

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y  
AUTOMÁTICA

*TRABAJO FIN DE GRADO*

FABRICACIÓN DEL ROBOT MINI-  
HUMANOIDE RAIDER, Y DISEÑO  
DE UNA ESTRATEGIA DE  
CAMINATA ESTABLE

---

*Autor: Daniel de Gracia Calvo*

*Tutor: Jorge Muñoz Yañez-Barnuevo*

*Director: Alberto Jardón Huete*

*Leganés, Febrero 2017*



Universidad  
Carlos III de Madrid



## AGRADECIMIENTOS

*Se podría decir que durante el desarrollo de este proyecto he aprendido más cosas sobre la materia que trata mi carrera, que durante los cuatro años de duración que tiene.*

*La multitud de contratiempos, imprevistos y demás problemas que han ido surgiendo durante el trascurso de este trabajo, han supuesto un verdadero reto para mí, tanto a nivel cognitivo como a nivel moral, puesto que había momentos en los que parecía que se llegaba a un “callejón sin salida” y todo lo que se había hecho previamente no era de utilidad.*

*Para el correcto desarrollo final de este proyecto, ha jugado un papel fundamental mi tutor, Jorge Muñoz Yañez-Barnuevo, ya que ha estado disponible en todo momento para cualquier duda que me surgiera, y me ha ayudado con sus conocimientos en la materia en la resolución del trabajo. Solo él y yo sabemos por la multitud de contratiempos y complicaciones por las que he pasado durante este trabajo, y en consecuencia, le estaré eternamente agradecido.*

*Agradecer también a mi familia todo el apoyo que me han proporcionado a todos los niveles, con mención especial para mis padres y mi hermano. ¡Gracias de corazón! por el apoyo y consejos proporcionados a lo largo de mi vida, sin los cuales estoy seguro de que esto no hubiera sido lo mismo.*

*Ha sido un placer y un verdadero reto afrontar este proyecto, enfrentarme a cada uno de los cientos de problemas que iban surgiendo sobre la marcha, y ante los cuales había que buscar una solución, pero bueno de eso trata la ingeniería... de ser ingenioso.*



## RESUMEN

En el presente documento se detalla el diseño y construcción del robot mini-humanoide RAIDER. La fabricación del robot parte de cero, desde la impresión de las piezas que forman el robot en impresora 3D, hasta su posterior montaje. Además del montaje mecánico del robot, también se realiza el diseño y fabricación de toda la electrónica del mini-humanoide, así como la adecuación e implantación de los sensores necesarios.

Adicionalmente, en este proyecto se diseña una estrategia de caminata estable para el robot, para que de esta forma, el robot sea capaz de caminar manteniendo el equilibrio. Para lograr dicho cometido, desempeñan un papel principal, la placa computadora *BeagleBone Black*, la controladora *OpenCM 9.04* encargada del movimiento de los servos del robot, así como la IMU, que dota al robot de la capacidad de sensar su inclinación.

Como aporte extra a este proyecto, se ha creado un manual de fabricación de RAIDER, ya que no existía ninguno hasta el momento, para que de esta manera en un futuro su fabricación sea rápida y sencilla.

Finalmente, una vez que el robot está fabricado y diseñada su caminata, se realizan una serie de pruebas para medir los resultados del proyecto y establecer las conclusiones del mismo.



## ÍNDICE

AGRADECIMIENTOS.....	3
RESUMEN .....	5
1 INTRODUCCIÓN .....	17
1.1 HISTORIA DE LA ROBÓTICA .....	17
1.1.1 ROBÓTICA Y FICCIÓN.....	18
1.1.2 LOS ROBOTS EN LA ACTUALIDAD .....	18
1.2 TIPOS DE ROBOTS.....	19
1.2.1 SEGÚN SU APLICACIÓN .....	19
1.2.2 SEGÚN EL TIPO DE CONTROL .....	22
1.2.3 SEGÚN SU GENERACIÓN .....	23
1.3 LA ROBÓTICA EN LA UC3M.....	24
1.4 SISTEMAS DE LOCOMOCIÓN EN ROBOTS .....	25
1.5 ROBOTS MINI-HUMANOIDES .....	30
2 OBJETIVOS .....	31
2.1 CONSTRUCCIÓN DE UN ROBOT MINI-HUMANOIDE .....	31
2.1.1 IMPRESIÓN DE LAS PIEZAS .....	31
2.1.2 MONTAJE MECÁNICO.....	32
2.1.3 MONTAJE ELECTRÓNICO .....	32
2.2 PUESTA EN MARCHA .....	32
2.3 DESARROLLO DE UNA APLICACIÓN PARA LA PLATAFORMA ROBÓTICA.....	32
3 ESTADO DEL ARTE .....	33
3.1 ESTADO DEL ARTE EN ROBÓTICA MINI-HUMANOIDE.....	33
3.1.1 ROBONOVA .....	33
3.1.2 KHR-3HV .....	34
3.1.3 ROBOVIE-X.....	35
3.1.4 BIOLOID .....	36
3.2 ESTADO DEL ARTE EN CAMINATA HUMANOIDE.....	38
3.2.1 MÉTODO BASADO EN LA SUPERPOSICIÓN DE MOVIMIENTOS.....	38
3.2.2 ESTRATEGIAS PARA MANTENER/RECUPERAR EL EQUILIBRIO EN HUMANOS .....	41
3.2.3 CAMINATAS ESTÁTICAMENTE ESTABLES Y DINÁMICAMENTE ESTABLES .....	43
3.2.4 ZERO MOMENT POINT (ZMP).....	45
4 MARCO REGULADOR.....	47
4.1 MARCO DE TRABAJO .....	47

4.1.1 EVOLUCIÓN DE LA ROBÓTICA HUMANOIDE EN LA UC3M.....	47
4.1.2 CAMPEONATO CEABOT.....	48
5 DESCRIPCIÓN DEL SISTEMA.....	51
5.1 ESTRUCTURA FÍSICA.....	51
5.2 ELECTRÓNICA.....	51
5.3 ESPECIFICACIONES TÉCNICAS DEL ROBOT.....	52
6 MONTAJE FÍSICO Y ELECTRÓNICO DEL ROBOT.....	54
6.1 IMPRESIÓN EN 3D DE LAS PIEZAS DEL ROBOT.....	54
6.2 MONTAJE FÍSICO.....	58
6.3 MONTAJE ELECTRÓNICO.....	59
6.4 PREPARACIÓN DEL SOFTWARE DE LA BEAGLEBONE BLACK.....	60
7 SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO Y SELECCIÓN DE LA TÉCNICA DE LOCOMOCIÓN.....	62
7.1 HERRAMIENTAS DE DESARROLLO.....	62
7.1.1 QT CREATOR.....	62
7.1.2 ROBOTIS OPENCM.....	63
7.2 TÉCNICA DE LOCOMOCIÓN.....	64
7.2.1 DESPLAZAMIENTO DE LA PROYECCIÓN DEL CDG SOBRE EL POLÍGONO DE SOPORTE DE FORMA MECÁNICA.....	68
7.2.2 CONTROL DE LA PROYECCIÓN DEL CENTRO DE GRAVEDAD MEDIANTE SENSORIZACIÓN.....	70
8 DESARROLLO DEL CÓDIGO.....	75
8.1 MÉTODOS DE PROGRAMACIÓN DE LA LOCOMOCIÓN EN ROBOTS.....	75
8.1.1 MÉTODOS ANALÍTICOS BASADOS EN EL <i>FEEDBACK</i> DE LOS SENSORES.....	75
8.1.2 PROGRAMACIÓN POR GUIADO.....	75
8.2 CÓDIGO DE LA OPENCM 9.04.....	76
8.2.1 FUNCIONES <i>MOV_VERTICAL</i> , <i>MOV_LATERAL</i> Y <i>MOV_FRONTAL</i> .....	77
8.2.2 FUNCIONES <i>STEP_LEFT</i> Y <i>STEP_RIGHT</i> .....	80
8.2.3 FUNCIONES <i>BEGIN_STEP_LEFT</i> Y <i>BEGIN_STEP_RIGHT</i> .....	81
8.2.4 FUNCIONES <i>PESO_ADELANTE</i> Y <i>PESO_ATRÁS</i> .....	82
8.2.5 FUNCIONES <i>REPOSO_LEFT</i> Y <i>REPOSO_RIGHT</i> .....	84
8.2.6 CÓDIGO PRINCIPAL O <i>MAIN</i> .....	84
8.3 CÓDIGO DE LA <i>BEAGLEBONE BLACK</i> .....	86
8.3.1 CONEXIÓN <i>SSH</i> PC-BEAGLEBONE BLACK.....	86
8.3.2 PROGRAMACIÓN DEL CÓDIGO.....	87
9 CONTRATIEMPOS.....	90
9.1 PIEZAS REALIZADAS CON IMPRESORA 3D.....	90



9.2 MONTAJE DEL ROBOT .....	91
9.3 MODIFICACIONES EN EL DISEÑO ORIGINAL DE RAIDER.....	91
9.4 PROBLEMAS CON LA <i>BEAGLEBONE BLACK</i> .....	92
9.5 BATERÍA LIPO.....	93
9.6 ELECCIÓN DE LA ESTRATEGIA DE CAMINATA.....	95
9.7 RETRASOS TEMPORALES POR DIFERENTES CAUSAS.....	95
10 PUEBAS Y RESULTADOS.....	97
10.1 CAMINATA SIN REALIZAR MOVIMIENTOS PREVIOS AL PASO .....	97
10.2 CAMINTA REALIZANDO MOVIMIENTOS PREVIOS AL PASO .....	98
11 CONCLUSIONES .....	100
12 ANEXOS .....	101
12.1 MANUAL DE MONTAJE DEL ROBOT RAIDER .....	101
12.1.1 MANUAL DE MONTAJE FÍSICO DEL ROBOT .....	101
12.1.2 MANUAL DEL MONTAJE ELCETRÓNICO DEL ROBOT .....	114
12.2 ID'S DE LOS SERVOS DEL ROBOT .....	119
12.3 ENTORNO SOCIO-ECONÓMICO Y PRESUPUESTO.....	120
12.3.1 ENTORNO SOCIO-ECONÓMICO .....	120
12.3.2 PRESUPUESTO .....	120
REFERENCIAS.....	122



# ÍNDICE DE FIGURAS

Figura 1.1: Robot Industrial.....	19
Figura 1.2: Robot Doméstico.....	20
Figura 1.3: Tipos de robots de servicio .....	20
Figura 1.4: Robot Médico.....	21
Figura 1.5: Robot Espacial .....	21
Figura 1.6: Robot de Aprendizaje.....	22
Figura 1.7: Robot 1ª Generación.....	23
Figura 1.8: Robot 2ª Generación.....	23
Figura 1.9: Robot 3ª Generación.....	24
Figura 1.10: Robot con ruedas .....	25
Figura 1.11: Robot oruga.....	26
Figura 1.12: Robot con patas .....	27
Figura 1.13: Polígono de soporte .....	28
Figura 1.14: Hormiga.....	28
Figura 1.15: Robot hexápodo.....	29
Figura 1.16: STriDER robot .....	30
Figura 3.1: Robonova .....	34
Figura 3.2: KHR-3HV .....	35
Figura 3.3: Robovie-X. ....	36
Figura 3.4: Bioloid Premium.....	37
Figura 3.5: Función Seno .....	39
Figura 3.6: Función Delta de Dirac .....	40
Figura 3.7: Función Pulso .....	40
Figura 3.8: Función Triangular.....	40
Figura 3.9: Ankle Strategy .....	41
Figura 3.10: Hip Strategy.....	42
Figura 3.11: Stepping Strategy .....	42
Figura 3.12: Suspensory Strategy.....	43
Figura 3.13: Esquema de las articulaciones de un bípedo .....	44
Figura 3.14: Fases en la caminata de un bípedo .....	44
Figura 4.1: CEABOT, prueba 1 .....	48
Figura 4.2: CEABOT, prueba 2 .....	49
Figura 4.3: CEABOT, prueba 3 .....	49
Figura 4.4: CEABOT, prueba 4 .....	49
Figura 5.1: Conexión de placas y sensores.....	52
Figura 6.1: Impresora "Blacky".....	54
Figura 6.2: Impresora "Hija-Resurrection" .....	55
Figura 6.3: Pieza originaria del robot Bioloid proporcionada por el departamento de robótica .....	55
Figura 6.4: Pieza originaria del robot Bioloid proporcionada por el departamento de robótica .....	56
Figura 6.5: Unión de dos motores.....	56
Figura 6.6: Tobillo del robot.....	57
Figura 6.7: Servo Dynamixel AX-12A.....	57
Figura 6.8: Mini-Humanoide modelo .....	58
Figura 6.9: Robot de Javier Isabel Hernández.....	59

Figura 6.10: Características técnicas de la BBB rev C .....	60
Figura 7.1: Qt Creator .....	63
Figura 7.2: Robotis OpenCM .....	64
Figura 7.3: Servo adicional en el extremo del brazo .....	65
Figura 7.4: Robot de Javier Isabel .....	66
Figura 7.5: Robot de este proyecto .....	66
Figura 7.6: Tool Center Point de un robot .....	67
Figura 7.7: Robot SIN realizar movimientos previos al paso .....	69
Figura 7.8: Robot realizando los movimientos previos al paso .....	69
Figura 7.9: Polígono de soporte sobre dos piernas .....	71
Figura 7.10: Ejes de coordenadas del robot .....	72
Figura 7.11: Robot en el pto A .....	73
Figura 7.12: Robot en el pto B desequilibrado .....	73
Figura 7.13: Robot en el pto B corregido .....	74
Figura 8.1: Trayectoria del pie en vuelo formada por cuatro puntos .....	76
Figura 8.2: TCP del robot .....	76
Figura 8.3: Posición de reposo .....	78
Figura 8.4: Función MovVertical .....	78
Figura 8.5: Función MovVertical (Perfil) .....	79
Figura 8.6: Función MovLateral .....	79
Figura 8.7: Función MovFrontal .....	80
Figura 8.8: Función Begin_Step_Left .....	81
Figura 8.9: Función Begin_Step_Right .....	82
Figura 8.10: Robot en reposo .....	83
Figura 8.11: Función Peso_Adelante .....	83
Figura 8.12: Diagrama de flujo del código de la OpenCM 9.04 .....	85
Figura 8.13: Dirección IP .....	86
Figura 8.14: Comando ssh -X .....	87
Figura 8.15: Sensor MPU 9250 .....	88
Figura 9.1: Upper_Leg fracturada .....	90
Figura 9.2: Batería LIPO 3S 11.1V 2200mah .....	93
Figura 9.3: Cargador Prophet Sport plus 50 W AC/DC .....	94
Figura 9.4: Cargador Prophet Sport plus 50 W AC/DC .....	94
Figura 12.1: Lower Leg .....	101
Figura 12.2: Foot .....	101
Figura 12.3: Ankle .....	102
Figura 12.4: Pie completo .....	102
Figura 12.5: Knee .....	103
Figura 12.6: Rodilla .....	103
Figura 12.7: Motores unidos .....	104
Figura 12.8: Upper Leg .....	104
Figura 12.9: Pierna completa (vista lateral) .....	105
Figura 12.10: Pierna completa (vista frontal) .....	105
Figura 12.11: Leg Bracket .....	106
Figura 12.12: Hip Chasis .....	106
Figura 12.13: Hip Top .....	107
Figura 12.14: Cadera .....	107
Figura 12.15: Cadera .....	107

Figura 12.16: Tren inferior del robot .....	108
Figura 12.17: Tren inferior del robot .....	109
Figura 12.18: Body .....	110
Figura 12.19: Pieza alza .....	110
Figura 12.20: Unión de la pieza Body con el servo de la cintura SIN alza .....	111
Figura 12.21: Unión de la pieza Body con el servo de la cintura CON alza .....	111
Figura 12.22: Brazo Bioloid .....	112
Figura 12.23: Servos del hombro interno en la pieza body .....	112
Figura 12.24: Parte superior del robot .....	113
Figura 12.25: Robot completo .....	113
Figura 12.26: Placa .....	114
Figura 12.27: Placa rectificada .....	114
Figura 12.28: Diagrama de la placa rectificada .....	115
Figura 12.29: OpenCM9.04 .....	115
Figura 12.30: Diagrama de conexionado de alimentación .....	116
Figura 12.31: Conexionado del UBEC y batería. Cables de entrada y salida del UBEC en rojo y negro. Cables de alimentación a la batería en rojo y negro. Conectores XT60 en amarillo .....	117
Figura 12.32: Montaje final de las placas .....	117



## ÍNDICE DE TABLAS

Tabla I: Características Robonova .....	34
Tabla II: Características KHR-3HV .....	35
Tabla III: Características Robovie-X .....	36
Tabla IV: Características Bioloid .....	37
Tabla V: Especificaciones técnicas del robot.....	53
Tabla VI: Pruebas sin movimientos previos al paso .....	97
Tabla VII: Gráfica pruebas sin movimientos previos al paso.....	98
Tabla VIII: Pruebas con movimientos previos al paso .....	98
Tabla IX: Gráfica pruebas con movimientos previos al paso.....	99
Tabla X: ID's de los servos .....	119
Tabla XI: Presupuesto Materiales y Equipos .....	121
Tabla XII: Presupuesto del personal .....	121
Tabla XIII: Resumen del presupuesto total .....	121





# 1 INTRODUCCIÓN

## 1.1 HISTORIA DE LA ROBÓTICA

El ser humano siempre ha buscado la forma de evitar realizar tareas tediosas en las que se emplea mucho tiempo y esfuerzo. Por ello a lo largo de la historia se han ido inventando y desarrollando diversos métodos y artilugios con los cuales realizar dichas tareas poco deseables.

Por este motivo surge la robótica, para que los humanos tengan a su disposición “esclavos” moralmente aceptados que realicen las tareas repetitivas y peligrosas que las personas no quieren realizar.

El origen de la robótica no tiene una fecha exacta como sí ocurre con otros acontecimientos. Se sabe que ya en la Alejandría del siglo I a.C se hallaron descripciones de más de 100 máquinas y autómatas. El famoso Leonardo da Vinci diseñó un robot humanoide en 1495, al cual llamo *Caballero Mecánico*.

La robótica ha ido evolucionando a lo largo de la historia, desde hechos que poco tienen que ver con lo que hoy en día entendemos por robótica, hasta los últimos acontecimientos que se producen continuamente hoy en día. De esta forma, a lo largo de la historia, han ido aconteciendo una serie de hitos que han hecho que la robótica evolucione hasta el concepto que tenemos de ella hoy en día. A continuación, se muestran algunos de estos acontecimientos que muestran de forma orientativa la evolución que ha seguido la robótica a lo largo del tiempo.

-En 1921 surge el primer autómata de ficción llamado *robot*.

-En 1956 aparece el primer robot comercial gracias a la compañía *Unimation*.

-En 1961 se instala el primer robot industrial creado también por la misma compañía.

-En 1970 la empresa alemana *KUKA* instala la primera línea de soldadura con robots industriales.

-En 1971 se funda JIRA, *Japanese Industrial Robotic Association*.

-En 1976 se teleoperan desde la Tierra los brazos de las sondas *Viking-1* y *Viking-2*, mandadas para tomar muestras de Marte.

-En el año 2000 aparece el primer robot humanoide capaz de desplazarse de forma bípeda e interactuar con las personas.

Sobre robótica se ha escrito multitud de documentos y libros, algunos con contenidos aplicables en la vida cotidiana, y otros de ficción. Es por eso por lo que hay que diferenciar entre la parte de ficción, y la parte veraz y real de la robótica.

### 1.1.1 ROBÓTICA Y FICCIÓN

La palabra *robot* proviene del término de origen checo "*robota*" que podría traducirse como "servidumbre compulsiva". Este nuevo término fue utilizado por primera vez por el autor checo Karel Capek en 1920, en su obra "*Rossum's Universal Robots*", cuyos protagonistas eran unos esclavos de apariencia humana que luchaban por su libertad.

Por otro lado, la palabra *robótica* fue utilizada por primera vez en 1941 por Isaac Asimov, el gran escritor y bioquímico de origen ruso, conocido mundialmente por sus grandes obras de ciencia ficción y en especial por la formulación de las "tres leyes de la robótica" que todo robot ha de cumplir.

Destacar que se han realizado cientos de películas y libros que tratan sobre robots pero desde un punto de vista poco real o de ficción. Quizá el máximo exponente de estas obras sea la película "Yo, Robot", basada en un relato de Isaac Asimov y que se desarrolla en Chicago en un futuro no muy lejano en el tiempo. Robots y humanos conviven en paz, con los robots estando al servicio de los humanos, hasta que ocurre un hecho concreto que hace que la relación entre humanos y robots ya no sea tan idílica. Esto es un claro ejemplo de la robótica de ficción, ya que a día de hoy es imposible llegar al nivel tan avanzado que tienen los robots que se muestran en la película.

### 1.1.2 LOS ROBOTS EN LA ACTUALIDAD

Hoy en día, el nivel y conocimiento actual de la robótica dista mucho de lo que se relata en las obras de ficción.

La robótica se ha dividido en múltiples ramas de trabajo según la función que el robot tenga que realizar. Una de las principales ramas son los robots industriales, que resultan indispensables en los procesos de fabricación actuales.

Joseph Engelberger tuvo un papel fundamental en la robótica industrial, ya que fue quién desarrolló, junto con George Devol, el primer robot industrial estadounidense, el *Unimate* (IFR), en 1959. Este robot pesaba dos toneladas y empleaba actuadores hidráulicos que llegaron a tener una precisión de una diezmilésima parte de pulgada [1].

Otra línea de trabajo dentro de la robótica es la robótica mini-humanoide. Esta rama todavía no está tan explotada y extendida como otras, ya que a día de hoy no tiene grandes aplicaciones para la vida cotidiana, pero es una parte de la robótica que posee un gran potencial y sobre la que están abiertas muchas líneas de investigación y concursos, como el CEABOT.

Debido a este potencial y a que resulta un mundo realmente interesante, es por lo que este trabajo trata sobre el desarrollo y programación de un robot mini-humanoide.

## 1.2 TIPOS DE ROBOTS

Debido a que la robótica es una materia que lleva siendo estudiada e investigada durante muchos años, en la actualidad el conocimiento que hay sobre ella es bastante cuantioso.

Debido a esto, existen multitud de tipos de robots que se diferencian unos de otros por diferentes parámetros. A continuación se realizará una breve clasificación en base a los tipos de robots:

### 1.2.1 SEGÚN SU APLICACIÓN

- Robots industriales: es la clase de robot más común, y se emplean en las diferentes tipos de procesos de fabricación existentes. Estos robots se caracterizan por estar formados por múltiples articulaciones que dotan al robot de un gran número de grados de libertad, lo que posibilita y facilita la realización de la tarea que tenga que hacer el robot (pintar, soldar, montaje de piezas, mecanizado, control de calidad.... entre otras muchas aplicaciones posibles).



*Figura 1.1: Robot Industrial*

- Robots de servicio: Según la Federación Internacional de Robótica (IFR), estos tipos de robots son robots que operan de forma automática o semi-automática, realizando tareas que facilitan el bienestar de los seres humanos. De esta forma, estos robots realizan tareas, excluyendo las tareas de manufacturas, que resultan repetitivas, difíciles, peligrosas o aburridas para las personas. Entre

los robots de servicio destacan los robots domésticos, que se ocupan de tareas domésticas como la limpieza, cocinar o la vigilancia.



Figura 1.2: Robot Doméstico

A continuación se pueden observar los tipos de robots de servicio según la IFR.

510	Robots de servicios a humanos (Personales, seguridad, entretenimiento, etc.)
	Interactivos
	No interactivos
	Otros
520	Robots de servicios a equipos (Mantenimiento, reparación, limpieza, etc.)
	Interactivos
	No interactivos
	Otros
530	Otros robots desarrollando funciones autónomas (vigilancia, transporte, adquisición de datos, etc.) y-o robots de servicio que no entran en 510 y 520

Figura 1.3: Tipos de robots de servicio

- **Robots médicos:** estos robots principalmente se centran en aspectos quirúrgicos en los que se requiere gran precisión. De esta forma, realizan una tarea muy complicada, y ayudan al cirujano.



Figura 1.4: Robot Médico

- Robots espaciales: este tipo de robots se emplean en el espacio para realizar tareas que si tuvieran que ser realizadas por un humano supondrían un alto coste económico y conllevarían un alto riesgo para la integridad de la persona, como puede ser la toma de muestras de un asteroide o la exploración de un planeta.



Figura 1.5: Robot Espacial

- Robots sociales: la principal característica de estos robots es la interacción social con los seres humanos. Es común que estos robots adquieran sus movimientos mediante imitación. Un operador guía al robot realizando el movimiento que quiere que aprenda, el robot lo memoriza, y de esta manera ya puede realizar dicho movimiento siempre que lo requiera sin necesidad del operador.



Figura 1.6: Robot de Aprendizaje

### 1.2.2 SEGÚN EL TIPO DE CONTROL

Esta clasificación diferencia a los robots según su método de control (ISO 8373) [2]. A continuación se muestran los principales tipos de control en robots:

- Secuencial: los movimientos se realizan siguiendo un orden, de esta forma hasta que no termina un movimiento no se inicia el siguiente. Los controladores son de dos estados (arriba-abajo, abierto-cerrado), y entre dichos estados no se puede controlar la operación ni la trayectoria a realizar.
- Controlado por trayectoria: los movimientos de tres o más ejes se desarrollan siguiendo instrucciones que especifican en el tiempo la trayectoria que hay que seguir para llegar a la siguiente posición.
- Control adaptativo: el sistema de control de estos robots tiene en cuenta la información sensorial que le llega de sus sensores. Además, poseen una capacidad de aprendizaje de las tareas que realizan de forma repetitiva.
- Teleoperados: son robots operados de forma remota por un ser humano, extendiendo las capacidades motoras y sensoriales del humano a un lugar remoto.

### 1.2.3 SEGÚN SU GENERACIÓN

Este tipo de clasificación es la más típica a la hora de enumerar los diferentes tipos de robots (IFR) [3].

- 1ª Generación: repite la tarea programada de forma secuencial sin tener en cuenta las alteraciones posibles de su entorno.



Figura 1.7: Robot 1ª Generación

- 2ª Generación: estos robots empiezan a tener en cuenta al entorno. Toman información limitada de él y actúan según está información. Son capaces de localizar, clasificar y detectar esfuerzos, y adaptar sus movimientos en consecuencia.

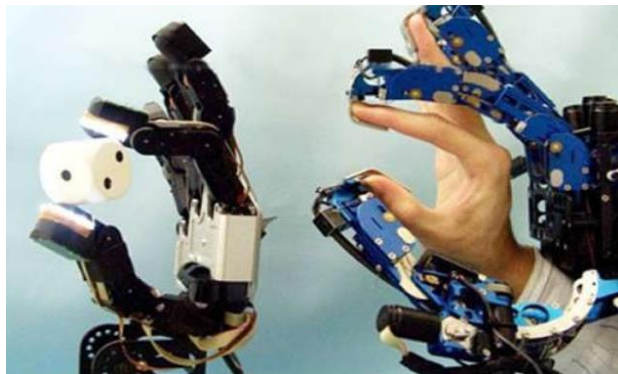


Figura 1.8: Robot 2ª Generación

- 3ª Generación: la programación de estos robots se realiza mediante el empleo de lenguaje natural. Pueden planificar tareas de forma automática.





Figura 1.9: Robot 3ª Generación

### 1.3 LA ROBÓTICA EN LA UC3M

La Universidad Carlos III (UC3M) es una de las universidades punteras a nivel europeo en robótica, y dispone de uno de los laboratorios más grandes de Europa en dicha especialidad.

Esta universidad tiene múltiples líneas de investigación dentro de la robótica, ya que es una materia muy amplia y abierta. La UC3M, en su página de *RoboticsLab* [4], muestra todas las líneas de investigación y proyectos que se están llevando a cabo como: Vehículos aéreos no tripulados (UAVs), Impresoras 3D Open Source, Robots mini-humanoides, *Artificial Muscles*, robots de asistencia de diseño portátil, control de manipuladores móviles, el desarrollo de una mano robótica que sea capaz de manipular objetos... y así hasta cerca de 35 líneas diferentes de investigación.

Esto muestra el nivel de interés que tiene esta universidad en la robótica y sus derivados.

Una de estas líneas de investigación, la de robots mini-humanoides, será el tema central sobre el que se centrará este trabajo de fin de grado, aunque también se tratarán otros temas como el de las Impresoras 3D, las cuales fueron necesarias para la realización del TFG.

En el capítulo 2 se detallarán los objetivos de este proyecto, el cual seguirá y estudiará varias de estas ramas de investigación, fundamentales para la realización de este trabajo. Entre estas ramas, además de las ya mencionadas anteriormente (robótica mini-humanoides, impresoras 3D), tiene especial importancia la rama de sistemas de locomoción en robots, puesto que para que un robot realice una caminata de forma



satisfactoria, su sistema de locomoción protagoniza un papel principal. Por ello, en el siguiente capítulo se explica qué es un sistema de locomoción y los diferentes tipos de locomociones que existen hoy en día.

#### 1.4 SISTEMAS DE LOCOMOCIÓN EN ROBOTS

El término locomoción hace referencia a la forma en la que algo o alguien (en el caso de este trabajo, un robot) se desplaza a lo largo del espacio.

Para conseguir este propósito existen varios sistemas de locomoción según los elementos que se utilicen para lograr dicho desplazamiento. A continuación se nombrarán los sistemas de locomoción más comunes en robótica:

- Sistema de locomoción basado en ruedas: esta clase de robots presenta grandes ventajas, como la de ser un sistema bastante sencillo de implementar y controlar. Además, las ruedas son un sistema barato, y su diseño es bastante simple en comparación con otros sistemas de locomoción.

El gran inconveniente que este sistema presenta, es el de los grados de libertad en su desplazamiento. Estos robots se pueden desplazar únicamente por terrenos en los que sea posible el desplazamiento mediante ruedas, quedando exentos terrenos con grandes irregularidades.



*Figura 1.10: Robot con ruedas*

- Sistema de locomoción de tipo oruga: pese a la sencillez de los sistemas de control de locomoción de estos robots, estos presentan grandes desventajas a la hora del desplazamiento, puesto que el movimiento queda bastante limitado a moverse únicamente por

superficies medianamente lisas y sin grandes irregularidades en el terreno (un escalón o un agujero, por ejemplo).

Además al existir deslizamiento se producen pobres resultados en odometría, lo que provoca que el control de posición del robot se complique. Como último inconveniente, mencionar que se requiere de mucha energía para realizar el giro de las ruedas.



*Figura 1.11: Robot oruga*

- Sistema de locomoción basado en el empleo de patas: este tipo de robots pueden moverse por todo tipo de terrenos, debido a la enorme libertad que les proporcionan las patas (ej: subir unas escaleras).

Por el contrario, estos robots presentan grandes dificultades, ya que al poseer muchos grados de libertad, el control del robot se vuelve bastante complicado. Además, este tipo de robots presentan una complejidad bastante alta a la hora de mantener su estabilidad, puesto que la superficie sobre la que están sostenidos, es menor que la de los robots con ruedas o la de los robot de tipo oruga. Esto, sumado al alto consumo de energía que supone el control de estabilidad de estos robots, hace que su complejidad sea bastante alta.



*Figura 1.12: Robot con patas*

Puesto que este trabajo trata sobre los robots mini-humanoides, nos centraremos en los sistemas de locomoción y estrategias de caminata típicos de robots bípedos.

La elección de la estrategia de caminata o locomoción de un robot es clave, ya que es uno de los aspectos básicos a la hora de dotar al robot de la autonomía necesaria para desplazarse de una manera fluida y sin perder el equilibrio, ya que tal y como se comentó líneas arriba, es bastante complicado que un robot bípedo mantenga la estabilidad.

Hay varias formas de estudiar la estabilidad del robot, a continuación se explicarán de forma breve los conceptos de estabilidad estática y dinámica, necesarios para comprender posteriormente las estrategias de caminata más empleadas en la actualidad que se detallan en el capítulo 3.2.

Estática/Dinámica: La estabilidad estática se logra a través del diseño mecánico del robot. Este tipo de estabilidad implica que el robot pueda estar parado sobre sus piernas y no perder el equilibrio y caerse. Los seres humanos no son estáticamente estables pero si son estables dinámicamente. Esto se debe a que los humanos tienen un control activo inconsciente, gracias al cual, controlan el balanceo de su cuerpo.

La estabilidad estática pierde complejidad conforme mayor número de apoyos tiene el robot. Esto se debe a que para que sea estable, la proyección del centro de gravedad (CDG) debe de estar dentro del polígono de soporte, que es el área formada por los puntos que están en contacto con el suelo.

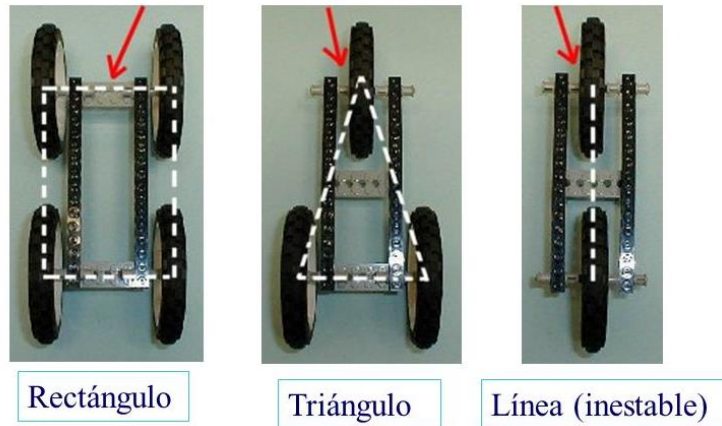


Figura 1.13: Polígono de soporte

En robots bípedos como el de este proyecto, conseguir que sea estable estáticamente es muy complicado ya que el polígono de soporte de estos robots es muy pequeño, mucho más pequeño que el propio robot en sí, por lo que conseguir este tipo de estabilidad se vuelve bastante complicado, salvo que el robot tenga unos pies enormes, algo que no sería viable.

Es por eso que para conseguir este tipo de estabilidad, cuantas más piernas tenga el robot más fácil será conseguir que sea estable.

### ¿Qué número de piernas es el más eficiente?

La respuesta a esta pregunta se encuentra en la naturaleza. Como se puede observar en muchos insectos, el número de patas del que disponen varía entre seis y ocho. Esto es debido a que a la hora de caminar se alternan las patas que están en el aire con las que están en contacto en el suelo, provocando de esta forma, que en animales/robots de seis patas, siempre estén en contacto con el suelo tres de las seis patas, y de esta forma la proyección de CDG esté dentro del polígono de soporte (que en este caso sería en todo momento un triángulo formado por las tres patas que están en contacto con el suelo), logrando así una continua estabilidad estática, esto se conoce como *tripod gait*. Los robots de seis patas o hexápodos son los más comunes en lo que a robots con piernas se refiere.



Figura 1.14: Hormiga



Figura 1.15: Robot hexápodo

### ¿Por qué no se utilizan tanto los robots de tres patas?

Se podría pensar que con un robot de tres patas se conseguiría la estabilidad estática, pero no es así. El motivo por el cual un robot de tres piernas es complicado que sea estable, es que a la hora de moverse, una de las piernas está en el aire mientras que las otras dos estarían en contacto con el suelo, provocando que de esta manera el polígono de soporte solo estaría formado por dos piernas, por lo que surgiría de nuevo el problema de estabilidad de los robots bípedos.

Conseguir estabilidad estática mientras el robot se mueve es muy ineficiente energéticamente hablando. Como alternativa a la estabilidad estática, surge la estabilidad dinámica que permite al robot mantenerse en equilibrio teniendo en cuenta la inercia propia.

Para conseguir este tipo de estabilidad, se requiere un control activo, por ejemplo utilizando el método del *inverse pendulum problem*, que supone un nivel de complejidad bastante mayor que el control empleado en un robot que se rige por el método de estabilidad estática.

A pesar de lo mencionado anteriormente, comentar que sí que existen ejemplos y prototipos de robots de tres patas que han tenido éxito, como el STriDER [5], robot desarrollado por el *Robotics & Mechanisms Laboratory* de *Virginia Tech*. Este robot se inspira en la forma de locomoción humana a pesar de estar formado por tres piernas. El STriDER al emplear únicamente tres piernas en lugar de seis u ocho, consigue reducir bastante el número de motores empleados y de esta forma la complejidad a la hora de sincronizar los diferentes motores de las articulaciones (*joints*) disminuye en gran medida, consiguiendo una mayor eficiencia energética en el movimiento.



Figura 1.16: STrider robot

Por otro lado, los robots móviles más comunes hoy en día son los que disponen de ruedas para moverse y no piernas, ya que las ruedas son más eficientes y más fáciles de controlar que las piernas, pero tienen un gran inconveniente, la limitación a la hora de moverse por terrenos irregulares en los que hay gran número de obstáculos y en los que el empleo de las piernas se hace indispensable.

## 1.5 ROBOTS MINI-HUMANOIDES

La robótica humanoide es una rama de la robótica que trata que los robots imiten tanto en apariencia como en movimiento (ya sea locomoción, movimientos gestuales o movimientos en estático) a los seres humanos.

Estos robots mini-humanoides pueden tener diferentes objetivos, como realizar tareas humanas, servir de herramienta de investigación para estudiar la locomoción bípeda, o servir de fines lúdicos (como, por ejemplo, juguetes para niños).

Como ya se ha mencionado anteriormente, este trabajo se centrará en la construcción de un robot mini-humanoides, de ahí que la robótica mini-humanoides suponga un punto esencial en este proyecto.

Más adelante, concretamente en el capítulo 3.1, se profundizará en el estado del arte de la robótica mini-humanoides. En el capítulo 3.2, se hará hincapié en el estado del arte en caminata humanoide, que es otro punto fundamental de este trabajo.

## 2 OBJETIVOS

A continuación se numeran los objetivos que este proyecto persigue. Para que su comprensión y realización sea más simple, se han dividido en varios subobjetivos.

2.1 Construcción de un robot mini-humanoide basado en otro diseño, pero con modificaciones necesarias para desarrollar los siguientes objetivos

2.2 Puesta en marcha del robot

2.3 Desarrollo de una aplicación de caminata

En los siguientes apartados, se procede a explicar de forma resumida en que consiste cada uno de estos objetivos. La explicación detallada de cada objetivo se mostrará más adelante.

### 2.1 CONSTRUCCIÓN DE UN ROBOT MINI-HUMANOIDE

El robot mini-humanoide en el que se basa el de este proyecto, es el realizado por Javier Isabel Hernández en su Trabajo de Fin de Grado “Desarrollo de una plataforma robótica mini-humanoide con visión artificial” [6]. Este mini-humanoide se trata de un robot *Bioloid*, al cual se le realizaron varias modificaciones para mejorar sus capacidades. A continuación se muestran cada uno de los subobjetivos en los que se divide la construcción del robot. En el capítulo 6, se contará de forma detallada todo este proceso.

#### 2.1.1 IMPRESIÓN DE LAS PIEZAS

Debido a que el robot mini-humanoide elegido es una nueva versión del robot empleado por Javier Isabel Hernández, es necesario realizar el diseño y fabricación de las nuevas piezas que se van a emplear para la construcción del robot.

En primer lugar, se procederá a la impresión de las piezas que se utilizarán para posteriormente montar el robot. Esta impresión se realizará en unas impresoras 3D situadas en los laboratorios de la Universidad Carlos III de Leganés, a partir de unos planos existentes de dichas piezas, realizados por Javier Isabel y disponibles en su repositorio de *GitHub* [7].

### 2.1.2 MONTAJE MECÁNICO

Una vez obtenidas todas las piezas necesarias, se procederá al montaje del robot. Aparte, tomando como modelo un robot mini-humanoide proporcionado por el departamento de robótica, de forma y estructura similar al futuro robot que se quiere construir, pero con algunas diferencias en su diseño, se montará el robot utilizando las piezas impresas en 3D, además de utilizar algunas piezas del robot Bioloid que se toma como modelo.

### 2.1.3 MONTAJE ELECTRÓNICO

Una vez construido el robot de forma mecánica, se pasará al montaje electrónico del mismo. Se instalarán los diferentes sensores que llevará el robot, como la IMU (giróscopo + acelerómetro + brújula). También se instalarán los componentes electrónicos que forman tanto el controlador principal como el controlador de locomoción, necesarios para el procesamiento de datos y control de los servos.

### 2.2 PUESTA EN MARCHA

Una vez realizada la construcción física y electrónica del robot se procederá a la programación y puesta en marcha del mismo, dotando así al robot de la capacidad de movimiento.

### 2.3 DESARROLLO DE UNA APLICACIÓN PARA LA PLATAFORMA ROBÓTICA

Por último se desarrollará una aplicación para que el robot sea capaz de desplazarse manteniendo en todo momento el equilibrio.

El hecho de que sea capaz de mantener el equilibrio, es un punto novedoso y fundamental en este proyecto, puesto que conseguir que un mini-humanoide no se caiga al desplazarse no resulta nada sencillo.

Para lograr esto, se empleará una IMU dotada de giróscopo, acelerómetro y brújula.



## 3 ESTADO DEL ARTE

A continuación se relatará el estado del arte en robótica mini-humanoide. Dentro de la robótica mini-humanoide se profundizará en: tipos de mini-humanoides y tipos de caminatas.

### 3.1 ESTADO DEL ARTE EN ROBÓTICA MINI-HUMANOIDE

Los robots mini-humanoides tienen una presencia cada vez mayor en el mercado, y sobre los cuales se están llevando a cabo gran cantidad de estudios y proyectos. A continuación se enumeran algunos de los robots mini-humanoides más comunes y mejor equipados que hay en el mercado.

#### 3.1.1 ROBONOVA

El **Robonova** (*figura 3.1*) es uno de los robots pioneros en lo que a robótica mini-humanoide se refiere. Fue uno de los robots más comercializados y utilizados a nivel global debido a que su equipamiento era verdaderamente bueno.

Desde que se fundó la Asociación de Robótica de la Universidad Carlos III, se han utilizado Robonovas en competiciones (CEABOT), y en diferentes trabajos/proyectos para su estudio y utilización en diferentes áreas.

El Robonova posee servos digitales HSR 8498HB que hacen la función de actuadores con un par de 7.4 kg·cm. Estos servos tienen la capacidad de leer posiciones y comunicárselas al controlador (*Motion Feedback*), que en este caso es un micro ATmega 128 sobre una placa base, cuyas principales características son: 40 pines GPIO, 3 salidas PWM, puerto serie, 8 entradas analógicas y capacidad para conectarse mediante I2C. Se pueden ampliar las capacidades de este robot instalándole más sensores y actuadores, debido a que dicho montaje es bastante sencillo. Por último decir que este robot tiene 16 grados de libertad, por lo que debido a esto, tiene una alta capacidad de movimiento. Su software está basado en lenguaje Basic.



Figura 3.1: Robonova

Par servos	7.4 kg·cm
Electrónica	ATMega 128 ( 40 pines GPIO, 3 salidas PWM, puerto serie, 8 entradas analógicas, I2C)
Número de servos	16

Tabla 1: Características Robonova

### 3.1.2 KHR-3HV

El **KHR-3HV**, mostrado en la *figura 3.2*, es actualmente uno de los mini-humanoides más avanzados debido a su gran dinamismo. Este robot es producido por el fabricante japonés Kondo.

Este robot cuenta con 17 servos KRS-2552HV que tienen una capacidad de 14 kg·cm de par máximo. El controlador del robot es una placa RCB-4 cuyas principales características son: 10 GPIOs y 10 entradas analógicas, que pueden controlar hasta 35 servos o actuadores.

A nivel de programación, el software empleado por el fabricante es el *Heart to Heart V4*.

Existen varias librerías abiertas para su utilización, debido a que el número de usuarios que utilizan este robot es bastante amplio, permitiendo de esta manera poder programar en lenguajes más comunes, como Python y C.



Figura 3.2: KHR-3HV

Par servos	14 kg·cm
Electrónica	RCB-4 ( 10 pines GPIO,10 entradas analógicas)
Número de servos	17 servos KRS-2552HV

Tabla II: Características KHR-3HV

### 3.1.3 ROBOVIE-X

El **Robovie-X**, tiene tres versiones que se diferencian en el número y tipo de servos que emplea. Estas tres versiones (Lite, Standard y Pro) son fabricadas por la empresa japonesa Vstone.

El modelo Standard, presentado en la *figura 3.3*, está compuesto por servos VS-S092J con un par máximo de 9.2 kg·cm. Estos actuadores proporcionan al robot la capacidad de tener hasta 17 grados de libertad. El Robovie-X emplea un controlador VS-C1 con 30 canales para la conexión de los servos.

Para programar el robot es necesario el programa RobovieMaker2, que el fabricante suministra junto con el robot. La programación del robot se basa en diagramas de flujo.



Figura 3.3: Robovie-X.

Par servos	9.2 kg-cm
Electrónica	VC-C1 ( 30 canales para la conexión de los servos )
Número de servos	17

Tabla III: Características Robovie-X

### 3.1.4 BIOLOID

El **robot Bioloid** (figura 3.4), se asimila mucho al empleado en este trabajo, ya que incluye 18 servos Dynamixel modelo AX-12A (como los empleados en la versión del robot de este proyecto) ó AX-18A, dependiendo de la versión. Una de las grandes ventajas que tiene el *Bioloid*, es que sus servos pueden realizar un control de su propia velocidad, par, etc, gracias al microcontrolador ATmega8. Esto provoca que sea posible procesar dentro del propio actuador información de bajo nivel, quitando de esta manera trabajo a la controladora del robot para que se centre en tareas de mayor nivel.

Dependiendo del kit, nos encontramos con diferentes tipos de controladoras de la marca Robotis (serie CM), concretamente, la CM-5 en el caso del Bioloid Comprehensive y la CM-510 o CM-530 (según qué versión) en el Bioloid Premium y GP.

#### Robotis CM-5:

Esta controladora viene con el *Bioloid Comprehensive* y dispone de un microcontrolador ATmega128. Esto hace que sea posible la conexión de sensores específicos de la marca *Robotis*, como el *Dynamixel AX-S1* [8].

Robotis CM-510:

Dispone de un microcontrolador ATmega2561, y nos la encontramos junto con la versión *Bioid Premium*. Al igual que la Robotis CM-5, esta controladora tiene la capacidad de conectar al bus TTL sensores de la misma marca, pero adicionalmente dispone de un puerto para conectar un receptor ZigBee (y teleoperar el robot), y cinco puertos para sensores analógicos.

Robotis CM-530:

Esta controladora es una evolución de la CM-510, y emplea un microcontrolador ARM Cortex STM32F103RE de 32 bits. La Robotis CM-530 viene junto con la versión de Bioid GT, y sus características son semejantes a las de su predecesora, CM-510 (cinco puertos de salida analógica para sensores, un puerto para Bluetooth, etc).

El lenguaje empleado (R+) para la programación de este robot, es bastante simple y muy visual debido a su uso para la educación (se emplea mucho este tipo de robot para robots educativos).

Esta sencillez también tiene desventajas, debido a que no permite emplear todas las posibilidades de uso de las que disponen los *Dynamixel*, por lo que es un lenguaje bastante limitado.



Figura 3.4: Bioloid Premium

Par servos	15 kg·cm
Electrónica	ATmega8
Número de servos	18 servos Dynamixel AX-12A

Tabla IV: Características Bioloid

A continuación se describen las estrategias de caminata más empleadas en la actualidad.

## 3.2 ESTADO DEL ARTE EN CAMINATA HUMANOIDE

En este capítulo se hablará de las estrategias de caminata que emplean los robots humanoides. Puesto que hay varias estrategias posibles a seguir, se nombrarán y explicarán únicamente las más comunes y relacionadas con el proyecto.

La elección de la estrategia de caminata a seguir es uno de los puntos claves y más complicados a la hora de realizar un robot mini-humanoide. De hecho no hay mejor ejemplo que los seres humanos para ver la complejidad que conlleva una caminata.

Hoy en día, la caminata humana sigue sin comprenderse completamente, por lo que esto da una muestra de lo complicado y crítico que resulta este punto.

### 3.2.1 MÉTODO BASADO EN LA SUPERPOSICIÓN DE MOVIMIENTOS

Este método [9] se basa en modelos trigonométricos. Para ello hay que dividir los diferentes conjuntos (pierna, brazo...) en subconjuntos formados por las articulaciones que los forman. Para obtener el movimiento final de cada articulación, hay que sumar el giro producido por cada movimiento en las distintas articulaciones.

Este método no sería necesario en robots con piernas o brazos que tuvieran dos o tres grados de libertad ya que en estos casos se podría emplear la cinemática directa para determinar la posición de cada articulación, pero para robots que tengan brazos o piernas de más de cuatro grados de libertad (como el de este proyecto), intentar emplear los métodos de cinemática directa e inversa sería muy poco útil, ya que los cálculos que habría que realizar se complicarían exponencialmente, lo que provoca que sea muy difícil de determinar la solución (si es que la hay), por no comentar la aparición de redundancias y singularidades al existir varios ejes alineados.

Para solucionar este problema, se divide el conjunto requerido, por ejemplo una pierna, en mecanismos más sencillos para ser analizados cada uno de forma individual. Cuando ya se han analizado cada uno de los subconjuntos, hay que unirlos para obtener de esta forma el movimiento final de la pierna. Para ello, hay que sumar la variación que en cada eje produce los diferentes estudios realizados al dividir la pierna en subconjuntos.

$$\Delta q_{\text{movimiento1}} + \Delta q_{\text{movimiento2}} + \dots + \Delta q_{\text{movimientoN}} = \Delta q_{\text{total}}$$

Para analizar cada uno de los subconjuntos, se emplean relaciones trigonométricas que relacionan cada uno de los ejes que influyen en el movimiento junto con un parámetro variable. Al estar basado en relaciones trigonométricas, posibilita que se pueda calcular analíticamente, lo que provoca que sea posible encontrar una solución rápida.

Otro de los factores fundamentales, es la señal con la que se gobiernan los ejes o articulaciones. Hay varios tipos de señales:

- **Señal sinusoidal:** esta trayectoria se basa en la función  $\sin(\omega t + \phi)$ . Esta función es continua al igual que su primera y segunda derivada, es por esto que no presenta discontinuidades en el tiempo y supone una de las posibilidades más apropiadas para realizar una trayectoria lo más suavizada posible, algo bastante recomendable en la caminata de un robot. Además al ser una función periódica solo es necesario analizarla en el intervalo  $(0 \text{ a } 2\pi)$ .

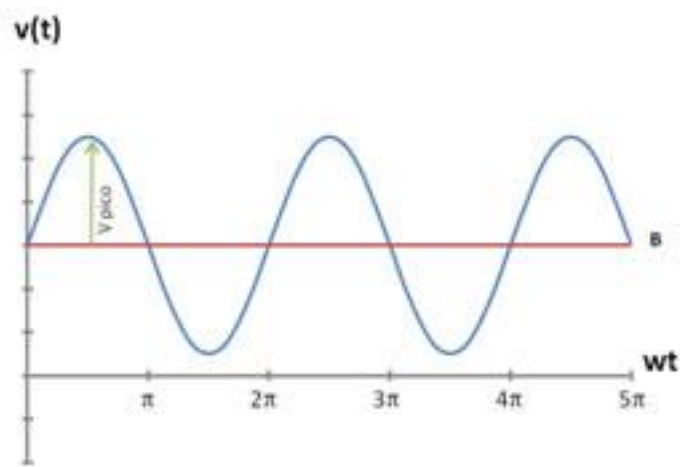


Figura 3.5: Función Seno

- **Señal de pulso:** este tipo de trayectoria se utiliza en ocasiones en las que interesa que el movimiento se realice en un momento determinado, y que posteriormente regrese a su estado de reposo. Esto se representa mediante la función delta de Dirac,  $y = \delta(x)$ . En la práctica, ese instante es demasiado corto como para que los motores actúen y se realice el movimiento deseado, es por eso que es necesario aumentar el ancho de dicho pulso.

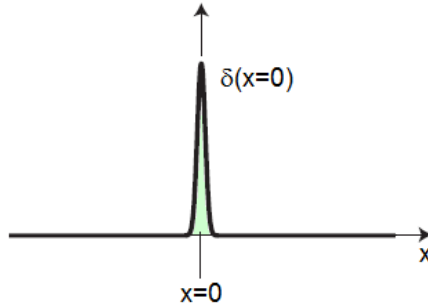


Figura 3.6: Función Delta de Dirac

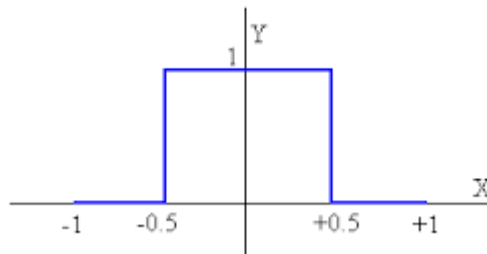


Figura 3.7: Función Pulso

Se puede observar en la *figura 3.6* como el pulso ocurre en un instante ínfimo en el cual no hay tiempo posible para que los motores actúen. Por eso al aumentar el ancho de pulso (*figura 3.7*) se consigue tener el tiempo suficiente para que los motores realicen los movimientos deseados.

- **Señal triangular:** Este tipo de trayectoria se utiliza cuando hay que llegar a dos puntos simétricos con respecto a un eje y en los que se emplea una velocidad constante pero de signo opuesto en cada uno de los sentidos.

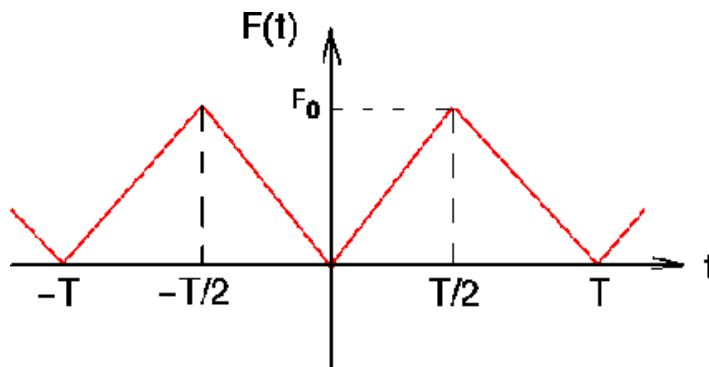


Figura 3.8: Función Triangular



### 3.2.2 ESTRATEGIAS PARA MANTENER/RECUPERAR EL EQUILIBRIO EN HUMANOS

Los seres humanos son un modelo a seguir para la creación y estudio de robots humanoides; Es por eso que comprender como los seres humanos son capaces de mantener el equilibrio y llevar a cabo trayectorias dinámicas sin desequilibrarse, es un punto fundamental en el ámbito de las estrategias de caminata.

Ante un factor que provoca balanceo y desequilibrio, los seres humanos tienen varias estrategias [10] según el tipo de factor que provoca el desequilibrio:

- **Ankle Strategy** (estrategia que emplea los tobillos): este tipo de estrategia es empleada cuando la perturbación es lenta y con una amplitud baja. Además, el contacto con la superficie sobre la que se encuentra el humano, debe de ser firme y con un área mayor que el de los pies.

Los movimientos de la cabeza están en fase con los de la cadera.



Figura 3.9: Ankle Strategy

- **Hip Strategy** (estrategia que emplea la cadera): esta estrategia se usa cuando la perturbación es rápida o con una amplitud grande. En este caso la superficie de contacto es inestable o con un área menor que el de los pies. Además, la cabeza realiza movimientos desfasados con respecto a los movimientos realizados por la cadera.

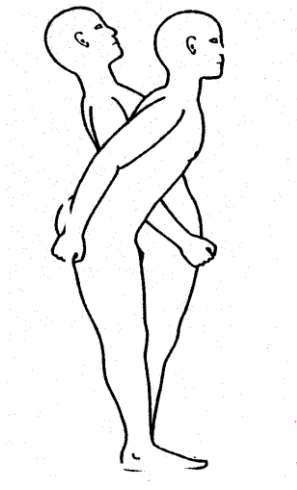


Figura 3.10: Hip Strategy

- **Stepping Strategy** (estrategia que emplea el dar un paso): este tipo de estrategia se utiliza para prevenir caídas ante perturbaciones rápidas, de gran amplitud o cuando alguna de las estrategias mencionadas anteriormente falla.

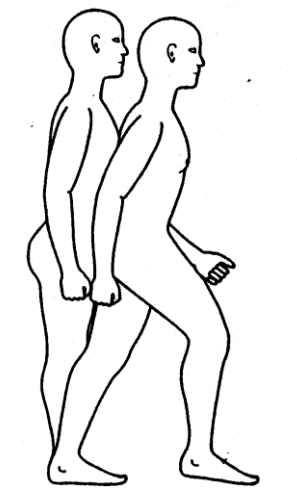


Figura 3.11: Stepping Strategy

- **Suspensory Strategy**: este tipo de estrategia consiste en doblar hacia adelante el tronco, flexionando la cadera y las rodillas, llegando incluso a una posición de cuclillas.

De esta forma, se consigue que la posición del centro de gravedad baje, logrando de esta forma mayor estabilidad.

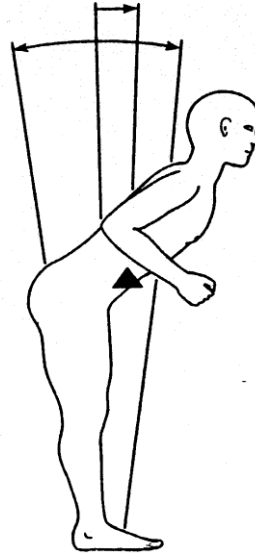


Figura 3.12: Suspensory Strategy

### 3.2.3 CAMINATAS ESTÁTICAMENTE ESTABLES Y DINÁMICAMENTE ESTABLES

En el capítulo 1.3 se introdujeron los conceptos de estabilidad estática y dinámica. En este capítulo se profundizará en estos conceptos [11] a la hora de emplearlos en la caminata del robot.

- **Caminata estáticamente estable:** este tipo de caminata se produce con movimientos lo suficientemente lentos como para despreciar los efectos de las fuerzas de inercia.

Una caminata estáticamente estable se logra enlazando diferentes posturas, todas ellas estáticamente estables (la proyección del centro de gravedad queda dentro del polígono de soporte). En caso de que el robot camine rápido, la inercia que se produciría, provocaría que la proyección del centro de gravedad pudiera salirse del polígono de soporte y de esta forma desestabilizarse.

Cuando un robot bípedo realiza un movimiento de caminar hacia adelante, dicho movimiento se puede dividir en dos: balanceo y avance.

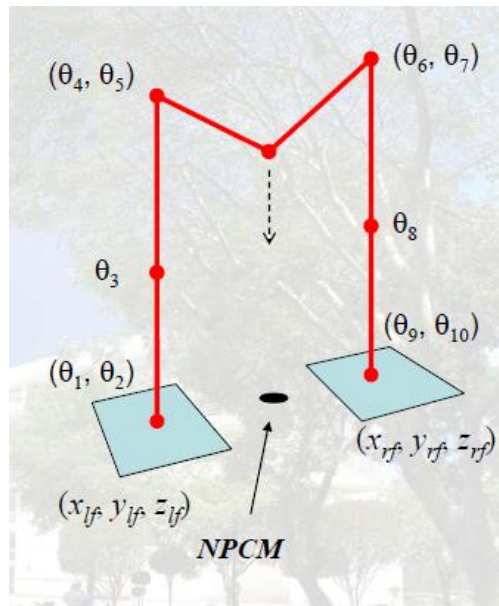


Figura 3.13: Esquema de las articulaciones de un bipedo

Para el movimiento de balanceo se emplean las articulaciones 2, 5, 7 y 10, mientras que para el de avance se utilizan las articulaciones 3, 4, 6 y 8. Las articulaciones restantes (1 y 9) se utilizan para mantener el pie en una posición paralela al suelo.

La numeración de cada una de las articulaciones mencionadas se puede observar en la *figura 3.13*

De esta forma, la manera en la que camina cualquier bipedo, ya sea humano o robot, se puede dividir en las fases que se aprecian en la *figura 3.14*

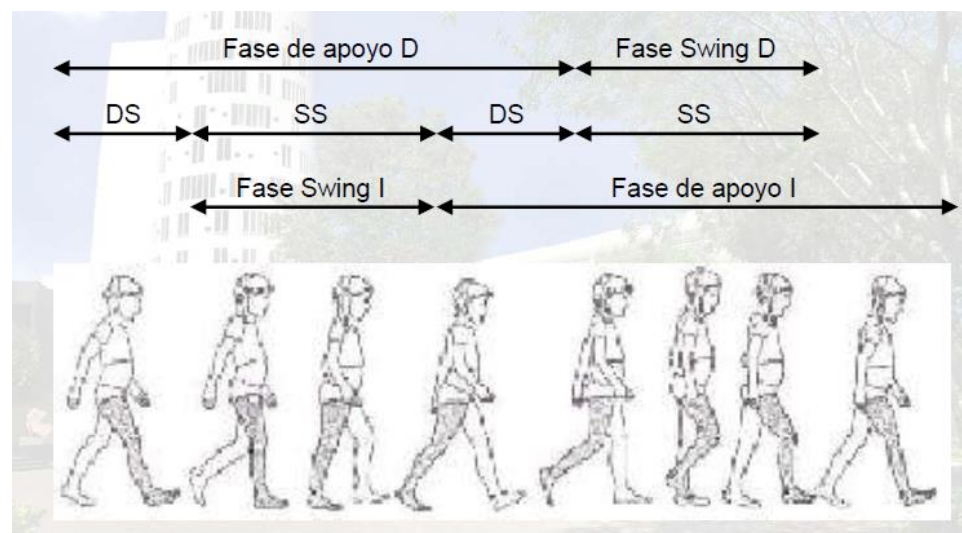


Figura 3.14: Fases en la caminata de un bipedo

Por tanto, de forma resumida la caminata estáticamente estable se caracteriza principalmente por:

- Aceleración despreciable en el centro de masas.
  - El bípedo estará estable si la proyección del centro de masas está dentro del polígono de soporte.
  - Los movimientos son sumamente lentos para no generar inercias.
- **Caminata dinámicamente estable:** este tipo de caminata se caracteriza porque tiene en cuenta las fuerzas que actúan sobre los puntos de soporte. Estas fuerzas provienen de los movimientos inerciales que realiza el bípedo, y posibilita que este se pueda mover a velocidades mayores, no como en la caminata estáticamente estable.
- Además, al tener en cuenta este tipo de fuerzas inerciales, el hecho de que la proyección del centro de gravedad esté dentro del polígono de soporte, no garantiza que el robot sea estable y no pierda el equilibrio.

#### 3.2.4 ZERO MOMENT POINT (ZMP)

El ZMP [12] o punto de momento nulo, es el punto en el cual el momento neto de las fuerzas externas es cero en los ejes  $x$  e  $y$ , (ejes horizontales del tobillo). Si este punto está dentro del polígono de soporte, entonces el robot realizará un movimiento estable dinámicamente. Cuanto más cercano esté el ZMP del centro geométrico del polígono de soporte (pie), mayor será la seguridad con la que se produzca la estabilidad. Esta estabilidad dinámica se garantiza siempre que el ZMP esté dentro de la superficie de soporte.

Esta idea surgió hace más de 35 años, y se aplicó por primera vez en Japón en 1984 en *Waseda University, Laboratory of Ichiro Kato*, en el robot *WL-10RD* [13]. Este robot fue el primero en estar equilibrado dinámicamente mediante este método.

A continuación se procede a realizar una breve y resumida explicación sobre el desarrollo de este método.

El suelo al estar en contacto con el pie genera una reacción sobre él. Esta reacción está formada por un momento  $M$  y una fuerza  $F$ . Con la articulación del tobillo ocurre lo mismo, se genera una reacción formada por una fuerza  $F_A$  y un momento  $M_A$ . Estas reacciones se pueden simplificar a una fuerza que crea un momento en el tobillo cuando se aplica en un punto concreto de la planta. Tanto las fuerzas como los momentos están

formadas por sus tres componentes (X,Y,Z). Suponiendo que no hay deslizamiento entre el pie y el suelo, las componentes horizontales de  $F$  y  $F_A$  se anulan al quedar compensadas unas con otras. Ocurre lo mismo con las componentes verticales de  $M$  y  $M_A$ , quedan compensadas una con la otra.

De esta forma, todo el problema se simplifica a estudiar las componentes verticales de  $F$  y  $F_A$ , y las componentes horizontales de  $M$  y  $M_A$ . Al tener las fuerzas una única componente, la componente  $Z$ , esta se puede compensar encontrando el punto en el que se anule con las componentes horizontales de los momentos. Este punto es el *Zero Moment Point*. Este ZMP siempre estará dentro del polígono de soporte cuando el robot sea estable, sin embargo cuando el robot pierde el equilibrio y cae, es debido a que el ZMP se encuentra fuera del polígono de soporte.

Este método tiene grandes ventajas:

- El ZMP asegura que las trayectorias de movimiento sean dinámicamente estables.
- Es posible ir corrigiendo sobre la marcha los movimientos y la posición del ZMP en caso de que este se desvíe de su posición previamente programada. Esta desviación ocurre debido a que una vez que el robot realiza los movimientos en el mundo real, hay perturbaciones (irregularidades en el suelo, agentes meteorológicos, rozamientos...) que hacen que la posición calculada del ZMP no se corresponda con la real.

Por otro lado, esta estrategia también presenta algunos inconvenientes, aunque hay que indicar que son más las ventajas que aporta este método que los problemas que causa.

- Al usar cinemática inversa, es necesaria una gran precisión para conseguir estabilidad a la hora de moverse.
- Aparecen errores numéricos debidos a la existencia de las singularidades, que provocan desequilibrios y comportamientos extraños del robot.
- Es un método bastante complejo en cuanto a capacidad de cálculo se refiere, aunque con el continuo avance en capacidad de proceso, ese problema irá disminuyendo paulatinamente.

Como último apunte acerca del *Zero Moment Point*, indicar que los robots bípedos que mejor se desenvuelven actualmente a la hora de moverse emplean el ZMP, un claro ejemplo de esto es el robot ASIMO de Honda [14].

## 4 MARCO REGULADOR

En este apartado se resaltan, en caso de que existieran, las principales normas y regulaciones que afectan a este proyecto.

Este tipo de normas y regulaciones de obligado cumplimiento afectan principalmente a la ubicación del proyecto en sí. Debido a que este trabajo se centra en la realización de un robot mini-humanoide, no hay ninguna normativa al respecto que el proyecto deba cumplir.

Es por eso que en el capítulo 4.1.2, se hablarán de las normas y pruebas del concurso CEABOT, concurso en el que se establece una competición entre distintos robots mini-humanoides, y cuyo desarrollo está regido por una serie de normas que los robots deben cumplir.

### 4.1 MARCO DE TRABAJO

En este apartado se comentan aquellos acontecimientos que han precedido e influido por diversas causas en el desarrollo de este proyecto.

#### 4.1.1 EVOLUCIÓN DE LA ROBÓTICA HUMANOIDE EN LA UC3M

La asociación de robótica de la Universidad Carlos III de Madrid, lleva trabajando desde 2006 con robots humanoides para la investigación y competición.

Estos robots han ido evolucionando dependiendo de la plataforma robótica que utilizaban.

En primer lugar, se utilizó la plataforma robótica *Robonova* de *Hitec*, en la que se realizaron mejoras para alojar sensores adicionales.

Posteriormente, se sustituyó la placa de *Robonova* por una *Arduino* con mayor libertad a la hora de la programación. Las ventajas de emplear una placa *Arduino* son:

- Sistema y lenguaje muy sencillo de utilizar
- Existen miles de librerías *open-source* disponibles para su uso

Estas placas tienen un inconveniente, y es que debido a esa sencillez no se pueden comunicar con ciertos elementos y periféricos (como pueden ser los servos *Dynamixel AX-12A* que se emplean en este trabajo).

Seguidamente, se adquirió un kit de *Bioloïd*, que al ser más moderno permitía un mejor control de los movimientos del robot, pero tenía el inconveniente de que sus capacidades de sensorización y programación eran bastantes limitadas.

Por eso a partir de 2013 se empezó a rediseñar las piezas con el objetivo de que, con piezas mecánicas nuevas y nueva electrónica, las capacidades del robot aumentarían. Para el rediseño de las piezas entra en juego un actor principal, las impresoras 3D, con las cuales el diseño y ejecución de las piezas se produce de una manera bastante simple y ágil. Gracias a las impresoras 3D de los laboratorios de la Universidad Carlos III, se han podido imprimir las piezas rediseñadas para realizar una nueva versión más avanzada del robot RAIDER.

#### 4.1.2 CAMPEONATO CEABOT

Un acontecimiento de gran importancia para la UC3M, es el CEABOT [15]. El CEABOT es un campeonato en el que compiten universidades de todo el país y gente a nivel particular, en el cual, sus robots mini-humanoides son sometidos a diferentes pruebas, y clasificados posteriormente según la puntuación ponderada en cada una de estas pruebas. A continuación se explicará en que consiste brevemente cada una de ellas.

1ª prueba: El robot realiza una carrera de ida y vuelta salvando unos obstáculos colocados de forma aleatoria por el jurado antes de comenzar la prueba. Este año incluye la novedad de que las paredes tendrán un código QR para facilitar al robot su ubicación. Ver *figura 4.1*.

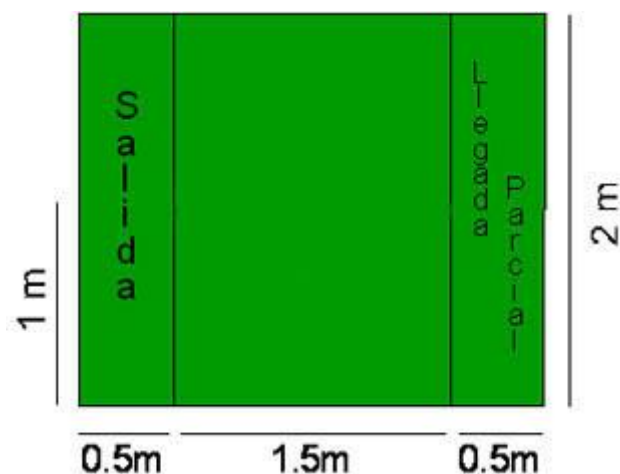


Figura 4.1: CEABOT, prueba 1

2ª prueba: El robot realiza una carrera (solo ida), teniendo que subir y bajar los peldaños (3 cm de altura) de una escalera como la de la *figura 4.2*.



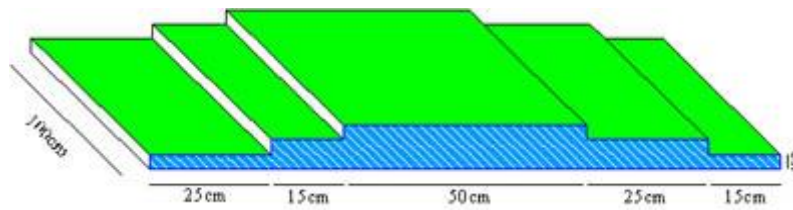


Figura 4.2: CEABOT, prueba 2

**3ª prueba:** Esta prueba consiste en una serie de combates entre robots. Los combates consistirán en 3 asaltos de 2 minutos cada uno. Entre asalto y asalto habrá un tiempo máximo de 1 minuto. Gana el robot que haya ganado más asaltos. En caso de empate a asaltos, se desempatará con la suma total de los puntos Yuhkoh conseguidos. En caso de empatar a suma de puntos Yuhkoh, se hará un asalto más. El escenario de combate se puede observar en la *figura 4.3*.

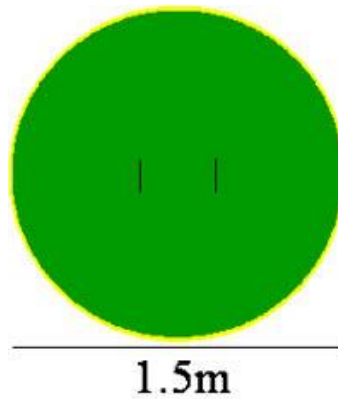


Figura 4.3: CEABOT, prueba 3

**4ª prueba:** El robot se coloca en el centro del escenario (*figura 4.4*) encarado al primer obstáculo. En dicho obstáculo, habrá un QR que tendrá que decodificar para saber dónde se encuentra el siguiente obstáculo. Cuantos más obstáculos localice en el menor tiempo, más puntos recibe.

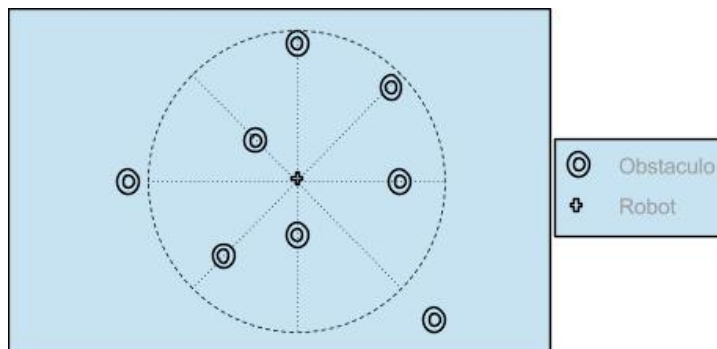


Figura 4.4: CEABOT, prueba 4

En este contexto, este proyecto parte del Trabajo de Fin de Grado mencionado anteriormente realizado por Javier Isabel Hernández, “Desarrollo de una plataforma robótica mini-humanoide con visión artificial” dirigido por Félix Rodríguez Cañadillas y tutorizado por Alberto Jardón Huete, ya que ha servido de guía tanto a nivel de documentación como a nivel práctico.

En dicho TFG también se realizaba el estudio y rediseño de un robot mini-humanoide a partir de una base comercial. Este robot actuaba mediante visión artificial, y tenía absoluta movilidad bípeda y control de caídas. Este proyecto se basó en la plataforma comercial *Bioid Comprehensive*, aunque se realizaron modificaciones sobre él. Empleaba un controlador para la locomoción CM9.04 y una *Beaglebone Black* como controlador principal, además de utilizar una cámara para la visión, sensores infrarrojos, brújula, giróscopo, servos *Dynamixel AX-12A*... Gracias a este trabajo se consiguieron mejoras que posteriormente se utilizaron para la competición CEABOT.

## 5 DESCRIPCIÓN DEL SISTEMA

En este capítulo se describirá, de forma general, como es el robot de este trabajo y de que partes y elementos está formado, sin entrar en detalles. El montaje detallado del robot completo se detallará en el capítulo seis.

### 5.1 ESTRUCTURA FÍSICA

Toda la estructura del robot está formada por piezas previamente diseñadas, y posteriormente impresas en 3D. Con estas piezas, y mediante el uso de los servos *Dynamixel*, se consigue la unión de las diferentes partes hasta formar la estructura física del robot.

El listado de piezas, las cuales se mostrarán en el capítulo 12.1, es el siguiente:

- *Body*
- *Body back*
- *Hip chasis*
- *Hip top*
- *Knee*
- *Knee damper*
- *Left & right ankle*
- *Left & right foot*
- *Left & right leg bracket*
- *Lower left & right leg*
- *Upper left & right leg*
- *Left & right arm*

Además, tal y como se ha mencionado anteriormente, se emplean 21 servos de la clase *Dynamixel AX-12A*, que dotan al robot de 21 grados de libertad, con los que se consigue un alto grado de movilidad y un gran número de combinaciones en sus movimientos.

### 5.2 ELECTRÓNICA

Una vez montada la estructura física, se pasa al apartado electrónico. Primero se suelda la placa del robot con sus diferentes elementos electrónicos (resistencias, condensadores, pines, etc), a esta placa se le unirá la controladora CM9.04 y la *Beaglebone black*, que son las encargadas del control de los servos y del control global

respectivamente. Además se le añade un acelerómetro, para dotar al robot de la capacidad de sensor su inclinación y conocer su posición actual. A diferencia del robot de Javier Isabel, este robot no lleva cámara ni brújula, ya que no son necesarios para el propósito requerido.

En la *figura 5.1* se muestra un esquema del conexionado de las diferentes placas y sensores del robot:

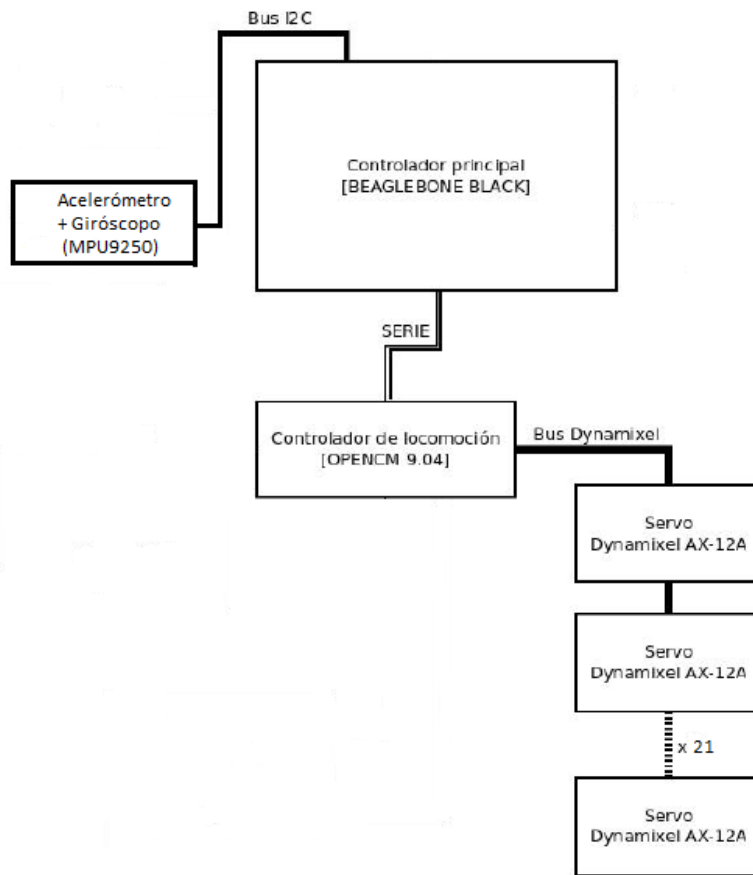


Figura 5.1: Conexionado de placas y sensores

### 5.3 ESPECIFICACIONES TÉCNICAS DEL ROBOT

En este apartado se comentan las especificaciones técnicas más importantes del robot de este proyecto.

El robot RAIDER de este trabajo mide 35 cm de altura y tiene una envergadura (medida desde el extremo de un brazo hasta el extremo del otro brazo) de 50 cm. En cuanto al peso del robot, decir que este mini-humanoide ronda los 2500 gramos.

En la *Tabla V* se muestran otras especificaciones técnicas del robot.

Par servos	15.3 kg·cm
Electrónica	<i>BeagleBone Black</i> rev C (AM3358 processor)
Número de servos	21

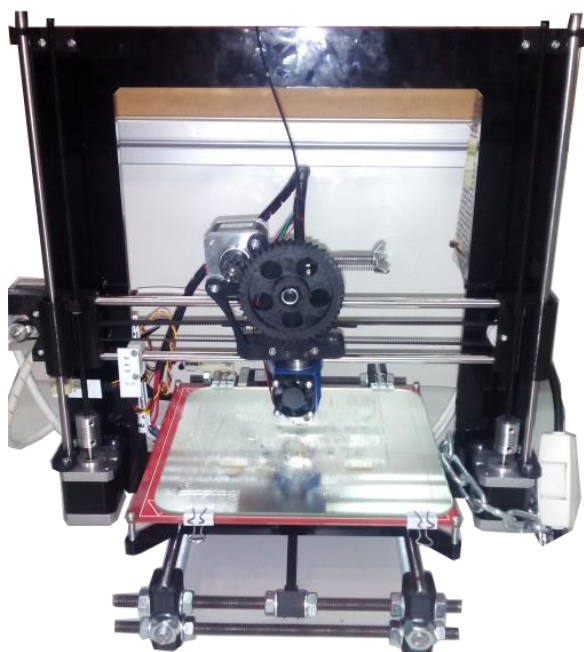
*Tabla V: Especificaciones técnicas del robot*

## 6 MONTAJE FÍSICO Y ELECTRÓNICO DEL ROBOT

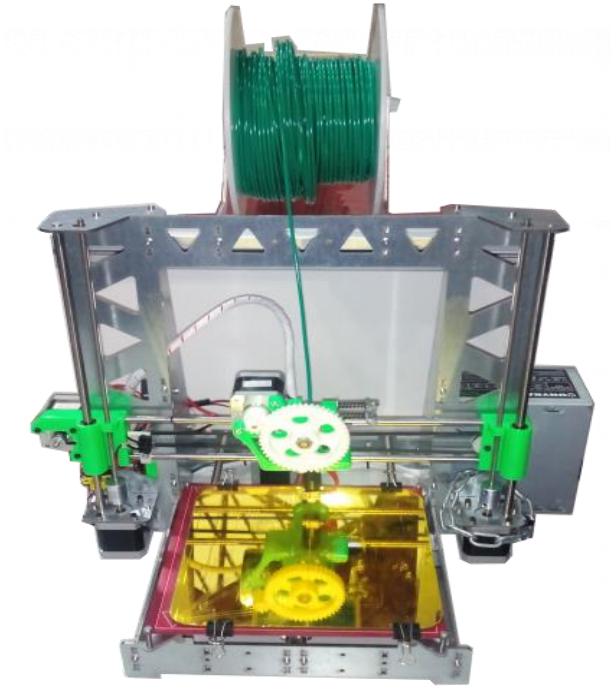
### 6.1 IMPRESIÓN EN 3D DE LAS PIEZAS DEL ROBOT

Una vez realizados los diferentes estudios sobre tipos de robots mini-humanoides y los distintos modos de locomoción o caminatas, el siguiente paso es el de la impresión de las piezas que se utilizan para el posterior montaje del robot, ya que en este trabajo no se parte de una plataforma robótica ya creada o comprada, si no que se crea el robot desde cero.

Dicha impresión se realiza en dos impresoras 3D situadas en el laboratorio 1.0.B06 de la Universidad Carlos III de Madrid. Estas impresoras (*figuras 6.1 y 6.2*) creadas por el departamento de robótica, facilitan enormemente la fabricación de piezas para cualquier propósito, ya que gozan de una gran flexibilidad y versatilidad.



*Figura 6.1: Impresora "Blacky"*



*Figura 6.2: Impresora "Hija-Resurrection"*

Partiendo de los planos de las piezas ya hechos y procedentes del proyecto de referencia, se imprime cada una de las piezas necesarias.

Otras piezas que también son empleadas en el montaje del robot, pero que no se han imprimido en estas impresoras 3D, si no que han sido proporcionadas por el departamento de Robótica, son las piezas de las *figuras 6.3 y 6.4*.



*Figura 6.3: Pieza originaria del robot Bioloid proporcionada por el departamento de robótica*

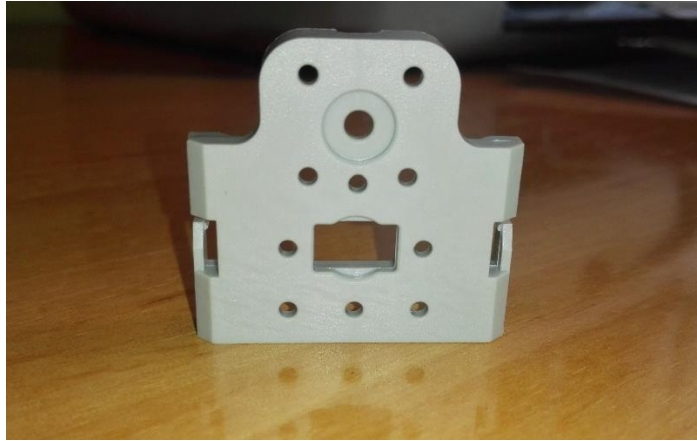


Figura 6.4: Pieza originaria del robot Bioloid proporcionada por el departamento de robótica

La pieza de la *figura 6.3*, ha sido empleada para unir dos servos, *figura 6.5*, mientras que la pieza de la *figura 6.4*, se ha utilizado para el montaje del pie, *figura 6.6*.



Figura 6.5: Unión de dos motores





Figura 6.6: Tobillo del robot

Para acabar, el último elemento necesario para el montaje del robot, a parte de los tornillos de distintas métricas, tuercas, etc, son los servos empleados en este proyecto, que como ya se indicó líneas arriba, son del modelo *Dynamixel AX-12A*, figura 6.7.



Figura 6.7: Servo Dynamixel AX-12A

## 6.2 MONTAJE FÍSICO

La construcción y montaje de este robot constituye uno de los puntos más complicados del proyecto, debido a que no existen planos ni instrucciones con los que seguir una serie de pasos para su montaje.

La única guía o ayuda con la que se cuenta, es un mini-humanoide Bioloid (*figura 6.8*) proporcionado por el departamento de Robótica, de estructura y forma similar pero con notables diferencias con respecto al robot a construir. Aparte de tenerlo como modelo, se emplean varias piezas, como son los brazos completos.

Es por este motivo por el cual en este trabajo, además de realizar los objetivos marcados para el mismo, adicionalmente se ha creado un manual de instrucciones para la construcción del robot mini-humanoide RAIDER. Este manual se adjunta en el anexo “**Manual de montaje del robot RAIDER**”, capítulo 12.1.1



*Figura 6.8: Mini-Humanoide modelo*

La otra ayuda con la que se cuenta, es una foto del robot empleado en el TFG de Javier Isabel Hernández, *figura 6.9*. Dicho robot si se asemeja en gran medida al de este proyecto, aunque con notables diferencias entre ambos.



Figura 6.9: Robot de Javier Isabel Hernández

### 6.3 MONTAJE ELECTRÓNICO

Una vez realizado el montaje físico, el siguiente paso es la electrónica del robot. Los elementos electrónicos que se emplean en este proyecto son:

- 2 Resistencias de 10 Kohmios de  $\frac{1}{4}$  W
- 2 Resistencias de 12 Kohmios de  $\frac{1}{4}$  W
- 2 condensadores de 100uF
- 2 tiras de pines rectos macho de 2x23
- 2 tiras de pines rectos hembra de 1x20
- 3 tiras de pines rectos macho de 1x4
- 1 tira de pines rectos macho de 1x3
- 1 tira de pines codo macho de 1x3

Además de estos elementos electrónicos, también se utiliza un conversor *UBEC* de 5V, una batería *LIPO* 3S de 11.1V 2200mah, así como cables para realizar diferentes conexiones.

El montaje electrónico viene detallado en el capítulo 12.1.2, **Manual de montaje electrónico**.

## 6.4 PREPARACIÓN DEL SOFTWARE DE LA BEAGLEBONE BLACK

La *BeagleBone Black* empleada en este proyecto es la edición *rev C*, cuyas características más importantes son:

- Procesador TI Sitara AM3358AZCZ100, 1GHz
- Memoria SDRAM de 512 Mb
- Memoria Onboard Flash de 4Gb
- Slot para tarjeta microSD
- Se alimenta de dos formas posibles: O mediante miniUSB o mediante fuente de alimentación externa de 5V

Las características completas de la BBB rev C se observan en la *figura 6.10*



Figura 6.10: Características técnicas de la BBB rev C

En este proyecto, se ha decidido iniciar la BBB desde una nueva imagen de *Debian* instalada en una tarjeta de memoria microSD, ya que la *BeagleBone Black* permite la

opción adicional de iniciar el sistema desde una imagen grabada en una tarjeta de memoria.

El motivo por el cual se decidió iniciar la BBB desde una tarjeta SD y no desde su sistema embebido por defecto, es que si se iniciaba desde la imagen de *Debian* que venía por defecto, la BBB era incapaz de realizar la *conexión ssh*.

De esta manera, se instaló en una tarjeta microSD la última imagen de Debian para BBB, más concretamente la versión 8.6 que data a fecha de 11/6/2016 (Las imágenes para las distintas *BeagleBones* se encuentran en la siguiente web [16]). Esta imagen ocupa alrededor de 4Gb, por lo que es necesario emplear una microSD igual o superior a dicho tamaño. Puesto que aparte de la nueva imagen, en la BBB hay que instalar otro tipo de librerías y programas como *QT Creator*, se decidió elegir una microSD de 32GB para así no tener problemas de almacenamiento de memoria en la placa.

Una vez instalada la nueva imagen en la tarjeta microSD, y expandida dicha imagen a todo el volumen de la tarjeta, se procedió a la instalación de las numerosas librerías necesarias para el correcto funcionamiento tanto de la *BeagleBone Black* como del código de RAIDER.

Este tipo de librerías (*Zbar*, *i2c*, *cmake* ó *fuzzylite*) hacen posible compilar el código de RAIDER sin errores o *warnings*.

Como último paso para dejar preparada la BBB para su posterior programación del código, quedaba descargar el código de RAIDER en la misma, para lo cual se procedió a realizar un comando "*git clone*" sobre el repositorio de código de RAIDER que hay en el perfil de Javier Isabel en *GitHub* [17].

Una vez realizadas las distintas instalaciones, la *BeagleBone Black* queda preparada y lista para programar el código del robot.

## 7 SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO Y SELECCIÓN DE LA TÉCNICA DE LOCOMOCIÓN

En este capítulo se detallarán las herramientas de desarrollo o programas informáticos que se han empleado para el desempeño de este proyecto, así como la elección de la técnica de locomoción que utilizará el robot.

### 7.1 HERRAMIENTAS DE DESARROLLO

Como se ha mencionado con anterioridad, el sistema de control del robot estará protagonizado por la *BeagleBone Black*, acompañada por la OpenCM 9.04 encargada del movimiento de los servos *Dynamixel*. Puesto que son dos placas distintas, para trabajar con cada una de ellas es necesario emplear una herramienta de desarrollo concreta. A continuación se nombran y explican las herramientas de desarrollo que se han utilizado en este proyecto para trabajar con la BBB y la OpenCM 9.04.

#### 7.1.1 QT CREATOR

Para desarrollar el código empleado en la BBB se utiliza el entorno de desarrollo integrado (IDE por sus siglas en inglés) *Qt Creator* [18], compatible con Linux, Windows y Mac.

*Qt Creator* puede trabajar con distintos lenguajes de programación como: C++, QML o JavaScript. En este proyecto el código de la BBB está escrito en C++, puesto que al ser uno de los lenguajes más comunes y desarrollados, dispone de una inmensidad de bibliotecas disponibles para su uso.

Además, es capaz de trabajar por defecto con archivos de tipo *CMake*. Gracias a estas características, *Qt Creator* supone una herramienta ideal para el desarrollo de este proyecto.

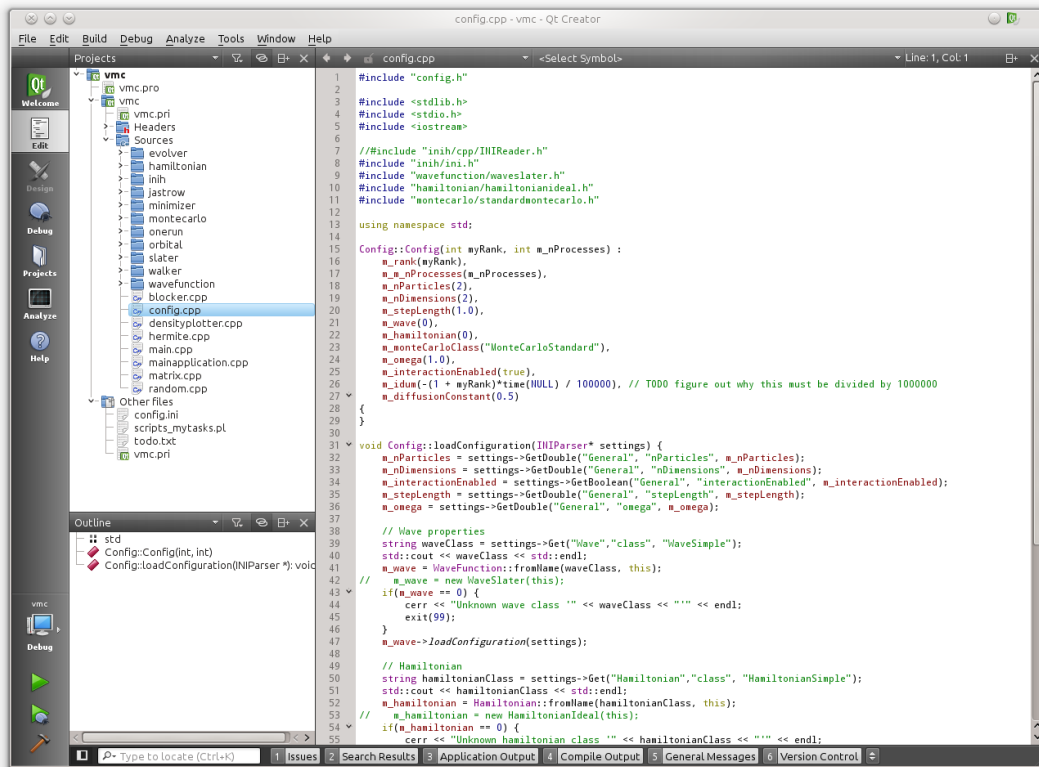


Figura 7.1: Qt Creator

## 7.1.2 ROBOTIS OPENCM

Esta IDE [19] de *Robotis* es una plataforma con *hardware* y *software* en *open-source* que permite, mediante el uso de lenguaje C/C++, la programación referente al movimiento de los servos.

*Robotis OpenCM* está basado en *Arduino IDE*, por lo que aparte de C/C++, también soporta las librerías de *Arduino*. Señalar que en este proyecto se ha empleado la versión 1.0.0, puesto que con versiones superiores había problemas a la hora de cargar el código en la OpenCM 9.04, cosa que no ocurre con dicha versión.



```
motion | ROBOTIS CM9 v1.0.0
Archivo Editar Sketch Herramientas Ayuda
motion raider_motion.cpp raider_motion.h
#include <Servo.h>
#include "raider_motion.h"

//STATE SIGNALS
#define STOP_BUTTON 0
#define STATE_LED 1

//BAUD RATE FOR COMMUNICATION
#define BBB_BAUD_RATE 9600
#define BT_BAUD_RATE 9600

Robot raider;

void setup() {
  // Serial2.begin(BT_BAUD_RATE); //Bluetooth
  Serial3.begin(BBB_BAUD_RATE); //Beaglebone
  // SerialUSB.begin();

  pinMode(SOARD_LED_PIN, OUTPUT);
  pinMode(STOP_BUTTON, INPUT);
  pinMode(STATE_LED, OUTPUT);

  raider.init();
}

bool flag=0;
```

Figura 7.2: Robotis OpenCM

## 7.2 TÉCNICA DE LOCOMOCIÓN

La técnica de locomoción del robot se basa en el código de RAIDER [20], pero adaptando dicho código al robot de este proyecto, ya que no son robots idénticos.

Entre las diferencias existentes entre ambos robots, cabe destacar una de vital importancia: tanto el número de servos como la disposición de los mismos es diferente. Mientras que el robot RAIDER dispone de 20 servos, el de este proyecto tiene 21. La diferencia en el número de servos se debe a que el mini-humanoide de este trabajo tiene un motor adicional en el extremo del brazo (este servo adicional se puede observar en la *figura 7.3*), proporcionando de esta forma dos grados de libertad extras (uno por cada brazo), haciendo que su movimiento sea más natural y humanoide. Por otro lado, no lleva servo en la cabeza, ya que este robot no dispone de cámara por lo que no es necesario.



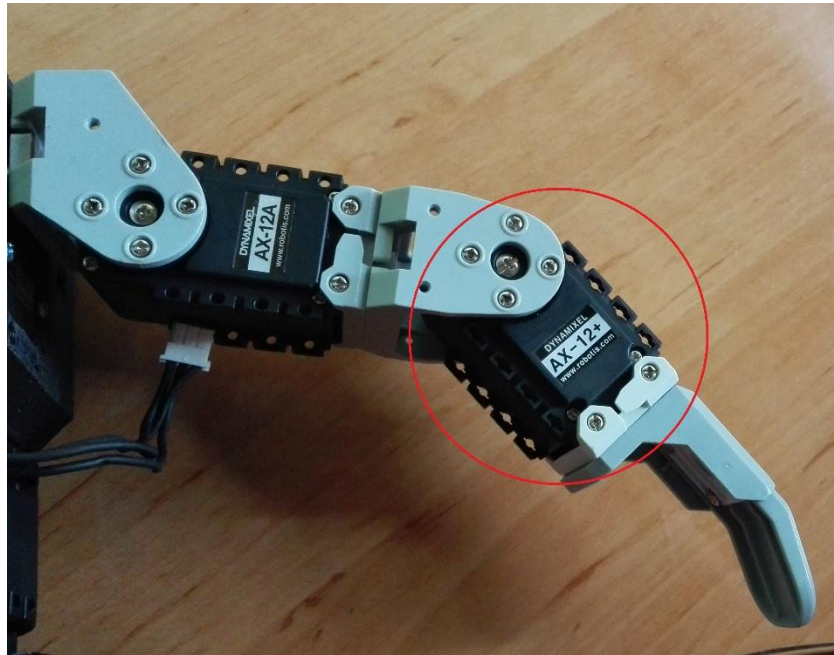


Figura 7.3: Servo adicional en el extremo del brazo

La secuencia de locomoción del robot se divide en varias fases, para que de esta forma su desarrollo y comprensión sea más sencilla. En este capítulo se explicará cada una de estas fases pero sin entrar en detalles de código (esto se hará en el capítulo 8).

En primer lugar, se ha parametrizado el paso izquierdo y el derecho mediante una serie de puntos. Estos puntos están definidos por sus tres coordenadas cartesianas y sus tres coordenadas de orientación, de esta forma, para que el robot dé un paso completo, debe de ir siguiendo estos puntos, yendo de uno hasta el siguiente hasta completar el paso.

Para seguir la trayectoria marcada por estos puntos, se utiliza las funciones *MovVertical*, *MovLateral* y *MovFrontal*. Estas funciones están basadas en las funciones originales de RAIDER, pero modificadas y adaptadas para que su empleo en este robot sea efectivo, puesto que las originales no son compatibles con este mini-humanoide (número de servos desigual, diferente colocación de los motores, distintas IDs adjudicadas...).

Comparando las *figuras 7.4* y *7.5* se observan las diferencias existentes entre los robots de ambos proyectos.



Figura 7.4: Robot de Javier Isabel

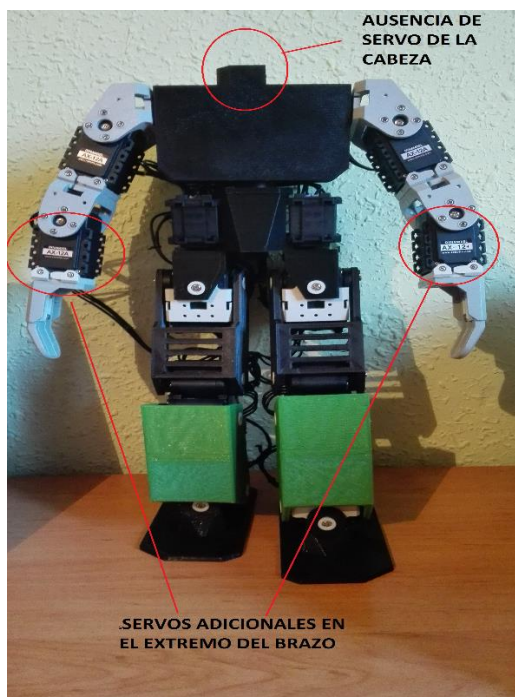
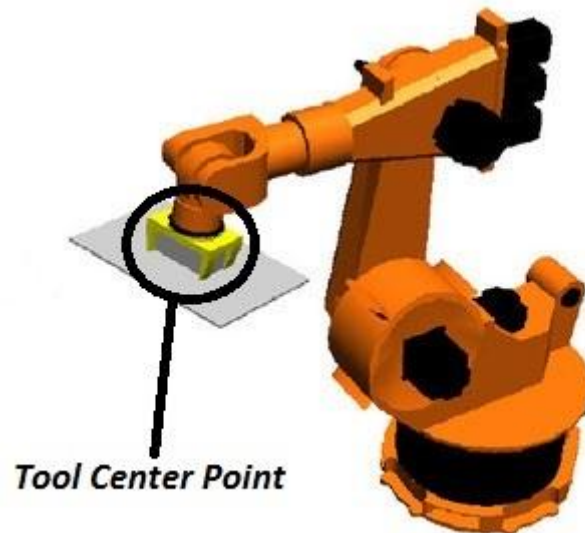


Figura 7.5: Robot de este proyecto

En robótica se trabaja con un concepto de vital importancia a la hora de realizar movimientos, este concepto es el de *Tool Center Point* (TCP). Este TCP o “punto central de la herramienta” sirve para referenciar una parte concreta del robot, la cual se va a mover de un lugar a otro. Como ejemplo típico de TCP están los robots que se emplean

en las cadenas de montaje de automóviles, en los cuales su TCP se encuentra en la punta del brazo robótico, que es donde tienen la herramienta (*Tool*) con la que trabajan, ya sea una pistola de pintura, una ventosa, un soldador, etc. Ver *figura 7.6*.



*Figura 7.6: Tool Center Point de un robot*

En el robot de este proyecto se toma como *Tool Center Point* el centro de la planta de cada uno de los pies. De esta forma, para que el robot dé un paso, el TCP debe de ir de un punto al siguiente de la trayectoria hasta completar todos los puntos de los que está formada dicha trayectoria. Para realizar este movimiento se emplean las funciones nombradas anteriormente según el movimiento que sea necesario realizar.

Simultáneamente a este proceso de locomoción, se lleva a cabo el proceso de equilibrado o estabilización del robot. Este es el punto novedoso y crítico de este proyecto, puesto que se quiere realizar una estrategia de caminata que sea estable en todo momento.

Como se ha mencionado líneas arriba, para que el robot camine, tiene que seguir la trayectoria marcada por unos puntos definidos con anterioridad. Si el robot únicamente se limitara a realizar los movimientos marcados por estos puntos, cuando el robot se encuentre sustentado solo por una pierna, como por ejemplo cuando da el paso derecho, el robot se caería hacia el lado derecho, ya que la proyección de su centro de gravedad (CDG) estaría fuera del polígono de soporte, que en este caso sería el área formada por el pie izquierdo, que es el que está en contacto con el suelo. Estos conceptos de CDG y de polígono de soporte están explicados en el capítulo 1.4.

Para evitar que el mini-humanoide se caiga, se realizan dos estrategias con las cuales el robot consigue mantener el equilibrio:

- 1- Desplazamiento de la proyección del CDG sobre el polígono de soporte de forma mecánica.
- 2- Control de la proyección del centro de gravedad mediante sensorización.

### 7.2.1 DESPLAZAMIENTO DE LA PROYECCIÓN DEL CDG SOBRE EL POLÍGONO DE SOPORTE DE FORMA MECÁNICA

Como se ha mencionado anteriormente, cuando el robot da un paso, la proyección del centro de gravedad queda fuera del polígono de soporte, más concretamente, hacia el lado de la pierna que da el paso y ligeramente hacia atrás con respecto al polígono de soporte. Para contrarrestar eso, el robot realiza una serie de movimientos previos al paso.

Cuando el robot va a dar un paso, por ejemplo, el paso derecho, antes de levantar la pierna derecha para dar el paso, el robot realiza un movimiento de inclinación hacia la izquierda, cargando gran parte del peso sobre dicha pierna.

Una vez realizado este movimiento, el mini-humanoide se inclina un par de grados hacia adelante, y gira el tronco hacia la izquierda.

Estos movimientos se realizan por los siguientes motivos. El primero de ellos, el de inclinación hacia la izquierda, logra desplazar la proyección del CDG hacia la izquierda, mientras que los movimientos de inclinación hacia adelante y giro del tronco hacia la izquierda, logran que dicha proyección se desplace hacia adelante. De esta forma la proyección del CDG queda desplazada hacia la izquierda y hacia adelante, quedando de esta manera dentro del polígono de soporte, que estaría formado por el pie izquierdo, logrando que al levantar la pierna derecha para dar el paso, el robot mantenga mejor su equilibrio.

De forma simétrica se realizaría el mismo proceso para dar el paso izquierdo.

En la *figura 7.7* se muestra al robot dando un paso SIN realizar los movimientos previos relatados, y en la *figura 7.8* realizando dichos movimientos.



Figura 7.7: Robot SIN realizar movimientos previos al paso

Como se aprecia en la *figura 7.7*, la proyección del CDG cuando el robot levanta la pierna izquierda para dar el paso, queda a la izquierda y ligeramente atrasada con respecto al pie derecho, que supondría el polígono de soporte, provocando que el robot se desequilibre hacia la izquierda.



Figura 7.8: Robot realizando los movimientos previos al paso

Por otro lado, en la *figura 7.8* se observa como al realizar los movimientos previos al paso descritos, la proyección del CDG se sitúa en el centro del pie derecho, proporcionando de esta manera la estabilidad deseada al robot.

## 7.2.2 CONTROL DE LA PROYECCIÓN DEL CENTRO DE GRAVEDAD MEDIANTE SENSORIZACIÓN

Los movimientos previos al paso, descritos anteriormente en el capítulo 7.2.1, ayudan en gran medida al robot a mantener su estabilidad, pero no garantizan que el robot se mantenga estable durante toda la caminata. Durante la misma, el robot sufre perturbaciones, ya que al andar puede encontrarse con terrenos con diferentes inclinaciones, rozamientos, agentes atmosféricos... que hacen que no valga simplemente con tener una trayectoria de paso diseñada de forma previa.

Para lograr que el robot sea estable en todo momento, hay que controlar de forma dinámica su estabilidad. ¿Cómo se logra esto? Mediante el control de la proyección de su centro de gravedad, es decir, mantener de forma continua que la proyección del CDG se mantenga dentro del polígono de soporte del robot. Dicho polígono de soporte varía según la posición en la que se encuentre el robot.

Por ejemplo, cuando el robot se encuentra sobre sus dos piernas, su polígono de soporte será mucho mayor que cuando el robot se encuentra únicamente sustentado por una de ellas. En la *figura 7.9* se observa como cuando el robot tiene los dos pies sobre el suelo, el polígono de soporte está formado por el área delimitada por los dos pies (línea blanca), siendo de esta forma mucho mayor que cuando el robot se encuentra sustentado solo sobre una de sus piernas, en cuyo caso, el polígono de soporte se corresponde con el área del pie que está apoyado en el suelo.



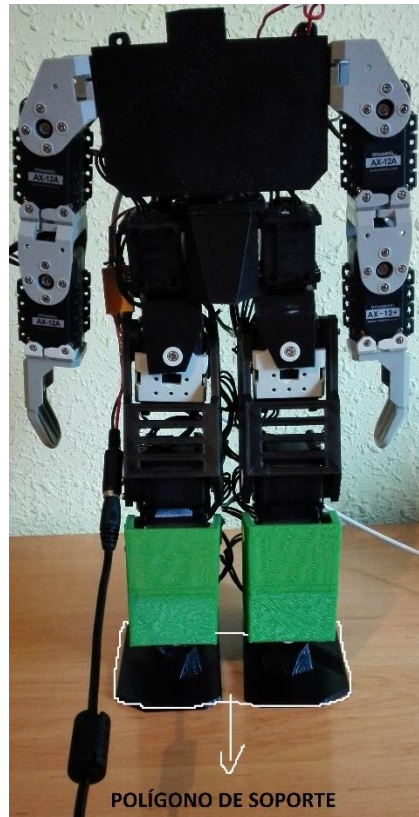


Figura 7.9: Polígono de soporte sobre dos piernas

Para lograr este propósito, el robot dispone de un IMU (*Inertial Measurement Unit*), concretamente el modelo *MPU 9250* [21], que se compone de un giróscopo de 3 ejes, un acelerómetro de 3 ejes y una brújula. Aunque esta IMU dispone de varios dispositivos, en este proyecto solo se utilizará el acelerómetro, ya que es lo que se requiere para realizar el control de equilibrio del robot.

El funcionamiento del control de equilibrio del robot se describe a continuación:

La BBB captura lecturas del acelerómetro cuando el TCP llegue al siguiente punto de la trayectoria del paso. Estas lecturas proporcionan el valor de  $G$ 's (unidad de medida de la aceleración, causada por el peso de un objeto o persona) en los ejes X, Y, Z. Solo se utilizarán las medidas de los ejes X e Y, puesto que el giro en Z es prácticamente despreciable debido a que el robot anda en línea recta.

Para que al lector le resulte más sencillo situar los ejes del robot, estos se pueden observar en la *figura 7.10*

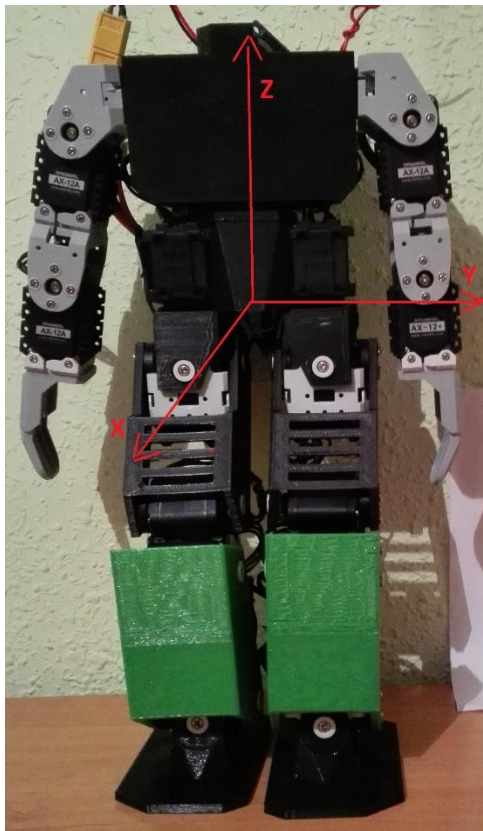


Figura 7.10: Ejes de coordenadas del robot

Como se puede apreciar, el giro en X del robot supone que este girará de forma lateral hacia la izquierda o derecha. Por otro lado, el giro en Y supondría un giro hacia adelante o hacia atrás. Finalmente, el giro en Z sería un cambio en la posición hacia la que “mira” el robot, pero como se ha indicado antes, el giro en Z se desprecia ya que el robot solo realiza trayectorias en línea recta.

Una vez que el acelerómetro toma las medidas en G's, estas se convierten a grados, para que su entendimiento sea más fácil a la hora de realizar el código. Partiendo de que el robot se encuentra en equilibrio cuando el giro en X e Y marcado por el acelerómetro es igual a cero grados, se realizan correcciones en la posición del robot cuando estos valores proporcionados por el acelerómetro son distintos de cero.

Si cuando el TCP (en este caso el pie del robot) llega al siguiente punto de la trayectoria, la lectura proporcionada por el acelerómetro es distinta de cero, entonces se corrige y almacena el valor medido por el acelerómetro, para que la siguiente vez que el robot dé ese paso y vaya a ese punto en concreto, a la vez que se esté moviendo la pierna que está dando el paso, también lo haga el tobillo de la otra pierna (el tobillo del pie que está en contacto con el suelo), ya que es mediante el movimiento del tobillo del pie que se mantiene en el suelo, como se corrige la posición del robot.



Para comprender de una forma más sencilla el funcionamiento del control de equilibrio del robot, se describe a continuación un simple ejemplo:

*Cuando el robot está dando el paso izquierdo, al ir de un punto A a un punto B de la trayectoria de ese paso, al llegar al punto B se leen los valores del acelerómetro. En caso de que estos valores sean distintos de cero (por ejemplo  $1^\circ$  en X y  $1^\circ$  en Y), se corrigen y almacenan. Cuando el robot vuelva a dar el siguiente paso izquierdo y vaya del punto A al punto B, aparte de realizar los movimientos oportunos en los servos de la pierna izquierda, también se moverán los servos del tobillo derecho, concretamente  $-1^\circ$  en X y  $-1^\circ$  en Y, de esta forma, el robot compensa el desequilibrio producido al ir del punto A al punto B cuando realiza el paso izquierdo.*



Figura 7.11: Robot en el pto A



Figura 7.12: Robot en el pto B desequilibrado



Figura 7.13: Robot en el pto B corregido

En la secuencia de imágenes que se muestra previa a este párrafo (*figuras 7.11, 7.12 y 7.13*), se observa el proceso descrito en el ejemplo anterior. En la *figura 7.11*, el robot se encuentra con los dos pies en el suelo (punto A), situación previa al paso. Al iniciar el paso, el robot realiza una serie de movimientos para levantar verticalmente la pierna izquierda y de esta manera llegar al punto B del paso. Como se observa en la *figura 7.12*, al llegar al punto B el robot está desequilibrado ya que se ha producido una inclinación del mismo hacia la izquierda. El valor de inclinación del robot es medido por el acelerómetro. Dicho valor es almacenado y enviado por puerto serie a la *OpenCM*, para que los servos del tobillo derecho (pie que NO está realizando el paso) corrijan la posición del robot hasta dejar al mismo en una posición de equilibrio (cero grados en el eje X y cero grados en el eje Y). En la *figura 7.13* se aprecia al robot en el punto B, pero con su posición ya corregida y en equilibrio.

## 8 DESARROLLO DEL CÓDIGO

En este capítulo se describirá con un mayor grado de detalle el código empleado en este proyecto, además de nombrar y explicar los principales métodos de programación de la locomoción en robots.

### 8.1 MÉTODOS DE PROGRAMACIÓN DE LA LOCOMOCIÓN EN ROBOTS

#### 8.1.1 MÉTODOS ANALÍTICOS BASADOS EN EL *FEEDBACK* DE LOS SENSORES

Este método construye movimientos basándose en la realimentación (*feedback*) de los sensores que dispone el robot, y en las propiedades geométricas del mismo. Un ejemplo de este tipo de caminata, es el método anteriormente mencionado, *Zero Moment Point (ZMP)*, que utiliza la cinemática inversa [22] del robot para realizar movimientos de forma controlada, moviendo las diferentes articulaciones del robot según sea necesario.

#### 8.1.2 PROGRAMACIÓN POR GUIADO

Un método alternativo para realizar trayectorias o movimientos en robots es el conocido como “programación por guiado”, método comúnmente empleado por los robots de aprendizaje, y que se basa en la imitación de la caminata del ser humano. Este método es muy útil cuando trabajamos con robots pequeños, como en el caso de los mini-humanoides, ya que no es necesario en este tipo de robots llegar a un nivel de complejidad tan alto como el empleado en el primer método descrito anteriormente.

La programación por guiado consiste en que una persona coloca al robot en una posición determinada para que posteriormente el robot la memorice. Tras grabar varias posiciones, se consigue un movimiento deseado, como por ejemplo, dar un paso al andar, imitando de esta forma el andar de un humano.

El Bioloid y el Robonova son dos robots que utilizan este método.

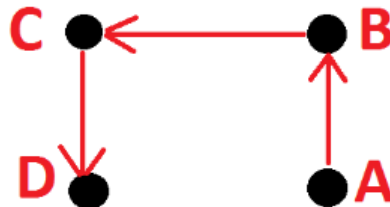
Una vez explicados los métodos principales en la programación de la locomoción en robots, ahora se procede a detallar el código empleado en este proyecto.

Tal y como se menciona en el capítulo 7.1, en este trabajo se emplean dos programas, *Robotis OpenCM* para el código utilizado por la *OpenCM 9.04*, y *Qt Creator* para el código de la *BeagleBone Black*. A continuación se explican cada uno de los códigos.

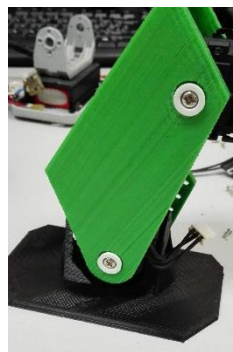
## 8.2 CÓDIGO DE LA OPENCM 9.04

El código empleado por la *OpenCM* sirve para realizar todos los movimientos de locomoción del robot, es decir, es el encargado de mover los servos necesarios para que el robot realice la caminata.

Para ello el robot debe de seguir una trayectoria diseñada previamente. Esta trayectoria está formada por una serie de puntos, cuyas coordenadas han sido calculadas con anterioridad, y por los cuales debe de ir pasando cada uno de los pies del robot cuando el robot da un paso. En la *figura 8.1* se puede observar cómo sería una trayectoria de paso formada por cuatro puntos vista de perfil.



*Figura 8.1: Trayectoria del pie en vuelo formada por cuatro puntos*



*Figura 8.2: TCP del robot*

Como se puede observar en las figuras anteriores, la trayectoria a seguir por el TCP del robot (*figura 8.2*) comenzaría en el punto A y terminaría en el punto D, tras haber pasado por el punto B y C, en ese orden. Por cada uno de los puntos señalizados en la *figura 8.1*, pasaría el pie del robot, más concretamente, por cada uno de esos puntos pasaría el

centro de la planta del pie, que sería exactamente el TCP del robot. De esta forma y de manera muy esquematizada, el robot consigue dar un paso y avanzar.

Las coordenadas de estos puntos están referenciadas con respecto al primer punto del paso (punto A), siendo las coordenadas del primer punto: (0,0,0,0,0,0). De esta forma, al restar las coordenadas del punto objetivo menos las del punto actual queda como resultado un incremento, siendo este incremento el avance del robot.

De esta manera, y a modo de ejemplo para que su comprensión sea más sencilla, se expone el siguiente ejemplo:

*El robot tiene que ir del punto A (0,0,0,0,0,0) al punto B (0,20,60,0,0,0). Al restar las coordenadas (B-A) queda un resultado de: (0,20,60,0,0,0), por lo que el robot tendría que realizar un movimiento en el eje X de 0, en el eje Y de 20, y en el eje Z de 60, manteniéndose con la misma orientación, ya que las diferencias en las tres últimas coordenadas, que son las de orientación, son cero.*

Este ejemplo se puede apreciar de forma ilustrada comparando las *figuras 7.11 y 7.13*, en las cuales el robot se desplaza desde el punto A hasta el punto B.

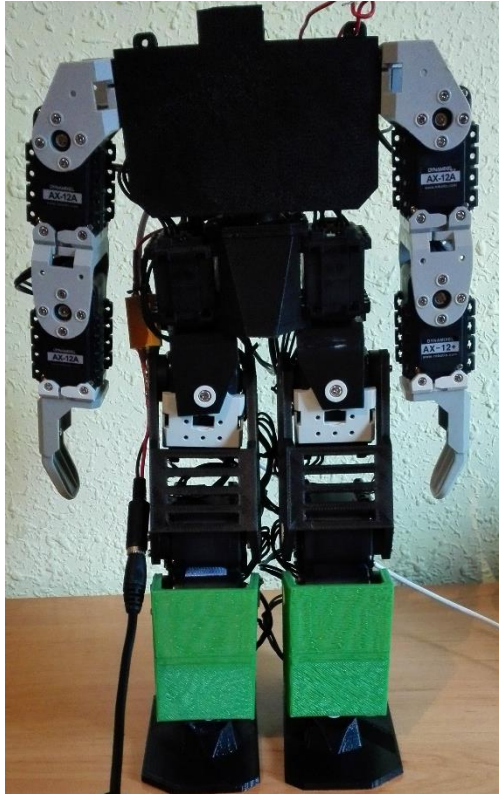
Estos incrementos se pasan posteriormente a la función de movimiento correspondiente:

- *MovFrontal*: para realizar los movimientos en el eje X.
- *MovLateral*: para realizar los movimientos en el eje Y.
- *MovVertical*: para realizar los movimientos en el eje Z.

### 8.2.1 FUNCIONES *MOV\_VERTICAL*, *MOV\_LATERAL* Y *MOV\_FRONTAL*

Estas funciones están basadas en las funciones originales del trabajo de Javier Isabel, pero han tenido que ser modificadas, puesto que al no tener el mismo número de servos, y al tener diferentes ID's (parámetro mediante el cual se identifica a cada uno de los servos) no eran compatibles con el robot de este proyecto. Se adjunta en el anexo "**ID's de los servos del robot**", el número ID que se ha adjudicado a cada uno de los servos del robot de este proyecto.

A continuación se puede observar los efectos que tienen cada una de estas tres funciones en las siguientes figuras, comparando la *figura 8.3* (posición de reposo) con el resto de figuras:



*Figura 8.3: Posición de reposo*



*Figura 8.4: Función MovVertical*



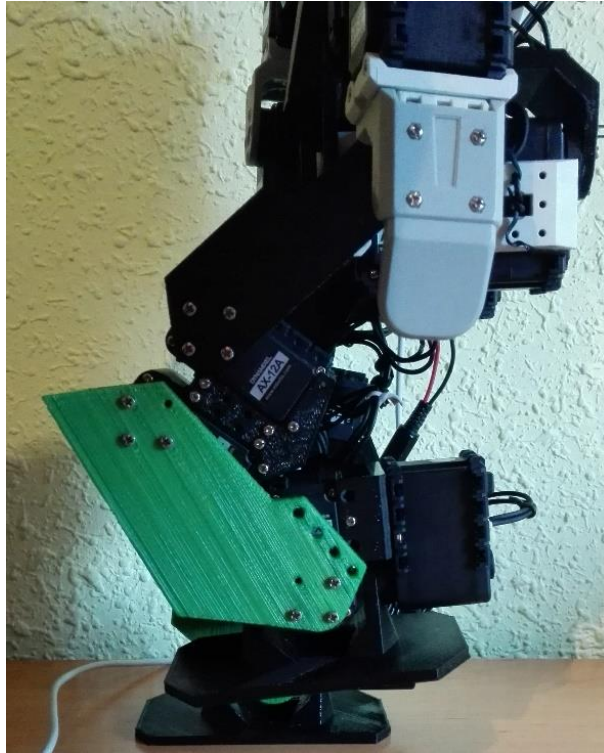


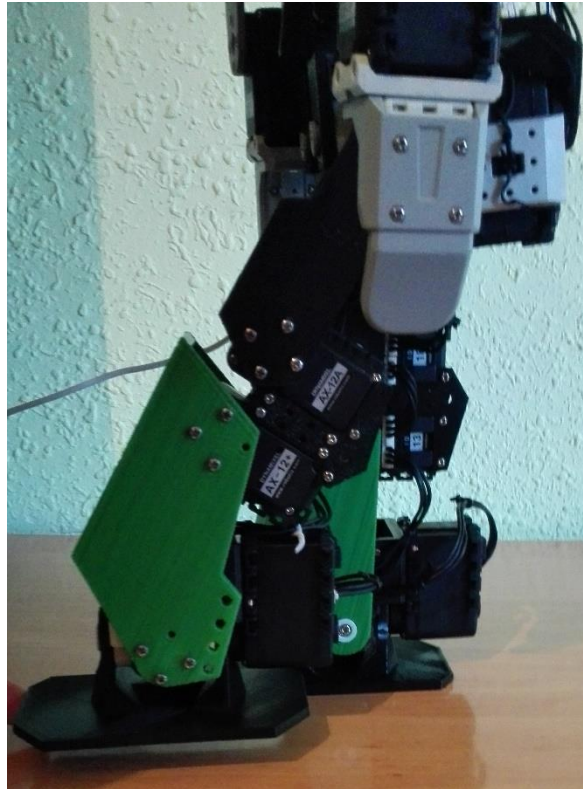
Figura 8.5: Función *MovVertical* (Perfil)

Como se puede observar, la función *MovVertical* mueve la pierna de forma vertical en el eje Z, mediante el movimiento simultáneo de cuatro servos: cadera delantera, los dos servos de la rodilla y el tobillo delantero.



Figura 8.6: Función *MovLateral*

En la *figura 8.6* se aprecia el movimiento en la pierna izquierda cuando se le aplica la función *MovLateral*. En este caso la pierna realiza un movimiento a lo largo del eje Y, mediante el movimiento de dos motores: el servo de la cintura trasera y el tobillo trasero.



*Figura 8.7: Función MovFrontal*

Por último, en la *figura 8.7* se observa el movimiento que realiza la pierna gracias a la función *MovFrontal*. Esta función provoca un movimiento lineal a lo largo del eje X mediante el movimiento de dos servos: cintura delantera y tobillo delantero.

### 8.2.2 FUNCIONES *STEP\_LEFT* Y *STEP\_RIGHT*

Para realizar la resta de coordenadas y pasar posteriormente esa diferencia a su respectiva función de movimiento (*MovVertical*, *MovLateral*, *MovFrontal*), se ha creado unas funciones llamadas *StepLeft* y *StepRight*, que como su nombre indica, son las encargadas de realizar el paso izquierdo y derecho, respectivamente. Estas funciones restan las coordenadas del punto objetivo menos las del punto en el que se encuentra el pie del robot, y posteriormente pasa el incremento en X a *MovFrontal*, el incremento en Y a *MovLateral*, y el incremento en Z a *MovVertical*.



Estas funciones serían las encargadas de realizar el paso como tal, pero como bien se ha indicado en el capítulo 7.2.1, el robot realiza una serie de movimientos previos al paso para posicionar la proyección de su CDG sobre el pie que se va a mantener en el suelo, es decir, su polígono de soporte. Estos movimientos previos constan de dos tipos de movimientos:

- 1- Movimiento de carga lateral del peso del robot (hacia la izquierda o derecha)
- 2- Movimiento de carga frontal del peso del robot (hacia adelante o atrás)

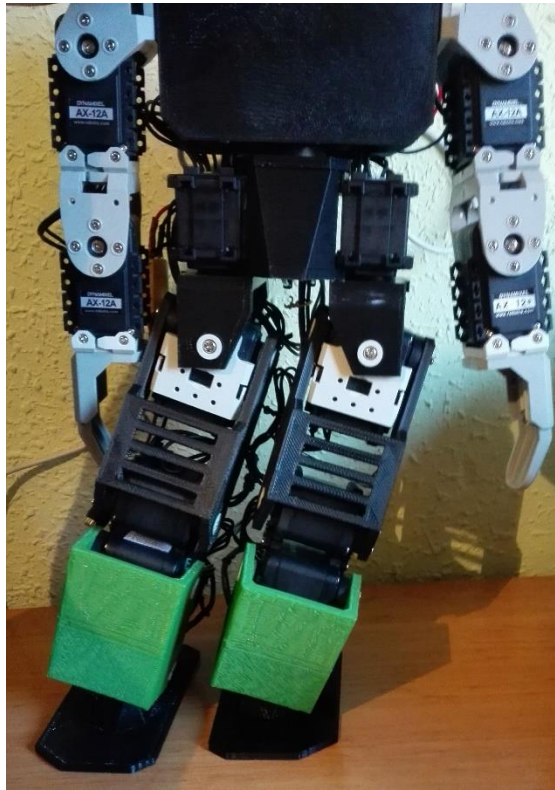
### 8.2.3 FUNCIONES *BEGIN\_STEP\_LEFT* Y *BEGIN\_STEP\_RIGHT*

Para realizar el primer movimiento se ha creado las funciones *Begin\_Step\_Left* y *Begin\_Step\_Right*. Estas funciones son las encargadas de desplazar el centro de gravedad sobre el pie derecho, en el caso de *Begin\_Step\_Left*, y sobre el pie izquierdo, en el caso de *Begin\_Step\_Right*, pero sin que los pies del robot se levanten todavía del suelo.



Figura 8.8: Función *Begin\_Step\_Left*

En la *figura 8.8* se puede ver como con la función *Begin\_Step\_Left* el centro de gravedad del robot se posiciona sobre la vertical del pie derecho, para que cuando el robot levante el pie izquierdo para dar el paso, mantenga mejor el equilibrio.



*Figura 8.9: Función Begin\_Step\_Right*

Ocurre lo mismo con la función *Begin\_Step\_Right* pero de forma simétrica. Su efecto se aprecia en la *figura 8.9*.

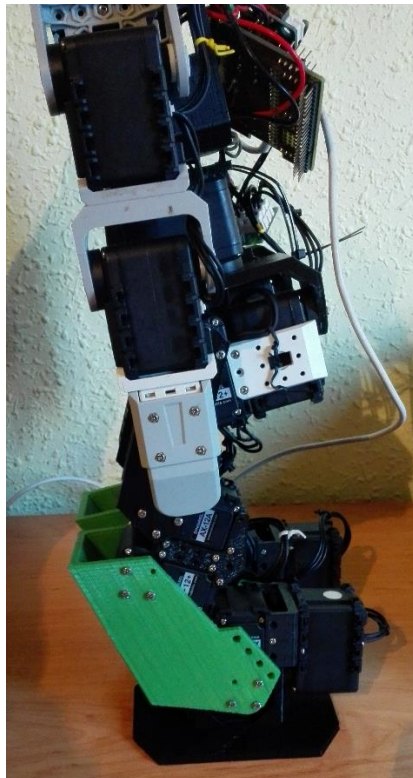
En ambas funciones, los servos que se utilizan son los de la cadera trasera, los del tobillo trasero y el de la cintura, siendo así un total de cinco servos moviéndose de forma simultánea.

#### 8.2.4 FUNCIONES *PESO\_ADELANTE* Y *PESO\_ATRÁS*

Por otro lado, para realizar el movimiento de carga frontal del robot, se han creado las funciones *Peso\_Adelante* y *Peso\_Atrás*. El cometido de estas funciones es simple, son las encargadas de inclinar al robot unos cinco grados hacia adelante y cinco grados hacia atrás respectivamente. Esta ligera inclinación del robot hacia adelante se debe a que cuando el robot levanta una de las piernas para dar el paso, la proyección del CDG quedaba ligeramente atrasada con respecto al polígono de soporte, por lo que el robot se desequilibraba hacia atrás.



*Figura 8.10: Robot en reposo*



*Figura 8.11: Función Peso\_Adelante*

Comparando la *figura 8.10* con la *figura 8.11*, se observa como en esta última el tronco del robot está inclinado unos cinco grados hacia adelante. Esto se debe a la función *Peso\_Adelante*.

Una vez que el robot ha dado el paso y tiene los dos pies sobre el suelo, se inclina hacia atrás mediante la función *Peso\_Atrás*, recuperando de esta forma su inclinación inicial (*figura 8.10*).

En estas funciones, los motores que entran en juego son únicamente dos, los del tobillo delantero, ya que son los encargados de manejar la inclinación del robot con respecto al eje Y.

### 8.2.5 FUNCIONES *REPOSO\_LEFT* Y *REPOSO\_RIGHT*

Por último, se han creado otras dos funciones, *Reposo\_Left* y *Reposo\_Right*. Estas funciones son las encargadas de que el robot recupere su posición de reposo (*figura 8.3*) tras haber dado un paso. Se podría decir que son la antítesis de las funciones *Begin\_Step\_Left* (*figura 8.8*) y *Begin\_Step\_Right* (*figura 8.9*), puesto que deshacen la inclinación que dichas funciones realizaban antes de dar el paso correspondiente. *Reposo\_Left* recupera la posición de reposo tras haber dado el paso izquierdo y *Reposo\_Right* tras haber dado el paso derecho.

### 8.2.6 CÓDIGO PRINCIPAL O *MAIN*

El código *main* es el código que está compuesto por las instrucciones que va a seguir el robot, realizando llamadas a las distintas funciones descritas anteriormente y empleando diferentes comandos de C++.

En primer lugar se establece una posición de reposo y se mueve al robot a dicha posición (*figura 8.3*). Una vez que el robot se encuentra en la posición de reposo, se inicia un *loop* (ciclo repetitivo) mediante el uso del comando *for*. Este ciclo se repite tantas veces como secuencias de pasos completos (un paso izquierdo y un paso derecho) se desee que realice el robot.

Ya dentro del *for*, se llama a la función *Begin\_Step\_Left* y a la función *Peso\_Adelante*, para que el robot desplace su centro de gravedad sobre el pie derecho. Acto seguido, se llama a la función *StepLeft* para que el robot de el paso izquierdo. Una vez que el pie izquierdo ha realizado el paso, se llama a la función *Peso\_Atrás* y a la función *Reposo\_Left* para que el robot vuelva a la posición de reposo inicial.

De manera simétrica se realiza el mismo proceso para dar el paso derecho, siguiendo el mismo orden que en el proceso descrito para dar el paso izquierdo, pero empleando las funciones *Begin\_Step\_Right*, *StepRight* y *Reposo\_Right*.

En la *figura 8.12* se puede apreciar un diagrama de flujo del funcionamiento del código de la *OpenCM 9.04*.

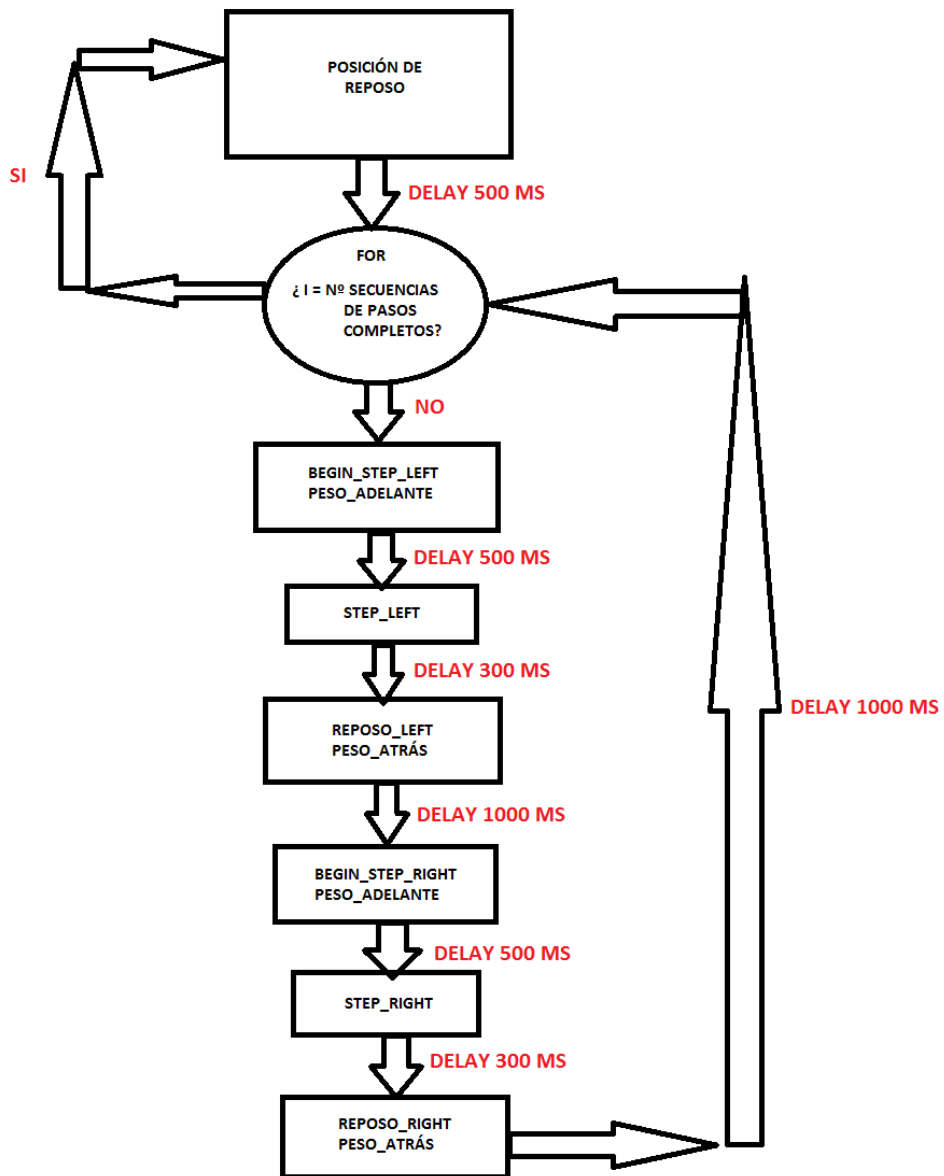


Figura 8.12: Diagrama de flujo del código de la OpenCM 9.04

## 8.3 CÓDIGO DE LA BEAGLEBONE BLACK

### 8.3.1 CONEXIÓN SSH PC-BEAGLEBONE BLACK

La *BeagleBone Black* es una placa computadora de *hardware*, es decir, es un ordenador, pero de tamaño reducido. Además, tiene la ventaja de que se pueden añadir periféricos como: pantalla, ratón, teclado y demás periféricos comunes en un PC.

A pesar de que la BBB permite la posibilidad de poder añadirla pantalla, ratón y teclado, es mucho más cómodo realizar una **conexión ssh** desde un ordenador con *Linux*, ya que de esta manera uno puede trabajar en la BBB desde su ordenador.

Como para programar el código de RAIDER en la BBB es necesario modificar y escribir código en el programa *QT Creator*, es imprescindible dotar a la BBB de la capacidad de poder ver lo que hay en ella y poder escribir en ella. Es por ello que la **conexión ssh** juega un factor principal.

Para realizar una **conexión ssh** es necesario que la dirección IP del dispositivo al cual se quiere conectar uno (en este caso la BBB) y la dirección del router al cual está conectado el ordenador desde el que se va a realizar dicha conexión, sean compatibles.

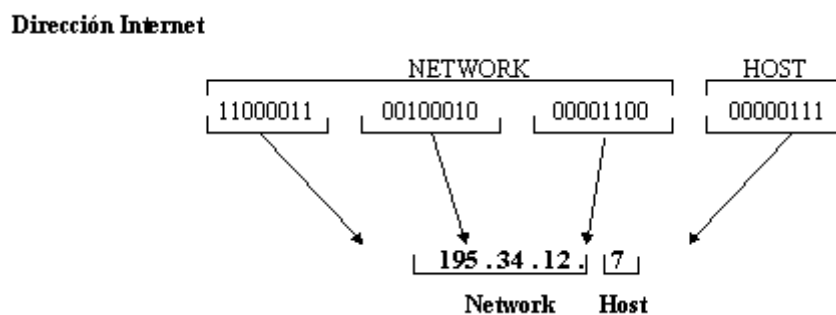


Figura 8.13: Dirección IP

Para que la dirección IP (*figura 8.13*) del router sea compatible con la de la BBB, se debe cumplir que la dirección IP de la BBB tenga los tres primeros números (Network) exactamente iguales a los de la IP del router (en este caso 192.168.1), y que el último número (Host) sea cualquier número de entre 0-255 exceptuando el número 1, puesto que este ya está siendo empleado por el router.

Por lo tanto, como la dirección IP que tenía la BBB de fábrica era 192.168.7.2, era necesario cambiar la dirección IP de la BBB a una que fuera compatible con la del router del lugar de trabajo en el que se ha realizado este proyecto. La dirección IP de este router era 192.168.1.1, por lo que se modificó la dirección IP de la BBB a 192.168.1.10.

De esta manera la BBB ya quedaba lista para que al abrir un terminal en un PC que estuviera conectado al mismo router al que estuviera conectada la BBB, y poner el comando “ssh -X usuario@192.168.1.10” (ver *figura 8.14*) en dicho terminal, se produjera la **conexión ssh** de forma satisfactoria.

```
daniel@daniel-HP-Notebook:~$ ssh -X debian@192.168.1.10
Debian GNU/Linux 7

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:tempwd]

debian@192.168.1.10's password:
Last login: Thu Sep 29 21:49:10 2016 from daniel-hp-notebook.local
debian@beaglebone:~$
```

Figura 8.14: Comando ssh -X

### 8.3.2 PROGRAMACIÓN DEL CÓDIGO

Una vez que se tiene la BBB preparada con todas las librerías instaladas, con el código de RAIDER descargado, y con la IP cambiada para así poder conectarse a ella mediante ssh, ya está lista para poder trabajar con el código para conseguir que el robot realice una estrategia de caminata estable.

Como ya se mencionó en el capítulo 7.2.1, gracias a la *OpenCM 9.04* y a las funciones diseñadas para dicha placa (explicadas en el capítulo 8.1), el robot realiza una serie de movimientos previos al paso que provocan el desplazamiento de su CDG sobre el polígono de soporte. A pesar de que con estos movimientos se logra que el robot mantenga el equilibrio en gran parte, no es suficiente para que logre realizar una caminata de forma estable en todo momento.

Para conseguir esto, hay que lograr que la proyección del CDG se encuentre dentro del polígono de soporte en todo momento, tal y como se describió en el capítulo 7.2.2.

Es por ello que en este proyecto juega un papel principal el sensor MPU 9250 (*figura 8.15*), ya que gracias a las lecturas proporcionadas por él, se consigue corregir la posición del robot.



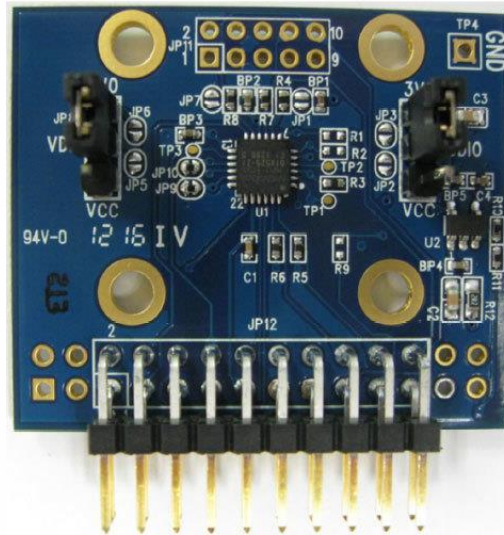


Figura 8.15: Sensor MPU 9250

En el capítulo 7.1 se menciona que para trabajar con el código empleado por la BBB se emplea el programa *QT Creator*. Gracias a este programa, y basándose en el código original de RAIDER, en este proyecto se propone una aplicación en la cual el robot es capaz de realizar una caminata de forma estable.

El funcionamiento de esta aplicación es el siguiente:

El robot comienza a dar un paso gracias a los movimientos pre-diseñados en la *OpenCM*. Cuando el TCP se aproxima a llegar al siguiente punto (se recuerda que el paso del robot estaba formado por una serie de puntos parametrizados previamente, y por los cuales iba pasando el TCP (pie) del robot hasta completar el paso), la *OpenCM* manda una señal por puerto serie a la BBB. Cuando la BBB recibe dicha señal, realiza una lectura de los valores proporcionados por la IMU, leyendo los siguientes registros del sensor:

- 0x3B y 0x3C, para saber la inclinación del sensor en el eje X
- 0x3D y 0x3E, para saber la inclinación del sensor en el eje Y
- 0x3F y 0x3G, para saber la inclinación del sensor en el eje Z (aunque los valores proporcionados por la inclinación en el eje Z se desprecian ya que son mínimos)

Una vez leídos estos valores, se realiza una conversión a grados, ya que el valor de estos registros está en G's. Estos valores se pasan por puerto serie a la *OpenCM* para que esta los almacene en una serie de variables. De esta forma, mientras realiza el paso, aparte de realizar los movimientos correspondientes para dar dicho paso, los servos del tobillo del pie que se mantiene en el suelo durante el paso, girarán un número de grados igual al almacenado en dichas variables pero con signo opuesto, para así contrarrestar el desequilibrio que se produce al ir de un punto en concreto al siguiente de la trayectoria del paso.

A modo de que el entendimiento del sistema de control de equilibrio sea más sencillo, se dispone de un ejemplo al final del capítulo 7.2.2 explicando este proceso.



De esta forma se consigue ir modificando la trayectoria de paso del robot sobre la marcha, algo totalmente necesario si se quiere lograr una estrategia de caminata estable, puesto que el robot no camina siempre sobre la misma superficie, ni sufre las mismas perturbaciones en todo momento.

## 9 CONTRATIEMPOS

A lo largo del desarrollo de este proyecto han surgido numerosos imprevistos y complicaciones a los que se ha tenido que hacer frente, y por los cuales el tiempo de ejecución ha sido más del deseado. A continuación se exponen los contratiempos más importantes que han surgido durante la ejecución de este trabajo, y las soluciones propuestas.

### 9.1 PIEZAS REALIZADAS CON IMPRESORA 3D

Tal y como se ha contado en el capítulo 6.1, el robot de este proyecto está formado por piezas impresas en 3D. Las impresoras con las cuales se han imprimido dichas piezas están situadas en la Universidad Carlos III de Madrid. Debido a que dichas impresoras están a disposición de aquellos alumnos, profesores e investigadores que lo requieran, no era sencillo disponer de ellas habitualmente. A esto hay que sumarle que debido a su mal uso, habitualmente una o las dos impresoras no estaban operativas, por lo que la impresión total de las piezas del robot llevo algunos meses.

Otro contratiempo relacionado con las piezas del robot, es que a pesar de que el material con el que están impresas las piezas es bastante resistente, en ocasiones ante determinados esfuerzos exigidos por el mini-humanoide, han llegado a ceder y fracturarse, tal y como se observa en la *figura 9.1*, en la que se puede observar a la pieza *upper\_leg* fracturada por uno de sus extremos.



*Figura 9.1: Upper\_Leg fracturada*

## 9.2 MONTAJE DEL ROBOT

Puesto que no existe ningún manual ni instrucciones sobre cómo realizar el montaje de RAIDER, se ha empleado una gran cantidad de tiempo buscando la manera en la que iban cada una de las piezas y cada uno de los servos empleados en el mini-humanoide. Además, también se tuvo que pensar en la manera en la que iban conexiados en serie, mediante cables TTL, cada uno de los servos, ya que tampoco viene explicado en ninguna parte como hacerlo.

Por este motivo, en este proyecto aparte de realizar la fabricación del robot y su posterior diseño de la estrategia de caminata, se ha creado también un **MANUAL DE INSTRUCCIONES de montaje de RAIDER**, para que así en el futuro su construcción sea rápida y sencilla. Este manual se puede observar en el Anexo “Manual de montaje del robot RAIDER”.

## 9.3 MODIFICACIONES EN EL DISEÑO ORIGINAL DE RAIDER

Durante el desarrollo del proyecto se observaron pequeños inconvenientes en el diseño del robot, a continuación se exponen:

- 1- Los cables TTL que van de los servos que hay en el interior del cuerpo (servos de los hombros internos) a los servos de la cintura, se cizallaban con la parte inferior de la pieza *body* al realizar un giro en la cintura. Es por eso que se diseñó una pieza a modo de alza para elevar unos cuantos milímetros la pieza *body* con respecto de las piezas que forman la cintura, para lograr de esta forma eliminar el efecto cizalla sobre estos cables. Esta pieza se puede observar en la *figura 12.19*, y el efecto que provoca su utilización en el robot se aprecia comparando las *figuras 12.20* y *12.21*
- 2- Se observó que durante la realización de la caminata, el robot sufría deslizamiento al apoyar el pie contra el suelo. Dicho deslizamiento se producía cuando la superficie sobre la que caminaba el robot tenía poco coeficiente de rozamiento,  $\mu$ , con respecto al plástico con el cual están hechas las piezas del pie del robot. Por ello, se pensó en poner en la parte inferior de cada uno de los pies, una especie de “plantilla” compuesta de un material con un alto coeficiente de rozamiento, para así evitar que el robot sufra deslizamientos al caminar.
- 3- En un principio la batería que alimenta al robot iba a ir colocada en la parte trasera del mismo, en la pieza *battery support*. Cuando en la práctica se intentó poner dicha pieza en su sitio correspondiente, se observó que el lugar en el cual iba colocada, era incompatible con el acto de caminata del robot, puesto que al realizar determinados giros, había piezas y cables del robot que

se enganchaban o colisionaban con el *battery support*. Es por esto que se decidió eliminar dicha pieza, y fijar la batería en el mismo lugar en el cual iba a ir originalmente, pero en vez de ir sustentada por la pieza *battery support*, sustentada por unas bridas. Con esta solución, aparte de eliminar el problema de que algunas piezas y cables se enganchen y choquen, también se logra eliminar peso en el robot, algo muy beneficioso para él, ya que cuanto más optimizado esté el peso total del mini-humanoide, más eficiente será el empleo de los servos y menor el consumo de batería.

#### 9.4 PROBLEMAS CON LA BEAGLEBONE BLACK

La *BeagleBone Black* ha sido uno de los focos principales de problemas y retrasos en este proyecto debido a varias causas que a continuación se exponen.

En un principio se iba a trabajar con la *BeagleBone* tal cual venía de fábrica, pero a la hora de intentar conectarse a la BBB mediante el comando *ssh* de Linux, surgían problemas, ya que la imagen de *Debian* que venía por defecto en la BBB no permitía conectarse (todo lo referente al cambio de IP de la BBB, conectarse mediante *ssh*, etc, viene explicado en el capítulo 8.3.1).

Una vez instalada la nueva imagen de *Debian* en la tarjeta SD (proceso descrito en el capítulo 6.4), se disponía de una BBB que en principio funcionaba sin ningún problema y sobre la cual se había realizado el cambio de IP de la misma (proceso descrito en el capítulo 8.3.1), por lo que estaba ya lista para pasar a la programación del código de RAIDER, hasta que un día sin razón aparente alguna dejó de funcionar. Se intentó alimentar de varias formas para comprobar que el problema no era la forma de alimentación... se alimentó vía USB, mediante batería e incluso mediante una fuente de alimentación de 5V cuyo conector hubo que fabricar y soldar de forma casera.

Una vez que se comprobó que el fallo era a nivel interno de la placa, y sobre la cual no se podía hacer nada más, se decidió pedir una nueva, con la mala suerte de que vino mal de fábrica y directamente ni se encendía, por lo que hubo que pedir una tercera BBB.

Fue con esta tercera placa con la que finalmente se ha podido realizar el proyecto, ya que esta placa no ha presentado ningún inconveniente a nivel de alimentación. Sin embargo, la BBB ha presentado un problema crítico a la hora de poder realizar el proyecto tal y como se tenía pensado en un principio.

Una vez que la BBB estaba ya preparada con todas la librerías instaladas, cambiada su IP a la correspondiente para poder realizar la conexión *ssh*, habiendo descargado el código de RAIDER, y habiendo solucionado todos los errores y *warnings* que salían al compilar dicho código... esta no dejaba ejecutar la app de caminata que se había creado para el robot puesto que aparecía un error. Dicho error estaba relacionado con la librería *fuzzylite*, y debido al cual el código no funcionaba. En un primer momento se pensó en eliminar esta librería, pero pronto se desestimó esta opción puesto que *fuzzylite* es una

librería indispensable para el funcionamiento del código. Tras semanas intentando buscar una solución a este problema, de haber buscado información en multitud de páginas webs, y ante la falta de tiempo latente, mi tutor y yo decidimos dejar lo poco que queda del trabajo como “Proyecto futuro”, para que este trabajo sea terminando por otro alumno en el futuro.

A pesar de esto, hay que decir que el proyecto está prácticamente acabado puesto que el robot ha sido fabricado y construido desde cero. Se ha realizado toda la electrónica necesaria para su funcionamiento correcto. Se ha diseñado y programado el código de la *OpenCM* necesario para que el robot camine, cosa que el robot hace. Queda únicamente como pendiente, el que se realice la continua sensorización del estado de equilibrio del mini-humanoide para que de esta forma realice la caminata completamente estable, algo que no se ha podido realizar, debido a los problemas que la BBB ha ocasionado en este proyecto, y por la escasez de tiempo del que se disponía.

## 9.5 BATERÍA LIPO

La batería empleada en este proyecto es una batería LIPO 3S de 11.1V y 2200mah (ver *figura 9.2*).



*Figura 9.2: Batería LIPO 3S 11.1V 2200mah*

Esta batería ha presentado fallos a la hora de cargarla, o mejor dicho, el cargador que venía con ella que se puede observar en las *figuras 9.3* y *9.4*.



Figura 9.3: Cargador Prophet Sport plus 50 W AC/DC



Figura 9.4: Cargador Prophet Sport plus 50 W AC/DC

El problema que producía este cargador era que solo cargaba de forma correcta las primeras veces. Cuando ya se habían realizado varias cargas de la batería, el cargador dejaba de funcionar. En un principio se pensó que el problema era debido a la batería, por lo que se pidió otra batería exactamente igual, por si acaso el problema se debía a que alguna de las tres celdas que forman la batería (3S) estuviera quemada. Pero el problema prosiguió al llegar la nueva batería, por lo que se dedujo que la causa del problema era el cargador y no la batería en sí.

Como solución se decidió alimentar al robot en vez de por batería, mediante un cargador de 12V conectado a la red. A pesar de que alimentando de esta forma se tiene como inconveniente el que se capa al robot en libertad de movimientos, tiene como ventaja el que no haya que estar cargando continuamente la batería, ya que cuando se alimentaba al robot mediante batería, se observó que a pleno rendimiento, el robot tenía un tiempo de autonomía aproximado de 10 minutos. Además, se obtiene otra ventaja, y es que al eliminar la batería se consigue disminuir el peso total del robot, mejorando las prestaciones del mismo.

## 9.6 ELECCIÓN DE LA ESTRATEGIA DE CAMINATA

La estrategia de caminata a seguir por el robot, no fue definida desde un principio tal y como ha sido finalmente.

Dicha estrategia se ha ido modificando durante el transcurso del proyecto, debido a las observaciones que se hacían al ver los fallos en la manera de caminar del robot.

En un primer lugar, se pensó realizar la estrategia de caminata de forma que el robot realizara los pasos sin hacer los movimientos previos que se relatan en el capítulo 7.2.1.

Se observó que de esta manera, el robot sufría grandes oscilaciones durante el paso, puesto que la proyección del CDG no quedaba dentro del polígono de soporte. Por este motivo, se decidió modificar la estrategia de caminata añadiéndole los movimientos previos al paso, mencionados anteriormente. Una vez modificado esto, se pudo apreciar una notable mejoría en el equilibrio del robot durante la caminata, algo que sumado al posterior control de equilibrio por sensorización (capítulo 7.2.2), hace que la estrategia de caminata del robot de este proyecto sea estable y satisfactoria.

## 9.7 RETRASOS TEMPORALES POR DIFERENTES CAUSAS

Durante la realización de este proyecto se han sufrido múltiples retrasos temporales debidos a diversas causas.

La impresión de las piezas del robot llevó un gran tiempo debido a que como se mencionó en el capítulo 9.1, las impresoras 3D de la universidad estaban habitualmente fuera de servicio. Además, el montaje del robot llevó más de lo esperado puesto que al no existir ningún manual de montaje del mismo, no había manera alguna de saber cómo iba cada una de las piezas.

El fallo de componentes ha sido otro de los motivos por los que este proyecto ha durado más tiempo de lo esperado. Como se ha mencionado líneas arriba, hubo que pedir varias placas BBB y varias baterías ya que estas fallaban, con el tiempo de espera que esto conlleva hasta que llegan los componentes deseados a la universidad.

Pero el motivo fundamental por el cual este proyecto ha sufrido un retraso temporal notable, es que empezó siendo un proyecto distinto y con un tutor diferente.

En un principio este proyecto estaba tutorizado por Félix Rodríguez Cañadillas y el objetivo de dicho proyecto era el "Desarrollo e implementación de un sistema dinámico de locomoción para un robot mini-humanoide". Este trabajo también se basaba en el robot mini-humanoide RAIDER fabricado en este proyecto, pero su objetivo final no era el mismo.

El cambio en el objetivo final del proyecto se debe a que en Junio de 2016 la universidad no renovó el contrato a Félix, por lo que tuvo que abandonar los proyectos que tenía a su cargo. Como consecuencia, se tuvo que buscar un nuevo tutor (Jorge Muñoz Yañez-Barnuevo), y ponerle al día de la temática del proyecto y buscar un nuevo enfoque que darle.

Esto supuso un *delay* de varios meses, pero finalmente se ha conseguido realizar un proyecto sólido de forma satisfactoria.



## 10 PRUEBAS Y RESULTADOS

En este capítulo se muestran las diferentes pruebas realizadas sobre el robot, y los resultados obtenidos de dichas pruebas. Para ello, a continuación se detallan diferentes parámetros de la caminata del robot para varios estilos de caminata.

Los parámetros a medir son:

- Tiempo que el robot se mantiene en equilibrio al caminar (TE).
- Distancia recorrida por el robot antes de perder el equilibrio (DR).
- Número de pasos izquierdos dados (NPI).
- Número de pasos derechos dados (NPD).

### 10.1 CAMINATA SIN REALIZAR MOVIMIENTOS PREVIOS AL PASO

En este apartado se muestran las pruebas realizadas cuando el robot camina sin realizar los movimientos previos al paso descritos en el capítulo 7.2.1.

	<b>TE(seg)</b>	<b>DR(cm)</b>	<b>NPI</b>	<b>NPD</b>
<b>PRUEBA 1</b>	0.44	0	0	0
<b>PRUEBA 2</b>	0.49	0	0	0
<b>PRUEBA 3</b>	0.42	0	0	0
<b>PRUEBA 4</b>	0.51	0	0	0
<b>PRUEBA 5</b>	0.48	0	0	0

*Tabla VI: Pruebas sin movimientos previos al paso*

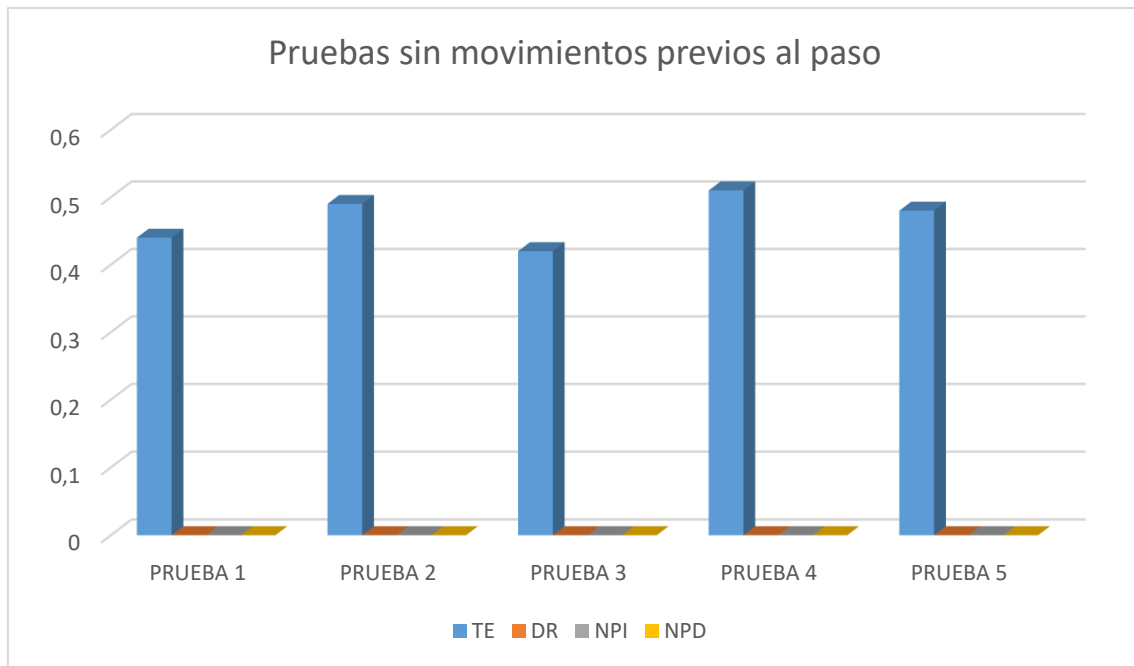


Tabla VII: Gráfica pruebas sin movimientos previos al paso

Como se puede observar en las *Tablas VI y VII*, no realizar los movimientos previos al paso resulta determinante para que la caminata no se realice de forma satisfactoria, puesto que tal y como se observa, el robot no es capaz de dar ni siquiera un paso, y por lo tanto, no recorre ninguna distancia.

## 10.2 CAMINTA REALIZANDO MOVIMIENTOS PREVIOS AL PASO

A continuación se muestran las pruebas y resultados de la caminata del robot realizando los movimientos previos al paso.

	TE(seg)	DR(cm)	NPI	NPD
<b>PRUEBA 1</b>	11.23	15.8	2	2
<b>PRUEBA 2</b>	14.14	19.3	3	2
<b>PRUEBA 3</b>	8.57	12.1	2	1
<b>PRUEBA 4</b>	22.26	31.0	4	4
<b>PRUEBA 5</b>	14.32	19.4	3	2

Tabla VIII: Pruebas con movimientos previos al paso

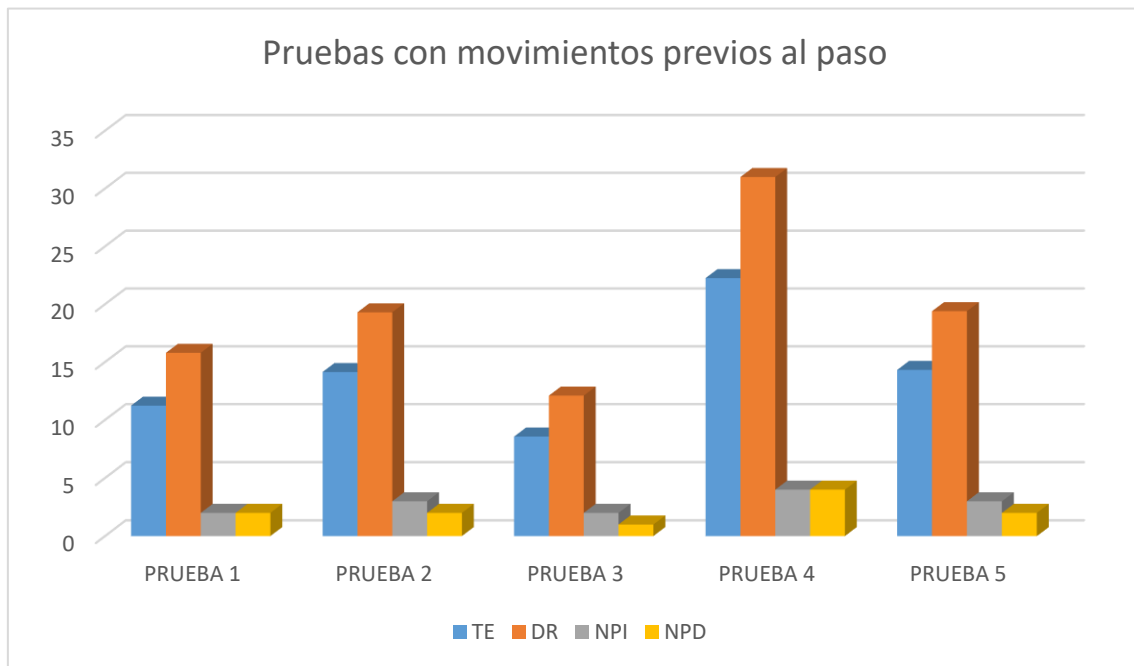


Tabla IX: Gráfica pruebas con movimientos previos al paso

En las *Tablas VIII y IX*, se observa el efecto que tienen los movimientos previos al paso en la caminata del robot. Gracias a dichos movimientos, el robot es capaz de caminar manteniendo el equilibrio durante un tiempo considerable (TE). El robot llegó a dar 8 pasos completos sin caerse en la prueba número 4, llegando a recorrer una distancia de 31 cm en algo más de 22 segundos, resultado bastante exitoso y satisfactorio.

En resumen, dados los resultados obtenidos en las diferentes tablas mostradas con anterioridad, se confirma en la práctica los conocimientos teóricos explicados en el capítulo 7.2.1, ya que es evidente vistos los resultados obtenidos, que al realizar los movimientos previos al paso descritos en dicho capítulo, al mover el CDG sobre el polígono de soporte, se consigue una estabilidad bastante fiable, hecho que no ocurre si dichos movimientos no se realizan.

## 11 CONCLUSIONES

Durante el desarrollo de este proyecto se han sufrido una serie de hechos que han provocado que debido a la falta de tiempo, no se haya podido completar al 100% los objetivos iniciales del mismo. Tal y como se menciona en el capítulo 2, los objetivos de este proyecto eran:

- **Construcción de un robot mini-humanoide**
- **Puesta en marcha del mismo**
- **Desarrollo de una aplicación de caminata**

El primer objetivo se ha completado al 100%, ya que se ha construido un robot mini-humanoide de forma completa, desde el diseño e impresión de las piezas, hasta los posteriores montajes mecánicos y electrónicos.

En lo referente al segundo objetivo, decir que también se ha completado al 100%, puesto que se han programado las diferentes placas empleadas en este proyecto, y se ha dotado al robot de capacidad de movimiento.

En cuanto al último objetivo, hay que mencionar que no se ha conseguido totalmente la consecución del mismo. Si bien es cierto, que se ha logrado que el robot camine y mantenga el equilibrio durante un tiempo y distancia considerables (capítulo 10.2), no se ha conseguido captar la lectura de datos de la IMU, para que de esta forma, el robot mantenga el equilibrio en todo momento.

Por esta razón, queda para *trabajos futuros*, conseguir que se realice la lectura de datos de la IMU y su posterior acondicionamiento, para que el robot corrija su posición de forma que mantenga su estabilidad en todo momento, ya que esto no se ha conseguido durante este proyecto debido a los continuos problemas ocasionados por la *BeagleBone Black* y la IMU, algo que sumado a la falta de tiempo disponible por otras causas, han hecho que no haya sido posible.

Adicionalmente, paralelo al desarrollo y consecución de los objetivos de este trabajo, se ha creado un manual de montaje del robot mini-humanoide RAIDER (Anexo 12.1), puesto que no existía hasta el momento ningún manual o documento que explicará paso a paso como construir dicho robot.

En resumen, señalar que el robot realiza en un muy alto porcentaje los objetivos marcados para este proyecto, objetivos que dicho queda, van más allá de un trabajo de fin de grado común, puesto que el desarrollo y fabricación de un robot mini-humanoide, y el diseño de su posterior estrategia de caminata, y que dicha estrategia sea estable y el robot camine, no es algo para nada sencillo.

## 12 ANEXOS

### 12.1 MANUAL DE MONTAJE DEL ROBOT RAIDER

#### 12.1.1 MANUAL DE MONTAJE FÍSICO DEL ROBOT

En este anexo se va a describir el orden cronológico para llevar a cabo la construcción de un robot mini-humanoide RAIDER, es decir, este anexo es un **Manual de Instrucciones** del robot mini-humanoide RAIDER.

En primer lugar, se construyen los pies uniendo las piezas *foot* (figura 12.2) y *lower leg* (figura 12.1) mediante un motor que va unido al *foot* gracias a la pieza de la figura 6.4. Toda esta unión se observa en la figura 6.6.

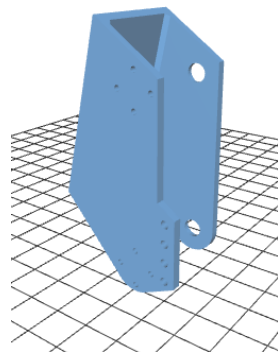


Figura 12.1: Lower Leg

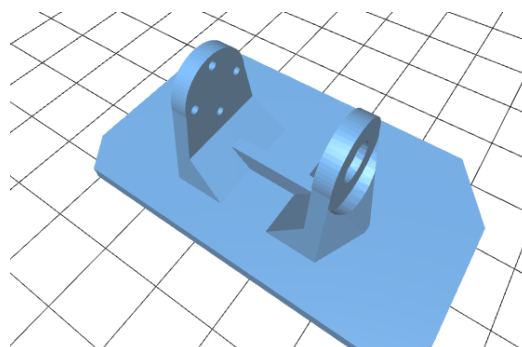


Figura 12.2: Foot

Posteriormente se añade otro motor por la parte trasera del pie y se une a él gracias a la pieza *ankle* (figura 12.3). De esta forma se dota al pie de dos grados de libertad al tener dos motores. El pie completo se puede observar en la figura 12.4.

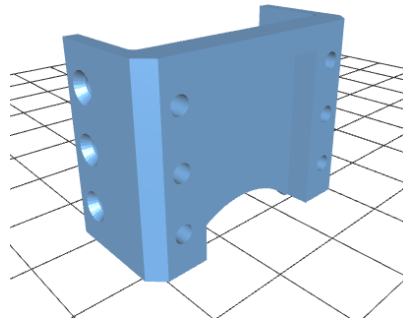


Figura 12.3: Ankle

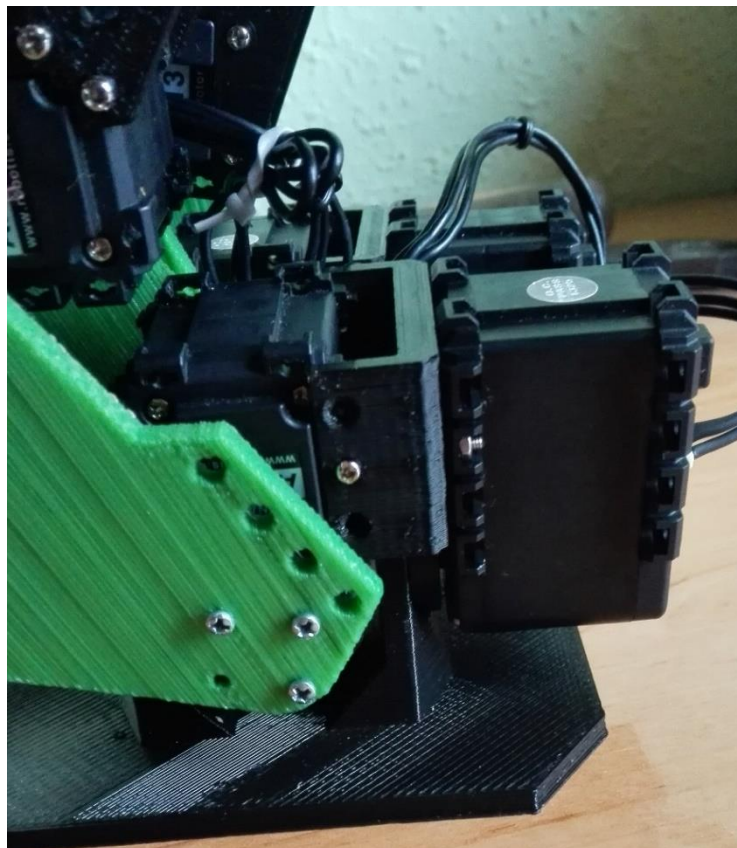


Figura 12.4: Pie completo

Seguidamente, se pasa a la construcción de la rodilla. Para ello es necesario unir dos motores en paralelo mediante la pieza *knee* (figura 12.5). Esta unión se puede apreciar en la figura 12.6.

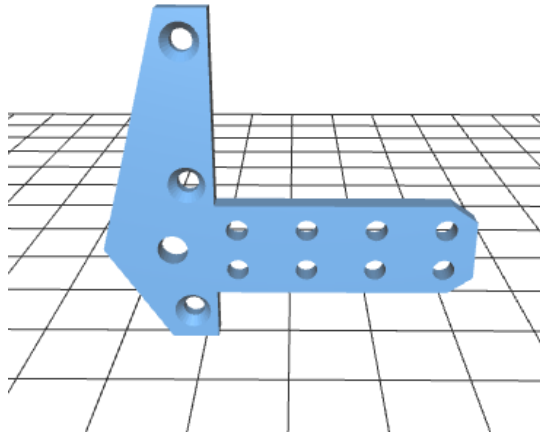


Figura 12.5: Knee



Figura 12.6: Rodilla

Por otro lado, se unen otros dos motores, uno en horizontal y otro en vertical, utilizando las piezas de las *figuras 6.3* y *6.4*, para obtener un resultado final que se puede apreciar en la *figura 12.7*



Figura 12.7: Motores unidos

Estos dos motores se unen a la pieza *upper leg* (figura 12.8), quedando el montaje de la figura 6.5

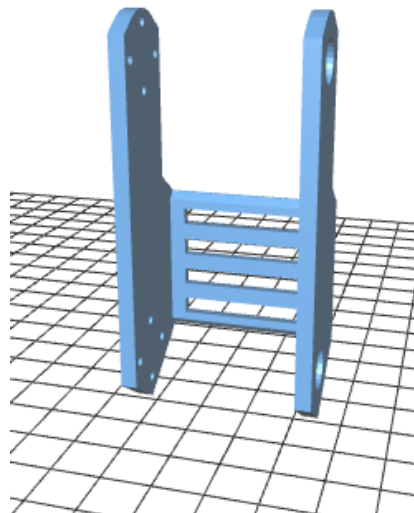


Figura 12.8: Upper Leg

Gracias a los dos motores unidos de la figura 12.6, se une la pieza *upper leg* (figura 12.8) con la parte inferior de la pierna montada anteriormente. De esta manera se obtiene la pierna completa del robot, figuras 12.9 y 12.10.





*Figura 12.9: Pierna completa (vista lateral)*



*Figura 12.10: Pierna completa (vista frontal)*

Una vez realizado el mismo proceso para la otra pierna, el siguiente paso es montar la cadera del robot.

Para ello se unen las piezas *leg bracket* (figura 12.11) con la pieza *hip chasis* (figura 12.12) mediante la unión de dos motores, uno encima de cada *leg bracket*. Estos motores se unen tanto al *hip chasis* como al *leg bracket* mediante tornillos.

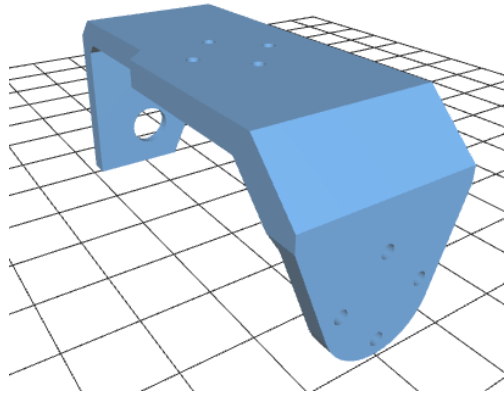


Figura 12.11: Leg Bracket

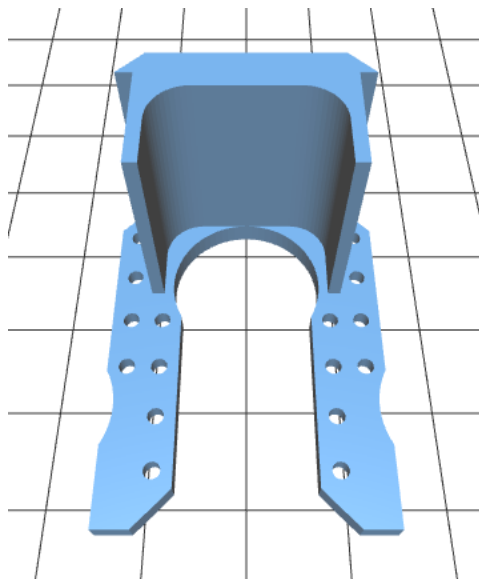


Figura 12.12: Hip Chasis

Para finalizar el montaje de la cadera, se añade la pieza *hip top* (figura 12.13), y se une también a los dos motores anteriores.

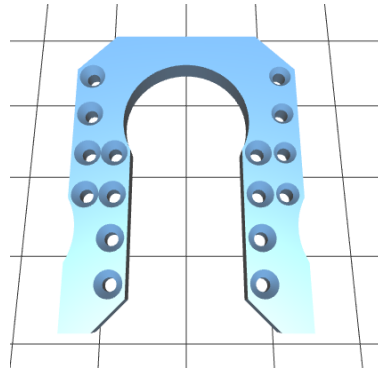


Figura 12.13: Hip Top

De esta forma obtenemos el montaje de la *figuras 12.14 y 12.15*



Figura 12.14: Cadera



Figura 12.15: Cadera

A continuación se procede a unir la cadera con las dos piernas. Este paso es muy sencillo ya que únicamente hay que unir mediante tornillos cada uno de los *leg bracket* de la cadera con su respectiva pierna, obteniendo como resultado todo el tren inferior del robot incluyendo hasta la cadera, *figuras 12.16 y 12.17*



*Figura 12.16: Tren inferior del robot*



Figura 12.17: Tren inferior del robot

El siguiente paso es unir el cuerpo del robot con el montaje obtenido hasta el momento. Para esto, se introduce un motor en el hueco que queda entre el *hip chasis* y el *hip top* y se une a dichas piezas mediante tornillos. Seguidamente se une la pieza *body* (figura 12.18) a este motor. A continuación, se coloca la pieza *alza* (figura 12.19) entre la pieza *body* y el motor situado en el hueco que queda entre las piezas *hip top* y *hip chasis*.



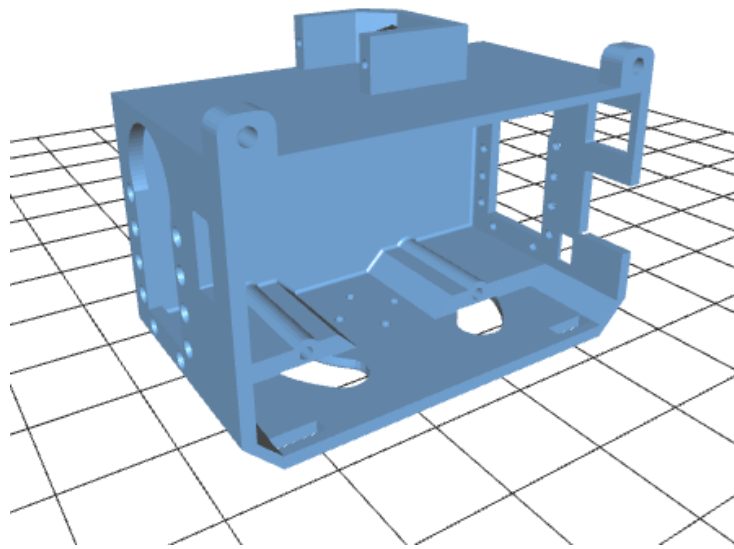


Figura 12.18: Body



Figura 12.19: Pieza alza

La diferencia entre que el robot lleve el *alza* o no se puede apreciar en las *figuras 12.20* y *12.21*



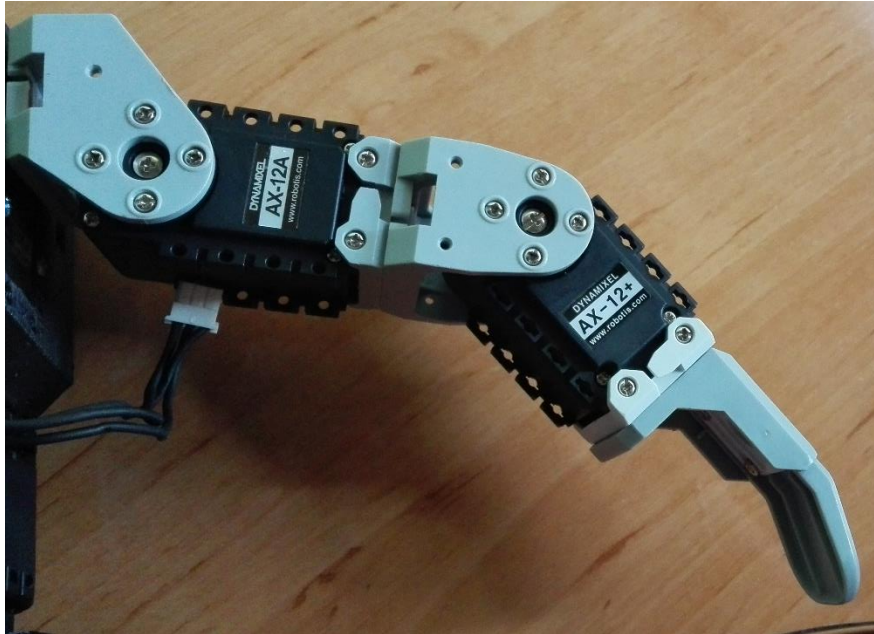
*Figura 12.20: Unión de la pieza Body con el servo de la cintura SIN alza*



*Figura 12.21: Unión de la pieza Body con el servo de la cintura CON alza*

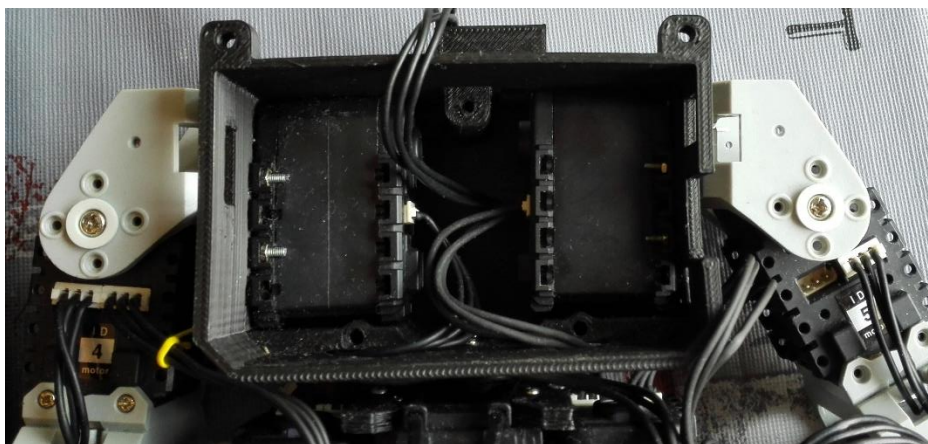
Se puede observar en las imágenes de arriba como al colocarle el alza, se crea un espacio entre el *body* y la cintura que permite que los cables TTL de la zona no se cizallen (este problema viene descrito en el capítulo 9.3).

Acto seguido, se añaden los brazos del robot. Estos brazos se toman tal cual del robot *Bioid* que tomamos como modelo (*figura 6.8*). Uno de estos brazos se puede observar en la *figura 12.22*



*Figura 12.22: Brazo Bioid*

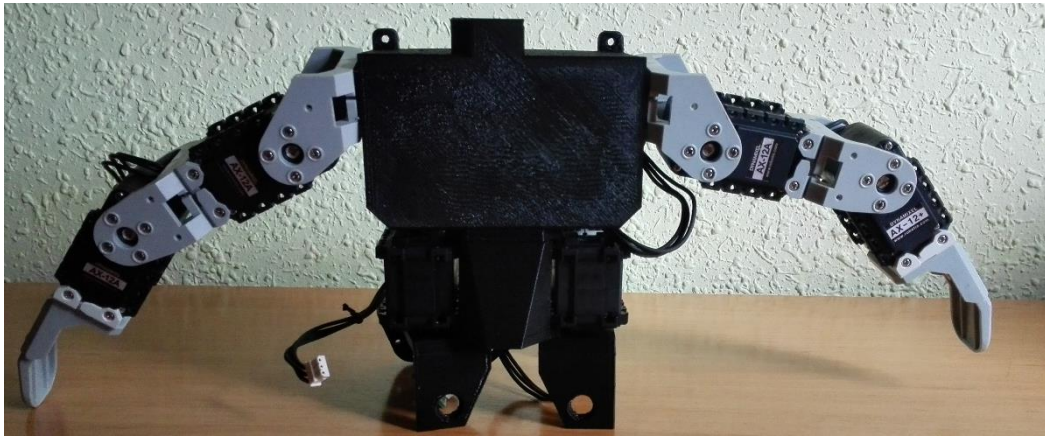
Cada brazo se une al cuerpo del robot mediante dos motores situados en el interior de la pieza *body*, ya que dentro de la misma hay una serie de agujeros cuyo fin es el de fijar dos servos para la posterior unión de los brazos del robot. La colocación de estos dos motores en el interior de la pieza *body* se puede observar en la *figura 12.23*



*Figura 12.23: Servos del hombro interno en la pieza body*



De esta forma ya queda montado el cuerpo con sus brazos como se puede observar en la *figura 12.24*



*Figura 12.24: Parte superior del robot*

Finalmente se puede observar en la *figura 12.25* al robot con su estructura física completa.

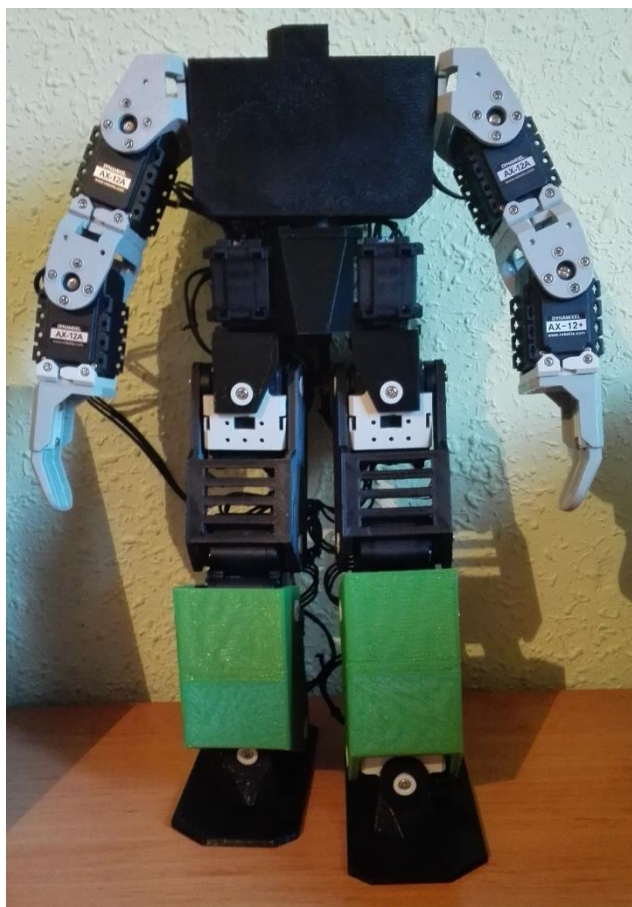


Figura 12.25: Robot completo

### 12.1.2 MANUAL DEL MONTAJE ELCTRÓNICO DEL ROBOT

Lo primero que hay que hacer es soldar la placa del robot con los elementos electrónicos necesarios para el correcto funcionamiento. Estos elementos son:

- 2 Resistencias de 10 Kohmios de  $\frac{1}{4}$  W
- 2 Resistencias de 12 Kohmios de  $\frac{1}{4}$  W
- 2 condensadores de 100uF
- 2 tiras de pines rectos macho de 2x23
- 2 tiras de pines rectos hembra de 1x20
- 3 tiras de pines rectos macho de 1x4
- 1 tira de pines rectos macho de 1x3
- 1 tira de pines codo macho de 1x3

Cuando se tienen todos estos componentes, se sueldan a la placa de la *figura 12.26*, obteniendo el resultado que se aprecia en la *figura 12.27*.

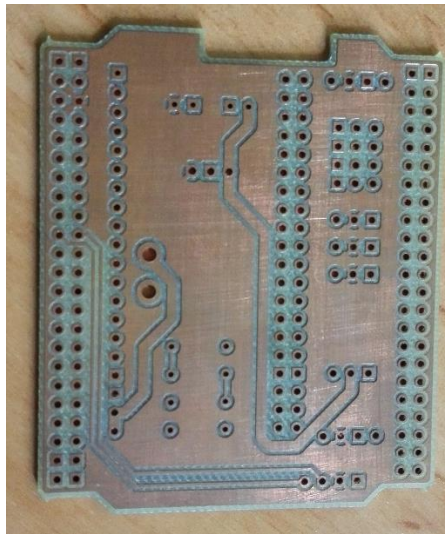


Figura 12.26: Placa

Como se puede observar en la *figura 12.27*, hay dos cables soldados que salen de dos pines. Este arreglo se debe a que en el diseño de la placa inicial había un error que se descubrió después de fabricar. Los pines de la comunicación serie de la OpenCM9.04 estaban desplazados un lugar hacia la izquierda, por lo que con esa simple solución de soldar un par de cables saliendo de los pines erróneos a los pines correctos se solventó el problema.

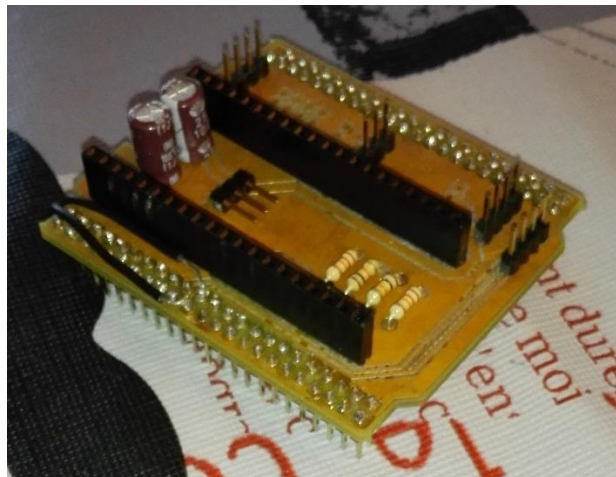


Figura 12.27: Placa rectificada

En la *figura 12.28* se puede observar un diagrama de la placa y las rectificaciones que se han realizado sobre ella.

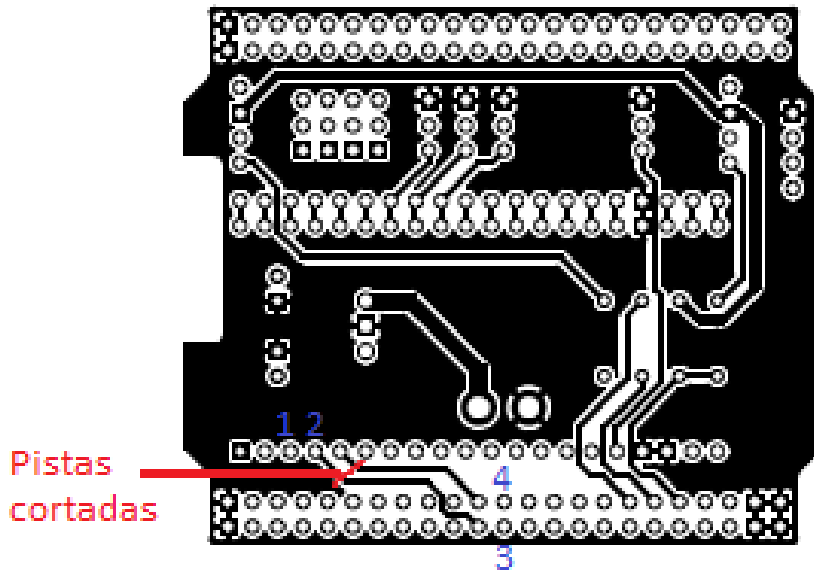


Figura 12.28: Diagrama de la placa rectificadora

Como se aprecia en la figura de arriba, se han cortado las dos pistas originales (señaladas en rojo) que unían erróneamente dos pines no deseados. Posteriormente se conectó el pin nombrado con el número 1 con el pin 3, soldando un cable a cada uno de los pines. De forma idéntica, se unió el pin 2 con el pin 4.

Una vez que se rectificó la placa de manera correcta, se sueldan dos tiras de pines rectos macho de 1x20 a la CM9.04 (figura 12.29) para que de esta forma luego se pueda unir la controladora CM9.04 con la placa rectificadora de la figura 12.27

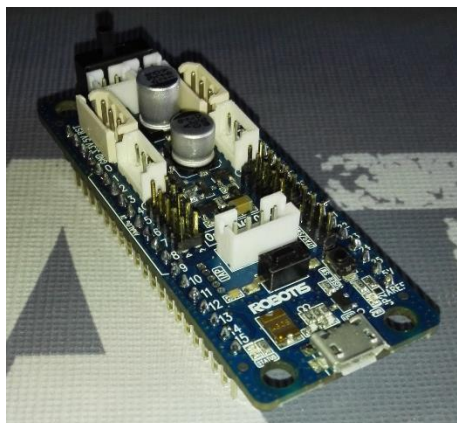


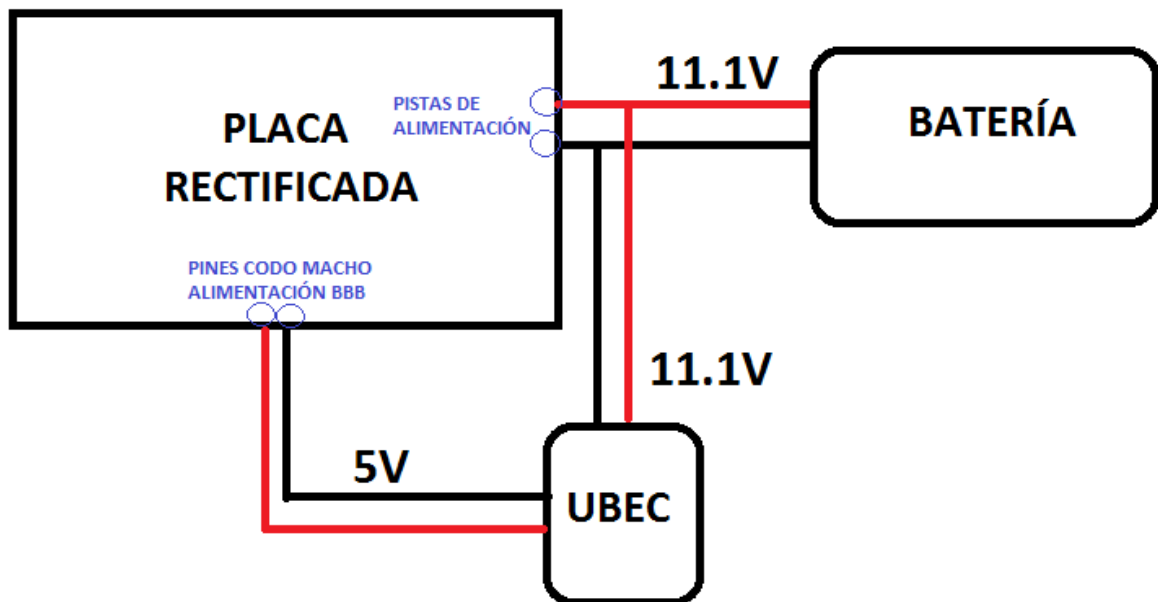
Figura 12.29: OpenCM9.04

Por otro lado, en la cara opuesta de la placa se sueldan cuatro cables a las pistas de alimentación diseñadas en la placa. Una pareja de cable rojo y negro corresponde a la alimentación de todas las placas mediante su conexión posterior a la batería. Para

realizar dicha conexión hay que soldar al otro extremo de los cables un conector XT60 macho para luego poder conectarlo con el conector XT60 hembra de la batería.

De las pistas de alimentación de la placa salen otros dos cables rojo y negro que alimentan al UBEC con 11.1V. El UBEC es un convertor que convierte los 11.1 V de entrada, en 5V de salida. Esto es necesario ya que las tensiones de alimentación de la CM9.04 y la BeagleBone Black son diferentes, concretamente, 11.1 y 5 voltios respectivamente, por lo que con el UBEC se consigue alimentar las dos placas empleando solo una batería de 11.1V. Los cables de salida del UBEC se conectan a la tira de pines codo macho, que es la que está conectada a la alimentación de la BeagleBone Black.

Todo el montaje de cables descrito anteriormente se puede observar en el diagrama de conexionado de la *figura 12.30* y en la *figura 12.31*



*Figura 12.30: Diagrama de conexionado de alimentación*



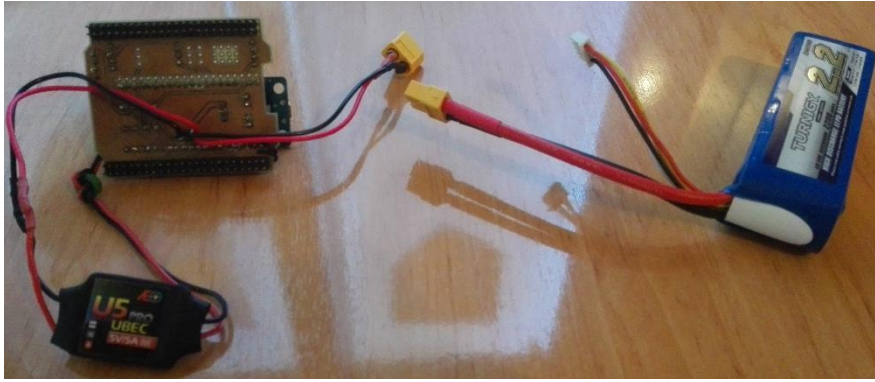


Figura 12.31: Conexión del UBEC y batería. Cables de entrada y salida del UBEC en rojo y negro. Cables de alimentación a la batería en rojo y negro. Conectores XT60 en amarillo

Finalmente la placa, la CM9.04 y la *BeagleBone Black* se conectan mediante las tiras de pines hembras y machos que se han soldado previamente, obteniendo el montaje completo que se observa en la *figura 12.32*. Este montaje posteriormente se introducirá en el interior de la pieza *body* diseñada específicamente para albergar en su interior a las diferentes placas necesarias para el control del robot.

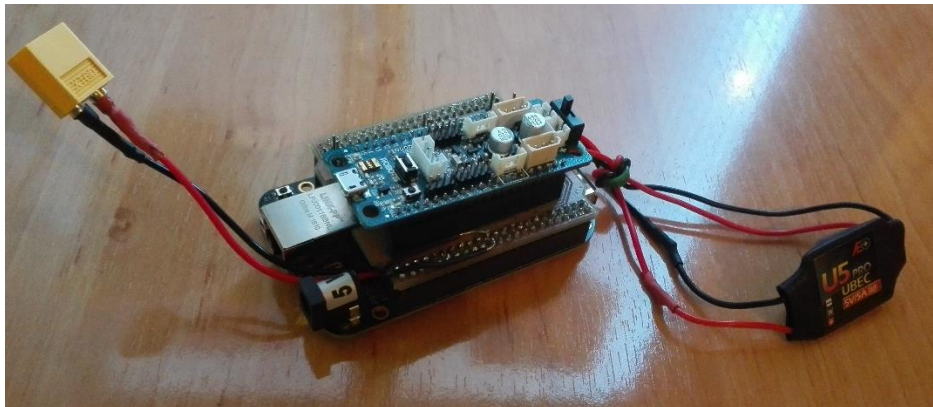


Figura 12.32: Montaje final de las placas

## 12.2 ID'S DE LOS SERVOS DEL ROBOT

En este anexo se identifica a cada uno de los servos o motores del robot con el número de identificación (ID) que se le ha adjudicado (*Tabla X*). De esta forma, en trabajos futuros se podrán emplear las funciones diseñadas en este proyecto si a cada uno de los servos se le adjudica la ID correspondiente que se dice a continuación.

Para adjudicar la ID de un servo *Dynamixel AX-12A*, es necesario modificar el registro 0x03, que es el que hace referencia a la ID del servo. Los diferentes registros de los que dispone este tipo de servo y las características técnicas del mismo se pueden observar en [23].

<b>SERVO</b>	<b>ID</b>
MANO IZQUIERDA	1
HOMBRO IZQUIERDO	2
HOMBRO IZQUIERDO INTERNO	3
HOMBRO DERECHO INTERNO	4
HOMBRO DERECHO	5
MANO DERECHA	6
CINTURA	7
CINTURA IZQUIERDA	8
CINTURA DERECHA	9
PELVIS IZQUIERDA DELANTERA	10
PELVIS IZQUIERDA TRASERA	11
RODILLA IZQUIERDA SUPERIOR	12
RODILLA IZQUIERDA INFERIOR	13
TOBILLO IZQUIERDO DELANTERO	14
TOBILLO IZQUIERDO TRASERO	15
PELVIS DERECHA DELANTERA	16
PELVIS DERECHA TRASERA	17
RODILLA DERECHA SUPERIOR	18
RODILLA DERECHA INFERIOR	19
TOBILLO DERECHO DELANTERO	20
TOBILLO DERECHO TRASERO	21

*Tabla X: ID's de los servos*

## 12.3 ENTORNO SOCIO-ECONÓMICO Y PRESUPUESTO

En este anexo se muestra el impacto socio-económico que supone este proyecto, evaluando los grupos sociales a los que podría estar dirigido, así como las aplicaciones pragmáticas del proyecto. Posteriormente, se realiza un desglose presupuestario de los materiales, equipos y personal de trabajo que han intervenido en la realización de este trabajo.

### 12.3.1 ENTORNO SOCIO-ECONÓMICO

Tal y como se ha mencionado en el capítulo 1.1.2, la robótica mini-humanoide es una rama de la robótica en continuo progreso e investigación. A día de hoy, la robótica mini-humanoide no tiene grandes aplicaciones en la vida cotidiana de las personas, sin embargo, el robot de este proyecto en particular, y los robots mini-humanoides en general, suponen una de las líneas de investigación con mayor potencial dentro de la robótica. Los mini-humanoides son una herramienta cada más importante y habitual para la enseñanza educativa en colegios e institutos, ya que los niños establecen un primer contacto con el mundo de la robótica y la automatización. Además, cada vez son más las investigaciones que se realizan en las universidades, teniendo como objeto de estudio la robótica mini-humanoide (como en el caso de este proyecto), debido al gran potencial que posee esta rama de la robótica.

### 12.3.2 PRESUPUESTO

En la *Tabla XI* se muestra el presupuesto referente a los materiales y equipos que se emplean en este trabajo.

DESCRIPCIÓN	CANTIDAD	COSTE UNITARIO(€)	COSTE TOTAL (€)
Impresora 3D Prusa Air 2	2	500.00	1000.00
Dynamixel AX-12A	21	44.90	942.90
OpenCM 9.04	1	19.90	19.90
BeagleBone Black	1	43.00	43.00
Batería LIPO 3S 11.1V 2200mah	1	16.95	16.95
MPU9250	1	12.37	12.37
Placa de expansión (virgen)	1	10.00	10.00
Resistencia 10KΩ	2	0.06	0.12
Resistencia 12KΩ	2	0.09	0.18
Condensador 100uF	2	0.49	0.98



Tira de pines rectos macho 2x23	2	1.73	3.46
Tira de pines rectos hembra 1x20	2	4.64	9.28
Tira de pines rectos macho 1x10	2	0.78	1.56
Tira de pines machos ángulo 90º 1x3	1	0.40	0.40
Tornillo Allen M2	124	0.15	18.60
Tornillo M3	10	0.09	0.90
Tuerca M3	10	0.08	0.80
<b>TOTAL</b>			<b>2081.40</b>

Tabla XI: Presupuesto Materiales y Equipos

En la *Tabla XII* se muestra el presupuesto referente al personal que ha realizado este proyecto.

DESCRIPCIÓN	HORAS DEDICADAS	COSTE POR HORA(€)	COSTE TOTAL (€)
Ingeniero	990	17	16830.00
<b>TOTAL</b>			<b>16830.00</b>

Tabla XII: Presupuesto del personal

Por último, en la *Tabla XIII* se muestra el coste del presupuesto total del proyecto.

DESCRIPCIÓN	COSTE TOTAL (€)
Materiales y Equipos	2081.40
Personal	16830.00
<b>TOTAL</b>	<b>18911.40</b>

Tabla XIII: Resumen del presupuesto total

Como se observa en la *Tabla XIII*, el coste total de este proyecto tiene un impacto económico de **18911.40 €**, siendo un 89% de dicho coste el empleado para el pago del personal de trabajo, en este caso, el ingeniero, por lo que el 11% restante (2081.40€) es el coste de los materiales y equipos necesarios para la realización del robot, un importe que entra dentro de las posibilidades de compra de un colegio o instituto para su empleo en la enseñanza educativa.

## REFERENCIAS

- [1] IFR, «IFR (International Federation of Robotics),» 2017. [En línea]. Available: <http://www.ifr.org/>.
- [2] ISO, «ISO (International Organization for Standardization),» 2017. [En línea]. Available: [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=55890](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=55890).
- [3] IFR, «IFR (International Federation of Robotics),» 2017. [En línea]. Available: <http://www.ifr.org/history/>.
- [4] UC3M, «RoboticsLab,» 2017. [En línea]. Available: <http://roboticslab.uc3m.es/roboticslab/>.
- [5] E. Guizzo, «IEEE SPECTRUM,» [En línea]. Available: [http://spectrum.ieee.org/autoton/robotics/robotics-software/this\\_three\\_legged\\_robot\\_walks\\_l](http://spectrum.ieee.org/autoton/robotics/robotics-software/this_three_legged_robot_walks_l).
- [6] F. R. C. A. J. C. B. Javier Isabel, «DESARROLLO DE UNA PLATAFORMA ROBÓTICA MINIHUMANOIDE,» 2015. [En línea]. Available: <http://www.ehu.eus/documents/3444171/4484749/104.pdf>.
- [7] J. Isabel, «GitHub,» 2017. [En línea]. Available: <https://github.com/JavierIH/raider/tree/develop/parts/stl>.
- [8] ROBOTIS, «ROBOTIS e-Manual,» 2017. [En línea]. Available: [http://support.robotis.com/en/product/auxdevice/sensor/dxl\\_ax\\_s1.htm](http://support.robotis.com/en/product/auxdevice/sensor/dxl_ax_s1.htm).
- [9] F. R. C. A. J. C. B. Roberto Herrera, «DESARROLLO DE UN ALGORITMO DE CAMINATA PARA EL ROBOT MINI-HUMANOIDE MYOD,» [En línea]. Available: <http://www.ehu.eus/documents/3444171/4484749/72.pdf>.
- [10] A. L. McDonough, «BALANCE & POSTURE,» 2017. [En línea]. Available: <https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj-1Pyd7PPRAhVrCsAKHfs4BUQQFggfMAA&url=http%3A%2F%2Fwww.nyu.edu%2Fclasses%2Fmcdonough%2FBalance%2520%26%2520Posture.ppt&usg=AFQjCNEQIxypm-34uqLUBpNe5i9950ozwA&sig2=SDo>.
- [11] A. Aceves, «ITESM (Institute of Technology and Higher Education),State of Mexico,» [En línea]. Available: [http://homepage.cem.itesm.mx/aaceves/Bogobots/seminario/Como\\_camina\\_robot.pdf](http://homepage.cem.itesm.mx/aaceves/Bogobots/seminario/Como_camina_robot.pdf).

- [12] M. V. & B. BOROVIAC, «Universität Bielefeld,» [En línea]. Available: <https://www.techfak.uni-bielefeld.de/~rhaschke/lehre/WS04/humanoids/papers/zmp-review.pdf>.
- [13] T. Luksch, «Technische Universität Kaiserslautern,» [En línea]. Available: <https://agrosy.cs.uni-kl.de/fileadmin/Literatur/Luksch10.pdf>.
- [14] HONDA, «Asimo Honda,» [En línea]. Available: <http://asimo.honda.com/downloads/pdf/asimo-technical-information.pdf>.
- [15] CEA (Comité Español de Automática), «CEA - Grupo Robótica,» 2017. [En línea]. Available: <http://www.ceautomatica.es/sites/default/files/upload/10/CEABOT/index.htm>.
- [16] BeagleBoard, «BeagleBoard,» 2017. [En línea]. Available: <https://beagleboard.org/latest-images>.
- [17] J. Isabel, «GitHub,» 2017. [En línea]. Available: <https://github.com/JavierIH/raider>.
- [18] QT, «QT Creator,» 2017. [En línea]. Available: <https://www.qt.io/ide/>.
- [19] ROBOTIS, «ROBOTIS OpenCM,» 2017. [En línea]. Available: [http://support.robotis.com/en/techsupport\\_eng.htm#software/robotis\\_opencm.htm](http://support.robotis.com/en/techsupport_eng.htm#software/robotis_opencm.htm).
- [20] J. Isabel, «GitHub,» 2017. [En línea]. Available: <https://github.com/JavierIH/raider/tree/develop/code>.
- [21] InvenSense, «InvenSense,» 2017. [En línea]. Available: <https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/>.
- [22] Á. Bravo Sancho, «EVALUACIÓN DE TÉCNICAS EVOLUTIVAS EN EL CÁLCULO DE LA CINEMÁTICA INVERSA DE ROBOTS MANIPULADORES,» 2012. [En línea]. Available: [http://e-archivo.uc3m.es/bitstream/handle/10016/16016/PFC\\_Alvaro\\_Bravo\\_Sancho.pdf?sequence=1](http://e-archivo.uc3m.es/bitstream/handle/10016/16016/PFC_Alvaro_Bravo_Sancho.pdf?sequence=1).
- [23] ROBOTIS, «ROBOTIS e-Manual,» 2017. [En línea]. Available: [http://support.robotis.com/en/product/actuator/dynamixel/ax\\_series/dxl\\_ax\\_actuator.htm](http://support.robotis.com/en/product/actuator/dynamixel/ax_series/dxl_ax_actuator.htm).