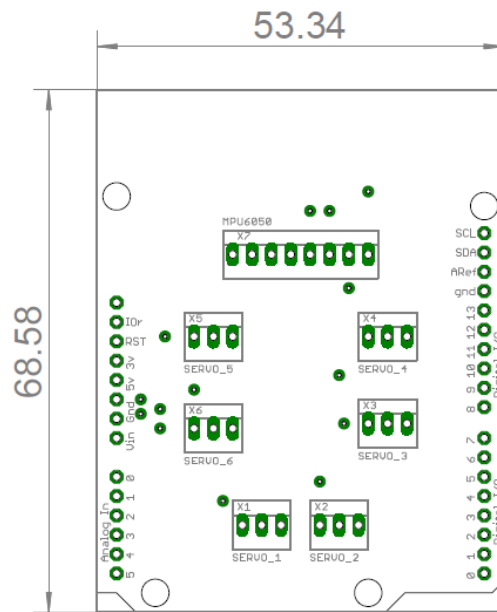
## ANEXO D. PLANOS DE DISEÑO DE LA TARJETA PCB

Este anexo contiene los diseños gráficos de la placa PCB fabricada para las conexiones internas de los servomotores y el dispositivo IMU sobre el Arduino.

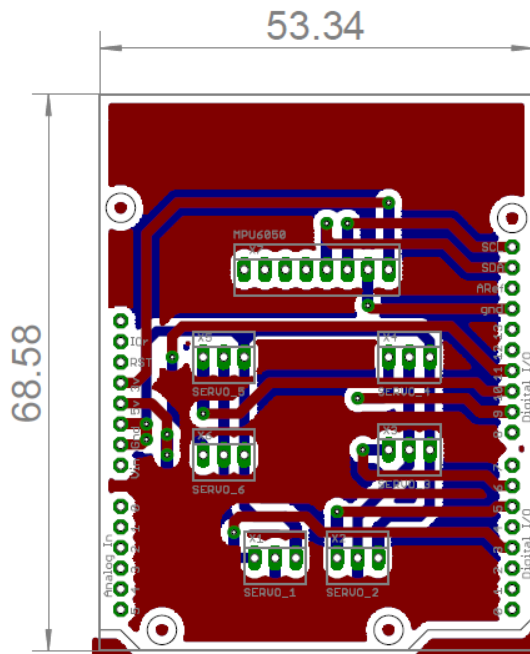
**Figura 72.** Vista en planta de la PCB.



**Figura 73.** Dimensión y distribución de pines sobre la PCB.



**Figura 74.** Enrutados eléctricos de la PCB.

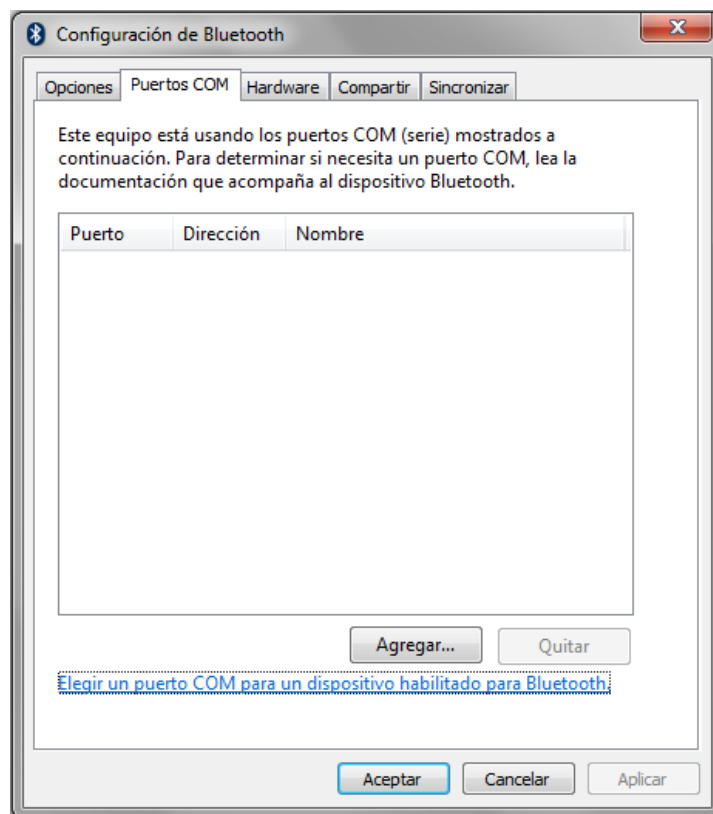


## ANEXO E. CREACIÓN DE UN DISPOSITIVO BLUETOOTH

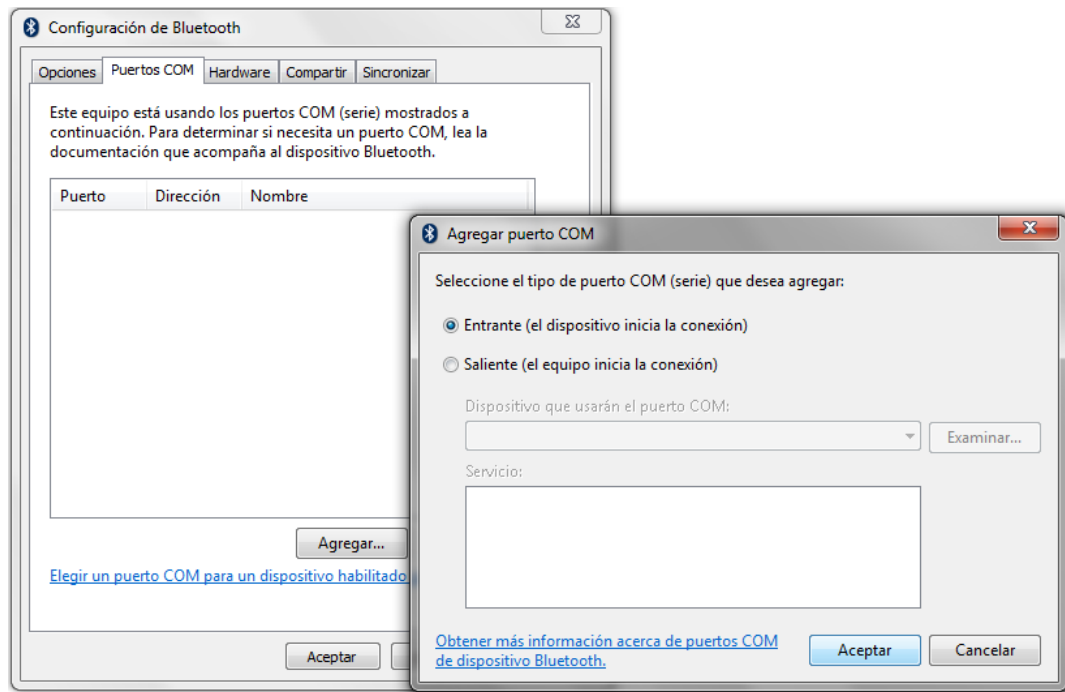
El presente anexo describe sistemáticamente el procedimiento para crear un dispositivo Bluetooth que comunique un Smartphone directamente con el computador que ejecuta el programa general de control de la plataforma Stewart diseñada en este proyecto y la creación de un puerto COM.

1. Acceder a la configuración *Bluetooth* del computador utilizando la siguiente ruta:

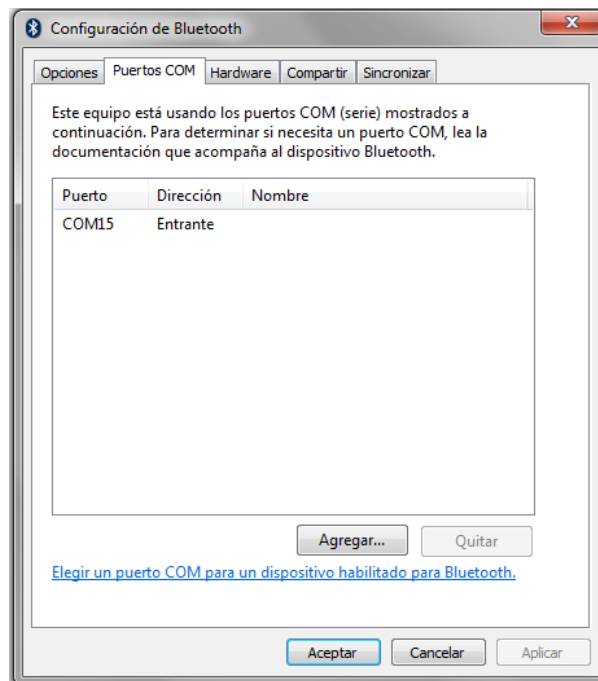
Panel de control > Dispositivos e Impresoras > Cambiar configuración de Bluetooth > Puertos COM. Aparecerá una ventana como la siguiente:



2. Se debe dar clic en la opción *Agregar...* donde se debe configurar el nuevo puerto COM como entrante de información tal como se ilustra en la siguiente imagen:

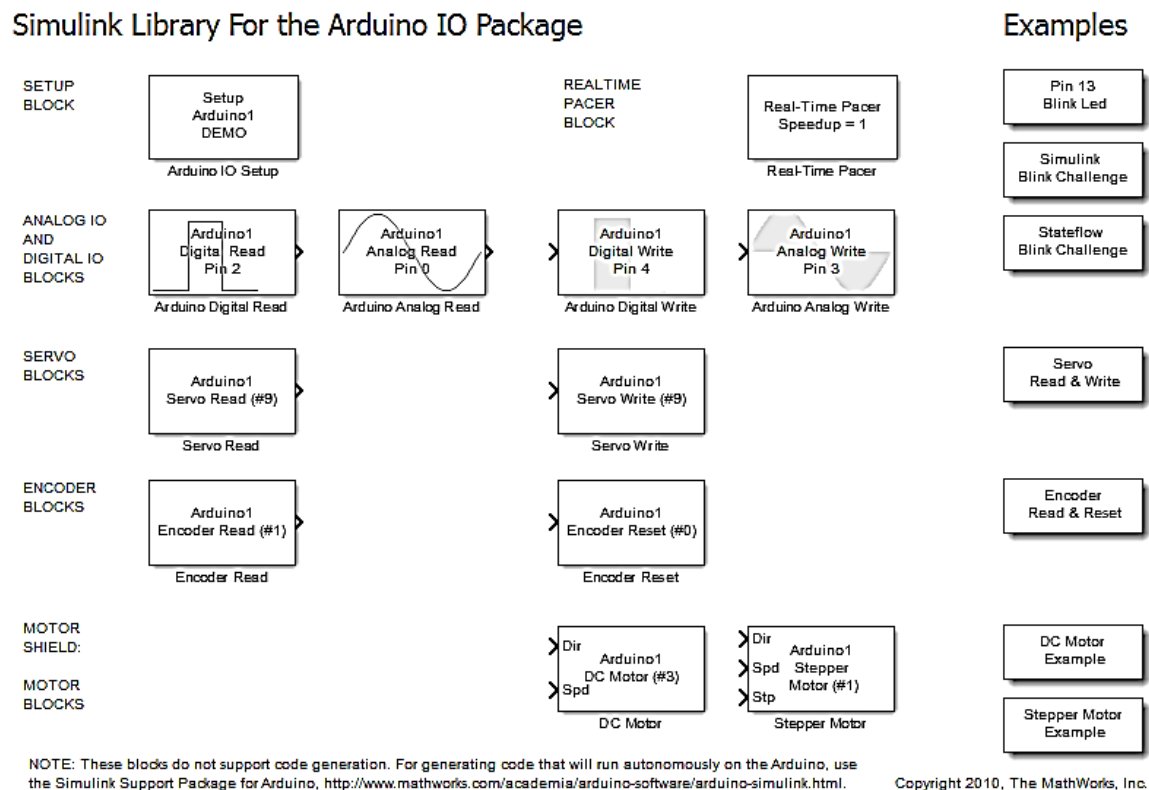


3. Se debe dar clic en el botón aceptar y con esto se ha creado el puerto COM, obteniendo la siguiente ventana en el explorador, en la cual se debe volver a aceptar la configuración.



## ANEXO F. INSTALACIÓN DE LA LIBRERÍA ARDUINO IO EN SIMULINK

Este anexo muestra el modo de la instalación de la librería Arduino IO en Matlab/Simulink que permite la comunicación analógica y digital con Arduino y de este modo poder obtener lectura o escritura sobre los puertos del Arduino a través del modelamiento de bloques en Simulink.



1. Descargar de la página de Mathworks el paquete ArduinoIO (ver el link: <http://www.mathworks.com/matlabcentral/fileexchange/47057-arduinoio-zip>).
2. Iniciar Matlab con el Current Folder del paquete descargado y escribir los siguientes comando en el Command Window:

```
>> install_arduino|
  Arduino folders added to the path
  Saved updated MATLAB path
```

3. Iniciar Simulink

## ANEXO G. INSTALACIÓN DEL COMPLEMENTO SIMMECHANICS

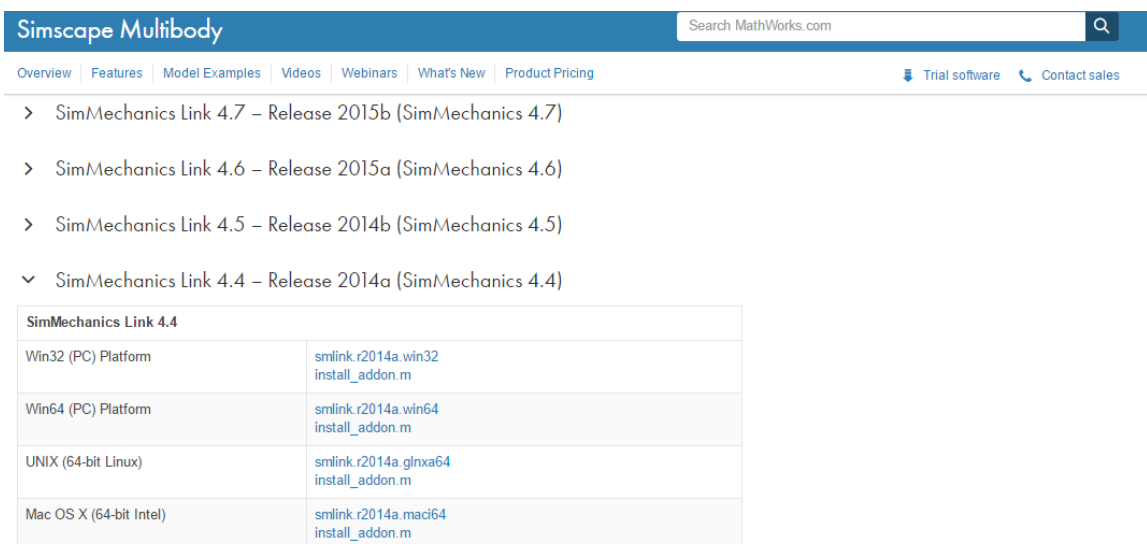
Este anexo contiene el instructivo para instalar el complemento Simmechanics en un ordenador con el *software* MatLab 2014. Sin embargo, en la página de Mathworks, es posible descargar el complemento para cualquier versión. Simmechanics es una librería disponible para Simulink que permite la simulación multicuerpo de sistemas mecánicos en tres dimensiones. Los bloques de simulación de Simmechanics representan componentes del sistema y sus relaciones, esta librería formula y resuelve las ecuaciones de movimiento teniendo en cuenta las propiedades respectivas de cada componente.

A continuación se listan los pasos para la instalación:

1. Ir al sitio web de descarga del complemento: Simscape Multibody Link

[https://de.mathworks.com/products/simmechanics/download\\_smlink\\_confirmation.html?elqsid=1484765873430&potential\\_use=Student](https://de.mathworks.com/products/simmechanics/download_smlink_confirmation.html?elqsid=1484765873430&potential_use=Student)

2. Escoger la pestaña referente a su versión de MatLab, en nuestro caso 2014a.



SimMechanics Link 4.4	
Win32 (PC) Platform	smlink_r2014a.win32 install_addon.m
Win64 (PC) Platform	smlink_r2014a.win64 install_addon.m
UNIX (64-bit Linux)	smlink_r2014a.glnxa64 install_addon.m
Mac OS X (64-bit Intel)	smlink_r2014a.maci64 install_addon.m

**3.** Escoger el sistema operativo del computador donde se va a instalar la librería y descargar los dos archivos disponibles, el archivo ZIP (smlink.r2014a) y el archivo con extensión de MatLab (install\_addon.m). Guardar dichos archivos preferiblemente en la carpeta predeterminada de archivos de Matlab.

**4.** Ejecutar MatLab como administrador y abrir la carpeta desde MatLab donde se guardaron los archivos del paso anterior.

**5.** Escribir el siguiente comando en el *comand window* de MatLab.

```
>> install_addon('smlink.r2014a.win64.zip')
```

Inmediatamente en el *comand window* aparecerá el siguiente mensaje:

```
Intalling smlink..
```

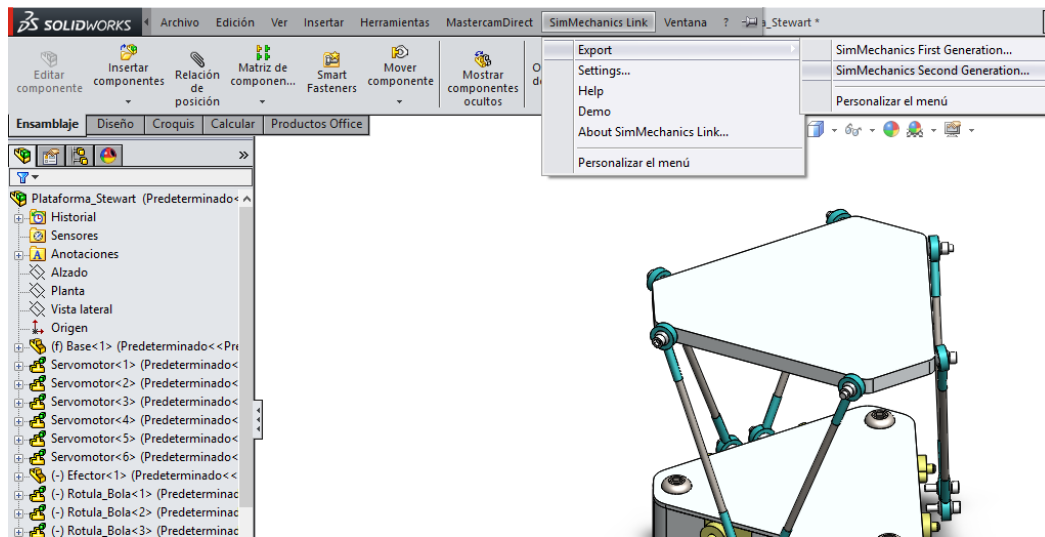
```
Extracting archive smlink.r2014a.win64.zip to C:\Program Files\MATLAB\R2014a...
```

**6.** Se debe habilitar el complemento en SolidWorks escribiendo el siguiente comando:

```
>> smlink_linksw
```

**7.** Antes de exportar el modelo CAD se debe habilitar el complemento en SolidWorks *SimMechanics Link*, dirigiéndose en la Barra de Herramientas de SolidWorks a la pestaña Herramientas > Complementos... > SimMechanics Link. Se debe chulear la opción SimMechanics Link.

**8.** Para exportar el modelo CAD en Simulink y crear un archivo con extensión .xml se debe dirigir a la Barra de Herramientas de SolidWorks y en la pestaña SimMechanics Link > Export > SimMechanics Second Generation... Tal como se muestra en la siguiente imagen:



9. Finalmente se importa el documento en MatLab escribiendo el comando *smimport* con el respectivo nombre del archivo CAD en el *comand window*, como se muestra a continuación:

```
>> smimport('Plataforma_Stewart.xml')
```