



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA MECATRÓNICA

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN MECATRÓNICA**

**AUTORES: MORILLO POVEDA EDUARDO ANDRÉS
PORTERO IBARRA JUAN CARLOS**

**TEMA: “DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO
DE ROBOT BÍPEDO AUTÓNOMO CON DESPLAZAMIENTO
DINÁMICO PARA EL LABORATORIO DE ROBÓTICA
INDUSTRIAL DE LA ESPE EXTENSIÓN LATACUNGA.”**

**DIRECTOR: ING. MARCO SINGAÑA
CODIRECTORA: ING. JACQUELINE LLANOS**

LATACUNGA, AGOSTO 2014

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**CARRERA DE INGENIERÍA MECATRÓNICA****CERTIFICADO**

ING. MARCO SINGAÑA (DIRECTOR)
ING. JACQUELINE LLANOS (CODIRECTORA)

CERTIFICAN:

Que el trabajo titulado **“Diseño y construcción de un prototipo de robot bípedo autónomo con desplazamiento dinámico para el laboratorio de Robótica Industrial de la ESPE Extensión Latacunga”** realizado por los señores: EDUARDO ANDRÉS MORILLO POVEDA y JUAN CARLOS PORTERO IBARRA, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE.

Latacunga, Agosto del 2014.

ING. MARCO SINGAÑA
DIRECTOR

ING. JACQUELINE LLANOS
CODIRECTORA

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

NOSOTROS, EDUARDO ANDRÉS MORILLO POVEDA
JUAN CARLOS PORTERO IBARRA

DECLARAMOS QUE:

El proyecto de grado denominado “**Diseño y construcción de un prototipo de robot bípedo autónomo con desplazamiento dinámico para el laboratorio de Robótica Industrial de la ESPE Extensión Latacunga**”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las referencias que constan en las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Agosto del 2014.

EDUARDO MORILLO POVEDA
CC. No.- 1802825958

JUAN PORTERO IBARRA.
CC. No.- 1804637674

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN

NOSOTROS, EDUARDO ANDRÉS MORILLO POVEDA
JUAN CARLOS PORTERO IBARRA

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo denominado **“Diseño y construcción de un prototipo de robot bípedo autónomo con desplazamiento dinámico para el laboratorio de Robótica Industrial de la ESPE Extensión Latacunga”** cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, Agosto del 2014.

EDUARDO MORILLO POVEDA
CC. No.- 1802825958

JUAN PORTERO IBARRA.
CC. No.- 1804637674

DEDICATORIA

Dedico este trabajo a mis padres Eduardo y Adela, que con todo su cariño, sabiduría y esfuerzo me han guiado durante toda mi vida; también dedico este trabajo a mi enamorada Valeria, sin ella mi vida no sería la misma, gracias a su amor incondicional e salido adelante en los momentos más difíciles y finalmente a mi amigo Juan Carlos, durante toda la carrera universitaria ha sido más que un compañero, un amigo de verdad y juntos logramos culminar este trabajo que nos costó mucho esfuerzo y dedicación pero finalmente lo logramos.

Eduardo Andrés.
eamp2289@gmail.com

A Dios por permitirme seguir aquí para llevar acabo mis sueños. A mis padres Cristóbal y Marlene que con su cariño, comprensión, ternura y sacrificio me han llevado a ser un profesional y más que eso una persona cabal y llena de principios. A mi tío Celiano que por causas del destino ya no se encuentra junto a mí, gracias por enseñarme que con valor y perseverancia todo es posible. A mis hermanos que con su ejemplo y consejos me ayudan a mejorar día con día. A mis dos sobrinitas que con sus ocurrencias me alegran el corazón y me inspiran a ser mejor. Finalmente a mi amigo Eduardo que más que un amigo se convirtió en un hermano, gracias por el apoyo y esfuerzo brindado sigamos adelante nos esperan más retos. Para ustedes y toda mi familia va dedicado la realización de este proyecto.

Juan Carlos.
jcportero89@hotmail.com

AGRADECIMIENTO

Agradecemos a Dios por brindarnos el despertar de cada día para lograr este sueño tan anhelado de culminar nuestra carrera universitaria.

El agradecimiento más especial y lleno de cariño a nuestros Padres, fuente inagotable de apoyo incondicional en la lucha por alcanzar nuestro sueño, gracias porque debido a sus sacrificios y su paciencia han guiado nuestro andar para que hoy en día podamos llegar a ser personas de bien y profesionales responsables.

A nuestros hermanos que han sido amigos sinceros e inseparables, que con su compañía durante toda nuestra vida fueron otra guía en nuestro andar, con cariño o frialdad un sabio consejo siempre nos podían entregar.

También no podemos olvidar aquellos compañeros de lucha que con sus ocurrencias y modo de actuar llenaron de buenas y malas experiencias, gracias a todos nuestros amigos y amigas.

A los docentes de la ESPE Extensión Latacunga que fueron los encargados de nuestra formación durante todos estos años.

Finalmente un agradecimiento especial a los ingenieros Marco Singaña y Jacqueline Llanos, quienes con sus conocimientos y palabras de apoyo nos supieron guiar y alentar a completar el desarrollo de nuestra tesis.

ÍNDICE DE CONTENIDOS

CERTIFICADO.....	I
AUTORÍA DE RESPONSABILIDAD.....	III
AUTORIZACIÓN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
ÍNDICE DE CONTENIDOS.....	VII
ÍNDICE DE TABLAS.....	XII
ÍNDICE DE FIGURAS.....	XIII
RESUMEN.....	XXI
ABSTRACT.....	XXII
PRESENTACIÓN.....	XXIII

CAPÍTULO I

GENERALIDADES.....	1
1.1. INTRODUCCIÓN.....	1
1.1.1. ROBÓTICA.....	1
1.1.2. ROBOT.....	2
1.1.3. LEYES DE LA ROBÓTICA.....	2
1.1.4. HISTORIA DE LA ROBÓTICA.....	3
1.1.5. MORFOLOGÍA DEL ROBOT.....	6
a. ESTRUCTURA MECÁNICA.....	6
b. SISTEMA DE ACCIONAMIENTO.....	6
c. SISTEMA SENSORIAL.....	6
d. SISTEMA DE CONTROL.....	7
e. ELEMENTOS TERMINALES.....	7
1.2. ROBOTS BÍPEDOS.....	8
1.2.1. ROBÓTICA MÓVIL.....	8
1.2.2. ROBÓTICA BÍPEDA.....	9
1.2.3. HISTORIA DE LA ROBÓTICA BÍPEDA.....	10
1.2.4. LOCOMOCIÓN HUMANA.....	22
a. CICLO DE LA MARCHA.....	23

1.2.5.	LOCOMOCIÓN DE ROBOTS BÍPEDOS.	24
a.	LOCOMOCIÓN ESTÁTICA.	24
b.	LOCOMOCIÓN DINÁMICA.	25
1.2.6.	CINEMÁTICA DE ROBOTS BÍPEDOS.	25
a.	CINEMÁTICA DIRECTA.	25
a.1	MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA.	26
a.2	REPRESENTACIÓN DE DENAVIT HARTENBERG (D-H).....	28
a.3	ALGORITMOS D-H PARA MODELADO CINEMÁTICO DIRECTO...	29
b.	CINEMÁTICA INVERSA.	31
1.2.7.	DINÁMICA DE ROBOTS BÍPEDOS.	32
1.2.8.	MATERIALES TÍPICAMENTE USADOS.	35
a.	MDF.	35
b.	ABS.	36
c.	ALUMINIO.	38
1.3.	CONTROLADORES AVANZADOS PARA ROBOTS.	38
1.3.1.	SISTEMAS DE CONTROL.	38
1.3.2.	CONFIGURACIONES DE SISTEMAS DE CONTROL.	39
a.	SISTEMAS DE CONTROL EN LAZO ABIERTO.	40
b.	SISTEMAS DE CONTROL EN LAZO CERRADO.	40
1.3.3.	TIPOS DE SISTEMAS DE CONTROL.	41
a.	SISTEMAS DE CONTROL CONVENCIONAL.	41
b.	SISTEMAS DE CONTROL AVANZADOS.	42
1.3.4.	TÉCNICA HEURÍSTICA.	43
a.	FUNCIÓN HEURÍSTICA.	44
1.4.	COMUNICACIÓN INALÁMBRICA.	47
1.4.1.	DEFINICIÓN.	47
1.4.2.	TECNOLOGÍA ZigBee.	47
1.5.	SERVOMOTORES.	49
1.5.1.	DEFINICIÓN.	50
1.5.2.	PARTES DE UN SERVOMOTOR.	50
1.6.	SOFTWARE.	52
1.6.1.	SolidWorks.	52
1.6.2.	Matlab.	52
a.	SimMechanics.	53
b.	Robotics Toolbox.	53
1.6.3.	Proteus.	54
1.6.4.	CURA.	54

1.6.5	PyPose.....	54
1.6.6	IDE de Arduino.	55

CAPÍTULO II

DISEÑO Y SELECCIÓN DE DISPOSITIVOS		56
2.1.	PARÁMETROS DE DISEÑO.	56
2.1.1.	PARÁMETROS MECÁNICOS.....	56
2.1.2.	PARÁMETROS DE COMUNICACIÓN Y ELECTRÓNICOS.....	57
2.1.3.	FUNCIONAMIENTO.....	57
2.2.	DISEÑO DEL SISTEMA MECÁNICO.	58
2.2.1.	DISEÑO DE LA ESTRUCTURA DEL ROBOT BÍPEDO.	58
2.2.2.	ANÁLISIS ESTÁTICO DE LA ESTRUCTURA DEL ROBOT.	66
2.2.3.	CINEMÁTICA DEL ROBOT BÍPEDO.	73
2.2.4.	DINÁMICA DEL ROBOT BÍPEDO.....	89
2.3.	DISEÑO DEL SISTEMA ELECTRÓNICO.....	94
2.3.1.	CIRCUITO DE ACCIONAMIENTO.....	94
2.3.2.	SERVOMOTORES.	96
2.3.3.	SENSOR.....	97
2.3.4.	TARJETA ELECTRÓNICA.	98
2.3.5.	DISPOSITIVOS DE COMUNICACIÓN.....	99
2.3.6.	FUENTE DE ENERGÍA.	100
2.4.	DISEÑO DEL SISTEMA DE CONTROL.	101
2.4.1.	MODELAMIENTO DEL ROBOT BÍPEDO.	101
2.4.2.	TRAYECTORIAS DE MOVIMIENTO DEL ROBOT BÍPEDO.....	106
2.4.3.	TÉCNICA DE CONTROL.	115
a.	DIAGRAMA DE FLUJO DEL SISTEMA DE CONTROL.	115
2.5.	SELECCIÓN DE ELEMENTOS Y MATERIALES.	117
2.5.1.	SERVOMOTORES.	117
2.5.2.	SENSOR.....	120
2.5.3.	TARJETA ELECTRÓNICA.	122
2.5.4.	DISPOSITIVOS DE COMUNICACIÓN.....	124
2.5.5.	FUENTE DE ENERGÍA.	128

CAPÍTULO III

CONSTRUCCIÓN E IMPLEMENTACIÓN.....		132
3.1.	MONTAJE DEL SISTEMA MECÁNICO.....	132
3.1.1.	FABRICACIÓN DE LA ESTRUCTURA DEL ROBOT.	132

3.1.2.	ENSAMBLAJE DEL ROBOT BÍPEDO.	135
a.	ENSAMBLAJE DE LAS EXTREMIDADES INFERIORES.	136
b.	CABLEADO DE LAS EXTREMIDADES INFERIORES.....	159
c.	ENSAMBLAJE DE LA PARTE SUPERIOR.....	163
3.2.	MONTAJE DEL SISTEMA ELECTRÓNICO.	173
3.2.1.	CONEXIONES DEL SISTEMA DE INSTRUMENTACIÓN.....	173
3.2.2.	CONEXIONES DEL CIRCUITO DE ACCIONAMIENTO.....	174
3.3.	IMPLEMENTACIÓN DEL SISTEMA DE CONTROL.	175
3.3.1.	INTEGRACIÓN TARJETA ARBOTIX-M CON ARDUINO.....	175
3.3.2.	ASIGNACIÓN DE ID EN LOS SERVOMOTORES.	179
3.3.3.	CONFIGURACIÓN MÓDULOS XBEE.	181
3.3.4.	SOFTWARE PARA ALMACENAR LAS POSICIONES.....	183
3.4.	INTEGRACIÓN DE SISTEMAS.....	187
3.5.	OBTENCIÓN DE LAS TRAYECTORIAS DEL ROBOT BÍPEDO.	196
3.6.	ALGORITMO DE CONTROL.....	200

CAPÍTULO IV

PRUEBAS Y RESULTADOS	207	
4.1.	PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA MECÁNICO.	207
4.1.1.	PRUEBAS DE RESISTENCIA DEL MATERIAL ABS.	207
4.1.2.	POSICIÓN DONDE SE REQUIERE MAYOR TORQUE.....	208
4.1.3.	VELOCIDAD DE DESPLAZAMIENTO DEL ROBOT BÍPEDO.	209
4.2.	PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA ELECTRÓNICO.....	213
4.2.1.	COMPROBACIÓN DE VOLTAJES.	213
4.2.2.	DURACIÓN DE LA BATERÍA.....	217
4.2.3.	DETECCIÓN DE OBSTÁCULOS.....	218
4.2.4.	DISTANCIA DE SENSADO.....	220
4.3.	PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA DE CONTROL.	221
4.3.1.	ERRORES DEL ALGORITMO DE CONTROL.....	221
4.3.2.	ACCIONES DEL SISTEMA DE CONTROL.	223
4.4.	PRUEBAS GENERALES DURANTE EL FUNCIONAMIENTO.	226
4.5.	VALIDACIÓN DE LA HIPÓTESIS.....	228
4.6.	ESTUDIO ECONÓMICO.	232

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES	234	
5.1.	CONCLUSIONES.....	234

5.2. RECOMENDACIONES.....	236
REFERENCIAS BIBLIOGRÁFICAS.	238
ANEXOS.	244

ÍNDICE DE TABLAS

Tabla 1.1. Historia de la robótica.	3
Tabla.1.2. Propiedades físico – mecánicas del MDF.	36
Tabla.1.3. Propiedades físico – mecánicas del ABS.	37
Tabla.1.4. Propiedades físico – mecánicas del aluminio.	38
Tabla 1.5. Características del XBee Serie 1.	49
Tabla 2.1. Elementos para el ensamblaje del robot bípedo.	67
Tabla 2.2. Parámetros D-H.	78
Tabla 2.3. Características de los servomotores.	117
Tabla 2.4. Especificaciones del servomotor Dynamixel AX-12A.	120
Tabla 2.5. Características de los sensores.	120
Tabla 2.6. Especificaciones del sensor Parallax PING.	121
Tabla 2.7. Características de las tarjetas electrónicas.	122
Tabla 2.8. Especificaciones de la tarjeta electrónica ArbotiX-M.	124
Tabla 2.9. Características de los dispositivos de comunicación.	125
Tabla 2.10. Especificaciones del UartSBee V4.	126
Tabla 2.11. Especificaciones del XBee S1.	128
Tabla 2.12. Características de las baterías.	129
Tabla 2.13. Especificaciones de la batería Tenergy.	131
Tabla 2.14. Especificaciones del Cargador Tenergy.	131
Tabla 3.1. Tiempos de impresión de las piezas del robot bípedo.	135
Tabla 3.2. Parámetros XBee para la comunicación.	183
Tabla 4.1. Parámetros del ensayo de tracción.	208
Tabla 4.2. Velocidad promedio.	213
Tabla 4.3. Resultados de la duración de la batería.	217
Tabla 4.4. Resultados de la detección de obstáculos.	220
Tabla 4.5. Resultados de la distancia de sensado.	221
Tabla 4.6. Datos observados para el equilibrio.	229
Tabla 4.7. Datos esperados para el equilibrio.	229
Tabla 4.8. Datos observados para el sensado.	231
Tabla 4.9. Datos esperados para el sensado.	231
Tabla 4.10. Costos del proyecto.	233

ÍNDICE DE FIGURAS

Figura 1.1. Robot móvil con patas.	9
Figura 1.2. Robot bípedo.	9
Figura 1.3. Robot WABOT-1.	11
Figura 1.4. Robot Wasubot.	11
Figura 1.5. Robot Honda E0.	12
Figura 1.6. Robots Honda a) E1, b) E2 y c) E3.	13
Figura 1.7. Robot Shadow Walker.	13
Figura 1.8. Robots Honda a) E4, b) E5 y c) E6.	14
Figura 1.9. Robot Honda P-1.	14
Figura 1.10. Robot Honda P-2.	15
Figura 1.11. Robot Honda P-3.	16
Figura 1.12. Robot HRP-1S.	16
Figura 1.13. Robot ASIMO 2000.	17
Figura 1.14. Robot QRIO.	17
Figura 1.15. Robot PARTNER.	18
Figura 1.16. Robot HOAP-3.	18
Figura 1.17. Robot del NICT.	19
Figura 1.18. Robot NAO.	19
Figura 1.19. Robot KOBIAN.	20
Figura 1.20. Robot ASIMO 2011.	20
Figura 1.21. Robot KIROBOT.	21
Figura 1.22. Robot Humanoide Bioloid.	21
Figura 1.23. Robot Surena II.	22
Figura 1.24. Componentes de la marcha.	24
Figura 1.25. Parámetro D-H para un eslabón giratorio.	28
Figura 1.26. Polígono de soporte.	34
Figura 1.27. Margen de estabilidad.	35
Figura 1.28. Sistemas de control.	39
Figura 1.29. Sistema de control en lazo abierto.	40
Figura 1.30. Sistema de control en lazo cerrado.	41
Figura 1.31. Estructura interna de un servomotor.	51
Figura 2.1. Extremidad inferior del cuerpo humano.	58
Figura 2.2. Pelvis del robot bípedo.	60
Figura 2.3. Cadera izquierda del robot bípedo.	61

Figura 2.4. Fémur izquierdo del robot bípedo.....	61
Figura 2.5. Tibia izquierda del robot bípedo.	62
Figura 2.6. Tobillo izquierdo del robot bípedo.....	62
Figura 2.7. Pie izquierdo del robot bípedo.	63
Figura 2.8. Ensamblaje de las extremidades inferiores del robot bípedo.....	63
Figura 2.9. Tronco del robot bípedo.....	64
Figura 2.10. Brazo izquierdo del robot bípedo.....	64
Figura 2.11. Cabeza del robot bípedo.	65
Figura 2.12. Ensamblaje del robot bípedo.	65
Figura 2.13. Ensamblaje del robot bípedo con materiales asignados.....	67
Figura 2.14. Vector gravedad del estudio estático.	68
Figura 2.15. Geometría fija en los pies.	69
Figura 2.16. Torque en las piezas.	69
Figura 2.17. Torques y fuerzas ejercidas en el robot bípedo.	70
Figura 2.18. Mallado de la estructura del robot bípedo.	70
Figura 2.19. Tensiones de Von Mises en la estructura del robot bípedo.	71
Figura 2.20. Factor de seguridad en la estructura del robot bípedo.....	73
Figura 2.21. Enumeración de eslabones D-H.....	74
Figura 2.22. Localización de los ejes en las articulares D-H.	75
Figura 2.23. Enumeración de articulaciones D-H.	75
Figura 2.24. Localización de los ejes D-H.	76
Figura 2.25. Obtención de los parámetros D-H.	77
Figura 2.26. Parámetros para obtener el comando <i>DHFactor</i>	81
Figura 2.27. Obtención de la longitud <i>L2</i>	83
Figura 2.28. Posición del efector final del robot bípedo.....	85
Figura 2.29. Pierna del robot bípedo en la posición inicial.	86
Figura 2.30. Pierna del robot bípedo visualización correcta.....	87
Figura 2.31. Valores articulares en la posición inicial.	87
Figura 2.32. Ejemplo cinemática directa del robot bípedo.....	88
Figura 2.33. Valores articulares del ejemplo de cinemática directa del robot bípedo.	89
Figura 2.34. Propiedades físicas del fémur izquierdo.....	90
Figura 2.35. Datos dinámicos de los eslabones.	91
Figura 2.36. Datos dinámicos en el SerialLink.....	92
Figura 2.37. Posición donde se requiere el mayor torque.	93
Figura 2.38. Diagrama de los componentes electrónicos.....	94
Figura 2.39. Circuito de accionamiento en ISIS.....	95
Figura 2.40. Circuito de accionamiento en ARES.	95

Figura 2.41. Enlace entre SimMechanics y SolidWorks.	102
Figura 2.42. Piezas del segundo ensamblaje.	103
Figura 2.43. Segundo ensamblaje de las extremidades inferiores.	103
Figura 2.44. Exportación del ensamblaje desde SolidWorks a Matlab.	104
Figura 2.45. Modelado del robot bípedo en SimMechanics.	104
Figura 2.46. Subsistemas del modelado SimMechanics.	105
Figura 2.47. Parámetros dentro de los bloques Body de SimMechanics.	105
Figura 2.48. Sensores y actuadores de SimMechanics.	106
Figura 2.49. Trayectoria de la cadera.	107
Figura 2.50. Trayectoria de la rodilla.	107
Figura 2.51. Trayectoria del tobillo.	108
Figura 2.52. Trayectoria de la cadera en Robotics Toolbox.	108
Figura 2.53. Trayectoria de la rodilla en Robotics Toolbox.	109
Figura 2.54. Trayectoria del tobillo en Robotics Toolbox.	110
Figura 2.55. Trayectorias de las articulaciones.	110
Figura 2.56. Subsistemas con la trayectoria implementada.	111
Figura 2.57. Secuencia de caminado del robot bípedo en SimMechanics.	113
Figura 2.58. Trayectoria de la cadera (Simulink).	113
Figura 2.59. Trayectoria de la rodilla (Simulink).	114
Figura 2.60. Trayectoria del tobillo (Simulink).	114
Figura 2.61. Diagrama de flujo del sistema de control.	116
Figura 2.62. Servomotor Dynamixel AX-12A.	119
Figura 2.63. Sensor Parallax PING.	121
Figura 2.64. Tarjeta electrónica ArbotiX-M.	123
Figura 2.65. UartSBee V4.	126
Figura 2.66. Cable FTDI (comunica el UartSBee V4 con la tarjeta).	127
Figura 2.67. Cables mini USB (comunica la PC con el UartSBee).	127
Figura 2.68. XBee S1.	127
Figura 2.69. Batería NiMH Tenenergy.	130
Figura 2.70. Cargador universal inteligente Tenenergy.	131
Figura 3.1. Diagrama de operación de procesos del fémur derecho externo.	133
Figura 3.2. Elementos de sujeción de los servomotores.	136
Figura 3.3. Disposición de tuercas en los servomotores ID1 e ID2.	136
Figura 3.4. Disposición de tuercas en los servomotores ID3 e ID4.	137
Figura 3.5. Disposición de tuercas en los servomotores ID5 e ID6.	137
Figura 3.6. Disposición de tuercas en los servomotores ID7 e ID8.	137
Figura 3.7. Disposición de tuercas en los servomotores ID9 e ID10.	138

Figura 3.8. Disposición de tuercas en los servomotores ID11 e ID12.....	138
Figura 3.9. Inserción de las tuercas en el servomotor 1.....	138
Figura 3.10. Coxis parte superior.....	139
Figura 3.11. Disposición de los tornillos S1 en el servomotor 1.....	139
Figura 3.12. Ensamblaje A.....	139
Figura 3.13. Ensamblaje A y tornillos S1 para formar el ensamblaje B.....	140
Figura 3.14. Ensamblaje B.....	140
Figura 3.15. Coxis parte inferior.....	141
Figura 3.16. Ensamblaje B y tornillos para formar el ensamblaje C.....	141
Figura 3.17. Ensamblaje C.....	141
Figura 3.18. Cadera derecha e izquierda.....	142
Figura 3.19. Ubicación de las piezas que conforman la cadera.....	142
Figura 3.20. Disposición de los tornillos S1 en la cadera.....	142
Figura 3.21. Ensamblaje D.....	143
Figura 3.22. Disposición de los tornillos S1 que sujetan los servomotores 3 y 4.....	143
Figura 3.23. Ubicación de los servomotores 3 y 4.....	144
Figura 3.24. Ensamblaje E.....	144
Figura 3.25. Ensamblaje E con el giro de los servomotores 3 y 4.....	145
Figura 3.26. Piezas del fémur de la pierna derecha e izquierda.....	145
Figura 3.27. Ensamblaje del fémur derecho externo.....	146
Figura 3.28. Ensamblaje del fémur izquierdo externo.....	146
Figura 3.29. Ensamblaje F.....	146
Figura 3.30. Disposición de las piezas y tornillos S1 para el ensamblaje G.....	147
Figura 3.31. Ensamblaje G.....	147
Figura 3.32. Ubicación de los servomotores 5 y 6.....	148
Figura 3.33. Disposición de los tornillos S1 para el ensamblaje H.....	148
Figura 3.34. Ensamblaje H.....	149
Figura 3.35. Piezas de la tibia de las piernas derecha e izquierda.....	149
Figura 3.36. Ensamblaje de la tibia derecha externa.....	150
Figura 3.37. Ensamblaje de la tibia izquierda externa.....	150
Figura 3.38. Ensamblaje I.....	150
Figura 3.39. Disposición de las piezas 6 y 7 junto con los tornillos S1.....	151
Figura 3.40. Ensamblaje J.....	151
Figura 3.41. Ubicación de los servomotores 7 y 8.....	152
Figura 3.42. Ensamblaje K.....	152
Figura 3.43. Piezas de los tobillos.....	153
Figura 3.44. Sub-ensamblaje de las piezas de los tobillos.....	153

Figura 3.45. Colocación de las piezas 9 y 10.	154
Figura 3.46. Disposición de tornillos S1 de la parte frontal de los tobillos.	154
Figura 3.47. Disposición de tornillos SB de la parte posterior de los tobillos.	154
Figura 3.48. Ensamblaje L.	155
Figura 3.49. Colocación de las piezas para los servomotores 9 y 10.	155
Figura 3.50. Ubicación de los servomotores 9 y 10.	156
Figura 3.51. Disposición de los tornillos S1 y SB del tobillo.	156
Figura 3.52. Ensamblaje M.	157
Figura 3.53. Colocación del pie izquierdo.	157
Figura 3.54. Disposición de los tornillos S1 en el pie izquierdo.	158
Figura 3.55. Colocación del pie derecho.	158
Figura 3.56. Disposición de los tornillos S1 en el pie derecho.	159
Figura 3.57. Ensamblaje de las extremidades inferiores.	159
Figura 3.58. Conexión del cable del servomotor 9.	160
Figura 3.59. Conexión del cable entre los servomotores 9 y 7.	160
Figura 3.60. Conexión del cable entre los servomotores 7 y 5.	161
Figura 3.61. Conexión del cable entre los servomotores 5 y 3.	161
Figura 3.62. Conexión de los servomotores de la pierna derecha.	162
Figura 3.63. Cableado de las extremidades inferiores.	162
Figura 3.64. Espalda del robot.	163
Figura 3.65. Correcto posicionamiento de la espalda.	163
Figura 3.66. Disposición de los tornillos S1 para la espalda.	164
Figura 3.67. Disposición de los tornillos S1 en el servomotor 11.	164
Figura 3.68. Ensamblaje N.	165
Figura 3.69. Disposición de los tornillos S1 en el servomotor 12.	165
Figura 3.70. Ensamblaje O.	166
Figura 3.71. Disposición de los tornillos para la tarjeta.	166
Figura 3.72. Colocación de la tarjeta.	166
Figura 3.73. Colocación de la batería.	167
Figura 3.74. Cableado de los servomotores 11 y 12.	167
Figura 3.75. Pecho del robot.	168
Figura 3.76. Disposición de los tornillos S1 para el pecho.	168
Figura 3.77. Ensamblaje P.	169
Figura 3.78. Cabeza del robot.	169
Figura 3.79. Cuello del robot.	169
Figura 3.80. Sub-ensamble Q.	170
Figura 3.81. Ubicación de los tornillos para el sensor.	170

Figura 3.82. Montaje del sensor.	170
Figura 3.83. Colocación de los cables del sensor.	171
Figura 3.84. Ensamblaje R.....	171
Figura 3.85. Disposición de los tornillos S1 para el brazo izquierdo.	172
Figura 3.86. Disposición de los tornillos S1 para el brazo derecho.....	172
Figura 3.87. Ensamblaje final del robot bípedo.	172
Figura 3.88. Colocación del circuito de accionamiento.	173
Figura 3.89. Conexión pines del sensor con la tarjeta Arbotix-M.	174
Figura 3.90. Conexión pines de la tarjeta Arbotix-M con el circuito de accionamiento. .	174
Figura 3.91. Versión de Arduino.	175
Figura 3.92. Pantalla principal del programa Arduino.....	176
Figura 3.93. Drivers FTDI.....	176
Figura 3.94. Distribución de las carpetas Arduino.	177
Figura 3.95. Comprobación Arbotix en Arduino.....	178
Figura 3.96. Hardware Arbotix-M con PC.	178
Figura 3.97. Jumper de energía Arbotix-M con PC.	179
Figura 3.98. Hardware Arbotix-M con servomotor.....	180
Figura 3.99. Jumper energía Arbotix-M con servomotor.	180
Figura 3.100. Pantalla principal DynaManager.....	181
Figura 3.101. Hardware XBee con PC.....	182
Figura 3.102. Comprobación del programa Python.	184
Figura 3.103. Pantalla principal del programa PyPose.....	185
Figura 3.104. Nuevo proyecto en PyPose.	186
Figura 3.105. Proyecto bípedo en PyPose.	186
Figura 3.106. Robot bípedo.	187
Figura 3.107. Componentes electrónicos.	187
Figura 3.108. Posición inicial en PyPose.....	188
Figura 3.109. Posición inicial.	188
Figura 3.110. Posición del brazo derecho en PyPose.....	189
Figura 3.111. Posición del brazo derecho.	189
Figura 3.112. Posición del brazo izquierdo en PyPose.	190
Figura 3.113. Posición del brazo izquierdo.....	190
Figura 3.114. Posición inclinada en PyPose.....	191
Figura 3.115. Posición inclinada.	191
Figura 3.116. Editor de secuencias.	192
Figura 3.117. Datos de la secuencia.	192
Figura 3.118. Opción export to AVR.	193

Figura 3.119. Librerías de los servomotores y secuencias en Arduino.	194
Figura 3.120. Librería de la secuencia en Arduino.	194
Figura 3.121. Agregar la secuencia en Arduino.	195
Figura 3.122. Características de la secuencia Arduino.	195
Figura 3.123. Agregar velocidad de transmisión Arduino.	196
Figura 3.124. Caminata del robot bípedo en el plano sagital.	197
Figura 3.125. Caminata del robot bípedo en el plano frontal.	197
Figura 3.126. Giro a la derecha del robot bípedo en el plano sagital.	198
Figura 3.127. Giro a la derecha del robot bípedo en el plano frontal.	198
Figura 3.128. Giro a la izquierda del robot bípedo en el plano sagital.	198
Figura 3.129. Giro a la izquierda del robot bípedo en el plano frontal.	199
Figura 3.130. Secuencia avanzar.	199
Figura 3.131. Secuencia giro derecha.	200
Figura 3.132. Secuencia giro izquierda.	200
Figura 3.133. Librerías de Arduino.	201
Figura 3.134. Velocidad de transmisión.	201
Figura 3.135. Declaración de variables.	201
Figura 3.136. Programación dentro del void setup.	202
Figura 3.137. Programación dentro del void loop.	203
Figura 3.138. Subprograma avanzar.	204
Figura 3.139. Subprograma sensor.	204
Figura 3.140. Subprograma girar.	205
Figura 3.141. Subprograma girar1.	205
Figura 3.142. Subprograma secuencias.	205
Figura 4.1. Posición donde se requiere mayor torque.	209
Figura 4.2. Tiempo de ejecución en el programa PyPose.	210
Figura 4.3. Desplazamiento del robot bípedo.	210
Figura 4.4 Velocidad vs Estabilidad.	212
Figura 4.5. Medición de voltaje en la batería.	214
Figura 4.6. Medición de voltaje del switch (estado OFF).	214
Figura 4.7. Medición de voltaje del switch (estado ON).	215
Figura 4.8. Voltaje en la entrada de la batería con los estados del switch.	215
Figura 4.9. Conexión de la etapa del botón.	216
Figura 4.10. Medición de voltaje en la salida del botón sin presionar.	216
Figura 4.11. Medición de voltaje en la salida del botón.	216
Figura 4.12. Obstáculo pared color verde claro.	218
Figura 4.13. Obstáculo madera color café oscuro.	218

Figura 4.14. Obstáculo espejo.	219
Figura 4.15. Obstáculo cartulina color blanco.	219
Figura 4.15. Obstáculo color negro.	219
Figura 4.17. Distancia de sensado.	220
Figura 4.18. Error 1 del algoritmo de control.	222
Figura 4.19. Error 2 del algoritmo de control.	222
Figura 4.20. Error 3 del algoritmo de control.	223
Figura 4.21. Robot bípedo en la posición erguida.	223
Figura 4.22. Robot bípedo caminando.	224
Figura 4.23. Robot bípedo curvando hacia la derecha 90 grados.	224
Figura 4.24. Robot bípedo curvando hacia la izquierda 180 grados.	225
Figura 4.25. Robot bípedo curvando hacia la izquierda 90 grados.	225
Figura 4.26. Prueba uno de la caminata del robot bípedo.	226
Figura 4.27. Prueba dos de la caminata del robot bípedo.	227
Figura 4.28. Prueba tres de la caminata del robot bípedo.	228

RESUMEN

En el presente proyecto se presenta el “Diseño y construcción de un prototipo de robot bípedo autónomo con desplazamiento dinámico para el laboratorio de Robótica Industrial de la ESPE Extensión Latacunga”. El objetivo del proyecto es desarrollar un robot bípedo capaz de caminar autónomamente, con la capacidad de detectar y esquivar los obstáculos que se encuentren frente a él a una distancia aproximada de 40 [cm]. La investigación se inicia al realizar un estado del arte de los fundamentos para la construcción de un prototipo de robot bípedo y las diferentes plataformas informáticas para su desarrollo; a partir de los parámetros de diseño obtenidos del estado del arte se diseña el robot bípedo en el programa SolidWorks en este mismo programa se realiza el análisis estático de la estructura; en el Robotics Toolbox de Matlab se efectúa el estudio cinemático y dinámico; el modelado matemático se obtiene mediante SimMechanics de Matlab de esta manera se simula el modelo físico del robot bípedo. Se realiza el diseño electrónico y la técnica de control dando paso a la selección y construcción de todos los componentes que conforman el robot bípedo; se implementan todos los sistemas y se obtienen las trayectorias permitiendo la realización de las distintas pruebas con sus respectivos análisis y resultados. Finalmente en base a los resultados obtenidos se emiten las conclusiones y recomendaciones del proyecto.

PALABRAS CLAVE:

INGENIERÍA MECATRÓNICA\ ROBÓTICA\ ROBOT BÍPEDO\
AUTÓNOMO\ DISEÑO Y CONSTRUCCIÓN

ABSTRACT

"Design and construction of an autonomous biped robot prototype with dynamic displacement for the Laboratory of Industrial Robotics ESPE Latacunga Extension" is presented in this project. The project objective is to develop a biped robot able to walk independently, with the ability to detect and avoid obstacles that are in front of him at a distance of approximately 40 [cm]. The investigation begins to performance the foundations for building a biped robot prototype and different computing platforms for development; from the design parameters of the prior art obtained the biped robot and static analysis of the structure are made in SolidWorks; kinematic, dynamic studies and mathematical modeling are obtained by Matlab, thus the physical model of the biped robot is simulated. Electronic design and control technology leading to the selection and construction of all components that make up the biped robot is performed; all systems are implemented and the trajectories are obtained of the different tests and their analysis results. Finally based on results, conclusions and recommendations of the project are issued.

KEY WORDS:

MECHATRONICS ENGINEERING\ ROBOTICS\ BIPED ROBOT\
AUTONOMOUS\ DESIGN AND CONSTRUCTION

PRESENTACIÓN

En el presente proyecto se describe el diseño y construcción de un prototipo de robot bípedo autónomo con desplazamiento dinámico para el laboratorio de robótica industrial de la ESPE Extensión Latacunga.

El Capítulo I, contiene el estado del arte de los robots bípedos e información y conocimientos necesarios de los temas fundamentales para el desarrollo del proyecto.

El Capítulo II, detalla de forma ordenada el diseño de las etapas mecánica, electrónica y de control en sus respectivos software; la selección de componentes de acuerdo a los parámetros establecidos para la creación del robot bípedo.

El Capítulo III, describe la construcción e implementación de los sistemas mecánico, electrónico y de control; la implementación del algoritmo de control y la obtención de las trayectorias que definen la caminata del robot bípedo.

En el Capítulo IV, se realizan pruebas y se emiten resultados de los sistemas mecánicos, electrónico y de control, para realizar pruebas de funcionamiento generales; además se presenta la validación de la hipótesis y el análisis económico de la realización del proyecto.

El Capítulo V, muestra las conclusiones y recomendaciones del proyecto.

Se incluye los anexos y las referencias bibliográficas para profundizar en los distintos temas.

CAPÍTULO 1

GENERALIDADES

1.1. INTRODUCCIÓN.

En este capítulo se desglosarán diferentes temas de interés, los mismos que tienen un orden, el cual va de acuerdo a como se debe ir avanzando en el conocimiento de cada uno de los conceptos que intervienen en un robot bípedo, conceptos como la evolución, desarrollo de la robótica y diversos argumentos más con los que se desarrollará este proyecto, al ir profundizando en este capítulo el conocimiento abarcará un campo amplio debido a que se involucrarán temas del campo electrónico, mecánico, software y de control.

1.1.1. ROBÓTICA.

La robótica es una rama de la tecnología integrada por un conjunto de conocimientos teóricos y prácticos que permiten el diseño, construcción, operación y aplicación de los robots; la robótica además combina diversas disciplinas como la física, mecánica, electrónica, informática, ingeniería de control e inteligencia artificial.

La palabra Robótica se origina cuando aparecen los primeros manipuladores tele-operados, inicialmente estos dispositivos no adoptaron la denominación de robot sino hasta que el operario fue sustituido por un programa de ordenador para dar las ordenes de movimiento al manipulador, todo esto debido a la necesidad de automatizar los procesos industriales para ganar tiempo de producción y generar líneas de manufactura con mayor eficiencia y menor intervención de la mano humana proclive a errores.

1.1.2. ROBOT.

En el término robot confluyen las imágenes de máquinas para la realización de trabajos productivos, de imitación de movimientos y comportamientos de seres vivos. Los robots actuales son obras de ingeniería y como tales concebidas para producir bienes y dar servicios o explotar recursos naturales. En nuestro siglo el desarrollo de máquinas ha estado fuertemente influido por el progreso tecnológico, de esta forma se pasa de máquinas que tienen como objetivo exclusivo la amplificación de la potencia muscular del hombre, sustituyéndolo en su trabajo físico, a máquinas o instrumentos que son también capaces de procesar información, complementando, o incluso sustituyendo, al hombre en algunas actividades intelectuales (Ollero, 2001, p. 1).

Desde el punto de vista literario o cinematográfico en el año de 1921 con la obra de teatro “RUR” (Rossum Universal Robota), del checo Karel Capek, es donde surge la palabra robot (robota en checo significa esclavo, o trabajador obligado). Más tarde esta visión fue reforzada por Fritz Lang en su película “Metropolis”, de 1926, y bastante después, una versión más humanizada es la que presenta Isaac Asimov en su serie de relatos, escritos a partir de 1942 (en los que, por cierto, se introduce por primera vez el término Robótica con el sentido de disciplina científica encargada de construir y programar robots) (Esteve, 2001).

1.1.3. LEYES DE LA ROBÓTICA.

Estas leyes fueron obra de Isaac Asimov en el relato “El círculo vicioso de Asimov” en el año 1942.

- Un robot no puede hacer daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.
- Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la primera Ley.

- Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la primera o la segunda Ley (Jiménez, 2013).

A pesar de que estas leyes están en medio del entorno de la ciencia ficción se puede utilizar en el mundo de la ciencia y tecnología, ya que su objetivo principal es proteger al ser humano y de ahí que viene una cuarta ley.

- Un robot no puede hacer daño a la humanidad o, por inacción, permitir que la humanidad sufra daño (Jiménez, 2013).

1.1.4. HISTORIA DE LA ROBÓTICA.

Desde el principio de la historia humana el hombre se ha sentido atraído y fascinado por máquinas y dispositivos que le ayuden a realizar movimientos que emulen su comportamiento, además de desempeñar tareas específicas; los griegos llamaban a estas máquinas “automatos” de la cual deriva la palabra *autómata* que hace referencia a los aparatos que imitan la forma y movimientos de un ser animado. A continuación en la Tabla 1.1 se muestra los autómatas más relevantes a lo largo de la historia.

Tabla 1.1. Historia de la robótica.

Año	Creador	Creación
62 A.C.	Herón de Alejandría	Teatro automático.
1200	Al-Jazari	Fuente del pavo real.
D.C.		
1500	Leonardo Da Vinci	Caballero con armadura medieval, fue el primer autómata con forma completamente humana.
1525	Juanelo Turriano	Monje autómata el cual caminaba y movía la cabeza.
1738	Jacques de Vaucanson	Pato autómata muy famoso que simulaba el aparato digestivo de un pato real.
1770	Jaquet-Droz	Creó diversos muñecos capaces de escribir, dibujar y tocar melodías.
1796	Hanzo Hosokawa	Muñeca que servía el té y era capaz de caminar.

Continúa 

1805	Familia Maillardet	Muñeca capaz de escribir en varios idiomas.
1912	Leonardo Torres	Creó el "Telekino" precursor de los vehículos teleoperados.
1921	Karel Capek	Obra de teatro R.U.R. aparece el término "robot".
1926	Fritz Lang	Lleva al cine la novela "Metrópolis".
1942	Isaac Asimov	Se le atribuye la creación del término "robotics" además q enunció las tres leyes de la robótica.
1948	R. C. Goertz	Crea el primer sistema de tele-manipulación.
1954	George C. Devol	Creó el primer robot programable, una máquina programada de transferencia de artículos.
1955	Denavit y Hartenberg	Representan y describen la geometría espacial de los elementos de un robot utilizando el álgebra matricial.
1959	Planet Corporation	Crean el primer robot comercial el cual estaba controlado por interruptores de finales de carrera.
1962	H.A. Ernest	Desarrolla una mano mecánica controlada por sensores táctiles.
1963	Marina Norteamericana	Desarrolló el primer robot submarino.
1966	Ole Molaug	Primer robot para aplicaciones de pintura.
1966	Standford Research Institute	Robot móvil "Shakey".
1967	NASA	"Surveyor-3" toma muestras de la luna por medio de un brazo robótico.
1977	George Lucas	Introduce en el cine los robots C3PO y R2D2, los cuales tenían autonomía e inteligencia artificial.
1980	Universidad Rhode Island	Sistema con visión capaz de captar piezas en orientaciones aleatorias.
1984	RDS (Robot Defense Systems)	"Prowler" el primer robot con fines militares.
1985	Universidad de Waseda	El profesor Ichiro desarrolla uno de los primeros robots humanoides, este era capaz de tocar el piano. La evolución de los robots humanoides se tratará en la página 10 de este documento, y no se tomarán en cuenta en esta sección.
1996	D Barrett	"RoboTuna" para el estudio del movimiento de especies marinas.
1999	LEGO	Lanza al mercado los robots Mindstorms.

Continúa 

1999	Sony	Lanza una mascota canina llamada "AIBO".
2003	Steve Potter	Creación de robots vinculados a tejido neuronal.
2005	Kuka Roboter GmbH	Montaje y Puesta en Servicio de la Primera Solución Robótica en Minería en Codelco Norte.
2006	Gil Weinberg	Presentó un robot llamado Haile, un robot baterista que usando un algoritmo puede escuchar el trabajo de un músico y crear un ritmo propio para acompañar la tonada.
2007	Touch Bionics	La primera mano "biónica" disponible comercialmente en el mundo, una prótesis con cinco dígitos que pueden ser funcionar individualmente.
2008	Automation Partnership	El Sistema glacial es un robot capaz de "cuidar" a millones de muestras biológicas en temperaturas de -80° centígrados.
2009	Instituto Nacional de Tecnología y Ciencia Industrial Avanzada de Tokio	Presenta al público a HRP-4C una ginoide que tiene una altura de 158 [cm] y pesa 43 [Kg]. Su inteligencia artificial le permite el reconocimiento y síntesis del habla.
2010	Hiroshi Ishiguro	El robot humanoide más avanzado del mundo llamado "Geminoid-DK", su apariencia y movimientos lo hacen semejante a un humano.
2011	Masahito Yamaguchi	Da a conocer el robot ciclista que es capaz de replicar las funciones motoras del ser humano, mantener el equilibrio, detectar y girar en las curvas y tiene la capacidad para frenar utilizando los pies.
2012	Equipo de la Universidad de Tecnología de Tokio, dirigido por Motomu Nakashima	"Swumanoid" es un robot humanoide que sabe nadar en diversos estilos. El humanoide fue hecho mediante el escaneo en 3D de los movimientos de un humano para permitir a los investigadores medir las diferentes fuerzas requeridas en los diferentes estilos de la natación.
2013	Rehink Robotics	Aparece Baxter, donde sus fabricantes aseguran que es una máquina con "sentido común", que se adapta a su entorno.

Fuente: (Barrientos, Penin, Balaguer, & Aracil, 2007, pp. 2-15), (Singaña, 2012, pp. 7-18), (Letraherido, 2009).

En la última década se han ido desarrollado cada vez más sorprendentes avances en el campo de la robótica, son muchos proyectos innovadores que cada día aparecen a lo largo del mundo, robots diseñados y creados por grandes empresas o por estudiantes universitarios, se han creado proyectos inimaginables hace algunos años, en campos de exploración marina, exploración subterránea, en el espacio, medicina, militar, académico, entretenimiento, entre otros.

1.1.5. MORFOLOGÍA DEL ROBOT.

Un robot está formado por los siguientes elementos:

a. ESTRUCTURA MECÁNICA.

Todo robot está constituido por una serie de eslabones unidos mecánicamente por articulaciones, las cuales permiten un movimiento relativo entre cada dos eslabones consecutivos, este movimiento puede ser de desplazamiento (prismático), de giro (rotación) o una combinación. Cada uno de los movimientos que puede realizar cada articulación con respecto a la anterior se denomina grado de libertad o GDL (Barrientos, Penin, Balaguer, & Aracil, 2007, pp. 31-32).

b. SISTEMA DE ACCIONAMIENTO.

Son los actuadores los que se encargan de generar el movimiento de las articulaciones del robot según las órdenes dadas por la unidad de control. En robótica se utilizan actuadores neumáticos, hidráulicos y eléctricos, entre cada una de estas fuentes de energía hay que considerar diferentes características que serán clave al momento de elegir el actuador ideal para cada aplicación, estas características son: potencia, controlabilidad, peso, volumen, precisión, velocidad, mantenimiento y coste.

c. SISTEMA SENSORIAL.

Para que un robot realice su trabajo con eficiencia, velocidad y precisión es necesario que tenga presente su propio desenvolvimiento y el estado en

el que se encuentra dentro de su entorno. La información de su propio estado se la consigue utilizando sensores internos también llamados propioceptivos y la información del estado de su entorno se adquiere con los sensores externos también llamados exteroceptivos.

- **Sensores Internos:** Estos sensores son ocupados básicamente para conocer la posición y velocidad de la estructura mecánica y sus articulaciones. Los sensores más utilizados para medir la posición son los encoders incrementales y absolutos, potenciómetros y LVDT (Transformador Diferencial de Variación Lineal); los sensores más utilizados para medir la velocidad son los tacogeneradores y LVT (Transductores de Velocidad Lineal).
- **Sensores Externos:** Para que un robot pueda interactuar con el medio en donde se encuentra se utilizan sensores externos, hay una gran variedad de estos sensores entre los más utilizados están los de ultrasonido para detectar obstáculos, unidades inerciales para controlar el equilibrio, de contacto, fuerza y torsión por medio de galgas de deformación y sensores de sonido.

d. SISTEMA DE CONTROL.

Los robot se pueden controlar por medio de software a través de algoritmos de control, en los últimos años se utiliza controladores avanzados como la lógica difusa, control adaptivo, control estocástico, redes neuronales, métodos heurísticos e inteligencia artificial; estos algoritmos se crean utilizando hardware digital como: computador, microcontroladores incluidos en tarjetas electrónicas o FPGA (del inglés Field Programmable Gate Array).

e. ELEMENTOS TERMINALES.

También son llamados efectores finales, éstos están encargados de interactuar y actuar directamente sobre el entorno del robot.

1.2. ROBOTS BÍPEDOS.

En este apartado se tratará el origen de los robots bípedos, historia, desarrollo, locomoción y propiedades físicas.

1.2.1. ROBÓTICA MÓVIL.

Ollero declara que el desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos, se trata también de incrementar la autonomía limitando todo lo posible la intervención humana (Ollero, 2001, p. 8).

El robot móvil hace uso de su capacidad de locomoción para interactuar con el entorno que lo rodea por ende necesitará un sistema de sensores que lo ayuden a localizar objetos (obstáculos) y definir la trayectoria para alcanzar su objetivo por medio de un sistema de control avanzado que le ayude a tomar decisiones.

Dentro de los robots móviles se pueden considerar los robots móviles terrestres (UGV), aéreos (UAV) y submarinos (AUV).

Los robots móviles terrestres se clasifican de acuerdo al modo de su desplazamiento:

- Robots móviles con ruedas.
- Robots móviles articulados.
- Robots móviles con patas.

Los robots móviles con patas se desplazan por aquellas zonas a las que un vehículo con ruedas no puede alcanzar, el uso de patas permite al robot desplazarse en terrenos escarpados evitando y pasando por encima de los obstáculos.

Algo muy importante en este tipo de robots es el equilibrio, debido al alto número de grados de libertad y a todos los movimientos que realizan las articulaciones para cumplir las trayectorias requeridas.

Se han desarrollado prototipos imitando a los animales de la naturaleza como se observa en la Figura 1.1 y también imitando al ser humano por eso de acuerdo al número de patas, este tipo de robots adquiere su denominación, como por ejemplo, los robots de dos patas se denominan bípedos, de cuatro patas cuadrúpedos, de seis patas hexápodos, etc., (Bonell, Diseño y Construcción de un Robot Humanoide, 2011, p. 18).



Figura 1.1. Robot móvil con patas.

Fuente: http://www.rpp.com.pe/2013-10-04-crean-robot-de-cuatro-patas-y-a-motor-que-corre-mas-rapido-que-bolt-noticia_636865.html

1.2.2. ROBÓTICA BÍPEDA.

Estos robots forman parte de la robótica móvil con patas, los cuales tienen dos extremidades inferiores para realizar su desplazamiento, por eso su nombre de bípedos, se puede observar un ejemplo de este tipo de robots en la Figura 1.2.

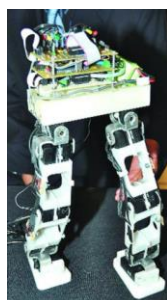


Figura 1.2. Robot bípedo.

Fuente: <http://www.cronica.com.mx/notas/2010/535397.html>

Cabe recalcar que dentro de este género los más conocidos son los humanoides, al querer imitar el comportamiento del caminado, acciones efectuadas y apariencia del ser humano que por excelencia es el ser vivo que mejor aprovecha la locomoción bípeda, pero no son los únicos modelos de robots bípedos, en la naturaleza muchos animales se desplazan de manera bípeda como los pingüinos y aves en general, canguros y otros marsupiales, algunos lagartos, simios y por eso se han realizado varios proyectos a lo largo de la historia que intentan imitar su comportamiento.

La caminata bípeda al momento de imitar es una de las más complicadas en comparación con cuadrúpedos o hexápodos debido a la complejidad de mantener el equilibrio, por eso, es esta la limitación de los proyectos de investigación científica en esta área de la robótica, muchas empresas ya consolidadas tienen grandes resultados pero esto debido a años de investigación, muchos ingenieros atrás del proyecto y presupuestos muy altos.

Además hay que aclarar que un robot bípedo no necesariamente puede ser un robot humanoide, pero todos los robots humanoides son robots bípedos.

1.2.3. HISTORIA DE LA ROBÓTICA BÍPEDA.

A finales de los años 60 en Japón se crea un nuevo interés por los robots que se asemejen al ser humano, tanto en su apariencia como comportamiento, fue un gran reto al inicio pero mientras pasaba el tiempo y con el conocimiento y tecnología cada vez más avanzados se han llegado a construir robots bípedos y humanoides sorprendentes hasta la actualidad.

A continuación se muestra los robots bípedos más relevantes a lo largo de la historia.

- **1970. Universidad de Waseda:** Creó el grupo de bioingeniería con el objetivo de construir robots que ayudaran en tareas de cuidado diario, enfermería y medicina; este proyecto se llamó "WABOT" y fue el inicio

de una amplia familia de robots bípedos y humanoides (López, 2012, p. 14).

- **1973. Universidad de Waseda:** Desarrolló el “WABOT-1” (Figura 1.3) este robot fue el primero con inteligencia artificial, tenía la capacidad de comunicarse con los seres humanos, de calcular distancias y de agarrar y transportar objetos.



Figura 1.3. Robot WABOT-1.

Fuente: http://www.humanoid.waseda.ac.jp/booklet/kato_2.html

- **1985. Universidad de Waseda:** El profesor Ichiro Kato creó el robot humanoide denominado “Wasubot” (Figura 1.4), este robot no podía caminar pero podía tocar el piano (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 12).



Figura 1.4. Robot Wasubot.

Fuente:

<http://books.google.com.ec/books?id=ixtL54iiDPUC&printsec=frontcover#v=onepage&q&f=false>

- **1986. Honda:** En el estudio de los principios fundamentales de la locomoción bípeda, Honda investigó y observó todas las formas de caminar (caminata rápida y caminata lenta), realizó numerosos experimentos y recogió una inmensa cantidad de datos con el robot “E0” mostrado en la Figura 1.5, el cual caminaba poniendo una pierna antes que la otra, se logró con éxito la locomoción, pero sólo podía caminar lentamente en línea recta y cada paso tardaba cinco segundos (Honda, 2014).



Figura 1.5. Robot Honda E0.

Fuente: <http://world.honda.com/ASIMO/history/e0/index.html>

- **1987 – 1991. Honda:** Inician la investigación del caminado rápido o también llamado caminado dinámico, Honda investigó y analizó la marcha humana a fondo (localización de las articulaciones de la pierna, rango de movimiento de las articulaciones, determinación de las dimensiones, peso y centro de gravedad de las piernas, medición de los torques que se producen mientras se produce la caminata, utilización de sensores para comprender como se desarrolla la caminata y determinación de las fuerzas de impacto son los análisis e investigaciones realizadas por Honda para la marcha humana). También se estudiaron los animales y otras formas de caminar, además de la marcha humana, también fueron investigados el movimiento y la ubicación de las articulaciones necesarias para caminar. El robot “E1” caminaba estáticamente a 0.25 [km/h], el “E2” fue el primer robot que podía caminar dinámicamente alcanzando una

velocidad de 1.2 [km/h], también podía subir o bajar escaleras; finalmente el “E3” alcanzó una velocidad de 3 [km/h] sobre superficies planas, todos estos robots se muestran en la Figura 1.6 (Honda, 2014).



Figura 1.6. Robots Honda a) E1, b) E2 y c) E3.

Fuente: http://world.honda.com/ASIMO/history/e1_e2_e3/index.html

- **1988.** Se crea el robot “Shadow Walker” (Figura 1.7), cuyo propósito es el de ayudar con la investigación y desarrollo para nuevos diseños y técnicas sobre equilibrio y locomoción humana.



Figura 1.7. Robot Shadow Walker.

Fuente: <http://www.lettraherido.com/19040302robots.htm>

- **1991 – 1993. Honda:** Honda llevó a cabo la investigación sobre la tecnología necesaria para lograr estabilizar la caminata y desarrolló con éxito tres tecnologías de control de postura al momento de caminar tomando en cuenta el punto de momento cero o ZMP, el

robot “E4” llega a una velocidad de 4.7 [km/h], el “E5” fue el primer modelo de locomoción autónoma y el modelo “E6” tenía un control autónomo de equilibrio, al pasar sobre superficies irregulares y al bajar y subir gradas, todos estos robots se muestran en la Figura 1.8 (Honda, 2014).



Figura 1.8. Robots Honda a) E4, b) E5 y c) E6.

Fuente: http://world.honda.com/ASIMO/history/e4_e5_e6/index.html

- **1992. Universidad de Waseda:** Para dar continuidad al proyecto “WABOT” se desarrolló el robot “WABIAN” con mejores características y prestaciones.
- **1993. Honda:** Presentó el robot “P-1” (Figura 1.9), siendo este el primer prototipo de un modelo de hombre de esta empresa, contaba con miembros superiores y todo su cuerpo imitando al del ser humano (Honda, 2014).



Figura 1.9. Robot Honda P-1.

Fuente: http://world.honda.com/ASIMO/history/p1_p2_p3/index.html

- **1996. Honda:** Presentó el robot “P-2” (Figura 1.10), un humanoide con apariencia de un humano en traje espacial, fue producto de diez años de investigación y más de 100 millones de dólares de inversión (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 12).

Además era capaz de compensar las fuerzas externas, de evitar colisiones y de caminar sobre superficies irregulares. Contaba con cámaras, acelerómetros, giroscopios y sensores de fuerza en los tobillos y muñecas lo que le permitía ser operado de forma inalámbrica (López, 2012, p. 17).



Figura 1.10. Robot Honda P-2.

Fuente: http://world.honda.com/ASIMO/history/p1_p2_p3/index.html

- **1997. Honda:** Presentó el robot “P-3” (Figura 1.11), una versión mejorada en peso y autonomía de su antepasado el “P-2” (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 12).

Este robot usaba demostraciones de ciclos de marcha realizados por humanos para generar las trayectorias de referencia que debía seguir, la técnica de control usada para la generación de las trayectorias se llama “punto objetivo de movimiento”, la misma que consiste en que el robot parte de la posición erguida y mediante una cámara detecta un punto objetivo, inmediatamente el algoritmo de control calcula la trayectoria con la menor cantidad de movimientos posibles para llegar a este punto, se usa una base de datos de ciclos

de marcha realizados por humanos (López, 2012, p. 17) y (Honda, 2014).



Figura 1.11. Robot Honda P-3.

Fuente: http://world.honda.com/ASIMO/history/p1_p2_p3/index.html

- **1998. Kawada Industries:** El “HRP” (Figura 1.12), tiene 30 grados de libertad, lo que le permite tener una sofisticada coordinación de las extremidades superiores, de esta manera tiene la facilidad de manipular objetos (Letraherido, 2009).



Figura 1.12. Robot HRP-1S.

Fuente: <http://www.letraherido.com/19040302robots.htm>

- **2000. Honda:** Lanza la última versión evolucionada del “P-3”, éste robot lo bautizaron como “ASIMO” (Advanced Step in Innovative Mobile) (Figura 1.13), su aplicación principal es la de dar asistencia a las personas; esta última versión dejó de ser una máquina automática y pasó a ser una máquina autónoma debido a que puede tomar

decisiones para determinar su comportamiento en conjunto con su entorno.



Figura 1.13. Robot ASIMO 2000.

Fuente: <http://world.honda.com/ASIMO/technology/index.html>

- **2003. Sony:** La empresa Japonesa lanza al mercado el robot “QRIO” (Quest for Curiosity) mostrado en la Figura 1.14, el cual es la evolución de varios de sus predecesores como el “SDR-4X” y “SDR-3X”, el control de movimiento está basado en la teoría del ZMP, este robot puede caminar, correr y saltar; su aplicación es la de servicio a los seres humanos y entretenimiento (Sony, 2014).



Figura 1.14. Robot QRIO.

Fuente: http://www.sony.net/SonyInfo/News/Press_Archive/200312/03-060E/

- **2005.** El “PARTNER” (Figura 1.15), es capaz de tocar la trompeta, ya que tiene la capacidad de mover sus labios como los de un ser humano con 1.80 [m] de altura, esto le permite cargar a una persona

de un lugar a otro, este fue el motivo de su creación, para ayudar a las personas de la tercera edad (Letraherido, 2009).



Figura 1.15. Robot PARTNER.

Fuente: <http://www.letraherido.com/19040302robots.htm>

- **2005. Fujitsu:** En este año salió la tercera versión del robot “HOAP” (Humanoid for Open Architecture Platform) que se muestra en la Figura 1.16, cuya primera versión se lanzó en el año 2001 y la segunda versión en el año 2003, es muy utilizado en los departamentos de ingeniería mecánica de las universidades, así como en las áreas de investigación robótica (García, 2012, p. 13).



Figura 1.16. Robot HOAP-3.

Fuente: García (2012). Interfaz de control cinemático para el humanoide HOAP-3 en Matlab para el cálculo off-line de trayectorias. P.13.

- **2007. Japan’s National Institute of Information and Communications Technology (NICT):** Desarrolla un robot que es

capaz de reconocer el lenguaje corporal de los humanos, es decir, sus gestos y tacto para de esta manera tener una comunicación más eficaz con los humanos, este robot se muestra en la Figura 1.17 (Letraherido, 2009).



Figura 1.17. Robot del NICT.

Fuente: <http://www.letraherido.com/19040302robots.htm>

- **2008. Aldebaran Robotics:** Esta empresa Francesa lanza al mercado el robot “NAO” mostrado en la Figura 1.18, que es un organismo autónomo, programable y de mediana estatura, es el robot oficial de la ROBOCUP con 21 grados de libertad y también es utilizada la versión de 25 grados de libertad en empresas y universidades (Wikipedia, Wikipedia, 2013).

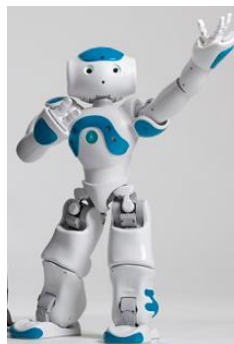


Figura 1.18. Robot NAO.

Fuente: [http://www.aldebaran-robotics.com/en/Discover-NAO/images-gallery.html#!ARGallery\[gallery1\]/1/](http://www.aldebaran-robotics.com/en/Discover-NAO/images-gallery.html#!ARGallery[gallery1]/1/)

- **2009. Tmsuk y la Universidad de Waseda:** “KOBIAN” (Figura 1.19), es el “Robot Humanoide Emocional” usa no solo su cara, sino su cuerpo entero, para demostrar sus emociones (Letraherido, 2009).



Figura 1.19. Robot KOBIAN.

Fuente: <http://www.letraherido.com/19040302robots.htm>

- **2011. Honda:** En este año se presenta la última versión de “ASIMO”, mostrada en la Figura 1.20, ya no es una máquina autónoma sino una máquina con autonomía capaz de tomar decisiones con relación a su entorno y a la gente que lo rodea, esta versión es el último logro tras las versiones del 2002, 2005 y 2007 (Honda, 2014).



Figura 1.20. Robot ASIMO 2011.

Fuente: <http://world.honda.com/ASIMO/technology/2011/index.html>

- **2013. Toyota:** Esta empresa Japonesa en conjunto con varias empresas más han desarrollado el robot “KIROBOT” que se muestra en la Figura 1.21, cuya misión es interactuar con personas que viven

solas, debido a su edad o algún problema, tiene una interfaz totalmente nueva en comparación con otros robots humanoides, éste robot también está presente en el espacio en la Estación Espacial Internacional.



Figura1.21. Robot KIROBOT.

Fuente: <http://kibo-robo.jp/en/gallery>

- **Actualidad:** En los últimos años se han creado muchos robots humanoides de pequeño tamaño cuya principal función es didáctica, utilizados mayoritariamente en el área académica, por citar algunos están los robots: “Rovonova 1”, “RoboSapien”, “i-Sobot”, “BRAT” de la empresa Lynxmotion y una especial mención al robot humanoide “Bioloid” mostrado en la Figura 1.22, el cual fue creado por la empresa coreana Robotis.



Figura 1.22. Robot Humanoide Bioloid.

Fuente: http://www.robotis.com/x/bioloid_en

También se han creado robots humanoides de tamaño casi natural a una persona, éstos han sido desarrollados en su mayoría por universidades en colaboración con grandes empresas, por citar algunos están los robots del grupo Partner Robot Project de Toyota, Los humanoides “MAHRU” y “AHRA” de Samsung, “Surena II” de la universidad de Teherán (Figura 1.23), “TOPIO” creado por TOSY Robotics y “REEM-A” de la empresa española PAL Robotics (López, 2012, pp. 19-22).



Figura 1.23. Robot Surena II.

Fuente: <http://www.fayerwayer.com/2010/07/surena-ii-el-primer-robot-humanoide-de-iran/>

1.2.4. LOCOMOCIÓN HUMANA.

La caminata humana es una actividad aprendida en la infancia, tarda un tiempo hasta que el sistema nervioso central coordine los movimientos con cada una de las extremidades y el tronco para lograr dar un par de pasos, se requieren además estabilidad y movilidad para lograr caminar de forma continua y poder realizar cambios de dirección o cambios de velocidad.

No todas las personas caminan de la misma manera, debido a la contextura, el calzado o alguna característica física o fisiológica propia de cada individuo, pero estas diferencias caen dentro de un cierto rango.

Para que un individuo pueda andar, el sistema locomotor debe cumplir los siguientes requisitos (Sáenz, 2012, p. 7).

- Cada pierna debe ser capaz de soportar el peso del cuerpo sin colapsar.
- El equilibrio debe ser mantenido tanto de forma estática como dinámica en apoyo simple.
- La pierna oscilante debe ser capaz de avanzar hasta la posición en que se pueda convertir en la pierna de soporte.
- Debe poder proporcionarse la fuerza suficiente para realizar el movimiento de las extremidades y avanzar el tronco.

Hay que tomar en cuenta la orientación del cuerpo humano, en la cual se consideran tres planos de corte, estos planos se denominan planos anatómicos (Cabás, 2009, p. 81).

- **Plano sagital:** Pasa desde la parte anterior del cuerpo (o segmento de éste) hasta la posterior, dividiendo a éste en dos mitades, izquierda y derecha.
- **Plano frontal:** Pasa desde un extremo lateral del cuerpo (o segmento de éste) hasta el otro, dividiendo a este en dos mitades, anterior y posterior.
- **Plano transversal:** Pasa horizontalmente el cuerpo (o un segmento de éste), dividiéndolo en mitades superior e inferior.

a. CICLO DE LA MARCHA.

El caminado es un proceso cíclico, cada ciclo comienza cuando un pie tiene contacto con el suelo y termina en el siguiente contacto con el suelo del mismo pie, a la distancia entre estos dos pasos se le llama paso completo, el ciclo está dividido en dos fases:

- **Fase de Apoyo:** Cuando una pierna está en contacto con el suelo, también se utiliza el término apoyo doble que es cuando las dos piernas están en contacto con el suelo.
- **Fase de Balanceo:** Cuando una pierna no contacta con el suelo.

En la Figura 1.24 se observa que la cantidad de tiempo ocupada por la fase de apoyo es 60% del ciclo, la fase de doble apoyo es del 20% y la fase de balanceo es de 40% del ciclo (Hernández, 2008, p. 39).

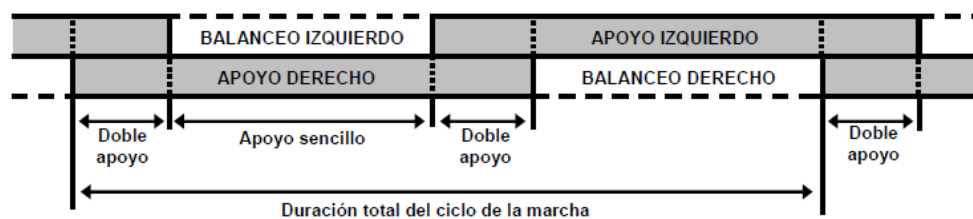


Figura 1.24. Componentes de la marcha.

Fuente: Francisco Hernández (2008). Diseño y construcción de prototipo neumático de prótesis de pierna humana. Capítulo 3. P.38.

1.2.5. LOCOMOCIÓN DE ROBOTS BÍPEDOS.

La locomoción bípeda en una persona consiste en una concatenación de movimientos controlados en respuesta a un sistema retroalimentado en tiempo real, en el caso de robots bípedos es algo similar.

a. LOCOMOCIÓN ESTÁTICA.

También se la conoce como caminata pasiva, esta caminata hace referencia a sistemas robóticos que caminan sin necesidad de control ni de actuadores, es así que su movimiento se genera únicamente por acción de la gravedad y por una pendiente en el piso. La principal ventaja de la caminata pasiva es que no se requiere de energía externa para caminar, pero la desventaja está en la direccionalidad limitada y la imposibilidad para subir pendientes (Santana, 2013, p. 3).

b. LOCOMOCIÓN DINÁMICA.

También se la conoce como caminata activa, la cual se relaciona directamente con robots que poseen un sistema de control complejo, utilizan actuadores y sensores, así el robot desarrolla tareas complejas como caminar en diferentes direcciones, subir escalones y permanecer en una pierna (Santana, 2013, p. 3).

Se utilizan criterios para lograr su estabilidad al momento de caminar, uno de los más utilizados es el punto de momento cero (ZMP), que se basa en mantener el centro de presión dentro del contacto del pie (Guzmán, 2010, p. 9).

1.2.6. CINEMÁTICA DE ROBOTS BÍPEDOS.

Antes de empezar con el estudio de la cinemática de un robot bípedo se necesita conocer de qué trata la cinemática de un robot, según (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 93): “La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia”.

Entonces, la cinemática de un robot bípedo describe la relación entre el movimiento de las articulaciones y el movimiento resultante de los cuerpos rígidos (eslabones) que lo constituyen; básicamente el análisis cinemático de un robot bípedo comprende el estudio de su movimiento con respecto a un sistema de referencia.

a. CINEMÁTICA DIRECTA.

En la cinemática directa, las posiciones de las articulaciones ya están determinadas y el problema radica en encontrar la configuración del efector final (Subir Kumar, 2010, p. 112). Por lo que se puede decir que la Cinemática directa consiste en que a partir de valores conocidos de las articulaciones y parámetros geométricos de los elementos del robot, se calcula la posición y orientación del efector final con respecto a un sistema de referencia.

Existen varios métodos para encontrar la cinemática directa de un robot, entre los más conocidos se tiene al uso de relaciones geométricas, el mismo que no es muy utilizado, por ser un método asistemático y sirve para robots de pocos grados de libertad, otra alternativa es la resolución de cinemática directa mediante cuaternios pero quizás el método más difundido sea aquel que utiliza matrices de transformación homogénea, por presentar grandes ventajas al momento de analizar un sistema complejo.

a.1 MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA.

Se define como matriz de transformación homogénea T a una matriz de dimensión 4×4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro (Borger, 2004). La matriz de transformación homogénea 4×4 relaciona dos sistemas de coordenadas, de forma que:

$$S_1 = T \cdot S_0 \quad \text{Ec.1.1}$$

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix} \quad \text{Ec.1.2}$$

En donde, S_1 representa el sistema de coordenadas móviles y S_0 es el sistema de coordenadas de referencia; en robótica se realiza una pequeña simplificación de modo que la matriz T se reduce a:

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ F_{1 \times 3} & W_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} \quad \text{Ec.1.3}$$

Que representa la posición y orientación de un sistema de coordenadas con respecto a un sistema de referencia.

Mediante una matriz de transformación homogénea se asocia a cada eslabón un sistema de referencia solidario, siendo posible representar rotaciones y traslaciones relativas entre los eslabones; una matriz de

transformación entre dos eslabones consecutivos se representa de la siguiente manera:

$${}^{i-1}A_i \quad \text{Ec.1.4}$$

Dónde i representa el último eslabón de la cadena.

Mediante estas matrices se puede representar ya sea total o parcialmente la cinemática de un robot, por ejemplo:

$$T = {}^0A_5 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \quad \text{Ec.1.5}$$

$${}^3A_5 = {}^3A_4 \cdot {}^4A_5 \quad \text{Ec.1.6}$$

Para la resolución del problema cinemático directo, se considera a la matriz de transformación como:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec.1.7}$$

Donde $[n_x n_y n_z]$, $[o_x o_y o_z]$ y $[a_x a_y a_z]$ son los vectores que conforman la matriz de rotación y $[p_x p_y p_z]$ es el vector de traslación.

Aplicando la ecuación 1.1 con S_0 en el origen (0,0,0), se obtiene $S_1 = [p_x p_y p_z]^T$ y se ha encontrado la posición del sistema coordenado S_1 con respecto al sistema S_0 . Para encontrar la orientación (ángulos de Euler):

$$\alpha = \arctan\left(\frac{a_y}{a_z}\right) \quad \text{Ec.1.8}$$

$$\beta = \arcsin(-a_x) \quad \text{Ec.1.9}$$

$$\gamma = \arccos\left(\frac{O_x}{n_x}\right)$$

Ec.1.10

Al contar con las ecuaciones Ec.1.8, Ec.1.9 y Ec.1.10, se puede decir que el problema cinemático directo ya sea de un robot bípedo u otro robot ha sido resuelto completamente.

a.2 REPRESENTACIÓN DE DENAVIT HARTENBERG (D-H).

Denavit y Hartenberg propusieron en 1955 un método matricial que permite establecer de manera sistemática un sistema de coordenadas S_i ligado a cada eslabón i de una cadena articulada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 96).

Según la representación D-H, escogiendo adecuadamente los sistemas de coordenadas asociados a cada eslabón, será posible pasar de uno al siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón tal como se muestran en la Figura 1.25.

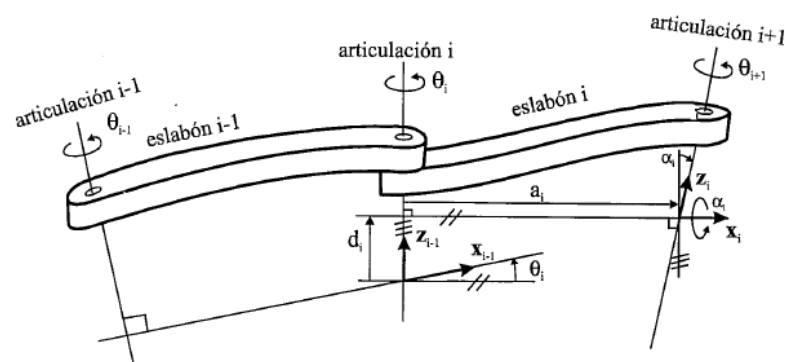


Figura 1.25. Parámetro D-H para un eslabón giratorio.

Fuente: Barrientos, Penin, Balaguer & Aracil (2007). Fundamentos de Robótica
p.98. Madrid: Mc Graw Hill

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i - 1$. Las transformaciones son las siguientes:

- Rotación alrededor del eje z_{i-1} un ángulo θ_i .
- Traslación a lo largo de z_{i-1} una distancia d_i .
- Traslación a lo largo de x_i una distancia a_i .
- Rotación alrededor del eje x_i , un ángulo α_i .

Dado que el producto de matrices no es conmutativo, las transformaciones se han de realizar en el orden indicado. De este modo se tiene que:

$${}^{i-1}A_i = T(z, \theta_i)T(0,0, d_i)T(a_i, 0,0)T(x, \alpha_i) \quad \text{Ec.1.11}$$

Realizando el producto de matrices:

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec.1.12}$$

a.3 ALGORITMOS D-H PARA MODELADO CINEMÁTICO DIRECTO.

Este procedimiento se lo realiza según (Barrientos, Penin, Balaguer, & Aracil, 2007, pp. 97 - 98).

D-H1. Numerar todos los eslabones comenzando con 1 (primer eslabón móvil de la cadena). Se enumerará como eslabón 0 a la base fija del robot.

D-H2. Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .

D-H3. Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

D-H4. Para i de 0 a $n - 1$ situar el eje z_i sobre el eje de la articulación $i + 1$.

D-H5. Situar el origen del sistema de la base S_0 en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0 .

D-H6. Para i de 1 a $n - 1$, situar el sistema S_i (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría S_i en el punto de corte. Si fuesen paralelos S_i se situaría en la articulación $i + 1$.

D-H7. Situar x_i en la línea normal común a z_{i-1} y z_i .

D-H8. Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .

D-H9. Situar el sistema S_n en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .

D-H10. Obtener θ_i como el ángulo que hay que girar que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.

D-H11. Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar S_{i-1} para que x_i y x_{i-1} quedasen alineados.

D-H12. Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo S_{i-1} para que su origen coincidiese con (S_i) .

D-H13. Obtener α_i como el ángulo que habría que girar en torno a x_i (que ahora coincidiría con x_{i-1}), para que el nuevo (S_{i-1}) coincidiese totalmente con (S_i) .

D-H14. Obtener las matrices de transformación ${}^{i-1}A_i$.

D-H15. Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1 \cdot {}^1A_2 \dots {}^{n-1}A_n$.

D-H16. La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base, en función de las n coordenadas articulares.

Una vez obtenidos los parámetros D-H, el cálculo de las relaciones entre eslabones consecutivos del robot es inmediato, ya que viene dado por las matrices A , que se calculan según la expresión general de la ecuación 1.12.

b. CINEMÁTICA INVERSA.

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot, $q = [q_1 q_2 \dots q_n]^T$ para que su extremo se posicione y oriente según una determinada localización espacial (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 108).

En otras palabras se dice que consiste en encontrar el estado de cada una de las articulaciones que conforman un robot, para lograr una posición y orientación del efector final, de manera que:

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma)$$

Ec.1.13

Este tipo de solución presenta las siguientes ventajas:

- En muchas aplicaciones, el problema inverso ha de resolverse en tiempo real (por ejemplo, en el seguimiento de una determinada trayectoria); una solución de tipo iterativo no garantiza tener la solución en el momento adecuado.

- Al contrario de lo que ocurrirá en el problema cinemático directo, con cierta frecuencia la solución del problema cinemático inverso no es única; existiendo diferentes n-uplas, $[q_1 q_2 \dots q_n]^T$ que posicionan y orientan el robot del mismo modo; en este caso una solución cerrada permite incluir determinadas reglas o restricciones que aseguren que la solución obtenida sea la más adecuada de entre las posibles (por ejemplo límites en los recorrido articulares).

De igual manera que en el problema cinemático directo, existen varios métodos de resolución, entre los cuales se destacan:

- **Método geométrico:** Este procedimiento es adecuado para robots de pocos grados de libertad o para el caso de que se consideren solo los primeros grados de libertad dedicados a posicionar el extremo.
- **Matriz de transformación homogénea:** Se trata de obtener el modelo cinemático inverso de un robot a partir del conocimiento de su modelo directo, es decir, suponiendo conocidas las relaciones que expresan el valor de la posición y orientación del extremo del robot en función de sus coordenadas articulares.
- **Desacoplo cinemático:** Aplicable a robots de 6 grados de libertad, en donde la resolución total es dividida en dos partes (posición y orientación), dedicando los tres primeros grados de libertad para posicionar el efector final y los tres restantes para orientar el mismo.

1.2.7. DINÁMICA DE ROBOTS BÍPEDOS.

Otro punto esencial para el desarrollo del robot bípedo es el estudio de las fuerzas que actúan sobre él, por lo cual, se hace necesario conocer la teoría de la dinámica en robots, la misma que se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina, por lo tanto, el modelo dinámico tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas que involucran el mismo (Barrientos, Penin, Balaguer, & Aracil, 2007, p. 131).

Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

- La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- Las fuerzas y pares aplicados en las articulaciones o en el extremo del robot.
- Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

El problema de la obtención del modelo dinámico de un robot es uno de los aspectos más complejos de la robótica, lo que ha llevado a ser obviado en numerosas ocasiones, por lo cual el modelo dinámico debe ser resuelto de manera iterativa mediante la utilización de un proceso numérico.

Es preciso señalar que si bien en la mayor parte de las aplicaciones reales de la robótica, las cargas e inercias manejadas no son suficiente como para originar deformaciones en los eslabones del robot, en determinadas ocasiones no ocurre así, siendo preciso considerar al robot como un conjunto de eslabones no rígidos.

Dentro de la dinámica referente a los robots bípedos se hace necesario el conocer algunos conceptos los mismos que se detallan a continuación.

- **COM.** Es el centro de masa o punto de masa concentrada de un robot.
- **Polígono de soporte.** Figura geométrica que se forma en el suelo por uno o por los dos pies del robot, de acuerdo a la fase en que se encuentre de la caminata; en el instante en que el robot se encuentra en único soporte, el polígono de soporte es el pie de apoyo, en cambio, si el bípedo se encuentra en doble soporte el polígono es la

figura que se forma por los dos pies, tal como se muestra en la Figura 1.26.

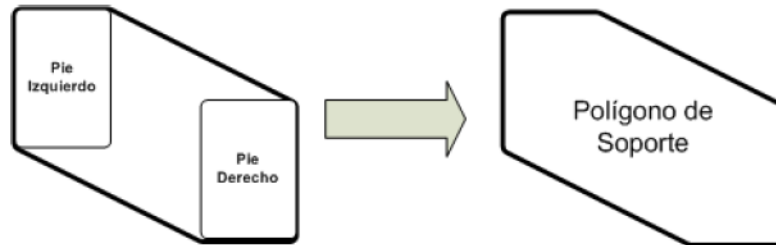


Figura 1.26. Polígono de soporte.

Fuente: Santana José Luis (2008). Control de Equilibrio en el Proceso de Caminado de un Robot Bípedo p.9. Guadalajara: Universidad de Guadalajara

- **ZMP.** Es el punto de momentos cero, es el punto dentro de la superficie de contacto en el cual la reacción neta del piso actúa en el momento en que el robot está dinámicamente estable.
- **FIR.** Es el índice de rotación del pie, es el punto en la superficie en el cual la reacción del piso tiene que actuar para que el pie permanezca estacionario.
- **COP.** Es el centro de presión o fuerza, es un punto en la superficie de contacto en donde se considera que actúa la fuerza neta de reacción del piso.
- **SM.** Es el margen de estabilidad, es la mínima distancia entre el FRI y los bordes del polígono de soporte, en donde SM_x es el margen de estabilidad en el eje X, y SM_y es el margen de estabilidad en el eje Z, tal como se observa en la Figura 1.27.

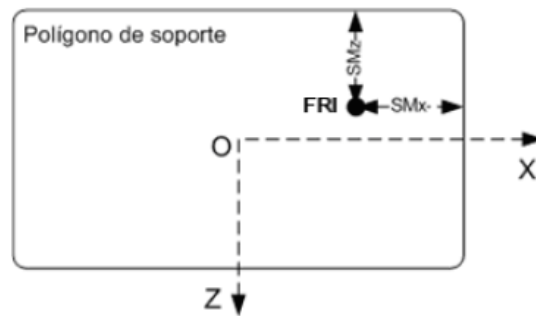


Figura 1.27. Margen de estabilidad.

Fuente: Santana José Luis (2008). Control de Equilibrio en el Proceso de Caminado de un Robot Bípedo p.10. Guadalajara: Universidad de Guadalajara

1.2.8. MATERIALES TÍPICAMENTE USADOS.

Debido a la diversidad de materiales existentes, nace la necesidad de elegir un material que cumpla con los requerimientos necesarios para que el proyecto se realice de una manera adecuada, los siguientes materiales cumplen con las características necesarias para que puedan ser tomados en cuenta para el desarrollo del robot bípedo.

- MDF.
- ABS.
- Aluminio.
- Fibra de vidrio.
- Fibra de carbono.

a. MDF.

Por sus siglas en inglés de Medium Density Fibreboard, es un tablero de fibras de madera de pino radiata unidas por adhesivos, las fibras de madera son obtenidas mediante un proceso termo-mecánico y unidas con adhesivo que polimeriza mediante altas presiones y temperaturas (MASISA, 2014).

Características:

- Facilidad en el acabado, que permite excelentes terminaciones, con un importante ahorro de pintura y un menor desgaste de herramientas.
- Consta de superficies 100% lisas con gran homogeneidad, lo que permite una mínima preparación de las superficies.
- Tiene resistencia al fuego, según la Norma: Nch 935/1 Of. 97 Certificado IDIEM N° 238.339 Resultado: Clasificado F-15.

En la Tabla 1.2 se detallan las propiedades mecánicas del MDF.

Tabla.1.2. Propiedades físico – mecánicas del MDF.

Propiedades	Unidad	Tolerancia	Espesor 3 [mm]
Densidad	[kg/m ³]	± 50	820
Flexión	[N/mm ²]	± 4.9	49
Tracción	[N/mm ²]	± 0.1	1.4
Módulo de Elasticidad	[N/mm ²]	± 290	4440
Límite Elástico	[N/mm ²]	± 50	50
Hinchamiento 24 horas	[%]	-	35

Fuente: MASISA Recuperado de: <http://www.masisa.com/per/productos/tableros/mdf/>

b. ABS.

El ABS es el nombre dado a una familia de termoplásticos, se le llama plástico de ingeniería, debido a que es un plástico cuya elaboración y procesamiento es más complejo que los plásticos comunes, como son las polioleofinas (polietileno, polipropileno). En la actualidad el ABS se produce por medio de la polimerización del estireno y el acrilonitrilo en presencia de polibutadieno, quedando como producto una estructura de polibutadieno, conteniendo cadenas de SAN (estireno acrilonitrilo) injertados en él (TecnologíaPlásticos, 2014).

Características:

- Los materiales de ABS tienen importantes propiedades de ingeniería, como buena resistencia mecánica y al impacto combinado con facilidad para el procesado.
- La resistencia al impacto de los plásticos ABS se ve incrementada al aumentar el porcentaje de contenido en butadieno pero disminuyen las propiedades de resistencia a la tensión y disminuye la temperatura de deformación por calor.
- Pueden ser extruidos, moldeados por inyección, soplado y prensado; generalmente los grados de bajo impacto son los que más fácil se procesan, al contrario de los de alto impacto los cuales son más dificultosos porque al tener un mayor contenido en caucho los hace más viscosos.
- A pesar de que no son altamente inflamables, mantienen la combustión; hay algunos tipos auto-extinguibles para cuando se requiere algún producto incombustible, otra solución consiste en aplicar algún retardante de llama.

En la Tabla 1.3 se detallan las propiedades físico - mecánicas del ABS.

Tabla.1.3. Propiedades físico – mecánicas del ABS.

ABS – POLIACRILONITRILO BUTADIENO ESTIRENO	
Alargamiento a la Rotura (%)	45
Coefficiente de Fricción	0,5
Módulo de Tracción [GPa]	2,1 - 2,4
Resistencia a la Tracción – Flexión [MPa]	41 – 45
Resistencia al Impacto Izod [$J m^{-1}$]	200 – 400
Absorción de Agua – en 24 horas (%)	0,3 – 0,7
Densidad [$g cm^{-3}$]	1,05
Limite elástico [MPa]	38.8
Resistencia a los Ultra-violetas	Pobre

Fuente: SIIM CO. SL. Recuperado de: <http://www.siim.com/docs/RAE-1018.pdf/>

c. ALUMINIO.

El aluminio puro es blando y tiene poca resistencia mecánica, pero puede formar aleaciones con otros elementos para aumentar su resistencia y adquirir varias propiedades útiles; las aleaciones de aluminio son ligeras, fuertes, y de fácil formación para muchos procesos de metalistería; son fáciles de ensamblar, fundir o maquinar y aceptan gran variedad de acabados. Por sus propiedades físicas, químicas y metalúrgicas, el aluminio se ha convertido en el metal no ferroso de mayor uso (Lenntech, 2014). En la Tabla 1.4 se detallan las propiedades físico-mecánicas.

Tabla.1.4. Propiedades físico – mecánicas del aluminio.

ALUMINIO	
Esfuerzo [N/mm²]	250
Módulo de Young [MPa] (elasticidad)	70000
Módulo de rigidez cortante [Kg/cm²]	280000
Módulo de Poisson	0.33
Densidad [gr/cm³]	2.7
Coefficiente de expansión lineal x10⁻⁶/C	24

Fuente: Automoción. SL. Recuperado de:<http://puentelara.blogspot.com/2013/02/aluminio-uso-en-el-automovil.html>

1.3. CONTROLADORES AVANZADOS PARA ROBOTS.

Debido a la investigación realizada se denota que dentro de los sistemas de control aplicados para el desarrollo de robots bípedos los que más destacan son los sistemas de control adaptativo, difuso, robusto y heurístico, pero cabe recalcar que es necesario conocer además parte de la teoría referente a los sistemas de control.

1.3.1. SISTEMAS DE CONTROL.

Un sistema de control integra diversos procesos con la finalidad de cumplir un conjunto de especificaciones en las salidas de los mismos, el control es un campo del conocimiento de gran utilidad en la ingeniería y la

ciencia, ha sido piedra angular en el desarrollo de la tecnología, ya que posibilita obtener un óptimo desempeño de sistemas autónomos.

Un sistema de control produce una salida o respuesta para una entrada o un estímulo dado, como se muestra en la Figura 1.28.

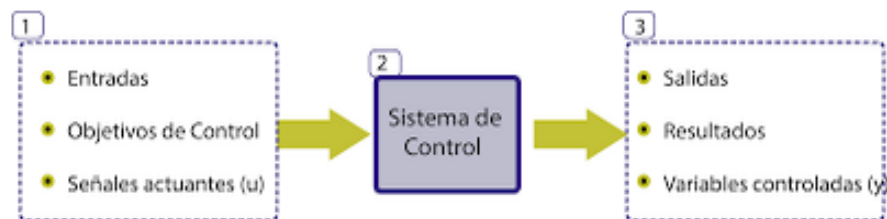


Figura 1.28. Sistemas de control.

Fuente: <http://carolina-sp.blogspot.com/>

La entrada representa una respuesta deseada y la salida representa una respuesta real. Existen dos factores que hacen que la salida sea diferente a la entrada:

- El primero se conoce como respuesta transitoria, el cual está presente cuando existe un cambio instantáneo de la entrada con respecto al cambio gradual de la salida; después de la respuesta transitoria, un sistema físico aproxima su respuesta en estado estable, en donde trata de aproximarse a la respuesta deseada.
- El segundo factor es la precisión, que puede ocasionar que la salida sea diferente a la entrada; esta diferencia es conocida como error en estado estable y en ocasiones depende de las características físicas del dispositivo que se desea controlar.

1.3.2. CONFIGURACIONES DE SISTEMAS DE CONTROL.

Las configuraciones de los sistemas de control son:

a. SISTEMAS DE CONTROL EN LAZO ABIERTO.

Son los sistemas en los cuales la salida no tiene efecto sobre la acción de control, en otras palabras, en un sistemas de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada (Ogata, 2003, p. 7). En la Figura 1.29 se muestra el diagrama de un sistema de control en lazo abierto.

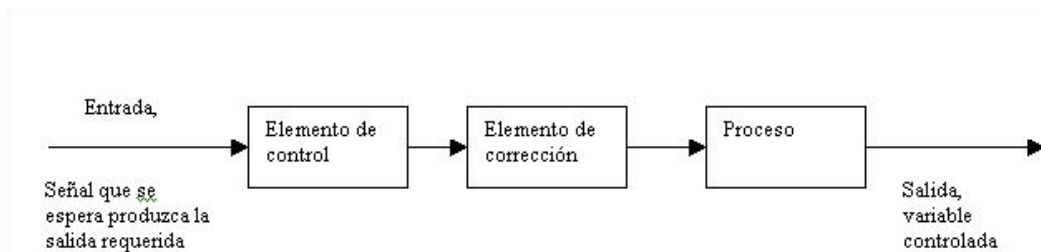


Figura 1.29. Sistema de control en lazo abierto.

Fuente: http://usuarios.multimania.es/automatica/temas/tema2/pags/la_lc/lalc.htm

Debido a que en los sistemas de control en lazo abierto la salida no se compara con la entrada de referencia, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración; ante la presencia de perturbaciones estos sistemas no realizan la tarea deseada.

b. SISTEMAS DE CONTROL EN LAZO CERRADO.

Estos sistemas miden la señal de la respuesta de salida y la comparan con la señal de la respuesta de entrada mediante una trayectoria de retroalimentación, la cual permite estimar las perturbaciones exteriores para que estas puedan ser corregidas (Ramírez Ramos, 2008, pp. 12-13).

El modelo matemático de estos procesos puede estar en el dominio de la frecuencia o del tiempo, los modelos en el dominio de la frecuencia requieren de una función de transferencia y se utilizan principalmente en sistemas lineales que no varían en el tiempo, haciendo de ésta su principal desventaja, los modelos en el dominio del tiempo permiten modelar procesos

más complejos, con múltiples entradas y salidas. La Figura 1.30 muestra el diagrama de un control en lazo cerrado.

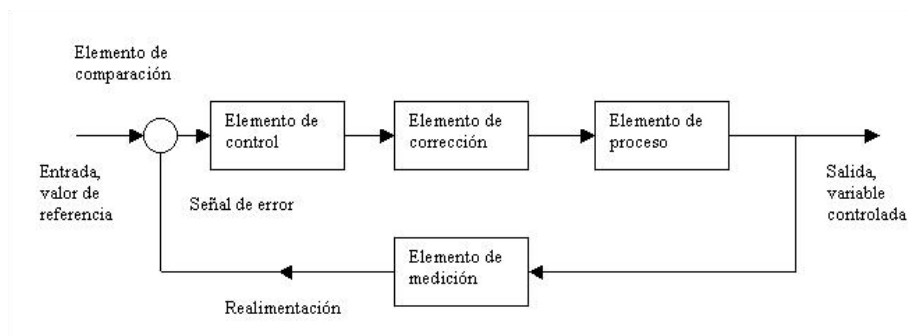


Figura 1.30. Sistema de control en lazo cerrado.

Fuente: http://usuarios.multimania.es/automatica/temas/tema2/pags/la_lc/lalc.htm

1.3.3. TIPOS DE SISTEMAS DE CONTROL.

Existen diferentes tipos de sistemas de control, entre los cuales se tienen:

a. SISTEMAS DE CONTROL CONVENCIONAL.

Estos tipos de controladores mejoran el error en estado estable y la respuesta transitoria, el modelo de cualquiera que sea el dominio que se elija requiere de modelos matemáticos que pueden ser demasiado complejos; estos sistemas de control realizan de manera eficiente la función de mantener estable un sistema, pero son vulnerables a cambios e incertidumbres que no se tomaron en cuenta cuando se diseñaron.

En estos tipos de sistemas de control se encuentran los siguientes:

- **Acción Encendido - Apagado:** En este control el elemento de actuación solo tiene dos posiciones fijas ON u OFF, este control es relativamente barato y simple y por eso se ocupa en muchas aplicaciones del hogar y la industria (Núñez, 2007, p. 9).
- **Acción Proporcional:** En este control el tiempo de subida se reduce, lo que permite incrementar el sobretiro y reducir el error en estado

estable; se lo conoce como un amplificador con ganancia ajustable (Núñez, 2007, p. 11).

- **Acción Integral:** En este control se disminuye y elimina el error de estado estacionario, cabe recalcar que este control añade un polo en el origen, con lo cual el sistema se vuelve menos estable; se lo conoce como control de reajuste o reset (EcuRed, 2014).
- **Acción Proporcional - Integral (PI):** En este control se decrementa el tiempo de subida, incrementando el sobre impulso y además el tiempo de estabilización, este control elimina el error de estado estable pero empeorará la respuesta transitoria (Núñez, 2007, p. 13).
- **Acción Proporcional - Derivativa (PD):** En este control se reduce el tiempo de estabilización y el sobre impulso, tiene el efecto de estabilizar el sistema mejorando la respuesta del mismo, tiene la ventaja de ser de previsión, pero su desventaja es que amplifica el ruido y suele provocar un efecto de saturación en el actuador; se lo conoce como control de velocidad (Núñez, 2007, pp. 14-15).
- **Acción Proporcional – Integral - Derivativa (PID):** En este control se tiene las ventajas de cada una de las acciones de control proporcional, integral y derivativo, es el más completo y que mejores resultados da dentro de los sistemas de control convencionales.

b. SISTEMAS DE CONTROL AVANZADOS.

En estos sistemas de control se puede mencionar que se encuentran aquellos que son utilizados para ser desarrollados en campos como la robótica debido a su respuesta ante perturbaciones que no son consideradas en su diseño, dentro de estos sistemas de control se tiene:

- **Control Adaptivo:** Es un tipo especial de control no lineal en el que el estado del proceso puede ser separado en dos escalas de tiempo que evolucionan a diferente velocidad, la escala lenta corresponde a los

cambios de los parámetros y por consiguiente a la velocidad con la cual los parámetros del regulador son modificados, y la escala rápida que corresponde a la dinámica de bucle ordinario de realimentación (Rodríguez & López, 1996, pp. 9-10). Un control adaptivo se puede utilizar en procesos industriales muy complejos donde la variación de los parámetros de un sistema no se puede determinar desde el inicio (Ramírez Ramos, 2008, p. 15).

- **Control Robusto:** Se considera que cualquier modelo matemático de un proceso va a ser en mayor o en menor grado impreciso o va a tener incertidumbres o errores de modelado, si se desea controlar de manera eficiente un proceso es necesario contar con la información de las posibles fuentes de incertidumbre y evaluar su efecto sobre el comportamiento del sistema (Ramírez Ramos, 2008, p. 15).
- **Control Difuso:** Un sistema de control difuso es una alternativa muy útil a los sistemas de control que requieren de un modelo matemático demasiado complejo; este tipo de sistema de control utiliza el conocimiento experto para generar una base de conocimientos, la cual le otorga al sistema la capacidad de tomar decisiones, la base del funcionamiento de este tipo de sistema de control es la lógica difusa.

1.3.4. TÉCNICA HEURÍSTICA.

Se puede definir la heurística como una técnica o procedimiento práctico o informal, para resolver problemas, también se la puede definir como un conjunto de reglas metodológicas no necesariamente formalizadas positivas y negativas, que sugieren o establecen cómo proceder y evitar problemas a la hora de generar soluciones y elaborar hipótesis. Según el matemático George Pólya la base de la heurística está en la experiencia de resolver problemas y en ver cómo otros lo hacen (Reyes, 2013).

a. FUNCIÓN HEURÍSTICA.

Se conoce como función heurística a aquella función numérica que permite cuantificar el beneficio de un cambio en el espacio de estados del problema a resolver, resultan útiles a la hora de optimizar los procesos de búsqueda, ya que intentan guiar la exploración del espacio de estados en la dirección más provechosa, sugiriendo el mejor camino a seguir cuando disponemos de varias alternativas, intentando llegar a la solución de manera intuitiva (Malkav, 2012).

La función heurística puede tener dos interpretaciones, por una parte la función puede ser una estimación de lo próximo que se encuentra el estado de un estado objetivo, bajo esta perspectiva los estados de menor valor heurístico son los preferidos; pero en otros casos puede suceder que lo que convenga sea maximizar esa función.

Dentro del campo de la robótica la mayoría de los robots y humanoides tienden a tener trayectorias o constar con bases de datos predeterminadas para realizar distintas acciones ya sean sencillas o complejas, por lo cual, en este campo la heurística permite encontrar la solución a dichos problemas de una forma eficiente y muy rápida, minimizando el número de nodos que son expandidos por el algoritmo de búsqueda hasta su finalización y con ello el tiempo de computación necesario.

A continuación se detallan las técnicas que involucran los procedimientos de búsqueda en el espacio de soluciones factibles:

- **Búsqueda Local:** Los algoritmos de búsqueda local parten de una solución inicial y aplicándole operadores de variación, la van alterando; si la solución alterada es mejor que la original, se acepta, si no lo es, se vuelve a la inicial. El procedimiento se repite hasta que no se consigue mejora en la solución; los procedimientos de búsqueda local más usados se basan en el gradiente (Reyes, 2013).

- **Hillclimbing:** Estos algoritmos se llaman de escalada (o de ascensión a la colina) porque tratan de elegir en cada paso un estado cuyo valor heurístico sea mayor que el del estado activo en ese momento, debido a que se tratan de algoritmos de búsqueda locales sólo van a encontrar el máximo local más cercano al punto de inicio. Estos algoritmos se dividen en dos grupos:
 - Los métodos irrevocables en este método, en el momento en que se encuentra un estado E que es más favorable que el que se está expandiendo, dicho estado E es devuelto sin generar el resto de estados hijos.
 - Los métodos tentativos en este caso, se generan todos los hijos de un estado, se calculan sus valores heurísticos y se determina uno de los estados de mejor valor heurístico; se compara dicho valor con el de él estado expandido y si es mejor, se devuelve ese estado como expansión.

En ambos casos, si ningún estado hijo es mejor que el estado que está siendo expandido, no se devuelve nada, lo que conllevará que la búsqueda termine sin haber encontrado un objetivo (Malagón, 2012).

- **Recocido Simulado:** Se basa en el proceso de recocido, en la cual los metales se funden y se forjan a la par que se enfrían, hasta conseguir una estructura cristalina, dura y resistente. Consiste en un ascenso de gradiente comenzando en un punto aleatorio; dependiendo de la temperatura es como se mueve por el espacio de búsqueda. La probabilidad de escoger una solución peor que la actual, descende con el tiempo; la solución (se aproxima a la solución óptima) se escoge de manera determinista evitando los mínimos locales (Reyes, 2013).
- **Iteración Básica:** En cada iteración, el método de recocido simulado evalúa algunos vecinos del estado actual s y probabilísticamente

decide entre efectuar una transición a un nuevo estado s' o quedarse en el estado s . Típicamente la comparación entre estados vecinos se repite hasta que se encuentre un estado óptimo que minimice la energía del sistema o hasta que se cumpla cierto tiempo computacional u otras condiciones (Reyes, 2013).

- **Vecindario de un Estado:** Está compuesto por todos los estados a los que se pueda llegar a partir de s mediante un cambio en la conformación del sistema. El método de evaluación de estados vecinos es fundamental para encontrar una solución óptima global al problema dado. Los algoritmos heurísticos, basados en buscar siempre un estado vecino mejor (con energía más baja) que el actual se detienen en el momento que encuentran un mínimo local de energía (Reyes, 2013).
- **Búsqueda Tabú:** Esta técnica en realidad es una meta-heurística, que guiará a otras técnicas para que no se quede en óptimos locales, explora el espacio de búsqueda de todas las soluciones por medio de movimientos que se harán sólo si la solución siguiente es mejor que la anterior. La parte medular de la búsqueda tabú se localiza en el proceso de memoria de corto plazo, y muchas de las consideraciones estratégicas en que se fundamenta este proceso reaparecen, amplificadas pero sin mayores modificaciones, en los procesos de memoria de largo plazo (Reyes, 2013).
- **Algoritmos Bio – Inspirados:** Basan su operación en los principios de la teoría Neo - Darwiniana de la evolución natural, los problemas combinatoriales pueden ser divididos en dos grandes grupos considerando la existencia de algoritmos polinomiales para resolver cada tipo de problema.

El primero es el problema tipo P (polinomial) para el cual existen algoritmos con esfuerzos computacionales de tipo polinomial para encontrar la solución óptima; y el segundo es el problema tipo NP (no

polinomial) para el cual no se conocen algoritmos con esfuerzos computacionales de tipo polinomial para encontrar la solución óptima; las técnicas heurísticas son algoritmos que encuentran soluciones de buena calidad para problemas combinatoriales complejos o sea para problemas tipo *NP* (Reyes, 2013).

1.4. COMUNICACIÓN INALÁMBRICA.

Teniendo en cuenta que la comunicación es de suma importancia en el envío y recepción de información, se denota que para el desarrollo del presente proyecto la comunicación inalámbrica es muy recursiva en todos sus aspectos para ser utilizada, debido a lo mencionado, en adelante se hace referencia a toda la teoría que es necesaria conocer para poder seleccionar la tecnología adecuada de comunicación inalámbrica de igual manera se detallan módulos específicamente usados en aplicaciones robóticas.

1.4.1. DEFINICIÓN.

La comunicación inalámbrica o sin cables es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales se encuentran: antenas, computadoras portátiles, PDA, teléfonos móviles, etc. (Wikipedia, Wikipedia, 2013).

1.4.2. TECNOLOGÍA ZigBee.

ZigBee (Oyarce, 2009) es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE 802.15.4, permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas, es útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.

Las comunicaciones ZigBee se pueden realizar en tres bandas 868 [MHz] para Europa, 915 [MHz] para Norte América y 2.4 [GHz] para todo el mundo, a diferencia de bluetooth, este protocolo realiza las comunicaciones a través de una única frecuencia, es decir, de un canal, normalmente puede escogerse un canal de entre 16 posibles para la banda libre de 2.4 [GHz], 10 canales para la banda de 915 [MHz] y 1 solo canal para 868 [MHz], el alcance depende de la potencia y antena del dispositivo, la velocidad de transmisión de datos de una red ZigBee es de hasta 256 [kbps]. Una red ZigBee la pueden formar, en teoría, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en una red esta cantidad de dispositivos (Huaca, 2009).

Una red ZigBee la forman básicamente 3 tipos de elementos: un único dispositivo Coordinador, dispositivos Routers y dispositivos Finales.

- **Módulos XBee.**

Una buena forma de agregar conectividad inalámbrica es utilizando los módulos XBee, estos módulos proveen 2 formas amigables de comunicación, transmisión serial transparente o modo AT y el modo API (interfaz de aplicación para la programación), que se configura mediante el parámetro AP, existen dos modos de trabajo, uno "normal" (AP = 1) y otro "con escape" (AP = 2).

Los módulos XBee pueden ser configurados desde el PC utilizando el programa X-CTU o bien desde un microcontrolador, estos dispositivos pueden comunicarse en arquitecturas punto a punto, punto a multi punto o en una red de malla. Para la elección del módulo XBee correcto primero se escoge el tipo de antena (chip, alambre o conector SMA que es un tipo de conector roscado para cable coaxial utilizado en microondas) y lo segundo a tener en cuenta es la potencia de transmisión (2 [mW] para 300 [ft] o 60 [mW] para hasta 1 [mi]).

Existen 2 series de módulos. La serie 1 y la serie 2 o también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out (pin de salida), sin embargo, no son compatibles entre sí ya que utilizan distintos chipset.

A continuación se muestra la Tabla 1.5 con las características del XBee de la Serie 1.

Tabla 1.5. Características del XBee Serie 1.

Característica	XBee Serie 1
Rango interior/urbano	mayor a 30 [m]
Rango línea de seguimiento de radio frecuencia en exterior	mayor a 100 [m]
Poder de transmisión de salida	1 [mW]
Velocidad de datos de radio frecuencia	250 [Kbps]
Sensibilidad del receptor	-92 [dbm]
Voltaje de alimentación	2.8 – 3.4 [V]
Corriente de transmisión (típica)	45 [mA]
Corriente de recepción (típica)	50 [mA]
Corriente de Standby	10 [µA]
Frecuencia	ISM 2.4 [GHz]
Dimensiones	0.0960 x 1
Temperatura de operación	-40 a 85 [°C]
Opciones de antena	chip, antena integrada
Topología de red	punto a punto, estrella
Número de canales	16 canales de secuencia directa
Opciones de filtración	canal & origen/destino

Fuente: XBEE.CL Recuperado de: <http://www.xbee.cl/diferencias.html>

1.5. SERVOMOTORES.

Debido a las prestaciones que presentan este tipo de actuadores eléctricos son los más utilizados en robótica, por lo cual es necesario conocer su teoría y constitución.

1.5.1. DEFINICIÓN.

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición; está conformado por un motor, una caja reductora y un circuito de control, los servos se utilizan frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a estos.

Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos (Ramirez, 2010).

La corriente que requiere depende del tamaño del servo, normalmente el fabricante indica cual es la corriente que consume, eso no significa mucho si todos los servomotores van a estar moviéndose todo el tiempo, la corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado.

Existen tres tipos de servomotores como son los de corriente alterna (CA), corriente continua (CC) y los de imanes permanentes o Brushless.

1.5.2. PARTES DE UN SERVOMOTOR.

Un servomotor está conformado por las siguientes partes:

- a. Cubierta superior.
- b. Flecha o eje.
- c. Engranajes Reductores.
- d. Cubierta.
- e. Motor DC.

f. Cubierta inferior.

g. Circuito de Control.

Todas estas partes integran el servomotor tal como se muestra en la Figura 1.31.

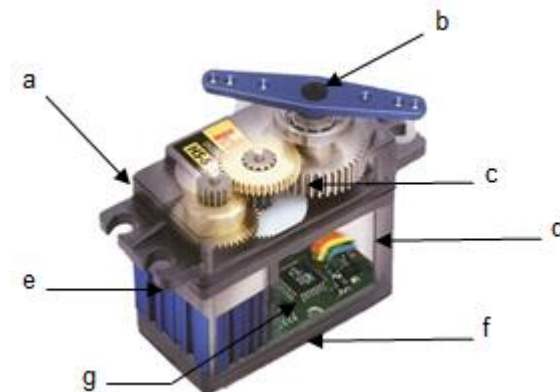


Figura 1.31. Estructura interna de un servomotor.

Fuente:

http://www.klugers.net/web/contenido/proyectos/Control_de_un_Servomotor/images/interior_servo.jpg

Motor DC.

Es el elemento que le brinda movilidad al servo, cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima, si el voltaje aplicado a sus dos terminales es inverso, el sentido de giro también se invierte.

Engranajes reductores.

Se encargan de convertir gran parte de la velocidad de giro del motor de corriente continua en torque.

Circuito de control.

Este circuito es el encargado del control de la posición del motor, recibe los pulsos de entrada y ubica al motor en su nueva posición dependiendo de los pulsos recibidos, los servomotores tienen 3 terminales:

- **Terminal positivo:** Recibe la alimentación del motor (4 a 12 [V]).
- **Terminal negativo:** Referencia tierra del motor (0 [V]).
- **Entrada de señal:** Recibe la señal de control del motor.

Los colores de los cables de cada terminal varían con cada fabricante, el cable del terminal positivo siempre es rojo, el del terminal negativo puede ser marrón o negro y el del terminal de entrada de señal suele ser de color blanco, naranja o amarillo (Ramirez, 2010).

1.6. SOFTWARE.

El desarrollo de las diferentes partes en el diseño de un robot bípedo provoca la necesidad de acudir a diferentes tipos de programas que permitan tanto la creación de la estructura, comprobación de los cálculos, como facilitar el desarrollo de la simulación o la creación de un programa para el control; por lo cual, se detallan a continuación los distintos programas que se utilizarán en este proyecto.

1.6.1. SolidWorks.

Es una completa solución de diseño en 3D que incorpora una poderosa herramienta de simulación y validación de diseño, ya sea de componentes mecánicos como de prototipos robóticos, estructurales, etc. Consta de herramientas específicas para la realización de estudios en base a los diseños creados que facilitan su entrega, así como la colaboración de CAD/CAM.

1.6.2. Matlab.

Matlab es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, visualización y programación, usando Matlab, se puede analizar los datos, desarrollar algoritmos y crear modelos y aplicaciones. El lenguaje, las herramientas y funciones matemáticas integradas que permiten explorar múltiples enfoques y llegar a una solución más rápida que con hojas

de cálculo o lenguajes de programación tradicionales, como C/C++ o Java (Mathworks, Matlab the language of technical computing, 2014).

Se puede utilizar Matlab para una gama de aplicaciones, incluyendo el procesamiento de señales, comunicaciones, procesamiento de imágenes y vídeo, sistemas de control, prueba y medida, finanzas computacionales, y la biología computacional.

Las herramientas de Matlab aplicadas en robótica son:

a. SimMechanics.

SimMechanics proporciona un entorno de simulación multi-cuerpo para sistemas mecánicos 3D, tales como robots, suspensiones de vehículos, equipos de construcción y equipo de aterrizaje de los aviones. Aquí se puede modelar el sistema multi-cuerpo mediante bloques que representan elementos de los cuerpos, las articulaciones, limitaciones, y de la fuerza, y luego SimMechanics formula y resuelve las ecuaciones de movimiento para el sistema mecánico completo. Una animación 3D generada automáticamente le permite visualizar la dinámica del sistema (Mathworks, SimMechanics Model and simulate multibody mechanical systems, 2014).

b. Robotics Toolbox.

El Robotics Toolbox (RTB) ofrece una amplia gama de funciones para la simulación de robots móviles y de brazos robóticos, esta toolbox data de principios de los años 1990, además presentaba un método muy general de representar la estructura de la serie de vínculos manipuladores utilizando matrices y posteriores objetos de Matlab. Arbitrarias series de enlaces manipuladores pueden ser creados dentro de él, otra prestación del toolbox es la de proporcionar funciones de avance, cinemática inversa y dinámica. El toolbox incluye funciones para la manipulación y conversión entre tipos de datos, tales como vectores, transformaciones homogéneas, cuaternios, que son necesarios para representar la posición y la orientación en 3 dimensiones (Corke, 2011, p. 7).

1.6.3 Proteus

Este software es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics, cuyos módulos principales son los siguientes: (Wikipedia, Wikipedia, 2014)

- **ISIS:** permite diseñar el plano eléctrico del circuito que se desea realizar con componentes muy variados, desde simples resistencias, hasta alguno que otro microprocesador o microcontrolador, incluyendo fuentes de alimentación, generadores de señales y muchos otros componentes con prestaciones diferentes.
- **ARES:** Es la herramienta de enrutado, ubicación y edición de componentes, se utiliza para la fabricación de placas de circuito impreso.

1.6.4 CURA.

Es un software que convierte archivos STL que contienen los diseño 3D, en piezas físicas en un solo entorno de trabajo por medio de impresoras 3D. Es una opción más como programa de laminado basado profundamente en Skeinforge pero con opciones de configuración más simples y una interfaz gráfica algo más intuitiva y sencilla. Tiene la capacidad de generar laminados mucho más rápido o generar varios laminados a la vez en paralelo (Makealo, 2014).

1.6.5 PyPose.

Es un simple código abierto que funciona como un programa para el almacenamiento de posición de los servomotores AX-12A de la marca Dynamixel, este software está escrito en Python y utiliza wxPython como interfaz gráfica. Fue creado para que los expertos en robótica puedan crear una serie de posiciones y luego ensamblarlas en una secuencia de movimiento (Hosting, 2012).

1.6.6 IDE de Arduino.

El entorno de desarrollo integrado de Arduino es un conjunto de herramientas software que permite escribir y probar programas con comodidad, en el caso de Arduino estos programas son llamados “sketch”, permite comprobar que no hayan errores y que además permite grabarlo en la memoria del microcontrolador de la placa Arduino, tarjetas derivadas y tarjetas compatibles, para que este controlador se convierta a partir de entonces en el ejecutor autónomo de dicho programa (Torrente, 2013, p. 129). Debido a que es un software de desarrollo libre éste realiza de una manera fácil el cambio de lenguaje de diferentes instrucciones realizadas en las líneas de código para que sean entendidas por el microcontrolador (Banzi, 2008).

CAPÍTULO 2

DISEÑO Y SELECCIÓN DE DISPOSITIVOS

2.1. PARÁMETROS DE DISEÑO.

Los parámetros de diseño considerados para el desarrollo del proyecto son: parámetros mecánicos, comunicación, electrónicos y de funcionamiento.

2.1.1. PARÁMETROS MECÁNICOS.

Dentro de los parámetros mecánicos se considera el tamaño, el peso y los grados de libertad del robot.

- Para asignar el tamaño del robot bípedo se ha realizado una investigación respecto a robots similares en proyectos de tesis y en artículos como los siguientes (Ortega & Contreras, 2008, p. 10), (Ramírez, 2009, pp. 21-24) y (Albero, Blanes, Benet, Simó, & Perez, 2013, p. 2), teniendo en cuenta esta investigación se establece que el tamaño del robot estará comprendido entre 25 y 50 [cm] de altura.
- De igual manera mediante la lectura comprensiva de tesis como las de (Chavolla, 2010, pp. 13-18) y (Ramírez, 2009, pp. 21-24) en donde se realiza un estado del arte de robots humanoides y teniendo en cuenta que el prototipo de robot bípedo será de uso práctico, el peso total del robot quedará comprendido entre 0.75 y 1.5 [kg].
- En base a la lectura de tesis y artículos como (Bonell, 2011), (Lezama, Sklar, & Tejera, 2005), (Chavolla, 2010), (Albero, Blanes, Benet, Simó, & Perez, 2013), (Ramírez, 2009) y teniendo en cuenta que el máximo número de grados de libertad para las extremidades inferiores es de 10 debido a lo mencionado en el perfil de tesis (para las extremidades superiores no existen restricciones), los grados de

libertad escogidos para las piernas son 10, de los cuales 4 serán los encargados de simular la cadera, 2 realizarán el giro de las rodillas (izquierda y derecha) y 4 simularán el movimiento de los tobillos (izquierdo y derecho), para la parte superior en los brazos se utilizarán 2 servomotores simulando las articulaciones de los hombros (izquierdo y derecho).

2.1.2. PARÁMETROS DE COMUNICACIÓN Y ELECTRÓNICOS.

- En este proyecto es importante definir el tipo de comunicación para el envío y la recepción de los datos, la comunicación entre los sensores, los servomotores y la tarjeta será de forma convencional (mediante cables); mientras que la comunicación entre el computador y la tarjeta electrónica es de libre selección ya sea por cable o tecnología inalámbrica.
- Después de haber investigado legos educativos como los de la marca Bioloid, NAO de Aldebaran y Mindstorms de LEGO, los mismos que dentro de una de sus configuraciones pueden desempeñarse como robot bípedos, se identifica claramente que los voltajes de los diferentes sensores y actuadores van desde 3 hasta 12 [V], por lo cual los niveles de voltaje que se establecen como parámetro de diseño se encuentran dentro de este rango, dependiendo de los elementos electrónicos y servomotores que se seleccionarán en el diseño del robot bípedo.

2.1.3. FUNCIONAMIENTO.

- El prototipo de robot bípedo camina hacia el frente de forma autónoma sobre una superficie lisa; el robot consta de un sensor de ultrasonido el cual detecta los obstáculos que se encuentren frente a él a una distancia aproximada de 40 [cm]; al sensar un obstáculo el robot detiene su marcha y el sistema de control procede a accionar los servomotores de cada una de las piernas, los cuales están

destinados a realizar los movimientos coordinados que hacen posible el giro hacia la derecha del robot para esquivar el obstáculo y así éste continúe su marcha mientras no exista otro obstáculo en frente.

2.2. DISEÑO DEL SISTEMA MECÁNICO.

En este apartado se tratará todo lo relacionado a la estructura del robot bípedo, su diseño, análisis y validación para pasar a la siguiente etapa que es el diseño del sistema electrónico.

2.2.1. DISEÑO DE LA ESTRUCTURA DEL ROBOT BÍPEDO.

Al igual que en el cuerpo humano, se busca que cada parte de la estructura del robot bípedo sea semejante a la distribución de las extremidades inferiores, como se muestra en la Figura 2.1; en la cual se indica la distribución de los huesos (pelvis, fémur, tibia y los conformados por el pie) y las articulaciones (cadera, rodilla y tobillo) de una pierna humana.



Figura 2.1. Extremidad inferior del cuerpo humano.

Fuente: <http://cmapserver.unavarra.es/rid=1MB07WGY1-23M7KN2-2ST/Extremidades.jpg>

En base a los parámetros mecánicos del literal 2.1.1 se inicia con el diseño de las piezas de la estructura del robot bípedo las cuales nacen de la creatividad e inventiva de los autores del proyecto; mediante la observación de robots humanoides de las marcas Bioloid, NAO y Kondo se comprendió el

funcionamiento de cada uno de los grados de libertad y la forma de las extremidades, con lo cual se procede a utilizar el programa SolidWorks que como se detalló en el primer capítulo éste permite la creación de prototipos y brinda los recursos necesarios para el análisis estructural de los mismos.

Primero se diseña cada una de las partes que van a conformar las extremidades inferiores del robot bípedo. en lo que se refiere al servomotor se utiliza un modelo de servomotor genérico para representar la ubicación de cada una de las articulaciones del robot y en los cuales van a estar unidas las piezas de la estructura, se escoge este modelo genérico debido a que el servomotor a utilizar se elegirá después de haber realizado el estudio dinámico del robot bípedo, el peso se estableció mediante un promedio realizado entre los servomotores que debido al tamaño, torque y forma pueden ser seleccionados para su utilización como en proyectos similares al presente caso, el promedio de los pesos es el siguiente:

- Servomotor Dynamixel MX-28: 72 [gr] (Trossen Robotics, Dynamixel MX-28T Robot Actuator, 2014)
- Servomotor Kondo KRS-788HV: 47.5 [gr] (Robot Shop, 2009)
- Servomotor Dynamixel AX-12A: 55 [gr] (Trossen Robotics, 2014)
- Servomotor Hitec HS422: 45.5 [gr] (SuperRobotica, 2002)

El peso del servomotor genérico es:

$$\frac{72 + 47.5 + 55 + 45.5}{4} = 55 \text{ [gr]}$$

De igual manera el torque de los servomotores genéricos se obtiene del siguiente promedio:

- Servomotor Dynamixel MX-28: 31.6 [kg*cm] (Trossen Robotics, Dynamixel MX-28T Robot Actuator, 2014)
- Servomotor Kondo KRS-788HV: 10 [kg*cm] (Robot Shop, 2009)

- Servomotor Dynamixel AX-12A: 15.3 [kg*cm] (Trossen Robotics, 2014)
- Servomotor Hitec HS422: 4.1 [kg*cm] (SuperRobotica, 2002)

El torque del servomotor genérico es:

$$\frac{31.6 + 10 + 15.3 + 4.1}{4} = 15.25 [Kg * cm]$$

Las piezas diseñadas del prototipo de robot bípedo se describen a continuación. Mientras que el diseño a escala real se incluye en el Anexo A dónde se indican todas las medidas en los respectivos planos.

- **Pelvis:** Estas dos piezas mostradas en la Figura 2.2 son la base donde van a estar sujetos los servomotores que van a ser los dos primeros grados de libertad de la cadera, en la parte inferior van a estar ubicadas las piernas y en la parte superior estará el tronco del robot, además se toma como el origen de referencia.

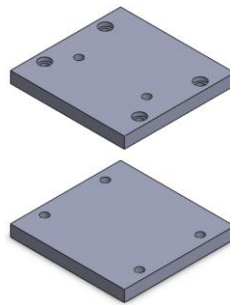


Figura 2.2. Pelvis del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Cadera:** Pieza articular de dos grados de libertad mostrada en la Figura 2.3, el primer grado es el que se une al servomotor de la pelvis y el segundo está unido al servomotor que da movimiento al fémur.

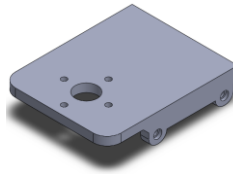


Figura 2.3. Cadera izquierda del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Fémur:** Se ha diseñado esta parte de manera que ocupe poco material y por ende tenga menor peso, por eso está diseñado en dos piezas. Se muestran estas piezas en la Figura 2.4 en la cual la parte superior va unida a la articulación de la cadera y la parte inferior al servomotor de la articulación de la rodilla.

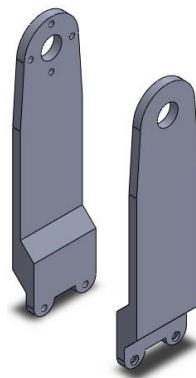


Figura 2.4. Fémur izquierdo del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Tibia:** Teniendo en cuenta que el peso de cada eslabón afecta la estabilidad como la carga que deben soportar los servomotores, el diseño de la tibia se divide en dos piezas para reducir el peso. La parte superior de los eslabones va unida a la articulación de la rodilla y la parte inferior al servomotor de la articulación del tobillo. La tibia se muestra en la Figura 2.5.

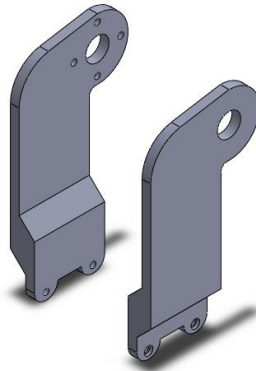


Figura 2.5. Tibia izquierda del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Tobillo:** Se muestra en la Figura 2.6, la parte superior va sujeta a la articulación del tobillo y la parte inferior va unida a la segunda articulación del tobillo que se une al pie.

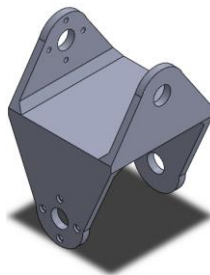


Figura 2.6. Tobillo izquierdo del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Pie:** El pie del robot bípedo se ha diseñado de tal manera que tenga la mayor superficie que se pueda en contacto con el piso asegurando de esta manera que la proyección del centro de masa se encuentre en el área de soporte para que el robot sea estable; tal como se muestra en la Figura 2.7. Para establecer el diseño del pie se consideraron los siguientes aspectos:
 - Se requiere que el área de cada pie sea grande y además que los pasos sean lo más estables posibles, para poder mantener las fuerzas dinámicas lo más bajas posibles.

- Si el punto de momento cero (ZMP) reside dentro del área de soporte del pie del robot (o ambos pies) sobre el piso, el robot se encuentra en balance dinámico.

Los aspectos anteriormente mencionados fueron tomados de (Torres, 2009, pp. 17,32) en base al punto de momento cero.

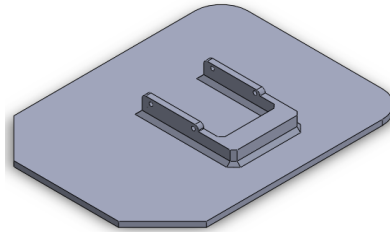


Figura 2.7. Pie izquierdo del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Al tener diseñadas todas las piezas de las extremidades inferiores se procede a realizar el ensamblaje el cual se muestra en la Figura 2.8, teniendo en cuenta que todas las relaciones de posición estén planificadas correctamente para que no existan errores y tenga un funcionamiento óptimo. La manera en que están ubicados los servomotores genéricos es para que cada articulación realice el movimiento deseado ya que estos actuadores tienen un rango de ángulos de trabajo y eso afecta al momento de planificar las trayectorias.

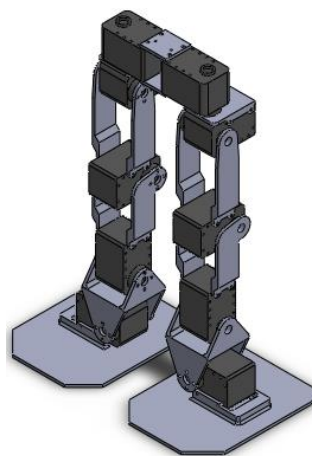


Figura 2.8. Ensamblaje de las extremidades inferiores del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

A continuación se procede a diseñar la parte superior del robot.

- **Tronco:** Un sub ensamblaje esencial que consta de 3 piezas, en el cual la parte inferior estará unida a la pelvis, a los lados los servomotores del hombro que dan movimiento a los brazos; en la parte interior del tronco va a ir alojada la batería, en la espalda la tarjeta electrónica y en la parte superior la cabeza, se indica el tronco del robot en la Figura 2.9.

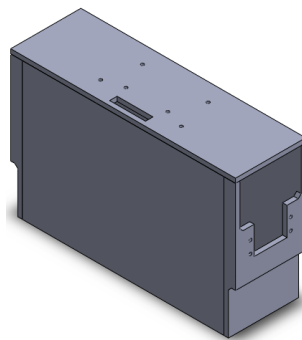


Figura 2.9. Tronco del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Brazo:** Conformado de dos piezas, como se muestra en la Figura 2.10, tanto izquierdo como derecho; van unidos a la articulación del hombro respectivo, solo tienen un grado de libertad el cual simulará el movimiento que realiza el brazo de una persona tanto adelante como hacia atrás al momento de caminar.

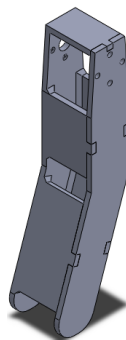


Figura 2.10. Brazo izquierdo del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

- **Cabeza:** Se muestra en la Figura 2.11, en la parte frontal va a estar alojado el sensor que detectará los obstáculos que estén frente al robot, debido a la forma que tiene el sensor éste simulará o dará la apariencia de ser los ojos del robot bípedo. En la parte posterior estará alojado un pequeño circuito con un switch y un pulsador.

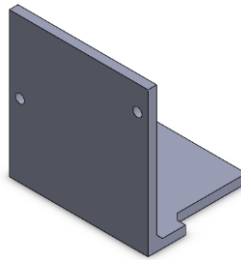


Figura 2.11. Cabeza del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Al tener todas las piezas diseñadas se procede a realizar el ensamblaje completo del prototipo de robot bípedo, el cual se muestra en la Figura 2.12.

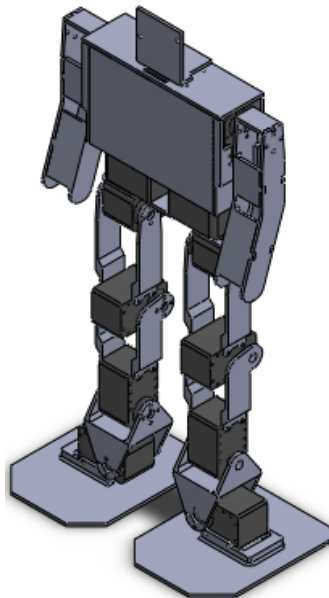


Figura 2.12. Ensamblaje del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

En base al estudio del arte realizado en el capítulo uno se concluyó que los materiales más utilizados en la construcción de prototipos de robots

bípedos son la madera, el aluminio y el polímero ABS, cada uno de éstos tienen propiedades diferentes pero con buenos resultados, también se ha comparado las propiedades físico – mecánicas correspondientes a cada material en las Tablas 1.2 (MDF), 1.3 (ABS) y 1.4 (Aluminio).

Para esta estructura se ha elegido al material ABS ya que su uso es relativamente nuevo y está en auge su ocupación, pudiendo realizar con este material diversos diseños de piezas a diferencia que con la madera y el aluminio los cuales logran buenos resultados en la fabricación de piezas 2D, pero en piezas complejas de diseño 3D como los planteados en este trabajo los resultados no son favorables y por eso se descarta su utilización. Además el ABS al ser trabajado en una impresora 3D permite el prototipado rápido de las piezas sin importar el número de ellas, ya que se va a obtener una forma igual a la esperada.

2.2.2. ANÁLISIS ESTÁTICO DE LA ESTRUCTURA DEL ROBOT.

Se tiene que realizar este análisis en la estructura para verificar que el diseño sea capaz de soportar el peso de todas las piezas y servomotores además del torque de los mismos.

En el software SolidWorks se asigna el material ABS a cada una de las piezas, se espera que sus propiedades físicas puedan resistir el peso y torques al que van a estar sometidas, de esta manera poder validar el diseño con este material después de haber analizado las tensiones de Von Mises y el factor de seguridad.

A los servomotores genéricos se les ha asignado como material un plástico ya que todavía no se sabe que servomotor se ocupará y por lo general la estructura de la mayoría de servomotores están construidos de ese material, también se tiene que asignar el peso obtenido a través del promedio en las propiedades físicas de la pieza del servomotor para que el estudio de un resultado óptimo.

Los cambios realizados en el ensamblaje se muestran en la Figura 2.13, se debe notar el cambio del color y textura debido a los materiales asignados.

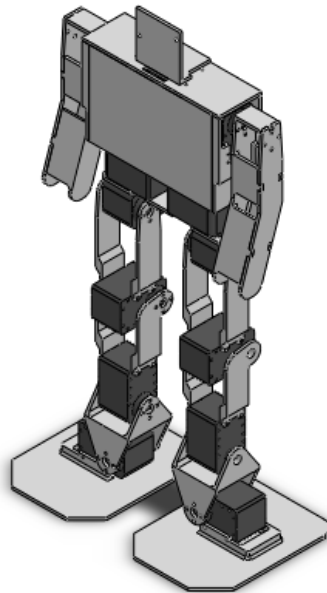


Figura 2.13. Ensamblaje del robot bípedo con materiales asignados.

Autores: Eduardo Morillo y Juan Portero.

En la Tabla 2.1 se indica la cantidad de piezas necesarias y el peso de cada una de ellas con los materiales ya asignados, se puede verificar que el peso total está dentro del rango estipulado en el literal 2.1.1. (Los datos de las piezas se han tomado de las propiedades físicas del programa SolidWorks y el peso de los servomotores del promedio realizado previamente).

Tabla 2.1. Elementos para el ensamblaje del robot bípedo.

Elemento	Cantidad	Peso Individual [gr]	Peso Global [gr]
Pelvis	2	2.1	4.2
Cadera	2	6.2	12.4
Fémur	2	15.1	30.2
Tibia	2	13.3	26.6
Tobillo	2	19.7	39.4
Pie	2	49.4	98.8
Tronco	1	149	149

Continúa 

Brazo	2	38.1	76.2
Cabeza	1	11.9	11.9
Servomotores	12	55	660
TOTAL	26		1108.7

Autores: Eduardo Morillo y Juan Portero.

Al tener lista la estructura y conocer que su peso está dentro de lo requerido, se debe crear un nuevo estudio estático en el paquete de SolidWorks Simulation, lo primero que hay que hacer es ingresar el valor del vector gravedad para que tenga influencia el peso de las piezas al momento de realizar los cálculos, la dirección de este vector automáticamente se fija en el centro de masa del ensamblaje como muestra la Figura 2.14.

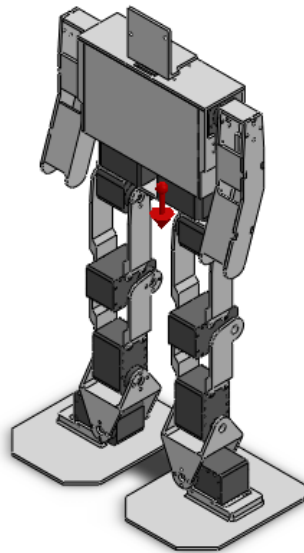


Figura 2.14. Vector gravedad del estudio estático.

Autores: Eduardo Morillo y Juan Portero.

Ahora se procede a simular que los pies están en el piso, para esto se crea una **sujeción** de **geometría fija** y se seleccionan las plantas de los pies como indica la Figura 2.15.

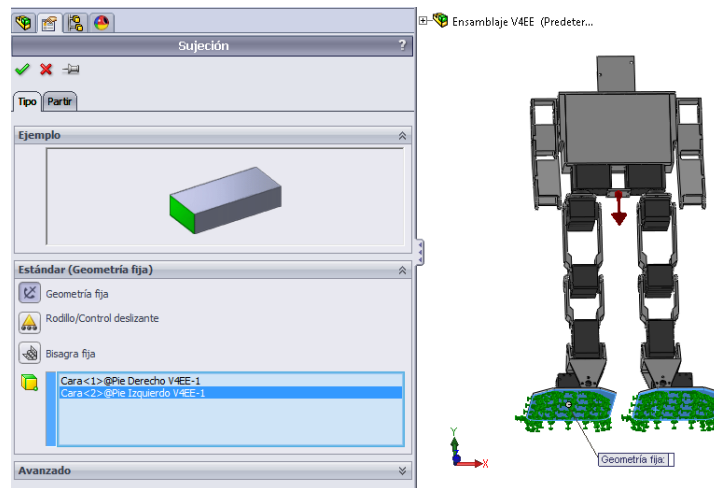


Figura 2.15. Geometría fija en los pies.

Autores: Eduardo Morillo y Juan Portero.

A continuación en el menú **cargas externas** se selecciona **torsión** para simular el torque al que va a estar sometida cada articulación del robot bípedo, el torque a ingresar se ha tomado del promedio realizado a los servomotores, se tiene que elegir la cara y el eje donde se aplica la torsión, esto quiere decir que va en la cara de la pieza que está unida a cada uno de los ejes de los servomotores como indica la Figura 2.16.

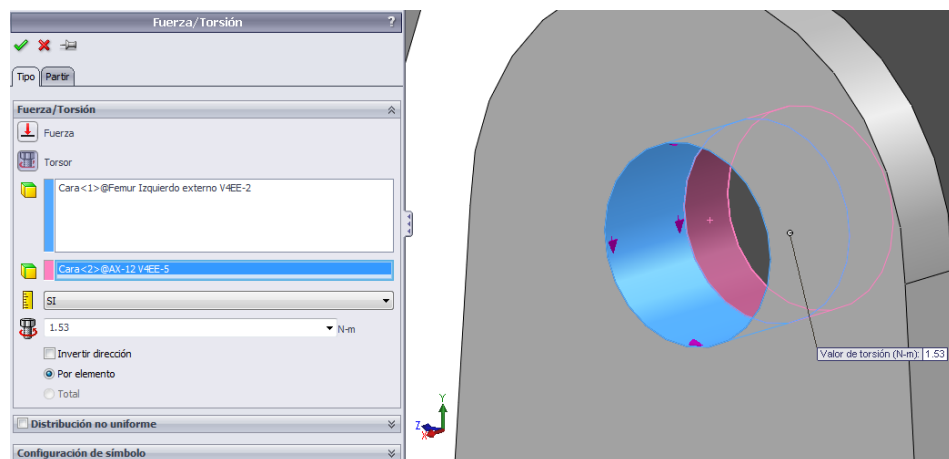


Figura 2.16. Torque en las piezas.

Autores: Eduardo Morillo y Juan Portero.

La Figura 2.17 muestra los torques de los servomotores y fuerzas que actúan sobre el robot bípedo.

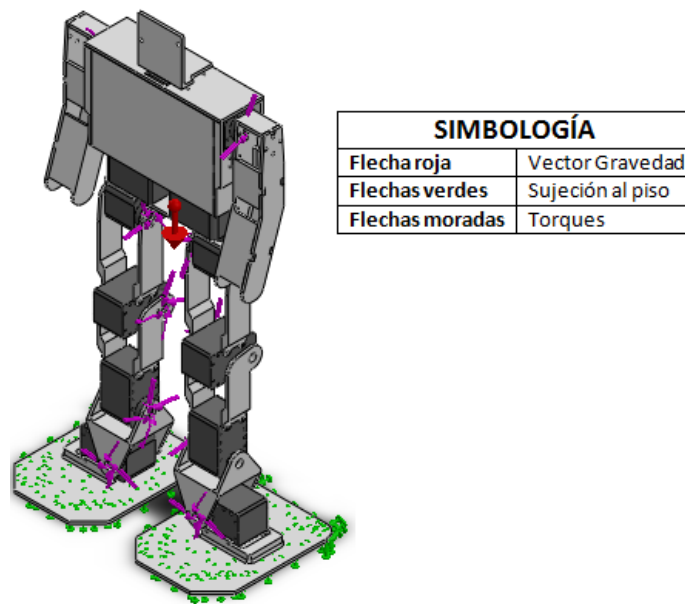


Figura 2.17. Torques y fuerzas ejercidas en el robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Se procede a crear la malla, entre más fina mejores resultados se obtendrán, pero depende mucho del procesador del computador y de la geometría de la estructura, para este proyecto se ha ocupado una malla de 5 [mm] debido a que como el diseño es de un prototipo no se necesita utilizar una malla de menor tamaño ya que esto complicaría el cálculo y con esta malla es suficiente para obtener buenos resultados los cuales se muestran en la Figura 2.18.

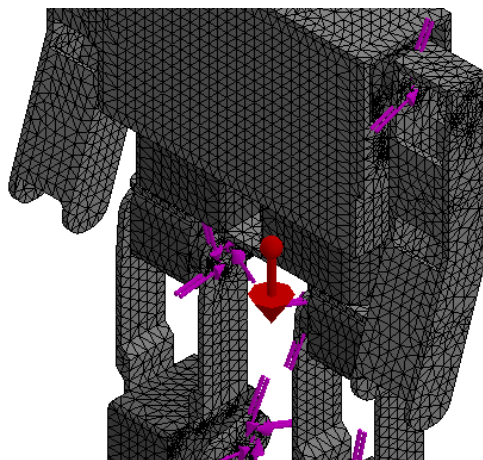


Figura 2.18. Mallado de la estructura del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Finalmente se realiza el análisis estático en la pestaña **ejecutar**, este proceso puede que tarde minutos hasta un par de horas dependiendo de la capacidad del procesador del computador.

En el análisis de esfuerzos, la teoría de Von Mises expone que un material dúctil (material que si se aplica una fuerza, puede deformarse sosteniblemente sin romperse a diferencia de los materiales frágiles) no cede en una ubicación cuando la tensión de Von Mises es menor al límite elástico (SolidWorks, 2011).

$$\sigma_{VonMises} < \sigma_{LimiteElastico}$$

Ec.2.14

En la Figura 2.19 se observa el valor máximo de la tensión de Von Mises que esta aplicada a la estructura del robot bípedo, este valor es de 11.55 [MPa], al aplicar la ecuación 2.1 se puede comprobar que este valor es inferior al valor del límite elástico el cual es de 38.8 [MPa] (Tabla 1.3) y por lo tanto no existe fallo en el material.

$$\sigma_{VonMises} < \sigma_{LimiteElastico}$$

$$11.55 [MPa] < 38.8 [MPa]$$

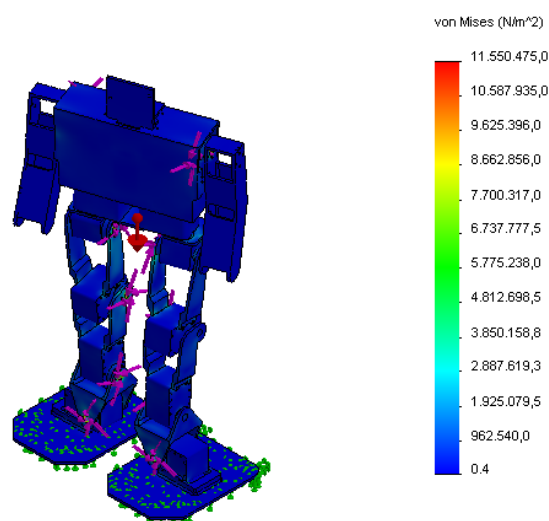


Figura 2.19. Tensiones de Von Mises en la estructura del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Además se considera que el esfuerzo de diseño tiene que ser mayor al esfuerzo máximo que soporta la estructura para que no exista fallo en la misma (Mott, 2006).

$$\sigma_d \geq \sigma' \quad \text{Ec.2.2}$$

Dónde:

- σ_d : Esfuerzo de diseño.
- σ' : Esfuerzo soportado por la estructura.

A su vez el esfuerzo de diseño es:

$$\sigma_d = \frac{S_y}{N} \quad \text{Ec.2.3}$$

Dónde:

- S_y : Resistencia a la flexión.
- N : Factor de seguridad.

El factor de seguridad es el índice de seguridad en el diseño, un factor de seguridad igual o superior a 1 significa que el material es seguro y es conveniente que su magnitud sea evaluada dependiendo el tipo de aplicación (González, 2009).

Al aplicar la ecuación 2.3 del esfuerzo de diseño y si se considera un factor de seguridad de 1.5 y la resistencia a la flexión de 45 [MPa] (Tabla 1.3) se obtiene:

$$\frac{S_y}{N} = \sigma_d$$

$$\frac{45 \text{ [MPa]}}{1.5} = 30 \text{ [MPa]}$$

Al aplicar la ecuación 2.2, se determinó que:

$$\sigma_d \geq \sigma'$$

$$30 [MPa] \geq 11.55 [MPa]$$

De esta manera se comprueba que el esfuerzo de diseño es mayor al esfuerzo que soporta la estructura.

Al analizar el gráfico del factor de seguridad mostrado en la Figura 2.20 se comprueba que el valor mínimo es de 3.36 siendo un valor óptimo para el diseño.

Finalmente en base a los resultados obtenidos en este análisis estático se concluye que la estructura resiste y no tiene fallos de seguridad estructural con el material asignado.

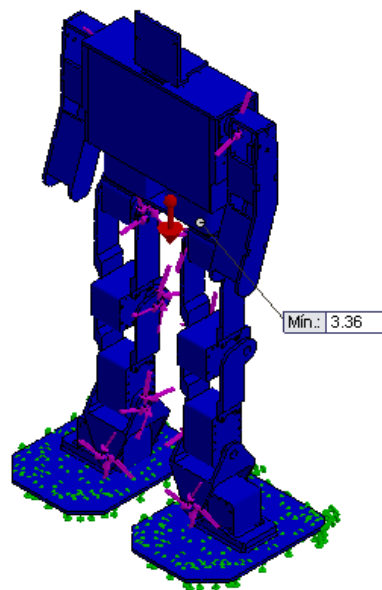


Figura 2.20. Factor de seguridad en la estructura del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

2.2.3. CINEMÁTICA DEL ROBOT BÍPEDO.

En este punto del diseño mecánico se va a obtener la cinemática directa de una de las piernas de la estructura del prototipo de robot bípedo, para

saber si la misma cumple con las posiciones que se desean, cabe recalcar que este estudio se realiza en una pierna debido a que las piernas son simétricas y los resultados son equivalentes y por ende se realizaría en vano el estudio en las dos piernas.

La resolución de la cinemática directa permite obtener la posición y orientación del efector final del robot (punta del pie) con respecto a un sistema de coordenadas de referencia fijo (primer servomotor de la cadera).

Primero se determinan los parámetros D-H mediante el procedimiento de (Vele, 2008, pp. 19-22), dicho procedimiento se aplicará a este proyecto y se lo realiza de la siguiente manera.

Identificación de articulaciones:

1. Se debe enumerar los eslabones comenzando con 1 en el primer eslabón móvil de la cadena cinemática y acabando con el último eslabón (pie del robot bípedo). Se numera como eslabón 0 a la base fija del robot bípedo, la Figura 2.21 muestra cómo van enumerados los eslabones para el caso del robot bípedo.

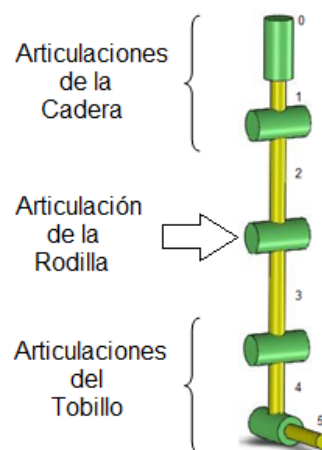


Figura 2.21. Enumeración de eslabones D-H.

Autores: Eduardo Morillo y Juan Portero.

2. Se debe localizar el eje de cada articulación como se muestra en la Figura 2.22, si ésta es rotativa el eje será su propio eje de giro, si es

prismática será el eje a lo largo del cual se produce el desplazamiento.

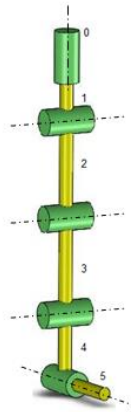


Figura 2.22. Localización de los ejes en las articulares D-H.

Autores: Eduardo Morillo y Juan Portero.

3. Se procede a enumerar cada articulación comenzando por 1 que sería la correspondiente al primer grado de libertad y acabando al llegar al pie del robot bípodo como se demuestra en la Figura 2.23.

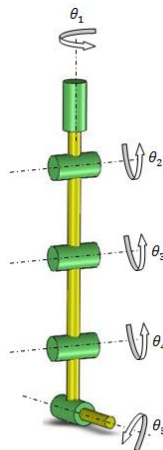


Figura 2.23. Enumeración de articulaciones D-H.

Autores: Eduardo Morillo y Juan Portero.

Localización de ejes:

1. Para i de 0 a $n - 1$, se coloca el eje Z_i , sobre el eje de la articulación $i + 1$.

2. Después se sitúa el origen del sistema de la base (S_0) en cualquier punto del eje Z_0 . Los ejes X_0 e Y_0 se situaran de modo que formen un sistema dextrógiro con Z_0 .
3. Para i de 1 a $n - 1$, se posiciona el sistema (S_i) (solidario al eslabón i) en la intersección del eje Z_i con la línea normal común a Z_{i-1} y Z_i . Si ambos ejes se cortasen se situaría (S_i) en el punto de corte. Si fuesen paralelos (S_i) se situaría en la articulación $i + 1$.
4. A continuación se coloca X_i en la línea normal común a Z_{i-1} y Z_i .
5. Después se sitúa Y_i de modo que forme un sistema dextrógiro con X_i y Z_i .
6. Y finalmente se ubica el sistema S_n en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y X_n sea normal a Z_{n-1} y Z_n .

Para encontrar el sentido de giro de las articulaciones se aplica la regla de la mano derecha, con el dedo pulgar siguiendo a Z_i . La Figura 2.24 indica la localización de todos los ejes de la pierna del robot bípedo al finalizar los pasos anteriormente señalados.

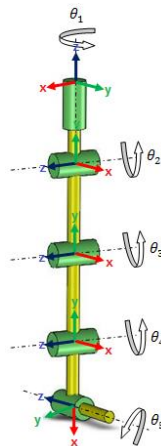


Figura 2.24. Localización de los ejes D-H.

Autores: Eduardo Morillo y Juan Portero.

Obtención de parámetros:

1. En primer lugar se encuentra θ_i como el ángulo que hay que girar en torno a Z_{i-1} para que X_{i-1} y X_i queden paralelos.
2. Después se halla d_i como la distancia, medida a lo largo de Z_{i-1} , que habría que desplazar (S_{i-1}) para que X_i y X_{i-1} quedasen alineados.
3. A continuación se encuentra a_i como la distancia medida a lo largo de X_i (que ahora coincidiría con X_{i-1}) que habría que desplazar el nuevo (S_{i-1}) para que su origen coincidiese con (S_i).
4. Finalmente se obtiene α_i como el ángulo que habría que girar en torno a X_i (que ahora coincidiría con X_{i-1}), para que el nuevo (S_{i-1}) coincidiese totalmente con (S_i).

Al seguir todo el procedimiento anterior se obtiene como resultado la Figura 2.25.

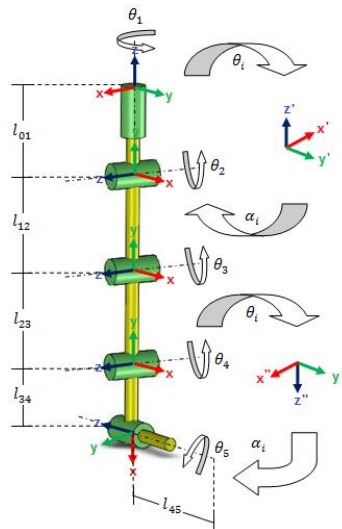


Figura 2.25. Obtención de los parámetros D-H.

Autores: Eduardo Morillo y Juan Portero.

A continuación se procede a obtener la Tabla 2.2 de los parámetros D-H de la Figura anterior que representa una de las piernas del robot bípedo.

Tabla 2.2. Parámetros D-H.

Articulación	θ	d	a	α
1	$\theta_1 + \pi/2$	l_{01}	0	$\pi/2$
2	θ_2	0	l_{12}	0
3	θ_3	0	l_{23}	0
4	$\theta_4 - \pi/2$	0	l_{34}	$\pi/2$
5	θ_5	0	l_{45}	0

Autores: Eduardo Morillo y Juan Portero

Una vez obtenidos los parámetros D-H para la pierna del robot bípedo se procede a la obtención de la matriz de transformación homogénea que permite conocer la localización del efector final con respecto al sistema de coordenadas de referencia con lo que queda resuelto el problema cinemático directo.

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$$

Desarrollando cada una de las matrices según la ecuación 1.12 se obtiene:

La matriz 0A_1 :

$${}^0A_1 = \begin{bmatrix} \cos\theta_1 & -\cos\pi/2 \sin\theta_1 & \sin\pi/2 \sin\theta_1 & a_1 \cos\theta_1 \\ \sin\theta_1 & \cos\pi/2 \cos\theta_1 & -\sin\pi/2 \cos\theta_1 & a_1 \sin\theta_1 \\ 0 & \sin\pi/2 & \cos\pi/2 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0A_1 = \begin{bmatrix} \cos(\theta_1 + \pi/2) & 0 & \sin(\theta_1 + \pi/2) & 0 \\ \sin(\theta_1 + \pi/2) & 0 & -\cos(\theta_1 + \pi/2) & 0 \\ 0 & 1 & 0 & l_{01} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0A_1 = \begin{bmatrix} -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & l_{01} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz 1A_2 :

$${}^1A_2 = \begin{bmatrix} \cos\theta_2 & -\cos(0)\text{sen}\theta_2 & \text{sen}(0)\text{sen}\theta_2 & a_2\cos\theta_2 \\ \text{sen}\theta_2 & \cos(0)\cos\theta_2 & -\text{sen}(0)\cos\theta_2 & a_2\text{sen}\theta_2 \\ 0 & \text{sen}(0) & \cos(0) & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & l_{12}\cos\theta_2 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & l_{12}\text{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz 2A_3 :

$${}^2A_3 = \begin{bmatrix} \cos\theta_3 & -\cos(0)\text{sen}\theta_3 & \text{sen}(0)\text{sen}\theta_3 & a_3\cos\theta_3 \\ \text{sen}\theta_3 & \cos(0)\cos\theta_3 & -\text{sen}(0)\cos\theta_3 & a_3\text{sen}\theta_3 \\ 0 & \text{sen}(0) & \cos(0) & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_{23}\cos\theta_3 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & l_{23}\text{sen}\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz 3A_4 :

$${}^0A_1 = \begin{bmatrix} \cos(\theta_4 - \pi/2) & -\cos\pi/2 \text{sen}(\theta_4 - \pi/2) & \text{sen}\pi/2 \text{sen}(\theta_4 - \pi/2) & a_5\cos(\theta_4 - \pi/2) \\ \text{sen}(\theta_4 - \pi/2) & \cos\pi/2 \cos(\theta_4 - \pi/2) & -\text{sen}\pi/2 \cos(\theta_4 - \pi/2) & a_5\text{sen}(\theta_4 - \pi/2) \\ 0 & \text{sen}\pi/2 & \cos\pi/2 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$${}^3A_4 = \begin{bmatrix} \cos(\theta_4 - \pi/2) & 0 & \text{sen}(\theta_4 - \pi/2) & a_4\cos(\theta_4 - \pi/2) \\ \text{sen}(\theta_4 - \pi/2) & 0 & -\cos(\theta_4 - \pi/2) & a_4\text{sen}(\theta_4 - \pi/2) \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$


$${}^3A_4 = \begin{bmatrix} \text{sen}\theta_4 & 0 & -\cos\theta_4 & l_{34}\cos\theta_4 \\ -\cos\theta_4 & 0 & -\text{sen}\theta_4 & -l_{34}\text{sen}\theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz 4A_5 :

$${}^4A_5 = \begin{bmatrix} \cos\theta_5 & -\cos(0)\sin\theta_5 & \sin(0)\sin\theta_5 & a_5\cos\theta_5 \\ \sin\theta_5 & \cos(0)\cos\theta_5 & -\sin(0)\cos\theta_5 & a_5\sin\theta_5 \\ 0 & \sin(0) & \cos(0) & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4A_5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & l_{45}\cos\theta_5 \\ \sin\theta_5 & \cos\theta_5 & 0 & l_{45}\sin\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matlab ofrece una poderosa herramienta gratuita creada por el Ingeniero Peter Corke que permite realizar el cálculo de la cinemática y dinámica de un robot, entre otras aplicaciones de robótica; este complemento es el Robotics Toolbox el cual se lo puede descargar desde el siguiente enlace http://petercorke.com/Robotics_Toolbox.html, hay que descargar según la versión de Matlab utilizada en el computador; se descarga el siguiente archivo  [robot-9.8.zip](#) que hay que descomprimirlo. Para proceder a utilizar el toolbox hay que ejecutar dentro de Matlab el siguiente archivo  [startup_rvc.m](#) a continuación sale un mensaje indicando que está listo para usarse, en ese momento se pueden utilizar sus funciones, cabe denotar que no se está instalando un programa sino solo se ejecuta el complemento, por eso cada vez que se inicie una nueva sesión de Matlab hay que volver a ejecutar el archivo mencionado anteriormente para poder utilizar el Robotics Toolbox.

Al ser un software gratuito en la misma página de descarga está un link para descargar el manual llamado  [robot.pdf](#), en el cual se explica cada una de sus funciones; además se puede adquirir el libro Robotics, Vision & Control (Corke, 2011) muy útil para entender la Toolbox y todas sus aplicaciones.

Después de esta pequeña introducción a la herramienta que se utilizará para el cálculo de la cinemática directa del robot bípedo, se explica paso a paso como crear el robot y como obtener la cinemática directa de una forma muy sencilla.

Se inicia comprendiendo la instrucción **DHFactor** la cual obtiene los parámetros D-H en base a las rotaciones y traslaciones que tiene que hacer el robot a partir de su base hasta la punta del pie, como se observó anteriormente solo se va a hacer el cálculo de una pierna a partir de la base de la pelvis hasta la punta del pie, en la Figura 2.26 se muestran las rotaciones y traslaciones necesarias para poder obtener la cadena cinemática del robot bípedo.

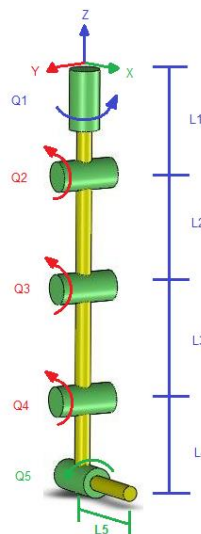


Figura 2.26. Parámetros para obtener el comando *DHFactor*.

Autores: Eduardo Morillo y Juan Portero.

La Figura 2.26 se interpreta de la siguiente manera:

- Rotación de un ángulo Q_1 alrededor del eje Z.
- Traslación de una longitud L_1 a lo largo del eje Z.
- Rotación de un ángulo Q_2 alrededor del eje X.
- Traslación de una longitud L_2 a lo largo del eje Z.
- Rotación de un ángulo Q_3 alrededor del eje X.
- Traslación de una longitud L_3 a lo largo del eje Z.
- Rotación de un ángulo Q_4 alrededor del eje X.

- Traslación de una longitud L_4 a lo largo del eje Z.
- Rotación de un ángulo Q_5 alrededor del eje Y.
- Traslación de una longitud L_5 a lo largo del eje Y.

En la ventana de comandos de Matlab se guarda en la variable **s** la secuencia de los parámetros los cuales van a ser la cadena cinemática como se muestra a continuación.

```
s = 'Rz(q1) .Tz(L1) .Rx(q2) .Tz(L2) .Rx(q3) .Tz(L3) .Rx(q4) .Tz(L4) .Ry(q5) .Ty(L5) ';
```

A continuación se ejecuta el comando **DHFactor** para obtener los parámetros D-H de la variable **s** y que van a ser guardados en una nueva variable llamada **dh**.

```
>> dh = DHFactor(s)
```

Para observar de una manera amigable los resultados se utiliza el comando.

```
>> dh.display
```

Mostrando los siguientes resultados, que si se comparan con los de la Tabla 2.2, no son los mismos pero al resolver las ecuaciones de la matriz de transformación homogénea se obtiene el mismo resultado y así se puede comprobar de una manera rápida y sencilla que todos los cálculos fueron realizados de manera correcta.

```
DH(q1+90, L1, 0, 90)
```

```
DH(q2-90, 0, -L2, 0)
```

```
DH(q3, 0, -L3, 0)
```

```
DH(q4, 0, -L4, -90)
```

```
DH(q5+90, L5, 0, 90)
```

A continuación se procede a ingresar los valores de las longitudes de cada eslabón, las medidas se tienen que introducir en metros [m].

```
>> L1 = 0.0205;
```

```
>> L2 = 0.0940;
```

```
>> L3 = 0.0940;
```

```
>> L4 = 0.0500;
```

```
>> L5 = 0.0569;
```

Estas longitudes se han obtenido desde SolidWorks midiendo la distancia entre cada eje de los servomotores ya que éstos representan las articulaciones, la distancia entre cada articulación se toma como si fuera la medida del eslabón. Se muestra en la Figura 2.27 la obtención de la longitud L_2 .

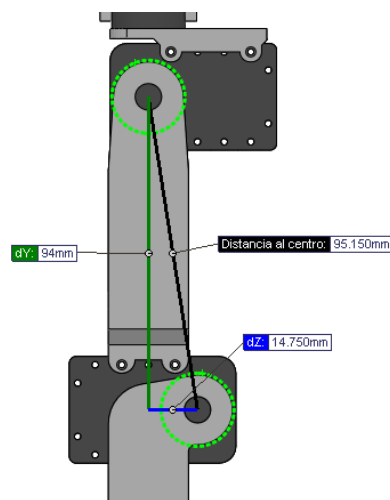


Figura 2.27. Obtención de la longitud L_2 .

Autores: Eduardo Morillo y Juan Portero.

A continuación se debe crear el robot en base a los parámetros D-H y las longitudes de los eslabones, creando lo que se conoce como **SerialLink** que no es más que el robot en sí, de la siguiente manera.

```
>> modelado = dh.command('pierna bipedo');
```

```
>> pierna = eval(modelado);
```

Se crea la variable **modelado** la cual contiene el comando `dh.command` que es el que crea el robot con el nombre **pierna bipedo** en base a los datos proporcionados por la variable **dh**. Con el comando **eval** se evalúan todos los datos ingresados y finalmente la variable que contiene al robot en forma de **SerialLink** se llama **pierna** que es la variable con la que se debe trabajar para obtener la cinemática, dinámica y todo lo que se requiera.

Para observar de una manera amigable los resultados se utiliza el comando.

```
>> pierna.display
```

Mostrando la tabla de los parámetros D-H (longitudes en metros y ángulos en radianes), el vector gravedad, la matriz de base de las transformaciones y la matriz de la transformación de la herramienta que en este caso es el pie.

```
pierna =

pierna bipedo (5 axis, RRRRR, stdDH)

+---+-----+-----+-----+-----+
| j |   theta |     d |     a |   alpha |
+---+-----+-----+-----+-----+
| 1 |    q1 | 0.0205|     0 |   1.571 |
| 2 |    q2 |     0 | -0.094|     0 |
| 3 |    q3 |     0 | -0.094|     0 |
| 4 |    q4 |     0 |  -0.05| -1.571 |
| 5 |    q5 | 0.0569|     0 |   1.571 |
+---+-----+-----+-----+-----+

grav =    0 base = 1 0 0 0 tool = 1 0 0 0
          0         0 1 0 0         0 1 0 0
        9.81        0 0 1 0         0 0 1 0
                   0 0 0 1         0 0 0 1
```

A continuación se calcula la cinemática directa para encontrar el efector final en cualquier posición de las articulaciones (servomotores), a través de la siguiente instrucción.

```
>> fkine(pierna, [0,0,0,0,0])
```

En este caso no se ha hecho ninguna variación en los ángulos ya que se considera esta posición como la posición inicial en la que va a estar totalmente erguido el robot bípedo, mostrando los siguientes resultados.

```
ans =
-1.0000    0.0000    0.0000   -0.0000    X
 0.0000    1.0000    0.0000    0.0569    Y
-0.0000    0.0000   -1.0000    0.2585    Z
      0          0          0      1.0000
```

El vector encerrado en el círculo muestra los valores de desplazamientos en los tres ejes del efector final con respecto al origen de referencia.

$X = 0$ [cm].

$Y = 5.69$ [cm].

$Z = 25.85$ [cm].

En la Figura 2.28 se interpreta este resultado, la distancia en el eje Y se encuentra marcada de color amarillo y la distancia en el eje Z se encuentra de color rojo.

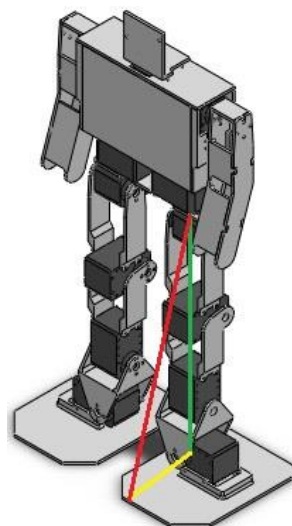


Figura 2.28. Posición del efector final del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Para visualizar el robot bípido en forma gráfica, se utiliza el siguiente comando.

```
>> pierna.plot([0,0,0,0,0], 'nobase', 'noshadow', 'noname');
```

El cual muestra al robot en la posición $[0,0,0,0,0]$, tal como se observa en la Figura 2.29.

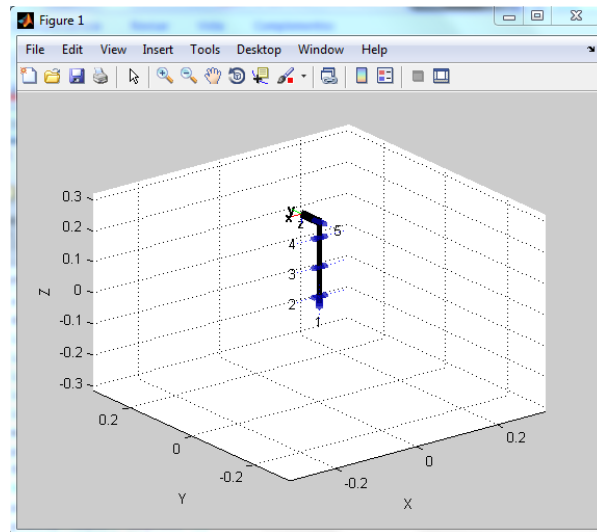


Figura 2.29. Pierna del robot bípido en la posición inicial.

Autores: Eduardo Morillo y Juan Portero.

Pero de esta manera no se puede apreciar que es una pierna, más bien parece ser un brazo robótico, para eso se da las siguientes instrucciones que solo cambian las direcciones de los ejes mas no afecta a los cálculos.

```
>> set(gca, 'Zdir', 'reverse');
>> set(gca, 'Ydir', 'reverse');
```

Mostrando la gráfica de la Figura 2.30 y ahora si pareciéndose a la Figura 2.26 que es como se inició todo este análisis cinemático en el Robotics Toolbox.

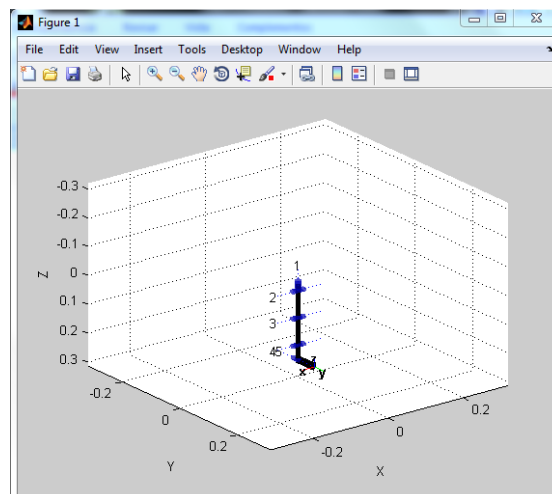


Figura 2.30. Pierna del robot bípodo visualización correcta.

Autores: Eduardo Morillo y Juan Portero.

Para mejorar aún más la gráfica se utiliza la instrucción mostrada a continuación la cual agrega un control para modificar el valor de los ángulos de las articulaciones, además que muestra la cinemática directa del efector final del robot en la parte superior del cuadro.

```
>> pierna.teach('degrees')
```

La instrucción **degrees** muestra los valores de los ángulos de las articulaciones en grados. La Figura 2.31 muestra el robot en la posición inicial.

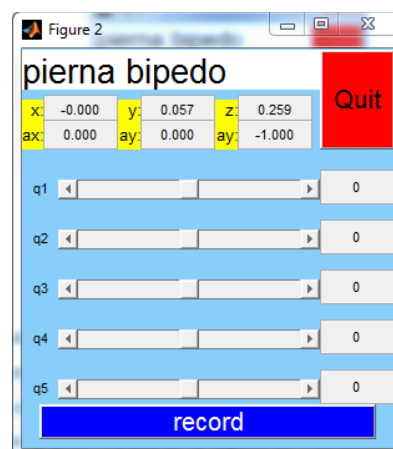


Figura 2.31. Valores articulares en la posición inicial.

Autores: Eduardo Morillo y Juan Portero.

Con todas estas instrucciones se ha terminado el cálculo de la cinemática directa del robot bípedo, como ejemplo se evalúa y se muestra el robot en la posición como que estuviera elevada la pierna, doblada la rodilla y con el pie paralelo al piso, mediante la siguiente instrucción.

```
>> fkine(pierna, [0, -pi/2, pi/2, 0, 0])
```

```
ans =
```

-1.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.1509
-0.0000	0.0000	-1.0000	0.1645
0	0	0	1.0000

Para la gráfica de la Figura 2.32.

```
>> pierna.plot([0, -pi/2, pi/2, 0, 0], 'nobase', 'noshadow', 'noname');
>> set(gca, 'Zdir', 'reverse');
>> set(gca, 'Ydir', 'reverse');
```

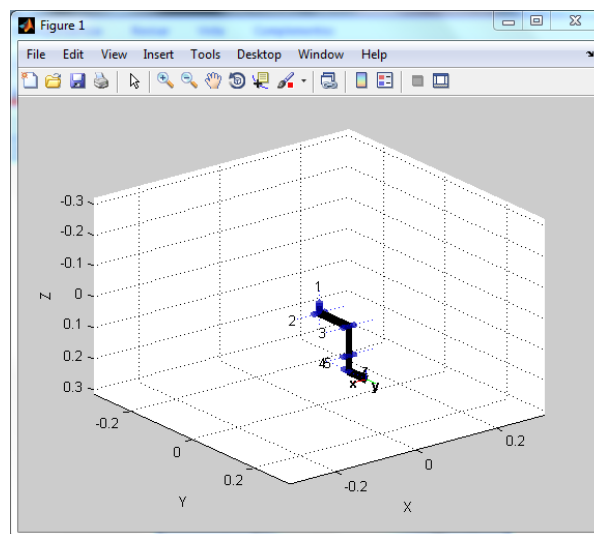


Figura 2.32. Ejemplo cinemática directa del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

```
>> pierna.teach('degrees');
```



Figura 2.33. Valores articulares del ejemplo de cinemática directa del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Como se puede observar en la Figura 2.33, los datos que están encerrados en los círculos rojos los cuales son entregados por la función ***fkine*** y la función ***teach*** coinciden entre sí, por lo tanto se concluye que todo fue realizado correctamente ya que son dos procedimientos distintos para obtener los valores de la cinemática directa.

2.2.4. DINÁMICA DEL ROBOT BÍPEDO.

El objetivo esencial de la resolución de la dinámica de un robot es la obtención de las fuerzas y torques aplicadas a cada una de sus articulaciones. De la misma manera como se resolvió la cinemática, mediante el Robotics Toolbox se puede resolver la dinámica de un robot por medio de funciones que permiten su cálculo.

Para poder resolver la dinámica de un robot se necesitan los datos de la masa, centro de gravedad y matriz de inercia de cada uno de los eslabones con respecto a la base del robot, obtener estos parámetros de manera matemática es complicado y conlleva mucho tiempo ya que las piezas tienen una forma irregular, pero gracias a la ayuda de SolidWorks se consiguen

estos datos de manera sencilla y estos valores se deben ingresar en el **SerialLink** creado anteriormente el cual fue llamado **pierna**.

Se seleccionan individualmente las propiedades físicas de cada eslabón del robot bípedo; de esta manera el programa muestra el cuadro de la Figura 2.34 en el que están todos los datos de ese eslabón (fémur izquierdo) y de los cuales se requieren la masa en kilogramos, vector del centro de gravedad en metros y la matriz de inercia en kilogramos por metro cuadrado.

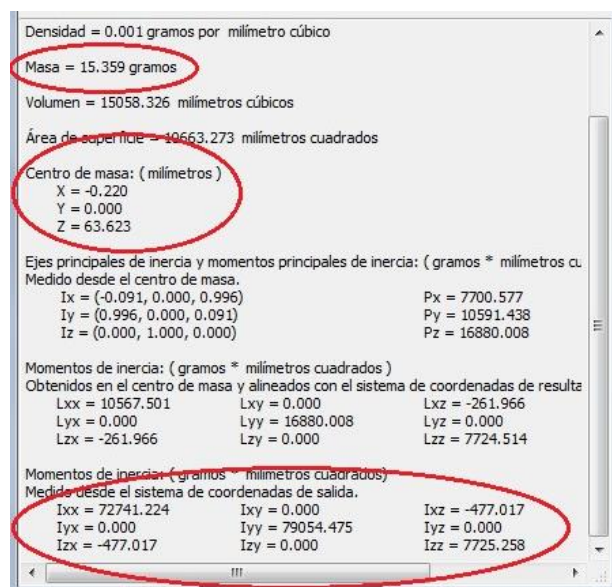


Figura 2.34. Propiedades físicas del fémur izquierdo.

Autores: Eduardo Morillo y Juan Portero.

Nótese que la masa del fémur es de 15.4 [gr] pero a este peso hay que sumar el peso del servomotor genérico de esta articulación para obtener el peso total del eslabón que es de 70.4 [gr], esto también se debe realizar para los otros cuatro eslabones (cadera, tibia, tobillo y pie); además al primer eslabón (cadera) se le suma el peso que va a soportar de la estructura superior, de la batería, sensor y tarjeta electrónica, se ha calculado que este peso es de 590 [gr], por eso el peso total de este eslabón es de 661 [gr].

Los datos de las masas, centros de gravedad y matrices de inercia de cada eslabón obtenidos mediante la ventana de propiedades físicas del

SolidWorks, se han guardado en un archivo *.m* de Matlab y se muestran en la Figura 2.35, donde m_1 , m_2 , m_3 , m_4 y m_5 son las masas de los eslabones, cog_1 , cog_2 , cog_3 , cog_4 y cog_5 son los centros de gravedad y I_1 , I_2 , I_3 , I_4 e I_5 son las matrices de inercias de cada uno de los eslabones respectivamente.

```

m1 = 0.6610;
m2 = 0.0704;
m3 = 0.0687;
m4 = 0.0749;
m5 = 0.1044;


cog1 = [0 -0.014 0.002];
cog2 = [0 0 0.064];
cog3 = [0 -0.002 0.142];
cog4 = [0 0 0.231];
cog5 = [0.021 0.005 0.280];

I1 = [2.2514e-06 0 0; 0 9.7771e-07 -1.9523e-07; 0 -1.9523e-07 3.1359e-06];
I2 = [7.2741e-05 0 -4.7702e-07; 0 7.9054e-05 0; -4.7702e-07 0 7.7253e-06];
I3 = [0.0002803 -5.0108e-08 -5.9791e-07; -5.0108e-08 0.0002854 -2.7438e-06; -5.9791e-07 -2.7438e-06 7.1594e-06];
I4 = [0.00106986 9.51e-10 -3.5562e-07; 9.51e-10 0.00106878 -9.8366e-07; -3.5562e-07 -9.8366e-07 1.0545e-05];
I5 = [0.0039523 4.3437e-06 0.00028748; 4.3437e-06 0.00394623 7.3231e-05; 0.00028748 7.3231e-05 0.00013233];

```

Figura 2.35. Datos dinámicos de los eslabones.

Autores: Eduardo Morillo y Juan Portero.

Una vez obtenidos los datos se procede a ingresarlos en el **SerialLink** que es el archivo  **pierna** previamente guardado como archivo *.mat* en el momento en que se realizó la cinemática del robot, así se lo puede abrir y editar cuando sea necesario, este archivo se lo abre de la siguiente manera.

```
>> load('pierna.mat')
```

Entonces aparece esta variable en el Workspace, al dar doble clic sobre la misma se muestran todas sus propiedades, para poder ingresar los datos de cada eslabón se tiene que abrir la sección **Links** en la cual están los cuatro eslabones del robot y finalmente ingresar en cada uno de ellos, como se muestra en la Figura 2.36, para llenar los datos de la masa representado con la letra ***m***, el vector 1x3 del centro de gravedad representado por la letra ***r*** y la matriz 3x3 de inercia representada por la letra ***I***.

Property	Value	Min	Max
theta	0	0	0
d	0.0205	0.0205	0.0205
alpha	1.5708	1.5708	1.5708
a	0	0	0
sigma	0	0	0
mdh	0	0	0
offset	1.5708	1.5708	1.5708
name	[]		
m	0.6610	0.6610	0.6610
r	[0 -0.0140 0.0020]	-0.0140	0.0020
I	[2.2514e-06 0 0; 0 9.7771e-07 -1.952...]	-1.952...	3.1359...
Jm	0	0	0
B	0	0	0
Tc	[0 0]	0	0
G	0	0	0

Figura 2.36. Datos dinámicos en el SerialLink.

Autores: Eduardo Morillo y Juan Portero.

Al completar el ingreso de todos los datos que requiere el Robotics Toolbox para realizar los cálculos de la dinámica del robot, se procede a verificar los mismos con la siguiente función.

```
>> pierna.dyn()
```

Finalmente se obtiene la resolución del problema dinámico inverso, el cual expresa las fuerzas y torques que intervienen en las articulaciones en función de la evolución de las coordenadas articulares y sus derivadas, esto es posible mediante la siguiente función.

```
>> Q = pierna.rne([0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0])
```

En la posición inicial con los ángulos [0,0,0,0,0], con velocidad [0,0,0,0,0] y aceleración [0,0,0,0,0] se obtienen los valores de torques que necesitan entregar los servomotores para poder mantener la estructura de pie ante la gravedad y su propio peso, mostrando los siguientes resultados, los valores se muestran en Newton por metro [Nm].

```
Q =
```

```
0    0.2318    0.2318    0.2331    0.0215
```

Cuando la pierna está totalmente estirada hacia el frente como se muestra en la Figura 2.37 es cuando se entrega el mayor torque, al comparar con los resultados obtenidos en la posición inicial.

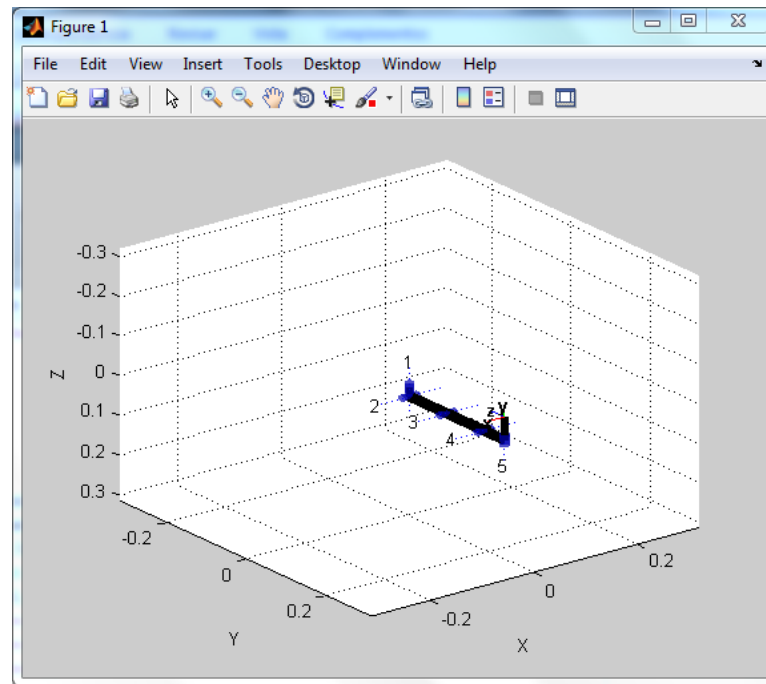


Figura 2.37. Posición donde se requiere el mayor torque.

Autores: Eduardo Morillo y Juan Portero.

Para esta posición los cálculos son los siguientes.

```
>> Q = pierna.rne([0,-pi/2,0,0,0], [0,0,0,0,0], [0,0,0,0,0])
```

Y los resultados se muestran de la siguiente manera.

```
Q =
-0.0000    0.2892   -0.0044   -0.2331   -0.0000
```

La interpretación de estos datos se da de la siguiente manera, de izquierda a derecha los valores de los torque articulares van desde q_1 hasta q_5 , es decir, $q_1 = 0$ [Nm], $q_2 = 0.2892$ [Nm], $q_3 = -0.0044$ [Nm], $q_4 = -0.2331$ [Nm] y $q_5 = 0$ [Nm].

Por lo tanto se concluye que los servomotores que se deben seleccionar deberán entregar un torque de por lo menos $0.2892 \text{ [Nm]} \approx 0.3 \text{ [Nm]}$ o superior para que el robot bípedo pueda soportar toda la carga y se mantenga en equilibrio.

2.3. DISEÑO DEL SISTEMA ELECTRÓNICO.

En base al funcionamiento descrito en el literal 2.1.3 se presenta a continuación en la Figura 2.38 un diagrama que contiene los elementos necesarios para cumplir con dicho funcionamiento, se indica cómo va a estar estructurada la etapa electrónica del prototipo de robot bípedo.

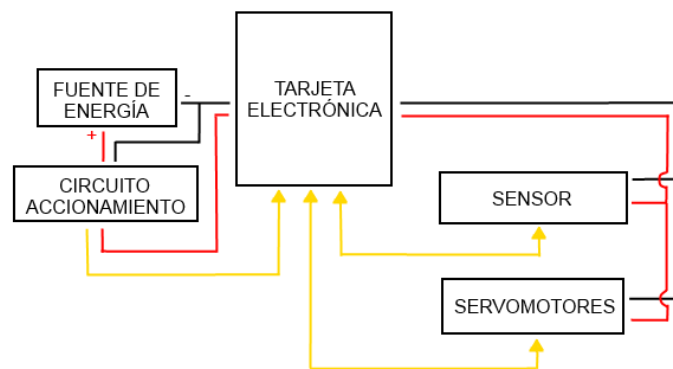


Figura 2.38. Diagrama de los componentes electrónicos.

Autores: Eduardo Morillo y Juan Portero.

Dónde:

- Línea Roja: Voltaje Positivo.
- Línea negra: Tierra
- Línea Amarilla: Línea de comunicación

A continuación se detalla cada uno de los bloques de la Figura 2.38

2.3.1. CIRCUITO DE ACCIONAMIENTO.

Este circuito está diseñado en el programa ISIS de Proteus, se indica en la Figura 2.39, consta de un switch al cual se le conectará el positivo de la

fuente de energía y al momento de accionar el mismo se energizará la tarjeta electrónica, además consta de un botón el cual está conectado a una resistencia pull-down (envía un cero lógico hasta que se presione el botón, en este caso enviará un 1 lógico), este botón servirá para indicar al robot que inicie con la caminata.

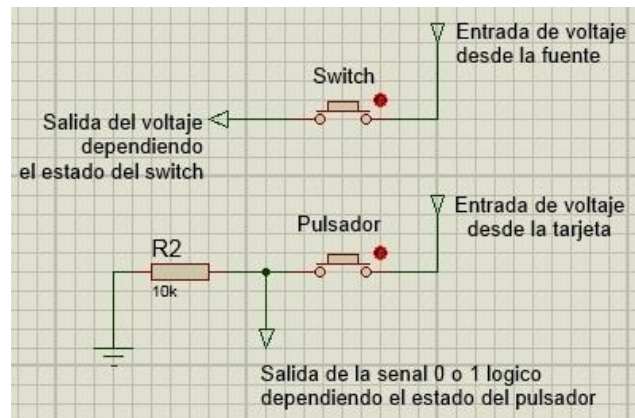


Figura 2.39. Circuito de accionamiento en ISIS.

Autores: Eduardo Morillo y Juan Portero.

A través de ARES de Proteus se transforma este circuito a PCB, se muestra en la Figura 2.40 el diseño a ser impreso para posteriormente pasarlo a baquelita.

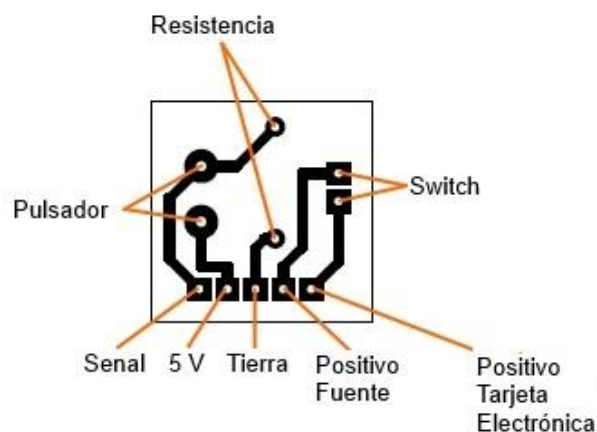


Figura 2.40. Circuito de accionamiento en ARES.

Autores: Eduardo Morillo y Juan Portero.

Gracias a la tecnología disponible en el mercado se va a seleccionar los demás componentes electrónicos, ya que existen muchas tarjetas, servomotores y sensores que se pueden utilizar para este proyecto, a continuación se definen los requerimientos, funciones y características que debe cumplir cada uno de éstos en base a los parámetros de diseño electrónico y de comunicación.

2.3.2. SERVOMOTORES.

Las funciones principales a cumplir son:

- Proporcionar el máximo torque calculado en el apartado de la dinámica que es de 0.3 [Nm] o uno mayor para que el funcionamiento del robot sea el adecuado.
- Que sus dimensiones se ajusten adecuadamente a la estructura diseñada en SolidWorks mediante el servomotor genérico.

Los servomotores deberán cumplir con las siguientes características:

- **Dimensiones:** Sus dimensiones deberán coincidir con las del servomotor genérico usado en el diseño mecánico dentro del programa SolidWorks para establecer las dimensiones de la estructura mecánica del robot.
- **Voltaje de Alimentación:** Deberán tener un voltaje de alimentación de 12 [V] que es el máximo establecido en los parámetros de diseño electrónicos o menor a este valor.
- **Temperatura de funcionamiento:** Permitirán su funcionamiento en un rango de temperatura posible de 5 hasta los 85 [°C].
- **Peso:** Debido al promedio realizado para establecer el peso de los servomotores en los cálculos realizados en el diseño mecánico el peso óptimo que deberán tener es de 55 [gr] para cumplir de manera adecuada con los cálculos realizados.

- **Comunicación:** Será la misma tecnología de comunicación que utiliza la tarjeta electrónica y los sensores para que no exista ningún problema de comunicación durante el envío y la recepción de datos.
- **Monitoreo de Variables:** Constarán de sensores internos o de un circuito de control interno que permita enviar los valores de posición, velocidad, torque y temperatura hacia la tarjeta para realizar el control de su funcionamiento.

2.3.3. SENSOR.

Para cumplir con el funcionamiento determinado en los parámetros de diseño donde se indica que el robot debe detectar un obstáculo frente a él se hace necesaria la utilización de un sensor que permita la detección de los mismos; para lo cual este debe contar con las siguientes características:

- **Comunicación:** Deberá ser compatible con la tecnología de comunicación utilizada por la tarjeta electrónica controladora.
- **Circuito de acondicionamiento:** No constará de ningún circuito de acondicionamiento para su conexión con la tarjeta pues se necesita minimizar el espacio a ocupar y el peso.
- **Voltaje de Alimentación:** Tendrá un voltaje de alimentación de 5 [V] o un voltaje igual al proporcionado por la tarjeta; caso contrario deberá permitir su alimentación mediante voltajes que pueden ser suministrados mediante reguladores de voltaje encontrados en el mercado.
- **Consumo de corriente:** El consumo de corriente deberá ser bajo debido a que la mayoría de las tarjetas controladoras usadas para los prototipos en robótica constan con un suministro de corriente bajo.
- **Distancia de sensado:** Permitirá sensar un objeto frente al robot a una distancia aproximada de 40 [cm].

- **Tamaño:** Deberá ser pequeño debido a que se montará sobre la cabeza del robot.

Las opciones a utilizar para cumplir con el funcionamiento que se espera de este sensor del robot y las características detalladas anteriormente son: sensores de ultrasonido, infrarrojos y radares.

2.3.4. TARJETA ELECTRÓNICA.

La tarjeta tiene la capacidad de manejar apropiadamente todos los recursos del robot, con esto se deben tener en cuenta las siguientes características:

- **Recursos:** Debe poseer variedad de interfaces para la comunicación con los distintos dispositivos que deberá interactuar, debe permitir el control 12 servomotores o más debido a los grados de libertad con los que cuenta el robot y debe contar con comunicación alámbrica e inalámbrica con el computador para cumplir con el parámetro de diseño referente a la comunicación.
- **Voltaje de Alimentación:** Debido a que el “cerebro” de la tarjeta será un microprocesador pudiendo ser éste un AVR o un PIC de alta gama, el voltaje de alimentación será de 5 [V] como es el caso de la mayoría de estos dispositivos.
- **Velocidad de Procesamiento:** Que brinde altas velocidades que permitan una correcta ejecución del programa y la realización de las acciones de control respectivas.
- **Campo de Aplicaciones:** Debe contar con aplicaciones orientadas hacia la robótica o el desarrollo de prototipos robóticos.
- **Método de Programación:** Capacidad de implementar programas de código abierto para el desarrollo del algoritmo de control.

- **Flexibilidad:** Posibilidad de estar sujeta a futuras mejoras en cuanto a su capacidad de integración de nuevos componentes; también debe permitir el uso de fuentes externas para actuadores que ocupen corrientes y voltajes elevados que no podrán ser suministrados por la tarjeta. En este caso debería soportar una fuente externa de 12 [V] que es el voltaje máximo establecido en el parámetro de diseño electrónico.
- **Consumo de energía:** Debe consumir la mínima cantidad de energía posible, pero este consumo no deberá afectar su desempeño para que tenga un adecuado funcionamiento.
- **Temperatura de Funcionamiento:** Debido a que el robot no se encontrará en ningún ambiente hostil no es necesario tomar muchas precauciones al tener en cuenta este valor.
- **Entradas/Salidas:** Debería constar con pines para entradas y salidas análogas y digitales que faciliten la implementación de sensores, actuadores y diferentes dispositivos que puedan ser utilizados para el funcionamiento del robot.

2.3.5. DISPOSITIVOS DE COMUNICACIÓN.

Los requerimientos que se necesita que cumplan son:

- Que permitan la comunicación alámbrica e inalámbrica entre la computadora y la tarjeta.
- Deberán permitir el libre tráfico de los datos enviados y recibidos por los servomotores, sensores y el sistema de control.

A partir de estos requerimientos proceden las siguientes características:

- **Frecuencia de Trabajo:** Debe ser libre, porque no se trabaja en grandes distancias en donde se hace necesario la regulación de frecuencias.

- **Flexibilidad:** Que se pueda conectar a otros tipos de redes.
- **Facilidad de conexión:** Debe ser un estándar que implementen el resto de equipos que lleva el robot.

2.3.6. FUENTE DE ENERGÍA.

El aspecto importante a tener en cuenta en este caso es que debe tener la capacidad de entregar el suministro de voltaje y corriente para que los servomotores, sensores, tarjeta electrónica y dispositivos de comunicación del robot funcionen adecuadamente en el tiempo requerido, además debe proporcionar los valores de voltaje y corriente propicios durante el tiempo que dura una prueba de funcionamiento ya que la fuente de energía si no es fija varia en sus valores durante el tiempo de descarga, por lo cual las características con las que debería contar son:

- **Portabilidad:** La fuente de energía deberá ser fácilmente almacenable y transportable, la mejor opción es una batería a diferencia de las pilas que no son recargables.
- **Relación Espacio/Peso:** Deberá ser liviana y ocupar el menor espacio de ser posible, se deberá seleccionar entre las opciones estandarizadas del mercado.
- **Tiempo de Duración:** Permitirá suministrar energía durante 30 [min] de uso promedio. Este tiempo de descarga se lo calcula dividiendo la capacidad de la batería por el consumo de los elementos electrónicos.
- **Precio:** El costo de la adquisición debe ser módico pues el proyecto a desarrollar es un prototipo por lo cual no se requiere de equipos muy sofisticados ni de un elevado costo.
- **Suministro de Corriente:** Debe ser capaz de suministrar la corriente necesaria que permitirá el correcto funcionamiento de todo el sistema electrónico del robot (servomotores, sensores y la tarjeta). Hay que

recordar que mientras el voltaje aportado por la fuente es idealmente constante, la intensidad aportada, en cambio, varía en cada momento según lo haga el consumo eléctrico del circuito al que se lo conecte (Torrente, 2013, p. 21).



Al concluir con el diseño electrónico, la definición de los equipos que se usarán en el robot se realizará mediante un método de elección en primer lugar del tipo de tecnología más conveniente, para luego proceder a elegir el dispositivo específico, esto se realizará en el punto 2.5 de la selección de elementos y materiales.

2.4. DISEÑO DEL SISTEMA DE CONTROL.

Antes de empezar con el diseño del sistema de control se realizará la simulación del modelado del robot bípedo para verificar por simulación su comportamiento, se lo hace utilizando los paquetes informáticos SimMechanics, Simulink y Robotics Toolbox que son complementos del software Matlab ya que esta es una herramienta poderosa en este campo de la ingeniería.

2.4.1. MODELAMIENTO DEL ROBOT BÍPEDO.

Como primer paso hay que descargar el software necesario el cual permite enlazar el SolidWorks con el entorno Matlab, este software se lo puede descargar desde MathWorks en la sección de productos y servicios, SimMechanics y SimMechanics Link, hay que tomar en cuenta que para descargar el SimMechanics Link este debe corresponder a la versión de Matlab instalada en el computador para que no haya errores al momento de la instalación.

Se descargan los archivos  `install_addon.m` y un archivo comprimido  `smlink.r2012b.win64.zip` que no se debe descomprimir, después se debe ingresar a Matlab como administrador y ubicarse en la carpeta que contenga estos dos archivos, a continuación se debe ejecutar el siguiente comando.

```
>> install_addon('smlink.r2012b.win64.zip')
```

Después de unos segundos aparece un mensaje informando que ha finalizado la instalación, seguido se ingresa la siguiente instrucción que permite activar SimMechanics en SolidWorks.

```
>> smlink_linksw
```

Ahora para activar el complemento en SolidWorks hay que ingresar a Herramientas, Complementos y activar como indica la Figura 2.41, hay que tener en cuenta que la opción SimMechanics Link esta activada solo para ensamblajes de SolidWorks mas no para piezas.

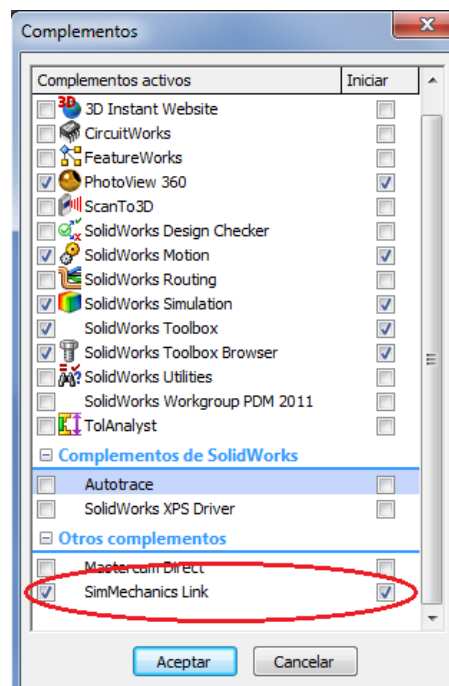


Figura 2.41. Enlace entre SimMechanics y SolidWorks.

Autores: Eduardo Morillo y Juan Portero.

Para evitar que se creen más bloques al momento de llevar el robot a SimMechanics, se ha editado el ensamblaje de la Figura 2.8 en un nuevo ensamblaje mostrado en la Figura 2.43 en el cual las piezas que conforman a cada eslabón son una sola y no esten separadas como en el ensamblaje anterior, un ejemplo se muestra en la Figura 2.42.

Con este ensamblaje se trabajará todo el diseño del sistema de control debido a que anteriormente se especificó que el control va a estar concentrado en la parte inferior del prototipo de robot bípedo.

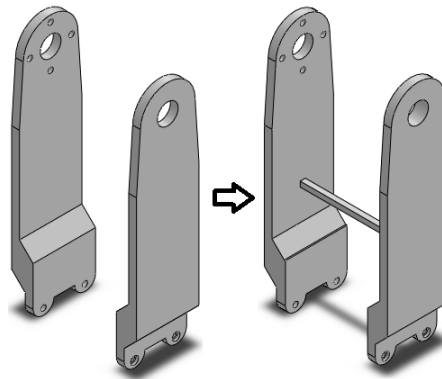


Figura 2.42. Piezas del segundo ensamblaje.

Autores: Eduardo Morillo y Juan Portero.

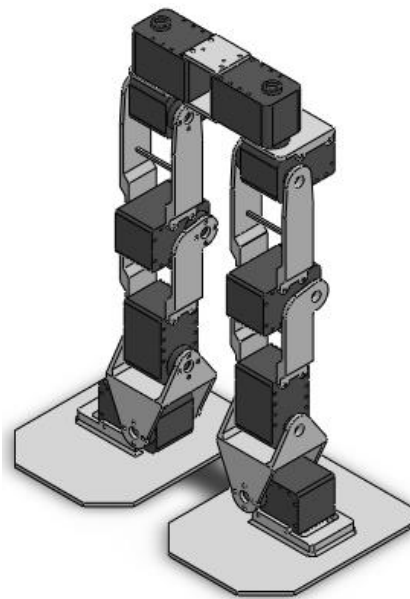



Figura 2.43. Segundo ensamblaje de las extremidades inferiores.

Autores: Eduardo Morillo y Juan Portero.

A este ensamblaje se lo exporta hacia Matlab como indica la Figura 2.44 y se crea el siguiente archivo:  Ensamblaje_MB.xml

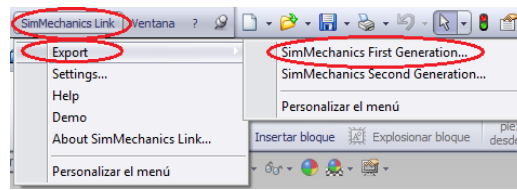


Figura 2.44. Exportación del ensamblaje desde SolidWorks a Matlab.

Autores: Eduardo Morillo y Juan Portero.

Para importar este archivo a Matlab se utiliza el siguiente comando, el cual crea el modelado del diseño mediante diagrama de bloques en el entorno de Simulink.

```
>> mech_import('Ensamblaje_MB.xml')
```

Una vez creado el modelado mostrado en la Figura 2.45, se procede a ordenar los bloques, para una mejor presentación se crean subsistemas (para las articulaciones) esto se hace seleccionando varios elementos y tipiendo el comando *Ctrl+G*. Para agregar una imagen al subsistema se tiene que seleccionar el mismo y con el comando *Ctrl+M* se edita su máscara, a continuación se ingresa el siguiente código `image(imread('AX-12.jpg'))`, la foto AX-12.jpg (imagen del servomotor) tiene que estar en la misma carpeta en la que se guarda todo el modelado para que no se presente un error.

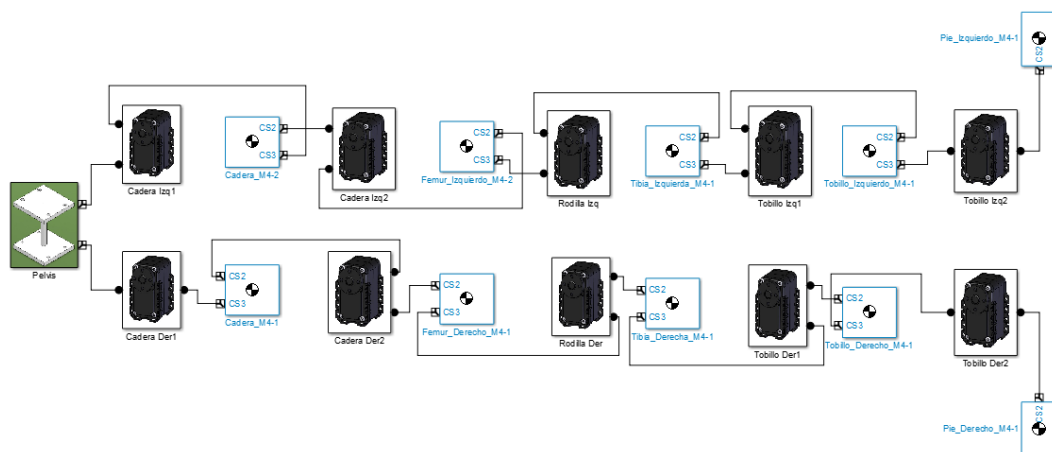


Figura 2.45. Modelado del robot bípedo en SimMechanics.

Autores: Eduardo Morillo y Juan Portero.

Dentro de cada subsistema como el que se muestra en la Figura 2.46 están los bloques de unión (weld), del cuerpo del servomotor (body) y de la articulación (revolute).

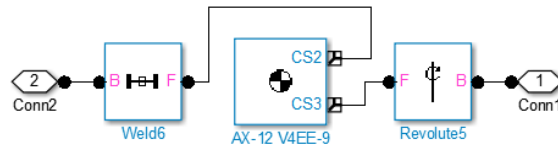


Figura 2.46. Subsistemas del modelado SimMechanics.

Autores: Eduardo Morillo y Juan Portero.

Algo importante que hay que recalcar es que cada uno de los bloques de las partes del robot, tienen incluido su masa (Mass), matriz de inercia (Inertia) y el centro de gravedad (CG), estos datos se muestran en la Figura 2.47, gracias a que se importan directamente estos parámetros desde el SolidWorks, no hace falta crear cuerpos nuevos (Body) y agregar toda esta información.

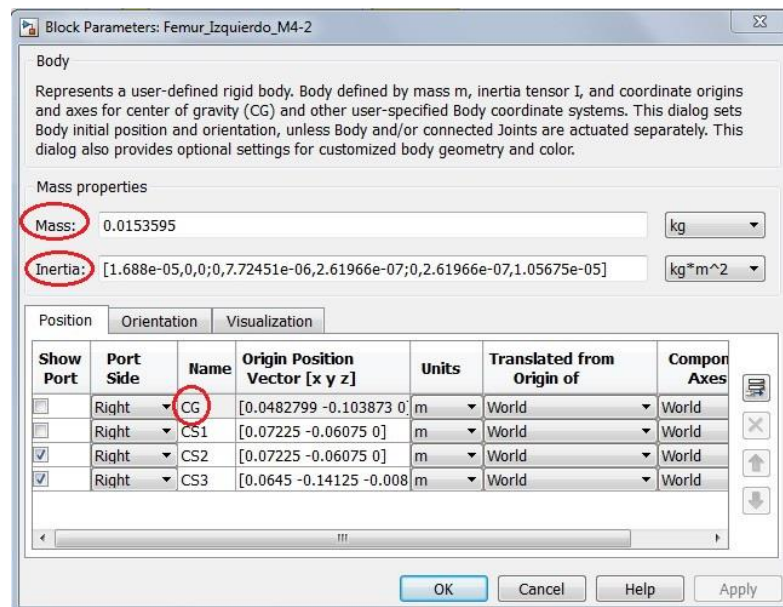


Figura 2.47. Parámetros dentro de los bloques Body de SimMechanics.

Autores: Eduardo Morillo y Juan Portero.

En el diagrama se pueden agregar nuevos bloques, los de SimMechanics se unen sin problemas entre ellos pero los bloques de las diferentes bibliotecas de Simulink tienen que unirse por medio de los bloques de sensores y actuadores mostrados en la Figura 2.48.

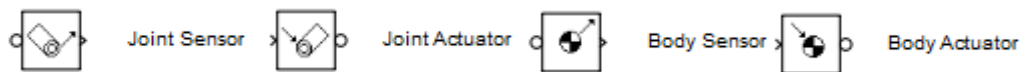


Figura 2.48. Sensores y actuadores de SimMechanics.

Autores: Eduardo Morillo y Juan Portero.

2.4.2. TRAYECTORIAS DE MOVIMIENTO DEL ROBOT BÍPEDO.

Tomando en cuenta lo citado en el literal 1.2.4 donde se hace un estudio breve sobre la locomoción del ser humano, se busca que el prototipo de robot bípedo realice los mismos movimientos que realiza una persona, al menos simulado porque en forma física se tienen otras trayectorias debido al tamaño y forma del robot además de la respuesta de los actuadores y todo lo que conlleva la gran diferencia entre la fisionomía del ser humano con la de este prototipo de robot bípedo.

Las articulaciones a tomar en cuenta para definir las trayectorias son la cadera, rodilla y tobillo desde la vista del plano sagital. En base a revisión de documentos como (Hernández, 2008, pp. 40-45) y (López, 2012, p. 39) se han obtenido las trayectorias que realiza una persona promedio y se muestran en las siguientes Figuras 2.49, 2.50 y 2.51; se observa los ángulos que cumple cada articulación en la fase del caminado y se muestra una gráfica de los mismos con respecto al porcentaje del ciclo.

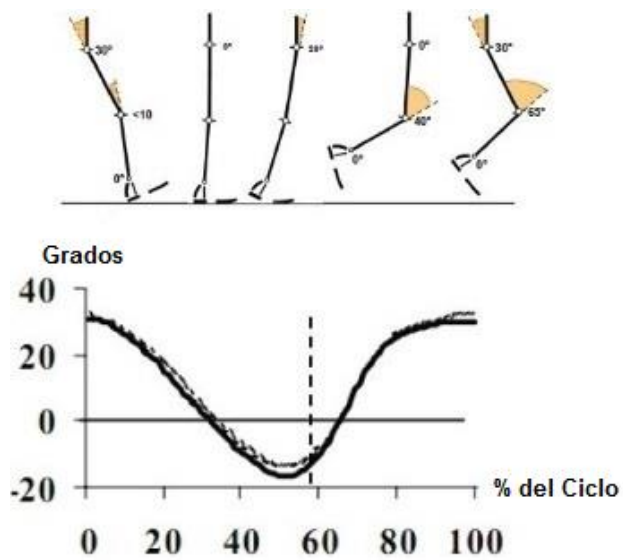


Figura 2.49. Trayectoria de la cadera.

Autores: Eduardo Morillo y Juan Portero.

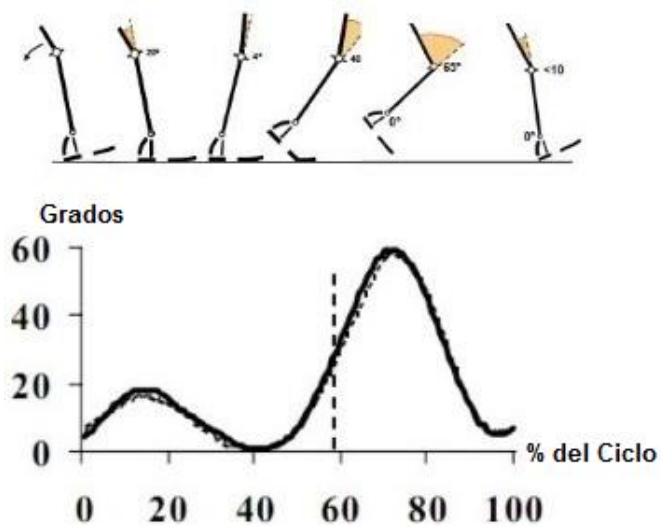


Figura 2.50. Trayectoria de la rodilla.

Autores: Eduardo Morillo y Juan Portero.

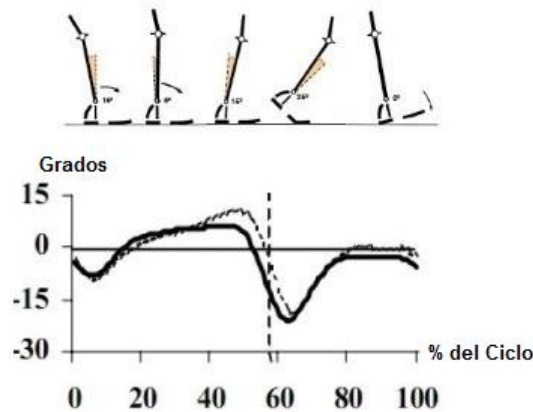


Figura 2.51. Trayectoria del tobillo.

Autores: Eduardo Morillo y Juan Portero.

Una vez definidas las trayectorias de las articulaciones se procede a llevarlas al entorno de Matlab, para esto se crean los archivos *.m* respectivos y se ingresan los datos como se indica en la Figura 2.52 para la cadera, Figura 2.53 para la rodilla y Figura 2.54 para el tobillo. Para crear la trayectoria de cada articulación se debe ingresar el punto inicial y punto final además de un intervalo para que el comando *jtraj* cree cada tramo, este comando crea un polinomio de quinto orden para suavizar la trayectoria de la posición, velocidad y aceleración (Corke, 2011, p. 43), *jtraj* solo funciona cuando se utiliza el Robotics Toolbox, así que hay que ejecutarlo previamente.

```

TrayectoriaCadera.m x
1   %Trayectoria Cadera
2   -   pc1=30;                               %Tramo 1
3   -   pc2=-20;
4   -   tc1=(0:1:50);                         %0a50
5   -   dc1=jtraj(pc1,pc2,tc1);
6   -   plot(dc1);
7   -   pc3=-20;                               %Tramo 2
8   -   pc4=30;
9   -   tc2=(0:1:50);                         %50a100
10  -   dc2=jtraj(pc3,pc4,tc2);
11  -   plot(dc2);

```

Figura 2.52. Trayectoria de la cadera en Robotics Toolbox.

Autores: Eduardo Morillo y Juan Portero.

```

TrayectoriaRodilla.m x
1 %Trayectoria Rodilla
2 - pr1=5; %Tramo 1
3 - pr2=20;
4 - tr1=(0:1:12); %0a12
5 - dr1=jtraj(pr1,pr2,tr1);
6 - plot(dr1);
7 - pr3=20; %Tramo 2
8 - pr4=0;
9 - tr2=(0:1:28); %12a40
10 - dr2=jtraj(pr3,pr4,tr2);
11 - plot(dr2);
12 - pr5=0; %Tramo 3
13 - pr6=60;
14 - tr3=(0:1:30); %40a70
15 - dr3=jtraj(pr5,pr6,tr3);
16 - plot(dr3);
17 - pr7=60; %Tramo 4
18 - pr8=5;
19 - tr4=(0:1:30); %70a100
20 - dr4=jtraj(pr7,pr8,tr4);
21 - plot(dr4);

```

Figura 2.53. Trayectoria de la rodilla en Robotics Toolbox.

Autores: Eduardo Morillo y Juan Portero.

```

TrayectoriaTobillo.m x
1 %Trayectoria Tobillo
2 - pt1=0; %Tramo 1
3 - pt2=-10;
4 - tt1=(0:1:7); %0a7
5 - dt1=jtraj(pt1,pt2,tt1);
6 - plot(dt1);
7 - pt3=-10; %Tramo 2
8 - pt4=5;
9 - tt2=(0:1:17); %7a24
10 - dt2=jtraj(pt3,pt4,tt2);
11 - plot(dt2);
12 - pt5=5; %Tramo 3
13 - pt6=15;
14 - tt3=(0:1:22); %24a46
15 - dt3=jtraj(pt5,pt6,tt3);
16 - plot(dt3);
17 - pt7=15; %Tramo 4
18 - pt8=-20;
19 - tt4=(0:1:19); %46a65
20 - dt4=jtraj(pt7,pt8,tt4);
21 - plot(dt4);

```

Continúa 

```

22 - pt9=-20; %Tramo 5
23 - pt10=0;
24 - tt5=(0:1:15); %65a80
25 - dt5=jtraj(pt9,pt10,tt5);
26 - plot(dt5);
27 - pt11=0; %Tramo 6
28 - pt12=0;
29 - tt6=(0:1:20); %80a100
30 - dt6=jtraj(pt11,pt12,tt6);
31 - plot(dt6);

```

Figura 2.54. Trayectoria del tobillo en Robotics Toolbox.

Autores: Eduardo Morillo y Juan Portero.

Al ejecutar cada uno de estos archivos *.m* se crean los archivos *.mat* que contienen toda la trayectoria de cada articulación, estos archivos son los que se van a utilizar en el SimMechanics para darles las trayectorias a los servomotores.

En la gráfica de la Figura 2.55 se observa la trayectoria de la cadera en color rojo, la trayectoria de la rodilla en color azul y la trayectoria del tobillo en color negro y al comparar esta gráfica con las Figuras 2.49 – 2.51 se observa que son idénticas y con esto concluye la obtención de las trayectorias de las articulaciones.

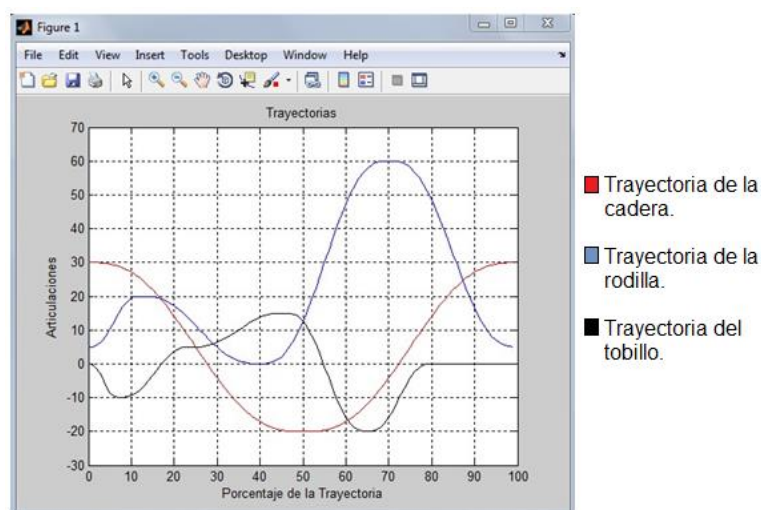


Figura 2.55. Trayectorias de las articulaciones.

Autores: Eduardo Morillo y Juan Portero.

A continuación se procede a editar cada subsistema del modelado del robot bípedo para que pueda leer los datos de cada articulación y así tener control sobre la misma, el nuevo diagrama queda como indica la Figura 2.56.

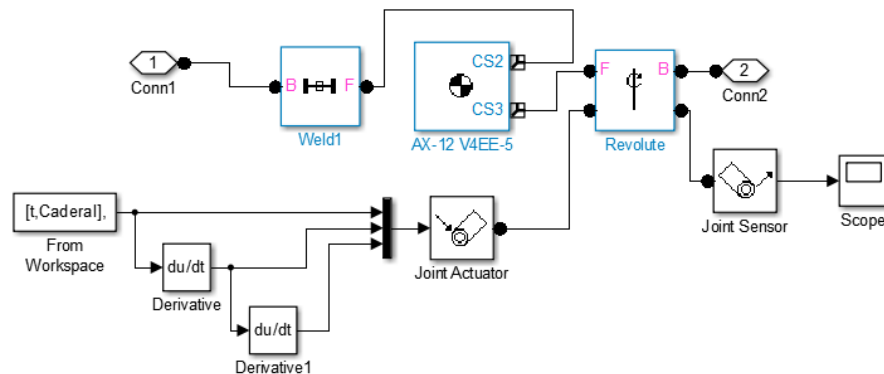
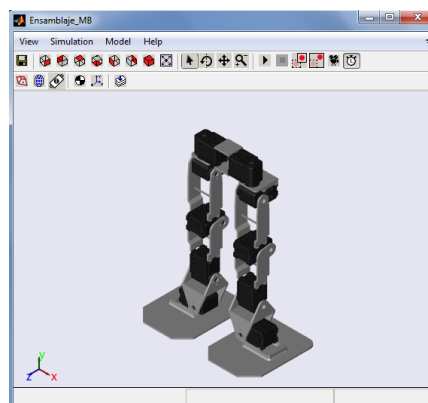


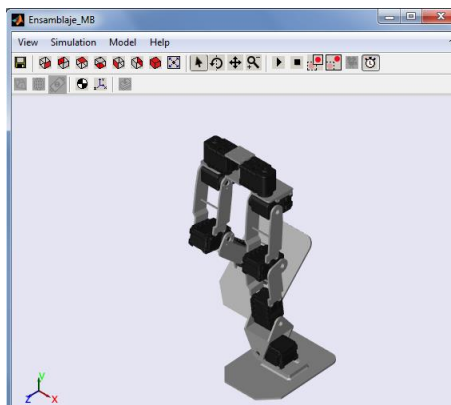
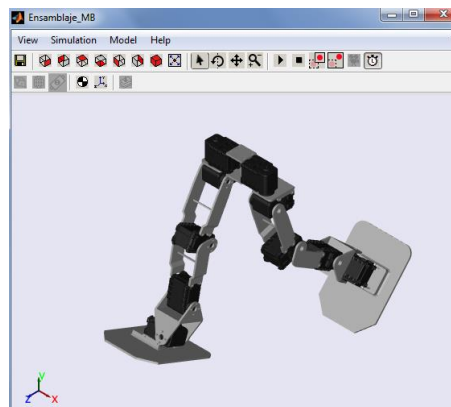
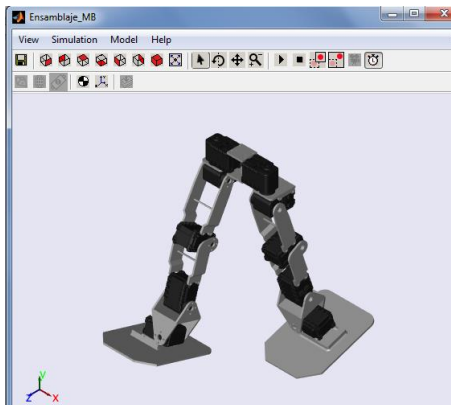
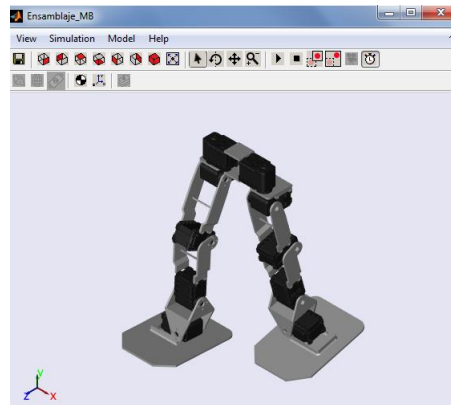
Figura 2.56. Subsistemas con la trayectoria implementada.

Autores: Eduardo Morillo y Juan Portero.

En la Figura 2.56 se observa algunos nuevos bloques, el bloque **From Workspace** que permite leer los datos de los archivos **.mat** que son la posición y al derivarlo se obtiene la velocidad y a su vez su derivada es la aceleración que se proporciona a la articulación, todos estos datos se unifican con un **mux** y se envían por medio de un **joint actuator** a la articulación de igual manera se ha ingresado un **scope** para observar cómo se cumple la trayectoria en el periodo de tiempo de ejecución.

Una vez ingresadas todas las trayectorias a todas las articulaciones se procede a la simulación, en la Figura 2.57 se indica la secuencia del caminado del robot.





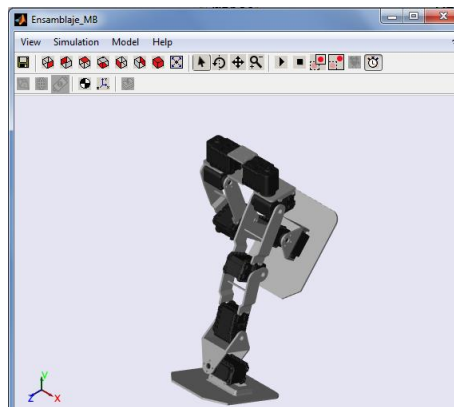


Figura 2.57. Secuencia de caminado del robot bípedo en SimMechanics.

Autores: Eduardo Morillo y Juan Portero.

Hay que recordar que a la primera articulación de la cadera y la segunda del tobillo se les da un valor de cero porque esas articulaciones realizan movimiento en otros planos.

Se ha realizado la simulación para que el robot bípedo realice 6 pasos, los resultados de las trayectorias se observan en los gráficos de los **scope** y se muestran en las Figuras 2.58, 2.59 y 2.60 para cada articulación respectivamente.

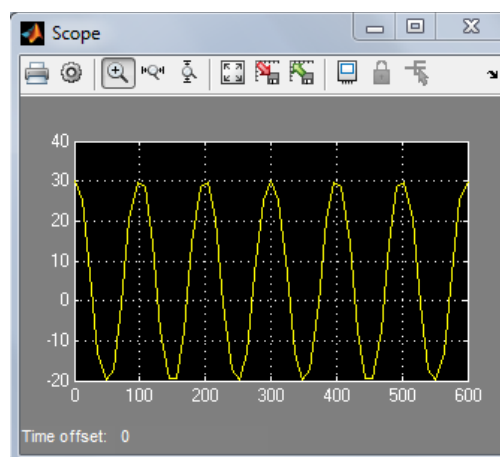


Figura 2.58. Trayectoria de la cadera (Simulink).

Autores: Eduardo Morillo y Juan Portero.

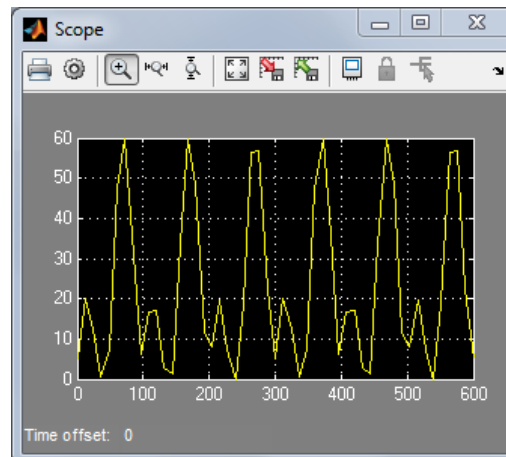


Figura 2.59. Trayectoria de la rodilla (Simulink).

Autores: Eduardo Morillo y Juan Portero.

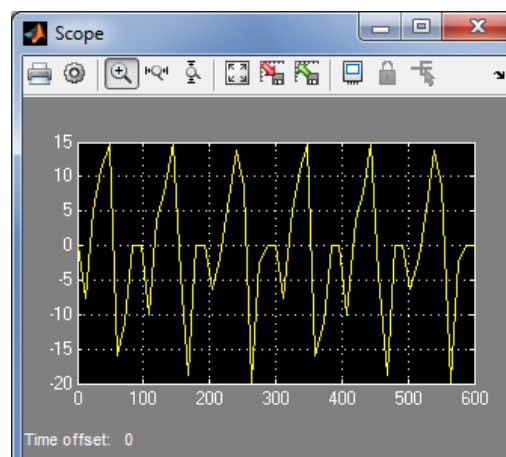


Figura 2.60. Trayectoria del tobillo (Simulink).

Autores: Eduardo Morillo y Juan Portero.

Al comparar las trayectorias obtenidas en las Figuras 2.58, 2.59 y 2.60 durante un ciclo que va de 0 a 100% del ciclo del caminado, con la Figura 2.55, se observa que son las mismas trayectorias, con lo que se concluye que el diseño realizado es óptimo pues todas las articulaciones cumplen con las trayectorias especificadas sin presentar ningún impedimento de tipo estructural, terminando así el estudio de los movimientos del robot bípedo en el entorno de Matlab.

2.4.3. TÉCNICA DE CONTROL.

La técnica que se va aplicar para el sistema de control es el método de la heurística, debido a que se contará con las trayectorias que realizarán cada una de las articulaciones o grados de libertad del robot bípedo, entonces esta técnica será la encargada de guiar al robot mientras éste se desplaza o realiza su marcha sobre la superficie plana, esta técnica es la encargada de tomar decisiones frente a los distintos obstáculos que se encuentren frente al robot para realizar una acción coherente para que el robot pueda continuar su marcha o girar hacia su derecha frente al obstáculo mientras aun encuentre uno frente a él. Otra característica de la heurística es que mediante esta toma decisiones siempre se elegirá aquella que más convenga para la solución de los problemas pues aquellas que no brinda una solución adecuada son descartadas mientras que las más utilizadas son guardadas para tenerlas como soluciones posibles para futuros problemas.

Las trayectorias que se implementarán en el robot bípedo (parte superior y parte inferior) serán generadas mediante el programa PyPose, pues se debe tener en cuenta que las trayectorias que son tomadas para la simulación de la parte inferior en Simulink son las del ser humano y que la simulación fue realizada en condiciones ideales, mientras que las que serán generadas en PyPose son influenciadas por factores como el peso de la parte superior, la gravedad que afecta a toda la estructura, la fricción de los engranajes de los servomotores y la fricción del contacto de los pies con el piso por lo cual éstas no serán iguales a las tomadas para la simulación.

a. DIAGRAMA DE FLUJO DEL SISTEMA DE CONTROL.

En la Figura 2.61 se indica el diagrama de flujo del funcionamiento del sistema de control del robot bípedo. Mediante el cambio de esta del switch el robot se encenderá y se pondrá en una posición inicial (erguido), con el accionamiento del botón el robot procede a iniciar su caminata, este continuará con la misma mientras no detecte un obstáculo frente a él, al detectar un obstáculo éste detendrá su marcha y evitará el obstáculo

mediante el giro a su derecha, al evitar el obstáculo este emitirá una señal de confirmación que permitirá que el robot continúe caminado hacia el frente mientras no exista otro obstáculo frente al robot, se da la finalización de la marcha al cambiar de estado el switch que conjuntamente con el botón estarán ubicados en la parte posterior de la cabeza del robot bípedo.

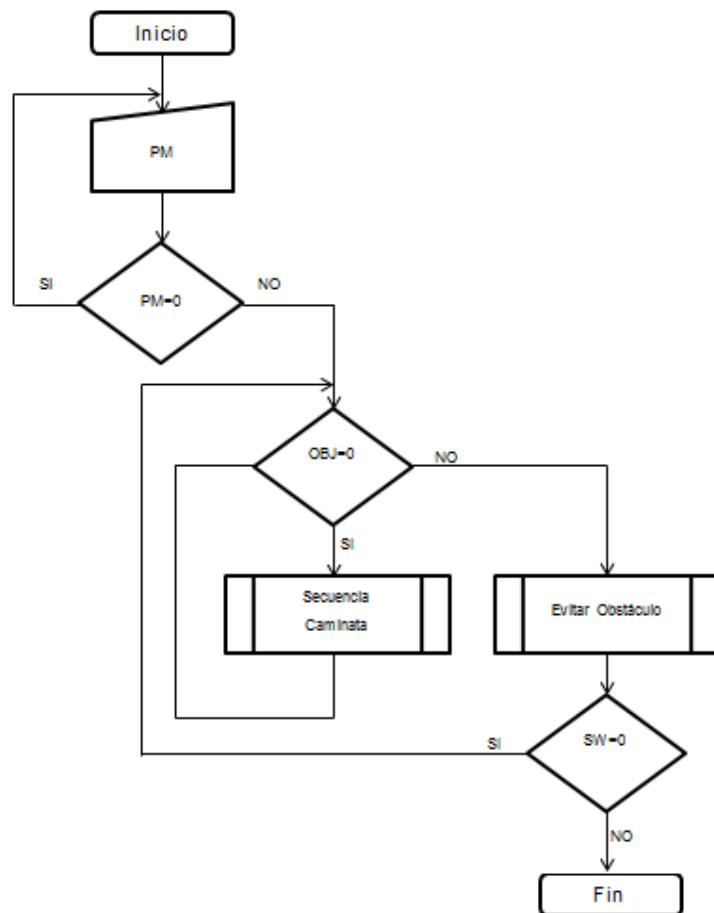


Figura 2.61. Diagrama de flujo del sistema de control.

Autores: Eduardo Morillo y Juan Portero.

Dónde:

- **PM:** Es la variable de la puesta en marcha del robot.
- **OBJ:** Es la variable del sensor que permitirá saber si existe un obstáculo o no frente al robot.

- **SW:** Es una variable que permitirá seguir con la caminata mientras no se cambie de estado el switch para terminar la marcha.
- **Secuencia Caminata:** Es un subprograma donde están las trayectorias para la caminata y donde se procede a verificar si el sensor detecta un obstáculo o no.
- **Evitar obstáculo:** Es un subprograma donde se encuentra la secuencia que realiza el robot para evitar el obstáculo aquí la variable SW y la variable OBJ regresan a sus valores iniciales para continuar con la caminata.

2.5. SELECCIÓN DE ELEMENTOS Y MATERIALES.

En este apartado se seleccionarán los dispositivos que cumplen con el diseño electrónico del prototipo de robot bípedo analizado anteriormente.

2.5.1. SERVOMOTORES.

La selección de los servomotores se realizará mediante la Tabla comparativa 2.3, en la cual se detallarán las características mencionadas en el correspondiente apartado del diseño electrónico, se verificará cuál de los servomotores detallados a continuación cumple en su totalidad con los requerimientos que se desea o aquel que se asemeje más a lo necesitado para el correcto funcionamiento del robot bípedo.

Tabla 2.3. Características de los servomotores.

	Dynamixel AX-12	Dynamixel MX-28T	Kondo KRS-788HV	Hitec HS-422
Torque	1.5 [Nm]	2.5 [Nm]	0.98 [Nm]	0.4 [Nm]
Dimensiones	32x50x40 [mm]	35.6x50.6x35.5 [mm]	41x35x21 [mm]	40.6x19.8x36.6 [mm]
Voltaje de Alimentación	12 [V]	12 [V]	9 a 12 [V]	4.8 a 6 [V]
Temperatura de Funcionamiento	-5 a 85 [°C]	-5 a 85 [°C]		-20 a 60 [°C]

Continúa 

Peso	55 [gr]	72 [gr]	47.5 [gr]	45.5 [gr]
Comunicación	TTL Half Duplex Async Serial	TTL Asynchronous Serial	Analog Parallel Network	RS-422
Monitoreo de Variables	Constan de un circuito integrado para monitorear velocidad, posición, temperatura y torque.	Constan de un circuito integrado para monitorear velocidad, posición, temperatura y torque.	No consta de ningún dispositivo para esa función.	No consta de ningún dispositivo para esa función.

Fuentes: <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>,
<http://www.trossenrobotics.com/dynamixel-mx-28-robot-actuator.aspx>,
<http://www.roboteshop.com/robot-catalogue/kondo/servos-motors/krs-788hv.html> y
<http://www.superrobotica.com/S330165.htm>.

A continuación se realizará una lista citando el valor deseado en cada característica y el nombre de los diferentes servomotores que cumplen con aquel valor y cuyo nombre constan en la Tabla 2.3.

- **Torque:** Mediante la Dinámica del robot se concluyó que el valor necesario para los actuadores de las articulaciones es de 0.3 [Nm] o uno mayor, como se observa en la Tabla 2.3 todos los servomotores cumplirían con esta característica.
- **Dimensiones:** Los servomotores genéricos diseñados anteriormente tienen las siguientes dimensiones 32x50x40 [mm], al fijarse en los valores de la Tabla 2.3 se puede apreciar que el servomotor Dynamixel AX-12A es el único que cumple con esta característica.
- **Voltaje de Alimentación:** Se busca un voltaje de alimentación de 12 [V], al observa la Tabla 2.3 se nota que los Servomotores Dynamixel AX-12A, Dynamixel MX-28T y Kondo KRS-788HV cumplen con el valor mencionado.

- **Temperatura de Funcionamiento:** Los servomotores deberán tener un rango de 5 hasta los 85 [°C], en la Tabla 2.3 se puede apreciar claramente que el servomotor Dynamixel AX-12A y Dynamixel MX-28T constan con este rango de valores buscados.
- **Peso:** El valor establecido es de 55 [gr], al comparar este valor en la Tabla 2.3 se denota que el servomotor Dynamixel AX-12A tiene el peso requerido.
- **Monitoreo de Variables:** Se mencionó que el servomotor debe permitir monitorear la velocidad, posición, torque y temperatura; mediante la mención anterior de la característica y con la apreciación de la tabla se concluye que los servomotores Dynamixel AX-12A y Dynamixel MX-28T cumplen con la misma.

El servomotor que ha sido mencionado en todas las características y por ende cumple con las mismas es el servomotor Dynamixel AX-12A mostrado en la Figura 2.62, por lo tanto éste será seleccionado para el desarrollo del proyecto. La Tabla 2.4 indica las especificaciones del mismo.



Figura 2.62. Servomotor Dynamixel AX-12A

Fuente: <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

Tabla 2.4. Especificaciones del servomotor Dynamixel AX-12A.

Especificaciones Dynamixel AX-12A	
Voltaje de operación	12 [V]
Torque Máximo	1.53 [Nm]
Velocidad sin carga	59 [rpm]
Peso	55 [gr]
Resolución	0.29°
Corriente Máxima	900 [mA]
Protocolo	TTL Half Duplex Async Serial
Temperatura de Operación	-5 a 85 [°C]

Fuente: <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

2.5.2. SENSOR.

El sensor encargado de detectar los obstáculos frente al prototipo de robot bípedo, se seleccionará por medio de la Tabla 2.5 con las diferentes características y los respectivos valores pertenecientes a cada sensor que se podría utilizar para este fin.

Tabla 2.5. Características de los sensores.

	Parallax PING (Sensor de Ultrasonido)	SRF04 (Sensor de Ultrasonido)	SHARP GP2Y0A02 (Sensor Infrarrojo)
Dimensiones	22x46x16 [mm]	20x43x17 [mm]	19X40X22 [mm]
Peso	9 [gr]	10 [gr]	5 [gr]
Voltaje de alimentación	5 [V]	5 [V]	5.5 [V]
Consumo de Corriente	35 [mA]	50 [mA]	50 [mA]
Distancia de Sensado	De 2 [cm] a 3 [m]	De 3 [cm] a 3 [m]	De 20 [cm] a 1.5 [m]
Tipo de comunicación	Bidireccional TTL	Bidireccional TTL	I2C
Circuito de acondicionamiento	No necesita de acondicionamiento.	Necesita de un circuito de	Necesita de un circuito de

Continúa 

Fuentes: <http://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf>,
<http://www.superrobotica.com/S320110.htm>,
<http://www.superrobotica.com/S320107.htm>,
<http://www.amazon.es/GP2Y0A02YK-Sharp-ranger-sensor-20150cm/dp/B00GYTPORK#productDetails>,
<http://repositorio.espe.edu.ec/bitstream/21000/4005/1/T-ESPEL-0505.pdf>.

Al analizar la Tabla 2.5 se eligió el sensor Parallax PING Ultrasonic Range, que debido a sus características resulta el sensor indicado para el desarrollo del robot bípedo debido a que cumple con los requerimientos del diseño electrónico y además cumple con el objetivo principal que es la detección de obstáculos frente al robot. La Figura 2.63 hace referencia al sensor seleccionado, además se presenta la Tabla 2.6 que contiene las especificaciones técnicas del mismo.



Figura 2.63. Sensor Parallax PING.

Fuente: <http://www.trossenrobotics.com/parallax-ping-ultrasonic-range-sensor.aspx>

Tabla 2.6. Especificaciones del sensor Parallax PING.

Especificaciones Parallax PING	
Rango de medición	De 2 [cm] hasta los 3 [m]
Suministro de voltaje	5 [V] \pm 10%
Suministro de Corriente	30 [mA] típica, 35 [mA] máxima
Tamaño	22 [mm] de alto x 46 [mm] ancho x 16 [mm] espesor
Entrada de disparador	Pulso TTL positivo, 2 [μ s min], 5 [μ s] típica.
Echo Pulse	Pulso TTL positivo, 115 [μ s] a 18,5 [μ s]
Echo Hold	750 [μ s] de caída del pulso de disparo
Burst Frequency	40 [kHz] para 200 [μ s]

Fuente: <http://www.trossenrobotics.com/parallax-ping-ultrasonic-range-sensor.aspx>

2.5.3. TARJETA ELECTRÓNICA.

Siendo la tarjeta electrónica controladora un elemento de vital importancia en su selección debido a que su objetivo principal es el de controlar todos los recursos del robot bípedo, se ha realizado la Tabla 2.7 la cual constará con las principales y más relevantes características a ser analizadas o comparadas entre las diferentes tarjetas que son utilizadas en aplicaciones robóticas y por ende opciones posibles a elegir.

Tabla 2.7. Características de las tarjetas electrónicas.

	Arduino Uno SMD	Arduino Mega	ArbotiX-M Robocontroller
Recursos	1 puerto serie UART TTL (5 [V]). 1 puerto virtual para comunicación serial con el computador. Soporta la comunicación I2C (TWI) y SPI. Permite el control de hasta 10 servomotores PWM.	4 puertos serie UART TTL (5 [V]). Soporta la comunicación I2C (TWI) y SPI. Permite el control de hasta 10 servomotores PWM.	2 puertos seriales, 1 dedicado para el control de los servomotores Dynamixel y el otro dedicado para XBee o a su vez FTDI. Comunicación I2C y SPI. Permite el control de hasta 18 servomotores seriales.
Voltaje de alimentación	7 – 12 [V]	7 – 12 [V]	6 – 16 [V]
Voltaje de Operación	5 [V]	5 [V]	5 [V]
Velocidad de procesamiento	16 [MHz]	16 [MHz]	16 [MHz]
Campo de aplicaciones	Utilizada para el desarrollo de proyectos de todo tipo exceptuando la industria.	Utilizada para el desarrollo de proyectos grandes de todo tipo exceptuando la industria.	Orientada al control de robot (legos y prototipos).
Método de programación	Arduino IDE	Arduino IDE	Arduino IDE

Continúa 

Flexibilidad	Permite la integración de una fuente externa.	Permite la integración de una fuente externa.	Permite la integración de dos fuentes externas.
Entradas y Salidas	14 entradas o salidas digitales de las cuales 6 proveen una para de PWM y 6 entradas analógicas.	54 entradas o salidas digitales de las cuales 14 proveen una para de PWM y 16 entradas analógicas.	28 entradas o salidas digitales de la cuales 8 pueden funcionar como entradas analógicas.

Fuentes: <http://arduino.cc/en/MainarduinoBoardUno>

<http://arduino.cc/es/Main/ArduinoBoardMega>

<http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx>

<http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

La tarjeta seleccionada es la Arbotix-M Robocontroller en base a la comparación de la Tabla 2.7, donde se observa que se cumplen de manera cabal todos los requerimientos y características que se detallan en la parte del diseño electrónico referente a la tarjeta de electrónica. En la Figura 2.64 se muestra una imagen de la tarjeta y a continuación de la misma se muestra la Tabla 2.8 con sus especificaciones.

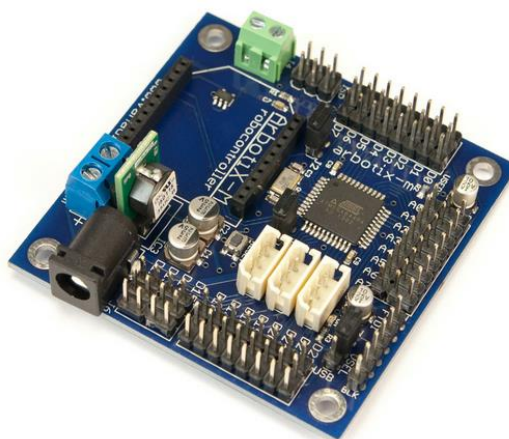


Figura 2.64. Tarjeta electrónica ArbotiX-M.

Fuente: <http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx>

Tabla 2.8. Especificaciones de la tarjeta electrónica ArbotiX-M.

Especificaciones ArbotiX-M	
Dimensiones	61 [mm] X 61 [mm]
Voltaje de alimentación	11 – 12 [V]
E / S digitales	28
Servomotores Permitidos	Hasta 18 servomotores con conectores de 3 pines (GND, V_{CC} , de la señal).
Puertos Serie	2, 1 dedicado al servo controlador y el otro a la programación XBEE/FTDI
Microcontrolador	AVR ATMEGA644p – 16MHz
Comunicación	I2C presente en los pines D16=SCL D17=SDA

Fuente: <http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx>

2.5.4. DISPOSITIVOS DE COMUNICACIÓN.

Para proceder con la selección de los dispositivos que se encargarán de la comunicación; tanto de los sensores y actuadores con la tarjeta, como de esta con el computador se ha procedido primero a detallar que comunicación tienen cada uno de los elementos ya seleccionados, facilitando de esta manera el establecer los diferentes dispositivos que se necesitarán.

De las Tablas 2.3, 2.5 y 2.7 se tiene que:

- Los servomotores tienen una comunicación TTL Half Duplex Async Serial.
- El sensor tiene una comunicación TTL bidireccional.
- La tarjeta cuenta con la comunicación serial para lo cual tiene 2 puertos seriales, 1 dedicado para el control de los servomotores Dynamixel y el otro dedicado para un XBee radio o para comunicación FTDI.

En base a lo anteriormente detallado se tiene que para la comunicación entre el computador y la tarjeta es necesario un dispositivo UartSBee y un XBee, el porqué de la selección de estos dispositivos es debido a que el

UartSBee permite la comunicación FTDI mediante un cable y también permite la comunicación inalámbrica mediante un XBee cumpliendo de esta manera con las tecnologías de comunicación que brinda la tarjeta seleccionada y con las características del diseño electrónico.

Para validar la elección de este dispositivo de comunicación se realiza la Tabla 2.9 en donde se comparan las características conjuntas que brinda este dispositivo con respecto a los dispositivos que lo brindarían de forma individual.

Tabla 2.9. Características de los dispositivos de comunicación.

	Cable USB FTDI	XBee Explorer USB	UartSBee V4 (USB-Xbee-TTL Interface)
Descripción	El cable FTDI es un conversor Serie-USB (TTL) que permite conectar dispositivos TTL por USB. El cable viene configurado para funcionar a 5 [V].	El XBee explorer USB permite conectar y utilizar cualquier módulo XBee directamente mediante un puerto USB conectado al computador admitiendo así la comunicación inalámbrica. Esta unidad funciona con todos los módulos XBee incluyendo la Serie S1 y la Serie 2.5, versión estándar y Pro.	UartSBee V4 es un dispositivo compacto con entrada USB que sirve como adaptador serial y viene equipado con un zócalo para XBee. Con FT232 integrado IC, que puede ser utilizado para la programación o la comunicación con las aplicaciones MCU.
Como se da la Comunicación	Este dispositivo se conecta al puerto USB del computador y el otro extremo se conecta hacia la tarjeta de acuerdo	Este dispositivo se conecta mediante un cable mini USB al PC y de esa forma se tiene acceso a los pines TX/RX del XBee y	Este dispositivo se conecta al computador mediante un cable mini USB, permite su alimentación y posterior

Continúa 

con los colores que con lo cual está funcionamiento.
 son establecidos listo para Quedando a libre
 para la comunicarse con el disposición del
 comunicación y Xbee de la tarjeta. usuario la
 alimentación. comunicación a
 utilizar ya sea
 mediante FTDI o un
 XBee.

Fuente: <https://www.sparkfun.com/products/8687>, <http://www.trossenrobotics.com/uartsbee>
<http://www.bricogeek.com/shop/391-cable-ftdi-5v.html>

Se concluye que la selección del UartSBee V4 mostrado en la Figura 2.65 es la óptima para el proyecto pues permite la comunicación alámbrica e inalámbrica de una forma adecuada pudiendo separar las mismas mediante un selector que alimenta a cada una de ellas. La Tabla 2.10 muestra las especificaciones del mismo.



Figura 2.65. UartSBee V4.

Fuente: <http://www.trossenrobotics.com/uartsbee>

Tabla 2.10. Especificaciones del UartSBee V4.

Especificaciones UartSBee V4	
Voltaje de alimentación	5 [V]
Doble Salida de Potencia	3,3 y 5 [V]
Protocolo	USB 2.0
Botón de Reset	para módulo XBee

Fuente: <http://www.trossenrobotics.com/uartsbee>

En las siguientes imágenes 2.66 y 2.67 se muestran los cables necesarios para la conexión de las diferentes comunicaciones.



Figura 2.66. Cable FTDI (comunica el UartSBee V4 con la tarjeta).

Fuente: <http://www.trossenrobotics.com/uartsbee>



Figura 2.67. Cables mini USB (comunica la PC con el UartSBee).

Fuente: <http://www.trossenrobotics.com/p/arbotix-robot-controller-starter-kit.aspx?relatedid=569>

El XBee seleccionado para trabajar en el UartSBee y la tarjeta electrónica es el XBee S1, se ha seleccionado teniendo en cuenta dos aspectos; el primero relacionado con la distancia de comunicación que es mínima y el segundo se debe a que el XBee Pro no es compatible con la tarjeta seleccionada. La imagen del XBee S1 se muestra en la Figura 2.68 seguida por sus especificaciones que constan en la Tabla 2.11.



Figura 2.68. XBee S1.

Fuente: <http://www.trossenrobotics.com/store/p/5200-XBee-1mW-Communication-Module.aspx>

Tabla 2.11. Especificaciones del XBee S1.

Especificaciones Xbee S1	
Voltaje de Alimentación	3.3 [V]
Suministro de Corriente	50 [mA]
Potencia de Salida	1 [mW]
Rango de Comunicación	3100 [m]
Pines de entrada ADC	6 de 10 [bit]
Pines de entrada/salida	8 digital
Comando para establecer o setear	AT o API

Fuente: <http://www.trossenrobotics.com/store/p/5200-XBee-1mW-Communication-Module.aspx>

En lo referente a la comunicación sensor - tarjeta y servomotores - tarjeta no habrá la necesidad de elegir ningún dispositivo debido a que la tarjeta ya cuenta con puertos dedicados para establecer estas comunicaciones y lo único que se necesitará es el uso de cables, teniendo en cuenta que tanto los servomotores y el sensor van montados sobre el robot, la comunicación inalámbrica entre ellos representaría un gasto innecesario.

2.5.5. FUENTE DE ENERGÍA.

Antes de iniciar con la selección de la fuente de energía se realizan los cálculos referentes al consumo de corriente de los servomotores y demás componentes ya seleccionados para establecer el suministro de corriente que debe entregar la batería o las baterías a utilizar.

Consumo de los servomotores:

$$M = 0,224 [N \cdot m]$$

$$\theta = 67^\circ \cdot \frac{\pi}{180^\circ} = 1,17[rad]$$

$$\omega = \frac{\theta}{t} = \frac{1,17[rad]}{0.169[s]} = 6.923 \left[\frac{rad}{s} \right]$$

$$P = M \cdot \omega = 0,224[N \cdot m] \cdot 6,923 \left[\frac{rad}{s} \right] = 1,55[W]$$

$$I = \frac{P}{V} = \frac{1,55[W]}{12[V]} = 0.129[A]$$

Al ser 12 servomotores el consumo es de 1.55 [A] a esto se debe sumar el consumo del sensor que es de 35 [mA] y el de la tarjeta que es de 55 [mA]; con lo cual se tendría un consumo total de 1.64 [A] requerido para el funcionamiento del robot.

En base a los parámetros requeridos en el diseño electrónico y al cálculo del consumo de corriente realizado, se ha elaborado la Tabla 2.12 limitándola al uso de baterías, teniendo en cuenta que un cable y el peso de una fuente propiamente dicha afectarían el funcionamiento del robot bípedo.

Tabla 2.12. Características de las baterías.

	Pack de baterías de Litio de Shenzhen Apollo	Pack de baterías NiMH Tenenergy	Pack de baterías Li-Po Shenzhen UFO
Voltaje Suministrado	12 [V]	12 [V]	12 [V]
Corriente Suministrada	5000 [mAh]	2000 [mAh]	4500 [mAh]
Tiempo de Duración	Depende de la carga a la que va a suministrar el voltaje y corriente.	Depende de la carga a la que va a suministrar el voltaje y corriente	Depende de la carga a la que va a suministrar el voltaje y corriente
Peso	275 [gr]	283,5 [gr]	50 [gr]
Dimensiones	38x55x68 [mm]	50x29x72 [mm]	66x135x35 [mm]
Precio	20 US Dólares	17.49 US Dólares	37 US Dólares

Fuente: <http://spanish.alibaba.com/goods/12v-lipo-battery.html>
<http://spanish.alibaba.com/p-detail/Bater%C3%ADas-de-litio-de-Shenzhen-12v-de-5000mAh-para-la-luz-de-emergencia-300000074353.html>

http://www.amazon.com/Tenergy-2000mAh-Battery-Leads-Airplanes/dp/B00408X4LU/ref=sr_1_fkmr0_3?ie=UTF8&qid=1377100282&sr=8-3-fkmr0&keywords=Ni-mh+12v+2.8a

Se ha elegido el pack de baterías NiMh (níquel-hidruro metálico) de 12 [V] y 2000 [mAh] de la marca Tenergy de la Figura 2.69, debido a que sus dimensiones se ajustan a las establecidas en la estructura del robot bípedo cumpliendo con la característica de la portabilidad, el voltaje y el consumo total de corriente necesarios para la fuente. Además su precio es asequible y finalmente para cumplir con todas las características se procede a calcular el tiempo de duración:

$$\text{tiempo de descarga} = \frac{\text{carga eléctrica de la batería}}{\text{consumo eléctrico de los dispositivos}} \quad \text{Ec.2.4}$$

$$\text{tiempo de descarga} = \frac{2000[\text{mA} \cdot \text{h}]}{1640[\text{mA}]}$$

$$\text{tiempo de descarga} = 1.22[\text{h}]$$

Como se observa el tiempo de duración es de 1.22 horas con lo cual se cumple el tiempo deseado de 30 [min] que era la característica planteada. El cargador de la misma se indica en la Figura 2.70.



Figura 2.69. Batería NiMH Tenergy.

Fuente: http://www.amazon.com/Tenergy-2000mAh-Battery-Leads-Airplanes/dp/B00408X4LU/ref=sr_1_fkmr0_3?ie=UTF8&qid=1377100282&sr=8-3-fkmr0&keywords=Ni-mh+12v+2.8a



Figura 2.70. Cargador universal inteligente Tenergy.

Fuente: http://www.amazon.com/Tenergy-Smart-Universal-Charger-Battery/dp/B001AVUAVC/ref=pd_bxgy_t_text_y

En la Tabla 2.13 se muestra las especificaciones del Pack de Baterías, seguido la Tabla 2.14 con las especificaciones de su cargador.

Tabla 2.13. Especificaciones de la batería Tenergy.

Especificaciones Baterías Tenergy	
Voltaje suministrado	12 [V]
Corriente suministrada	2000 [mAh]
Dimensiones	50 [mm] ancho; 29 [mm] altura; 72 [mm] espesor
Constituido	10 células AA de NiMH

Fuente: http://www.amazon.com/Tenergy-2000mAh-Battery-Leads-Airplanes/dp/B00408X4LU/ref=sr_1_fkmr0_3?ie=UTF8&qid=1377100282&sr=8-3-fkmr0&keywords=Ni-mh+12v+2.8a

Tabla 2.14. Especificaciones del Cargador Tenergy.

Especificaciones Cargador universal inteligente Tenergy	
Voltaje de Alimentación	120 – 240 [V]
Selector de Corriente	0.9 y 1.8 [A]
Fusible de Protección	para sobre corriente, cortocircuito y protección de polaridad inversa
Tiempo de Carga	1 a 2 Horas

Fuente: Fuente: http://www.amazon.com/Tenergy-Smart-Universal-Charger-Battery/dp/B001AVUAVC/ref=pd_bxgy_t_text_y

CAPÍTULO 3

CONSTRUCCIÓN E IMPLEMENTACIÓN

3.1. MONTAJE DEL SISTEMA MECÁNICO.

Al haber concluido con la fase de diseño del robot bípedo se procede a la construcción de las diferentes piezas que lo conforman, en esta sección se describen los procesos para la elaboración de las partes mecánicas que forman el robot, cada una de las piezas diseñadas en SolidWorks se realizan usando diagramas de operación de procesos, que permite detallar de forma clara las operaciones realizadas.

3.1.1. FABRICACIÓN DE LA ESTRUCTURA DEL ROBOT.

Las piezas de la estructura del robot bípedo son maquinadas en la impresora 3D Ultimaker, debido a que es el método más sencillo para trabajar con ABS que es el material que ha sido seleccionado para dicho fin. A continuación se detalla el diagrama de operación de procesos del fémur derecho externo de la Figura 3.1, este diagrama se generaliza para la elaboración de todas las partes del robot.

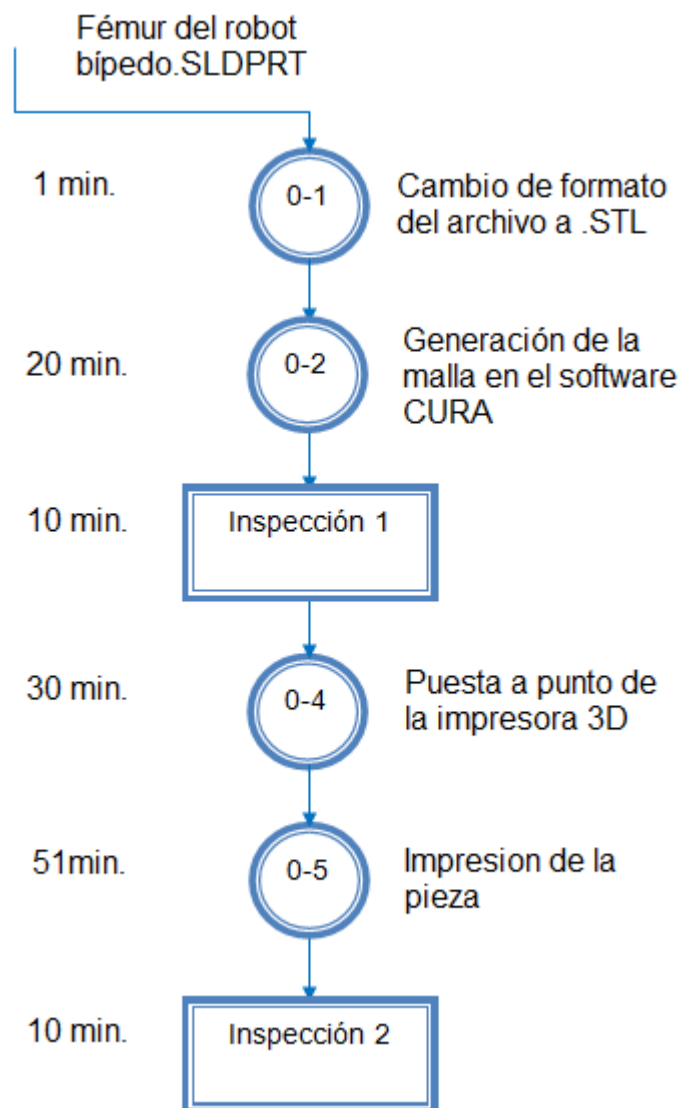


Figura 3.1. Diagrama de operación de procesos del fémur derecho externo.

Autores: Eduardo Morillo y Juan Portero.

La primera etapa del diagrama corresponde a cambiar el formato del archivo de la extensión *.SLDPRT* que es con la que se diseñó la pieza en SolidWorks a la extensión *.STL* que es con la que trabaja la impresora 3D, dentro del programa SolidWorks en la pestaña **archivo** se ingresa en **guardar como** y a continuación se selecciona **grabar en tipo** en la que se elige el formato *.STL*.

Este archivo es llevado al software de código abierto CURA con el cual se controla a la impresora 3D, el software permite dividir en capas al diseño, de esta manera la impresora sabe que capas tiene que imprimir.

En la primera inspección dentro del software CURA se verifica que la malla y la posición en la que se va a imprimir la pieza optimicen el tiempo y el material ocupado, este proceso se realiza dependiendo de la forma de la pieza.

Se prosigue con la puesta a punto de la impresora 3D en la cual hay que asegurarse que la boquilla de extrusión llegue a la temperatura óptima, este proceso tarda alrededor de 30 minutos.

Con los requerimientos previos realizados se procede a la impresión de la pieza en 3 dimensiones, este proceso varía dependiendo de la pieza a construir.

Finalmente impresa la pieza se realiza una última verificación de la calidad y medidas de la misma.

Finalizado el diagrama a continuación se muestra una tabla con los tiempos en los cuales son elaboradas cada una de las piezas debido a que resultaría de forma repetitiva realizar un diagrama para cada una de ellas. Los tiempos detallados en la Tabla 3.1 son los generados por el programa CURA para la impresión de cada parte del robot.

Los diferentes colores con los que se fabrican las piezas es una condición de apariencia y no representa ninguna diferencia dentro de sus propiedades mecánicas.

Tabla 3.1. Tiempos de impresión de las piezas del robot bípedo.

Nombre de la Pieza	Numero de Impresiones	Tiempo de la Impresión	Color
Fémur derecho externo	1	51 [min]	Azul
Fémur izquierdo externo	1	51 [min]	Azul
Fémur derecho interno	1	53 [min]	Azul
Fémur izquierdo interno	1	53 [min]	Azul
Tibia derecha externa	1	45 [min]	Azul
Tibia izquierda externa	1	46 [min]	Azul
Tibia derecha interna	1	45 [min]	Azul
Tibia izquierda interna	1	48 [min]	Azul
Tobillo derecho	1	140 [min]	Azul
Tobillo izquierdo	1	139 [min]	Azul
Pecho	1	335 [min]	Azul
Espalda	1	491 [min]	Azul
Cuello	1	142 [min]	Azul
Pie derecho	1	312 [min]	Negro
Pie izquierdo	1	311 [min]	Negro
Cadera derecha	1	49 [min]	Negro
Cadera izquierda	1	49 [min]	Negro
Cabeza	1	87 [min]	Negro
Pelvis abajo	1	20 [min]	Negro
Pelvis arriba	1	21 [min]	Negro
Brazo izquierdo	1	208 [min]	Negro
Brazo derecho	1	206 [min]	Negro
Tapas para los Brazos	2	156 [min]	Negro

Autores: Eduardo Morillo y Juan Portero.

3.1.2. ENSAMBLAJE DEL ROBOT BÍPEDO.

Con todas las piezas construidas y verificadas sus medidas y calidad, se procede a ensamblar el robot, para lo cual se unen las piezas y servomotores por medio de tuercas y tornillos que facilitan el ensamblaje y el des-ensamblaje para realizar tareas como su almacenamiento o mantenimiento. En adelante se muestra la forma en la que se ensamblan cada una de las partes que conforman el robot, terminando con el ensamble completo. Las piezas que sujetan los componentes y transmiten el

movimiento de los servomotores hacia las diferentes partes del robot (BU y WA) vienen junto con los servomotores. La Figura 3.2 muestra las piezas que vienen con cada servomotor, en el Anexo A se indican los planos.

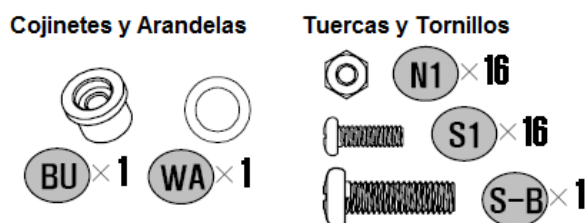


Figura 3.2. Elementos de sujeción de los servomotores.

Fuente: <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

a. ENSAMBLAJE DE LAS EXTREMIDADES INFERIORES.

Como procedimiento previo al ensamblaje se colocan en todos los servomotores las tuercas N1 que sujetan a los distintos eslabones que conforman el robot, cabe recalcar que la ubicación de estas tuercas es distinta por cada servomotor. Las Figura 3.3, 3.4, 3.5, 3.6, 3.7 y 3.8 muestran cómo van ubicadas las tuercas en cada uno de los servomotores.



Figura 3.3. Disposición de tuercas en los servomotores ID1 e ID2.

Autores: Eduardo Morillo y Juan Portero.

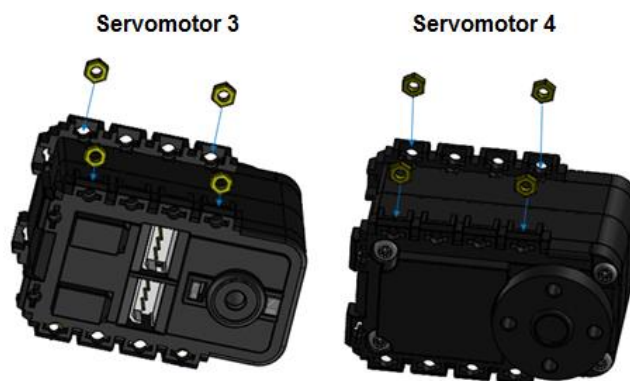


Figura 3.4. Disposición de tuercas en los servomotores ID3 e ID4.

Autores: Eduardo Morillo y Juan Portero.

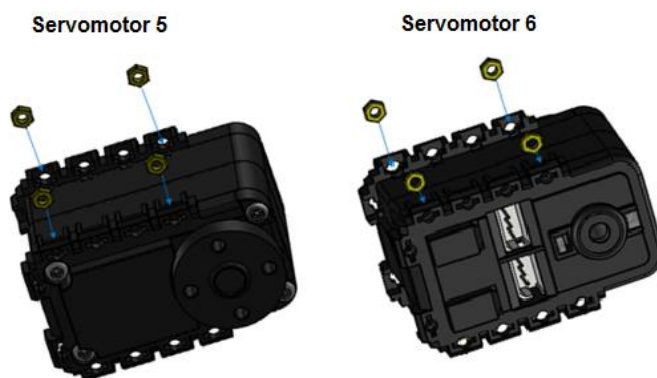


Figura 3.5. Disposición de tuercas en los servomotores ID5 e ID6.

Autores: Eduardo Morillo y Juan Portero.

Servomotor 7 y 8



Figura 3.6. Disposición de tuercas en los servomotores ID7 e ID8.

Autores: Eduardo Morillo y Juan Portero.

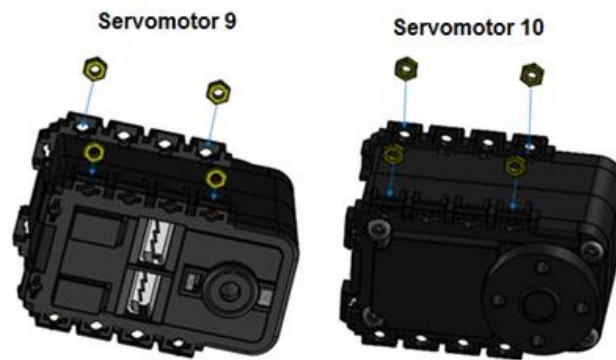


Figura 3.7. Disposición de tuercas en los servomotores ID9 e ID10.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.8. Disposición de tuercas en los servomotores ID11 e ID12.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.9. Inserción de las tuercas en el servomotor 1.

Autores: Eduardo Morillo y Juan Portero.

Una vez ubicadas las tuercas en cada servomotor se procede a ensamblar.

1. Se toma la pieza de la Figura 3.10 y se la coloca de manera que los orificios del servomotor 1 dispuestos para los tornillos, estén alineados correctamente; después se colocan los tornillos como se observa en la Figura 3.11 y ajustarlos; una vez ajustados los tornillos se obtiene el ensamblaje A como se muestra en la Figura 3.12.

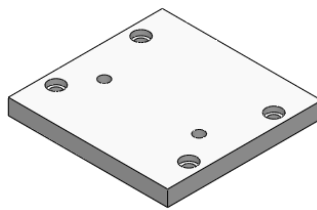


Figura 3.10. Cosis parte superior.

Autores: Eduardo Morillo y Juan Portero.

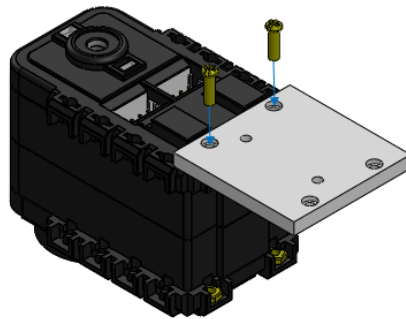


Figura 3.11. Disposición de los tornillos S1 en el servomotor 1.

Autores: Eduardo Morillo y Juan Portero.

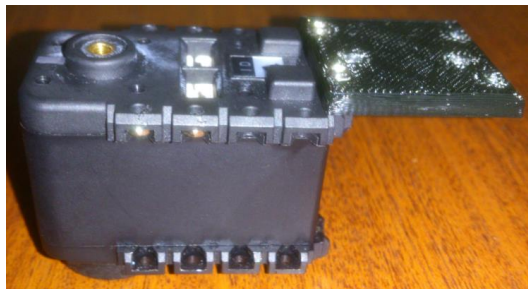


Figura 3.12 Ensamblaje A.

Autores: Eduardo Morillo y Juan Portero.

2. Se toma el ensamblaje A y el servomotor 2, el mismo que será dispuesto de forma que los tornillos ingresen libremente y por ende los orificios del ensamblaje estarán alineados con los del servomotor 2, este procedimiento se observa en la Figura 3.13. Al finalizar este literal se tendrá el ensamblaje B el mismo que se muestra en la Figura 3.14.

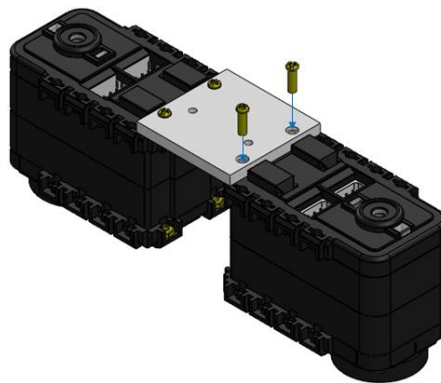


Figura 3.13. Ensamblaje A y tornillos S1 para forma el ensamblaje B.

Autores: Eduardo Morillo y Juan Portero.

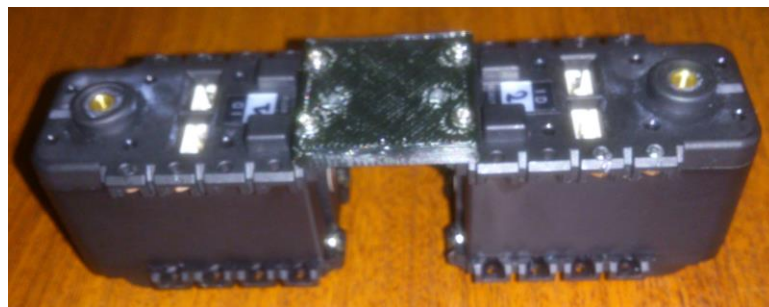


Figura 3.14. Ensamblaje B.

Autores: Eduardo Morillo y Juan Portero.

3. Se toma la pieza mostrada en la Figura 3.15 y se voltea al ensamblaje B, para que sea más sencillo colocarla y que así encaje con los agujeros de los servomotores 1 y 2, como se muestra en la Figura 3.16; en donde, además se muestra la forma correcta en la que se procede a colocar los tornillos y posteriormente se realiza su ajuste,

teniendo de esta manera el ensamblaje C que se observa en la Figura 3.17.

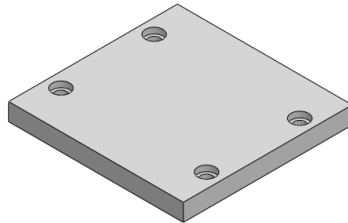


Figura 3.15. Caxis parte inferior.

Autores: Eduardo Morillo y Juan Portero.

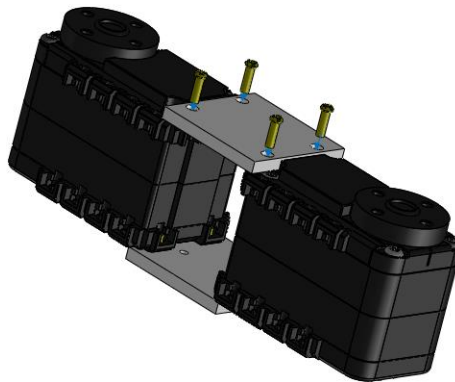


Figura 3.16. Ensamblaje B y tornillos para formar el ensamblaje C.

Autores: Eduardo Morillo y Juan Portero.

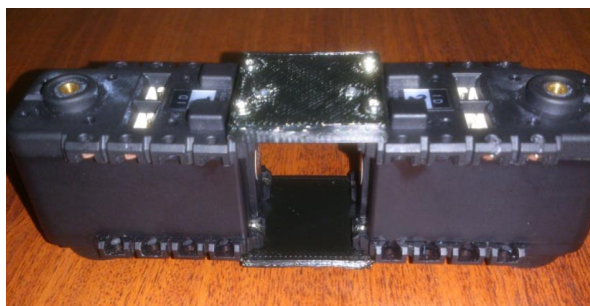


Figura 3.17. Ensamblaje C.

Autores: Eduardo Morillo y Juan Portero.

4. Se toman las piezas de la cadera mostradas en la Figura 3.18, correspondientes a la pierna derecha e izquierda, respectivamente. Después se las coloca en los servomotores 1 y 2 de la pierna derecha

y pierna izquierda; teniendo el ensamblaje C volteado para facilitar la colocación de los tornillos como se observa en la figura 3.19. Por último se procede a colocar los tornillos como se muestra en la Figura 3.20 y se los ajusta para formar el ensamblaje D que se presenta en la Figura 3.21.

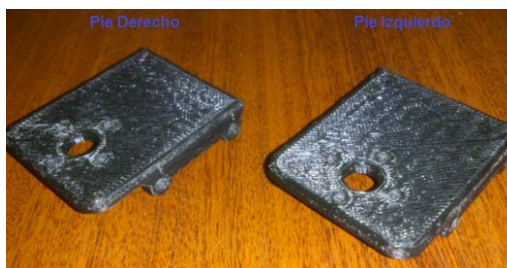


Figura 3.18. Cadera derecha e izquierda.

Autores: Eduardo Morillo y Juan Portero.

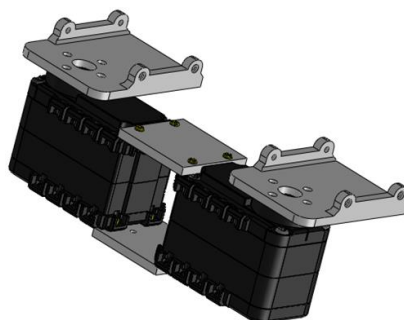


Figura 3.19. Ubicación piezas que conforman la cadera.

Autores: Eduardo Morillo y Juan Portero.

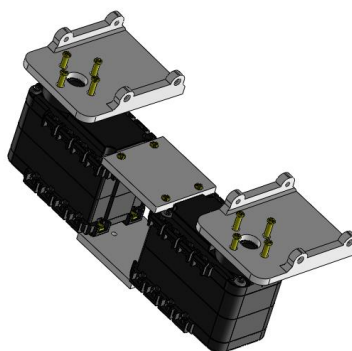


Figura 3.20. Disposición de los tornillos S1 en la cadera.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.21. Ensamblaje D.

Autores: Eduardo Morillo y Juan Portero.

5. En la Figura 3.22 se muestra el lugar donde van los tornillos S1 para ajustar los servomotores 3 y 4 que conforman el segundo grado de libertad de la cadera.

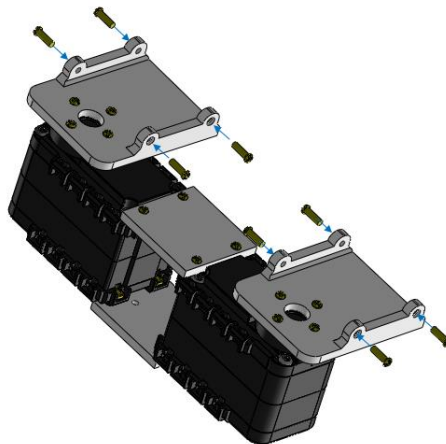


Figura 3.22. Disposición de los tornillos S1 que sujetan los servomotores 3 y 4.

Autores: Eduardo Morillo y Juan Portero.

6. La Figura 3.23 indica cómo se colocan los servomotores 3 y 4 para que puedan girar de forma correcta los ángulos necesarios que se requieren para el movimiento.

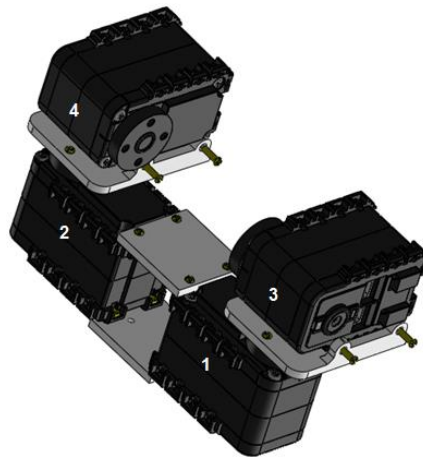


Figura 3.23. Ubicación servomotores 3 y 4.

Autores: Eduardo Morillo y Juan Portero.

- Se procede a ajustar los tornillos, teniendo de esta manera el ensamblaje E: el mismo que se representa en la Figura 3.24 en su posición normal debido a que en anteriores figuras se cambió su posición para facilitar el ensamblaje.

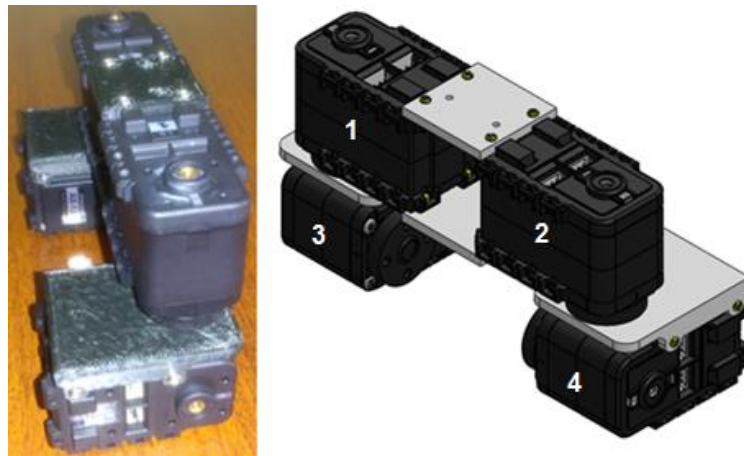


Figura 3.24. Ensamblaje E.

Autores: Eduardo Morillo y Juan Portero.

- Se giran los servomotores para obtener la posición mostrada en la Figura 3.25, que facilitará el ensamblaje de las demás partes del robot en los siguientes puntos.

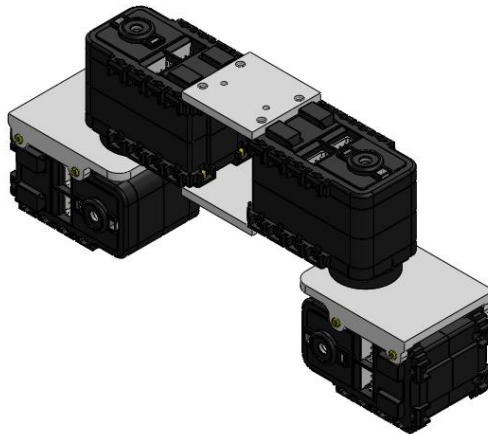


Figura 3.25. Ensamblaje E con el giro de los servomotores 3 y 4.

Autores: Eduardo Morillo y Juan Portero.

9. Se toman las piezas mostradas en la Figura 3.26, que son aquellas que conforman el fémur de la pierna derecha y la pierna izquierda, respectivamente.

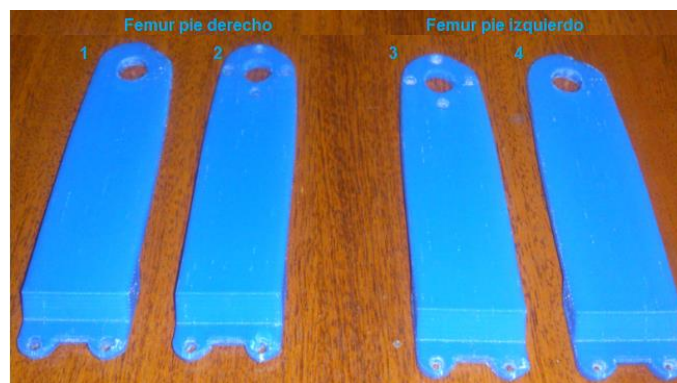


Figura 3.26. Piezas del fémur de la pierna derecha e izquierda.

Autores: Eduardo Morillo y Juan Portero.

10. Se colocan las piezas 1 y 4 de la Figura 3.26 como se muestra en las Figura 3.27 y 3.28, respectivamente; las mismas que muestran una vista simplificada de la pierna derecha e izquierda con lo que permiten observar de mejor manera cómo encaja cada una de las partes para formar en ensamblaje F de la Figura 3.29.

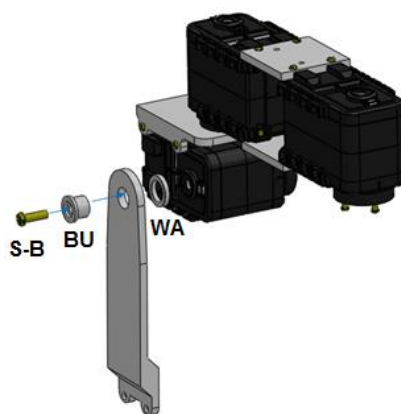


Figura 3.27. Ensamble del fémur derecho externo.

Autores: Eduardo Morillo y Juan Portero.

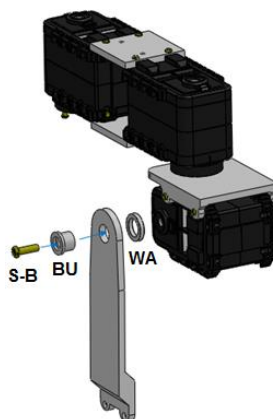


Figura 3.28. Ensamble del fémur izquierdo externo.

Autores: Eduardo Morillo y Juan Portero.

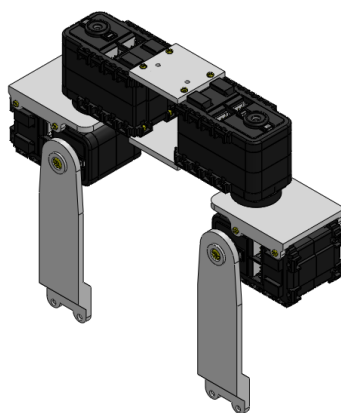


Figura 3.29. Ensamblaje F.

Autores: Eduardo Morillo y Juan Portero.

11. A continuación se debe girar el ensamble F y se procede a la colocación de las piezas 2 y 3 de la Figura 3.26 en la pierna derecha e izquierda, respectivamente: asegurándose que encajen los huecos de las mismas con la de los servomotores, para poner los tornillos S1 según las posiciones que indica la Figura 3.30.

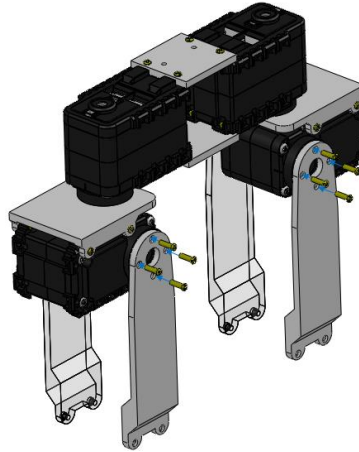


Figura 3.30. Disposición de las piezas y tornillos S1 para el ensamblaje G.

Autores: Eduardo Morillo y Juan Portero.

Teniendo como resultado el ensamblaje G, al regresar los servomotores a su posición normal (Figura 3.31).



Figura 3.31. Ensamblaje G.

Autores: Eduardo Morillo y Juan Portero.

12. Se colocan los servomotores de la forma que se muestra en la Figura 3.32 después se colocan los tornillos y ajustan los mismos en la posición donde se encuentran las tuercas que permiten el ajuste con los servomotores (Figura 3.33) para obtener el ensamble mostrado en la Figura 3.34.

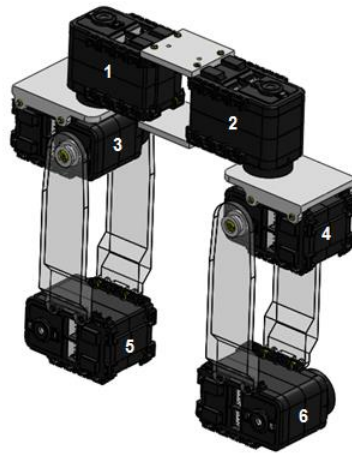


Figura 3.32. Ubicación de los servomotores 5 y 6.

Autores: Eduardo Morillo y Juan Portero.

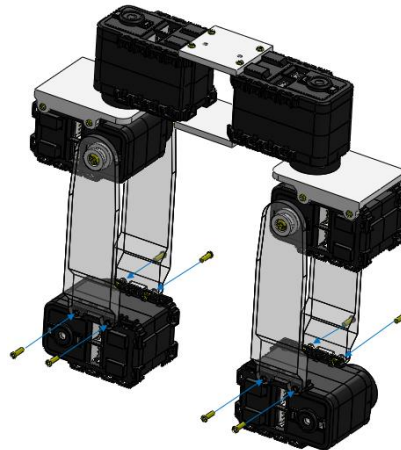


Figura 3.33. Disposición de los tornillos S1 para el ensamblaje H.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.34. Ensamblaje H.

Autores: Eduardo Morillo y Juan Portero.

13. Se toman las piezas referentes a la tibia del pie izquierdo y del pie derecho, como se muestra en la Figura 3.35.

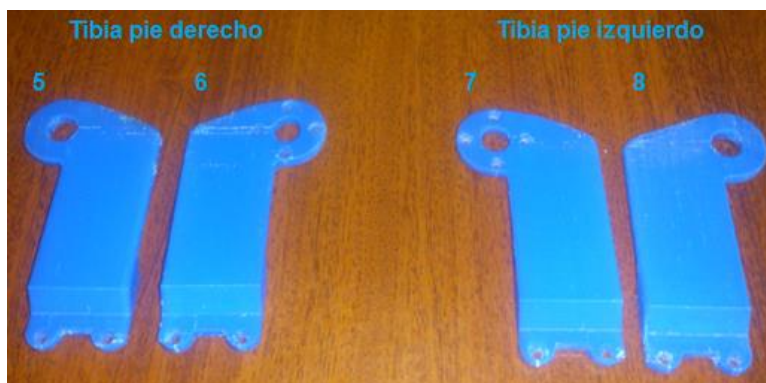


Figura 3.35. Piezas de la tibia de las piernas derecha e izquierda.

Autores: Eduardo Morillo y Juan Portero.

14. Se colocan las piezas 5 y 8 en la pierna derecha e izquierda respectivamente, este procedimiento se detalla en las Figuras 3.36 y 3.37 en donde se muestra de forma simplificada cada pierna y la forma de ubicar los demás componentes que en su totalidad conforman el ensamblaje I mostrado en la Figura 3.38.

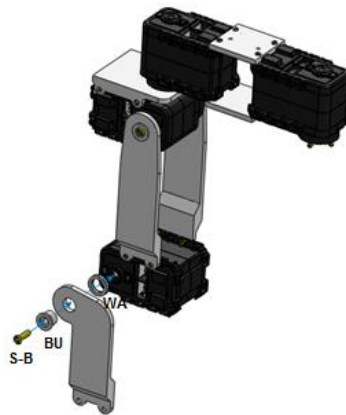


Figura 3.36. Ensamble de la tibia derecha externa.

Autores: Eduardo Morillo y Juan Portero.

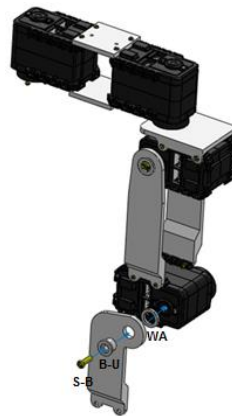


Figura 3.37. Ensamble de la tibia izquierda externa.

Autores: Eduardo Morillo y Juan Portero.

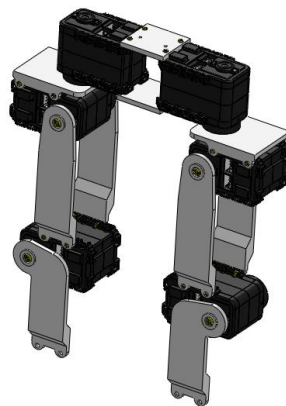


Figura 3.38. Ensamblaje I.

Autores: Eduardo Morillo y Juan Portero.

15. A continuación se ensamblan las piezas 6 y 7 de la Figura 3.35 en su respectiva pierna teniendo en cuenta que los huecos encajan exactamente con los que consta el servomotor y deben estar dispuestos de la forma que se indica en la Figura 3.39; conjuntamente se indica cómo se ubican los tornillos. La Figura 3.40 que corresponde al ensamblaje J, el mismo que muestra la evolución parcial del ensamblaje con la posición correcta de las piezas hasta este literal.

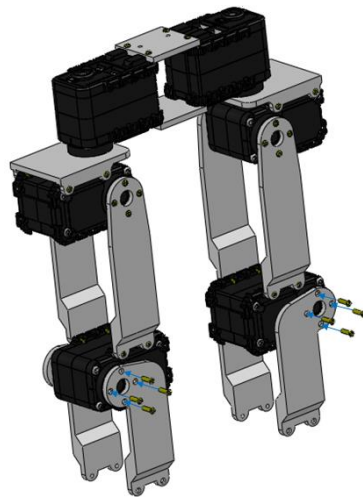


Figura 3.39. Disposición de las piezas 6 y 7 junto con los tornillos S1.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.40. Ensamblaje J.

Autores: Eduardo Morillo y Juan Portero.

16. Se procede al ensamblaje del primer grado de libertad de los tobillos para lo que se toman los servomotores 7 y 8 colocándolos como indica la Figura 3.41, que muestra la disposición que deben tener los servomotores y tornillos que soportan los mismos. La Figura 3.42 muestra el ensamblaje K de este literal.

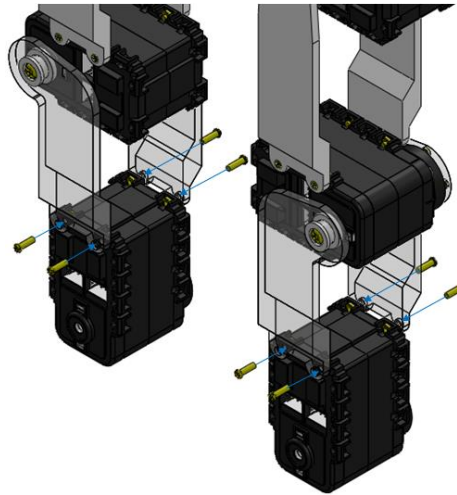


Figura 3.41. Ubicación de los servomotores 7 y 8.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.42. Ensamblaje K.

Autores: Eduardo Morillo y Juan Portero.

17. Para ensamblar el eslabón que conecta el segundo grado de libertad de los tobillos se toman las piezas 9 y 10 mostradas en la Figura 3.43 y se realiza un sub-ensamblaje como se indica en la Figura 3.44.

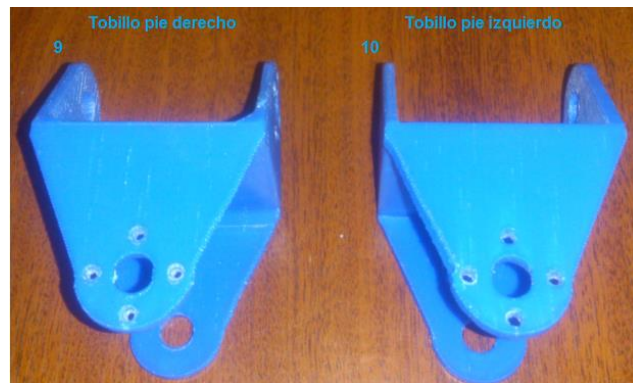


Figura 3.43. Piezas de los tobillos.

Autores: Eduardo Morillo y Juan Portero.

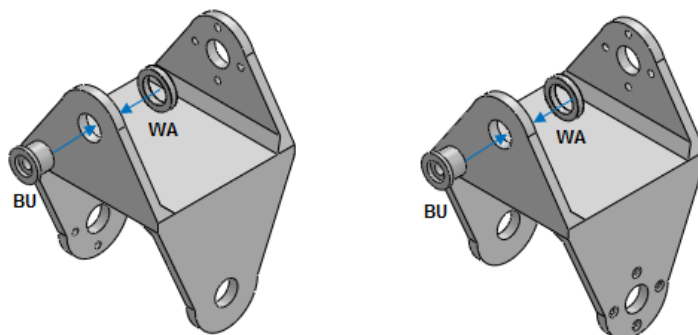


Figura 3.44. Sub-ensamblaje piezas de los tobillos.

Autores: Eduardo Morillo y Juan Portero.

18. Se introduce a presión el sub-ensamble dentro del ensamble K como se muestra en la Figura 3.45 de forma que encajen correctamente los tornillos que transmite el movimiento y que aseguran al servomotor con el eslabón.

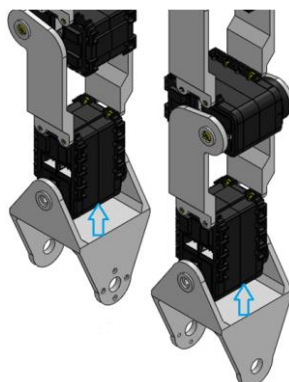


Figura 3.45. Colocación de las piezas 9 y 10.

Autores: Eduardo Morillo y Juan Portero.

19. Se colocan y ajustan los tornillos de los tobillos de la parte frontal y posterior de los servomotores 7 y 8, como se muestra en las Figuras 3.46 y 3.47.

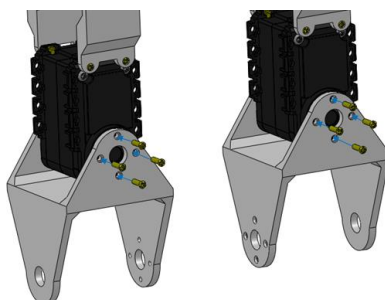


Figura 3.46. Disposición de tornillos S1 de la parte frontal de los tobillos.

Autores: Eduardo Morillo y Juan Portero.

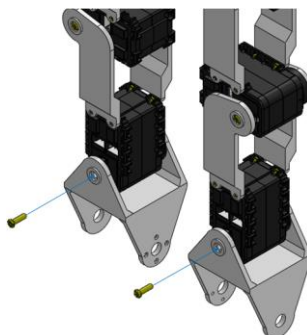


Figura 3.47. Disposición de tornillos SB de la parte posterior de los tobillos.

Autores: Eduardo Morillo y Juan Portero.

20. Al posicionar los servomotores a su posición original se tendrá el ensamble L, como el mostrado en la Figura 3.48.



Figura 3.48. Ensamblaje L.

Autores: Eduardo Morillo y Juan Portero.

21. Se agregan las piezas para transmitir el movimiento del segundo grado de libertad de los tobillos, dado por los servomotores 9 y 10 en su respectiva pierna. La forma de colocar las piezas para la transmisión del movimiento se muestran en la Figura 3.49.

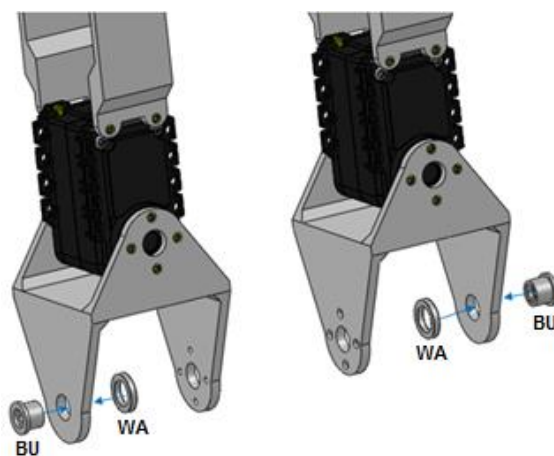


Figura 3.49. Colocación de las piezas para los servomotores 9 y 10.

Autores: Eduardo Morillo y Juan Portero.

22. Se colocan los servomotores 9 y 10 que representan el último grado de libertad del tobillo, este procedimiento se realiza presionando ligeramente los mismos, debido a que las piezas fueron diseñadas para encajar exactamente. La Figura 3.50 muestra como es la disposición en la que se deben colocar los servomotores.

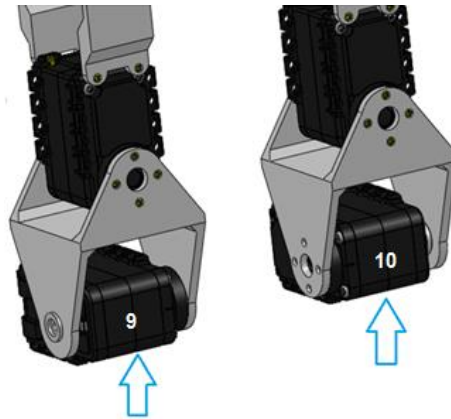


Figura 3.50. Ubicación de los servomotores 9 y 10.

Autores: Eduardo Morillo y Juan Portero.

23. Una vez colocados los servomotores se procede a regresar las piernas a su posición original para facilitar el ajuste de los servomotores colocados en el literal anterior. Los tornillos para este literal van a ser colocados como se observa en la Figura 3.51.

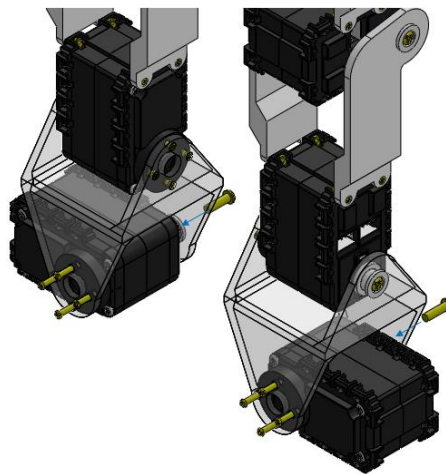


Figura 3.51. Disposición de los tornillos S1 y SB del tobillo.

Autores: Eduardo Morillo y Juan Portero.

24. Ajustados los tornillos se tiene como resultado el ensamblaje M que se aprecia en la Figura 3.52.



Figura 3.52. Ensamblaje M.

Autores: Eduardo Morillo y Juan Portero.

25. Realizado el ensamblaje M, se coloca el pie izquierdo del robot para que éste encaje con los huecos del servomotor donde se colocan los tornillos para su sujeción. La forma para colocarlo se muestra en la Figura 3.53.

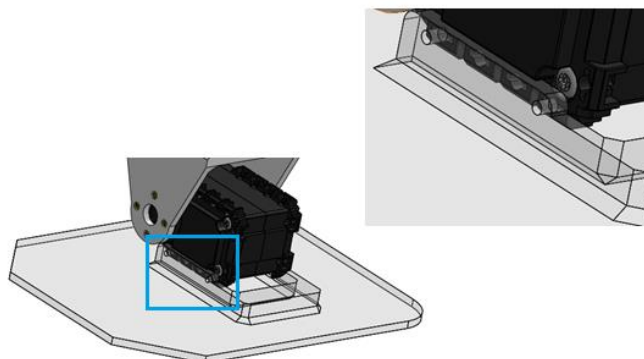


Figura 3.53. Colocación del pie izquierdo.

Autores: Eduardo Morillo y Juan Portero.

26. Se colocan y se ajustan los tornillos de la parte frontal y posterior del pie izquierdo como se muestra en la Figura 3.54.

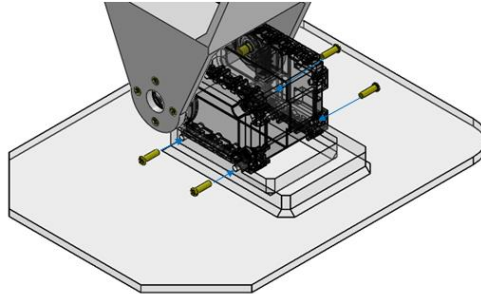


Figura 3.54. Disposición de los tornillos S1 en el pie izquierdo.

Autores: Eduardo Morillo y Juan Portero.

27. Se coloca el pie derecho de igual manera como el pie izquierdo como se observa en la Figura 3.55.

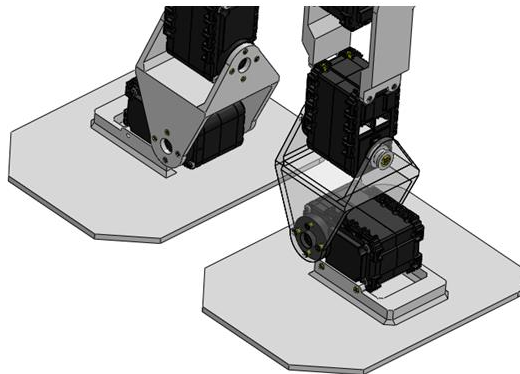


Figura 3.55. Colocación del pie derecho.

Autores: Eduardo Morillo y Juan Portero.

28. Finalmente se colocan y se ajustan los tornillos del pie derecho como se muestra en la Figura 3.56, para dar como resultado el ensamble completo de las extremidades inferiores del robot bípedo en la Figura 3.57.

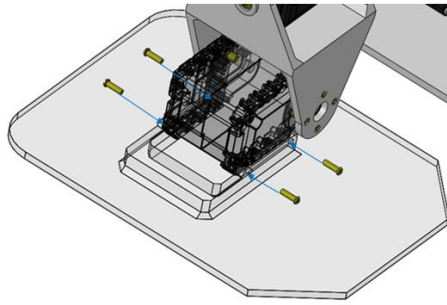


Figura 3.56. Disposición de los tornillos S1 en el pie derecho.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.57. Ensamblaje de las extremidades inferiores.

Autores: Eduardo Morillo y Juan Portero.

b. CABLEADO DE LAS EXTREMIDADES INFERIORES.

1. Se inicia con la colocación del primer cable de comunicación en el primer puerto (izquierdo) del servomotor 9 que pertenece al último grado de libertad del tobillo derecho, como se muestra en la Figura 3.58.



Figura 3.58. Conexión del cable del servomotor 9.

Autores: Eduardo Morillo y Juan Portero.

2. El otro extremo del cable se conecta al primer puerto del servomotor 7 que representa el primer grado de libertad del tobillo, de igual manera se conecta en el segundo puerto otro cable. Este procedimiento se observa en la Figura 3.59.



Figura 3.59. Conexión del cable entre los servomotores 9 y 7.

Autores: Eduardo Morillo y Juan Portero.

3. El otro extremo del cable se conecta al primer puerto del servomotor 5 de la rodilla; con lo que se establece la comunicación de los servomotores 5, 7 y 9. A continuación se toma otro cable y se lo coloca en el segundo puerto de comunicación del servomotor (Figura 3.60).



Figura 3.60. Conexión del cable entre los servomotores 7 y 5.

Autores: Eduardo Morillo y Juan Portero.

4. El extremo libre del cable se conecta al primer puerto de comunicación del servomotor 3; quedando comunicados de esta manera los servomotores 3, 5, 7 y 9. A continuación se conecta un nuevo cable al segundo puerto. Este procedimiento se muestra en la Figura 3.61.



Figura 3.61. Conexión del cable entre los servomotores 5 y 3.

Autores: Eduardo Morillo y Juan Portero.

5. El extremo del cable se conecta al primer puerto del servomotor 1; teniendo de esta manera comunicados a los servomotores 1, 3, 5, 7 y 9 que conforman la pierna derecha del robot (Figura 3.62).



Figura 3.62. Conexión de los servomotores de la pierna derecha.

Autores: Eduardo Morillo y Juan Portero.

6. Finalmente se coloca un cable en el segundo puerto del servomotor 1, que será el encargado de comunicar todos estos servomotores con la tarjeta electrónica. Este mismo procedimiento se repite para la pierna izquierda dando como resultado el cableado de las extremidades inferiores del robot bípedo como se indica en la Figura 3.63.



Figura 3.63. Cableado de las extremidades inferiores.

Autores: Eduardo Morillo y Juan Portero.

c. ENSAMBLAJE DE LA PARTE SUPERIOR.

1. Se coloca la pieza mostrada en la Figura 3.64 de una manera adecuada en la que cuadren los agujeros de la pieza con los de los servomotores 1 y 2, en donde van colocados los tornillos S1 para asegurar dicha pieza. En la Figura 3.65 se hace un acercamiento para ver la correcta ubicación de la espalda.

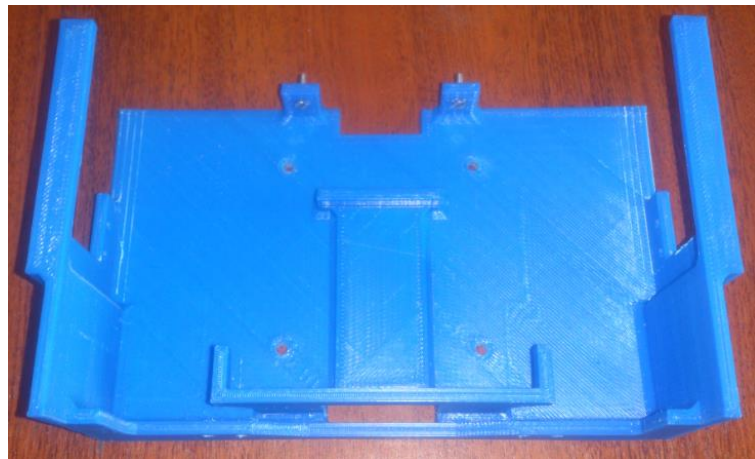


Figura 3.64. Espalda del robot.

Autores: Eduardo Morillo y Juan Portero.

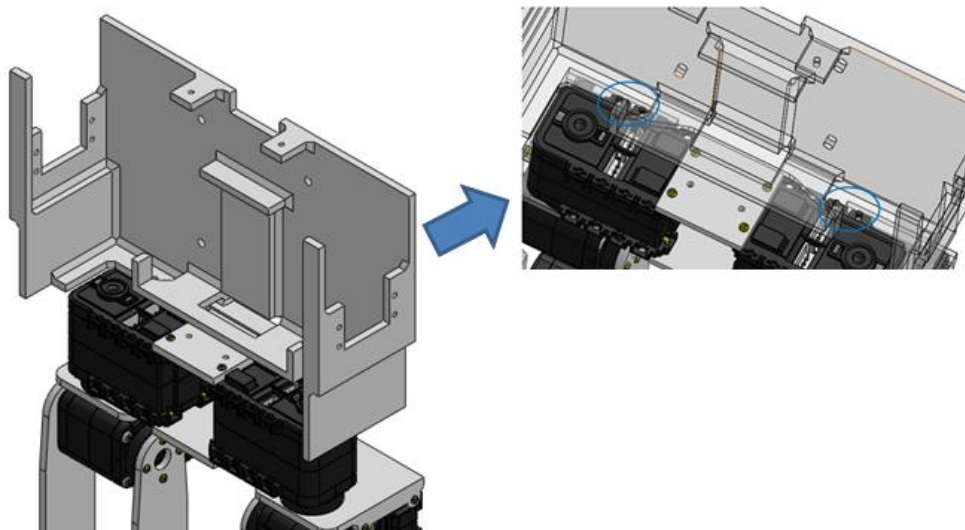


Figura 3.65. Correcto posicionamiento de la espalda.

Autores: Eduardo Morillo y Juan Portero.

2. Se toman cuatro tornillos S1 que son los encargados de ajustar la espalda con la parte inferior del robot; los mismo que deben ser colocados como se muestra en la Figura 3.66 para su ajuste.

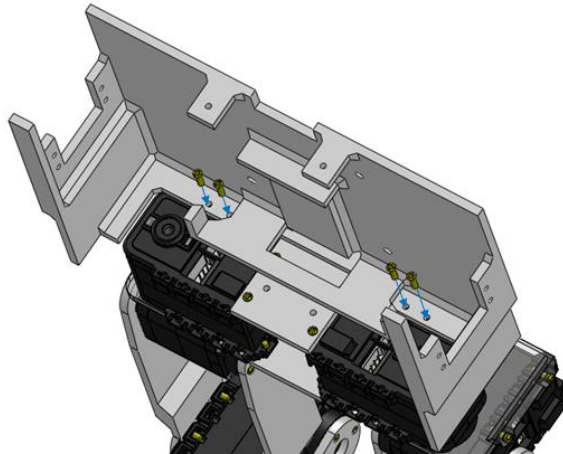


Figura 3.66. Disposición de los tornillos S1 para la espalda.

Autores: Eduardo Morillo y Juan Portero.

3. Se coloca el servomotor 11 del brazo derecho conjuntamente con los tornillos que lo asegurarán al torso del robot, tal como se indica en la Figura 3.67. Al completar este literal se obtiene el ensamblaje N mostrado en la Figura 3.68.

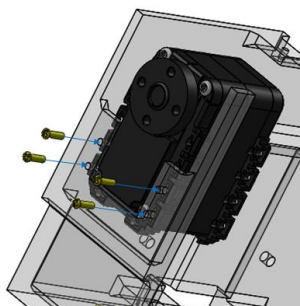


Figura 3.67. Disposición de los tornillos S1 en el servomotor 11.

Autores: Eduardo Morillo y Juan Portero.

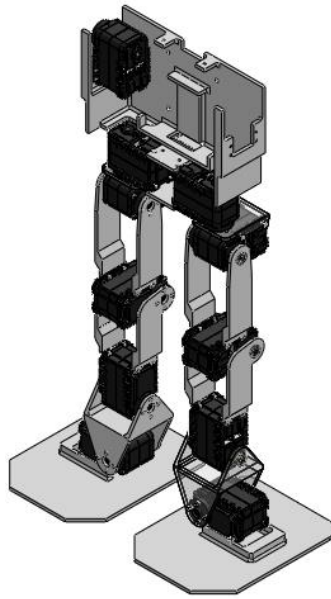


Figura 3.68. Ensamblaje N.

Autores: Eduardo Morillo y Juan Portero.

4. Se coloca el servomotor 12 perteneciente al brazo izquierdo junto con los tornillos que lo sujetan al torso del robot y así proceder a su ajuste, este procedimiento se muestra en la Figura 3.69.

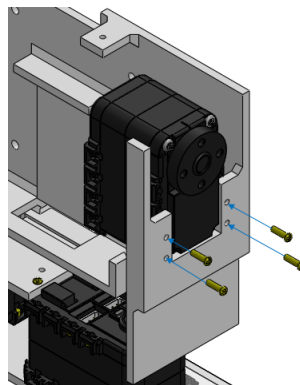


Figura 3.69. Disposición de los tornillos S1 en el servomotor 12.

Autores: Eduardo Morillo y Juan Portero.

5. Se obtiene el ensamblaje O, tal como se muestra en la Figura 3.70.

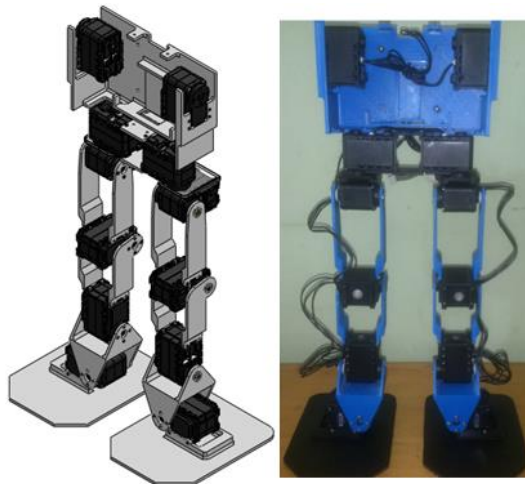


Figura 3.70. Ensamblaje O.

Autores: Eduardo Morillo y Juan Portero.

6. Se coloca la tarjeta electrónica conjuntamente con los tornillos y tuercas que la sujetan a la parte posterior de la espalda (Figuras 3.71 y 3.72).

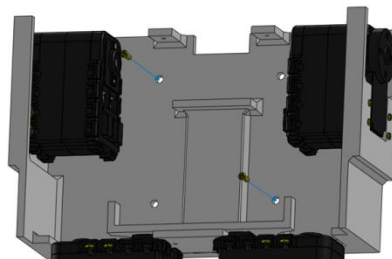


Figura 3.71. Disposición de los tornillos para la tarjeta.

Autores: Eduardo Morillo y Juan Portero.

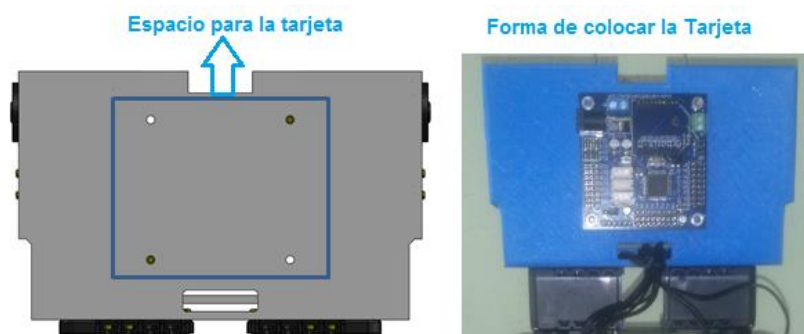


Figura 3.72. Colocación de la tarjeta.

Autores: Eduardo Morillo y Juan Portero.

7. Antes de ensamblar todo el tronco del robot se coloca la batería en el lugar dispuesto para la misma, como se observa en la Figura 3.73. Se debe recordar que los cables de la batería deben estar dispuestos hacia la parte superior, facilitando de esta manera su posterior conexión.

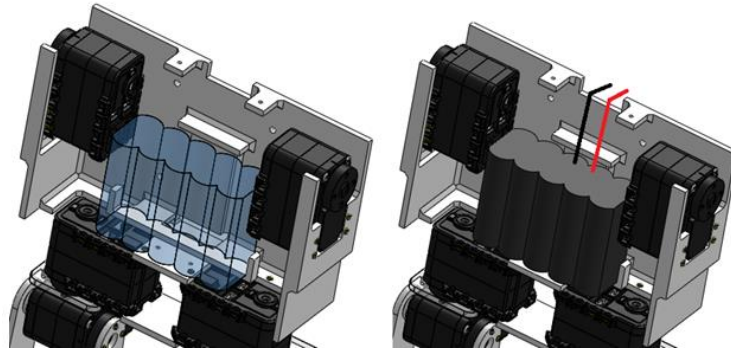


Figura 3.73. Colocación de la batería.

Autores: Eduardo Morillo y Juan Portero.

8. Se conectan los cables de los servomotores 11 y 12 de los brazos para no tener inconvenientes, al igual que los cables de la batería, se los deja orientados hacia la parte superior como se indica en la Figura 3.74.



Figura 3.74. Cableado de los servomotores 11 y 12.

Autores: Eduardo Morillo y Juan Portero.

9. Se cierra la parte frontal con la colocación del pecho mostrado en la Figura 3.75, el cual se posiciona de forma que coincidan los agujeros de la pieza con los de los servomotores 1 y 2 para que quepan los tornillos S1, como se indica en la Figura 3.76 la cual es una representación simplificada de como colocar la pieza y los tornillos.

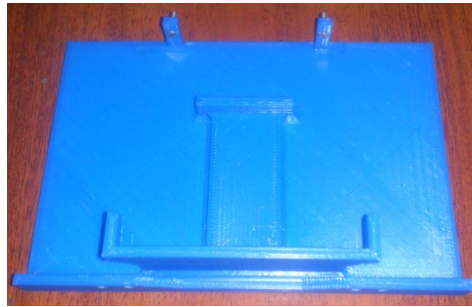


Figura 3.75. Pecho del robot.

Autores: Eduardo Morillo y Juan Portero.

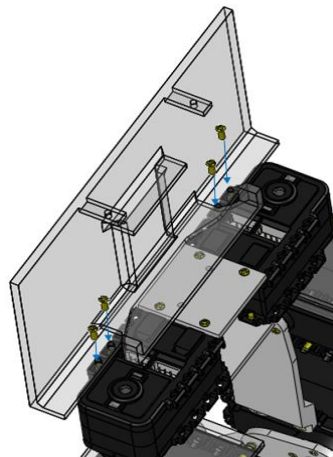


Figura 3.76. Disposición de los tornillos S1 para el pecho.

Autores: Eduardo Morillo y Juan Portero.

10. Al completar el procedimiento anterior se puede apreciar en la Figura 3.77 el ensamblaje P, donde estas dos partes conforman el tronco del robot bípedo.

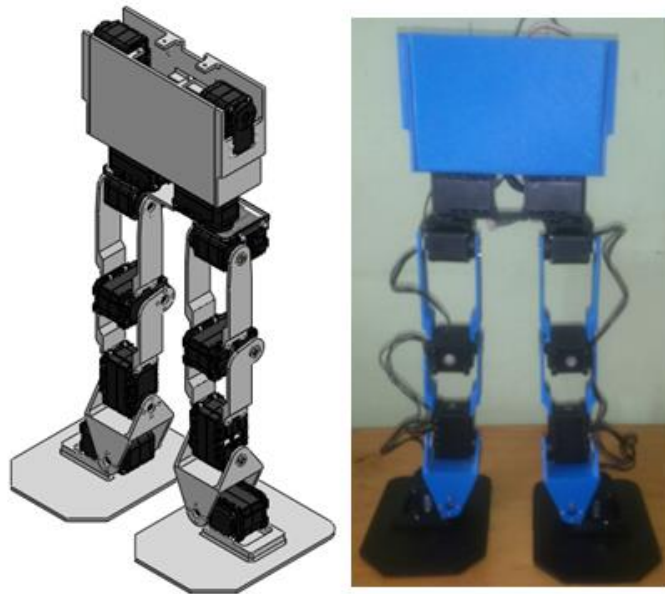


Figura 3.77. Ensamblaje P.

Autores: Eduardo Morillo y Juan Portero.

11. Se realiza el sub-ensamble Q de las piezas de las Figuras 3.78 y 3.79, este sub-ensamblaje representan el cuello y la cabeza del robot. En este caso se realiza una función similar al de una tapa sellando el tronco del robot, la forma de realizarlo es detallada en la Figura 3.80.

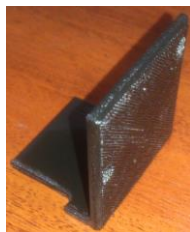


Figura 3.78. Cabeza del robot.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.79. Cuello del robot.

Autores: Eduardo Morillo y Juan Portero.

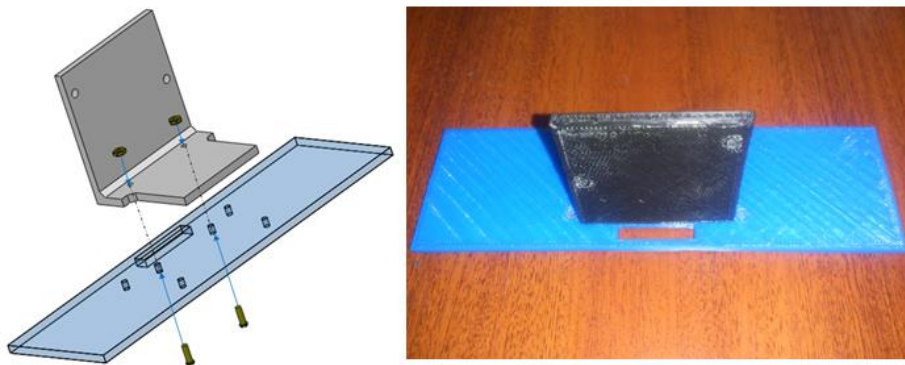


Figura 3.80. Sub-ensamble Q.

Autores: Eduardo Morillo y Juan Portero.

12. Al tener listo el sub-ensamblaje Q se monta el sensor de ultrasonido, como se indica en las Figura 3.81 y 3.82; donde se muestra la ubicación de los tornillos y la forma de colocar el sensor en la cabeza, respectivamente.

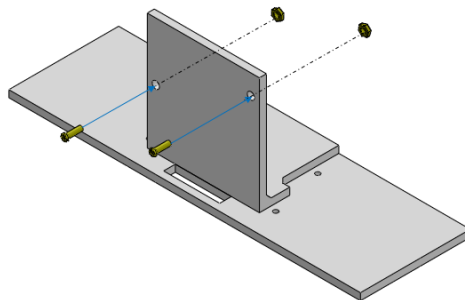


Figura 3.81. Ubicación de los tornillos para el sensor.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.82. Montaje del sensor.

Autores: Eduardo Morillo y Juan Portero.

13. Se une el ensamblaje P con el sub-ensamblaje Q, obteniendo el ensamblaje R, se observa la forma que toma el robot, siendo similar a la de un humanoide, como se puede notar en la Figura 3.84. La Figura 3.83 muestra la forma en la que se colocan los cables de este ensamblaje.



Figura 3.83. Colocación de los cables del sensor.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.84. Ensamblaje R.

Autores: Eduardo Morillo y Juan Portero.

14. Finalmente se colocan los brazos del robot en el ensamblaje R, como se muestran en las Figura 3.85 y 3.86, teniendo como resultado en ensamblaje final del robot indicado en la Figura 3.87.

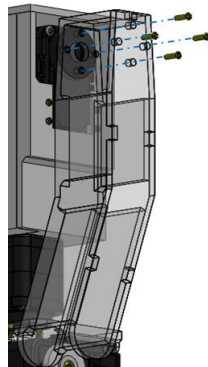


Figura 3.85. Disposición de los tornillos S1 para el brazo izquierdo.

Autores: Eduardo Morillo y Juan Portero.

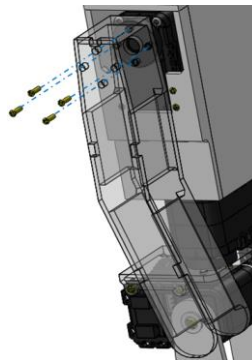


Figura 3.86. Disposición de los tornillos S1 para el brazo derecho.

Autores: Eduardo Morillo y Juan Portero.

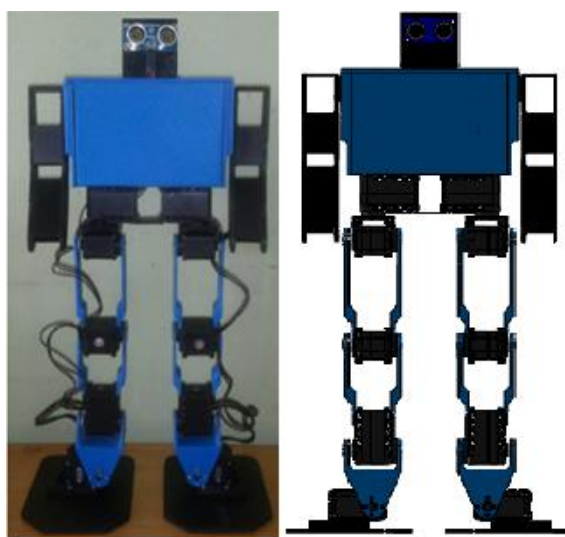


Figura 3.87. Ensamblaje final del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

3.2. MONTAJE DEL SISTEMA ELECTRÓNICO.

Al haber culminado el procedimiento de la selección de los diferentes componentes electrónicos que conforman la etapa electrónica y de control del robot; contando además con la etapa mecánica ya ensamblada, en esta sección se describe como van conectados cada uno de los bloques que conforman el robot bípedo, pues como se observó anteriormente los elementos ya fueron montados sobre la estructura.

La Figura 3.88 muestra donde se ubica el circuito de accionamiento del robot, dicha ubicación se escoge debido a que en ese punto convergen los cables necesarios para la alimentación y control de la tarjeta electrónica.

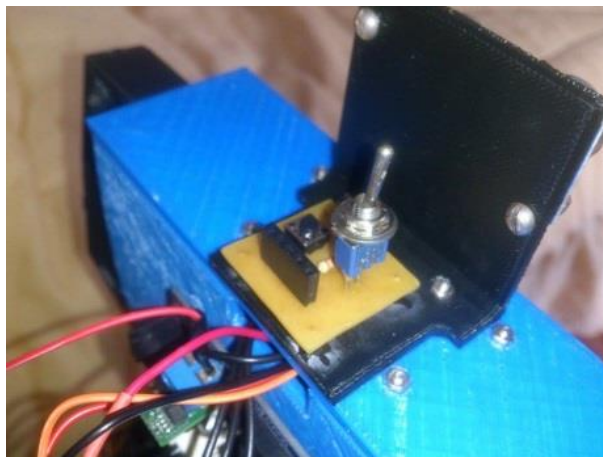


Figura 3.88. Colocación del circuito de accionamiento.

Autores: Eduardo Morillo y Juan Portero.

3.2.1. CONEXIONES DEL SISTEMA DE INSTRUMENTACIÓN.

El sistema de instrumentación está constituido por el sensor de ultrasonido Parallax PING y la tarjeta electrónica Arbotix-M; el sensor envía una señal de ultrasonido y mide el tiempo de rebote del sonido, para entregar la distancia por medio de su pin de salida de señal, donde se mide el ancho de pulso PWM en función de la distancia del obstáculo. Las conexiones se muestran en la Figura 3.89.

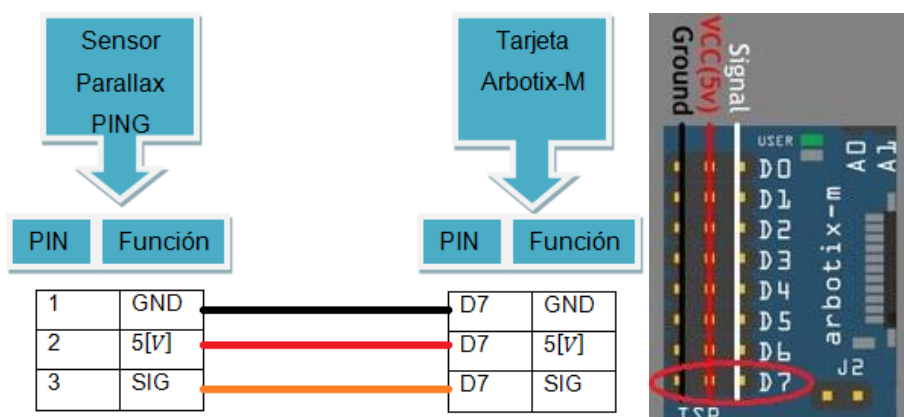


Figura 3.89. Conexión pines del sensor con la tarjeta Arbotix-M.

Autores: Eduardo Morillo y Juan Portero.

3.2.2. CONEXIONES DEL CIRCUITO DE ACCIONAMIENTO.

El circuito de accionamiento consta con un pulsador que está conectado a una entrada digital de la tarjeta electrónica, asignado al pulso de inicio de la caminata. También incluye un switch que es el encargado de cortar el suministro de energía hacia la tarjeta de esta manera permite resetear el robot; cabe aclarar que el negativo de la batería va conectado a la entrada de voltaje negativa de la fuente. Las conexiones se muestran en la Figura 3.90.

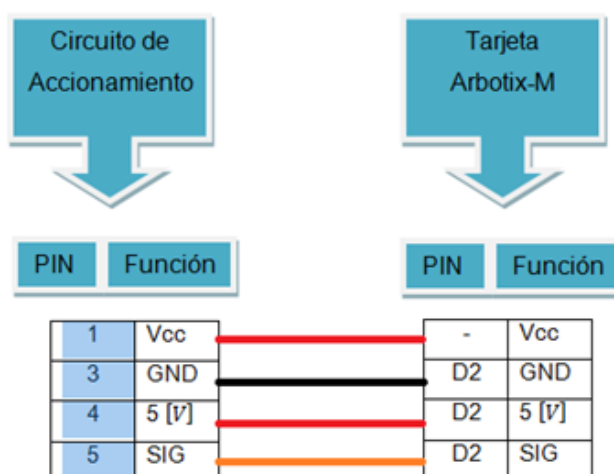


Figura 3.90. Conexión pines de la tarjeta Arbotix-M con el circuito de accionamiento.

Autores: Eduardo Morillo y Juan Portero.

3.3. IMPLEMENTACIÓN DEL SISTEMA DE CONTROL.

Al tener ensamblada la estructura y realizado el montaje de todos los componentes, en este apartado se realizará la implementación del sistema de control por medio de la programación en Arduino (literal 3.6), para que el robot realice las acciones previstas en el literal 2.1.3 del funcionamiento; la programación se realizará en base al diagrama de flujo de la Figura 2.61; como el sistema de control está basado en el método de la heurística las trayectorias del caminado y los giros del robot bípedo se obtendrán en el literal 3.5.

3.3.1. INTEGRACIÓN TARJETA ARBOTIX-M CON ARDUINO.

La tarjeta electrónica Arbotix-M que se ha escogido para este trabajo se la controla mediante el IDE de Arduino, la instalación de este programa es muy sencilla y el paquete de instalación se lo encuentra en la web de la empresa, la versión utilizada es la 1.0.5 como se muestra en la Figura 3.91.



Figura 3.91. Versión de Arduino.

Fuente: <http://arduino.cc/en/Main/Software>

Una vez instalado el programa, se tiene que ejecutar el mismo para que se cree la carpeta Arduino en la carpeta documentos del computador. En la Figura 3.92 se indica su entorno.

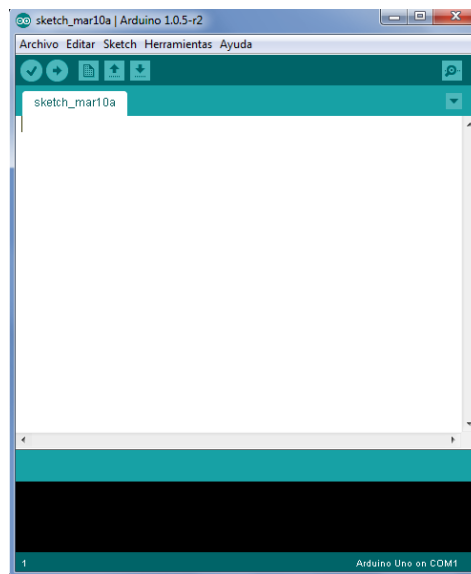


Figura 3.92. Pantalla principal del programa Arduino.

Autores: Eduardo Morillo y Juan Portero.

A continuación hay que instalar los drivers FTDI, que permitirán que el UartSBee funcione adecuadamente y establezca la comunicación entre la tarjeta con el computador.

Estos drivers se encuentran en el link <http://www.ftdichip.com/Drivers/VCP.htm>; en la Figura 3.93 se indica la versión utilizada en base a las características del computador.

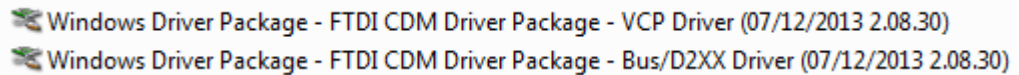
Currently Supported VCP Drivers:

Operating System	Release Date	Processor Architecture						
		x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4
Windows 8.1	2013-10-21	2.08.30 8.1	2.08.30 8.1	-	-	-	-	-
Windows*	2013-08-01	2.08.30	2.08.30	-	-	-	-	-
Linux	2009-05-14	1.5.0	1.5.0	-	-	-	-	-
Mac OS X	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-
Windows CE 4.2-5.2**	2012-01-06	1.1.0.10	-	-	1.1.0.14	1.1.0.10	1.1.0.10	1.1.0.10
Windows CE 6.0	2012-01-06	1.1.0.10	-	-	1.1.0.14	1.1.0.10	1.1.0.10	1.1.0.10

Figura 3.93. Drivers FTDI.

Fuente: <http://www.ftdichip.com/Drivers/VCP.htm>

Una vez descargado el archivo *.zip* que contiene el instalador, se ejecuta el archivo *CDM.exe* respectivo, si no hubo ningún problema aparecerá la siguiente pantalla.



Al tener lista la comunicación, se procede a descargar los archivos necesarios para integrar la tarjeta electrónica con el entorno de Arduino. En el siguiente link se pueden descargar todos estos archivos necesarios <https://github.com/trossenrobotics/arbotix/archive/master.zip>, en este archivo *.zip* se encuentran 3 carpetas: Arbotix Sketches, hardware y libraries. Para que se puedan usar estas carpetas hay que moverlas a la carpeta Arduino que se creó anteriormente en la carpeta mis documentos, si no se las mueve a esa dirección no van a tener efecto, en la Figura 3.94 se puede apreciar la ruta de las carpetas.

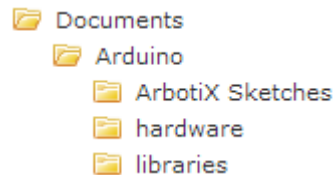


Figura 3.94. Distribución de las carpetas Arduino.

Fuente: <http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

Después hay que abrir el programa Arduino para verificar si se realizó correctamente todos los pasos anteriores, al ingresar en las pestañas que indica la Figura 3.95; si muestra los resultados que están en color rojo, la tarjeta esta lista para ser programada.

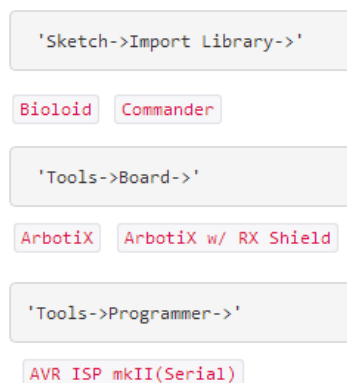


Figura 3.95. Comprobación Arbotix en Arduino.

Fuente: <http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

Finalmente para conectar la tarjeta con el computador, se tiene que realizar de la manera que indica la Figura 3.96, el cable USB se conectar al UartSBee, el cual se conecta por medio del cable FTDI a la tarjeta Arbotix-M.

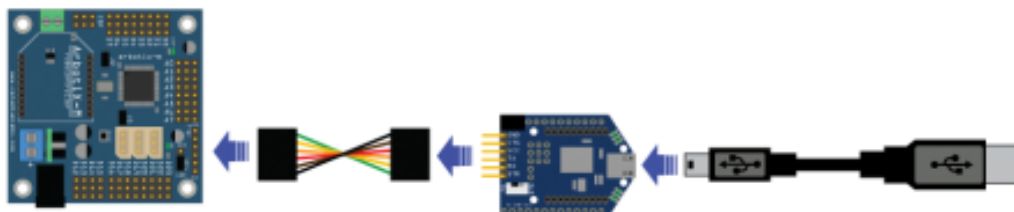


Figura 3.96. Hardware Arbotix-M con PC.

Fuente: <http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

Tomar en cuenta que en la tarjeta hay un jumper el cual para esta conexión tiene que estar colocado como indica la Figura 3.97 para que el voltaje de alimentación lo obtenga del computador mismo. El cable FTDI es de conexión cruzada y en la tarjeta electrónica y UartSBee están indicados los colores para una fácil conexión; el selector de voltaje del UartSBee tiene que estar colocado en 5 [V]. Considerar además que al utilizar la comunicación FTDI no puede estar conectado nunca el Xbee ya que utilizan el mismo puerto serial y puede dañarse la tarjeta electrónica.

Con esta configuración se cargarán siempre los programas desde el entorno Arduino hacia la tarjeta Arbotix-M.

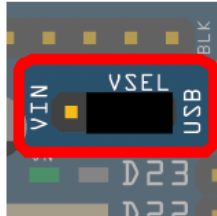


Figura 3.97. Jumper de energía Arbotix-M con PC.

Fuente: <http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

3.3.2. ASIGNACIÓN DE ID EN LOS SERVOMOTORES.

Este Software sirve para asignar el ID a cada servomotor, ya que los mismos vienen de fábrica con el número 1, este ID se utiliza para que cada servomotor sepa que tarea tiene que realizar en cada momento; la comunicación es a través de un BUS de datos y con el ID se pueden diferenciar entre ellos.

Este programa necesita que se tenga previamente instalado JAVA en el computador.

En el siguiente link se puede descargar la última versión de DynaManager <https://github.com/trossenrobotics/dynaManager/releases>; una vez descargado e instalado el software se debe descargar a la Arbotix-M el archivo *ROS.ino* desde Arduino de la siguiente manera **File->Sketchbook->ArbotiX Sketches->ROS**, previamente seleccionado el puerto serial COM y la tarjeta Arbotix en el menú **herramientas**; una vez descargado el archivo se procede a conectar el hardware como indica la Figura 3.98, tener en cuenta que ahora el jumper de la tarjeta Arbotix-M tiene que estar como indica la Figura 3.99, ya que va a tomar la energía de la fuente externa de 12 [V] y no del computador, esta fuente externa puede ser un cargador o una batería conectada a la torreta azul. El servomotor se tiene que conectar en los sócalos especiales para los servomotores Dynamixel de

la tarjeta electrónica (sócalos blancos), no importa en cuál de ellos se lo conecte, pero siempre tiene que ser un servomotor a la vez, debido a las características propias del programa el cual permite el manejo de solo un servomotor.

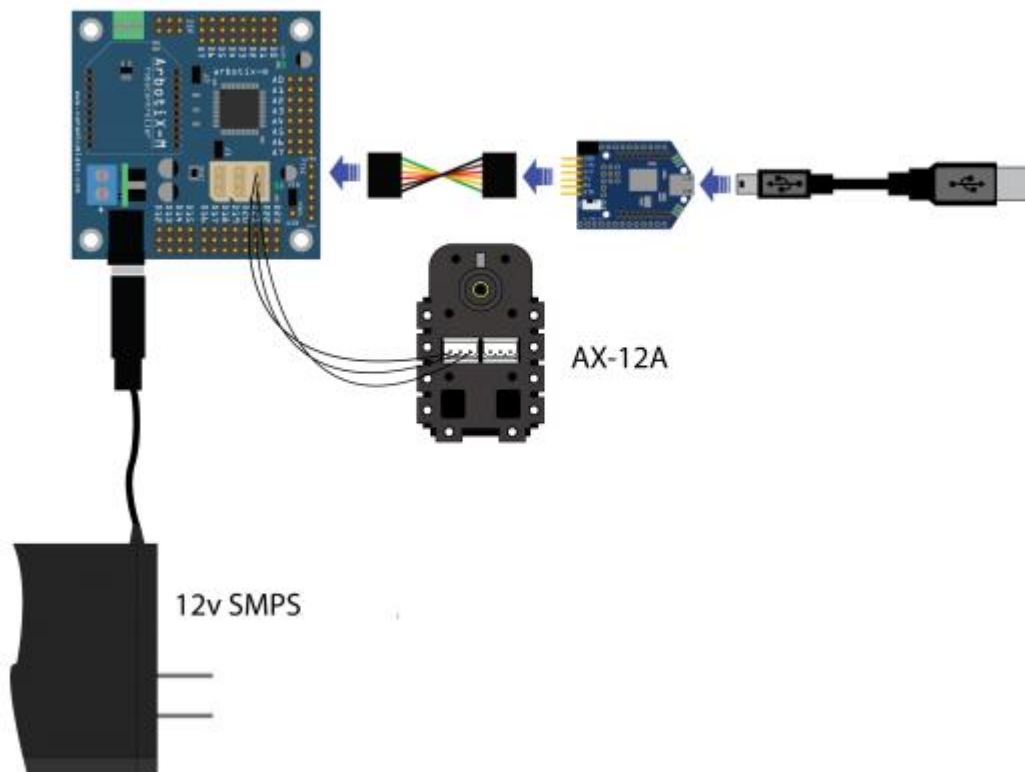


Figura 3.98. Hardware Arbotix-M con servomotor.

Fuente: <http://learn.trossenrobotics.com/arbotix/1-using-the-tr-dynamixel-servo-tool#&panel1-1>



Figura 3.99. Jumper energía Arbotix-M con servomotor.

Fuente: <http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide>

Después se tiene que abrir el programa DynaManager y aparece la pantalla de la Figura 3.100, aquí se seleccionan el puerto COM de comunicación y en la pestaña **SET ID/BAUD** se asigna el número de ID para cada servomotor, considerar que este número también se escribe en la parte posterior del servomotor para un fácil manejo del mismo. En este programa también se puede verificar el movimiento manual del servomotor en la pestaña **TEST SERVO**.



Figura 3.100. Pantalla principal DynaManager.

Autores: Eduardo Morillo y Juan Portero.

3.3.3. CONFIGURACIÓN MÓDULOS XBEE.

Los dos módulos XBee son utilizados para almacenar las posiciones de la caminata del robot bípedo, la tarjeta Arbotix-M tiene un sócalo específico para la conexión de un XBee, de igual manera la tarjeta UartSBee tiene un sócalo específico para éste módulo, de esta manera se puede establecer la comunicación inalámbrica ya que el módulo XBee #1 va a estar en el UartSBee que se conecta con el computador y el XBee #2 será conectado a

la tarjeta Arbotix-M que está colocada en el robot bípedo. Para lograr la transmisión correcta se configuran los módulos para enviar y recibir la información, esto se logra creando una red PAN, los XBee tienen el mismo ID de esta red, también hay que configurar la velocidad de comunicación y la dirección de destino.

Primero se necesita instalar la versión 5.2.8.6 del software X-CTU, se lo puede descargar del link <http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities>. También se necesita descargar los perfiles de cada XBee del link <https://github.com/trossenrobotics/xbeeProfilesInterbotiX/archive/master.zip>.

Ahora se procede a configurar los dos XBee, se tiene que conectar el hardware (Cable USB, UartSBee, módulo XBee) como indica la Figura 3.101, se puede reemplazar el UartSBee por un XBee explorer si se desea.

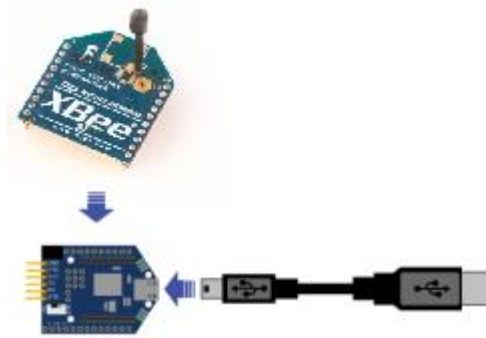


Figura 3.101. Hardware XBee con PC.

Autores: Eduardo Morillo y Juan Portero.

Colocar el selector de energía del UartSBee en 3v3 para que se habilite la comunicación con el XBee.

Al tener conectado el primer XBee se abre el programa X-CTU, seleccionar el puerto COM de comunicación, colocar la velocidad en 38400 baud; se lee los datos del dispositivo en la pestaña **read** y como paso seguido en la pestaña **load** se elige el archivo *xbee_for_commander.pro* que

se descargó anteriormente, finalmente se pulsa el botón **write** para cargar esta información en el XBee.

Para el segundo XBee se procede de la misma manera, pero se carga el archivo *xbee_for_robocontroller.pro*, en la Tabla 3.2 se indican los valores que se han modificado.

Tabla 3.2. Parámetros XBee para la comunicación.

PARÁMETRO	XBee #1	Xbee #2
PAN ID	3400	3400
MY ID	1	2
DL ID	2	1
Data Rate	5	5


Fuente: <http://learn.trossenrobotics.com/arbotix/10-configuring-xbees-for-the-arbotix#&panel1-3>

3.3.4. SOFTWARE PARA ALMACENAR LAS POSICIONES.

Para almacenar las posiciones del robot bípedo se va a utilizar el programa PyPose, el cual es una herramienta creada para ayudar de forma considerable a las personas que se encuentren realizando trabajos en el área de la robótica y más aún para los del área de humanoides, bípedos, cuadrúpedos, hexápodos, etc., ya que permite realizar secuencias de movimientos de todos los servomotores al mismo tiempo y no uno a la vez, esta ventaja ayuda a perfeccionar las trayectorias de un robot.

Este programa está construido en base al lenguaje Python, utiliza el programa wxPython para darle un entorno gráfico a la aplicación, además de esto se necesita el software PySerial para realizar la comunicación entre el computador y el robot.

Para este proyecto se ha instalado la versión 2.7.6 de Python, se lo puede encontrar en el link <https://www.python.org/downloads/>, ya que es software libre al igual que todos los programas que se han visto en los puntos anteriores, a continuación del siguiente link

<http://sourceforge.net/projects/wxpython/files/wxPython/> se descarga la versión de wxPython que encaje con la versión de Python instalada y con las características del computador, para este proyecto se ha instalado wxPython 2.8, finalmente del link <http://sourceforge.net/projects/pyserial/files/pyserial/>, hay que descargar la versión de PySerial que concuerde con la versión de Python, para este proyecto es la versión 2.7; una vez descargado ingresar a propiedades del sistema con clic derecho en Mi PC, seguido a opciones avanzadas del sistema, variables de entorno, en las variables del sistema encontrar la opción **path** y editarla para ingresar la línea **C:\PythonXX** (las siglas **XX** son la versión utilizada de Python, para este proyecto la línea quedaría de la siguiente manera **C:\Python27**), seguido se tiene pulsar aceptar y reiniciar el computador, tipear las teclas Windows+r y escribir cmd para ingresar al código del sistema; digitar **python** en la pantalla se indica la versión de Python y otras características como lo indicado en la Figura 3.102 si todo se realizó correctamente, finalmente copiar la carpeta *pyserial-x.x* y el archivo *pyserial-x.x.tar.gz* previamente descargados a la dirección C:/PythonXX/Lib/site-packages; hay que volver a ingresar al código del sistema y navegar mediante los distintos comandos que hay que tipiar para llegar a la dirección C:/PythonXX/Lib/site-packages/pyserial-X.X que es la carpeta que contiene el archivo  *setup.py*, una vez en esta dirección se ingresa el comando **python setup.py install** con esto todo queda listo para la utilización del programa PyPose.

```

C:\windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>python
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```


Figura 3.102. Comprobación del programa Python.

Autores: Eduardo Morillo y Juan Portero.

El siguiente paso es cargar en la tarjeta Arbotix-M el archivo *pypose.ino* de la misma manera como cuando se cargó el programa *ROS.ino* (página

179), la tarjeta queda cargada con la programación necesaria para la captura de posiciones.

Una vez cargado el software en la tarjeta, se procede a conectar el hardware, verificar que el computador este bien conectado al UartSBee a través del cable USB, en el UartSBee se conecta el XBee #1 y se coloca el selector en 3v3; en la tarjeta se conectan los servomotores, el XBee #2 y la fuente externa que brinde los 12 [V] que necesitan los servomotores para su funcionamiento.

El software PyPose se lo descarga del link <https://code.google.com/p/arbotix/downloads/detail?name=arbotix-0015.zip&can=2&q=>, en la carpeta pypose se encuentra el archivo  PyPose.py el cual hay que ejecutarlo para abrir el programa, aparecerá la pantalla de la Figura 3.103.

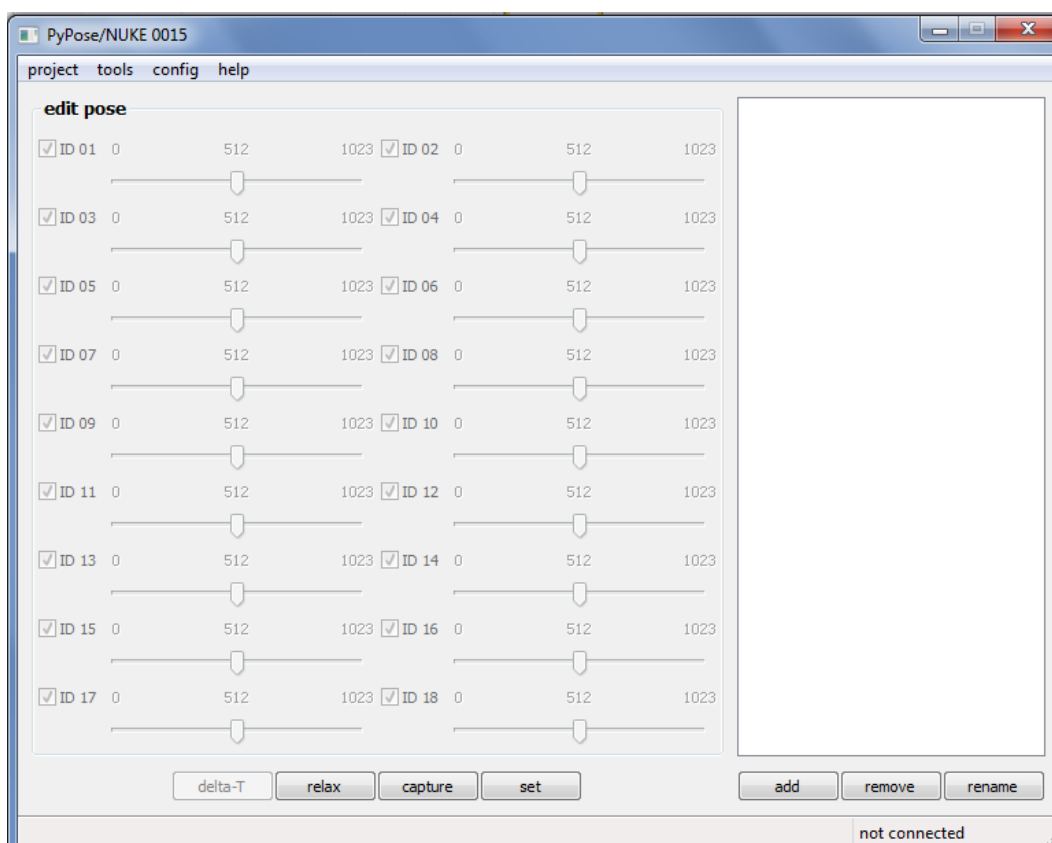


Figura 3.103. Pantalla principal del programa PyPose.

Autores: Eduardo Morillo y Juan Portero.

Para definir el robot bípedo hay que crear un **nuevo proyecto**, se muestra en la Figura 3.104, aquí se ingresa el nombre del robot (*Bipedo*), el número de servomotores a ocupar (12) y la resolución de los grados; hay dos opciones, 1024, que serían 3.41 espacios por grado que se mueve el servomotor, o la opción 4096, que serían 13.65 espacios por cada grado; para este proyecto se ha escogido una resolución de 1024.

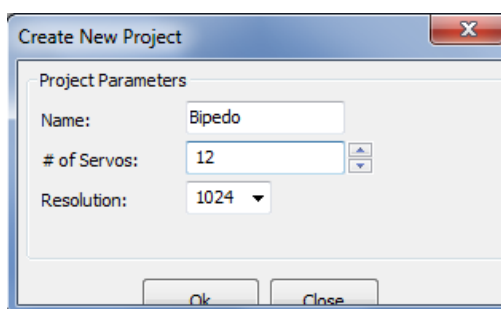


Figura 3.104. Nuevo proyecto en PyPose.

Autores: Eduardo Morillo y Juan Portero.

El siguiente paso es seleccionar el puerto de comunicación en la pestaña **config**, de esta manera queda listo el programa para iniciar con el almacenamiento de posiciones, como indica la Figura 3.105.

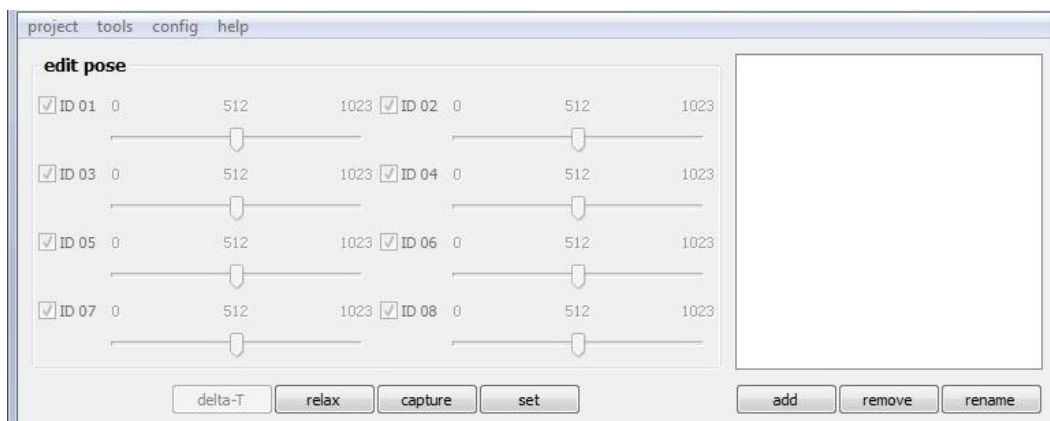


Figura 3.105. Proyecto bípedo en PyPose.

Autores: Eduardo Morillo y Juan Portero.

3.4. INTEGRACIÓN DE SISTEMAS.

Al tener listo todo lo relacionado a hardware y software se procede a integrar todos los sistemas, la estructura esta lista mecánicamente como lo indica la Figura 3.106.

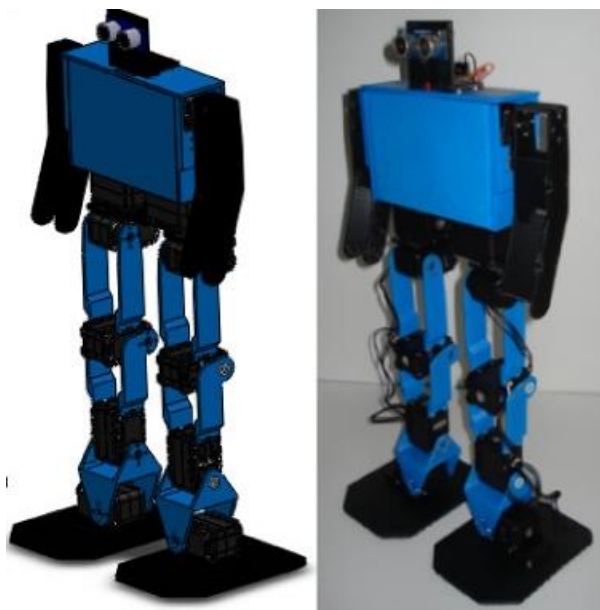


Figura 3.106. Robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

También electrónica y eléctricamente, como lo mostrado en la Figura 3.107; además los programas PyPose y Arduino funcionan correctamente.

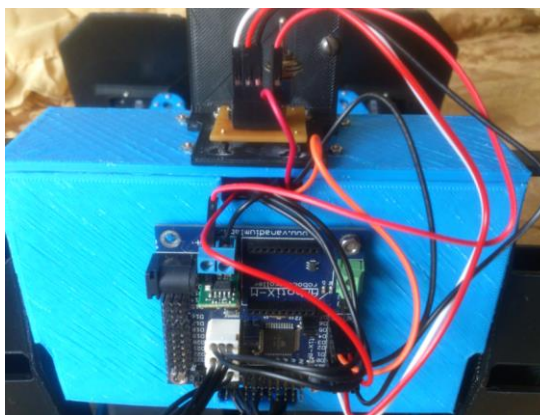


Figura 3.107. Componentes electrónicos.

Autores: Eduardo Morillo y Juan Portero.

A continuación se detalla cómo trabajan estos bloques en conjunto para crear una posición y una secuencia de posiciones; estas posiciones son solo un ejemplo, las utilizadas para las trayectorias del robot bípedo se especifican en el literal 3.5 al igual que el algoritmo de control mediante la programación en el Arduino IDE para que el robot camine autónomamente en el literal 3.6.

La primera posición se la ha denominado “inicial”, se muestra en la Figura 3.108 y es cuando todos los servomotores van a estar en la posición de cero grados de movimiento, aquí el robot va a estar totalmente erguido como muestra la Figura 3.109.

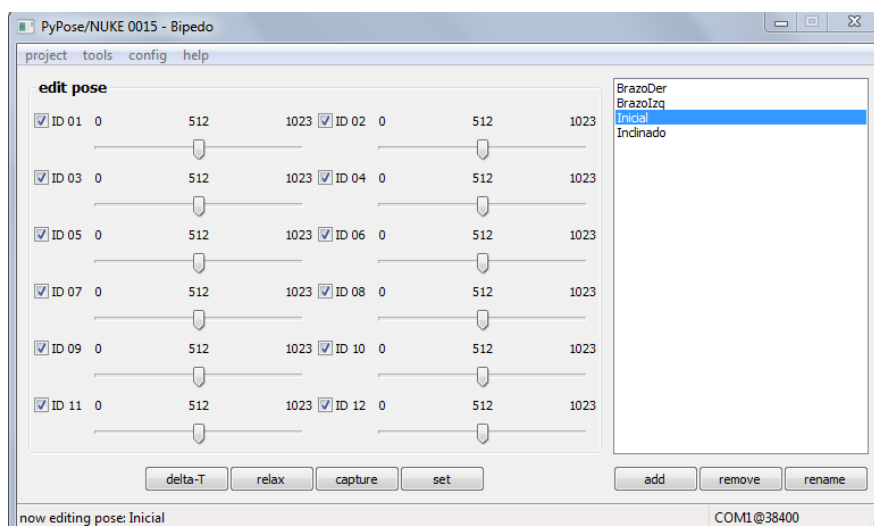


Figura 3.108. Posición inicial en PyPose.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.109. Posición inicial.

Autores: Eduardo Morillo y Juan Portero.

La siguiente posición se la ha llamado “BrazoDer” (Figura 3.110), en esta posición el robot va a levantar el brazo derecho 70 grados, como se aprecia en la Figura 3.111.

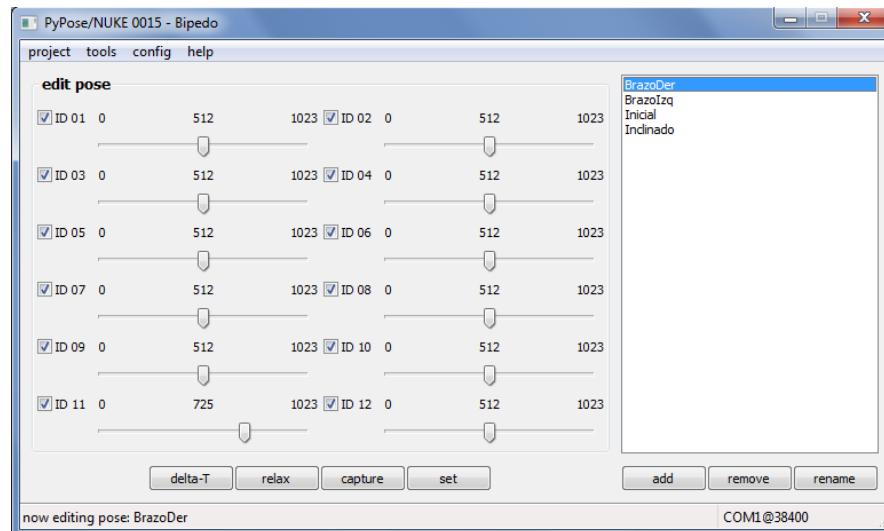


Figura 3.110. Posición del brazo derecho en PyPose.

Autores: Eduardo Morillo y Juan Portero.

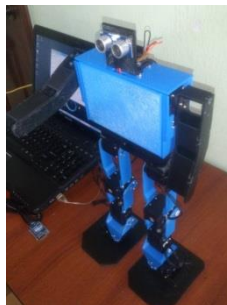


Figura 3.111. Posición del brazo derecho.

Autores: Eduardo Morillo y Juan Portero.

De igual manera la siguiente posición levanta el brazo izquierdo la misma cantidad de grados, se llamó a esta posición “BrazoIzq”, (Figura 3.112), se observa que los dos servomotores tienen un valor y en la Figura 3.113 se aprecia físicamente esta posición asignada.

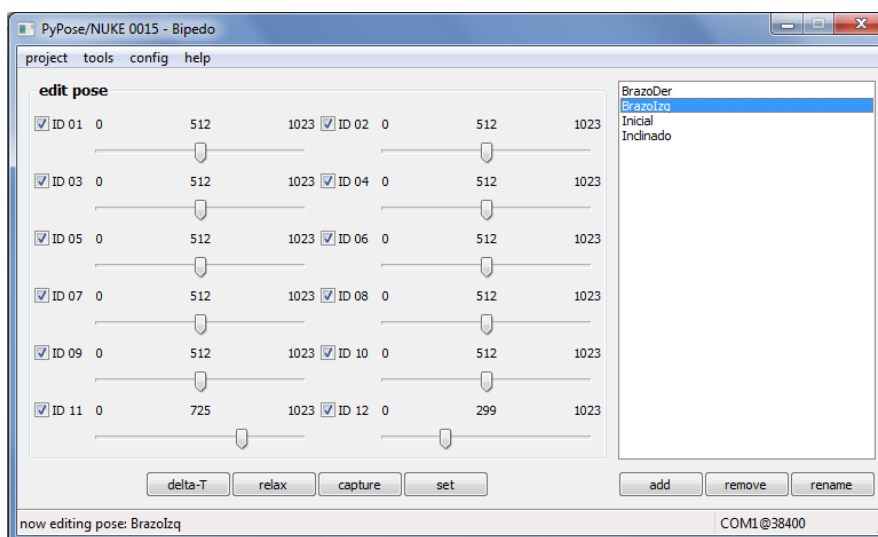


Figura 3.112. Posición del brazo izquierdo en PyPose.

Autores: Eduardo Morillo y Juan Portero.



Figura 3.113. Posición del brazo izquierdo.

Autores: Eduardo Morillo y Juan Portero.

Hasta el momento solo se ha modificado un servomotor por posición, también se puede modificar todos los servomotores que se desee; en la nueva posición “inclinado” como se observa en la Figura 3.114, y físicamente en la Figura 3.115.

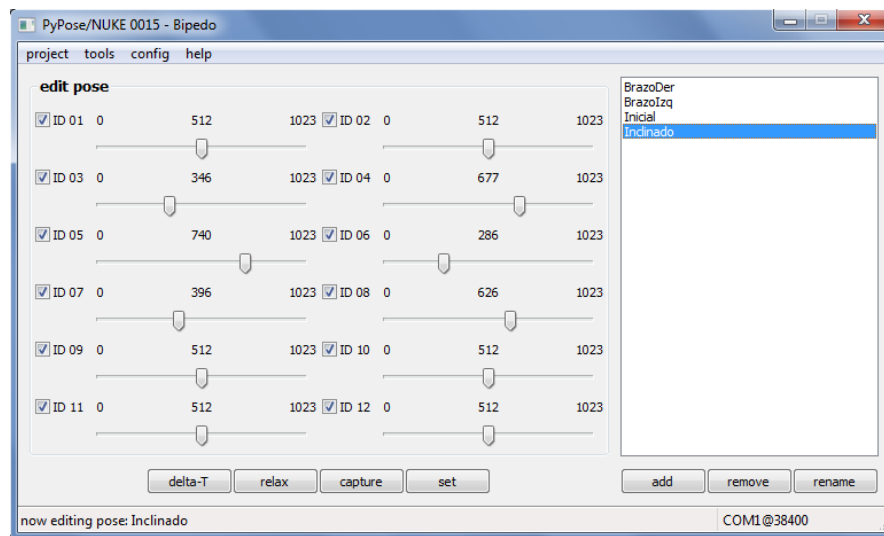


Figura 3.114. Posición inclinada en PyPose.

Autores: Eduardo Morillo y Juan Portero.

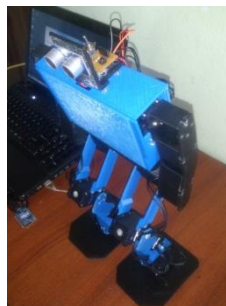


Figura 3.115. Posición inclinada.

Autores: Eduardo Morillo y Juan Portero.

De igual manera si se desea colocar física y manualmente al robot en una posición y luego almacenarla, se debe presionar el botón **relax** para que se retire el torque a los servomotores; a continuación se mueve al robot como se desee y con el botón **capture** se guardan los valores de cada servomotor, estos valores se pueden guardar en una posición nueva la cual se crea con el botón **add**, de esta manera se obtienen varias posiciones.

Si se desea usar varias posiciones a la vez se tiene que utilizar el **editor de secuencias** que se encuentra en la pestaña **tools**, al ingresar a esa opción la pantalla indica la Figura 3.116.

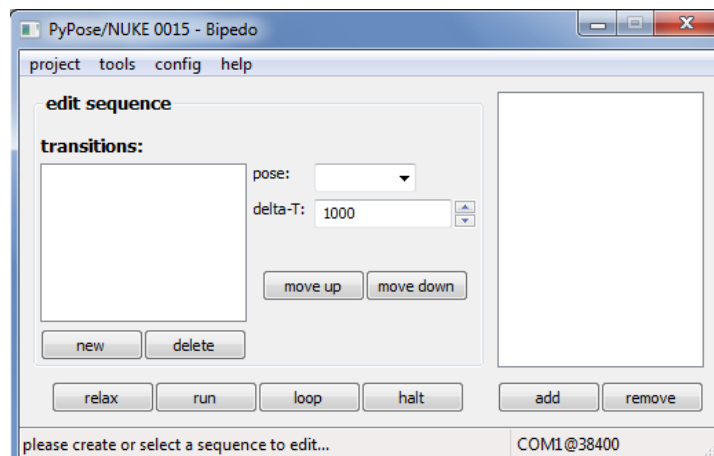


Figura 3.116. Editor de secuencias.

Autores: Eduardo Morillo y Juan Portero.

En esta nueva ventana con el botón **add** se agrega una nueva secuencia, para este ejemplo se ha nombrado como “secuencia” mostrada en la Figura 3.117, a continuación con el botón **new** se agrega una nueva posición y el tiempo que tarda en ejecutarse, en la pestaña **pose** se elige la posición deseada y de esta manera se va armando la secuencia para posteriormente con el botón **run** correr el mini programa una sola vez, con el botón **loop** se ejecuta la secuencia cíclicamente hasta que se presione el botón **halt** para detenerla.

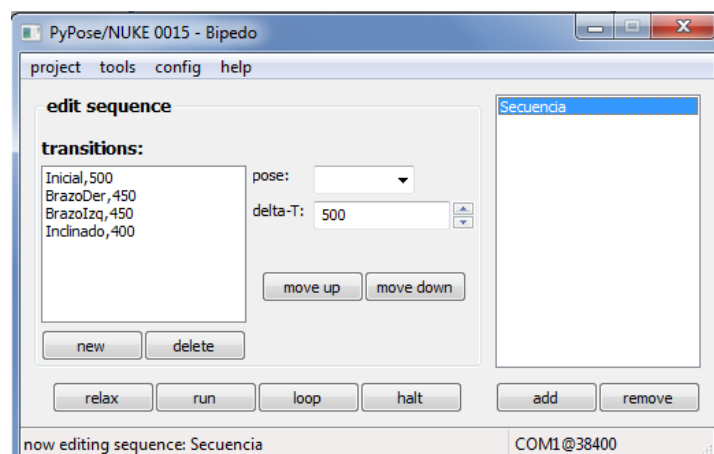


Figura 3.117. Datos de la secuencia.

Autores: Eduardo Morillo y Juan Portero.

De esta manera solo se puede reproducir la secuencia pero sin tener control sobre ella, si se desea que el robot la reproduzca autónomamente o al presionar un pulsador desde el PyPose es imposible, pero con la opción **export to AVR** ubicado en la pestaña **tools** esto es posible, ya que con esta opción se exporta un archivo `.h` como se indica en la Figura 3.118, este archivo se puede utilizar en el entorno Arduino y con esto se tiene un campo muy amplio de posibles aplicaciones, en este caso en el robot bípedo.

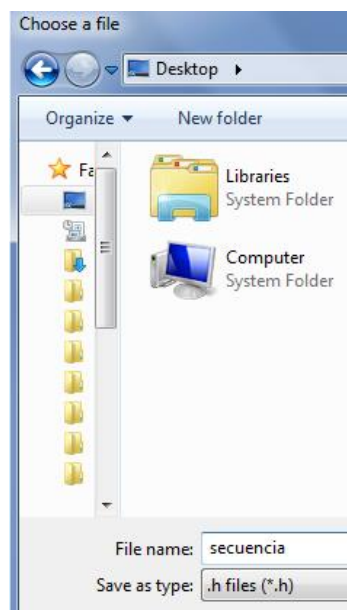


Figura 3.118. Opción export to AVR.

Autores: Eduardo Morillo y Juan Portero.

Al abrir el software Arduino se tiene que guardar un sketch nuevo, en ese instante se crea una carpeta con el nombre que se haya asignado, a continuación se mueve el archivo `secuencia.h` a esta carpeta, de esta manera el entorno Arduino va a poder leer las posiciones cuando se requiera.

Pero para que se puedan utilizar estas posiciones o secuencias se requiere conocer un par de librerías que ya se adjuntaron anteriormente a la carpeta Arduino ubicada en documentos, la librería `ax12.h` que implementa la comunicación entre Arduino y los servomotores, la librería

BioloidController.h que permite cargar las posiciones a la memoria Flash de la tarjeta y la manipulación de las poses o secuencias.

En el sketch de Arduino se ingresa estas librerías como indica la Figura 3.119.

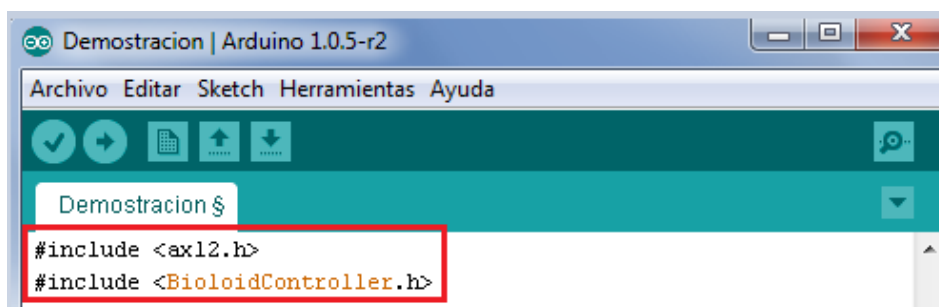


Figura 3.119. Librerías de los servomotores y secuencias en Arduino.

Autores: Eduardo Morillo y Juan Portero.

También se debe incluir el archivo que contiene a las posiciones y secuencias, en la Figura 3.120 se indica lo dicho.

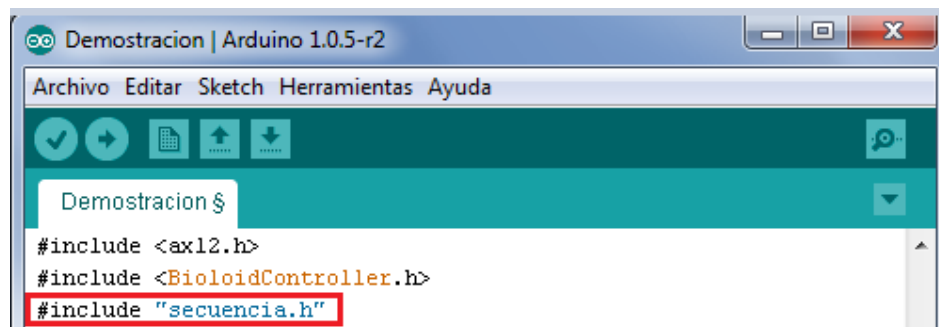


Figura 3.120. Librería de la secuencia en Arduino.

Autores: Eduardo Morillo y Juan Portero.

Para que el programa pueda incluir este archivo como una nueva pestaña dentro del mismo programa, se tiene que guardar y cerrar; al abrir nuevamente Arduino ya incluye el archivo como indica la Figura 3.121.

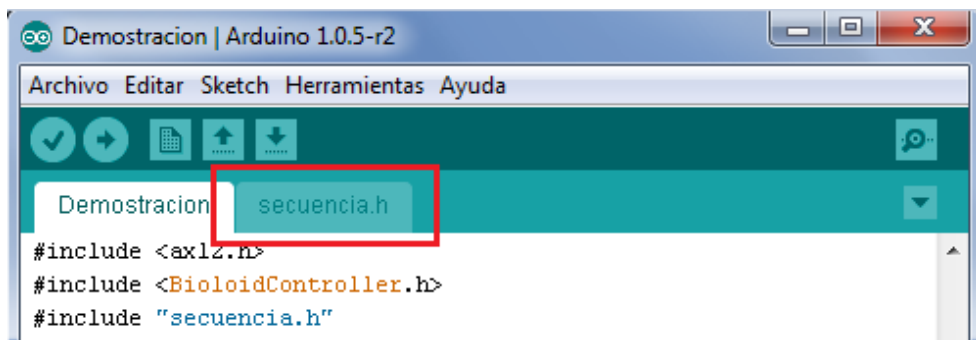


Figura 3.121. Agregar la secuencia en Arduino.

Autores: Eduardo Morillo y Juan Portero.

En esta nueva pestaña (Figura 3.122) se puede apreciar que en las primeras líneas está el nombre con que se creó el robot en PyPose, también se observa las posiciones y las secuencias que se hayan realizado. A este archivo no hay que modificarlo ya que desde el programa principal (Demostracion) se va a llamar a una posición, algunas posiciones o a una secuencia según se requiera.

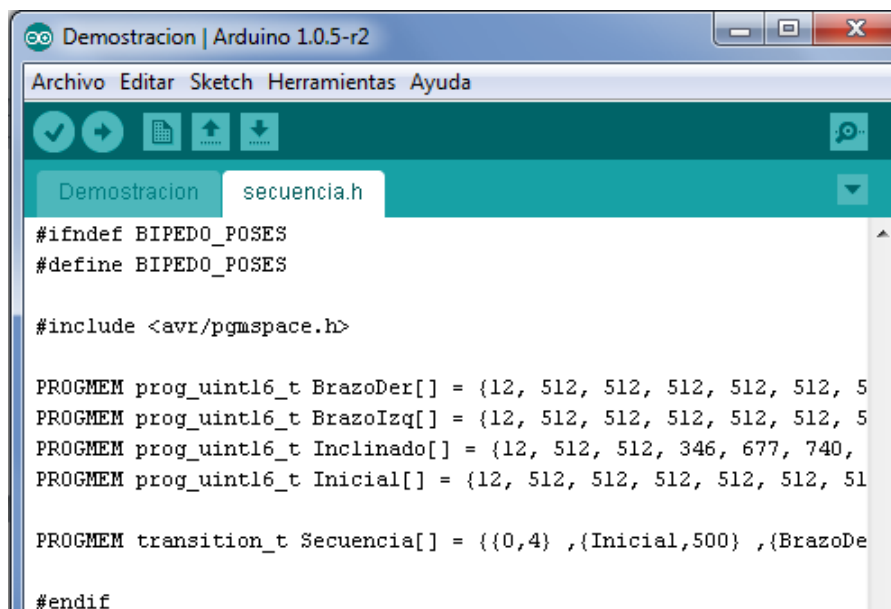
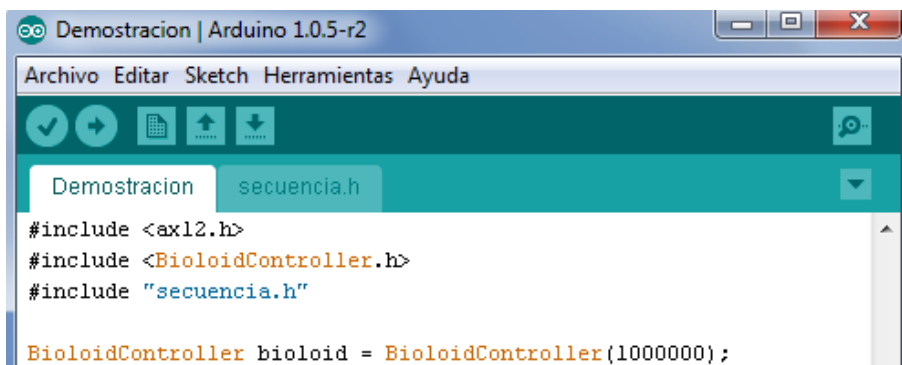


Figura 3.122. Características de la secuencia Arduino.

Autores: Eduardo Morillo y Juan Portero.

Al regresar al programa principal, algo que tampoco tiene que faltar para un correcto funcionamiento es crear la instancia de control de Bioloide,

generalmente este valor es de 1 [Mbps]; en la Figura 3.123 se observa este valor y como es su escritura.



```

Demostracion | Arduino 1.0.5-r2
Archivo Editar Sketch Herramientas Ayuda
Demostracion secuencia.h
#include <ax12.h>
#include <BioloidController.h>
#include "secuencia.h"
BioloidController bioloid = BioloidController(1000000);

```

Figura 3.123. Agregar velocidad de transmisión Arduino.

Autores: Eduardo Morillo y Juan Portero.

En las siguientes líneas de programación se podrán agregar constantes, variables, funciones, operaciones, la utilización de las posiciones y secuencias e infinidad de aspectos según la aplicación deseada.

3.5. OBTENCIÓN DE LAS TRAYECTORIAS DEL ROBOT BÍPEDO.

De la misma manera que el proceso realizado en el literal 3.4 donde se obtuvieron algunas posiciones de ejemplo, se procede a obtener cada una de las posiciones de las trayectorias para la caminata, giro hacia la derecha y giro hacia la izquierda del robot bípedo; estas posiciones serán unificadas en varias secuencias para llevar a cabo lo requerido de la mejor manera posible; no se indicará los valores de los servomotores para cada una de las posiciones ya que es algo metódico, en cambio se mostrará al robot realizando cada una de estas posiciones en las siguientes figuras.

En las Figuras 3.124 y 3.125 se muestran las trayectorias de la caminata del robot bípedo tanto en el plano sagital como en el plano frontal, esta trayectoria consta de diez posiciones distintas.

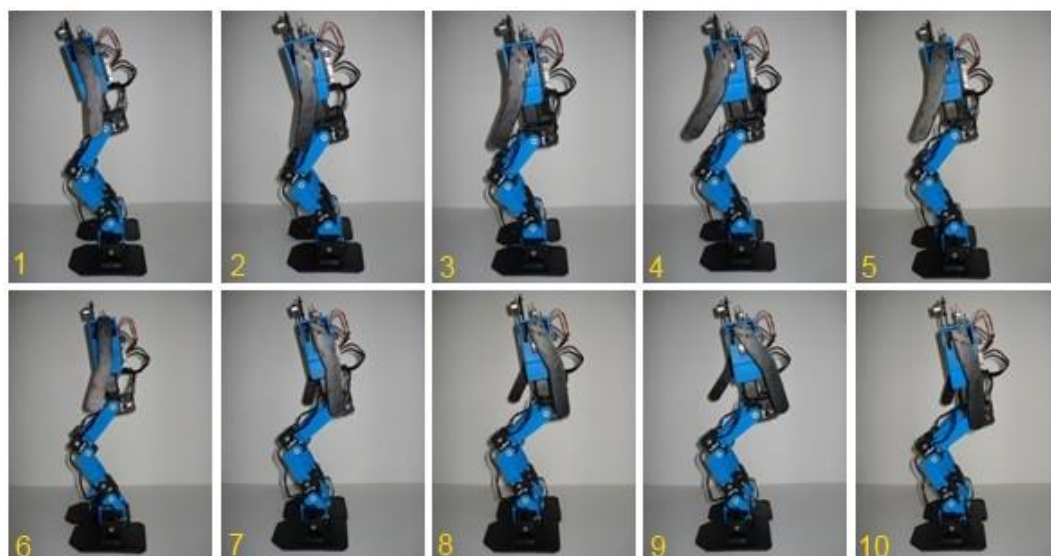


Figura 3.124. Caminata del robot bípedo en el plano sagital.

Autores: Eduardo Morillo y Juan Portero.

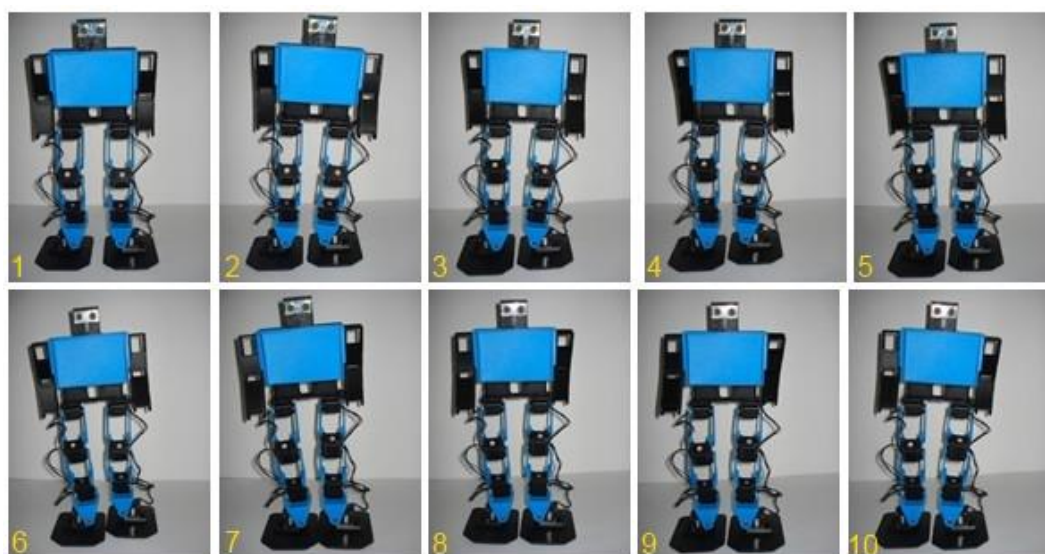


Figura 3.125. Caminata del robot bípedo en el plano frontal.

Autores: Eduardo Morillo y Juan Portero.

En las Figuras 3.126 y 3.127 se muestran las trayectorias del giro hacia la derecha del robot bípedo tanto en el plano sagital como en el plano frontal, esta trayectoria consta de cinco posiciones distintas.



Figura 3.126. Giro a la derecha del robot bípedo en el plano sagital.

Autores: Eduardo Morillo y Juan Portero.

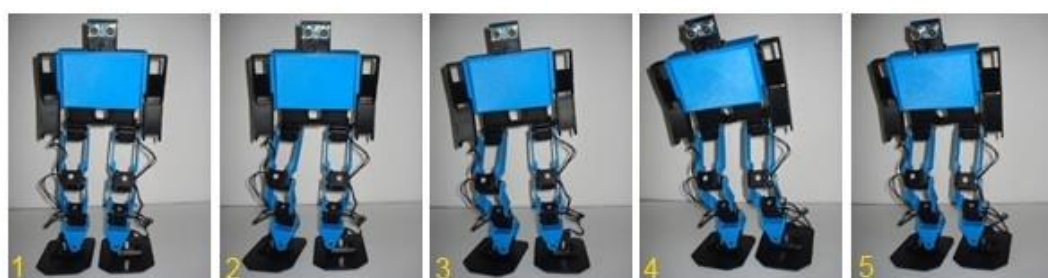


Figura 3.127. Giro a la derecha del robot bípedo en el plano frontal.

Autores: Eduardo Morillo y Juan Portero.

En las Figuras 3.128 y 3.129 se muestran las trayectorias del giro hacia la izquierda del robot bípedo tanto en el plano sagital como en el plano frontal, esta trayectoria consta de cinco posiciones distintas.



Figura 3.128. Giro a la izquierda del robot bípedo en el plano sagital.

Autores: Eduardo Morillo y Juan Portero.

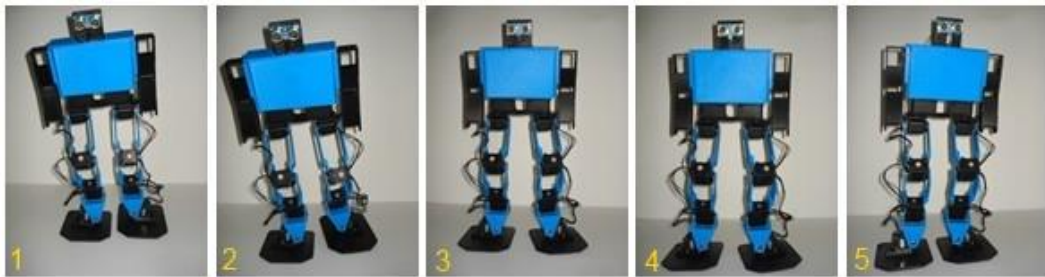


Figura 3.129. Giro a la izquierda del robot bípedo en el plano frontal.

Autores: Eduardo Morillo y Juan Portero.

Cada una de las 20 posiciones obtenidas se ha evaluado individualmente para que no haya problemas de tipo estructural ni ponga en riesgo la integridad del robot bípedo.

Al tener listas todas las posiciones para que el robot cumpla con las trayectorias esperadas, se crean las diferentes secuencias que serán utilizadas por el algoritmo de control.

La secuencia “avanzar” que se muestra en la Figura 3.130 es la encargada de permitir al robot realizar la caminata hacia el frente, se indica las posiciones que la conforman y la velocidad de interpolación entre cada posición.

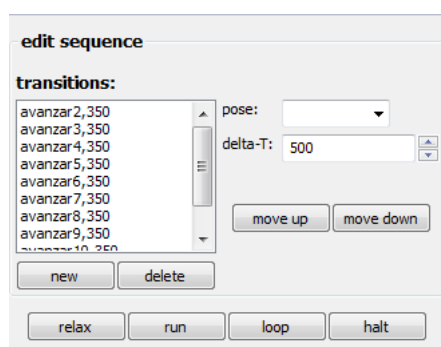


Figura 3.130. Secuencia avanzar.

Autores: Eduardo Morillo y Juan Portero.

La secuencia “giro” que se muestra en la Figura 3.131, permite al robot girar hacia la derecha, al repetir varias veces esta secuencia se obtiene el giro de 90 grados del robot bípedo hacia esa dirección.

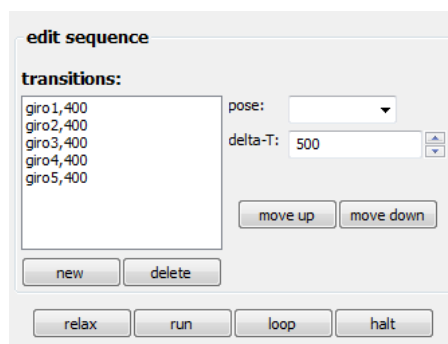


Figura 3.131. Secuencia giro derecha.

Autores: Eduardo Morillo y Juan Portero.

La secuencia “giroizq” que se indica en la Figura 3.132, es la encargada de hacer que el robot gire hacia la izquierda.

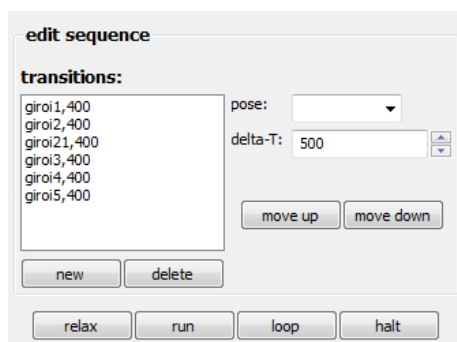


Figura 3.132. Secuencia giro izquierda.

Autores: Eduardo Morillo y Juan Portero.

Estas tres secuencias obtenidas de las trayectorias del robot bípedo son las que utiliza el algoritmo de control.

3.6. ALGORITMO DE CONTROL.

Para la mejor comprensión sobre cómo se desarrolló el algoritmo de control en el IDE de Arduino, se lo ha dividido en varias fracciones de programación como se detalla a continuación.

1. Se incluyen las librerías **#include <ax12.h>**, **#include <BioloidController.h>** y **#include "poses1.h"** como se muestra en la Figura 3.133. Dentro de estas librerías se encuentran los diferentes comandos o sentencias de los dispositivos que conforman el robot. La descripción de estas bibliotecas se encuentra en las páginas 193 y 194.

```
#include <ax12.h>
#include <BioloidController.h>
#include "poses1.h"
```

Figura 3.133. Librerías de Arduino.

Autores: Eduardo Morillo y Juan Portero.

2. Se agrega la velocidad de transmisión a la que trabajará la tarjeta Arbotix-M como se observa en la Figura 3.134.

```
BioloidController bioloid = BioloidController(1000000);
```

Figura 3.134. Velocidad de transmisión.

Autores: Eduardo Morillo y Juan Portero.

3. Se declaran las variables a utilizar dentro del programa y los subprogramas. La Figura 3.135 mostrada a continuación indica el nombre de las variables y su tipo.

```
int botonON = 2;
int leerON = 0;
int datosensor = 0;
int paro = 0;
int arranque = 0;
int caso = 0;
int numero = 0;
//sensor
const int pingPin = 7;
long duracion = 0;
long cm = 0;
```

Figura 3.135. Declaración de variables.

Autores: Eduardo Morillo y Juan Portero.

4. Dentro del **void setup()** se declara a la variable **botonON** como una entrada digital y a continuación se programan las líneas de código que permiten que cuando el switch de la señal de encendido el robot se coloque en la posición Inicial (similar a la posición cuando una persona esta erguida). Este literal se observa en la Figura 3.136.

```

void setup(){
  pinMode(botonON, INPUT);

  delay(100);
  bioloid.loadPose(Inicial);
  bioloid.readPose();
  bioloid.interpolateSetup(500);
  while(bioloid.interpolating > 0){
    bioloid.interpolateStep();
    delay(3);}
  }

```

Figura 3.136. Programación dentro del void setup.

Autores: Eduardo Morillo y Juan Portero.

5. Dentro del **void loop()**, se procede a leer el estado del botón (**botonON**) que permite iniciar la caminata y se lo guarda en la variable **leerON**, seguido se lee el dato del sensor acumulando su valor en la variable **datosensor** mediante la llamada al subprograma **sensor()** que retorna el valor de la distancia que está midiendo el sensor, tercero se realiza una decisión en la cual se pregunta si **leerON** es igual a uno entonces la variable **arranque** se coloca en uno, como cuarto paso se toma otra decisión en la cual se verifica si **arranque** es igual a 1 y la distancia del sensor (**datosensor**) es mayor que cuarenta centímetros se realiza la llamada al subprograma **avanzar()**, quinto mediante una decisión se verifica si el **arranque** es igual a uno y si la distancia (**datosensor**) es menor o igual que 40 entonces la variable **caso** toma un valor al llamar al subprograma **secuencias()**, al contar con este valor se ingresa al **switch** donde dependiendo del valor de **caso** se realiza una acción; en el caso uno se repetirá el subprograma **giro()** 6 veces, en el segundo caso se

realiza el subprograma **giro1()** 10 veces y en el caso 3 el subprograma **giro1()** 4 veces. La programación descrita se indica en la Figura 3.137.

```

void loop(){
  leerON = digitalRead(botonON);
  datosensor = sensor();
  delay(5);
  if(leerON == HIGH){
    arranque = 1;}
  if(arranque == 1 && datosensor > 40){
    avanzar();}
  if(arranque == 1 && datosensor <= 40){
    caso = secuencias();
    switch(caso){
      case 1:
        for(int i=0; i <= 6; i++){
          girar();}
        break;
      case 2:
        for(int i=0; i <= 10; i++){
          girarl();}
        break;
      case 3:
        for(int i=0; i <= 4; i++){
          girarl();}
        break;
    }
  }
}

```

Figura 3.137. Programación dentro del void loop.

Autores: Eduardo Morillo y Juan Portero.

6. Se realizan las líneas de código del subprograma **avanzar()** donde se inicializa la variable **número** en cero, después se ejecuta la secuencia “avanzar” obtenida por medio del programa PyPose que se la ha llamado “a”; esta secuencia se ejecuta infinitamente hasta que deje de ejecutarse el subprograma **avanzar()**. La programación de este literal se muestra en la Figura 3.138.

```

void avanzar(){
  numero = 0;
  bioloid.playSeq(a);
  do{
    bioloid.play();
  }
  while(bioloid.playing);
}

```

Figura 3.138. Subprograma avanzar.

Autores: Eduardo Morillo y Juan Portero.

7. Se detalla las líneas de código del subprograma ***int sensor()*** donde se coloca el modo del pin como salida ***pinMode(pingPin, OUTPUT)*** para enviar un pulso en bajo durante 2 [µs] ***digitalWrite(pingPin,LOW)***, esto asegura que el pulso en alto que se realiza después por 5 [ms] este limpio. Después se vuelve a poner en bajo el pulso y se coloca el ping en modo de entrada con lo cual se procede a medir el pulso de entrada en alto asignándolo a la variable ***duración*** con la cual se realiza una transformación del tiempo en que duro en rebotar la señal convirtiéndola en distancia (centímetros) que es el valor que retorna este subprograma. La programación descrita se observa en la Figura 3.139.

```

int sensor(){
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  pinMode(pingPin, INPUT);
  duracion = pulseIn(pingPin, HIGH);

  cm = duracion/29/2; //convierte tiempo a distancia
  return cm;
}

```

Figura 3.139. Subprograma sensor.

Autores: Eduardo Morillo y Juan Portero.

8. En el subprograma **girar** se procede a colocar las líneas de código para ejecutar la secuencia llamada “giro” obtenida previamente mediante el software PyPose, que permite el giro del robot hacia la derecha. La Figura 3.140 muestra lo detallado en este literal.

```
void girar(){
  bioloid.playSeq(giro);
  do{
    bioloid.play();
  }
  while(bioloid.playing);
}
```

Figura 3.140. Subprograma girar.

Autores: Eduardo Morillo y Juan Portero.

9. El subprograma **girar1** ejecuta la secuencia llamada “giroizq” la cual permite que el robot gire hacia la izquierda, esta secuencia se realizó en PyPose y sus líneas de código se observan en la Figura 3.141.

```
void girarl(){
  bioloid.playSeq(giroizq);
  do{
    bioloid.play();
  }
  while(bioloid.playing);
}
```

Figura 3.141. Subprograma girar1.

Autores: Eduardo Morillo y Juan Portero.

10. En el subprograma **secuencias** mostrado en la Figura 3.142 se realiza una suma para cambiar el valor de la variable **caso** según las decisiones tomadas en el **void loop()**.

```
int secuencias(){
  numero = numero + 1;
  return numero;
}
```

Figura 3.142. Subprograma secuencias.

Autores: Eduardo Morillo y Juan Portero.

De esta manera concluye la implementación del sistema de control que rige sobre el robot bípedo y que permite realizar todas las acciones que han sido previstas para el funcionamiento. En el Anexo B se incluye esta programación realizada en el IDE de Arduino.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

4.1. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA MECÁNICO.

Las pruebas a las que se ha sometido al sistema mecánico se dividen en tres fases, la primera es la prueba de resistencia del material es decir el ABS con el que está construida la estructura del robot bípedo, la segunda prueba es la colocación del robot bípedo en la posición donde los servomotores necesitan entregar más torque para así verificar si puede cumplir con esta posición y la tercera prueba es la velocidad a la que camina el robot bípedo ya que depende de la respuesta de los servomotores y de la forma de los eslabones.

4.1.1. PRUEBAS DE RESISTENCIA DEL MATERIAL ABS.

Este proyecto no se centra en el estudio del ABS ya que éste material con el que está constituida la estructura del robot bípedo es una de varias etapas necesarias para cumplir con la hipótesis y los objetivos del robot bípedo, por esta razón se han tomado los datos del estudio realizado por (Chancusi & Navarrete, 2014, pp. 153-159), donde se realizaron ensayos de tracción en base a la norma ASTM D638-10 y se usaron probetas del Tipo I para polímeros, se somete a cada probeta a un esfuerzo axial de tracción creciente hasta que se produce la rotura, obteniendo los datos de la Tabla 4.1.

Tabla 4.1. Parámetros del ensayo de tracción.

# de probeta	σ [MPa]	δ [mm]	ξ	E [MPa]
1	20.40	0.49	0.010	2082.0
2	20.30	0.36	0.007	2818.1
3	20.31	0.55	0.011	1847.4

Fuente: Chancusi & Navarrete (2014). Diseño y construcción de una impresora 3D auto-replicable controlada inalámbricamente para el prototipado de piezas plásticas de bajo costo, mediante software libre p.155. Latacunga.

Dónde:

- σ : Esfuerzo axial [MPa].
- δ : Diferencia de longitudes final e inicial [mm].
- ξ : Deformación convencional.
- E: Módulo de Young [MPa].

El esfuerzo máximo obtenido en las pruebas del material ABS es de 20.40 [MPa], con este esfuerzo hay una deformación de apenas 0.5 [mm]; el esfuerzo máximo del diseño del robot bípedo es de 11.55 [MPa] (página 71), con lo cual se concluye que la estructura no corre riesgo de ruptura ya que está sometida a un máximo de 56,6% de todo el esfuerzo que resiste el material.

4.1.2. POSICIÓN DONDE SE REQUIERE MAYOR TORQUE.

En el literal 2.2.4 donde se realiza la dinámica del robot bípedo, se tiene la posición donde los servomotores entregan el mayor torque para mantener la posición de la pierna como muestra la Figura 2.37, sobre todo el servomotor de la segunda articulación de la cadera es decir el servomotor q_2 que entrega un torque de 0.2892 [Nm] según lo calculado, al colocar mediante el programa PyPose al robot en esta posición como muestra la Figura 4.1 se observa que el robot mantiene sin problemas esta posición,

pero no mantiene el equilibrio esto debido a que el punto de momento cero se encuentra fuera del polígono formado por el pie que esta en contacto con el piso.



Figura 4.1. Posición donde se requiere mayor torque.

Autores: Eduardo Morillo y Juan Portero.

Se concluye que la estructura resiste estar en esta posición, que es la que mayor fuerza requiere por parte de los servomotores, de esta manera no se tiene fallo en la estructura del robot bípedo.

4.1.3. VELOCIDAD DE DESPLAZAMIENTO DEL ROBOT BÍPEDO.

Este literal comprende de dos objetivos: el primero permite establecer a partir de que velocidad el robot no mantiene el equilibrio debido a la inestabilidad que se genera, el segundo consiste en determinar la velocidad promedio a la cual se desplaza el robot.

La prueba que permite cumplir con el primer objetivo se realiza de la siguiente manera:

Se establece en el programa PyPose el tiempo en el cual se ejecuta cada posición de la secuencia “avanzar”, este dato se ingresa en la casilla ***delta-T***, se muestra en la Figura 4.2 un ***delta-T*** de 500.

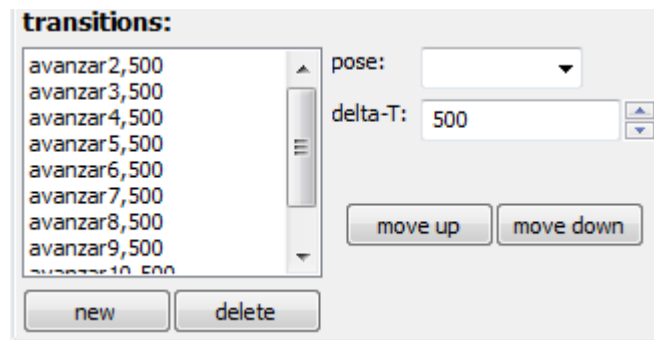


Figura 4.2. Tiempo de ejecución en el programa PyPose.

Autores: Eduardo Morillo y Juan Portero.

Se coloca una regla, cinta métrica o un flexómetro en el piso, extendido de forma que el cero sea desde donde empieza a caminar el robot, para de esta manera medir el desplazamiento realizado por el robot bípedo dentro de un tiempo establecido de 30 [s] (Figura 4.3).

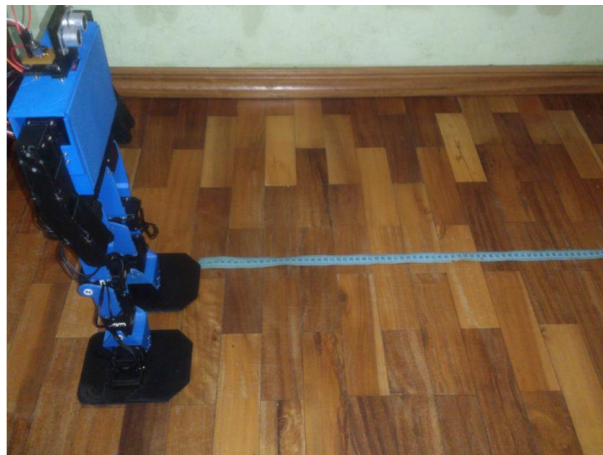


Figura 4.3. Desplazamiento del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Una vez que se cuenta con el tiempo y el desplazamiento que realizó el robot, se calcula la velocidad con la que el robot bípedo camina. A continuación se detallan los cálculos realizados para las pruebas planteadas de la forma descrita anteriormente:

- **Prueba 1.**

Tiempo en el PyPose = 500 [ms]

Tiempo de la prueba = 30 [s]

Espacio recorrido = 23.5 [cm]

$$v_1 = \frac{e}{t} = \frac{23.5 [cm]}{30[s]} = 0.78 \left[\frac{cm}{s} \right]$$

- **Prueba 2.**

Tiempo en el Pypose = 400 [ms]

Tiempo de la prueba = 30 [s]

Espacio recorrido = 32 [cm]

$$v_1 = \frac{e}{t} = \frac{32 [cm]}{30[s]} = 1.06 \left[\frac{cm}{s} \right]$$

- **Prueba 3.**

Tiempo en el Pypose = 350 [ms]

Tiempo de la prueba = 30 [s]

Espacio recorrido = 37 [cm]

$$v_1 = \frac{e}{t} = \frac{37 [cm]}{30[s]} = 1.23 \left[\frac{cm}{s} \right]$$

- **Prueba 4.**

Tiempo en el Pypose = 300 [ms]

Tiempo de la prueba = 30 [s]

Espacio recorrido = 42 [cm]

$$v_1 = \frac{e}{t} = \frac{42 [cm]}{30[s]} = 1.4 \left[\frac{cm}{s} \right]$$

- **Prueba 5.**

Tiempo en el Pypose = 250 [ms]

Tiempo de la prueba = 30 [s]

Espacio recorrido = 46.5 [cm]

$$v_1 = \frac{e}{t} = \frac{46.5 [cm]}{30[s]} = 1.55 \left[\frac{cm}{s} \right]$$

Una vez obtenidos los resultados de las pruebas realizadas; se establece que a velocidades mayores a 1.4 [cm/s] el robot se vuelve muy inestable, debido a que mientras más rápido se mueven los servomotores estos no llegan exactamente a la posición deseada, sino que conforme sigue el caminado se pierden algunos grados deformando de esta manera las trayectorias trazadas para el caminado del robot bípedo. La Figura 4.4 muestra la gráfica de la velocidad vs la estabilidad para este prototipo de robot bípedo.

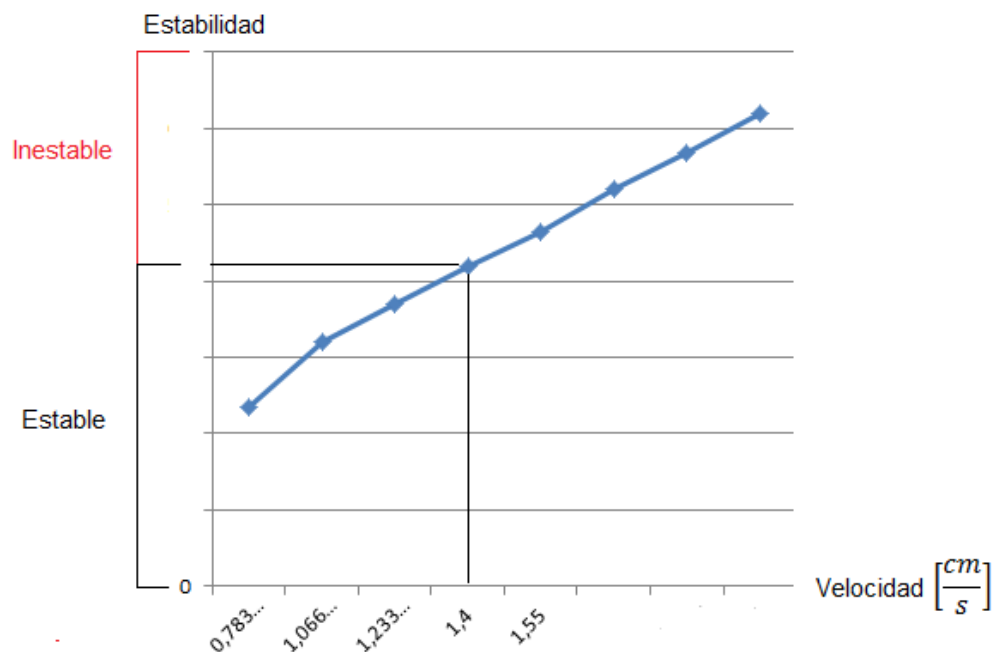


Figura 4.4 Velocidad vs Estabilidad.

Autores: Eduardo Morillo y Juan Portero.

Para cumplir con el segundo objetivo, la prueba se realiza de forma similar a la primera, con la diferencia que los valores establecidos para el tiempo del programa PyPose y el tiempo de la prueba se mantienen, para obtener el desplazamiento del robot 5 veces y a su vez la velocidad. Una vez

obtenida la Tabla 4.2 se realiza un promedio de la velocidad, con lo cual, se determina la velocidad promedio a la que se desplaza el robot.

Tabla 4.2. Velocidad promedio.

Tiempo PyPose	Tiempo Prueba	Desplazamiento	Velocidad
500 [ms]	30 [s]	23.5[cm]	0.78 $\left[\frac{cm}{s}\right]$
500 [ms]	30 [s]	23.7[cm]	0.79 $\left[\frac{cm}{s}\right]$
500 [ms]	30 [s]	24[cm]	0.80 $\left[\frac{cm}{s}\right]$
500 [ms]	30 [s]	23.45[cm]	0.78 $\left[\frac{cm}{s}\right]$
500 [ms]	30 [s]	23.2[cm]	0.77 $\left[\frac{cm}{s}\right]$

Autores: Eduardo Morillo y Juan Portero.

$$\text{Velocidad promedio} = \frac{(0.78+0.79+0.80+0.78+0.77)}{5} \left[\frac{cm}{s}\right]$$

$$\text{Velocidad promedio} = 0.78 \left[\frac{cm}{s}\right]$$

Una vez concluida la prueba determinando como resultado de esta prueba que el robot se mueve a una velocidad promedio de $0.78 \left[\frac{cm}{s}\right]$.

4.2. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA ELECTRÓNICO.

El objetivo de las pruebas que van a realizarse en el sistema electrónico tiene como fin comprobar el correcto funcionamiento de los dispositivos. Las pruebas que se realizan en el sistema electrónico son: medición y comprobación de los voltajes entregados por los distintos elementos, duración de la batería, detección de obstáculos por parte del sensor y la distancia de sensado.

4.2.1. COMPROBACIÓN DE VOLTAJES.

1. Como primer paso se comprueba el voltaje entregado por la batería que es la fuente de energía, con el multímetro se mide el voltaje como indica la Figura 4.5, el valor ideal es de 12 [V] pero al tratarse

de una batería al estar totalmente cargada ésta tiene un poco más de voltaje de lo ideal; además poco a poco se descarga, en este caso se tendría que volver a cargarla ya que los servomotores correrían riesgo de daño por trabajar con bajo voltaje.



Figura 4.5. Medición de voltaje en la batería.

Autores: Eduardo Morillo y Juan Portero.

2. Se conecta el positivo de la batería en el lugar destinado del circuito de accionamiento y se comprueba el voltaje que entrega el pin de salida del switch, en los dos estados del mismo, como indican las Figuras 4.6 y 4.7.

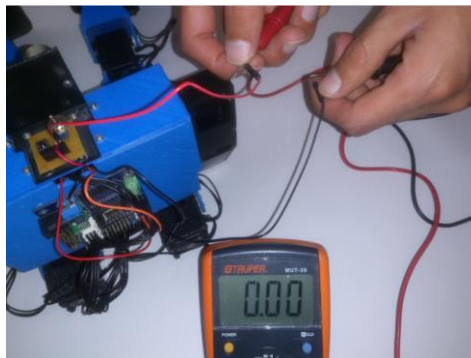


Figura 4.6. Medición de voltaje del switch (estado OFF).

Autores: Eduardo Morillo y Juan Portero.

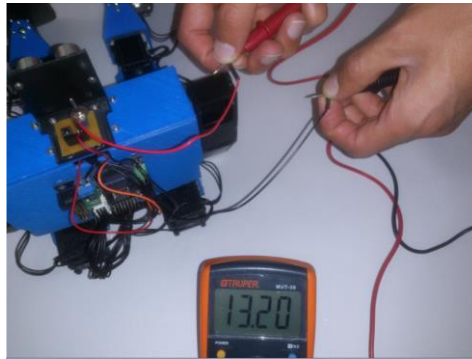


Figura 4.7. Medición de voltaje del switch (estado ON).

Autores: Eduardo Morillo y Juan Portero.

3. El pin de salida del switch se conecta a la entrada positiva de la tarjeta electrónica, de igual manera el cable negativo de la batería se conecta a la entrada negativa de la tarjeta electrónica, de esta manera ya se utiliza el circuito del switch permitiendo o no el paso de energía hacia la tarjeta como lo indica la Figura 4.8.

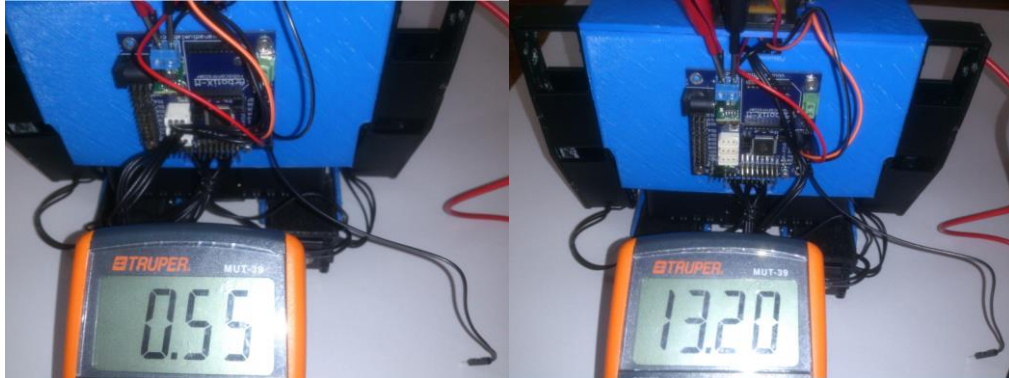


Figura 4.8. Voltaje en la entrada de la batería con los estados del switch.

Autores: Eduardo Morillo y Juan Portero.

4. Desde la tarjeta electrónica se lleva al circuito de accionamiento 5 [V] y tierra por medio de dos cables para que funcione la etapa del botón, cada cable se conecta en el sitio indicado, como se muestra en la Figura 4.9; a continuación se mide el voltaje en el pin de salida de señal sin presionar el botón como indica la Figura 4.10 y al presionar el botón como se indica en la Figura 4.11.

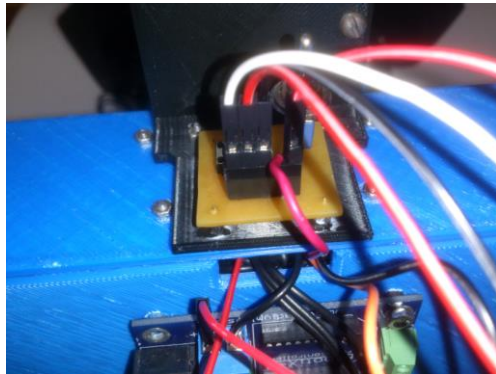


Figura 4.9. Conexión de la etapa del botón.

Autores: Eduardo Morillo y Juan Portero.

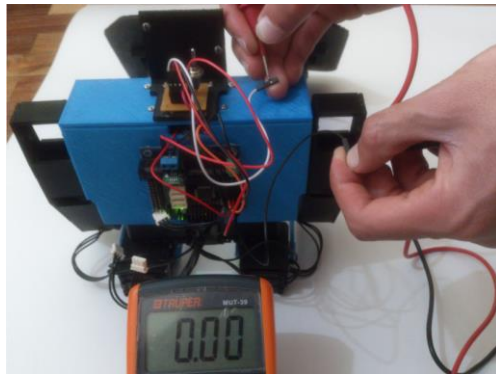


Figura 4.10. Medición de voltaje en la salida del botón sin presionar.

Autores: Eduardo Morillo y Juan Portero.

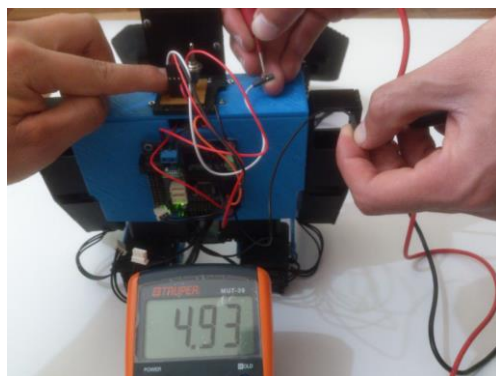


Figura 4.11. Medición de voltaje en la salida del botón.

Autores: Eduardo Morillo y Juan Portero.

Al concluir las pruebas de la medición de los voltajes, el resultado obtenido es que estos valores son los correctos para permitir el funcionamiento del robot sin ningún inconveniente.

4.2.2. DURACIÓN DE LA BATERÍA.

El objetivo de esta prueba es comprobar cuanto tiempo dura la batería suministrando el voltaje suficiente a los servomotores y distintos componentes electrónicos para el funcionamiento del robot bípedo, para realizar esta prueba se mide el voltaje cada 10 [min] durante 70 [min] con la batería totalmente cargada y el robot en pleno funcionamiento de caminado, al realizar esta prueba se describe anteriormente se obtiene la Tabla 4.3.

Tabla 4.3. Resultados de la duración de la batería.

Tiempo	Voltaje Medido	Funciona el Robot	Porcentaje de descarga
10 [min]	12.3 [V]	Si	100%
20 [min]	11.8 [V]	Si	98.33%
30 [min]	10.4 [V]	Si	86.67%
40 [min]	9.1 [V]	Si	75.83%
50 [min]	8.6 [V]	Si	71.67%
60 [min]	7.5 [V]	No	62.5%

Autores: Eduardo Morillo y Juan Portero.

De la Tabla 4.3 se deduce que el tiempo en el que se descarga la batería de forma real se diferencia del obtenido mediante los cálculos de la página 132, del capítulo de diseño, el tiempo obtenido mediante cálculos es de 73.2 [min] y el obtenido cuando el robot bípedo está funcionando es de 60 [min], con lo cual la diferencia de tiempo es de 13.2 [min], ésta diferencia se da debido a que en los cálculos se tiene un funcionamiento ideal de los componentes donde la batería va a estar en una descarga constante y no paulatina como es de forma real.

4.2.3. DETECCIÓN DE OBSTÁCULOS.

En este literal el objetivo es poner a prueba el sensor de ultrasonido, para verificar si la constitución, el color o la textura de los obstáculos afectan la detección de los mismos. La prueba se realiza colocando al frente del robot diferentes tipos de obstáculos y mediante un tabla se detalla si el sensor fue o no capaz de sensar los obstáculos.

A continuación se enlistan los diferentes casos que fueron tomados para realizar esta prueba.

1. Se coloca al robot frente a una pared de color verde clara a una distancia de 40 [cm] aproximadamente como se muestra en la Figura 4.12.



Figura 4.12. Obstáculo pared color verde claro.

Autores: Eduardo Morillo y Juan Portero.

2. El obstáculo que se ubica frente al robot a una distancia de 40 [cm] aproximadamente es madera color café oscuro (Figura 4.13).

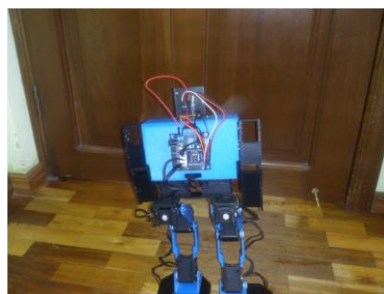


Figura 4.13. Obstáculo madera color café oscuro.

Autores: Eduardo Morillo y Juan Portero.

3. El material que se coloca frente al robot en este literal a una distancia 40 [cm] aproximadamente es un espejo tal como se muestra en la Figura 4.14.



Figura 4.14. Obstáculo espejo.

Autores: Eduardo Morillo y Juan Portero.

4. Se ubica una cartulina de color blanco como obstáculo para el robot (Figura 4.15).



Figura 4.15. Obstáculo cartulina color blanco.

Autores: Eduardo Morillo y Juan Portero.

5. Se coloca un obstáculo de color negro, como se muestra en la Figura 4.15.



Figura 4.15. Obstáculo color negro.

Autores: Eduardo Morillo y Juan Portero.

Tabla 4.4. Resultados de la detección de obstáculos.

Caso	Si detecta	No detecta
1 (pared color verde claro).	X	
2 (madera color café oscuro).	X	
3 (espejo).	X	
4 (cartulina color blanco).	X	
5 (obstáculo de color negro).	X	

Autores: Eduardo Morillo y Juan Portero.

En base a los datos obtenidos de la Tabla 4.4, se obtiene como resultado, que el sensor es capaz de detectar todos los obstáculos que se le ha puesto en frente en los diferentes casos, esto debido a que el sensor emite una señal de sonido y en todos estos materiales el sonido rebotó permitiendo al sensor obtener la distancia que separa al robot bípedo del obstáculo.

4.2.4. DISTANCIA DE SENSADO.

Debido a que en el sistema de control se establece una distancia adecuada para que el robot realice el giro se proceden a realizar varias pruebas que tiene por objetivo verificar la distancia real a la cual el sensor detiene el robot bípedo, para poder realizar el giro; a continuación se muestra la Figura 4.17, dónde se encuentra las pruebas realizadas; la distancia real se establece mediante la medición con una cinta métrica ubicada en el piso entre el robot y el obstáculo.

**Figura 4.17. Distancia de sensado.**

Autores: Eduardo Morillo y Juan Portero.

En base a las pruebas realizadas se obtiene la Tabla 4.5 en la cual se muestra los valores medidos junto con el establecido por software.

Tabla 4.5. Resultados de la distancia de sensado.

Número de Prueba	Distancia Establecida [cm]	Distancia Medida [cm]
1	40	38
2	40	39
3	40	38.5
4	40	39,3
5	40	38,8

Autores: Eduardo Morillo y Juan Portero.

De la Tabla 4.5 se obtiene como resultado, que se produce un pequeño error en la distancia a la cual se requiere que el sensor detenga el robot, se establece que este error se produce debido al tiempo de retardo que se le da al mismo para que el subprograma *sensor()* retorne el dato hacia el programa.

4.3. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA DE CONTROL.

Una vez establecido el algoritmo de control que regirá el funcionamiento del robot bípedo (literal 3.6), se comprueba que cada una de las acciones se realicen según lo planeado, para lo cual, se plantean los siguientes objetivos: encontrar y solucionar los problemas que se producen en el algoritmo de control y probar que todas las acciones de control sean realizadas correctamente por el robot.

4.3.1. ERRORES DEL ALGORITMO DE CONTROL.

Para determinar los errores que se producen en el sistema de control y así cumplir con el primer objetivo planteado se procede a probar el robot bípedo con el algoritmo de control en la tarjeta; los posibles errores que se produzcan serán enlistados conjuntamente con la solución que se ha tomado para los mismos.

Durante las pruebas de funcionamiento del sistema de control se produjeron los siguientes errores:

- La secuencia de caminado solo se repetía una vez por lo cual se procedió a verificar porque el ciclo no se repetía en un bucle infinito, encontrando el error en la variable que servía para que el ciclo se repita; la solución para este error fue colocar una variable propia de la biblioteca **bioloid robocontroller** dedicada para este fin (Figura 4.18).

Error	Corrección
<pre>void avanzar(){ numero = 0; bioloid.playSeq(a); do{ bioloid.play(); } while(paros == 1); }</pre>	<pre>void avanzar(){ numero = 0; bioloid.playSeq(a); do{ bioloid.play(); } while(bioloid.playing); }</pre>

Figura 4.18. Error 1 del algoritmo de control.

Autores: Eduardo Morillo y Juan Portero.

- Durante el giro hacia la derecha se notó que el robot debía repetir un mayor número de veces la secuencia del giro pues con las establecidas por primera vez no se lograba alcanzar esquivar el obstáculo. La solución se muestra en la Figura 4.19.

Error	Corrección
<pre>case 1: for(int i=0; i <= 4; i++){ girar();} break;</pre>	<pre>case 1: for(int i=0; i <= 6; i++){ girar();} break;</pre>

Figura 4.19. Error 2 del algoritmo de control.

Autores: Eduardo Morillo y Juan Portero.

- Un nuevo error se produjo en el giro hacia la izquierda donde el número de veces con que se repetía la secuencia de giro no era el

adecuado para lograr esquivar los obstáculos cuando se encontraban presentes al frente del robot y a su derecha; para lo cual, se aumentó las repeticiones del ciclo repetitivo (Figura 4.20).

Error	Corrección
<pre>case 2: for(int i=0; i <= 5; i++){ girarl();} break;</pre>	<pre>case 2: for(int i=0; i <= 10; i++){ girarl();} break;</pre>

Figura 4.20. Error 3 del algoritmo de control.

Autores: Eduardo Morillo y Juan Portero.

4.3.2. ACCIONES DEL SISTEMA DE CONTROL.

El objetivo de este apartado es realizar pruebas del funcionamiento del robot bípedo para verificar si cada una se está cumpliendo a cabalidad, la forma para realizar las distintas pruebas y el resultado obtenido se detallan a continuación:

1. La primera prueba es colocar al robot en posición erguida al cambiar el estado del switch para que permita el paso de energía. El resultado obtenido es satisfactorio pues el robot cumple correctamente la acción del sistema de control (Figura 4.21).

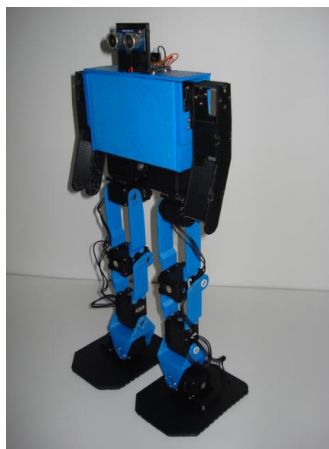


Figura 4.21. Robot bípedo en la posición erguida.

Autores: Eduardo Morillo y Juan Portero.

2. La segunda prueba del sistema de control, es que el robot empiece su marcha hacia el frente al presionar el botón. Esta acción se ejecuta de forma correcta como se observa en la Figura 4.22. Donde se ha representado al robot bípedo por medio de un círculo y la dirección por medio de una línea roja, esto se ha realizado para que en los próximos pasos se comprenda de una manera gráfica lo que se citará en los siguientes puntos.



Figura 4.22. Robot bípedo caminando.

Autores: Eduardo Morillo y Juan Portero.

3. La tercera prueba consiste en que el robot debe girar hacia la derecha al tener un obstáculo en frente a una distancia de 40 [cm] aproximadamente, al probar esta acción el resultado es favorable como muestra la Figura 4.23, donde el robot gira al detectar el obstáculos, si al terminar de girar 90 grados hacia la derecha el robot no detecta otro obstáculo a 40 [cm] éste continuará con el caminado del paso 2.

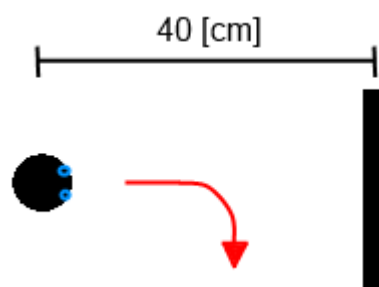


Figura 4.23. Robot bípedo curvando hacia la derecha 90 grados.

Autores: Eduardo Morillo y Juan Portero.

4. La cuarta prueba sirve para verificar, si el robot gira hacia la izquierda si tiene un nuevo obstáculo al frente a una distancia aproximada de 40 [cm], esta acción se ejecuta teniendo un resultado favorable como

se muestra en la Figura 4.24. Si al terminar de girar 180 grados hacia la izquierda el robot no detecta otro obstáculo a 40 [cm] éste continuará con el caminado.

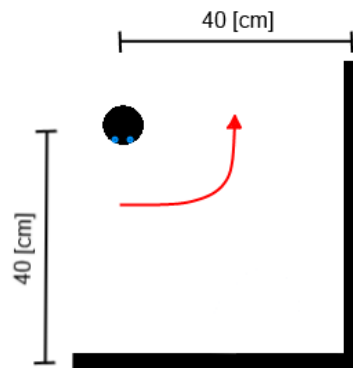


Figura 4.24. Robot bípido curvando hacia la izquierda 180 grados.

Autores: Eduardo Morillo y Juan Portero.

5. La quinta prueba en determinar si el robot gira nuevamente hacia la izquierda si tiene un nuevo obstáculo al frente a una distancia de 40 [cm] aproximadamente, esta acción se ejecuta de forma coherente como se muestra en la Figura 4.25. Si al terminar de girar 90 grados hacia la izquierda el robot no detecta otro obstáculo a 40 [cm], éste continuará con el caminado.

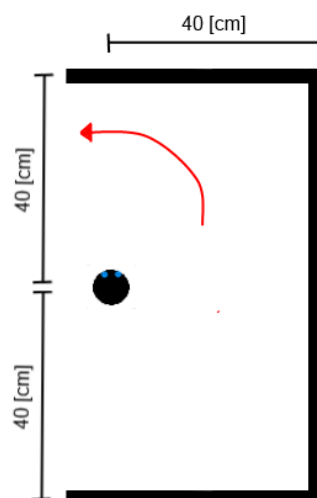


Figura 4.25. Robot bípido curvando hacia la izquierda 90 grados.

Autores: Eduardo Morillo y Juan Portero.

4.4. PRUEBAS GENERALES DURANTE EL FUNCIONAMIENTO.

Al tener unificados todos los sistemas que conforman el prototipo de robot bípedo, además de realizadas las pruebas pertinentes, se realizan las pruebas generales, en las que el robot tiene que actuar como lo requerido y diseñado en los capítulos anteriores.

1. La primera prueba que se realizó fue que el robot al ser encendido mediante el switch se ponga en la posición erguido, a continuación se inició la marcha de manera autónoma al pulsar el botón, el robot bípedo caminó hacia el frente hasta que encontró un obstáculo a una distancia aproximada de 40 [cm], al detectar el obstáculo procedió a girar a la derecha 90 grados como se esperaba, al no tener otro obstáculo a 40 [cm] de distancia procedió con el caminado hacia el frente. A continuación se indican varias fotografías donde se aprecia lo dicho en este párrafo (Figura 4.26).

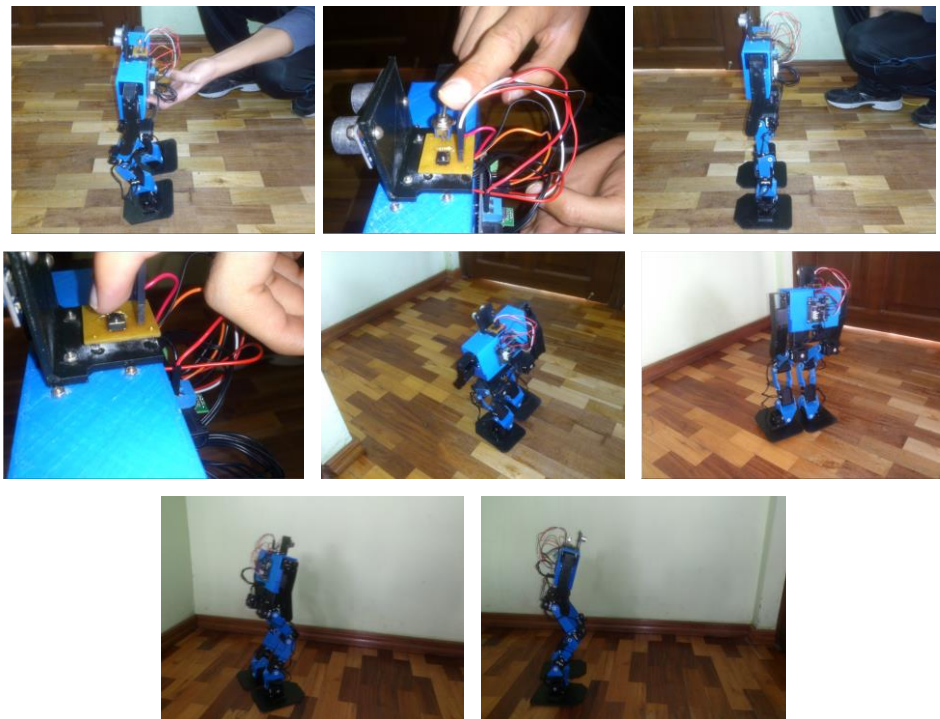


Figura 4.26. Prueba uno de la caminata del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

2. En la segunda prueba el robot bípedo tenía en frente y a su derecha un obstáculo, respectivamente; ambos a una distancia aproximada de 40 [cm]. Al detectar el obstáculo del frente, el robot procedió a girar hacia la derecha, una vez realizado el giro detectó el siguiente obstáculo y procedió a girar 180 grados hacia su izquierda como lo esperado, al realizar el giro y no encontrarse con un nuevo obstáculo frente del robot el mismo continuó con la caminata. A continuación se indican varias fotografías de este procedimiento (Figura 4.27).



Figura 4.27. Prueba dos de la caminata del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

3. En la tercera prueba el robot bípedo tenía en frente, derecha e izquierda un obstáculo respectivamente, los tres a una distancia aproximada de 40 [cm], al detectar el primer y segundo obstáculo el robot realizó los giros respectivos, al detectar el tercer obstáculo el robot giró 90 grados hacia su izquierda como lo esperado, al no volver a encontrarse con ningún obstáculo el robot bípedo continuó con la

caminata hacia el frente. A continuación se indican varias fotografías de lo explicado anteriormente (Figura 4.28).



Figura 4.28. Prueba tres de la caminata del robot bípedo.

Autores: Eduardo Morillo y Juan Portero.

Al concluir con las pruebas realizadas a todos los sistemas del robot bípedo, se obtiene como resultado el correcto funcionamiento del robot y por lo tanto se da por finalizado el capítulo y el proyecto.

4.5. VALIDACIÓN DE LA HIPÓTESIS.

La hipótesis planteada para el presente proyecto fue:

¿El diseño y la construcción del prototipo de robot bípedo autónomo con desplazamiento dinámico permitirán que éste sea capaz de caminar manteniendo el equilibrio y detectar los obstáculos presentes en su trayectoria?

La comprobación de la misma se la realiza por medio de la prueba de chi-cuadrado (Suárez, 2012), esta prueba tiene la siguiente ecuación:

$$\lambda^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad \text{Ec.4.1}$$

Dónde:

- λ^2 : Chi cuadrado.
- O_i : Es la frecuencia absoluta observada o empírica.

- E_i : Es la frecuencia esperada o deseada.

Se tiene que comprender que el termino frecuencia para este proyecto se refiere a valores numéricos.

A continuación se realizan los cálculos respectivos para encontrar los valores necesarios de la ecuación 4.1, para obtener el valor de chi-cuadrado y de esta manera comparar éste valor con los de la Tabla del Anexo C, para de esta manera validar la hipótesis nula (H_0) o la hipótesis alternativa (H_1).

En este proyecto hay dos variables dependientes que son el equilibrio y detección de obstáculos por ende se van a realizar dos cálculos distintos para validar estas dos hipótesis.

Los datos de las Tablas 4.6 – 4.9 son las pruebas que se realizaron al robot bípedo. El nivel significativo para el cual se va a trabajar en los dos casos es del 95%, debido a que se espera una alta confiabilidad de rendimiento del prototipo de robot bípedo, no se da el 100% de confiabilidad porque siempre se espera algún tipo de falla en un prototipo.

Variable dependiente: Equilibrio.

H_0 : El robot no pierde el equilibrio durante cada una de las acciones que realiza o en todas las acciones se mantiene de igual forma el equilibrio.

H_1 : El robot tiende a perder equilibrio durante todas las acciones.

Tabla 4.6. Datos observados para el equilibrio.

Mantiene el equilibrio	Durante la Caminata	En el giro de 90° a la derecha.	En el giro de 90° a la izquierda.	En el giro de 180° a la izquierda.	TOTAL
SI	85	72	50	80	287
NO	15	8	10	20	53
TOTAL	100	80	60	100	340

Autores: Eduardo Morillo y Juan Portero.

Tabla 4.7. Datos esperados para el equilibrio.

	<i>i</i>	O	E	O-E	$(O - E)^2$	$\frac{(O - E)^2}{E}$
Durante la Caminata. SI	1	85	84,411	0,588	0,346	0,004
Durante la Caminata. NO	2	72	67,529	4,470	19,986	0,295
Durante el giro a la derecha. SI	3	50	50,647	0,647	0,418	0,008
Durante el giro a la derecha. NO	4	80	84,411	-4,411	19,463	0,23
Durante el giro a la izquierda 90°. SI	5	15	15,588	-0,588	0,346	0,022
Durante el giro a la izquierda 90°. NO	6	8	12,470	-4,470	19,986	1,602
Durante el giro a la izquierda 180°. SI	7	10	9,352	0,647	0,418	0,0447
Durante el giro a la izquierda 180°. NO	8	20	15,588	4,411	19,463	1,248
					λ^2	3,45714724

Autores: Eduardo Morillo y Juan Portero.

$$\text{Grados de libertad} = (2 - 1) * (4 - 1) = 3$$

Con los grados de libertad y con un nivel significativo de 0,05 se busca en la Tabla del Anexo C, referente a la distribución del chi-cuadrado, obteniendo un valor de 12,8381 para λ^2 .

El resultado obtenido (3,4571) es menor del que debería ser el valor de chi cuadrado (12,8381), por lo que p es menor de 0,05% y por lo tanto se acepta la hipótesis nula en la cual el robot no pierde el equilibrio, ni en la caminata, ni en cada giro que realiza; se dice que el robot mantiene el equilibrio de la misma manera en todas las acciones.

Variable dependiente: Detección de obstáculos.

Los datos que se toman para hacer la prueba del chi-cuadrado de la detección de obstáculos son los mismos que para la caminata pues resultaría muy reiterativo realizar las pruebas una y otra vez dado que el robot cumple las acciones de forma secuencial.

H_0 : No existe fallo en la detección de obstáculos por parte del robot en su trayectoria.

H_1 : El robot falla en la detección de obstáculos en su trayectoria.

Tabla 4.8. Datos observados para la detección de obstáculos.

Detecta el obstáculo	Durante la Caminata	Después de girar 90 ° a la derecha.	Después de girar 90 ° a la izquierda.	Después de girar 180 ° a la izquierda.	TOTAL
SI	85	72	50	80	287
NO	15	8	10	20	53
TOTAL	100	80	60	100	340

Autores: Eduardo Morillo y Juan Portero.

Tabla 4.9. Datos esperados para la detección de obstáculos.

	<i>i</i>	O	E	O-E	$(O - E)^2$	$\frac{(O - E)^2}{E}$
Durante la Caminata SI	1	85	84,411	0,588	0,346	0,004
Durante la Caminata NO	2	72	67,529	4,470	19,986	0,295
Después de girar 90 ° a la derecha. SI	3	50	50,647	0,647	0,418	0,008
Después de girar 90 ° a la derecha. NO	4	80	84,411	-4,411	19,463	0,23
Después de girar 90 ° a la izquierda. SI	5	15	15,588	-0,588	0,346	0,022
Después de girar 90 ° a la izquierda. NO	6	8	12,470	-4,470	19,986	1,602
Después de girar 180 ° a la derecha. SI	7	10	9,352	0,647	0,418	0,0447
Después de girar 180 ° a la derecha. NO	8	20	15,588	4,411	19,463	1,248
					λ^2	3,4571472

Autores: Eduardo Morillo y Juan Portero.

$$\text{Grados de libertad} = (2 - 1) * (4 - 1) = 3$$

Con los grados de libertad y con el nivel significativo de 0,05 se busca en la Tabla del Anexo C, referente a la distribución del chi-cuadrado, obteniendo un valor de 12,8381 para λ^2 .

El resultado obtenido (3,4571) es menor del que debería ser el valor de chi cuadrado (12,8381), por lo que p es menor de 0,05% y, por lo tanto, se acepta la hipótesis nula en la cual no existe fallo en la detección de obstáculos por parte del robot bípedo en su trayectoria.

Mediante los resultados obtenidos de la prueba de chi-cuadrado referente al equilibrio y a la detección de obstáculos los cuales confirman que el robot funciona de manera correcta. Se concluye que se construyó un robot capaz de mantener el equilibrio durante la caminata y detectar obstáculos a una distancia de 40 [cm] aproximadamente, además la caminata dinámica se realiza sin detenerse; cumpliendo de esta manera con la hipótesis planteada para la realización de este proyecto.

4.6. ESTUDIO ECONÓMICO.

Una vez implementado el robot de forma exitosa y cumpliendo con su funcionamiento, se hace necesario cuantificar su valor económico para identificar la inversión realizada y así poder compararla con legos o robots existentes en el mercado, dando un precedente para proyectos futuros que se desarrollen para el laboratorio de Robótica Industrial de la ESPE Extensión Latacunga.

Para la implementación del proyecto se empleó materiales electrónicos, dispositivos de control y procesos de maquinado los mismos que se detallan en la Tabla 4.10. Los valores se han obtenido el día lunes 21 de Julio del 2014. Los elementos marcados con el símbolo *, han sido adquiridos en los Estados Unidos ya que en el país tienen un alto costo y algunos no se comercializan en el país.

Tabla 4.10. Costos del proyecto.

Descripción	Cantidad	P. Unit.	Total
*Servomotores AX-12A	12	44.90	538.8
*Tarjeta Arbotix-M	1	39.95	39.95
*Sensor Parallax PING	1	30	30
*Xbee S1	2	21.95	43,9
*UartSBee V4	1	20.9	20,9
*Cargador Tenergy	1	18.50	18.50
*Bateria 12 [V] Tenergy 2000 [mAh] NiMH	1	23.95	23.95
Piezas construidas en ABS	24	-	200
Material electrónico	1	3	3
Envios dentro de los Estados Unidos	1	22.33	22.33
Envios hacia el Ecuador	1	32.35	32.35
Total			973,68

Autores: Eduardo Morillo y Juan Portero.

De acuerdo a la Tabla 4.9 se realizó una inversión total de \$ 973,68 USD, que en comparación a legos existentes como el de Bioloid el cual tiene un costo de \$ 1260,00 USD, se obtiene un ahorro de 286,32 USD lo que permite la posibilidad de desarrollar prototipos de fabricación nacional con un ahorro considerable.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.

- Se diseñó y construyó un prototipo de robot bípedo autónomo con desplazamiento dinámico con las siguientes características: velocidad de desplazamiento $0.78 \left[\frac{cm}{s} \right]$, peso 1.4 [Kg] y altura de 46,5 [cm].
- A velocidades menores a $1.4 \left[\frac{cm}{s} \right]$ el robot no pierde el equilibrio, por lo cual se define como velocidad de trabajo del robot $0.78 \left[\frac{cm}{s} \right]$.
- Mediante los cálculos realizados en conjunto con el software Matlab, se diseñó y simuló el modelo matemático del robot bípedo.
- Como concepción del diseño y la implementación realizada del robot bípedo, se tiene que el movimiento del caminado que el robot ejecuta se asemeja al del ser humano.
- El robot funciona correctamente de forma autónoma, permitiendo garantizar de esta forma que el algoritmo de control diseñado por medio del software libre de Arduino cumple los requerimientos exigidos para el desarrollo del proyecto.
- El uso del material ABS permite el ahorro de tiempo en la construcción de la estructura del robot bípedo, porque realiza el prototipado rápido de las piezas, ya sea que estas tengan formas simples o complejas.
- Se cumplió con los valores del tamaño y el peso del robot bípedo, los cuales están dentro de los rangos que se dieron en los parámetros de diseño.

- La selección de elementos fue la correcta, ya que con estos componentes electrónicos se obtuvieron los resultados esperados de funcionamiento del robot bípedo.
- La concepción del robot bípedo se dio con nuevas tecnologías dándole así un realce para dejar atrás tecnologías obsoletas, de esta manera continuar acorde con el avance de la tecnología a nivel mundial.
- Factores como la fricción con el piso, construcción de los componentes y resolución de la impresora 3D afectan a la hora de realizar el caminado haciendo que el robot tienda a desviarse.
- Los motores utilizados tiene un excelente desempeño para realización de proyectos del tipo robótico.
- La detección de obstáculos es limitada debido a que se cuenta con un solo sensor y además por su ubicación; razón por la cual se define una distancia adecuada para realizar el giro.
- La tarjeta electrónica usada cumplen de forma correcta con todo lo establecido para las acciones de control y de almacenamiento de datos utilizados para el proyecto.
- El diseño del pie mediante la teoría del ZMP, fue el correcto pues como se observó el robot mantiene el equilibrio mientras el punto de momento cero se encuentren dentro del polígono formando por los pies.

5.2. RECOMENDACIONES.

- Al momento de realizar el ensamblaje del prototipo de robot bípedo se debe verificar que las ranuras con las que constan los servomotores estén alineadas, de esta manera los servomotores estarán en la posición inicial para evitar errores al momento que se ejecuten las trayectorias.
- Verificar todas las conexiones de los distintos componentes electrónicos antes de encender el robot bípedo para evitar que se produzcan cortocircuitos, debido a que mientras el robot camina el movimiento tiende a mover los cables de su posición correcta.
- Debido al trabajo realizado en las distintas fases de prueba se propone que la tecnología de los servomotores Dynamixel AX-12A facilitan el desarrollo de proyectos robóticos, por lo que se sugiere utilizarlos para el desarrollo de prototipos de este tipo.
- Usar la tarjeta Arbotix-M para el control de los servomotores Dynamixel AX-12A, pues algunos de los puertos de esta tarjeta electrónica son dedicados exclusivamente para estos servomotores, ahorrando el trabajo de diseñar circuitos electrónicos que permitan la comunicación con estos servomotores.
- Trabajar con software libre porque existen programas como Python e IDE de Arduino, estos programas brindan un potencial increíble para el desarrollo de aplicaciones robóticas; además que exceptúan el pago de licencias.
- Descartar el uso de madera como material para la construcción de un robot bípedo, debido a que presenta problemas del tipo estructural al momento de poner en funcionamiento el robot, haciendo inestable y susceptible a fracturas pues las piezas no pueden conformarse íntegramente sino que necesitan ser pegadas entre sí.

- La implementación de las trayectorias para el robot bípedo como las que realiza el ser humano necesitan de mayores implicaciones investigativas debido a que el ser humano cuenta con músculos y mayor número de grados de libertad, los mismo que son limitantes en un prototipo, se recomienda continuar con la investigación en este tema.
- Realizar la captura de movimientos mediante cámaras de la trayectoria que realiza el ser humano en el plano sagital al momento de caminar, esto permitirá contar con una base de datos para la futura implementación de las trayectorias en otros prototipos no solamente del tipo robótico sino también servirían para prótesis inteligentes, con lo que se facilitará el desarrollo de prototipos más avanzados y complejos.
- El uso de un número mayor de sensores para la detección de los obstáculos para que el robot bípedo tenga un campo de visión más amplio y un mejor desenvolvimiento en diferentes entornos.
- Para mejorar los prototipos de robots bípedos se debe realizar investigaciones previas como la construcción de un transporte personal con auto-balance (segway), este proyecto ayudaría a conocer de mejor manera como funciona un giroscopio y como implementarlo para después aplicarlo en un robot bípedo.
- En base a la investigación realizada en este proyecto se recomienda continuar con el desarrollo de prototipos de robots ya sean bípedos, humanoides, etc.
- Para el desarrollo de robots bípedos o humanoides es necesario revisar textos de aprendizaje de temas sobre robótica móvil.
- Realizar prácticas más complejas sobre la dinámica y cinemática de robots en el entorno de Matlab para facilitar y mejorar el aprendizaje.

REFERENCIAS BIBLIOGRÁFICAS.

- Banzi, M. (2008). USA: O'REILLY.
- Barrientos, A., Penin, L., Balaguer, C., Aracil, R. (2007). *Fundamentos de Robótica*. España: Mc Graw Hill.
- Bonell, M. (2011). Diseño y Construcción de un Robot Humanoide. Barcelona: Universidad Politécnica de Cataluña. Departamento de Ingeniería Electrónica.
- Cabás, L. (2009). *Mecatrónica bioinspirada de robots humanoides de tamaño natural*. Madrid: Universidad Carlos III de Madrid. Departamento de Sistemas y Automática.
- Chancusi, S., Navarrete, P. (2014). *Diseño y construcción de una impresora 3D auto-replicable controlada inalámbricamente para el prototipado de piezas plásticas de bajo costo, mediante software libre*. Latacunga: Universidad de la Fuerzas Armadas ESPE. Departamento de Mecánica. Ingeniería Mecatrónica.
- Chavolla, E. (2010). Algoritmos genéticos modificados para el balance de robots . Guadalajara: Universidad de Guadalajara. Departamento de Electrónica.
- Corke, P. (2011). *Robotics, Vision and Control fundamental algorithms in Matlab*. Queensland: Springer.
- Esteve, J. (2001). *Apuntes de Robótica*. Valencia: Propia autoría.
- García, D. (2012). *Interfaz de control cinemático para el humanoide HOAP-3 en Matlab para el cálculo off-line de trayectorias*. Madrid: Universidad Carlos III de Madrid. Departamento de Sistemas y Automática.

- Guzmán, C. (2010). *Construcción de un robot bípedo basado en caminado dinámico*. Cuernavaca: CENIDET. Departamento de Ingeniería Mecatrónica.
- Hernández, F. (2008). *Diseño y construcción de prototipo neumático de prótesis de pierna humana*. Puebla: Universidad de las Américas Puebla. Departamento de Computación, Electrónica y Mecatrónica.
- Lezama, D., Sklar, A., Tejera, G. (2005). aHuman-Droid Prototype: Primeros pasos en robótica bípeda. *Workshop del CAFR2005*.
- López, R. (2012). *Planificación y optimización de la caminata de un robot bípedo*. México: Universidad Nacional Autónoma de México.
- Malagón, C. (2012). Búsqueda Heurística.
- Mott, R. (2006). *Diseño de Elementos de Máquinas*. México: Pearson Educación S.A.
- Núñez, F. (2007). *Control de movimiento empleando Labview, un enfoque didáctico*. Puebla: Universidad de las Américas Puebla. Departamento de Computación, Electrónica y Mecatrónica.
- Ogata, K. (2003). *Ingeniería de Control Moderna*. Madrid: Pearson Education.
- Ollero, A. (2001). *Robótica Manipuladores y Robots Móviles*. Barcelona: Marcombo.
- Ortega, E., Contreras, J. (2008). Diseño y Contrucción de un Robot Bípedo. *Revista Internacional de Educación en Ingeniería*, 10.
- Ramírez Ramos, O. (2008). Simulación en simmechanics de un sistema de control difuso para el robot udlap. Puebla; Universidad de las Américas Puebla. Departamento de Computación, Electrónica y Mecatrónica.

- Ramírez, J. G. (2009). Diseño y control de Kokone, un pequeño robot humanoide. Tamaulipas: CINVESTAV. Laboratorio de Tecnologías de Información.
- Rodríguez, F., López, M. (1996). *Control adaptivo y robusto*. Sevilla: Secretariado de publicaciones de la universidad de Sevilla.
- Sáenz, F. (2012). *Generación de trayectorias para el tren inferior del robot humanoide TEO subiendo escaleras*. Madrid: Universidad Carlos III de Madrid. Departamento de Sistemas y Automática.
- Santana, J. (2013). *Control del equilibrio en el proceso de caminado de un robot bípedo*. Guadalajara: Universidad de Guadalajara. Departamento de Electrónica.
- Singaña, M. (2012). Robótica Industrial. Ecuador.
- Suárez, M. (2012). *Interaprendizaje de Probabilidades y Estadística Inferencial con Excel, Winstats y Graph*. Ibarra: Imprenta M & V.
- Subir Kumar, S. (2010). *Introducción a la Robótica*. México: Mc Graw Hill.
- Torrente, O. (2013). *ARDUINO Curso Práctico de formación*. México D.F.: Alfaomega Grupo Editor.
- Torres, J. (2009). Diseño y Construcción de un Robot Bípedo Caminante. Monterrico: Universidad Perunana de Ciencias Aplicadas.
- Vele, O. L. (2008). Cinemática de un Robto Bípedo. México.

NETGRAFÍA

- Albero, M., Blanes, F., Benet, G. (2013). *YABIRO: PROTOTIPO DE ROBOT BÍPEDO AUTÓNOMO*. Recuperado el 18 de Marzo de 2014, de <http://www.ceautomatica.uji.es/old/actividades/jornadas/XXIV/documentos/ro/48.pdf>

Borger, A. (2004). *Herramientas Matemáticas para la Localización Espacial*. Recuperado el 10 de Febrero de 2014, de http://www.industriaynegocios.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20de%20Aut/trabajos/2004/Rob%C3%B3tica/seminario%202004%20robotica/Seminario_Robotica/Documentos/Herramientas%20Matematicas.htm

EcuRed. (2014). *EcuRed*. Recuperado el 06 de Mayo de 2014, de EcuRed: http://www.ecured.cu/index.php/Control_integral

González, G. (2009). *mailxmail.com*. Recuperado el 19 de Mayo de 2014, de mailxmail.com: <http://www.mailxmail.com/curso-calculo-componentes-transmisiones-mecanicas/coeficiente-seguridad-admisible-1>

Honda. (2014). *Honda The Power of Dreams*. Recuperado el 22 de Enero 22 de 2014, de Honda The Power of Dreams: <http://world.honda.com/ASIMO/>

Hosting, G. P. (2012). *PyPose*. Recuperado el 14 de Abril de 2014, de <http://code.google.com/p/arbotix/wiki/home>

Huaca, J. A. (2009). *Redes inalámbricas*. Recuperado el 22 de Enero 22 de 2014, de redes inalámbricas: <http://technical.blogspot.com/>

Jiménez, J. (2013). *ladelec.com*. Recuperado el 13 de Enero de 2014, de ladelec.com: <http://www.ladelec.com/teoria/informacion-tecnica/290-las-3-leyes-de-la-robotica>

Lenntech. (2014). *Aluminio - AI*. Recuperado el 22 de Enero 22 de 2014, de Aluminio - AI: <http://www.lenntech.es/periodica/elementos/al.htm>

Letraherido. (2009). *Letraherido*. Recuperado el 22 de Enero 22 de 2014, de Letraherido: <http://www.letraherido.com/19040302robots.htm>

Makealo. (2014). *Makealo 3D Printers Company*. Recuperado el 22 de Enero de 2014, de Makealo 3D Printers Company: <http://makealo.com/software/>

Malkav. (2012). *Sistemas Expertos*. Recuperado el 5 de Mayo de 2014, de Sistemas Expertos: <http://sistexpertosmalkav.blogspot.com/2012/04/232-heuristicas-y-funcion-de-control.html>

MASISA. (2014). *MDF MASISA*. Recuperado el 14 de Abril de 2014, de <http://www.masisa.com/chi/productos/tableros/mdf/>

Mathworks. (2014). *Matlab the lenguaje of technical computing*. Recuperado el 22 de Enero 22 de 2014, de Matlab the lenguaje of technical computing: <http://mathworks.com/products/matlab/>

Mathworks. (2014). *SimMechanics Model and simulate multibody mechanical systems*. Recuperado el 22 de Enero 22 de 2014, de SimMechanics Model and simulate multibody mechanical systems: <http://mathworks.com/products/simmechanics/>

Ramirez, Y. (2010). *Monografias.com*. Recuperado el 20 de Enero 20 de 2014, de Monografias.com: <http://www.monografias.com/trabajos60/servo-motores/servo-motores.shtml>

Reyes, C. (2013). *Técnicas Heurísticas*. Recuperado el 5 de Mayo 5 de 2014, de Técnicas Heurísticas: <http://inteligenciaartificial801b.blogspot.com/2013/02/tecnicas-heuristicas.html>

Robot Shop. (2009). *KRS-788HV*. Recuperado el 20 de Marzo de 2014, de <http://www.roboteshop.com/robot-catalogue/kondo/servos-motors/krs-788hv.html>

SolidWorks. (2011). *SolidWorks Web Help*. Recuperado el 15 de Abril 15 de 2014, de SolidWorks Web Help: http://help.solidworks.com/2011/spanish/SolidWorks/cworks/LegacyHelp/Simulation/Checking_stresses/prot_vonm.html?id=dab4434bf64d487d8cbbf45682c1f4ef#Pg0

Sony. (2014). *Sony*. Recuperado el 22 Enero de 2014, de Sony: http://www.sony.net/SonyInfo/News/Press_Archive/200312/03-060E/

SuperRobotica. (2002). *Servo Motor Hitec HS422 S330165*. Recuperado el 20 de Marzo de 2014, de <http://www.superrobotica.com/S330165.htm>

TecnologíaPlasticos. (2014). *Tecnología de Plasticos*. Recuperado el 22 de Enero de 2014, de Tecnología de Plasticos: <http://tecnologiadelosplasticos.blogspot.com/2011/06/abs.html>

Trossen Robotics. (2014). *Dynamixel AX-12A Robot Actuator*. Recuperado el 20 de Marzo de 2014, de <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

Trossen Robotics. (2014). *Dynamixel MX-28T Robot Actuator*. Recuperado el 20 de Marzo de 2014, de <http://www.trossenrobotics.com/dynamixel-mx-28-robot-actuator.aspx>

Wikipedia. (2013). *Wikipedia*. Recuperado el 22 de Enero de 2014, de Wikipedia: [http://es.wikipedia.org/wiki/Nao_\(robot\)](http://es.wikipedia.org/wiki/Nao_(robot))

Wikipedia. (2013). *Wikipedia*. Recuperado el 20 de Enero de 2014, de Wikipedia: http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1mbrica

Wikipedia. (2014). *Wikipedia*. Recuperado el 10 de Julio de 2014, de Wikipedia: [http://es.wikipedia.org/wiki/Proteus_\(electrónica\)](http://es.wikipedia.org/wiki/Proteus_(electrónica))

ANEXOS.

- **Anexo A:** Planos mecánicos.
- **Anexo B:** Algoritmo de control.
- **Anexo C:** Distribución chi-cuadrado.
- **Anexo D:** Diagramas esquemáticos de los componentes electrónicos.

Latacunga, Agosto de 2014.

Eduardo Morillo
AUTOR

Juan Portero
AUTOR

Ing. Marco Singaña
DIRECTOR

Ing. Jacqueline Llanos
CODIRECTORA

Ing. Vicente Hallo
DIRECTOR DE CARRERA

Dr. Freddy Jaramillo
SECRETARIO ACADÉMICO