ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DISEÑO Y APLICACIÓN DE UN SISTEMA DE CONTROL PARA LA CONDUCCIÓN DE UN VEHÍCULO ELÉCTRICO POR EL ROBOT BÍPEDO HUMANOIDE NAO USANDO TÉCNICAS DE VISIÓN ARTIFICIAL

CARLOS ALBERTO CARRANCO QUIÑONEZ

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN INDUSTRIAL

Riobamba – Ecuador

Junio 2018

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado "Diseño y aplicación de un sistema de control para la conducción de un vehículo eléctrico por el robot bípedo humanoide NAO usando técnicas de visión", de responsabilidad del Ingeniero Carlos Alberto Carranco Quiñonez, ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

Dr. Juan Vargas Guambo M.Sc. **PRESIDENTE**	FIRMA
Ing. Javier José Gavilanes Carrión M.Sc. **DIRECTOR DE TESIS**	FIRMA
Ing. Patricio German Encalada Ruiz M.Sc. **MIEMBRO DEL TRIBUNAL**	FIRMA
Ing. Gabriel Alfonso Delgado Oleas M.Sc. **MIEMBRO DEL TRIBUNAL**	FIRMA

Riobamba, junio de 2018

DERECHOS INTELECTUALES

Yo, Carlos Alberto Carranco Quiñonez declaro que soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo y que el patrimonio intelectual generado por la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

_____

CARLOS ALBERTO CARRANCO QUIÑONEZ

N° cedula: 1713629564

DECLARACIÓN DE AUTENTICIDAD

Yo, Carlos Alberto Carranco Quiñonez, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

_____

CARLOS ALBERTO CARRANCO QUIÑONEZ

N° cedula: 1713629564

DEDICATORIA

Dedico este trabajo a mi esposa Myrian que gracias a tu apoyo se puede ver culminado con éxito, a mis chiquillos Karlita y David que este esfuerzo que hemos realizado sea un incentivo y ejemplo para ustedes.

Carlos Alberto Carranco Q.

AGRADECIMIENTO

En primer lugar, a Dios quien me ha brindado la vida y la sabiduría para finalizar con éxito este proyecto y permitirme conocer a las personas indicadas.

A la Escuela Superior Politécnica de Chimborazo, por darme la oportunidad y permitirme poder alcanzar un éxito profesional más, en mi vida.

A la Universidad Politécnica Salesiana y a todas las personas que permitieron realizar mi proyecto en el campus sur.

A mi director de tesis, M.Sc. Javier Gavilanes, quien confió y me motivo, en la elección de mi proyecto y sus ideas qué muy acertadamente han sido un aporte.

A la familia Yausin Ocaña por abrirme las puertas de su hogar, en el transcurso de mis estudios.

A mis padres por inculcarme en los inicios de mi vida, esa fuerza y constancia para poder superar las distintas situaciones que se presentan en mi vida.

A mis hermanos Darío, Miguel y David que fueron participes de este proyecto.

A mis suegros David Girón y Blanca Villa por darme ese apoyo incondicional, junto a mi familia.

A mis cuñad@s por el apoyo que me brindan.

A Paulina Bolaños Logroño por brindarme su amistad, apoyo y animo en el aula y desarrollo de mi proyecto.

A todos quienes, con una frase de aliento, me animaron a culminar con éxito.

Dios les bendiga.

Carlos Alberto Carranco Q.

CONTENIDO

RESUMEN ..... **xi**

CAPÍTULO I **1**

1. INTRODUCCIÓN **1**

1.1. Hipótesis **2**

1.2. Diseño **2**

1.3. Antecedentes **2**

1.4. Robots autónomos aplicados en la conducción de un vehículo **3**

1.5. Objetivos de la investigación **11**

1.5.1. Objetivo General ***11***

1.5.2. Objetivos específicos ***11***

1.6. Justificación de la investigación **12**

CAPÍTULO II **13**

2. MARCO TEÓRICO **13**

2.1. Robótica **13**

2.1.1. Robótica bípeda ***13***

2.2. Robot NAO **13**

2.2.1 NAOqi os: ***15***

2.2.1.1. Módulo ALProxy *16*

2.2.1.2. Módulo ALMotion *17*

2.3. Cinemática del robot **21**

2.3.1 Matriz de rotación ***22***

2.3.2 Piernas del robot NAO ***25***

2.3.3 Pierna izquierda. ***26***

2.3.4 Pierna Derecha. ***29***

2.3.5 Movimientos robot NAO ***29***

2.4. Visión artificial **30**

2.4.1 Visión con el robot NAO ***30***

2.4.2 Módulo ALVideoDevice ***31***

2.4.3 Open CV con el robot NAO ***32***

2.5. Python **32**

2.6. Spyder **33**

2.7. Choregraphe **33**

CAPÍTULO III **36**

3. Metodología **36**

3.1. Robot NAO ***37***

3.1.1. Posición sentado del robot NAO ***37***

3.2. Coche eléctrico ***40***

3.3. Sistema de visión ***42***

3.3.1. Calibración de cámara ***43***

3.3.1.1. Proceso calibración cámara robot nao *44*

3.3.2. Campo de visión y conducción ***45***

3.3.3. Procesamiento de las líneas ***46***

3.3.3.1. Obtener Imagen de la cámara superior del robot NAO. *47*

3.3.3.2. Función grayscale *48*

3.3.3.3. Función Gaussian Blur *49*

3.3.3.4. Función undistort *49*

3.3.3.5. Función x_thresh *50*

3.3.3.6. Función mag_thresh *51*

3.3.3.7. Función dir_threshold *52*

3.3.3.8. Funciones HLS y HSV *53*

3.3.3.9. Función warp *54*

3.3.3.10. Función lane_detector y fit *56*

3.3.3.11. Función curvature *57*

3.3.3.12. Detección de personas *58*

3.3.3.13. Procesamiento de la imagen de la cámara del robot NAO. *59*

CAPÍTULO IV **61**

4. Resultados **61**

4.1. Pruebas con imágenes con cámara del robot NAO ***61***

4.2. Detección de personas con cámara del robot NAO en el desplazamiento ***63***

4.3. Espera al detectar personas frente al vehículo. ***64***

4.4. Posición de brazos, pies y piernas para inicio del desplazamiento. ***66***

4.5. Movimientos de la cabeza para seguimiento de líneas. ***68***

4.6. Movimiento de pie derecho aceleración y desaceleración del coche. ***70***

4.7. Movimiento de brazos en el desplazamiento del coche. ***72***

CONCLUSIONES **86**

RECOMENDACIONES **87**

BIBLIOGRAFÍA **88**

ANEXOS

PRESUPUESTO

ÍNDICE DE TABLAS

Tabla 1-1: Ventajas y desventajas de las tecnologías de detección 2
Tabla 1-2: Características del robot NAO..... 15
Tabla 2-2: Detalle de articulaciones del robot NAO. 21
Tabla 3-2: Resoluciones soportadas cámara robot NAO y cuadros por segundo 31
Tabla 4-2: Detalle de paneles Choregraphe 34
Tabla 1-3: Puntos de origen y destino detección de la pista 55
Tabla 2-3: Características Webcam USB con LED 59
Tabla 1-4: Datos de los sensores por articulación (motores encendidos) 81

ÍNDICE DE FIGURAS

Figura 1-1: Niveles de autonomía de un vehículo 4
Figura 2-1: NAO CAR proyecto de Epitech (Montpellier) 5
Figura 3-1: Robot NAO sobre el coche eléctrico 5
Figura 4-1: Aplicación del algoritmo detección carril 6
Figura 5-1: Robot NAO sobre un coche simple diseñado para la conducción 6
Figura 6-1: MotoBot 7
Figura 7-1: Coche Autónomo con Raspberry Pi 8
Figura 8-1: Spirit of Berlin 9
Figura 9-1: Implementación carro autónomo con Raspberry Pi 9
Figura 10-1: Flujograma del procesamiento de la imagen de una pista 10
Figura 1-2: Partes y red de sensores del robot NAO 14
Figura 2-2: Estructira del NAOqi 16
Figura 3-2: Estructura robot nao con el Naoqi 16
Figura 4-2: Cuadro que muestra los robots NAO que se encuentran enlazados 17
Figura 5-2: Partes de articulaciones del robot NAO 17
Figura 6-2: Articulaciones del robot NAO 18
Figura 7-2: Ángulos de la cabeza del robot NAO 18
Figura 8-2: Ángulos de movimiento del brazo derecho robot NAO 19
Figura 9-2: Ángulos de movimiento del brazo izquierdo robot NAO 19
Figura 10-2: Ángulos de movimiento del torso robot NAO 20
Figura 11-2: Ángulos de movimiento pierna derecha robot NAO 20
Figura 12-2: Ángulos de movimiento pierna izquierda robot NAO 21
Figura 13-2: Movimientos de los ejes x, y, z 22
Figura 14-2: Grados de libertad cabeza del robot 22
Figura 15-2: Rango movimiento brazo izquierdo (izquierda) 23
Figura 16-2: Esquema cinemático de las piernas del robot 26
Figura 17-2: Rango movimiento pierna izquierda (izquierda), dimensiones vista frontal robot (derecha) 26
Figura 18-2: Ubicación de cámaras en el robot NAO 30
Figura 19-2: Ángulo de visión (izquierda), ángulo de visión "cámaras robot NAO" 31
Figura 20-2: Interface de Spyder 33
Figura 21-2: Paneles de barra de herramientas en Choregraphe 34
Figura 22-2: Uso de Timeline para obtener las posiciones del robot 35
Figura 23-2: Datos obtenidos en Timeline capturado de cada articulación 35

Figura 1-3: Posición sentado en Choregraphe 38
Figura 2-3: Selección de la cabeza para captura de posiciones 38
Figura 3-3: Guarda los daros de articulación elegida 39
Figura 4-3: Exportar datos para usar en script de Python 40
Figura 5-3: Desarrollo del script en Python con los datos capturados 40
Figura 6-3: a) Mando armado volante, b) Mando montado en la columna dirección 41
Figura 7-3: Estructura del procesamiento de las líneas por cámara del robot NAO 42
Figura 8-3: Diagrama de bloque de detección de pista 42
Figura 9-3: Parámetros intrínsecos de una cámara 43
Figura 10-3: Detección de esquinas 44
Figura 11-3: Imagen original (izquierda), imagen sin distorsión (derecha) 45
Figura 12-3: Pista imagen original (izquierda), pista imagen sin distorsión (derecha) 45
Figura 13-3: Campo visual horizontal 46
Figura 14-3: Archivo creado al ejecutar camera_calibration.py 47
Figura 15-3: Funciones del script 48
Figura 16-3: Imagen capturada con la cámara superior del robot NAO 48
Figura 17-3: Imagen en Gris cámara superior robot NAO 49
Figura 18-3: Suavizado imagen cámara superior robot NAO 49
Figura 19-3: Resultado de aplicar la función undistort 50
Figura 20-3: Parámetros de calibración cámara, resolución 320x240 50
Figura 21-3: Valor absoluto Sobel *x* (izquierda), Valor absoluto Sobel *y* (derecha) 51
Figura 22-3: Imágenes obtenidas en la función magnitud del gradiente 52
Figura 23-3: Resultado obtenido al aplicar la función de dirección de magnitud 53
Figura 24-3: Imagen obtenida al trabajar con la función HLS 54
Figura 25-3: Imagen perspectiva inversa calculada por medio de correspondencias 4 puntos .. 54
Figura 26-3: a) Imagen origen, b) Región de interés, c) Resultado perspectiva “vista de ave”. 55
Figura 27-3: Resultado de aplicar las funciones a la imagen capturada por la cámara del NAO 56
Figura 28-3: Aplicación de la función warp a la imagen capturada (derecha) 57
Figura 29-3: Detección lineas para determinar la curva 58
Figura 30-3: a) Imagen origen, b) Identificación de líneas con función warp, c) Resultado final 58
Figura 31-3: Detección personas y líneas (izquierda), Detección de solo persona (derecha) 59
Figura 32-3: Ubicación webcam en coche. 60
Figura 33-3: Robot NAO sentado en el coche: vista frontal (izquierda), vista lateral (derecha) 60
Figura 1-4: Secuencia imágenes, correspondiente a la pista de conducción del robot NAO. 63
Figura 2-4: Imágenes detección de personas en el trayecto y con el vehículo detenido 63

Figura 3-4: Imágenes detección de personas en el trayecto y con el vehículo desplazándose .. 64
Figura 4-4: Robot acelerando sin detectar a la persona (coche detenido)..... 64
Figura 5-4: Robot acelerando no se detecta personas (coche desplazándose)..... 65
Figura 6-4: Robot sin acelerar detecta una persona, frente al coche detenido..... 65
Figura 7-4: Robot sin acelerar detecta persona, frente al coche (al desplazarse)..... 65
Figura 8-4: Posición sentado, brazos al volante, posición piernas y pies (sin aceleración)..... 67
Figura 9-4: Análisis movimiento de la cabeza..... 68
Figura 10-4: Giro a la izquierda de la cabeza con valor del ángulo..... 69
Figura 11-4: Giro al centro de la cabeza con valor del ángulo 69
Figura 12-4: Giro al lado derecho de la cabeza con valor negativo del ángulo 70
Figura 13-4: Posición desaceleración pie derecho..... 70
Figura 14-4: Posición aceleración pie derecho 71
Figura 15-4: Leyenda del sensor pie derecho 71
Figura 16-4: Inicio de aceleración sobre el pie derecho 72
Figura 17-4: Finalización de aceleración sobre el pie derecho..... 72
Figura 18-4: Análisis de posición del brazo izquierdo 73
Figura 19-4: Inicio de giro al lado izquierdo (brazo izquierdo)..... 73
Figura 20-4: Posición de brazos a la izquierda 74
Figura 21-4: Activación de los motores para mover los brazos..... 74
Figura 22-4: Giro de volante a la derecha grafica de ambos brazos 75
Figura 23-4: Giro completo a la derecha por los brazos (ángulos brazo derecho)..... 75
Figura 24-4: Giro completo a la derecha por brazos (ángulos brazo izquierdo)..... 76
Figura 25-4: Activación y desactivación de los motores de brazos 76
Figura 26-4: Estructura de datos con el NAOQI..... 77
Figura 27-4: Estructura procesamiento imagen para realizar acciones de corrección del coche78
Figura 28-4: Comparación datos con procesamiento imagen y movimientos del robot NAO .. 79
Figura 29-4: Comparación dato al detectar una persona, en la acción del pedal del coche..... 80
Figura 30-4: Datos de los sensores por articulación (motores encendidos)..... 82
Figura 31-4: Datos de los sensores articulación (BRAZOS) 83
Figura 32-4: Datos de los sensores articulación (BRAZOS y CUERPO) 84
Figura 33-4: Datos de los sensores articulación (PIE)..... 85

RESUMEN

El principal objetivo del proyecto es el desarrollo de un sistema de control para la conducción de un vehículo eléctrico por el robot bípedo humanoide NAO usando técnicas de visión artificial. Debido a las múltiples aplicaciones, que se le ha dado al robot humanoide NAO, uno de los motivos ha sido proponer un sistema de navegación, con el uso de los dispositivos propios que tiene el robot. En el desarrollo de la investigación, se ha hecho uso de los distintos software que permiten el trabajo con el robot humanoide NAO, tales como: Choregraphe, Python al igual que las librerías de visión artificial de Open CV, que han sido utilizadas para el procesamiento de imágenes adquiridas en tiempo real con la cámara superior del robot NAO, siendo estas: la detección de líneas, personas y mediante estos datos, el robot corrija la trayectoria ejecutando la acción al volante, en el momento del desplazamiento con el coche. El procesamiento de las imágenes se lleva a cabo por el computador, para lo cual es necesario que la plataforma móvil (coche eléctrico) que está montado el robot NAO, tenga una velocidad mínima determinada, para poder realizar las maniobras respectivas y que el coche se mantenga dentro de la pista. Para la verificación de la hipótesis se realiza pruebas que determinen el funcionamiento para procesar la imagen y los movimientos del robot en el momento de desplazarse con el coche, obteniendo resultados favorables en el procesamiento de la imagen que se adquiere, al igual que los movimientos del robot tanto en el pie derecho para acelerar y los brazos para maniobrar el volante. Para finalizar es recomendable realizar pruebas en una pista de fondo negro y las líneas que delimitan el carril de color amarillo en el lado izquierdo y color blanco en el lado derecho.

Palabras clave: <ROBOT NAO CONDUCTOR> <DETECCIÓN DE LÍNEAS CON TÉCNICAS DE VISIÓN ARTIFICIAL> <DETECCIÓN DE LÍNEAS DE CARRIL POR ROBOT NAO> <CONDUCCIÓN COCHE ROBOT NAO> <DETECCIÓN DE PERSONAS POR EL ROBOT NAO> <PROCESAMIENTO IMÁGENES ROBOT NAO> <OPENCV (VISIÓN ARTIFICIAL)>

ABSTRACT

The main objective of the project is the development of a control system for the conduction of an electric vehicle by the NAO humanoid biped robot using artificial vision techniques. Due to the multiple applications, which have been given to the robot NAO, one of the reasons has been to propose a navigation system, with the use of devices that the robot have. In the development of the research, it was used the various software that allow the work with the robot humanoid NAO, such as: Choregraphe, Python as well as the open CV artificial vision libraries, which have been used for the processing of Images acquired in real time with the superior camera of the NAO robot, being these: the detection of lines, people and through this data, the robot correct the trajectory by executing the action at the wheel, at the moment of the displacement with the car. The processing of the images is carried out by the computer, for which it is necessary that the mobile platform (electric car) that is assembling the robot NAO, has a certain minimum speed, to be able to carry out the respective maneuvers and the car is Keep inside the track. For the verification of the hypothesis is carried out tests that determine the operation to process the image and the movements of the robot at the moment of moving with the car, obtaining favorable results in the processing of the image that is acquired, just like the robot moves both on the right foot to accelerate and the arms to maneuver the steering wheel. To finish it is advisable to perform tests on a black background track and the lines that delimit the yellow lane on the left side and white color on the right side.

Key words: <NAO DRIVER ROBOT>, <DETECTION OF LINES WITH ARTIFICIAL VISION TECHNIQUES>, <DETECTION OF RAIL LINES BY NAO ROBOT>, <DRIVING CAR ROBOT NAO>, <DETECTION OF PEOPLE BY THE NAO ROBOT>, <PROCESSING ROBOT NAO IMAGES>, <OPENCV (ARTIFICIAL VISION)>

CAPÍTULO I

1. INTRODUCCIÓN

Una pequeña pérdida de concentración puede hacer que el vehículo se desvíe de su trayectoria y pase al otro carril o se salga de la carretera. Más de un tercio de los accidentes de camión o incidentes graves en las carreteras están relacionados con salidas accidentales del carril, en especial al final de un largo y agotador día de trabajo, ya que los conductores son más susceptibles de distraerse por factores externos

Los accidentes en carreteras son atribuidos a errores del conductor, la somnolencia, la fatiga o las distracciones son al parecer ser las causas principales.

El sistema de advertencia de salida de carril, está basado en la tecnología Machine Visión. Una cámara situada detrás del parabrisas que controla la posición del vehículo en relación a las líneas de la carretera, capaz de adaptarse a las distintas condiciones que se presenten. Si el vehículo va a salirse del carril por cualquiera de los dos lados, se silencia la radio y se emite un distintivo sonido de banda sonora por el altavoz del lado correspondiente para advertir al conductor que debe desviarse hacia el otro lado. Las falsas alarmas se evitan mediante un sistema lógico de reconocimiento de los cambios de carril intencionados. (DAF Trucks N.V., 2018)

La aplicación conjunta de la robótica y visión artificial para la conducción de un coche eléctrico por el robot bípedo humanoide NAO, es una adaptación reducida de un vehículo autónomo, tales como los prototipos de Google, Tesla que se conducen de forma autónoma, con la implementación de sensores: cámaras, ultrasonidos, GPS de alta precisión e instrumentos de localización láser conocidos como *lidar*. (Will Knight, 2015)

Estos dispositivos implementados en el vehículo crean una imagen del ambiente que le rodea, para poder conducir mayor seguridad. (Will Knight, 2015)

Algunos sistemas constan de sensores para la detección de las líneas del carril, disponen de una unidad de control electrónica que se encarga del procesamiento de las señales recibidas por los sensores, calculando en tiempo real la posición y trayectoria del vehículo con referencia a las líneas delimitadoras del carril, disponiendo de una interfaz de usuario que controla la puesta en marcha del sistema y de los sistemas de aviso al usuario (acústico, óptico y/o mediante vibraciones del volante o el asiento).

Las tecnologías de detección utilizadas al momento son tres. La más simple y de un costo bajo utiliza sensores infrarrojos que detectan las líneas de la carretera, en tanto que los dos restantes usan la visión artificial y el escaneado láser del entorno. (FITSA, 2007)

Tabla 1-1: Ventajas y desventajas de las tecnologías de detección

	VENTAJAS	DESVENTAJAS
Sensores IR	• Sencillez • Bajo costo	• No prevé trayectoria. • No distingue entre señalización horizontal
Visión artificial	• Predice trayectoria • Distingue entre diversos tipos de señalización horizontal	• Problemas en curvas • Problemas en condiciones de visibilidad adversa (niebla…). • Alto coste
Scanner Laser	• Acoplamiento con sistemas de alerta de cambio involuntario de carril. • Implementación en vehículos	• Muy alto costo.

Fuente: FITSA 2007

1.1. Hipótesis

Se propone diseñar un sistema de control e implementarlo en un robot humanoide bípedo, para la conducción de un coche eléctrico, tratando de resolver de una manera eficiente las maniobras a realizar por parte del robot, basándose en la información obtenida de imágenes capturadas y debidamente procesadas en tiempo real, trazando la trayectoria a seguir y evitando colisiones con personas, permitiendo así controlar la dirección y el desplazamiento de un coche eléctrico.

1.2. Diseño

a. Localización

Quito - Pichincha.

Laboratorios de electrónica, Universidad Politécnica Salesiana, campus sur.

1.3. Antecedentes

La robótica y visión artificial, nos han permitido emprender nuevas aplicaciones en la conducción autónoma de un vehículo, brindando serenidad en las distintas situaciones que habitualmente el hombre realiza en el trayecto de rutina o retenciones en una pista de manejo.

En Ecuador no se han propuesto nuevas aplicaciones con robots en la conducción autónoma de un transporte, se podrían introducir una flota de robots para la navegación de vehículos realizando ciertas modificaciones en el vehículo de transporte terrestre, para la navegación en lugares determinados, por ejemplo en hospitales para ofrecer comidas, entrega de medicamentos, transportar instrumentos estériles, desechos, ropa y artículos de farmacia en todo el edificio sin intervenir en ciertos casos la ayuda de ningún ser humano. Otro caso sería implementarlos en

aeropuertos, el guiado de personas en museos, plantas industriales, centros educativos, almacenes entre otras múltiples aplicaciones que podríamos efectuar en nuestro medio.

1.4. Robots autónomos aplicados en la conducción de un vehículo

Todos los fabricantes de vehículos están incorporando a sus modelos sistemas de ayuda a la conducción en mayor o menor medida. Estos sistemas, como son el frenado de emergencia autónomo, mantenimiento de carril, detección del ángulo muerto o aparcamiento automático, hacen que el conductor se sienta cada vez más apoyado en la conducción diaria, ya que suponen una gran ayuda en situaciones de peligro o en las rutinas diarias al volante. Estas tendencias en los vehículos van encaminadas a la conducción autónoma, es decir, que no sea necesario la participación del conductor para realizar el trayecto. En este proceso de transformación la Sociedad de Ingenieros del Automóvil (SAE) ha realizado una clasificación de los vehículos con relación a los sistemas que cada uno incorpora y las acciones realizadas sin colaboración del conductor. A continuación, se describen los diferentes niveles de esta clasificación: (Centro Zaragoza, 2016)

Nivel 0. El coche no tiene ningún sistema automatizado que le permita tomar el control, solamente puede disponer de algún sistema de advertencia. (Centro Zaragoza, 2016)

Nivel 1: El ser humano, lleva el control de todos los aspectos en la conducción del vehículo, mientras dura el viaje. El coche dispone de sistemas de apoyo para controlar la velocidad crucero o la asistencia de mantenerse dentro del carril por el que se circula. (Centro Zaragoza, 2016)

Nivel 2. El coche puede calificarse como semiautónomo. En este caso, el conductor debe estar atento por si se inicia una situación de inseguridad y tomar los mandos con presteza, además el sistema se desactivará cuando el conductor tome el control. Un vehículo de estas características es el Mercedes Clase E con su sistema Drive Pilot, que lleva la conducción a un estado intermedio entre la conducción tradicional y la conducción totalmente autónoma. (Centro Zaragoza, 2016)

Nivel 3. El coche puede circular de forma totalmente autónoma en entornos controlados como en autopistas. En esta categoría aparece el Tesla Model S con su sistema Autopilot que por defecto se encuentra siempre desconectado. El conductor lo puede activar voluntariamente y el sistema se encargará de conducir el vehículo siempre supervisado por el conductor, además el sistema verifica que el conductor permanece alerta y con las manos al volante. (Centro Zaragoza, 2016)

Nivel 4: La evolución de los sistemas de conducción autónoma se perfecciona poco a poco hasta el punto de no requerir la intervención humana en su totalidad. Controlando el tráfico y las diversas condiciones del medio, definiéndose la mejor ruta (inclusive las alternas) sabiendo responder ante cada situación. (Centro Zaragoza, 2016)

Nivel 5: El vehículo es capaz de dirigirse a cualquier sitio, sin mandos de ningún tipo, con una interfaz donde se introduciría las órdenes, requeridas por el usuario.(Siemens S.A., 2007)

Hasta el momento en nuestro entorno, aún estamos estancados entre la fase 1 y 2. Aunque pocas son las aplicaciones en el nivel 3, queda una brecha grande por recorrer, para llegar a ver en las carreteras un vehículo capaz de controlarse por sí mismo con un sistema bastante robusto en las maniobras. Aproximadamente, la conducción autónoma total, será en el 2030. (Siemens S.A., 2007)

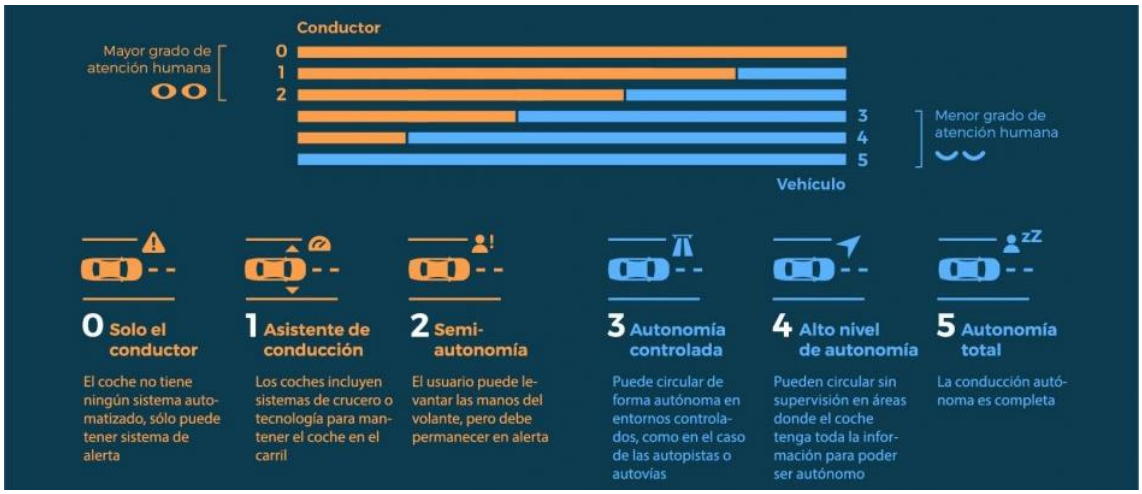


Figura 1-1: Niveles de autonomía de un vehículo

Fuente: Leire Pérez km77.com

Antecedentes bibliográficos

A. NAO CAR.

El robot NAO de la empresa Aldebaran Robotics, es capaz de conducir un mini coche eléctrico para niños. Es el resultado de una colaboración entre cuatro estudiantes de Epitech (Montpellier). Nao puede conducir de forma autónoma interpretando una carretera, definiendo los movimientos que le permitan conducir. Él es capaz de iniciar, ejecutar, parar y volver atrás.

Con la cámara frontal el robot Nao recupera la imagen de ruta indicada por una línea roja. A partir de los datos recogidos realiza los movimientos adecuados. Basándose en los datos y su propia orientación, es capaz de calcular la dirección en la que ejecuta la maniobra sobre la línea.

Con todos estos pasos, Nao demuestra su capacidad para conducir de forma autónoma a lo largo de una ruta predefinida.

“Nao Car es la continuidad de la investigación actual dirigida a reducir las barreras entre el hombre y la máquina. En este caso, el robot Nao que conduce un coche para niños se demuestra su capacidad para interactuar sin problemas con objetos diseñados para los humanos”. (GeekMag, 2012)

Figura 2-1: NAO CAR proyecto de Epitech (Montpellier)

Fuente: Epitech, 2012

B. Sensor basado en la navegación móvil utilizando el robot humanoide nao.

Este trabajo esta propuesto sobre un sistema de navegación por el robot humanoide NAO sobre un coche eléctrico para niños (plataforma móvil), con la ayuda de sensores infrarrojos para esquivar obstáculos, el robot hace los movimientos respectivos para el manejo del coche eléctrico. Con la ayuda de una tarjeta Arduino Uno se controla a los motores y la recepción de las señales de los sensores infrarrojos para conseguir el objetivo. El programa principal se ejecuta con Choregraphe interactuando con el robot. (Ariffin et al., 2015)

Figura 3-1: Robot NAO sobre el coche eléctrico

Fuente: Ariffin et al., 2015

C. Búsqueda avanzada de carriles

En este proyecto se ejecuta varias versiones, el objetivo es elaborar un software para identificar los límites de carril en un vídeo (Priyanka Dwivedi, 2017). Para cada imagen del video, se detecta las líneas del carril de conducción a cada lado del coche y las resalta visualmente. El algoritmo de detección de carriles funcionará bien en caminos no curvos con una sola cámara. En las

imágenes, las condiciones climáticas son buenas. Se procesa las imágenes sobre sombras, colores de línea diferentes y carriles continuos / discontinuos.(SqalliFollow, 2017)

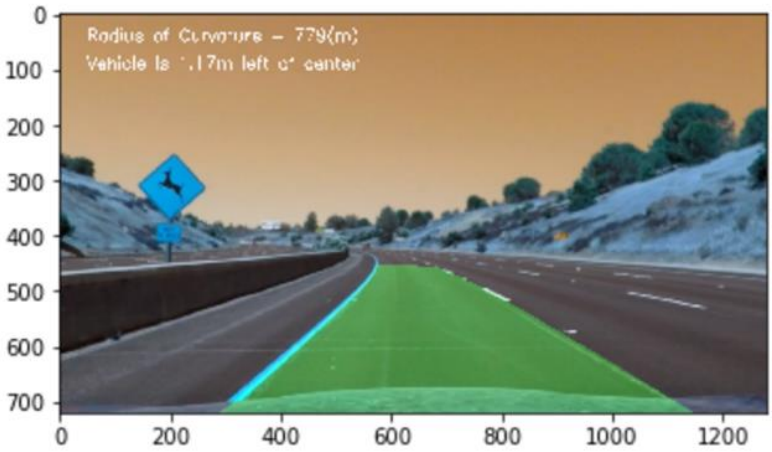


Figura 4-1: Aplicación del algoritmo detección carril

Fuente: Priyanka Dwivedi, 2017

D. Vehículo simple para el transporte de un robot humanoide nao

Presenta un soporte de montaje simple que sin el uso de piezas móviles es capaz de permitir que un robot humanoide Nao pueda montarlo, desmontarlo y conducir un vehículo robótico simple. Al tomar ventaja de la forma del robot humanoide Nao, se evita el uso de piezas móviles en el soporte de montaje. Permitiendo que el costo del soporte sea bajo y de fácil fabricación.

Además, mediante la utilización de la forma del torso del robot Nao, el soporte de montaje es capaz de bloquear al robot en su lugar mientras el vehículo está en movimiento, lo que permite que el vehículo realice cambios bruscos sin desalojar al Nao del soporte. (Suddrey, Cunningham-Nelson, Richards, Gariano, & Maire, 2014)

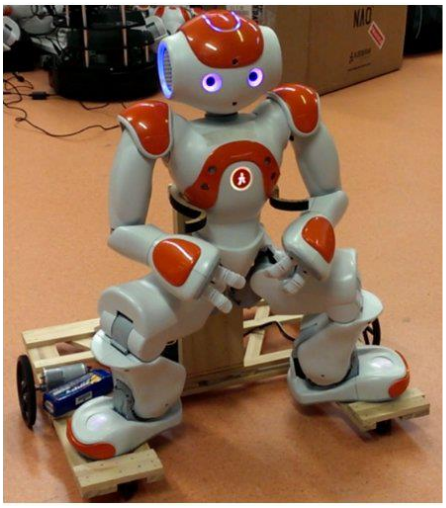


Figura 5-1: Robot NAO sobre un coche simple diseñado para la conducción

Fuente: Suddrey, Cunningham-Nelson, Richards, Gariano, & Maire, 2014

E. MOTOBOT

Es un sistema automatizado que se podría ajustar en cualquier motocicleta, haciendo un proyecto apropiado, además puede conducir a una velocidad de hasta 200 km/h.

Motobot de Yamaha es un proyecto de robótica dentro del departamento de I+D en la compañía. El objetivo, es el desarrollo tecnológico que permita crear en el futuro sistemas de conducción autónomos en sus motos. Para lo cual, han creado un robot y enseñarle a conducir y los datos obtenidos usarlos para mejoras de la conducción creando algo similar a lo que se implementa en los coches: un software que tome el control del vehículo. (J. C. González, 2015)

Figura 6-1: MotoBot

Fuente: J. C. González, 2015.

F. Coche autónomo con raspberry PI

Proyecto desarrollado por Zeng Wang es una versión reducida de un sistema de auto-conducción (usando un coche radio control) con Raspberry Pi 2 Model B – Placa base (ARM Quad-Core 900 MHz, 1 GB RAM, 4 x USB, HDMI, RJ-45), Arduino y un computador con software de código abierto.

El uso de una Raspberry Pi con una cámara incorporada, un sensor ultrasónico como entradas, envían las imágenes al computador en forma de video encargándose del procesamiento en tiempo real del video, realizando el reconocimiento de objetos (señales de tránsito) y medición de distancia ordenando las señales correspondientes de la dirección mediante una placa Arduino para el control del coche de RC. (Navarro, 2016)

Figura 7-1: Coche Autónomo con Raspberry Pi

Fuente: Navarro, 2016

G. Visión artificial aplicada a vehículos autónomos Spirit of Berlin

Aquí se trata sobre dos conceptos modernos de la ingeniería. Por un lado, las técnicas de visión por computadora y, por el otro, el diseño de automóviles con capacidades de guiado independiente o autónomo. Tales conceptos pueden parecer muy alejados de los elementos de confort que utilizamos actualmente. Sin embargo, es posible entender rápidamente su potencial imaginando autobuses del transporte público que a través de cámaras de visión artificial puedan evitar el contacto con otros vehículos, capaces de aproximarse lentamente y sin errores a la parada o a la estación de servicio. Además, pueden inteligentemente reducir la velocidad según el congestionamiento vial o bien regular su desplazamiento por la cantidad de personas que en ese momento ocupan dicha unidad. Es precisamente este tipo de aplicaciones de los sistemas de visión en automóviles los que han capturado un profundo interés de parte de la comunidad científica en últimas fechas. La operación autónoma desde un punto de vista técnico es el resultado de una integración coordinada de información proveniente de distintos sensores, entre los cuales, aquellos basados en visión artificial toman un rol preponderante.(Pérez Cisneros, Marco Antonio ; Cuevas Jiménez, Erik Valdemar ; Zaldívar Navarro, 2009)

Figura 8-1: Spirit of Berlin

Fuente: Zaldívar Navarro, 2009

H. Diseño e implementación de un carro autónomo usando raspberry PI

El proyecto tiene como objetivo construir un prototipo de carro autónomo con visión monocular usando Raspberry Pi para el procesamiento. Una cámara de alta definición con un sensor de ultrasonido que es usado para proveer datos en tiempo real. El coche es capaz de alcanzar el destino de forma segura e inteligentemente, evitando el peligro para seres humanos. Varios algoritmos como la detección de líneas, detección de obstáculos se combinan para proporcionar el control sobre el coche.(Gurjashan Singh Pannu, Pritha Gupta, 2015)

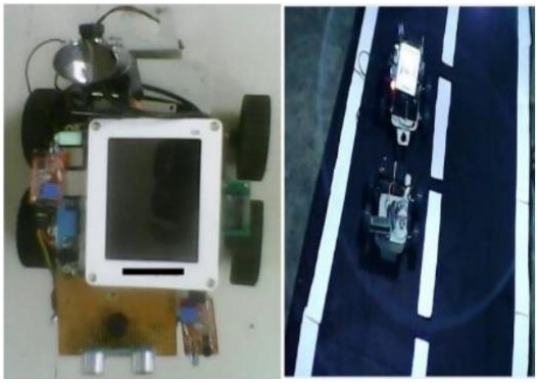


Figura 9-1: Implementación carro autónomo con Raspberry Pi

Fuente: Pritha Gupta, 2015

I. Procesamiento de imagen para sistemas de carril con detección de borde usando técnicas transformada de Hough.

Proyecto que ha sido llevado a revolucionar la seguridad en sistemas automotrices con la ayuda de procesamiento de imagen, que ayuda al conductor en la conducción eficiente. Una cámara de alta velocidad escanea la imagen de la escena del camino. Las imágenes se procesan en la

memoria, posteriormente estas imágenes pasan por TÉCNICAS DE DETECCIÓN DE BORDE. Esta técnica de detección de bordes consiste en un algoritmo que detecta los carriles en la carretera. Luego, después de usar la transformada de Hough estas líneas son detectadas y comparadas con la posición actual del coche. Esto ayuda al sistema a poner un control constante en el coche sobre su posición en el carril. Este concepto también ayuda en la revolución futura de la conducción automática en carreteras. Además, el uso de procesadores de alta velocidad como procesadores Blackfinn hace que sea muy sencillo implementar este sistema en tiempo real.(Lakshmi, Rajya, Mounika, 2010)

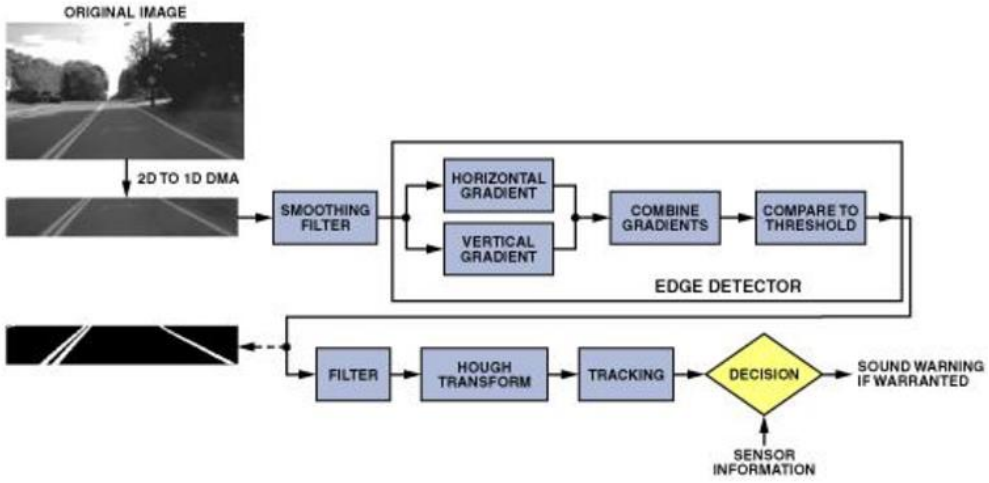


Figura 10-1: Flujograma del procesamiento de la imagen de una pista

Fuente: Us et al., 2010

1.5. Objetivos de la investigación

1.5.1. Objetivo General

Diseñar y aplicar un sistema de control para la conducción de un vehículo eléctrico por el robot humanoide NAO usando técnicas de visión artificial.

1.5.2. Objetivos específicos

1. Desarrollar el sistema de movimientos del robot para que sea capaz de realizar las maniobras correctas y satisfactorias en el entorno de forma aproximada. Además de corregir sus movimientos de forma automática, producidos en el momento de la conducción.
2. Procesar las imágenes obtenidas en tiempo real en el instante de la conducción, para realizar las maniobras respectivas en la pista de manejo.
3. Implementar el reconocimiento de la pista en el robot NAO usando un algoritmo visual que sea rápido y robusto en las variaciones del ambiente.
4. Determinar los dispositivos propios del robot bípedo humanoide NAO para el control del vehículo.
5. Analizar el método más factible para la programación del robot bípedo humanoide NAO.
6. Evaluar el comportamiento y la utilidad del robot bípedo humanoide NAO, en la pista de manejo con pruebas experimentales.

1.6. Justificación de la investigación

Esta investigación está enfocada en estudiar el procesamiento de imágenes por el robot humanoide NAO para los movimientos que éste deberá realizar en el manejo de un coche eléctrico para niños y en un futuro poder realizar esta aplicación en un vehículo motorizado destinado para el ser humano, apoyado de las distintas aplicaciones tecnológicas.

El robot humanoide NAO tiene una gran aplicación en modelo conceptual y teórico, el uso en la investigación se da en la actualidad para experimentos prácticos, para poder emplearlos en distintas tareas que para el ser humano parecen triviales.

Se implementa una pista adecuada delimitando el lugar que será ocupado por el robot NAO, en ciertos casos similar a una pista de carretera relacionada con automotores.

En la conducción de un vehículo se debe tener en cuenta varios factores, el de mayor relevancia el evitar accidentes causados por imprudencia del conductor o peatón, sean por: distracción, cansancio, falta de ángulo de observación, no mantener la distancia adecuada con otro vehículo, conducción bajo efectos del alcohol o sustancias que alteran a la salud física y mental del ser humano, exceso de velocidad, irrespeto de las señales de tránsito, desperfectos mecánicos del automóvil entre otros. Con la ayuda de las nuevas tecnologías aplicadas a un vehículo y principalmente en la conducción se tiene mejor control ya que contrarresta ciertos factores causantes de eventos trágicos para el ser humano.

CAPÍTULO II

2. MARCO TEÓRICO

Se explica los fundamentos teóricos más importantes para el proyecto, en el desarrollo de la programación se tiene ciertos requerimientos, en este caso visión artificial y los movimientos propios que tendrá que desarrollar el robot para las maniobras requeridas, esto basados en las imágenes que se capturen en tiempo real.

2.1. Robótica

La robótica se encuentra asociada al estudio de las tecnologías básicas del robot. Este estudio se relaciona con la investigación teórica - práctica, dividiéndose en el diseño del robot, su mecánica, la planeación y control de trayectoria, su programación e inteligencia artificial. (Rafael, 2006)

2.1.1. Robótica bípeda

La robótica bípeda conforma un área de investigación con gran crecimiento en los últimos años. Si bien el desplazamiento mediante ruedas es más eficiente y permite mayor velocidad, los robots con patas son más versátiles y pueden desplazarse en terrenos irregulares. En particular, los bípedos son esencialmente aptos para manejarse en nuestro entorno, por contar con características similares a los seres humanos, por lo que sin tener que modificar nuestros hogares y lugares de trabajo estos robots pueden realizar tareas por nosotros, siendo particularmente interesante su aplicación en trabajos que ponen en riesgo la salud o la vida de las personas. (Gutiérrez González, Ribacoba Menoyo Tutores, & Etxebarria Ecenarro Alfredo García Arribas, 2007)

2.2. Robot NAO

En América Latina, varias universidades y organizaciones trabajan con los robots humanoides NAO desarrollándolos en diferentes ámbitos, como la inteligencia artificial, procesamiento de señales (audio y video, imágenes entre otros). En el campo de la medicina e incluso en relaciones sociales principalmente con niños y personas de la tercera edad, también ha sido incluido este robot (Moreno Wilfrido, 2016). Una de las aplicaciones que se podría enfatizar es la mejora en el aprendizaje de niños con autismo estimulando los procesos cognitivos y perceptivos a través de *Autism Solution for Kids (ASK).*(“NAO Educación - Aliverobots”, s/f)

El robot humanoide NAO mide 58 centímetros, es interactivo y totalmente programable, con 25 grados de libertad. Dispone de una red de sensores que incluye dos cámaras, cuatro micrófonos, nueve sensores táctiles, sonares y sensores de presión, además de un sintetizador de voz y dos parlantes de alta fidelidad. También dispone de un SDK para el desarrollo de software, con una interfaz amigable, compatible con Windows, Linux y Mac, en la que se pueden usar distintos

lenguajes de programación, como C++, Python, JAVA, .NET y MATLAB. (Moreno Wilfrido, 2016)

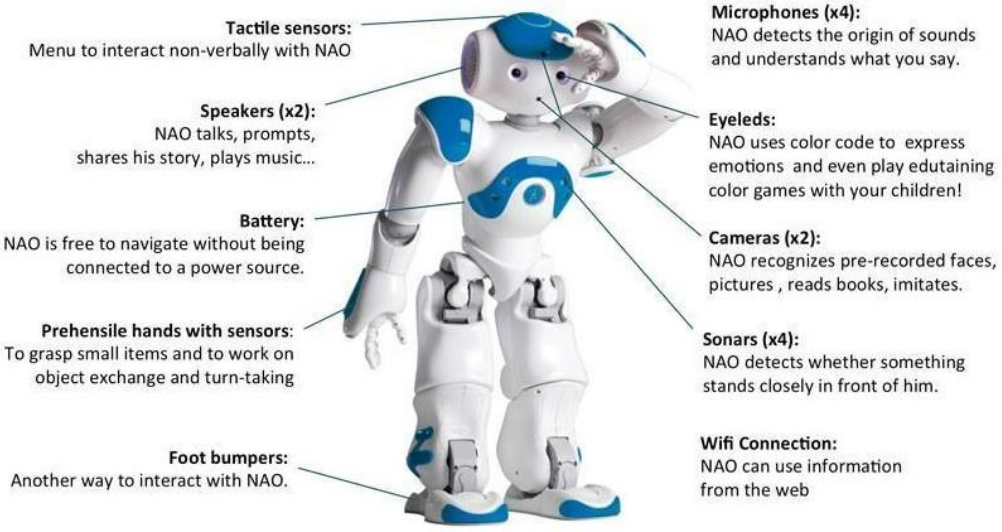


Figura 1-2: Partes y red de sensores del robot NAO

Fuente: ZeitgeistLab.ca ©, 2016

Aldebaran Robotics es actualmente líder en robot humanoides, desde el 2011 lanzó una nueva generación de los robots NAO, con grandes avances para la interacción del robot con el medio. Las nuevas generaciones de robots NAO son totalmente programables siendo uno de los robots a nivel mundial de gran uso. En la Figura 2-2, una de las mejoras del robot NAO trata de la computadora a bordo, con un procesador Intel Atom de 1.6 GHz se ejecuta un kernel de Linux y el middleware de Aldebaran llamado NAOqi. Disponible con dos cámaras de alta definición, conectadas a un FPGA permitiendo la recepción simultanea de imágenes con mejor velocidad y desempeño, lo que permite un mejor procesamiento de imágenes, incluso con baja iluminación.(Ocampo, 2014)

Tabla 1-2: Características del robot NAO

Altura	58 cm
Peso	4.3 Kg
Energía	Batería Li-Ion, 6 celdas en seria, Vnom = 21.6 V, I = 2 A
Autonomía	60 minutos (uso activo), 90 minutos (uso normal)
Grados de libertad	25 grados
CPU	Intel Atom @ 1.6 GHz
Desarrollado en el SO	NAOqi (Basado en Linux)
SO compatibles	Windows, Mac OS, Linux
Lenguaje de programación	C++, Python, Java, MATLAB, Urbi, C, .NET
Visión	Dos cámaras 1280x960 HD
Conectividad	Ethernet, Wi-Fi
Sensores	Giroscopio, Acelerómetro, Bumpers, Sonar, I/R

Fuente: Aldebaran documentation, 2012

2.2.1 NAOqi os:

Aldebaran Robotics desarrollo un Framework llamado NAOqi, es el sistema operativo que se ejecuta internamente en el robot NAO, permite crear nuevas funciones para el robot siendo, rápido, seguro y multi-plataforma. EL NAOqi permite una comunicación homogénea entre los distintos módulos que el robot NAO permite trabajar (motricidad, audio, video), de igual manera la programación y la compartición de recursos será de manera homogénea.(Lluch, 2016)

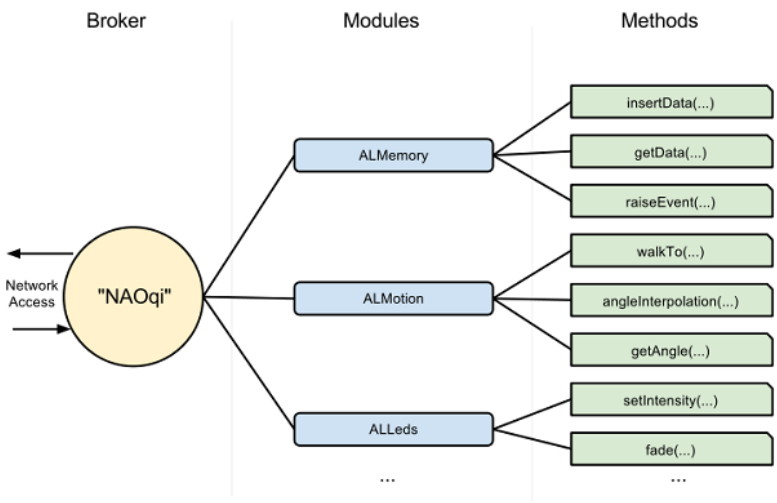


Figura 2-2: Estructira del NAOqi

Fuente: Aldebaran documentation, 2012

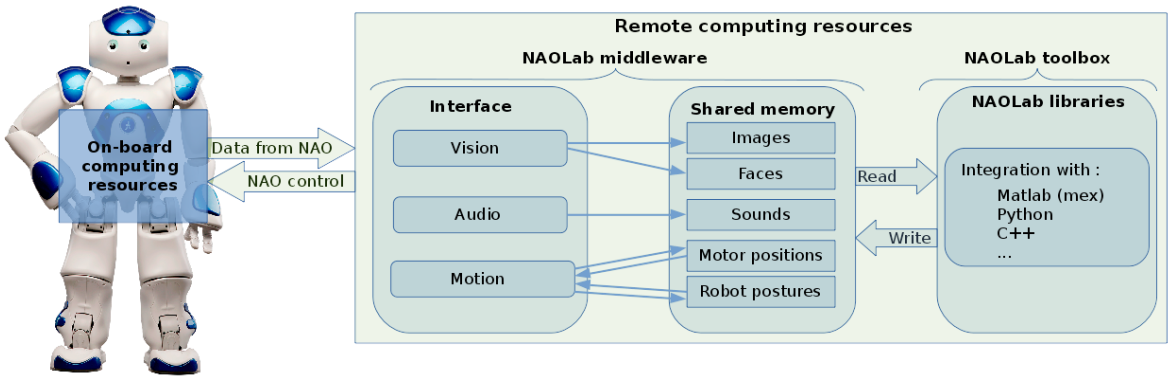


Figura 3-2: Estructura robot nao con el Naoqi

Fuente: Soraya Arias, 2015

2.2.1.1. Módulo ALProxy

El Módulo ALProxy permite la conexión proxy con el robot NAO, este debe ser llamado en cada módulo que se precise usar en el robot, su configuración es la siguiente:

ALProxy ("nombre_modulo", "IP_robot", PORT).

nombre_modulo: Es el módulo el cual se quiere conectar y usar en el robot NAO

IP_robot: Es la dirección IP del broker en el cual el módulo está ejecutándose sobre el robot NAO.

PORT: Es el número del puerto del broker del robot NAO

Los datos de IP y PORT se los puede visualizar en Choregraphe al hacer clic sobre el icono

Connect to , luego se despliega un cuadro de dialogo ver **¡Error! No se encuentra el origen de la referencia.**-2.

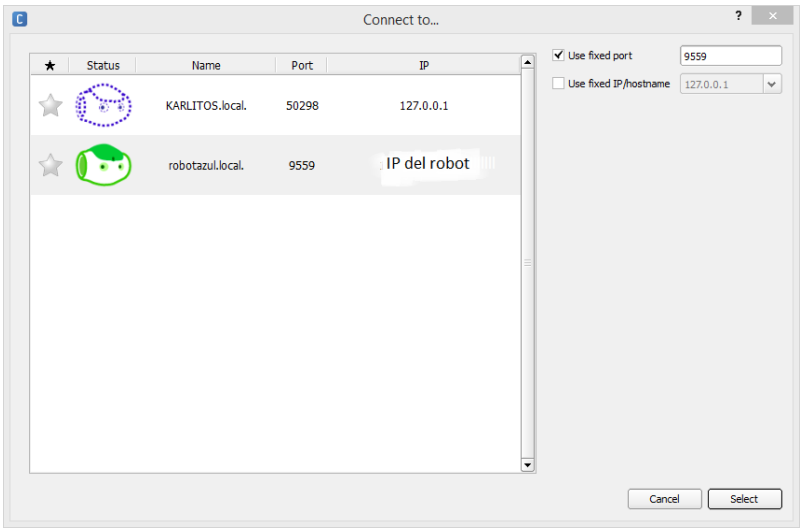


Figura 4-2: Cuadro que muestra los robots NAO que se encuentran enlazados

Realizado por: CAR, Carlos Alberto, 2017

2.2.1.2. Módulo ALMotion

El Módulo ALMotion es el encargado del movimiento en general del robot NAO, manipulando los motores que actúan sobre las articulaciones del robot. Cuando se requiera el movimiento de cierta articulación se hace el llamado mediante ALMotion que funciona en ciclos tardando 20 ms.(Ocampo, 2014)

Se detalla las articulaciones del robot NAO acopladas **¡Error! No se encuentra el origen de la referencia.**-2, en la Figura 5-2 se tiene una visión general en relación entre los diferentes nombres y se especifica qué se incluye el modelo con la junta, forma como se encuentran enlazados con al robot ya que un factor importante en el momento de poner en funcionamiento las articulaciones al realizar una aplicación.

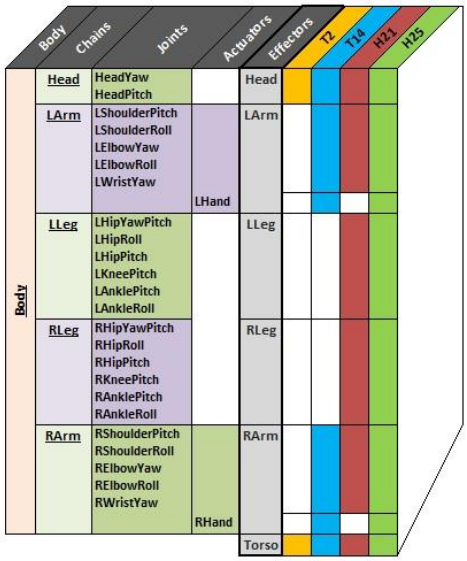


Figura 5-2: Partes de articulaciones del robot NAO

Fuente: Aldebaran documentation, 2012

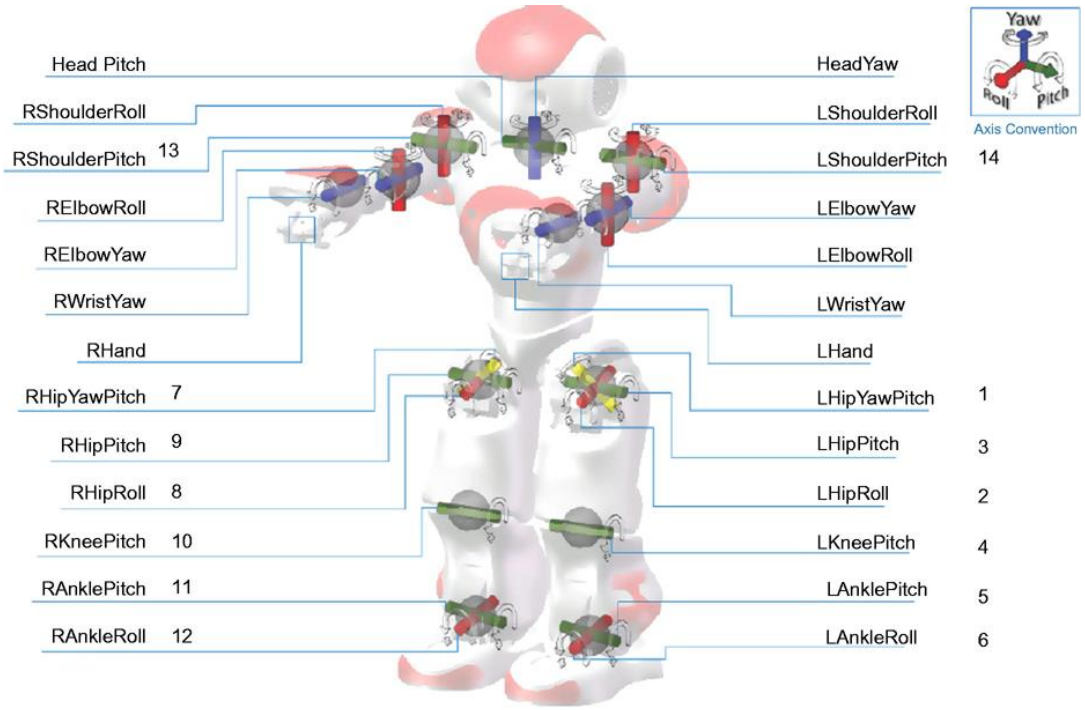


Figura 6-2: Articulaciones del robot NAO

Fuente: Aldebaran documentation, 2012

En las siguientes figuras se puede observar los ángulos de cada parte anteriormente detalladas, con lo cual se puede complementar con el ángulo en el momento de trabajar con el robot físicamente o con la simulación en Choregraphe, o de igual manera tomando los valores con el robot real y guardar las posiciones para luego trabajar solamente con la simulación y poder ver el comportamiento en las distintas aplicaciones.

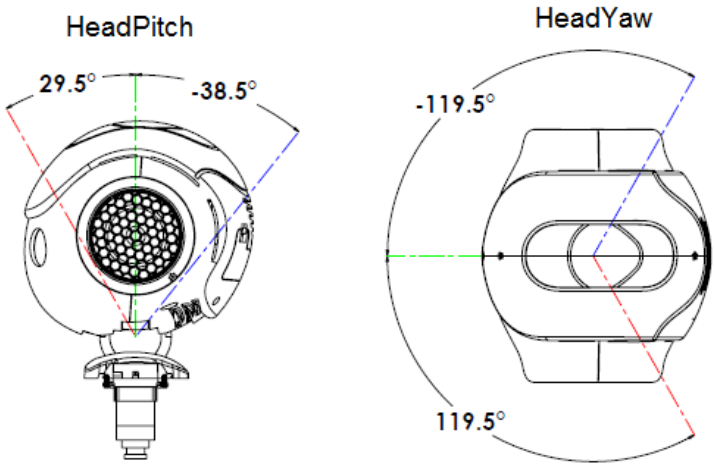


Figura 7-2: Ángulos de la cabeza del robot NAO

Fuente: Aldebaran documentation, 2012

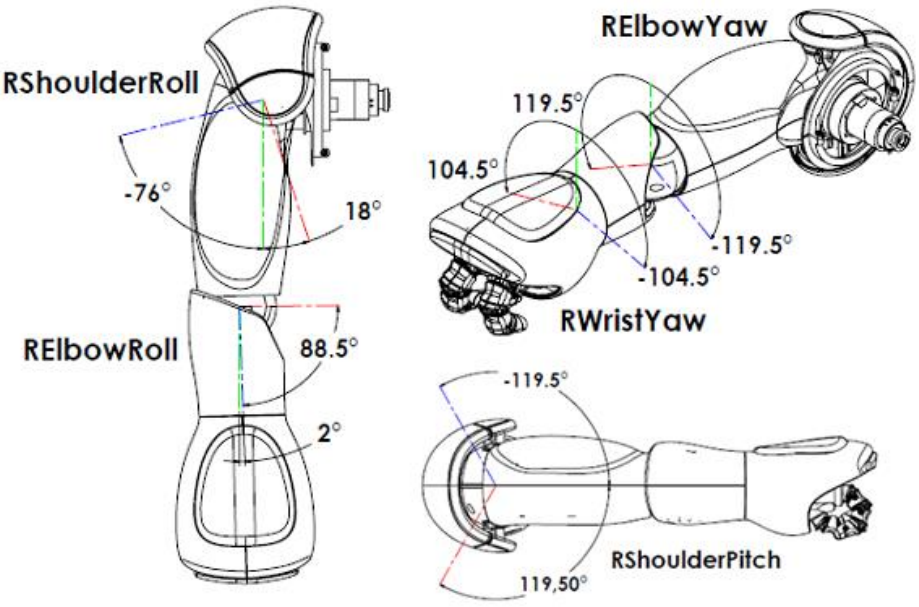


Figura 8-2: Ángulos de movimiento del brazo derecho robot NAO

Fuente: Aldebaran documentation, 2012

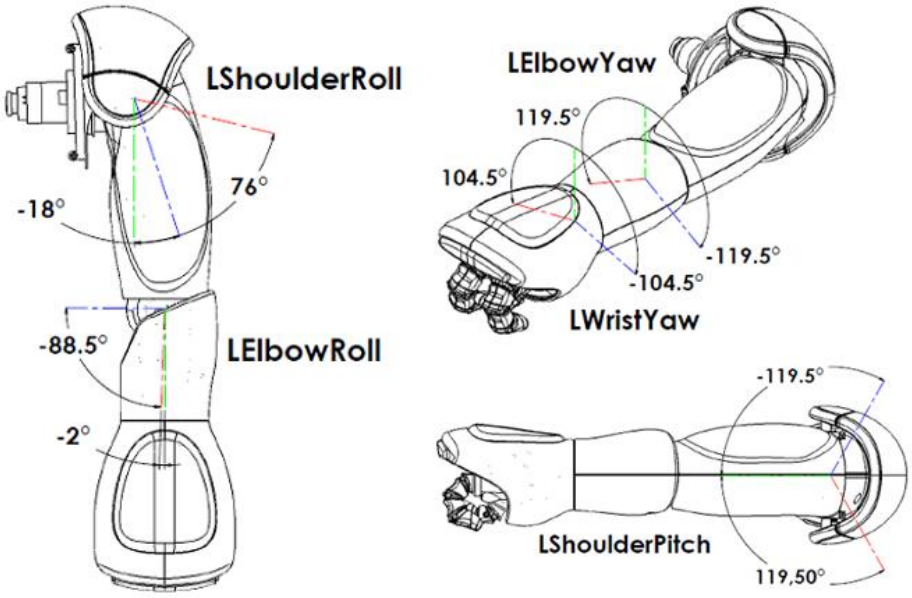


Figura 9-2: Ángulos de movimiento del brazo izquierdo robot NAO

Fuente: Aldebaran documentation, 2012

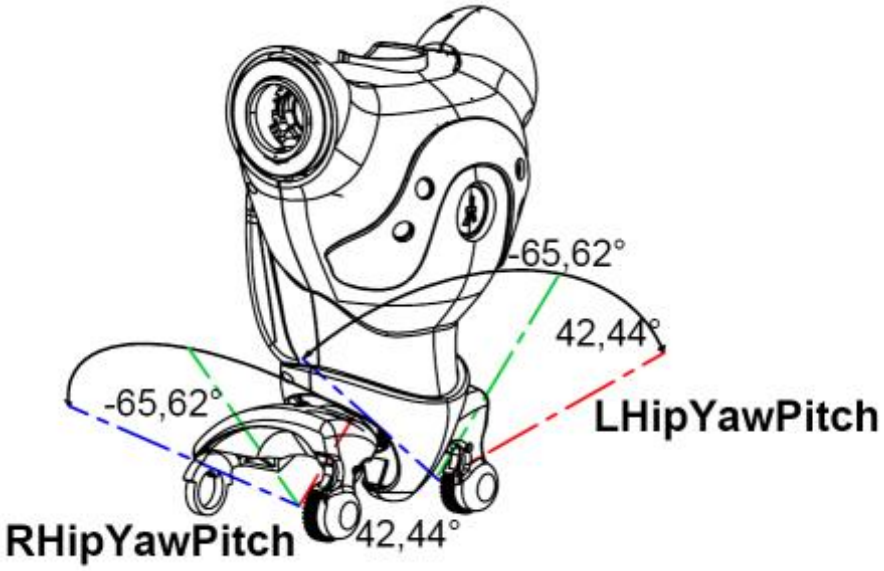


Figura 10-2: Ángulos de movimiento del torso robot NAO

Fuente: Aldebaran documentation, 2012

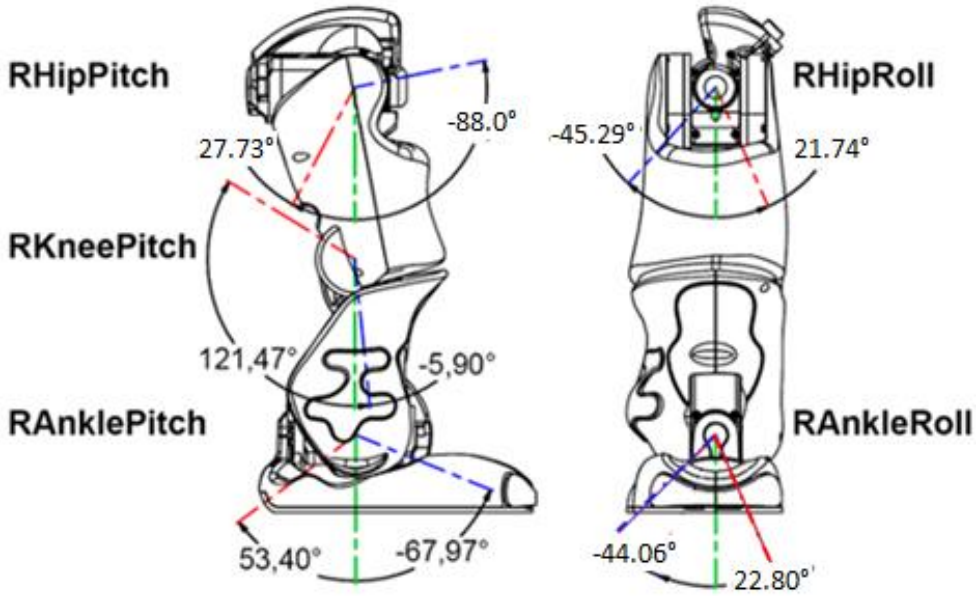


Figura 11-2: Ángulos de movimiento pierna derecha robot NAO

Fuente: Aldebaran documentation, 2012

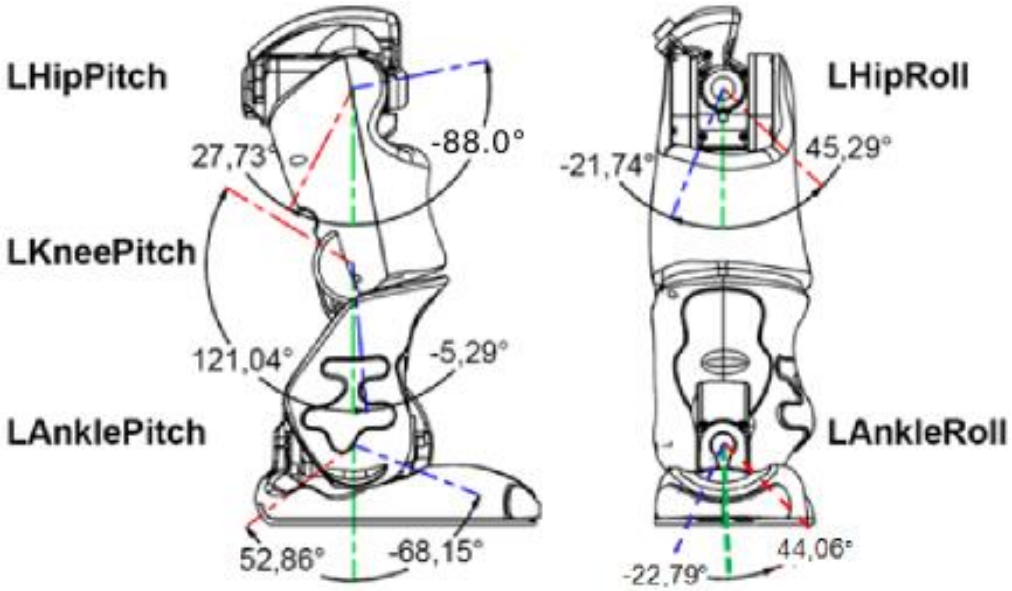


Figura 12-2: Ángulos de movimiento pierna izquierda robot NAO

Fuente: Aldebaran documentation, 2012

2.3. Cinemática del robot

A continuación, se detalla los movimientos de las articulaciones que componen al robot NAO, se puede observar en la Tabla 2-2.

Tabla 2-2: Detalle de articulaciones del robot NAO.

Cabeza	HeadPitch	Flexión/Extensión cuello
	HeadYaw	Rotación Cuello
Brazos (R/L Derecha/izquierda)	R/LShoulderPitch	Flexión/Extensión hombro
	R/LShoulderRoll	Abd/aducción hombro
	R/LElbowYaw	Rotación codo
	R/LElbowRoll	Flexión/Extensión codo
	R/LWristYaw	Rotación muñeca
	R/LHand	Abrir / Cerrar mano
Pelvis	HipYawPith	Articulación pélvica a 45°
Piernas (R/L Derecha/Izquierda)	R/LHipPith	Flexión/Extensión cadera
	R/LHipRoll	Abd/Aducción cadera
	R/LAnklePith	Flexión/Extensión tobillo
	R/LAnkleRoll	Abd/Aducción tobillo
	R/LKneePith	Flexión/Extensión rodilla

Fuente: Jesús Bueno Gómez, 2012

2.3.1 Matriz de rotación

Los movimientos del robot se detallan en base a la matriz de rotación, que es un modelo matemático del resultado del producto de rotaciones de los ejes x, y, z. (Aldebaran, 2017)

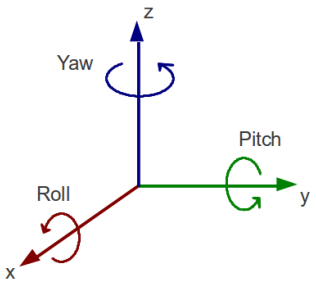


Figura 13-2: Movimientos de los ejes x, y, z

Fuente: Aldebaran documentation, 2012

La matriz resultante se forma de las ecuaciones.

$$R_{xyz} = R_{z,\emptyset} R_{y,\theta} R_{z,\gamma}$$

$$Rxyz = \begin{bmatrix} C_\theta & -S_\theta & 0 & 0 \\ S_\theta & C_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\theta & 0 & S_\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S_\theta & 0 & C_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_\gamma & -S_\gamma & 0 \\ 0 & S_\gamma & C_\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rxyz = \begin{bmatrix} C_\emptyset C_\theta & -S_\emptyset C_\gamma + C_\emptyset S_\theta S_\gamma & S_{\emptyset C_\gamma} + C_\emptyset S_\theta S_\gamma & 0 \\ S_\emptyset C_\theta & C_\emptyset C_\gamma + S_\emptyset S_\theta S_\gamma & -C_\theta S_\gamma + S_\emptyset S_\theta C_\gamma & 0 \\ -S_\theta & C_\theta S_\gamma & C_\theta C_\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

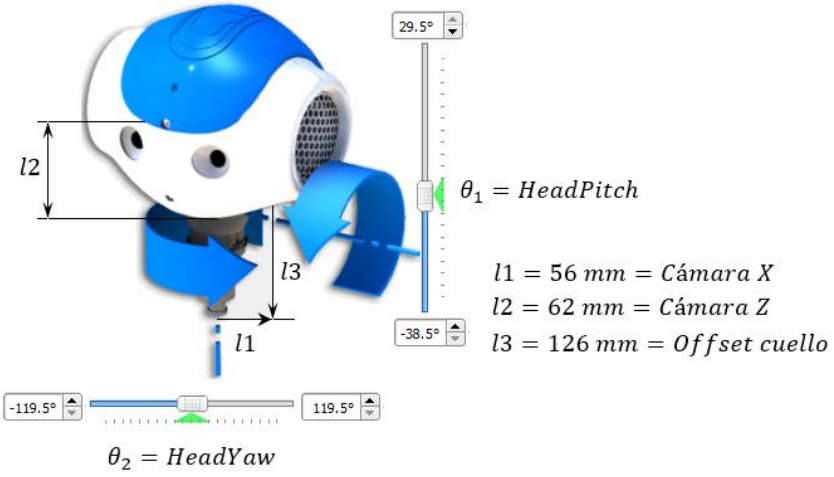


Figura 14-2: Grados de libertad cabeza del robot

Realizado por: CARRANCO, Carlos Alberto, 2017

A continuación, se muestra la ecuación donde esta aplicada para el movimiento de la cabeza con respecto al eje z. (Navas, Gabriel, Rivera, 2017)

$$\theta_1 = \left[\pm acos\left(\frac{px}{l2\cos(\theta_2 - {}^{\pi}/_{2}) - l1\sin(\theta_2 - {}^{\pi}/_{2})}\right)\right]$$

En la siguiente ecuación de muestra refiriéndose al movimiento de la cabeza en relación al eje x. (Kofinas, 2012)

$$\theta_2 = \left(\pi - \sin^{-1}\left(\frac{-pz + l3}{\sqrt{l1^2 + l2^2}}\right) - \tan^{-1}\left(\frac{l1}{l2}\right) + \frac{\pi}{2}\right)$$

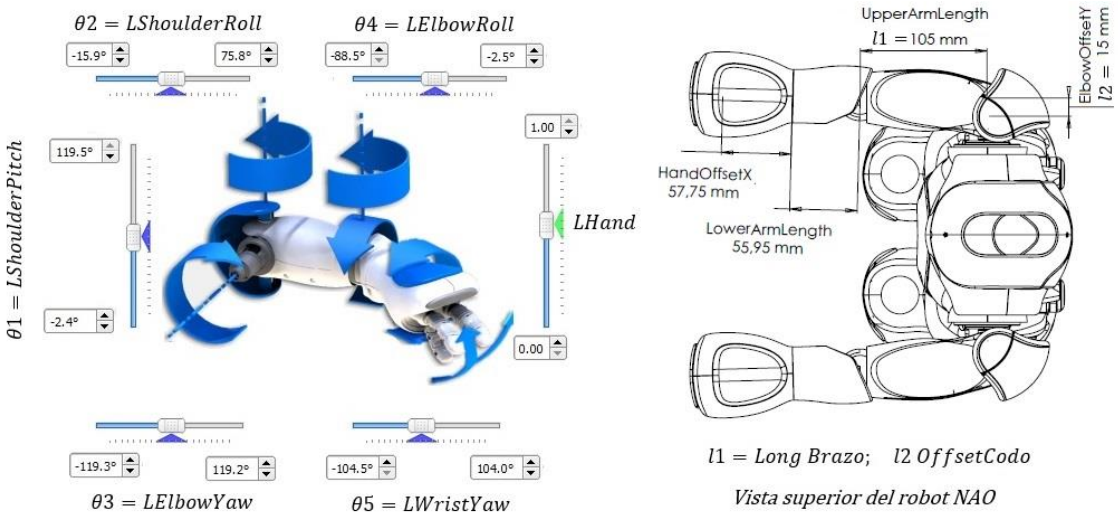


Figura 15-2: Rango movimiento brazo izquierdo (izquierda)

Realizado por: CARRANCO, Carlos Alberto, 2017

$$D = A^0_{Base} D^1_0 D^2_1 D^3_2 D^4_3 R_z({}^{\pi}/_{2}) A^{End}_4 \qquad \text{Ecuación 1.1}$$

$$D' = (A^0_{Base})^{-1} D \left(A^{End}_4\right)^{-1} \left(R_z({}^{\pi}/_{2})\right)^{-1} \qquad \text{Ecuación 1.2}$$

$$D" = (D')^{-1} \qquad \text{Ecuación 1.3}$$

$$D" = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A continuación se describe las ecuaciones de la matriz: (Kofinas, 2012)

$$r_{11} = \cos\theta_2 \sin\theta_1 \sin\theta_3 + \cos\theta_1(\cos\hat{\theta}_2 \cos\theta_3 \cos\theta_4 - \sin\hat{\theta}_2 \sin\theta_4)$$

$$r_{12} = \cos\theta_3 \cos\theta_4 \sin\hat{\theta}_2 + \cos\hat{\theta}_2 \sin\theta_4$$

$$r_{13} = \cos\hat{\theta}_2 \cos\theta_3 \cos\theta_4 \sin\theta_1 + \cos\theta_1 \cos\theta_4 \sin\theta_3 + \sin\theta_1 \sin\hat{\theta}_2 \sin\theta_4$$

$$r_{14} = \mathrm{L1}\cos\theta_3 \cos\theta_4 + L2 \sin\theta_4$$

$$r_{21} = -\sin\theta_1 \sin\theta_3 \sin\theta_4 - \cos\theta_1(\cos\theta_4 \sin\hat{\theta}_2 + \cos\hat{\theta}_2 \cos\theta_3 \sin\theta_4)$$

$$r_{22} = \cos\hat{\theta}_2 \cos\theta_4 - \cos\theta_3 \sin\hat{\theta}_2 \sin\theta_4$$

$$r_{23} = \cos\theta_4 \sin\theta_1 \sin\hat{\theta}_2 + (\cos\hat{\theta}_2 \cos\theta_3 \sin\theta_1 - \cos\theta_1 \sin\theta_3)\sin\theta_4$$

$$r_{24} = \mathrm{L2}\cos\theta_4 + L1 \cos\theta_3 \sin\theta_4$$

$$r_{31} = \cos\theta_3 \sin\theta_1 - \cos\theta_1 \cos\hat{\theta}_2 \sin\theta_3$$

$$r_{32} = -\sin\hat{\theta}_2 \sin\theta_3$$

$$r_{33} = \cos\theta_1 \cos\theta_3 + \cos\hat{\theta}_2 \sin\theta_1 \sin\theta_3$$

$$r_{34} = \mathrm{L1} \sin\theta_3$$

El parámetro $\hat{\theta}_2$ es el parámetro de θ Denavit-Hartenberg para la segunda juntura. El movimiento del hombro (cabeceo) se encuentra representado en la ecuación 1.1. (Kofinas, 2012)

$$\theta_3 = \begin{cases} \sin^{-1}\left(\frac{D"_{3,4}}{L1}\right) \\ \pi - \sin^{-1}\left(\frac{D"_{3,4}}{L1}\right) \end{cases}$$

La ecuación 1.2, hombro (alabeo).

$$D"_{14} = -L1 \cos\theta_3 \cos\theta_4 + L2 \sin\theta_4$$

$$\sin\theta_4 = \frac{D"_{14} + L1 \cos\theta_3 \cos\theta_3 \cos\theta_4}{L2}$$

$$D"_{24} = L2 \cos\theta_4 + L2 \cos\theta_3 \frac{D"_{14} + L1 \cos\theta_3 \cos\theta_4}{L2}$$

$$\cos\theta_4(L2^2 + L1^2 cos^2\theta_3) = L2D"_{24} - L1D"_{14} \cos\theta_3$$

$$\theta_4 = \pm\cos^{-1}\left(\frac{L2D"_{24} - L1D"_{14} \cos\theta_3}{L2^2 + L1^2 cos^2\theta_3}\right)$$

Para el desplazamiento de la muñeca, se asigna la ecuación 1.3

$$D''' = D'^{(D_2^3)^{-1}(D_3^4)^{-1}}$$

$$\hat{\theta}_2 = \tan^{-1}\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right)$$

$$\theta_2 = \hat{\theta}_2 - {}^{\pi}/_{2}$$

$$\theta_2 = \tan^{-1}\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right) - \frac{\pi}{2}$$

Los movimientos que puede realizar el codo son en guiñada y alabeo, descritos en las ecuaciones 2.10 y 2.11 respectivamente.

$$\theta_1 = \tan^{-1}\left(\frac{D'''_{1,3}}{D'''_{3,3}}\right)$$

$$\theta_1 = \tan^{-1}\left(\frac{D'''_{3,1}}{D'''_{3,3}}\right)$$

Brazo derecho

El brazo derecho e izquierdo son simétricos, por lo tanto las ecuaciones para el brazo derecho solo cambian en las junturas "*roll" y* las distancias a lo largo del eje *y* (RShoulderRoll, RElbowRoll). (Navas, Gabriel, Rivera, 2017)

La diferencia es que ElbowOffsetY ahora es negativo, entonces L1 = - ElbowOffsetY.

Las ecuaciones para la cinemática e inversa del brazo derecho son: (Kofinas, 2012)

$$D' = (A^0_{Base})^{-1} D \left(A^{End}_4\right)^{-1} \left(R_z\left(\pi/_2\right)\right)^{-1}$$

$$D'' = (D')^{-1}$$

$$\theta_3 = \begin{cases} \sin^{-1}\left(\frac{D"_{3,4}}{L1}\right) \\ \pi - \sin^{-1}\left(\frac{D"_{3,4}}{L1}\right) \end{cases}$$

$$\theta_4 = \pm \cos^{-1}\left(\frac{L2D''_{2,4} - L1D''_{1,4}\cos\theta_3}{L2^2 + L1^2 cos^2\,\theta_3}\right)$$

$$D''' = D'^{(D^3_2)^{-1}(D^4_3)^{-1}}$$

$$\theta_2 = \tan^{-1}\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right) - \frac{\pi}{2}$$

$$\theta_1 = \tan^{-1}\left(\frac{D'''_{1,3}}{D'''_{3,3}}\right)$$

2.3.2 Piernas del robot NAO

El esquema cinemático de las piernas del robot NAO se muestra en la **Figura 16**-2. Cada pierna posee seis articulaciones, dos articulaciones de la cadera son controladas por un solo motor.(E Fierro, Pamanes, A Santibáñez, Ruiz, & Ollervides, 2016)

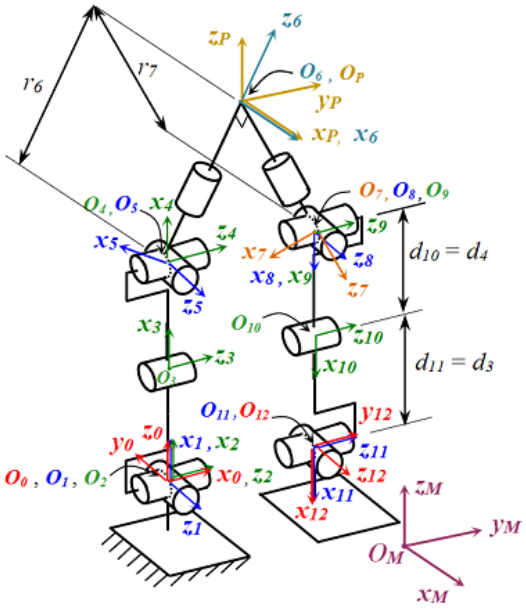


Figura 16-2: Esquema cinemático de las piernas del robot

Fuente: Alfonso Pamanes, 2017

2.3.3 Pierna izquierda.

En la Figura 17-2, se observan los siguientes movimientos:

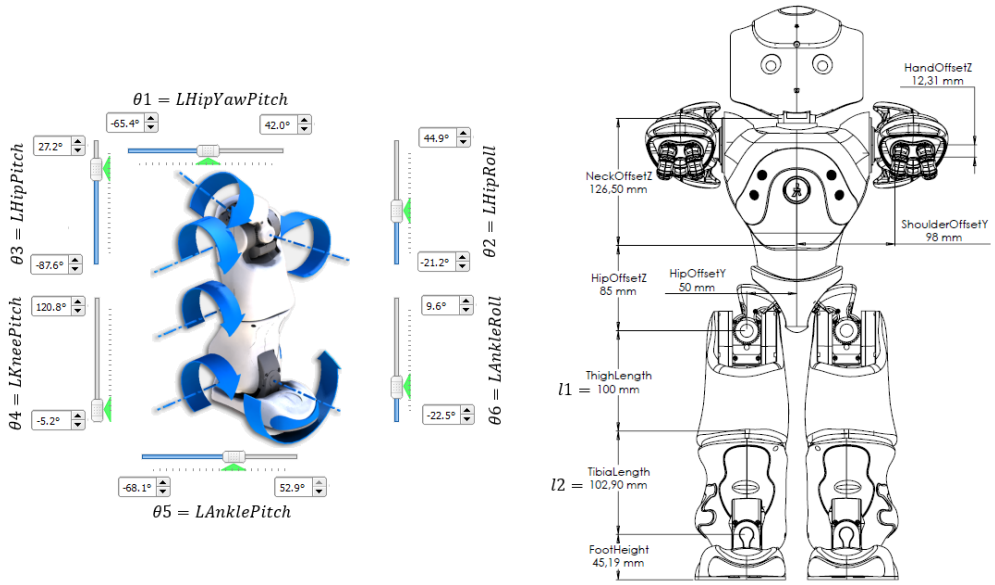


Figura 17-2: Rango movimiento pierna izquierda (izquierda), dimensiones vista frontal robot (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

A partir de estos movimientos se puede construir la ecuación para un sistema no lineal: (Navas, Gabriel, Rivera, 2017)

$$D = A_{Base}^{0} D_0^1 D_1^2 D_2^3 D_3^4 D_4^5 D_5^6 R_z(\pi) R_y\left(-\pi/2\right) A_6^{End}$$

$$\widehat{D} = (A_{Base}^{0})^{-1} D \left(A_6^{Final}\right)^{-1}$$

$$\widehat{D} = R_x\left(\pi/4\right)\widehat{D}$$

$$D' = \left(\widetilde{D}\right)^{-1}$$

Las ecuaciones de traslación del sistema resueltas se obtienen de la siguiente forma:

$$D' = l2 \sin\theta_5 - L1 \sin(\theta_4 + \theta_5)$$

$$D'_{(2,4)} = (L2 \cos\theta_5 + L1 \cos(\theta_4 + \theta_5)) \sin\theta_6$$

$$D'_{(3,4)} = (L2 \cos\theta_5 + L1 \cos(\theta_4 + \theta_5)) \cos\theta_6$$

En las ecuaciones anteriores se observa las variables L1 y L2 que representan la longitud del muslo y la tibia respectivamente. Para calcular la distancia del triángulo formado desde la base de la pierna hasta el final de la misma se utilizará la siguiente ecuación. (Navas, Gabriel, Rivera, 2017)

$$d = \sqrt{(s_x - p'_x)^2 + \left(s_y - p'_y\right)^2 + (s_z - p'_z)^2}$$

Donde (s_x, s_y, s_z) = (0, 0, 0) es el nuevo origen y (p'_x, p'_y, p'_z) = ($D'_{(1,4)}$, $D'_{(2,4)}$, $D'_{(3,4)}$) es la posición del nuevo punto final. Usando la ley de cosenos se obtiene el ángulo interno y de la tibia: (Navas, Gabriel, Rivera, 2017)

$$\theta'_4 = \cos^{-1}\left(\frac{L1^2 + L2^2 - d^2}{2L1L2}\right)$$

$$\theta''_4 = \pi - \theta'_4$$

$$\theta_4 = \pm\,\theta''_4$$

$$\theta_4 = \pm\left(\pi - \cos^{-1}\left(\frac{L1^2 + L2^2 - \|\bar{0} - \bar{p}\|^2}{2L1L2}\right)\right)$$

Para encontrar el ángulo θ_6 se usará las ecuaciones de traslación $D'_{(2,4)}$, $D'_{(3,4)}$:

$$\frac{D'_{(2,4)}}{D'_{(2,4)}} = \frac{p'_y}{p'_z}$$

$$\frac{(L2 \cos\theta_5 + L1 \cos(\theta_4 + \theta_5)) \sin\theta_6}{(L2 \cos\theta_5 + L1 \cos(\theta_4 + \theta_6)) \cos\theta_6} = \frac{p'_y}{p'_z}$$

$$\theta_6 = \tan^{-1}\left(\frac{p'_y}{p'_z}\right) \; si \; \left((L2 \cos\theta_5 + L1 \cos(\theta_4 + \theta_5))\right) \neq 0$$

Calculo del ángulo del tobillo θ_5:

$$\widetilde{D'} = \widetilde{D}\left(D_5^6 R_z(\pi) R_y\left(-\pi/2\right)\right)^{-1}$$

$$D'' = \left(\widetilde{D'}\right)^{-1}$$

La nueva posición final sería (p''_x, p''_y, p''_z) = ($D'_{(1,4)}$, $D'_{(2,4)}$, $D'_{(3,4)}$), siendo así las nuevas ecuaciones de traslación: (Navas, Gabriel, Rivera, 2017)

$$D''_{(1,4)} = L2\cos\theta_5 + L1(\cos\theta_5\cos\theta_4 - \sin\theta_5\sin\theta_4)$$

$$D''_{(2,4)} = -L2\sin\theta_5 - L1(\sin\theta_5\cos\theta_4 + \cos\theta_5\sin\theta_4)$$

$$D_{(3,4)} = 0$$

Teniendo en cuenta que:

$$D''_{(1.4)} = p''_X$$

$$(L2 + L1\cos\theta_4)\cos\theta_5 = p''_x + L1\sin\theta_5\sin\theta_4$$

$$\cos\theta_5 = \frac{p''_x + L1\sin\theta_5\sin\theta_4}{L2 + L1\cos\theta_4} \qquad si\ L2 + L1\cos\theta_4 \neq 0$$

$$\sin\theta_5(-L2 - L1\cos\theta_4) - L1\cos\theta_5\sin\theta_4 = p''_y$$

$$\sin\theta_5(-L2 - L1\cos\theta_4) - L1\frac{p''_x + L1\sin\theta_5\sin\theta_4}{L2 + L1\cos\theta_4}\sin\theta_4 = p''_y$$

$$-\sin\theta_5(L2 + L1\cos\theta_4) - \frac{L1\,p''_x\sin\theta_4}{L2 + L1\cos\theta_4} - \frac{L1^2\sin\theta_5\sin^2\theta_4}{L2 + L1\cos\theta_4} = p''_y$$

$$-\sin\theta_5(L2 + L1\cos\theta_4) - L1^2\sin\theta_5\sin^2\theta_4 = p''_y(L2 + L1\cos\theta_4) + L1\,p''_x\sin\theta_4$$

$$\theta_5 = \sin^{-1}\left(\frac{p''_y(L2 + L1\cos\theta_4) + L1\,p''_x\sin\theta_4}{L1^2\sin^2\theta_4 + (L2 + L1\cos\theta_4)^2}\right)$$

La división es siempre factible, porque $L1^2\sin^2\theta_4 + (L2 + L1\cos\theta_4)^2$ es mayor que cero para cualquier valor de θ_4. (Navas, Gabriel, Rivera, 2017)

Finalmente, la deducción de las ecuaciones para los ángulos $\theta_3, \theta_2\ y\ \theta_1$ a partir de las ecuaciones de traslación obtenidas en D'''. (Navas, Gabriel, Rivera, 2017)

$$D''' = \widetilde{D'}\left(D_3^4 D_4^5\right)^{-1}$$

A continuación, se describen las ecuaciones de traslación obtenidas de D'''.

$$r_{11=}\cos\hat{\theta}_1\cos\hat{\theta}_2\cos\theta_4 - \sin\hat{\theta}_1\sin\theta_3$$

$$r_{12} = -\cos\theta_3\sin\hat{\theta}_1 - \cos\hat{\theta}_1\cos\hat{\theta}_2\sin\theta_3$$

$$r_{13} = \cos\hat{\theta}_1\sin\hat{\theta}_2$$

$$r_{21} = -\cos\theta_3\sin\hat{\theta}_2$$

$$r_{22} = \sin\hat{\theta}_2\sin\theta_3$$

$$r_{23} = \cos\hat{\theta}_2$$

$$r_{31} = -\cos\hat{\theta}_2\cos\theta_3\sin\hat{\theta}_1 - \cos\hat{\theta}_1\sin\theta_3$$

$$r_{32} = -\cos\hat{\theta}_1\cos\theta_3 + \cos\hat{\theta}_2\sin\hat{\theta}_1\sin\theta_3$$

$$r_{33} = -\sin\hat{\theta}_1\sin\hat{\theta}_2$$

$$\hat{\theta}_2 = \cos^{-1}D'''_{(2,3)}$$

$$\theta_2 = \hat{\theta}_2 - {}^{\pi}/_{4}$$

$$\theta_2 = \cos^{-1} D'''_{(2,3)} - {}^{\pi}/_{4}$$

$$\theta_3 = \sin^{-1}\left(\frac{D'''_{(2,2)}}{\sin\left(\theta_2 + {}^{\pi}/_{4}\right)}\right)$$

$$\hat{\theta}_1 = \cos^{-1}\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + {}^{\pi}/_{4}\right)}\right)$$

$$\theta_1 = \hat{\theta}_1 + {}^{\pi}/_{2}$$

$$\theta_1 = \cos^{-1}\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + {}^{\pi}/_{4}\right)}\right) + \frac{\pi}{2}$$

2.3.4 Pierna Derecha.

Como los movimientos de la pierna izquierda son iguales en la pierna derecha:

$$D_{base}^{pierna\ izq.}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = M D_{base}^{pierna\ der.}(\theta_1, -\theta_2, \theta_3, \theta_4, \theta_5, -\theta_6) M$$

Las ecuaciones para los movimientos de la pierna derecha, se muestra a continuación:

$$\theta_1 = \cos^{-1}\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + {}^{\pi}/_{4}\right)}\right) + \frac{\pi}{2}$$

$$\theta_2 = \cos^{-1} D'''_{(2,3)} - {}^{\pi}/_{4}$$

$$\theta_1 = \sin^{-1}\left(\frac{D'''_{(2,2)}}{\sin\left(\theta_2 + {}^{\pi}/_{4}\right)}\right)$$

$$\theta_4 = \pm\left(\pi - \cos^{-1}\left(\frac{L1^2 + L2^2 - \|\bar{0} - \bar{p}\|^2}{2L1L2}\right)\right)$$

$$\theta_5 = \sin^{-1}\left(-\frac{p''_y(L2 + L1\cos\theta_4) + L1p''_x \sin\theta_4}{L1^2 \sin^2\theta_4 + (L2 + L1\cos\theta_4)}\right)$$

$$\theta_6 = \begin{cases} \tan^{-1}\left(\dfrac{D'_{(2,4)}}{D'_{(3,4)}}\right) & si\ \left((L2\cos\theta_5 + L1\cos(\theta_4 + \theta_5))\right) \neq 0 \end{cases}$$

2.3.5 Movimientos robot NAO

Aldebaran Robotics presenta dos formas diferentes para la programación de la plataforma robótica NAO, la primera es mediante el uso del software Choregraphe y la segunda es mediante un lenguaje de programación, el API para acceder al control de movimiento y uso de sensores se encuentra disponible en los lenguajes: C++, Python, .Net, Java Matlab y Urbi; de estos lenguajes únicamente C++ y Python permiten ser ejecutados directamente en el robot NAO, el resto de ellos únicamente permite el control remoto del robot a través de una conexión por red. (Ocampo, 2014)

2.4. Visión artificial

Uno de los sentidos más importantes de los seres humanos es la visión. Ésta es empleada para obtener la información visual del entorno físico. Según Aristóteles, "Visión es saber que hay y donde mediante la vista". De hecho, se calcula que más de 70% de las tareas del cerebro son empleadas en el análisis de la información visual. El refrán popular de "Una imagen vale más que mil palabras" tiene mucho que ver con los aspectos cognitivos de la especie humana. (Santillan, 2014).

La visión artificial consiste en la captación de imágenes en línea mediante cámaras CCD y su posterior tratamiento a través de técnicas de procesamiento avanzadas, permitiendo así poder intervenir sobre un proceso (modificación de variables del mismo) o producto (detección de unidades defectuosas), para el control de calidad y seguridad de toda la producción. (Artificial, 2012)

Un sistema de visión artificial:

- Capta una imagen de un objeto real
- La convierte en formato digital
    - La procesa mediante un ordenador
    - Obtiene unos resultados del proceso.(Artificial, 2012)

2.4.1 Visión con el robot NAO

El robot nao dispone de dos cámaras de video ubicadas en la parte frontal de la cabeza Figura 18-2, se hace uso de la cámara superior la cual va adquirir la imagen del entorno por donde el robot realizará la navegación.

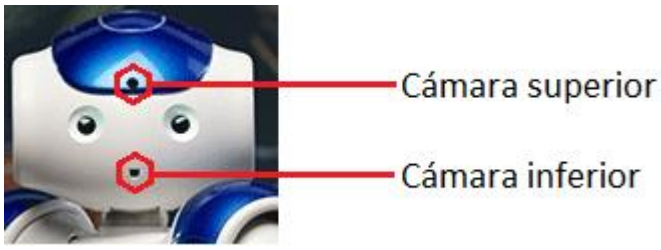


Figura 18-2: Ubicación de cámaras en el robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

Estas cámaras son de alta definición, con resolución de 1280 x 960 píxeles en espacio de color YUV422 con un máximo de 30 cuadros por segundo, las características de los ángulos se pueden observar en la Figura 19-2.

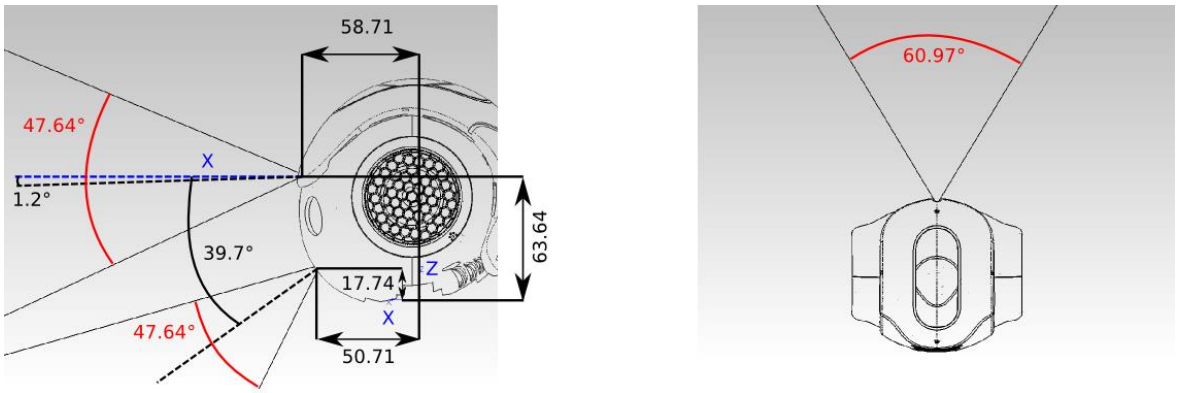


Figura 19-2: Ángulo de visión (izquierda), ángulo de visión "cámaras robot NAO"

Fuente: Aldebaran documentation, 2012

Otro de los factores bastante importantes en la adquisición de la imagen con el robot NAO es la resolución, se realiza varias pruebas con las distintas resoluciones y la que más se ajusta al proyecto es la de 320 x 240px, debido al tiempo de procesamiento de la imagen, en la Tabla 3-2 se detalla los parámetros que se puede trabajar con la cámara superior del robot NAO.

Tabla 3-2: Resoluciones soportadas cámara robot NAO y cuadros por segundo

Nombre del ID del parámetro	Valor de ID	Descripción	Soporte cuadros por segundo
AL:: kQQQQVGA	8	Imagen de 40 * 30px	from 1 to 30 fps
AL:: kQQQVGA	7	Imagen de 80 * 60px	from 1 to 30 fps
AL:: kQQVGA	0	Imagen de 160 * 120px	from 1 to 30 fps
AL:: kQVGA	1	Imagen de 320 * 240px	from 1 to 30 fps
AL:: kVGA	2	Imagen de 640 * 480px	from 1 to 30 fps
AL:: k4VGA	3	Imagen de 1280 * 960px	from 1 to 30 fps

Fuente: Aldebaran documentation, 2012

2.4.2 Módulo ALVideoDevice

El módulo ALVideoDevice es de los más básicos dentro del conjunto de módulos de visión, sin embargo, es el de mayor importancia, ya que provee de las imágenes para luego ser procesadas, este módulo por lo tanto es de más bajo nivel, pues se encarga de solicitar la imagen de la cámara directamente. Este módulo provee en todo momento a otros módulos con imágenes si fuera necesario, ajustando las configuraciones mínimas de todos ellos, de tal forma que usen la cantidad de recursos necesariamente posibles. Como el espacio de color nativo de las cámaras es YUV422, el mejor rendimiento de procesamiento de imágenes se obtiene cuando se utiliza este mismo espacio. Para imágenes de alta definición, la velocidad que puede procesar el CPU Atom se encuentra alrededor de los cinco cuadros por segundo. Para solicitarle imágenes al módulo ALVideoDevice, se le manda una solicitud para subscribirse al módulo, para esto se le deben enviar los siguientes cuatro parámetros: (Ocampo, 2014)

- Nombre: Se utiliza como identificador, es el nombre con el que el módulo ALVideoDevice va a manejar la suscripción.
- Resolución: El tamaño de la imagen que se desea recibir, procesar imágenes de mayor tamaño resulta en un mayor uso de recursos del procesador. Las resoluciones disponibles se pueden visualizar en la Tabla 3-2.
- Espacio de color: Se elige el espacio de color en el que se trabajara la imagen, si el espacio solicitado es distinto del nativo (YUV422), el módulo ALVideoDevice se encarga de hacer la trasformación, lo que utiliza recursos de procesador. Los principales espacios son los siguientes: – YUV422 (formato nativo) – YUV (24 bits) – Y (8 bits) – RGB (24 bits) – BGR (24 bits) – HSY (24 bits). (Ocampo, 2014)
- Cuadros por segundo: Se selecciona la velocidad de cuadros por segundo que se requieren, con un máximo de 30 fps.(Ocampo, 2014)

2.4.3 Open CV con el robot NAO

Para el procesamiento de las imágenes obtenidas por la cámara del robot NAO además de trabajar con el módulo ALVideoDevice, trabajará en conjunto con OpenCV, siendo una biblioteca open source para C/C++ en el procesamiento de imágenes y visión artificial, desarrollada inicialmente por Intel, multiplataforma, realiza operaciones básicas con matrices y procesamiento de imágenes. Permite la visualización de datos y extraer información de imágenes y videos.(Furfaro, 2010)

NAOqi soporta la versión 2.3.1 de OpenCV tanto para compilación como para compilación cruzada, sin embargo, dentro del NAOqi no se incluyen todas las librerías de OpenCV, por lo que la programación se la realiza en Python realizando la comunicación con NAOqi para el procesamiento de las imágenes.(Ocampo, 2014)

2.5. Python

Python es un lenguaje de programación orientado a objetos, creado por Guido van Rossum teniendo un gran crecimiento en los últimos años, debido principalmente a su adopción por parte de Google.(Cordoba, 2011)

Está siendo adoptado por científicos de distintas disciplinas (astrónomos, biólogos, físicos y científicos sociales) siendo así una alternativa gratuita a Matlab basada en software libre. Ya que Python es un lenguaje de programación fácil de aprender con una sintaxis sencilla e intuitiva. Dispone de una documentación de soporte, librerías científicas, capacidades para graficar en dos y tres dimensiones, así como un amplio repertorio de librerías libres para realizar tareas como creación de sitios web, interacción con bases de datos, creación de interfaces gráficas multi-plataforma, entre otros). (Cordoba, 2011)

En la aplicación de este proyecto se elige la programación con Python porque es un lenguaje relacionado con Choregraphe ya que ambos son intuitivos y sencillos de aprender.

2.6. Spyder

Es un Entorno de Desarrollo Integrado (o IDE por sus siglas en inglés), creado para hacer más sencilla y amigable la programación de simulaciones científicas. El desarrollo de su interfaz es similar a *Matlab* incorporando características especiales, facilitando el desarrollo de programas con Python.(Cordoba, 2011)

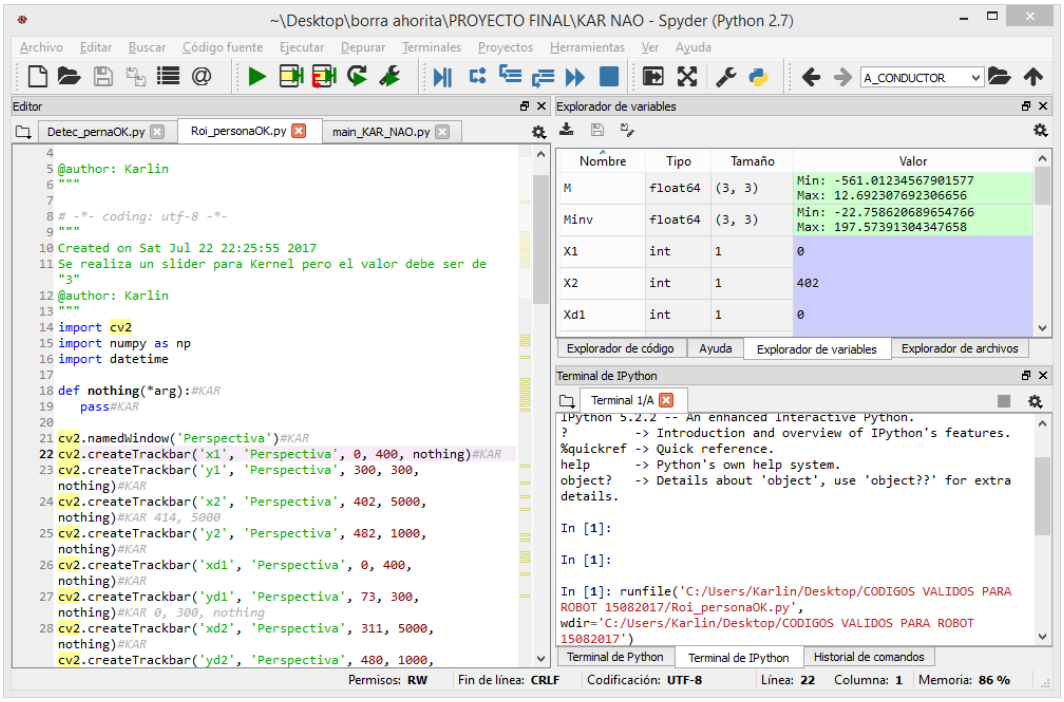


Figura 20-2: Interface de Spyder

Realizado por: CARRANCO, Carlos Alberto, 2017

Se elige el IDE de Spyder Figura 20-2, ya que es una aplicación similar a la de Matlab, donde se puede visualizar en el Explorador de las variables sus datos obtenidos al ejecutar algún script y si algún dato esta erróneo, fuera de rango o verificar alguna variable se puede visualizar sin ningún inconveniente.

2.7. Choregraphe

Choregraphe es un software que puede ser utilizado por los sistemas operativos Linux, Windows y Mac, diseñado y desarrollado por Aldebaran Robotics permitiendo a los usuarios la creación de nuevos movimientos e interacciones de una forma rápida y sencilla. Con Choregraphe la programación es gráfica y pseudocódigo con Python, creándose secuencias definidas para el uso del usuario o creadas dependiendo de lo que se requiera.

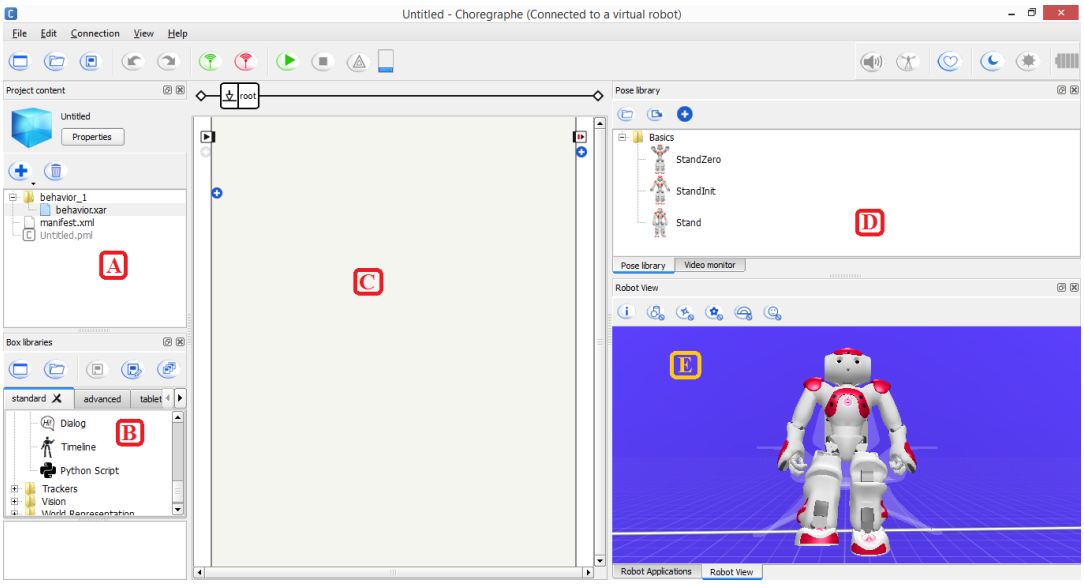


Figura 21-2: Paneles de barra de herramientas en Choregraphe

Realizado por: CARRANCO, Carlos Alberto, 2017

Tabla 4-2: Detalle de paneles Choregraphe

Parte	Panel
A	Contenido del proyecto
B	Cajas de librerías
C	Diagrama de Flujo (Área de programación)
D	Librería de posiciones y monitor de video
E	Robot virtual y aplicaciones del robot.

Fuente: Aldebaran documentation, 2012

Choregraphe en su interfaz, incluye un simulador del robot en el costado derecho, sin embargo, el simulador presenta un modelo sencillo del robot, donde no toma en cuenta las fuerzas externas al ejecutar los movimientos.(Ocampo, 2014)

En el proyecto se toma como referencia las posiciones y movimientos para el robot NAO, en el momento que tenga que realizar la acción respectiva, pero todos los datos son tomados con la ayuda de la caja de bloque Timeline, ubicada en el Box libraries otra de las opciones, es realizando clic derecho en el área de trabajo, tal como se muestra en la Figura 22-2.

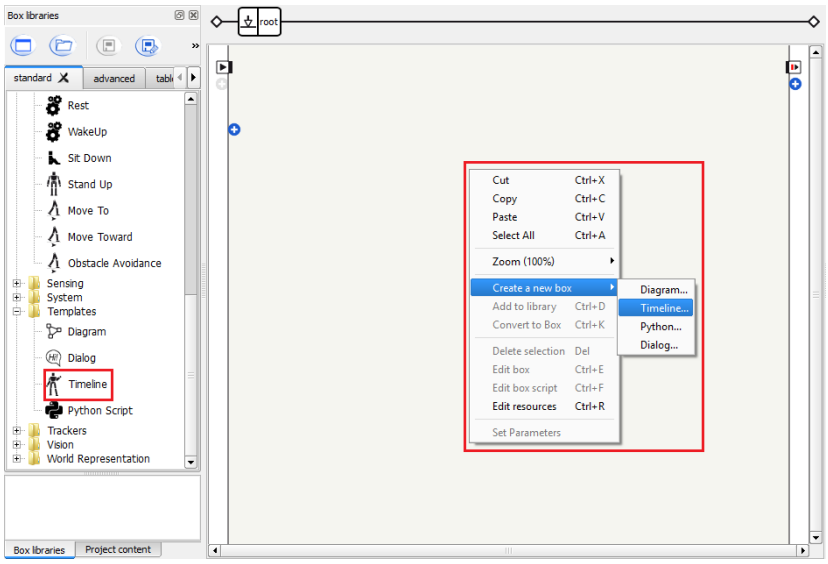


Figura 22-2: Uso de Timeline para obtener las posiciones del robot

Realizado por: CARRANCO, Carlos Alberto, 2017

Este proceso es de gran ayuda ya que para la programación solamente se ingresa los datos de las articulaciones y se hace el llamado dependiendo del evento que se desee que el robot NAO realice, por ejemplo, que gire el volante a la izquierda o derecha o lo mantenga recto, otro caso sería que, al detectar una persona con la cámara, deje de accionar el pedal para no traccionar con el vehículo, si no detecta puede seguir con el desplazamiento.

Más adelante se explicará el proceso de accionamiento, en el momento que se comience a procesar la imagen con la cámara del robot NAO.

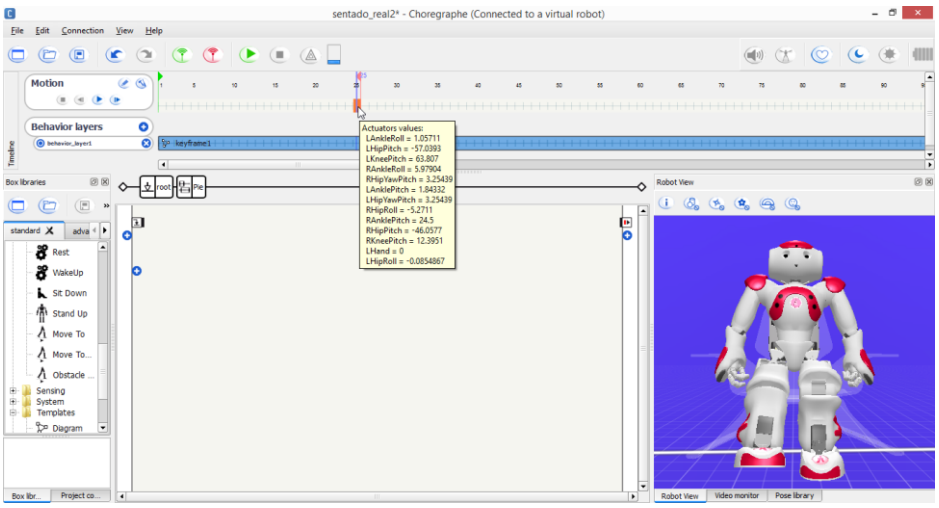


Figura 23-2: Datos obtenidos en Timeline capturado de cada articulación

Realizado por: CARRANCO, Carlos Alberto, 2017

CAPÍTULO III

3. Metodología

En el presente trabajo de investigación, la técnica de visión artificial es un factor importante en la adquisición de las imágenes mediante la cámara del robot humanoide NAO, para realizar el procesamiento y poder obtener resultados favorables en los movimientos en brazos y pie del robot en el control del coche eléctrico.

Para la identificación de los límites de un carril se hace la adaptación del código de (Priyanka Dwivedi, 2017), donde se realiza las pruebas respectivas con el robot sentado sobre un coche eléctrico en tiempo real, al igual que el uso de las librerías de OpenCV en la identificación de personas para realizar la acción respectiva en los controles del coche. La idea es sincronizar la fase de procesamiento de imagen, así como los movimientos del robot NAO hacia los controles del coche.

Para el desarrollo del proyecto es indispensable definir las siguientes secciones con la respectiva descripción de cada una para luego tratar cada proceso que se lleva a cabo para el resultado final.

a. ROBOT NAO
    i. El robot humanoide NAO debe estar en la posición sentado sobre una plataforma móvil (vehículo eléctrico para niños), tal que pueda alcanzar con su pie derecho el pedal de accionamiento.
    ii. Posición de los brazos y manos sobre el volante, para poder realizar el accionamiento tanto en rectas, izquierda o derecha.
    iii. Posición de la cabeza del robot para que detecte las líneas.
    iv. Conexión con el cable Ethernet conectado a una laptop.
    v. Energía suficiente para realizar las respectivas pruebas.
b. COMPUTADOR
    i. La aplicación se ejecutó desde un computador con los siguientes programas y librerías actualizadas:
        - Python versión 2.7.5 32 bits, Qt 4.8.7
        - Python 2.7 numpy 1.12.1
        - Python 2.7 matplolib 1.3.0
        - Python v2.7(x32), GPL v4.11.4 on Windows
        - Python 2.7 PIL versión 4.2.1
        - Python 2.7 pynaoqi 2.1.4.13
        - Open CV (versión 2.4.9)

- Choregraphe y Monitor 2.1.4.13 (Software que viene para instalarlo con el robot NAO)
- IDLE Spyder 3.1.3.

ii. Al momento de ejecutar se deberá tener en cuenta las siguientes configuraciones con el robot NAO:

- Dirección IP y Puerto del robot.
- Resolución de la cámara.

c. COCHE ELÉCTRICO

i. Volante debe ser tipo T con medidas específicas para que el robot pueda manipular.
ii. El pedal para acelerar debe ser de accionamiento suave (poseer un final de carrera de accionamiento leve).
iii. Desplazamiento con velocidad baja, media y alta, controlado con una tarjeta Arduino mega 256.
iv. L298N driver para el control del motor del coche.
v. El coche con dirección sin accionamiento eléctrico ya que causa que el volante sea duro lo cual dificulta el control para realizar movimientos por el robot.
vi. Un soporte para laptop.
vii. Cinturón para sujeción del robot NAO.
viii. Energía suficiente para realizar las respectivas pruebas.

3.1. Robot NAO

En la manipulación con el robot NAO, que ejercerá sobre el volante y el pedal; se detalla la forma de obtener los movimientos y relacionarlos con la programación del sistema de visión.

Cabe indicar que las posiciones fueron tomadas con el robot físico, con la ayuda del software Choregraphe y el robot virtual para lograr capturar las posiciones e implementarlas en la manipulación del volante.

3.1.1. Posición sentado del robot NAO

Se ubica al robot en el asiento del vehículo y se procede a capturar las posiciones, con respecto a las articulaciones que influyen en el robot, en la Figura 1-3 se puede observar la posición final.

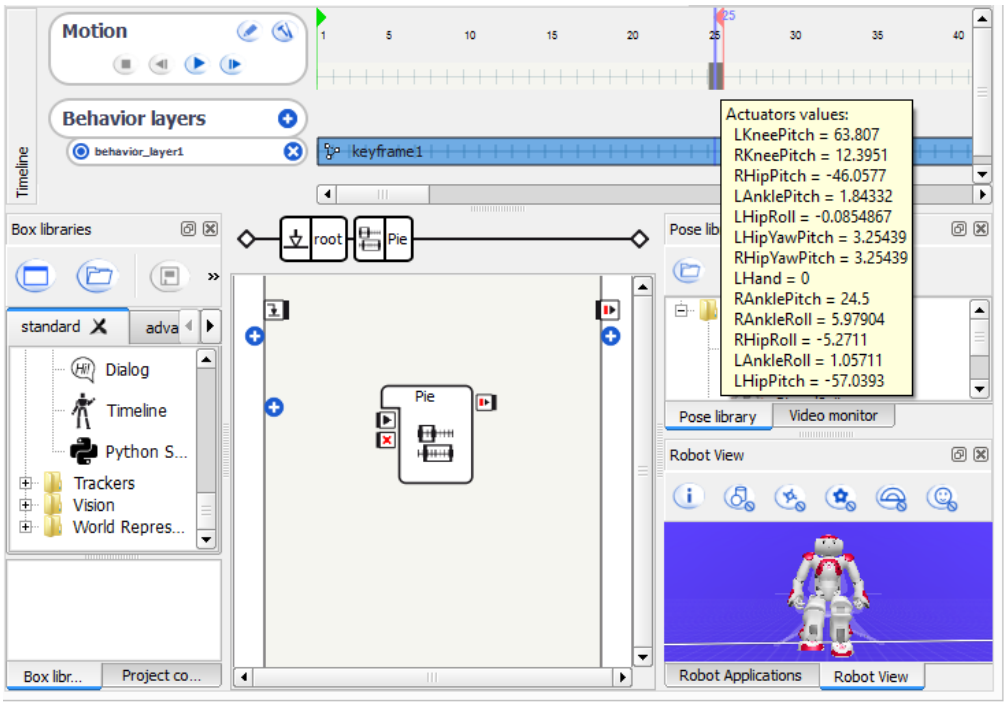


Figura 1-3: Posición sentado en Choregraphe

Fuente: CARRANCO, Carlos Alberto, 2017

A continuación, se detalla los pasos a seguir.

1. En el panel de Box libraries se ubica al Timeline, se arrastra al panel de programación.
2. Doble clic en Box Timeline y se abre una ventana de Timeline.
3. Para proceder a capturar las posiciones respectivas de cada articulación se debe seleccionar una parte del cuerpo del robot Nao: sea brazo (izquierdo o derecho), pierna (izquierda o derecha) y cabeza para proceder a manipular al robot y así queden las posiciones correctas para: tomar el volante, accione el acelerador y pueda obtener la visualización correcta de las líneas de la pista.

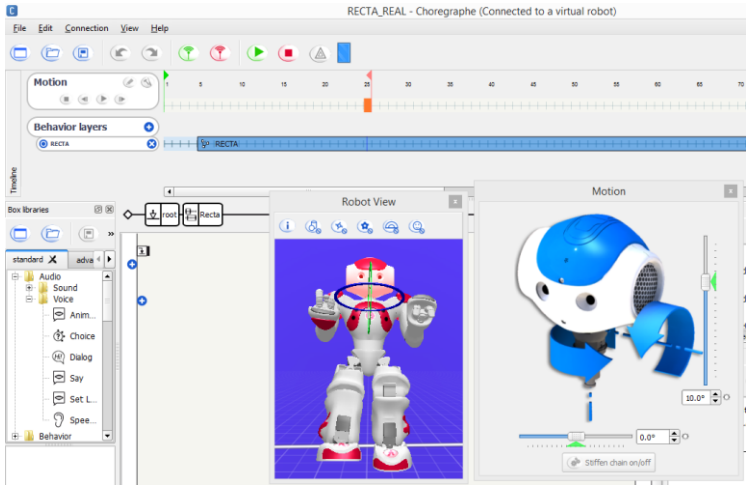


Figura 2-3: Selección de la cabeza para captura de posiciones

Realizado por: CARRANCO, Carlos Alberto, 2017

4. Al estar en las posiciones deseadas para el robot se debe activar el botón Stiffen chain on/off, para poner rígidas las articulaciones y queden en la posición que deseamos determinar.

Stiffen chain on/off		Rigidez desactivada, pueden moverse de forma manual al robot físico, pero los comandos no tendrán ninguna acción.
		Estado intermedio dependiendo del valor de rigidez
		El robot podra moverse unicamente enviando un comando

5. Luego de poner rígidas las articulaciones se debe dirigir a la regla del tiempo, para proceder a capturar las posiciones de las articulaciones sea por: todo el cuerpo (whole body), head (cabeza), brazos (arms), piernas (legs).

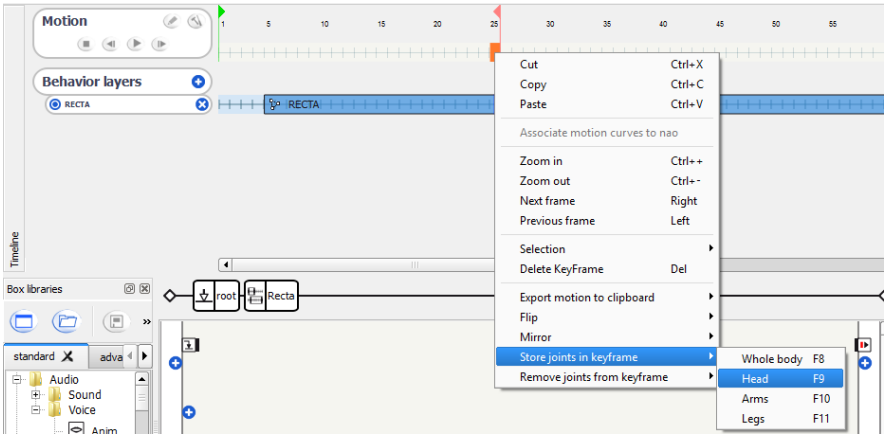


Figura 3-3: Guarda los daros de articulación elegida

Realizado por: CARRANCO, Carlos Alberto, 2017

6. Después de tener al robot en la posición deseada, como la programación en su mayoría se encuentra desarrollada en lenguaje Python, nos dirigimos al lugar donde se capturo las articulaciones, dar clic derecho y nos ubicaremos en ***Export motion to clipboard*** para obtener las posiciones a ser utilizadas con Python.

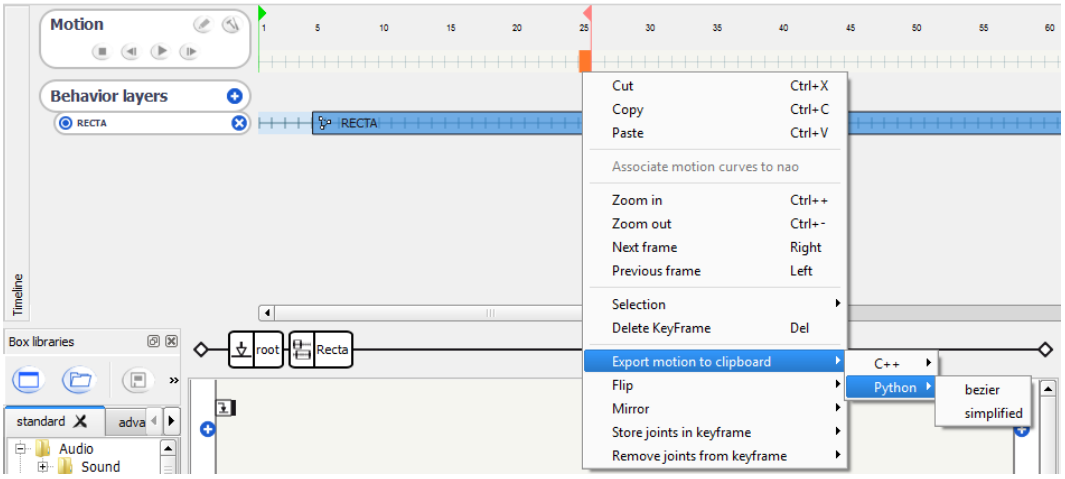


Figura 4-3: Exportar datos para usar en script de Python

Realizado por: CARRANCO, Carlos Alberto, 2017

7. En este caso se captura los datos dando clic en simplified, quedando copiado los datos de la articulación seleccionadas, y nos dirigimos a un editor de texto donde se realizará un pegado para proceder a realizar la respectiva programación en Python.

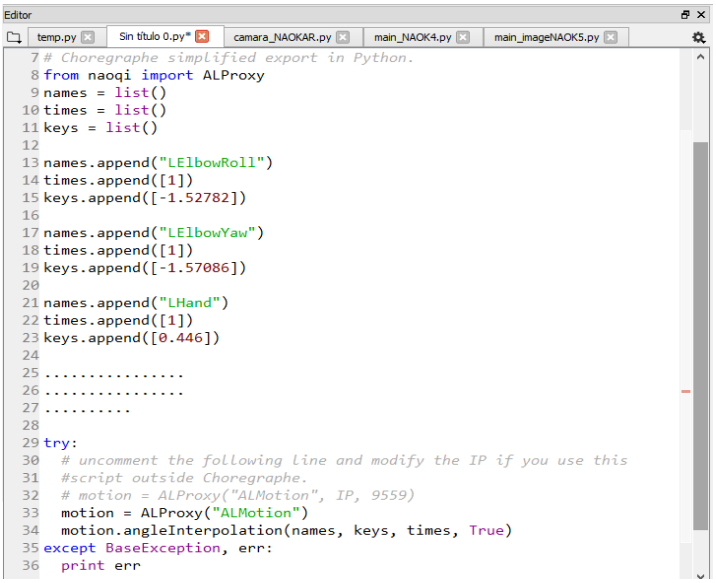


Figura 5-3: Desarrollo del script en Python con los datos capturados

Realizado por: CARRANCO, Carlos Alberto, 2017

8. De esta forma se realizará la captura de cada articulación para poderla manipular de forma independiente.

3.2. Coche eléctrico

Para que el robot pueda realizar de forma adecuada el accionamiento del volante se debe tener en cuenta los siguientes cambios en el coche eléctrico.

1. El coche tiene un motor eléctrico para el comando por radio frecuencia en la dirección, por lo que se procede a desconectar el mecanismo del motor hacia las ruedas, quedando como un coche común y corriente para niños.
2. Se cambia el volante, debido a que el robot NAO, no tiene la suficiente fuerza y ángulo en los brazos para el giro total del volante a los lados derecho o izquierdo. Se reemplaza con los siguientes componentes.
    - ***a.*** Manija de fuerza llave T (Corrediza) mando 3/8” x7.1/2”
    - ***b.*** Socket Adaptador de 3/8” (F) x ¼” (M).
    - ***c.*** Extensión de mando 1/4" 2" & 4" (L)
    - ***d.*** Copa de mando ¼” de hexagonal ¼”

Figura 6-3: a) Mando armado volante, b) Mando montado en la columna dirección
Realizado por: CARRANCO, Carlos Alberto, 2017

3. Control de tracción para el movimiento del vehículo. En el inicio de arranque y la velocidad que lleva el coche, se toma como alternativa el controlar al motor con modulación ancho de pulso, para que en el inicio del desplazamiento sea suave y reducir la velocidad para realizar el procesamiento de la imagen de mejor manera. En el coche se dispone de tres velocidades (Alta, Media, Baja).

   Para esto se utiliza una tarjeta Arduino MEGA 256 y un driver L298N para el control del motor eléctrico.

   La modulación por ancho de pulsos de una señal o fuente de energía es una técnica adecuada para modificar el ciclo de trabajo de una señal (análoga o digital), ya sea para transmitir datos mediante un canal de comunicación o para el control de energía que se envía a una carga. (Enrique, 2016)

   El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período duty cycle = (tiempo que la salida está a uno o HIGH) / (periodo de la función). (Enrique, 2016)

3.3. Sistema de visión

El uso del sistema de visión mediante la cámara del robot, permite el reconocimiento del entorno, para tomar la acción adecuada, realizando el procesamiento de búsqueda de líneas que delimitan a la pista siendo un factor muy importante, para mantener al vehículo en línea recta o corregir la desviación sobre el volante y a su vez al coche en el momento del desplazamiento.

En este trabajo se toma como punto de partida, el código fuente de Self Driving Car Nano Degree (Priyanka Dwivedi, 2017), pero aplicándolo en tiempo real, con resolución distinta de cámara y las maniobras la realizará el robot NAO.

A continuación, se detalla los métodos para el uso y el procesamiento de la imagen.

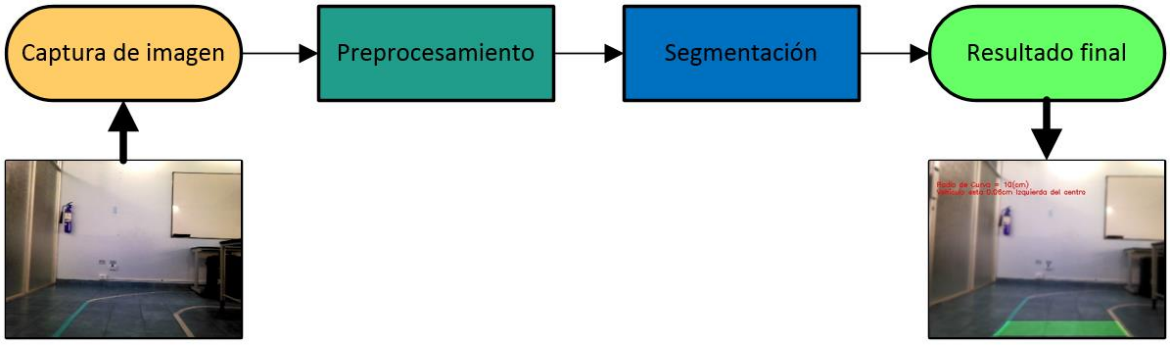


Figura 7-3: Estructura del procesamiento de las líneas por cámara del robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

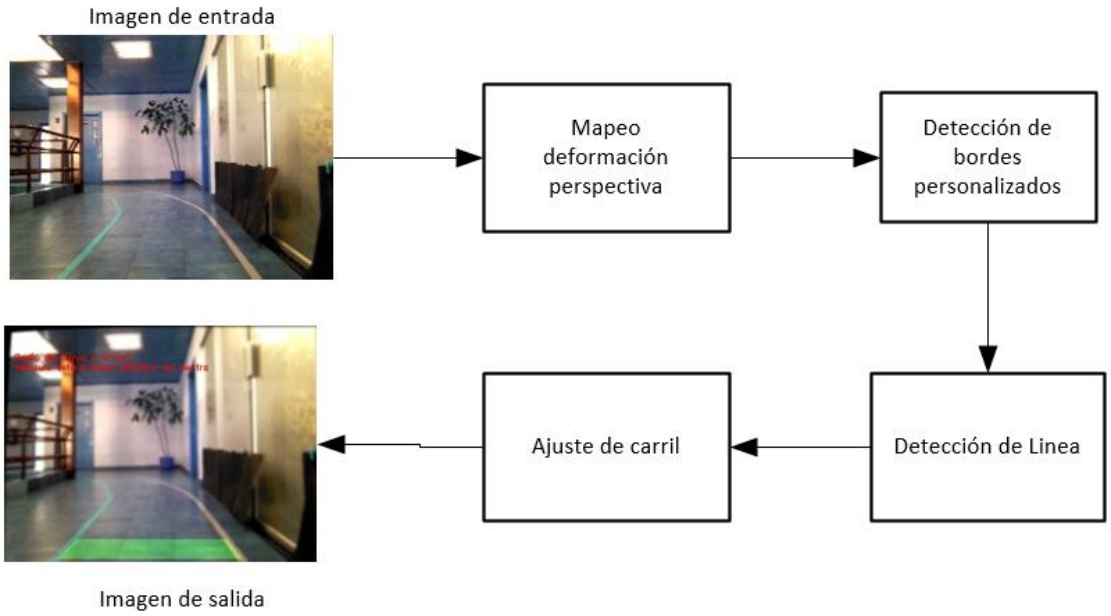


Figura 8-3: Diagrama de bloque de detección de pista

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.1. Calibración de cámara

La calibración es factor importante para los sistemas de visión. En este trabajo se permite la obtención de los parámetros intrínsecos y extrínsecos (internos y externos) de la cámara. Para el uso de algoritmos de posicionamiento es indispensable disponer de estos parámetros usados en la obtención de imágenes. Estos parámetros se adquieren con la calibración de cámara. (Florencia, Lanzaro; Fuentes, 2014)

Parámetros intrínsecos:

Los parámetros intrínsecos definen la geometría interna y la óptica de la cámara. Éstos determinan cómo la cámara proyecta los puntos del mundo 3D al plano de la imagen en 2D, siendo constantes en tanto no varíen las características y posiciones relativas entre la óptica y el sensor imagen Figura 9-3. (Florencia, Lanzaro; Fuentes, 2014)

Punto principal:

El punto principal (punto C) es el punto intersección entre el plano de la imagen y el eje óptico (recta perpendicular al plano de la imagen que pasa por el centro de cámara O). Las coordenadas de este punto vienen dadas en píxeles, y son expresadas respecto al sistema solidario al plano de la imagen (O',x',y').(Florencia, Lanzaro; Fuentes, 2014)

Distancia focal:
La distancia focal de una cámara es la distancia existente entre ésta (punto O) y el punto principal. Las coordenadas de este punto vienen dadas en píxeles horizontales y verticales.(Florencia, Lanzaro; Fuentes, 2014)

Centro óptico.
Punto donde se localiza la cámara. Por defecto, se considera en el centro de coordenadas, observando en dirección de Z negativo y con Y positivo hacia arriba.(Florencia, Lanzaro; Fuentes, 2014)

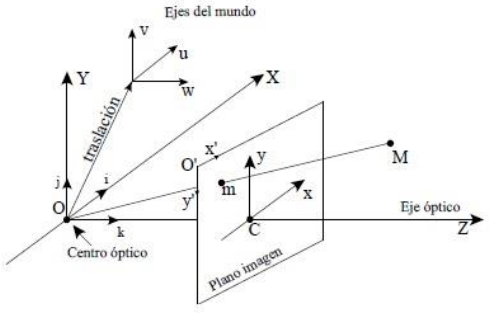


Figura 9-3: Parámetros intrínsecos de una cámara

Fuente: Florencia Lanzaro, 2014

Parámetros extrínsecos:

Definen la relación (traslación y rotación) entre un sistema de coordenadas absoluto (exterior a la cámara) y otro vinculado a la cámara. Incluyen seis parámetros: tres para la traslación (Tx, Ty, Tz) y tres para los ángulos rotados sobre cada uno de los ejes (α, β, γ). (J. I. González, 2003)

3.3.1.1. Proceso calibración cámara robot nao

Para obtener estos parámetros se ejecuta el script camera_calibration. Cabe mencionar que el objetivo principal de la aplicación de esta etapa, es la disminución de la distorsión tangencial y radial.

Al realizar la calibración en primer lugar, se debe colocar con imágenes capturadas de un tablero de ajedrez, para el caso se usa 6x4 casillas obtenidas con la cámara superior del robot y con la resolución que se desea ya definida. Se realiza distintas pruebas de las resoluciones y dimensiones, disponibles con la cámara, con un patrón de medidas conocidas (tablero de ajedrez) en distintas posiciones. Al ejecutar el script en Python se podrán obtener las características detectando las esquinas con precisión, para proceder a conseguir los parámetros antes mencionados, en la Figura 10-3 se muestra una de las gráficas de la calibración de cámara superior del robot NAO, las imágenes deberán ser al menos unas 20 capturas y en distintas posiciones del tablero de ajedrez, para que se pueda realizar un buen procesamiento de las líneas de la pista.

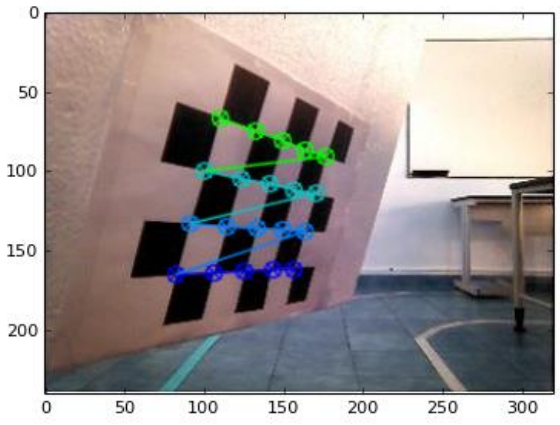


Figura 10-3: Detección de esquinas

Realizado por: CARRANCO, Carlos Alberto, 2017

Se ejecuta otro script para aplicar la corrección que se observa con la disminución de la distorsión Figura 11-3 aplicada en un tablero de ajedrez, lo mismo se realiza a la pista de prueba Figura 12-3, se puede observar que nos indica la escala a la cual se ha elegido la cámara del robot que es de 320 x 240 y el procesamiento que se obtiene.

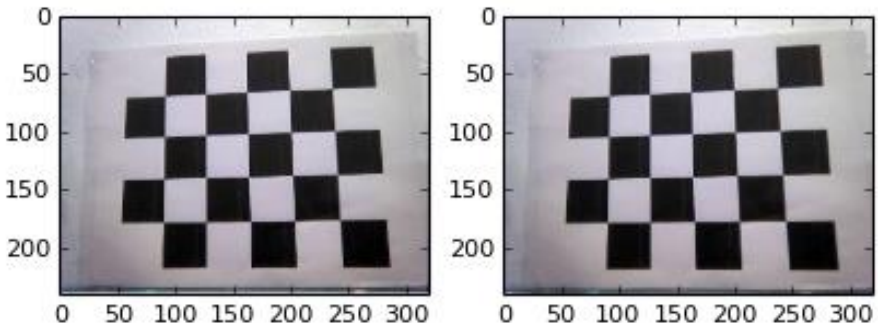


Figura 11-3: Imagen original (izquierda), imagen sin distorsión (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

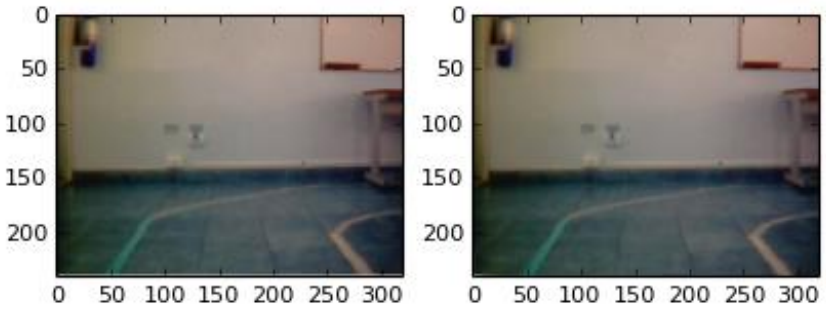


Figura 12-3: Pista imagen original (izquierda), pista imagen sin distorsión (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.2. Campo de visión y conducción

El área de visión perfecta se produce en un cono con un ángulo de apertura de 3 grados, en el que todos los objetos que se encuentran dentro de esta área se perciben con absoluta nitidez. (MAPFRE, s/f)

La visión sigue siendo clara con un ángulo de cinco o seis grados. En la medida que el ángulo de apertura se ensancha, las imágenes captadas se dejan de percibir con claridad, hasta perderse alrededor de un ángulo de 20 grados. (MAPFRE, s/f)

En el plano vertical se reduce el cono de visión aproximándose a un tercio de las cifras dadas para el plano horizontal. (MAPFRE, s/f)

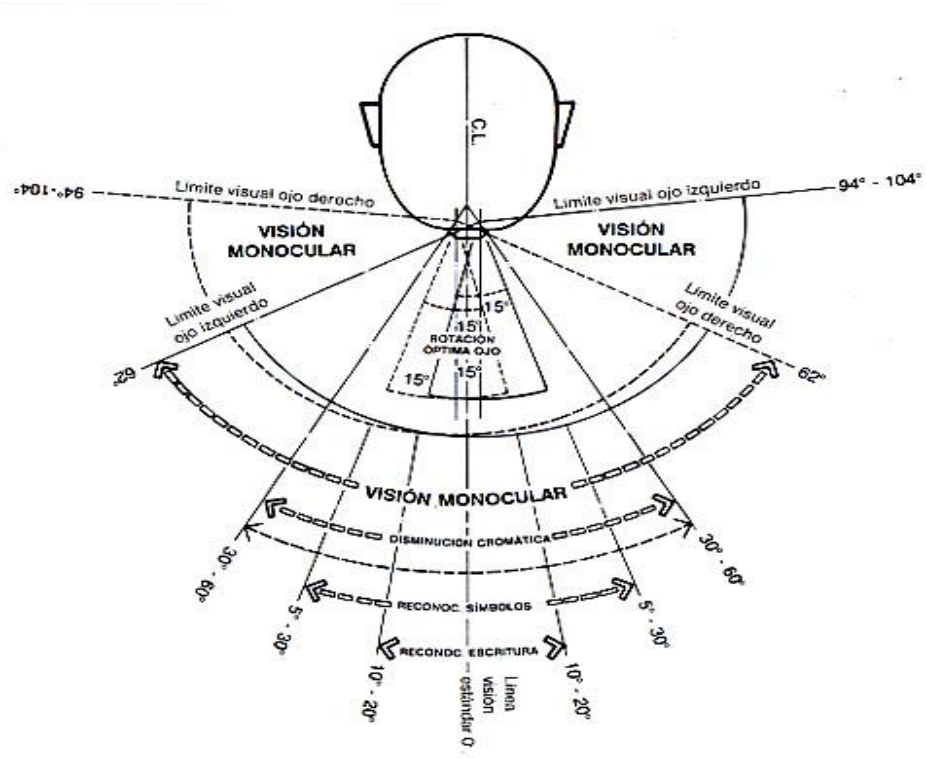


Figura 13-3: Campo visual horizontal

Fuente: Yunesky, 2013

El campo de visión está representado por un valor ajustable en muchos video juegos y que afecta directamente a la distancia de visión y tamaño de los objetos que se muestran en pantalla.

Los seres humanos tenemos los ojos relativamente juntos y colocados en el frontal de la cara, lo que permite una visión focalizada muy exacta en cuanto a tamaños y distancias con casi 180° de campo de visión horizontal y 100° de visión vertical.(Sim, 2012)

En el desarrollo del proyecto se hace uso de la cámara con dos objetivos:

1. Visualización de la pista. (visión horizontal)
2. Identificación de personas frente al vehículo. (visión horizontal – vertical)

3.3.3. Procesamiento de las líneas

El programa que se ejecuta debe realizar la detección de líneas de la pista, similar al código base e implementándose en el procesamiento la identificación de personas con funciones ya disponibles en Open CV.

En primer lugar, se debe importar las librerías que se vayan usar como se puede observar se hace el llamado de las librerías tanto del robot NAO, Open CV, Image, manejo de array y plots para graficar el procesamiento realizado.

```
import numpy as np
import vision_definitions
from PIL import Image
from naoqi import ALProxy
import datetime
import cv2
import pickle
import matplotlib.pyplot as plt
```

Archivo generado, al ejecutar el script camera_calibration.py ver Figura 14-3.

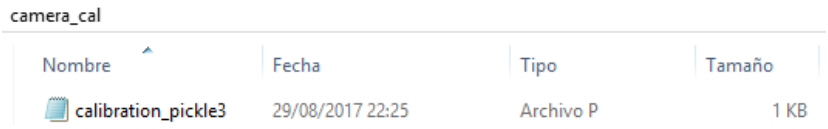


Figura 14-3: Archivo creado al ejecutar camera_calibration.py

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.1. Obtener Imagen de la cámara superior del robot NAO.

Para el proceso de obtener la imagen con la cámara del robot NAO se debe tener en cuenta los puntos tratados en el MóduloALVideoDevice. A continuación, se puede ver la configuración realizada con la cámara del robot NAO, ya que al hacer pruebas específicamente en la variable ***resolution***, mientras el valor es más alto se obtiene una mejor calidad de imagen en el procesamiento, pero existe un retardo entre cuatro a siete segundos, si es la más baja no realiza el procesamiento de la imagen como se requiere.

```
import vision_definitions
from naoqi import ALProxy

# DATOS DE CONFIGURACIÓN PARA EL ROBOT
IP = "IP ROBOT NAO"
PORT = 9559
cameraid=0 # Configuración para la cámara superior

camProxy = ALProxy("ALVideoDevice", IP, PORT)
resolution = vision_definitions.kQVGA # 320x240
colorSpace = vision_definitions.kBGRColorSpace
videoClient = camProxy.subscribe("python_client", resolution, colorSpace, 5)
camProxy.setParam(vision_definitions.kCameraSelectID, cameraid)
```

Al ejecutar el script, se tiene distintos procesos para el tratamiento de la imagen (Figura 15-3), esta imagen se obtendrá por medio de la cámara del robot NAO Figura 16-3.

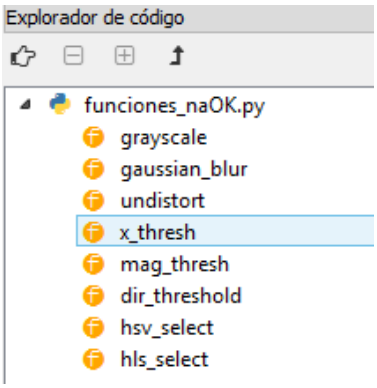


Figura 15-3: Funciones del script

Realizado por: CARRANCO, Carlos Alberto, 2017

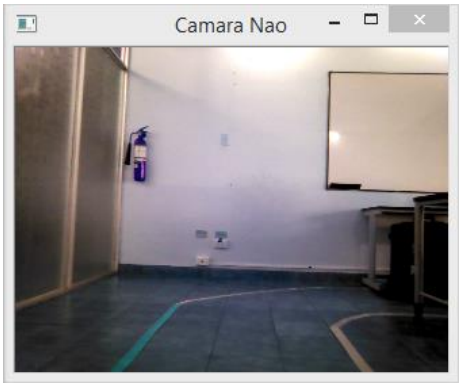


Figura 16-3: Imagen capturada con la cámara superior del robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

Se detalla los procesos realizados en la imagen que se obtiene con la cámara, con la ayuda de las librerías de Open CV para Python.

3.3.3.2. Función grayscale

Se crea una función llamada grayscale, para realizar la conversión a escala de grises, reduciendo el número de canales aplicando a la imagen en tiempo real Figura 17-3.

```
def grayscale(img):
    return cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

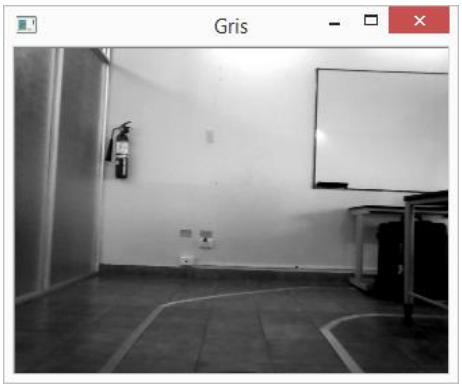


Figura 17-3: Imagen en Gris cámara superior robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.3. Función Gaussian Blur

Se crea una función gaussian_blur para reducir el ruido de la imagen. Este proceso es el mismo que convolucionar la imagen con una función gaussiana, suaviza los bordes **Figura 18**-3.

```
def gaussian_blur(img, kernel_size):
    """Applies a Gaussian Noise kernel"""
    return cv2.GaussianBlur(img, (kernel_size, kernel_size), 0)
```

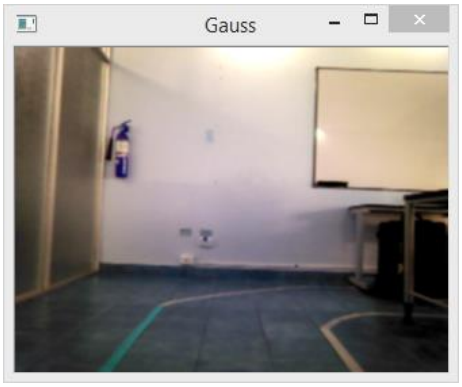


Figura 18-3: Suavizado imagen cámara superior robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.4. Función undistort

Se crea una función undistort, llamando al archivo que contiene los parámetros de la calibración y corrección de distorsión de la cámara del robot NAO.

Hay dos pasos principales para el proceso: usar las imágenes de tablero de ajedrez para los puntos de imagen y puntos de objeto, y luego usar las funciones de OpenCV cv2.calibrateCamera( ) y cv2.undistort ( ) para calcular la calibración y la no distorsión.

```
def undistort(img):
    cal_pickle = pickle.load( open("camera_cal/calibration_pickle3.p", "rb"))
    mtx = cal_pickle["mtx"]
    dist = cal_pickle["dist"]
    undist = cv2.undistort(img, mtx, dist, None, mtx)
    return undist
```

Figura 19-3: Resultado de aplicar la función undistort

Realizado por: CARRANCO, Carlos Alberto, 2017

Al abrir el archivo creado con el script camera_calibration.py se obtiene los siguientes datos que el programa lo interpretara al ejecutarse en la función undistort.

```
(dp0S'mtx'p1cnumpy.core.multiarray_reconstructp2
(cnumpyndarrayp3(I0tp4S'b'p5tp6Rp7(I1(I3I3tp8
cnumpydtypep9(S'f8'p10I0I1tp11Rp12(I3S'<'p13NNNI-
1I-1I0tp14bI00S"\xc5\x91\xf5N\xb4Vq@
\x00\x00\x00\x00\x00\x00\x00\x009\x14e
\x00\xc5\xf7b@
\x00\x00\x00\x00\x00\x00\x00\x00\x7f\xaf\xb2/
\xa4Uq@:L'\x81\xbc\xb2\\@
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\xf0?"p15
tp16bsS'dist'p17g2(g3(I0tp18g5tp19Rp20(I1(I1I5
tp21g12I00S':\xd8\x19|\x00U\xa4\xbf\xa4\xe8a\x16!
\xbe\xbfx\xbf\x191\xc54T\xbf\x0b/P\xedf\xc2~\xbf
\x01t\x86&\x91]\xd6?'p22tp23bs.
```

Figura 20-3: Parámetros de calibración cámara, resolución 320x240

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.5. Función x_thresh

En esta función, se aplica el operador Sobel a una imagen, es una forma de tomar la derivada de la imagen en la dirección x o y. Al aplicar los operadores para Sobelx y Sobely, se obtiene el siguiente resultado Figura 21-3.

```
def x_thresh(img, sobel_kernel=3, thresh=(0, 255)):
    ......
    .........
    .........
    return sxbinary
```

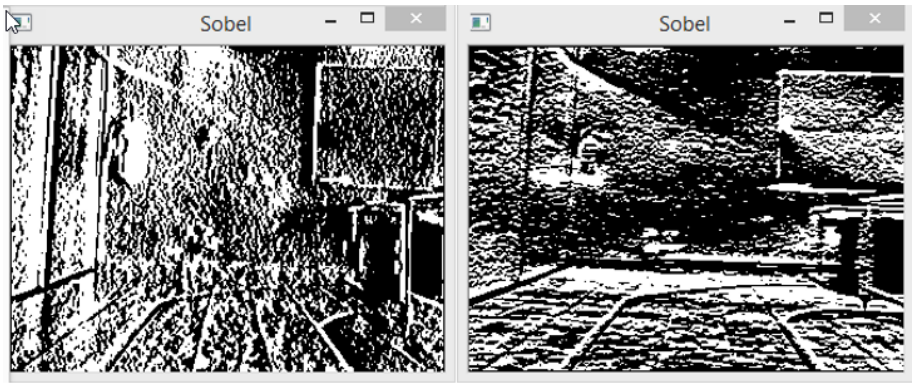


Figura 21-3: Valor absoluto Sobel *x* (izquierda), Valor absoluto Sobel *y* (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 21-3, se observa los gradientes tomados en las direcciones x e y detectando las líneas de la pista obteniéndose varios bordes. Al tomar el gradiente en la dirección x se resalta los bordes más cercanos a la vertical y en la dirección y los bordes cercanos a la horizontal.

3.3.3.6. Función mag_thresh

Esta función calcula la magnitud del gradiente o valor absoluto del gradiente. Con el resultado que se obtuvo anteriormente, se toma el gradiente en *x* o *y*, así se podrá establecer umbrales para la identificación de pixeles en un rango determinado. Al manipular los umbrales, se puede encontrar el gradiente *x*, obteniéndose las líneas de la pista, al igual que se puede ver las líneas en el degradado *y*.(hahnsang, 2016c)

```
47 def mag_thresh(img, sobel_kernel=3, thresh=(0, 255)):
48     # Convert to grayscale
49     gray = grayscale(img)
50     # Take both Sobel x and y gradients
51     sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=sobel_kernel)
52     sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=sobel_kernel)
53     # Calculate the gradient magnitude
54     gradmag = np.sqrt(sobelx**2 + sobely**2)
55     # Rescale to 8 bit
56     scale_factor = np.max(gradmag)/255
57     gradmag = (gradmag/scale_factor).astype(np.uint8)
58     # Create a binary image of ones where threshold is met, zeros otherwise
59     binary_output = np.zeros_like(gradmag)
60     binary_output[(gradmag >= thresh[0]) & (gradmag <= thresh[1])] = 1
61
62     # Return the binary image
63     return binary_output
```

Se debe considerar el tamaño de la región en la imagen de la que se tomará el gradiente. El tamaño predeterminado del núcleo de Sobel es de tres si se lo cambia este número deberá ser impar. Se puede observar en la Figura 22-3, el resultado del gradiente escalado a ocho bits, la imagen binaria y el Sobel identificados en cada frame.

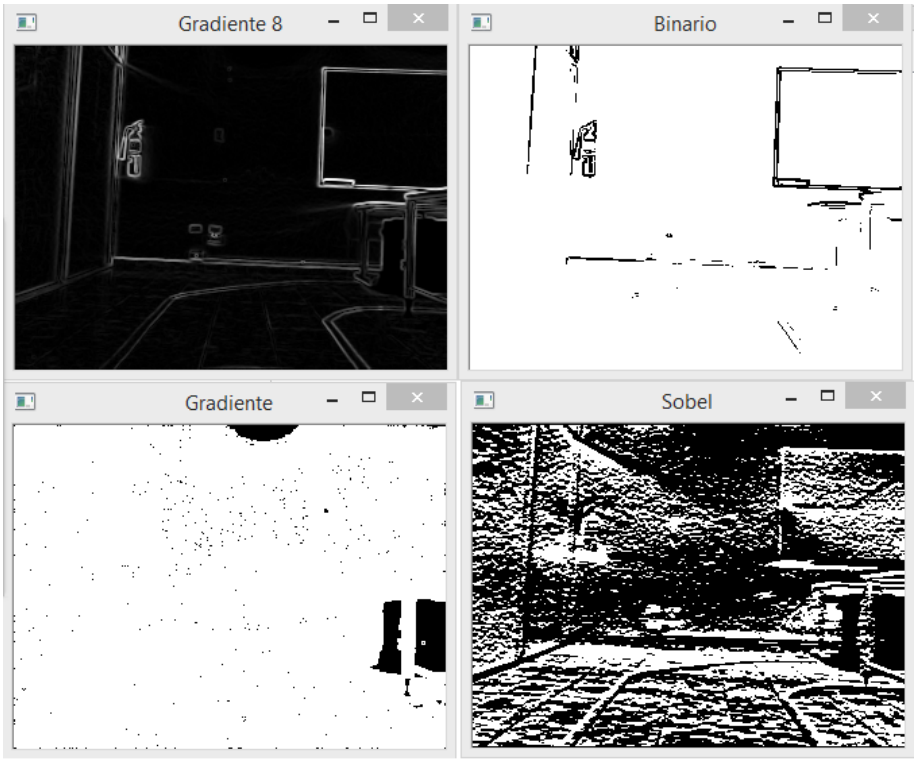


Figura 22-3: Imágenes obtenidas en la función magnitud del gradiente

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.7. Función dir_threshold

En esta función, se aplica para obtener los bordes de la pista en una orientación en particular. Cada pixel resultante contiene un valor para el ángulo del gradiente con respecto a la horizontal en unidades de radianes. Convirtiéndose en un rango de -π/2 a π/2. Una orientación de 0 involucra una línea horizontal y orientación +/- π/2 involucra línea vertical. (hahnsang, 2016a)

A continuación, se puede observar el código que se aplica en la función para calcular la dirección del gradiente y en la Figura 23-3 los resultados obtenidos.

```
65 def dir_threshold(img, sobel_kernel=3, thresh=(0, np.pi/2)):
66
67     gray = grayscale(img)
68     # Take both Sobel x and y gradients
69     sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=sobel_kernel)
70     sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=sobel_kernel)
71     # Calculate the gradient magnitude
72     abs_sobelx = np.absolute(sobelx)
73     abs_sobely = np.absolute(sobely)
74
75     dir_grad = np.arctan2(abs_sobely, abs_sobelx)
76
77     binary_output = np.zeros_like(dir_grad)
78     binary_output[(dir_grad >= thresh[0]) & (dir_grad <= thresh[1])] = 1
79
80     return binary_output
```

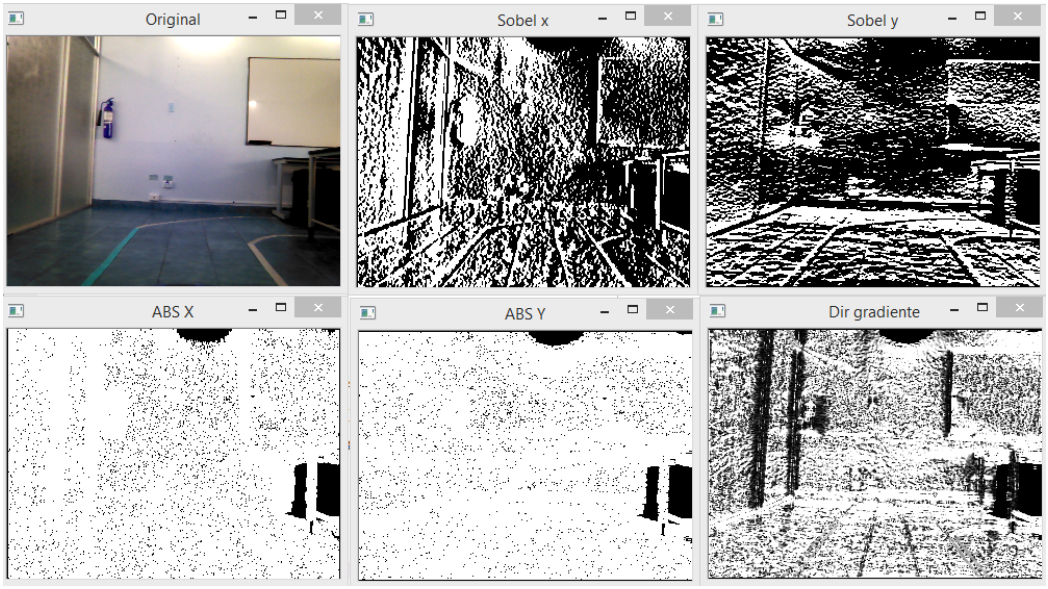


Figura 23-3: Resultado obtenido al aplicar la función de dirección de magnitud

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.8. Funciones HLS y HSV

La función Sobel no cumplirá con la búsqueda de la línea amarilla, por lo que es necesario aplicar otro espacio de colores como lo es del modelo HLS, éste se lo conoce como modelo sensible de color, por tomar los atributos de la percepción humana con la luz. (Ariagna, 2015)

Los modelos HLS y HSV incluyen otros dos parámetros adicionales al matiz para conseguir el color, siendo estos la saturación (en ambos) y el valor (en HSV) o la luminosidad o tono (en HSL).

La diferencia entre HSV y HSL es que en HSV la saturación va del color puro al blanco, y en HSL la saturación va del color puro al gris medio, y el tono, en HSV va desde el negro al color, y en HSL va desde el negro al blanco. (Herrera, 2014)

Con las funciones se hace un filtro para escoger solamente elementos amarillos y blancos, usando la librería de Opencv. La dimensión HSV es adecuada para hacer esto, ya que aísla el color (tonalidad), la cantidad de color (saturación) y el brillo (valor). Se define el rango de amarillo independiente en el brillo realizando pruebas para obtener los valores de las líneas a identificar. En la Figura 24-3 se puede observar la aplicación de la función HLS sobre la pista.

Figura 24-3: Imagen obtenida al trabajar con la función HLS

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.9. Función warp

La función warp realiza la deformación de la imagen, como si se estuviese viendo desde arriba (vista de ave). Transforma los puntos de la imagen que se obtiene por la cámara del robot NAO en puntos de imagen diferentes, como ejemplo se puede ver en la Figura 25-3.

Se considera que la cámara está en una posición fija, y la posición relativa de la pista son siempre las mismas, se debe tener en cuenta que la cámara está ubicada en la cabeza del robot NAO.

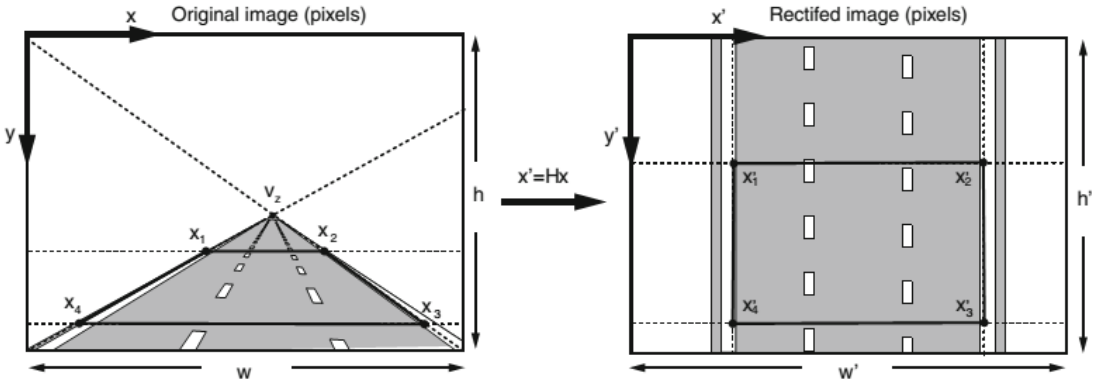


Figura 25-3: Imagen perspectiva inversa calculada por medio de correspondencias 4 puntos

Fuente: Nieto, Marco, 2012

La función warp de OpenCV necesita cuatro puntos: de origen (src) y destino (dst), como se puede ver en la Tabla 6-3, definidos en el código de la función y aplicados con la cámara del robot NAO para el lugar al cual se requiere realizar la detección de las líneas.

```
def warp(img):
    img_size = (img.shape[1], img.shape[0])

    src = np.array([[0,218],[320,218],[320,242],[0,242]],np.float32)
    dst = np.array([[0,0],[320,0],[320,288],[0,288]],np.float32)

    M = cv2.getPerspectiveTransform(src, dst)

    #inverse
    Minv = cv2.getPerspectiveTransform(dst, src)

    #create a warped image
    warped = cv2.warpPerspective(img, M, img_size, flags=cv2.INTER_LINEAR)

    unpersp = cv2.warpPerspective(warped, Minv, img_size, flags=cv2.INTER_LINEAR)

    return warped, unpersp, Minv
```

Basándose con respecto a la imagen que es capturada por la cámara del robot NAO y aplicando los valores de la Tabla 1-3 se define la región de interés, el resultado final se puede observar en la Figura 26-3. Cabe indicar que para obtener los valores src y dst, se realiza un script para determinar el área a procesar.

Tabla 1-3: Puntos de origen y destino detección de la pista

Fuente (src) (x, y)	Destino (dst) (x, y)
[0, 218]	[0, 0]
[320, 218]	[320, 0]
[320, 242]	[320, 288]
[0, 242]	[0, 288]

Realizado por: CARRANCO, Carlos Alberto, 2017

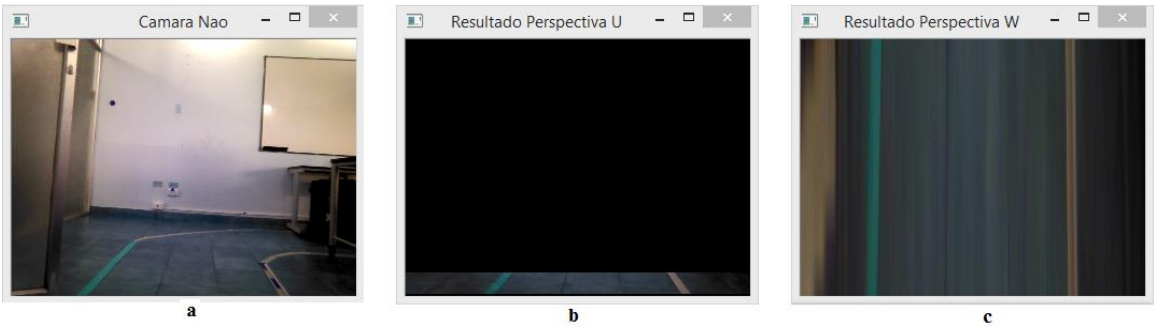


Figura 26-3: a) Imagen origen, b) Región de interés, c) Resultado perspectiva "vista de ave"

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.10. Función lane_detector y fit

En la función lane_detector lo que se realizará, es la aplicación de las funciones antes mencionadas ya que solamente queda aplicar la búsqueda precisa de las líneas donde el vehículo se desplazará con el robot NAO. En la función se procesa la imagen sin distorsión, se aplica la función gausiana con un kernel ya definido, para luego aplicar la función del gradiente: dirección, magnitud y orientación, descritos anteriormente. Se procesa el color, los valores determinados en pruebas independientes del rango más ideal, se combinan los colores ya que debemos recordar que se utiliza la línea amarilla en el lado izquierdo y línea blanca lado derecho se puede observar en la Figura 27-3. Para finalizar se aplica la función warp determinando específicamente la posición de la pista en relación a la pantalla donde se puede observar el resultado final Figura 28-3

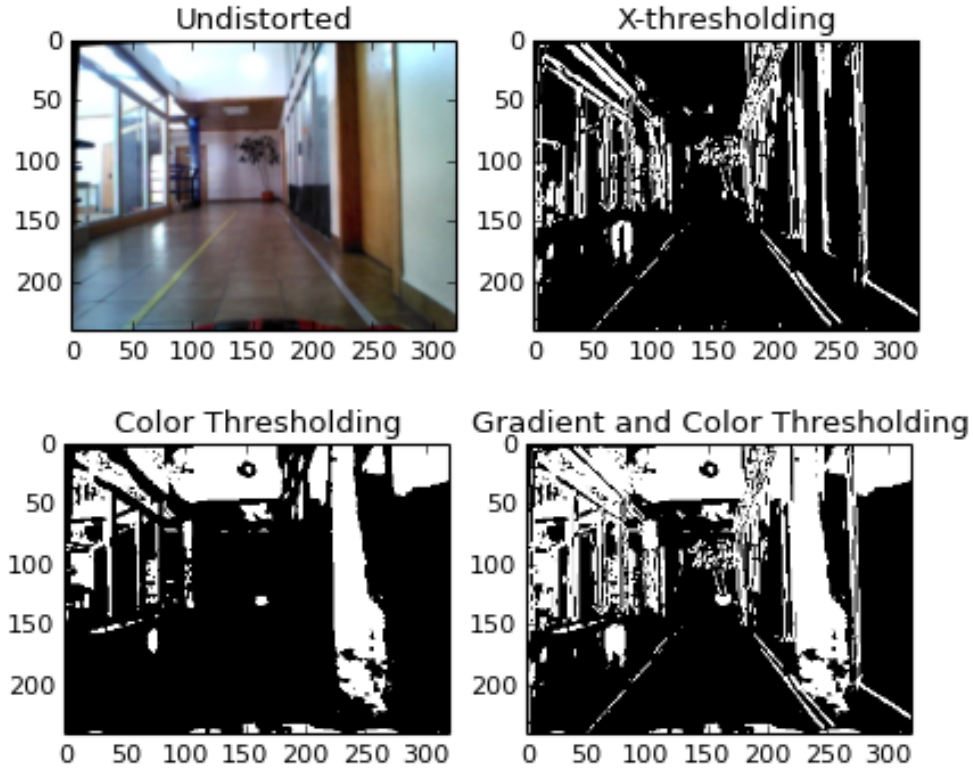


Figura 27-3: Resultado de aplicar las funciones a la imagen capturada por la cámara del NAO

Fuente: CARRANCO, Carlos Alberto, 2017

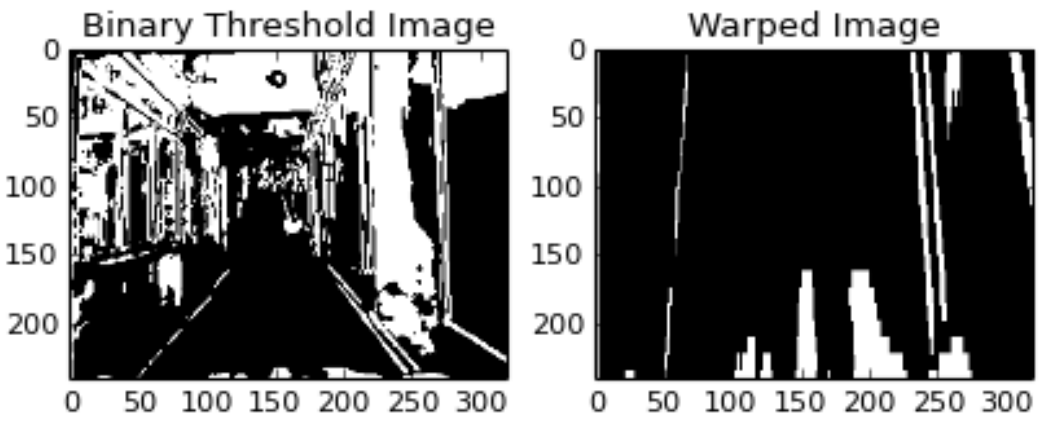


Figura 28-3: Aplicación de la función warp a la imagen capturada (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

```
def lane_detector(image):

    # IMagen sin distorsión
    undist = undistort(image)

    # Define el tamaño del kernel aplicando la funcion Gaussiana
    apply_blur = True
    if apply_blur:
        kernel_size = 5
        undist = gaussian_blur(undist, kernel_size)

    # Define parametros para el gradiente
    sxbinary = x_thresh(undist, sobel_kernel=3, thresh = (22,100))
    mag_binary = mag_thresh(undist, sobel_kernel=3, thresh=(40, 100))
    dir_binary = dir_threshold(undist, sobel_kernel=15, thresh=(0.7, 1.3))

    # Define parametros para colores
    s_binary = hls_select(undist, thresh=(90, 255))

    #Combina varios colores
    color_binary = np.dstack(( np.zeros_like(sxbinary), sxbinary, s_binary))

    # Combina dos colores de umbral
    combined_binary1 = np.zeros_like(sxbinary)
    combined_binary1[(s_binary == 1) | (sxbinary == 1)] = 1

    combined_binary2 = np.zeros_like(sxbinary)
    combined_binary2[(s_binary == 1) | (sxbinary == 1)| (mag_binary == 1)] = 1

    # Aplica la transformacion de perspectiva
    warped_im, _ , Minv = warp(combined_binary1)

    return undist, sxbinary, s_binary, combined_binary1, warped_im, Minv
```

3.3.3.11. Función curvature

Esta función determina el ángulo de dirección que el robot deberá corregir sea izquierda o derecha para tratar de mantener al vehículo dentro de la pista.

Una manera de calcular la curva de la línea, es ajustarla a un polinomio de segundo grado, a partir de esto se podrá extraer la información que sea útil. Para la línea de la pista que se encuentra cerca de la vertical, puede ajustarse una línea usando la formula $f(y) = Ay^2 + By + C$ donde A, B y C son los coeficientes. (hahnsang, 2016b)

Siendo que A nos entrega el dato de la curva de la línea, B indica la dirección que la línea apunta y C la posición de la línea, basándose en lo lejos que se encuentra de la izquierda de la imagen (y=0).(hahnsang, 2016b)

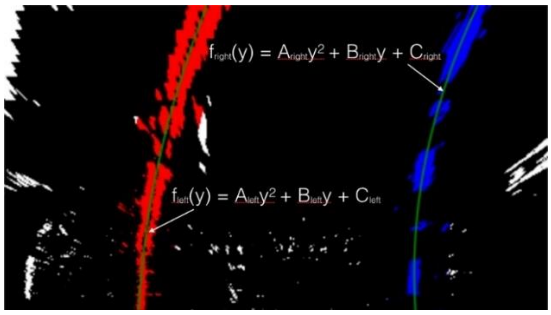
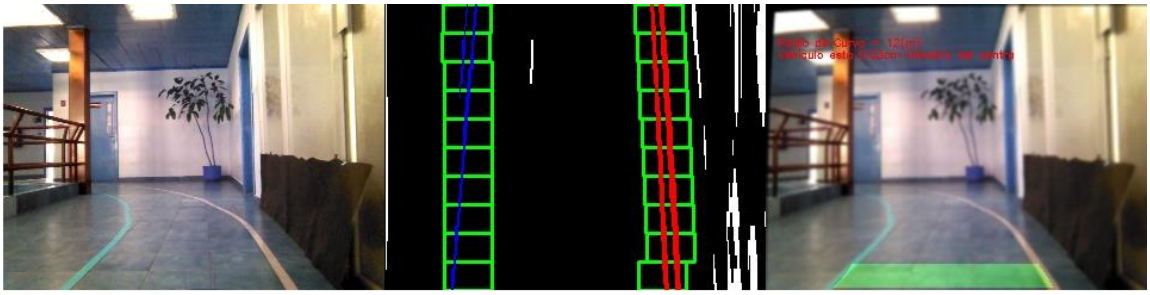


Figura 29-3: Detección lineas para determinar la curva

Fuente: claudiu, coldvision.io, 2017

A continuación, en la Figura 30-3 se muestra la aplicación final de las funciones antes indicadas.



a **b** **c**

Figura 30-3: a) Imagen origen, b) Identificación de líneas con función warp, c) Resultado final

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.12. Detección de personas

Para la detección de personas, se aplica el detector HOG + Linear SVM integrado en la librería de OpenCV, lo que nos permite detectar personas tanto en imágenes como en secuencia de video. (Rosebrock, 2015a) (Rosebrock, 2015b)

En la Figura 31-3 se muestra el resultado de la detección de persona con la cámara del robot NAO.

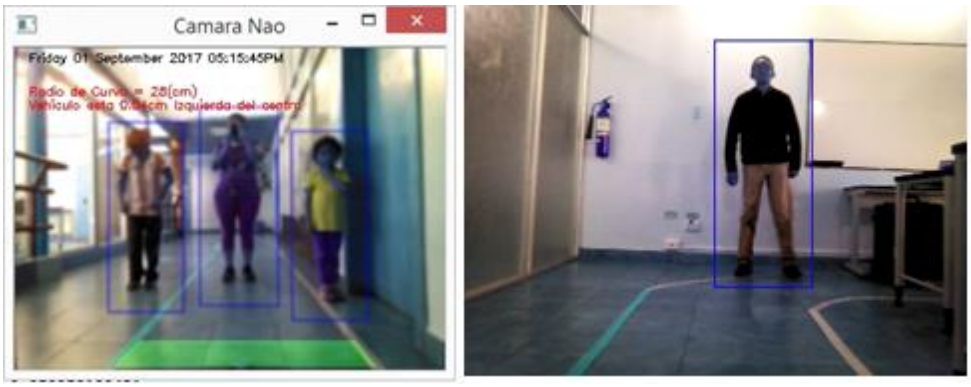


Figura 31-3: Detección personas y líneas (izquierda), Detección de solo persona (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

3.3.3.13. Procesamiento de la imagen de la cámara del robot NAO.

Todo lo descrito anteriormente se lleva a cabo para la detección de líneas y personas, pero cabe mencionar que el procesamiento es llevarlo con la obtención de imágenes por medio de la cámara del robot NAO.

En un principio el proceso se realiza simulando la obtención de imágenes, con una webcam LogiLink® Webcam USB con LED, características en la Tabla 6-3, para luego implementar la programación con el robot NAO, obteniendo similares resultados con ciertas diferencias como se indicó anteriormente.

Tabla 2-3: Características Webcam USB con LED

Connector: USB Port
Sensor: 300K CMOS, interpolated to 8M pixels
Max. resolution: 640x480
Color: 24 Bit True Color
Video framerate: up to 30 fps
360 degrees swiveling
With 6 LEDs
Plug & Play

Fuente: LogiLink

Al simular la cámara del robot NAO, ver Figura 32-3, se puedo enlazar con Choregraphe y por ende los movimientos del robot para poder observar el comportamiento del robot en las situaciones que se presente al detectar las líneas de la pista y detectar las personas en la trayectoria.

Figura 32-3: Ubicación webcam en coche.

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 33-3 se puede observar el robot NAO sentado en el coche, la altura casi es la misma en el momento de realizar el procesamiento de la imagen con la webcam.

Figura 33-3: Robot NAO sentado en el coche: vista frontal (izquierda), vista lateral (derecha)

Realizado por: CARRANCO, Carlos Alberto, 2017

CAPÍTULO IV

4. Resultados

Este capítulo tiene como finalidad manifestar la técnica aplicada directamente en el robot NAO, realizando el procesamiento de imagen y la conducción como se ha ido explicando en capítulos anteriores.

Los resultados conseguidos en las pruebas con la implementación del código (Priyanka Dwivedi, 2017) son favorables, con las respectivas modificaciones en valores que deben adaptarse específicamente a la cámara del robot NAO.

Siendo una situación la luz del ambiente y el piso (baldosa) un factor que en un 20% no favorece a la detección de las líneas, pero en la mayoría de las pruebas realizadas se obtiene resultados favorables en la detección de las líneas de carril. Cabe mencionar que en la detección de personas no hubo inconvenientes ya que el proceso si lo trabaja al 100%, llevando a cabo la acción sobre el pedal del acelerador y por ende sin el desplazamiento del coche, con respecto a distancia de detección desde el coche hacia la persona aproximadamente el dato mínimo es de un metro y máximo realizada la pruebas fue de ocho metros.

Se realiza pruebas con imágenes capturadas y pruebas directas con el robot, al igual que se toma el procesamiento de imágenes que demuestren el funcionamiento de la parte de visión artificial y la parte mecánica del robot NAO. Por lo cual que se analizará con el programa ejecutando en Python, Choregraphe y Monitor (estos dos últimos, son softwares que sirven para la programación y visualización de datos del robot respectivamente).

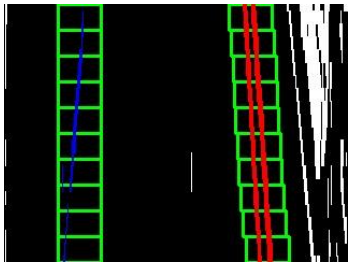
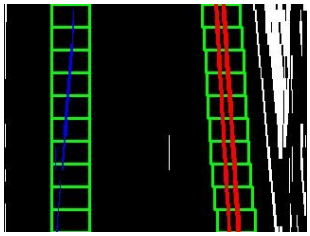
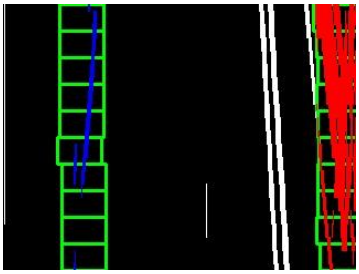
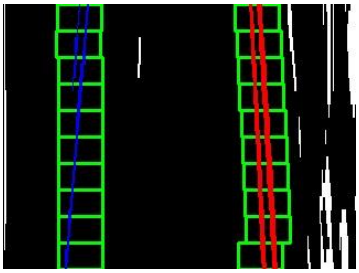
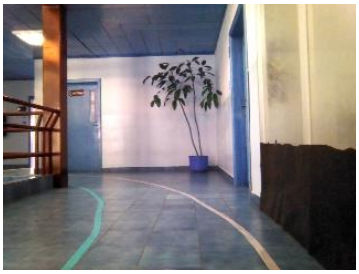
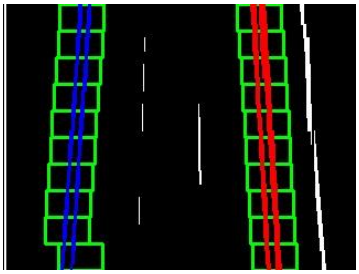
A continuación, se muestran las pruebas realizadas, con el robot tanto en procesamiento de imagen como en movimientos propios del robot NAO.

4.1. Pruebas con imágenes con cámara del robot NAO

Se obtiene imágenes con la cámara del robot NAO, en el trayecto que realiza, tomadas con la cámara del robot NAO, a una resolución de 320x420 pixeles. Al igual que se respaldan de videos donde se puede observar la secuencia del procesamiento detectando las líneas de la pista, para mostrar las pruebas, se realizan varios ensayos que muestran el funcionamiento del proyecto.

Como primer punto, es demostrar el reconocimiento de la pista con la ayuda de las líneas izquierda (color amarillo) y derecha (color blanco) con valores determinados en pruebas realizadas antes de aplicarlas directamente. En la Figura 1-4 se puede observar la progresión de los frames para las líneas izquierda y derecha, pintándose así el ROI ya definido y ajustándose a medida que la imagen es cambiante en su trayectoria, se puede visualizar que tenemos la imagen de origen, que

después de haber realizado los procesos de imagen se muestra el warp y finalizando con el objetivo de indicar el área definida.

Imagen de origen	Warp	Resultado final
		
		
		
		
		

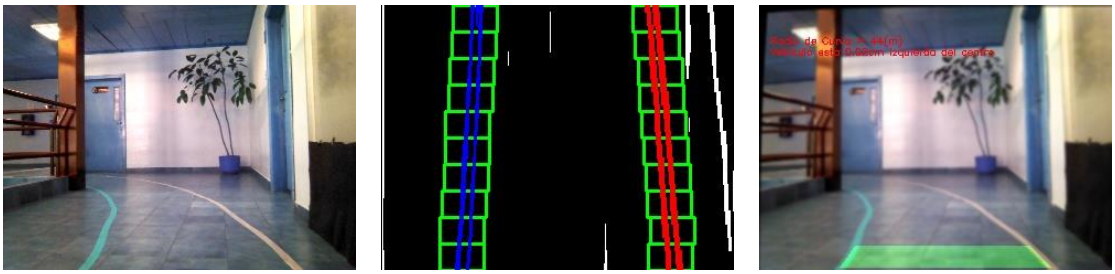


Figura 1-4: Secuencia imágenes, correspondiente a la pista de conducción del robot NAO.

Realizado por: CARRANCO, Carlos Alberto, 2017

4.2. Detección de personas con cámara del robot NAO en el desplazamiento

En la ejecución del script de Python, se puede observar en la Figura 2-4, la detección de líneas al igual que la detección de una persona. En estas imágenes que se efectúan con la cámara del robot NAO, cuya finalidad es identificar, para tomar alguna acción sobre el pedal.

Figura 2-4: Imágenes detección de personas en el trayecto y con el vehículo detenido

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 3-4 están tres personas las cuales son identificadas y el robot deberá realizar la misma acción descrita anteriormente, estas pruebas se realizan con el vehículo en desplazamiento y de igual manera detecta las líneas de la pista.

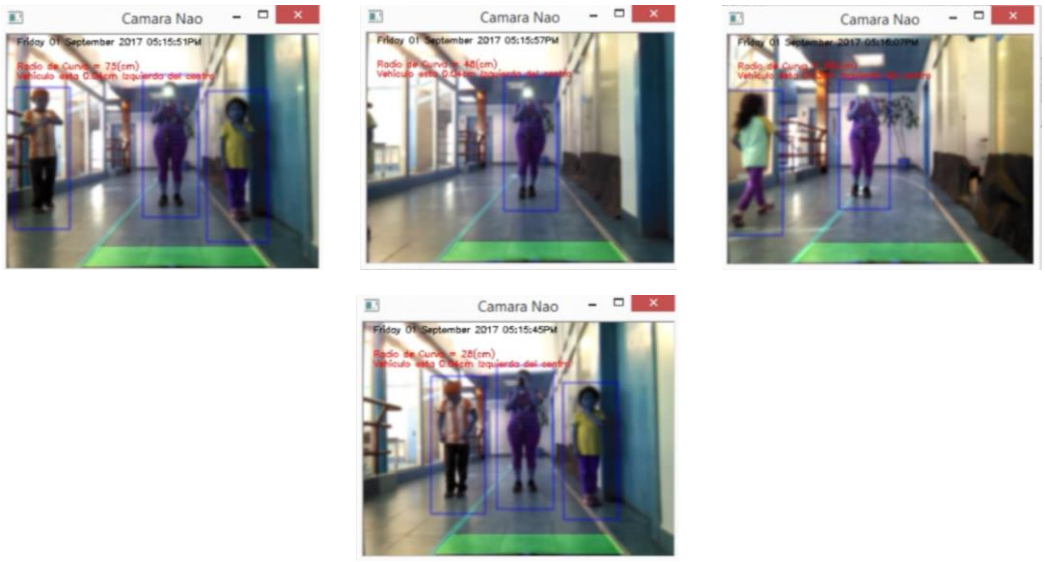


Figura 3-4: Imágenes detección de personas en el trayecto y con el vehículo desplazándose

Realizado por: CARRANCO, Carlos Alberto, 2017

4.3. Espera al detectar personas frente al vehículo.

En este test, luego de haber realizado la acción respectiva sobre el robot NAO al detectar una persona frente al coche, el robot deja de acelerar con el pie derecho y a la vez que empieza a decir un mensaje "Tenga cuidado" ver Figura 7-4, en el caso de que la persona no sea detectada Figura 4-4, el robot envía un mensaje de voz "Acelera", a continuación en las siguientes figuras se puede visualizar en el robot simulado, que se encuentra enlazado al robot real al momento que se envía los mensajes y la posición del pie del robot en las distintas situaciones, más adelante se podrá visualizar la posición del pie en relación al ángulo.

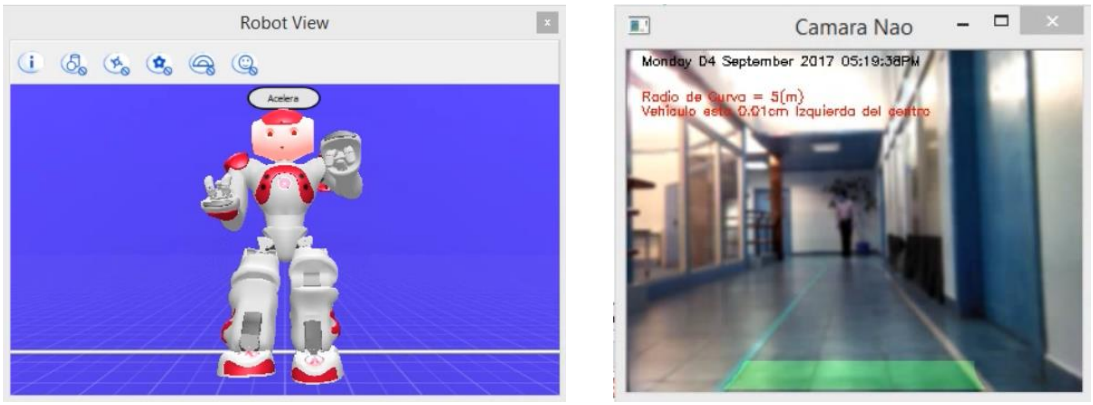


Figura 4-4: Robot acelerando sin detectar a la persona (coche detenido)

Realizado por: CARRANCO, Carlos Alberto, 2017

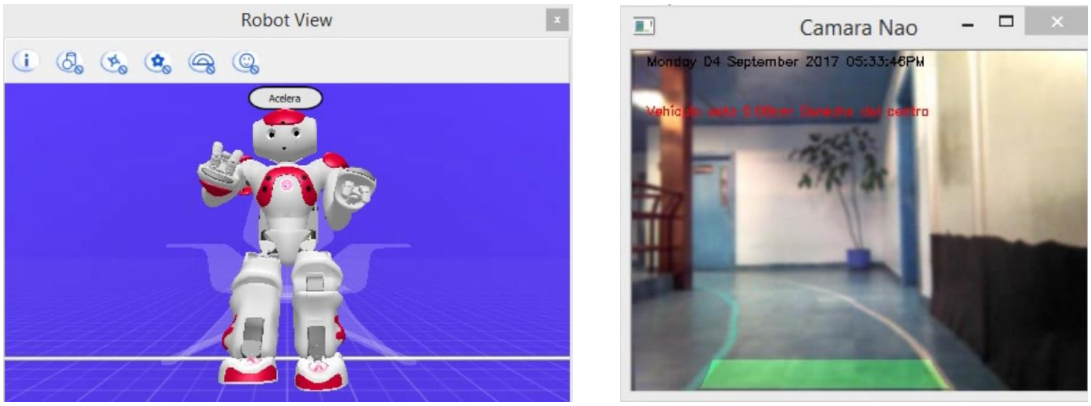


Figura 5-4: Robot acelerando no se detecta personas (coche desplazándose)

Realizado por: CARRANCO, Carlos Alberto, 2017

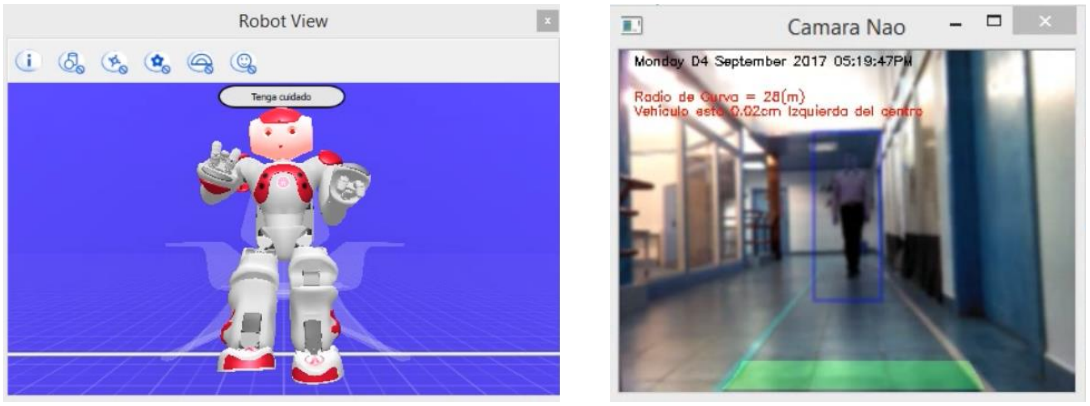


Figura 6-4: Robot sin acelerar detecta una persona, frente al coche detenido

Realizado por: CARRANCO, Carlos Alberto, 2017

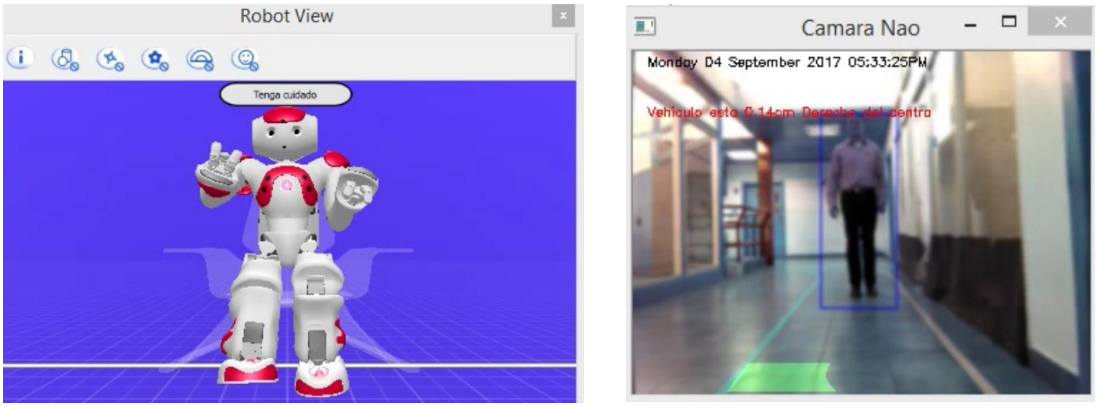


Figura 7-4: Robot sin acelerar detecta persona, frente al coche (al desplazarse)

Realizado por: CARRANCO, Carlos Alberto, 2017

4.4. Posición de brazos, pies y piernas para inicio del desplazamiento.

Antes de ejecutar el script en Python, se deberá posicionar el robot NAO en el asiento y mantener el coche en el centro de la pista, para lo cual se trabaja la referencia de movimientos de los brazos tanto izquierdo como derecho para girar el volante, en la Figura 8-4, se puede observar que se ha alineado el coche dentro de la pista y con los brazos del robot en posición recta tomando el volante.

De igual manera se toma en cuenta la posición de las piernas, así como de los pies tanto derecho como izquierdo, cabe mencionar que el pie derecho realiza la acción de aceleración y desaceleración sobre el pedal dependiendo de la acción programada. El pie izquierdo sirve de apoyo para el robot NAO sobre el coche en el momento que este pasa sentado, en la Figura 8-4, se puede visualizar la posición de las piernas con sus respectivos ángulos.

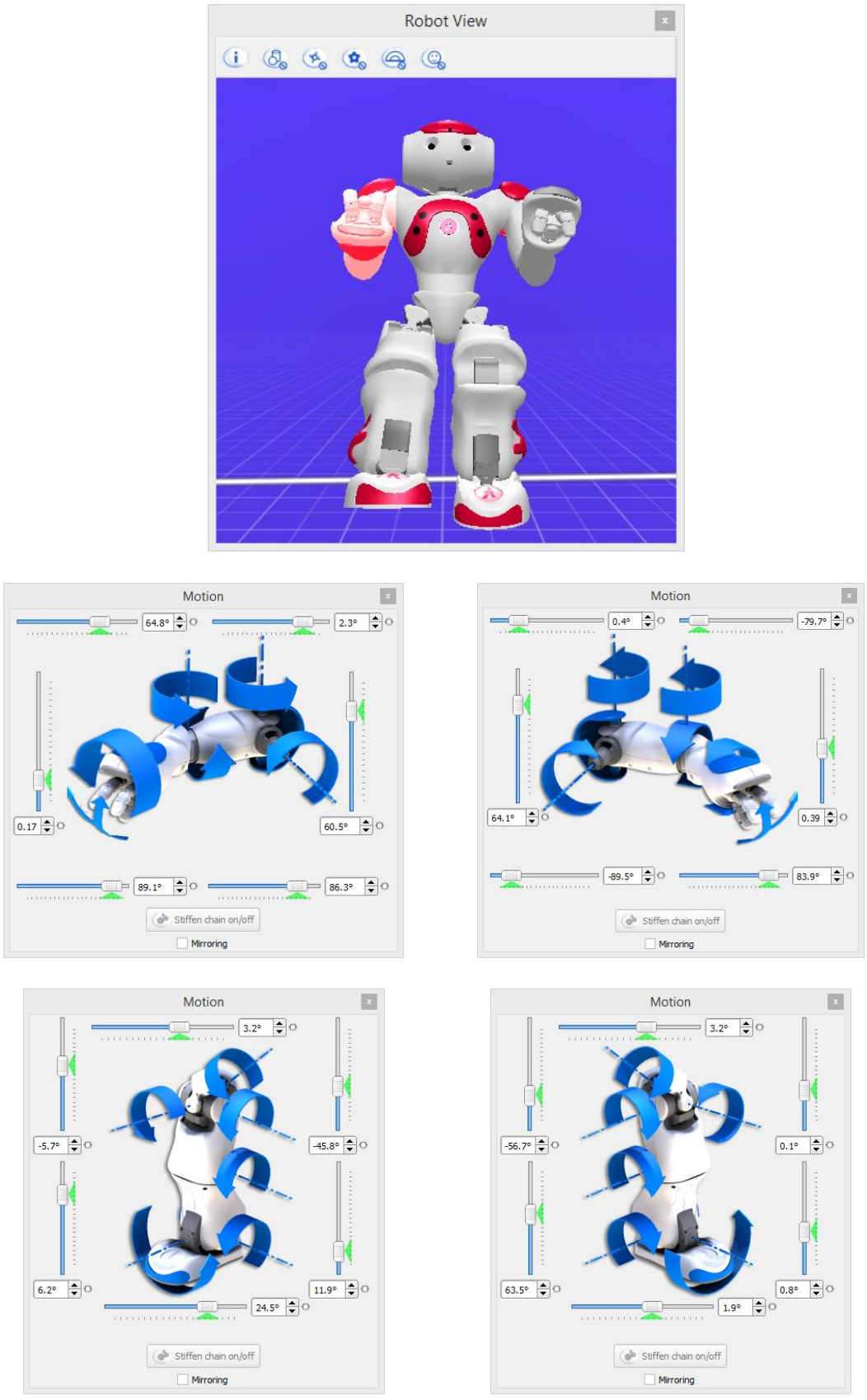


Figura 8-4: Posición sentado, brazos al volante, posición piernas y pies (sin aceleración)

Realizado por: CARRANCO, Carlos Alberto, 2017

4.5. Movimientos de la cabeza para seguimiento de líneas.

Se realiza un movimiento a la cabeza del robot para mantener a la vista las líneas de la pista y poder realizar el procesamiento y así mantener en el centro de la pista al coche.

En la Figura 9-4, se puede observar los datos tomados del movimiento de la cabeza del robot.

En el numeral uno, se puede interpretar el giro de la cabeza el robot hacia el lado izquierdo, para tratar de mantener las líneas de la pista en la trayectoria y a su vez realiza el movimiento de los brazos en la manipulación del volante. En el numeral dos, la cabeza ha realizado un giro de regreso, por lo que se puede visualizar valores negativos hasta llegar al numeral tres, que indica que la cabeza del robot NAO está alineada aproximadamente 0° con respecto a las líneas de la pista. Con relación al tiempo será dependiendo del procesamiento de la imagen, así como la velocidad que tenga el coche en alinearse a la posición central del carril.

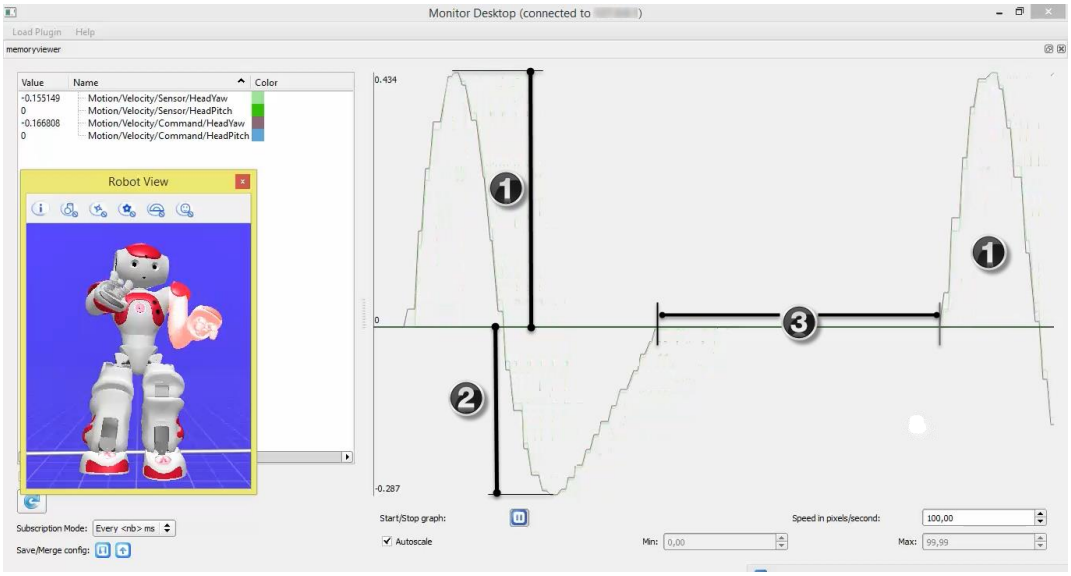


Figura 9-4: Análisis movimiento de la cabeza

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 10-4, Figura 11-4 y Figura 12-4, se puede observar cada secuencia de giro de la cabeza del robot NAO, con las respectivas posiciones, gráficas y ángulos que ha tomado dependiendo de la posición del coche con referencia al centro de la pista.

En las figuras se puede observar el dato de forma analógica, capturado por el sensor de velocidad que se encuentra acoplado a la articulación de la cabeza y su vez se visualiza el dato en valor numérico que toma con referencia al tiempo.

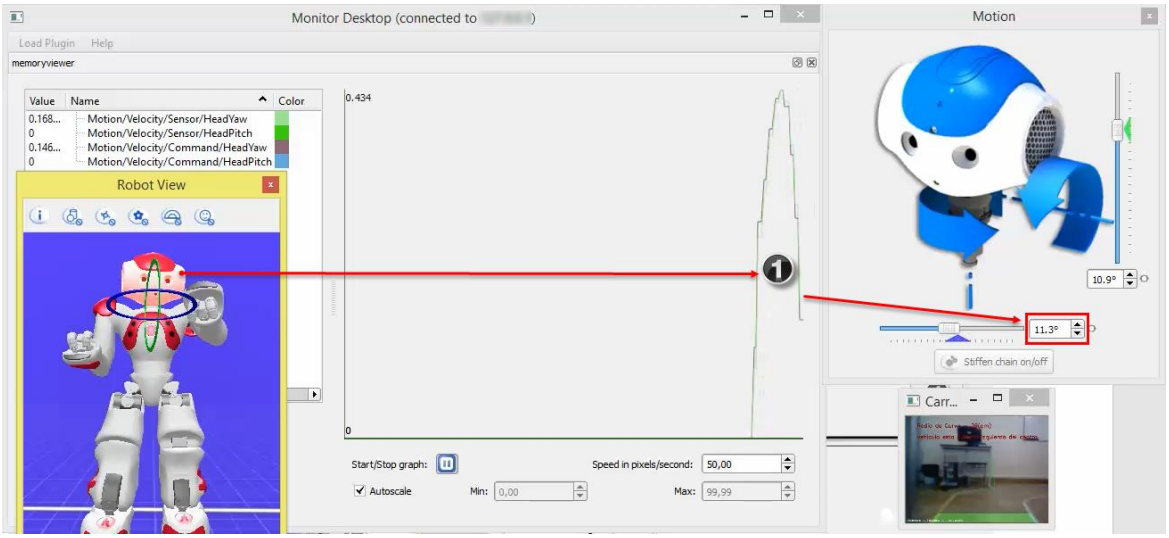


Figura 10-4: Giro a la izquierda de la cabeza con valor del ángulo

Realizado por: CARRANCO, Carlos Alberto, 2017

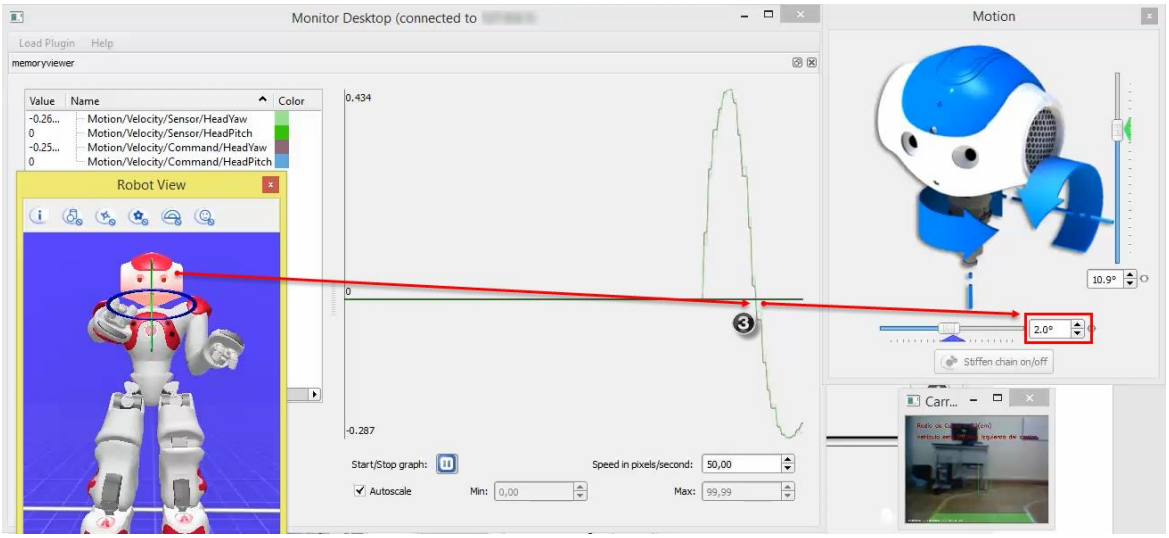


Figura 11-4: Giro al centro de la cabeza con valor del ángulo

Realizado por: CARRANCO, Carlos Alberto, 2017

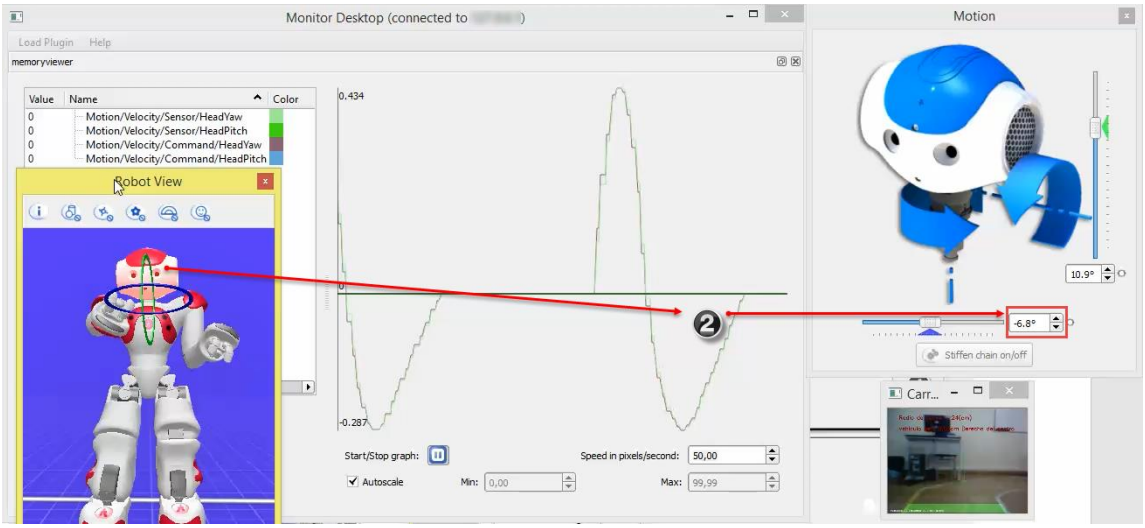


Figura 12-4: Giro al lado derecho de la cabeza con valor negativo del ángulo

Realizado por: CARRANCO, Carlos Alberto, 2017

4.6. Movimiento de pie derecho aceleración y desaceleración del coche.

En la Figura 13-4 y Figura 14-4, se puede visualizar el movimiento con referencia al tiempo que realizó el pie derecho sobre el pedal para acelerar o desacelerar, esta operación tiene relación con las acciones que antes se mencionaron con respecto, a que si se tiene que desplazar con el vehículo o al momento que se detecte una persona frente al coche.

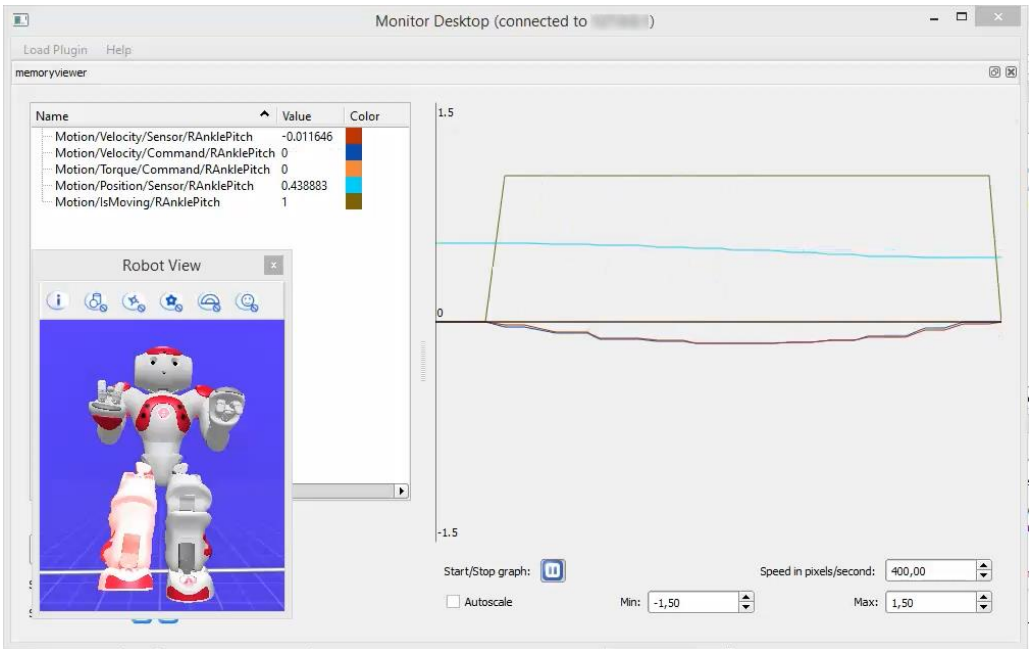


Figura 13-4: Posición desaceleración pie derecho

Realizado por: CARRANCO, Carlos Alberto, 2017

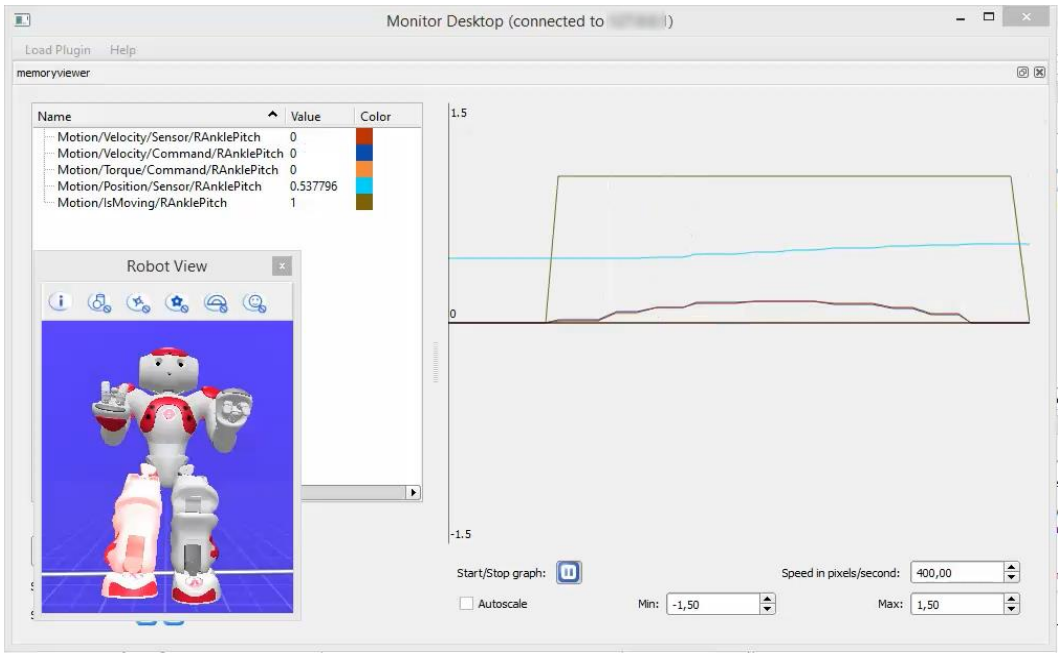


Figura 14-4: Posición aceleración pie derecho

Realizado por: CARRANCO, Carlos Alberto, 2017

A la Figura 15-4, hace referencia al valor que el sensor del pie derecho (Sensor/RAnklePitch), va tomando mientras el pie realiza la acción sobre el pedal de aceleración, se puede visualizar que toma distintos valores mientras (IsMoving/RAnklePitch) se active ya que indica el movimiento del motor de esa articulación.

Sensor/RAnklePitch	Ángulo de la articulación (en radianes). Actualización del contenido: El sensor utilizado es un codificador rotatorio magnético (MRE), utilizado como potenciómetro. Es un valor preciso de 12bits (de 0 a 4095) en rad.
IsMoving/RAnklePitch	Indica el momento que se activa el motor del pie derecho para el movimiento valores entre 0 y 1. 1 = activado; 0 = desactivado

Figura 15-4: Leyenda del sensor pie derecho

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 16-4 se observa que los valores del Sensor/RAnklePitch ha cambiado, debido a que el robot Nao mediante la cámara no ha detectado ninguna persona u obstaculo al frente del coche por lo que se procede acelerar esto hasta que IsMoving/RAnklePitch este en el valor de uno.

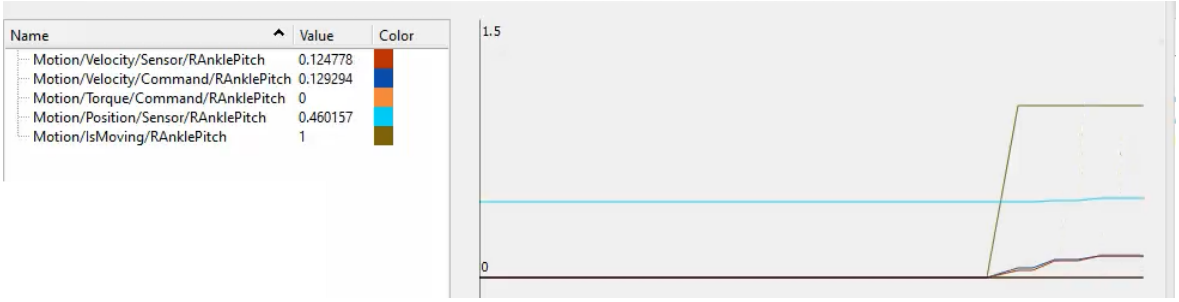


Figura 16-4: Inicio de aceleración sobre el pie derecho

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 16-4, Figura 17-4 se observa que los valores del Sensor/RAnklePitch ahora son de cero ya en el momento del desplazamiento del coche el robot Nao mediante la cámara ha detectado alguna persona u obstaculo al frente, por lo que se procede ha no relizar la accion de acelerado sobre el pedal por lo que IsMoving/RAnklePitch tiene un valor de 0.

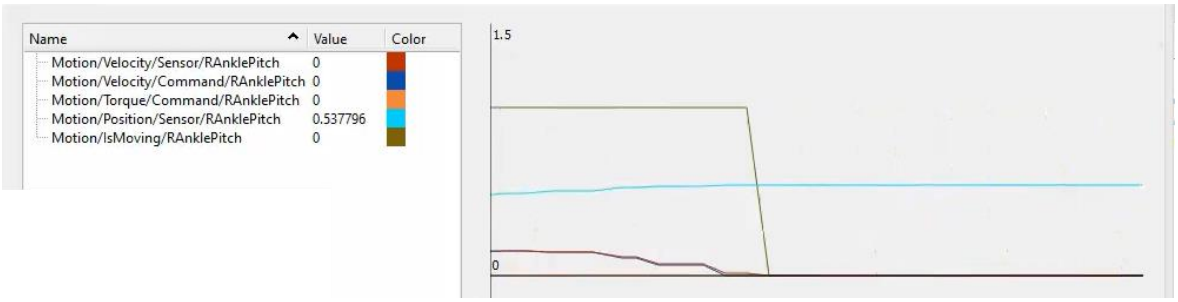


Figura 17-4: Finalización de aceleración sobre el pie derecho

Realizado por: CARRANCO, Carlos Alberto, 2017

4.7. Movimiento de brazos en el desplazamiento del coche.

En el momento que el vehículo se encuentre desplazándose en la pista, habrá instantes en que el robot NAO, deberá corregir la trayectoria del coche, tratando de mantenerlo dentro de la pista. Teniendo en cuenta el valor asignado para el centro del mismo, está relacionado con la programación de la detección de líneas, el que se puede visualizar en el momento de ejecutar el script de Python.

A continuación, podemos observar las distintas posiciones que el robot va tomando con respecto al volante que es manipulado por los brazos y manos del robot NAO, sea iniciando las ruedas en

posición recta, o en el desplazamiento del coche corregir al lado derecho o izquierdo dependiendo de la posición del coche con respecto a la cámara del robot.

Como se describió en la Figura 15-4, refiriendo al significado de las gráficas, datos que son tomados de los sensores que están ubicados en las articulaciones, de igual manera se lo hace con la de los brazos. Se observa en la Figura 18-4 hasta la Figura 24-4 las diversas posiciones que tiene cada brazo, visto el dato en forma de ángulo, así como el dato que el sensor está captando en [radianes].

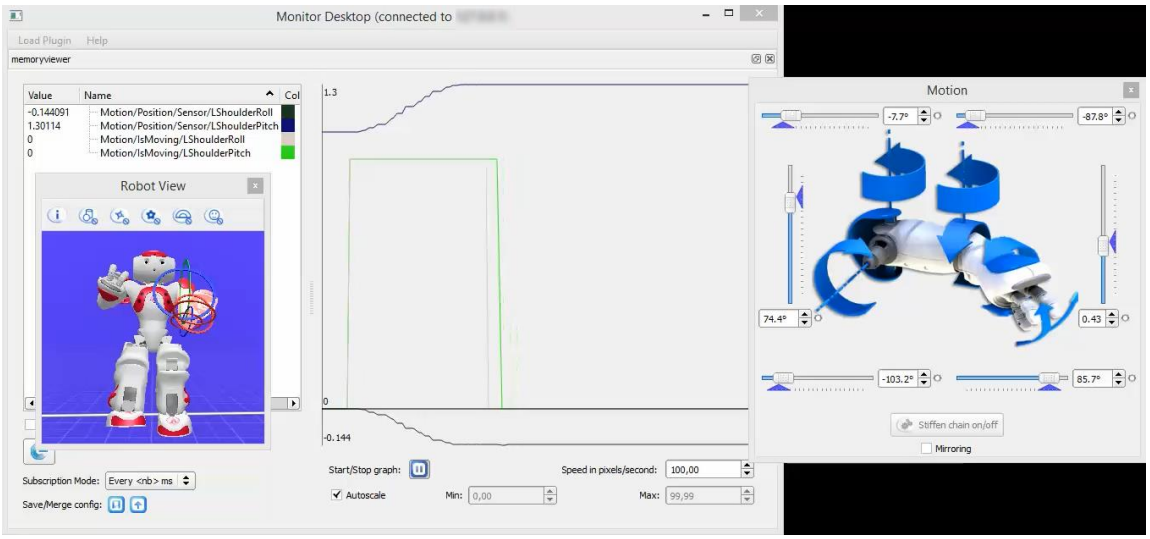


Figura 18-4: Análisis de posición del brazo izquierdo

Realizado por: CARRANCO, Carlos Alberto, 2017

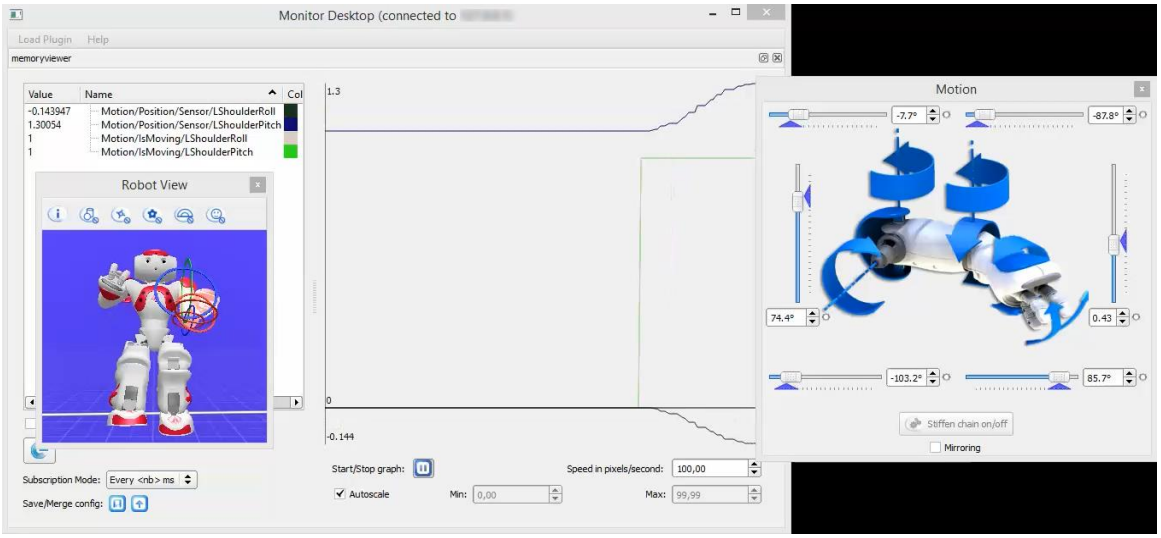


Figura 19-4: Inicio de giro al lado izquierdo (brazo izquierdo)

Realizado por: CARRANCO, Carlos Alberto, 2017

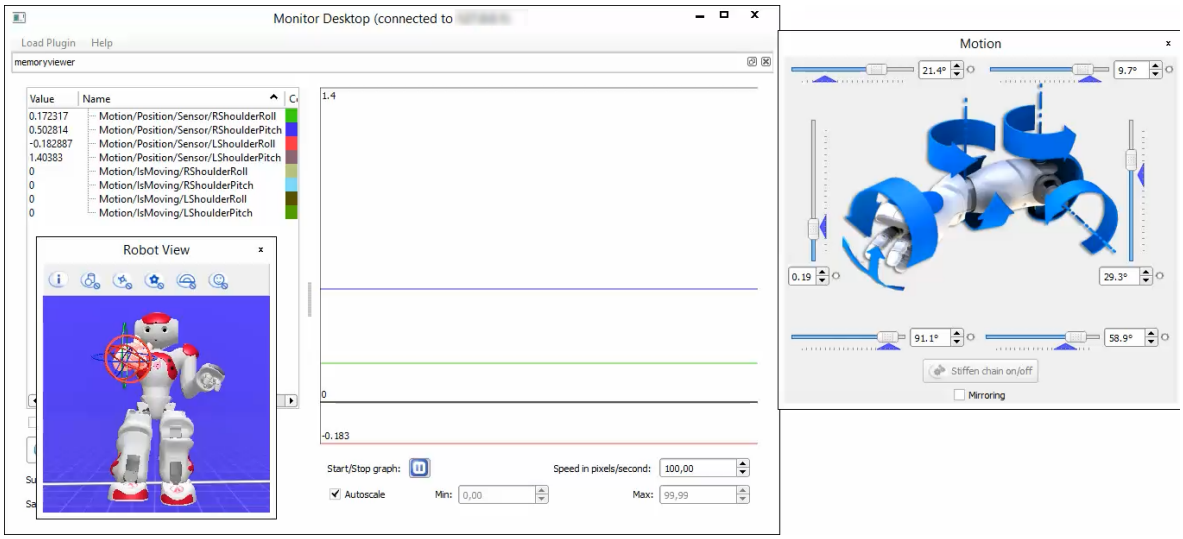


Figura 20-4: Posición de brazos a la izquierda

Realizado por: CARRANCO, Carlos Alberto, 2017

En la leyenda que muestra al capturar los datos de los sensores se puede observar en la Figura 21-4 valores en radianes de los Sensor/RShoulderRoll, Sensor/RShoulderPitch, Sensor/LShoulderRoll, Sensor/RShoulderPitch. Los valores de IsMoving/RShoulderPitch, IsMoving/LShoulderPitch son de activación del actuador para realizar el movimiento que depende la captura de la imagen procesada.

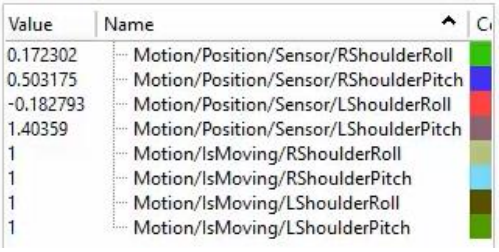


Figura 21-4: Activación de los motores para mover los brazos

Realizado por: CARRANCO, Carlos Alberto, 2017

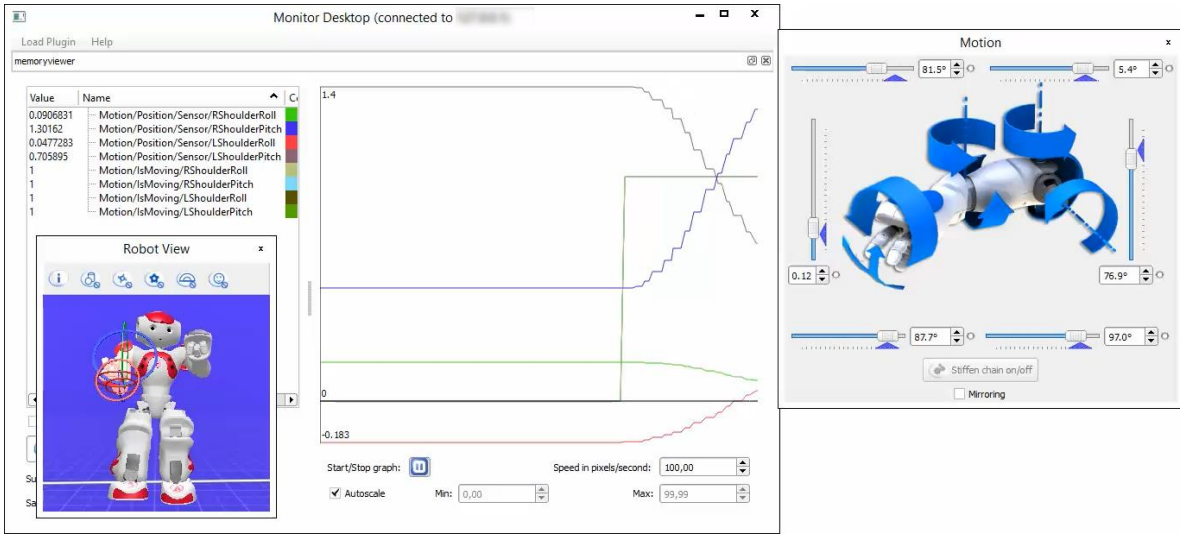


Figura 22-4: Giro de volante a la derecha grafica de ambos brazos

Realizado por: CARRANCO, Carlos Alberto, 2017

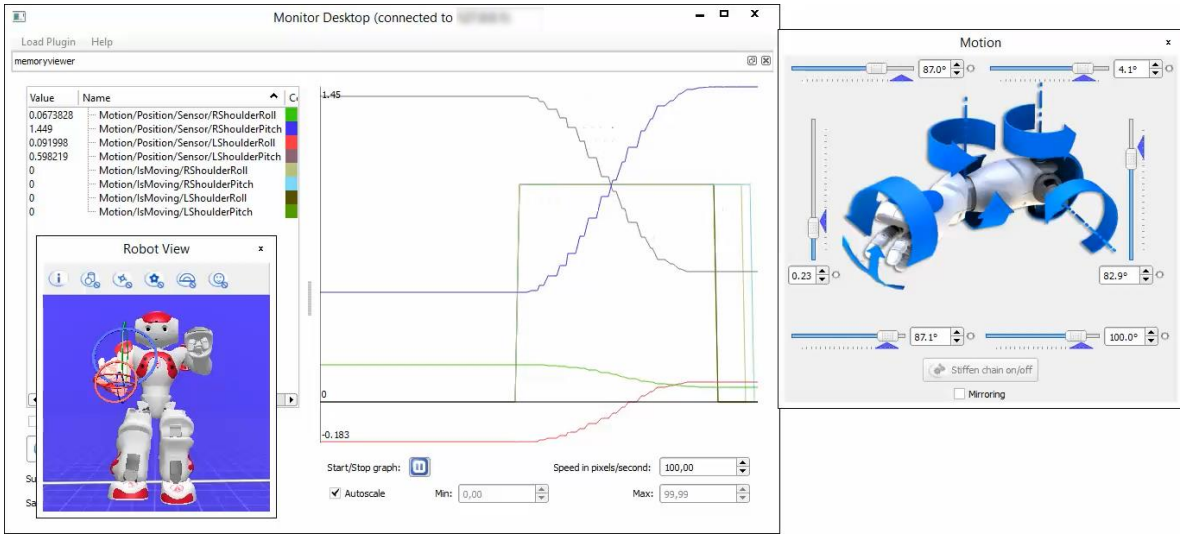


Figura 23-4: Giro completo a la derecha por los brazos (ángulos brazo derecho)

Realizado por: CARRANCO, Carlos Alberto, 2017

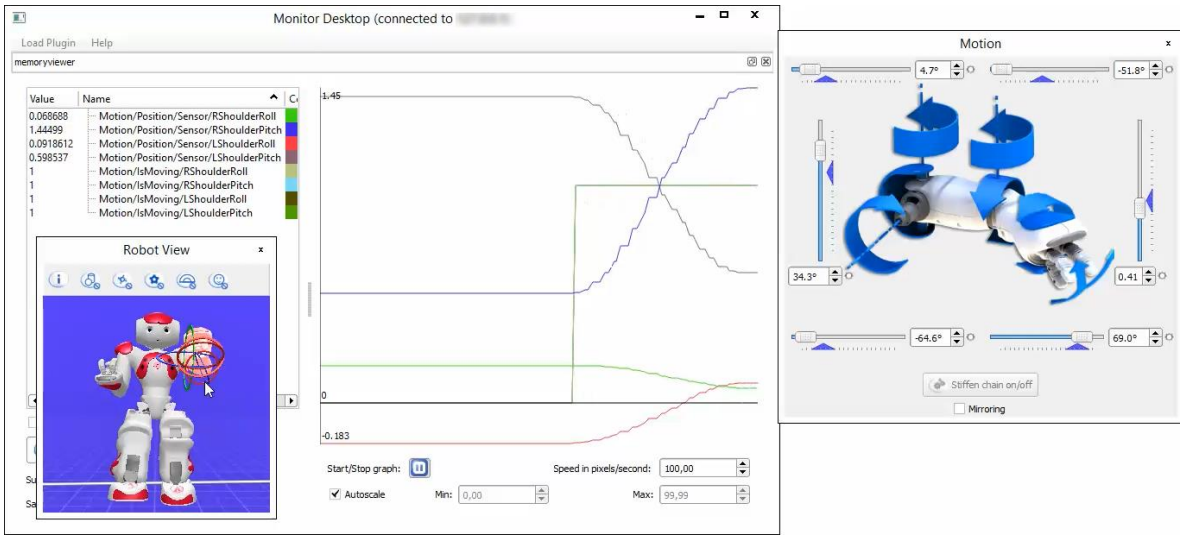


Figura 24-4: Giro completo a la derecha por brazos (ángulos brazo izquierdo)

Realizado por: CARRANCO, Carlos Alberto, 2017

En la Figura 25-4, se observa que los motores se han activado para realizar un giro al lado izquierdo mientras sigue visualizando las líneas de la pista como se dijo anteriormente para tratar de mantener en el centro del carril al coche.

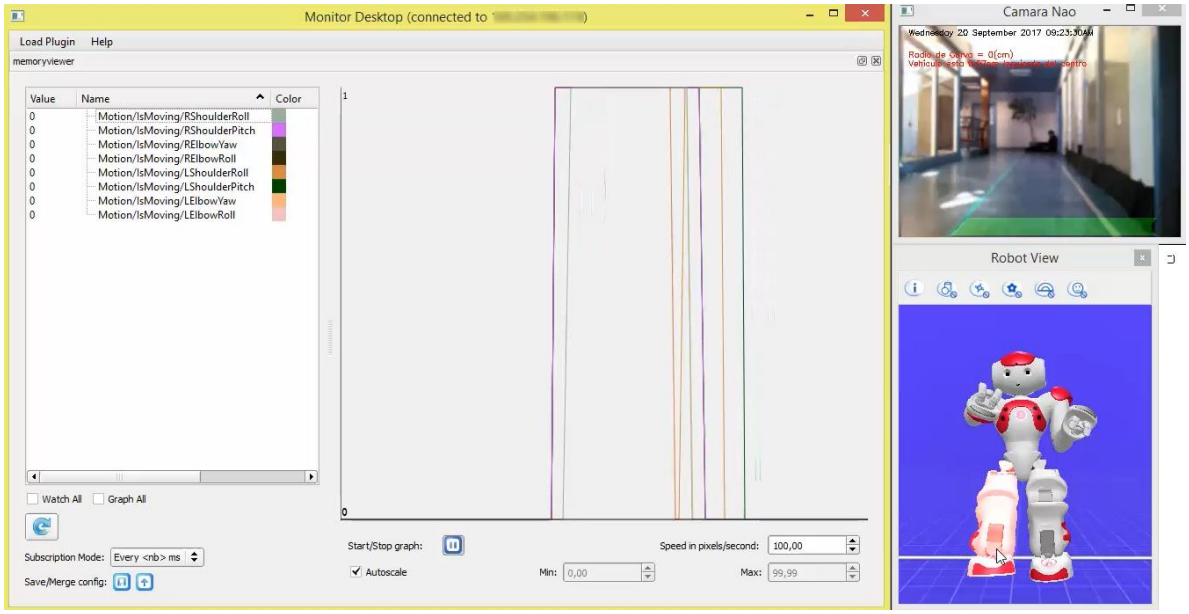


Figura 25-4: Activación y desactivación de los motores de brazos

Realizado por: CARRANCO, Carlos Alberto, 2017

A continuación, se puede observar en la Figura 26-4, Figura 27-4, Figura 28-4, Figura 29-4, el proceso de tratamiento de la imagen, así como la acción que realizan las articulaciones del robot NAO sobre el acelerador al igual que los movimientos de los brazos.

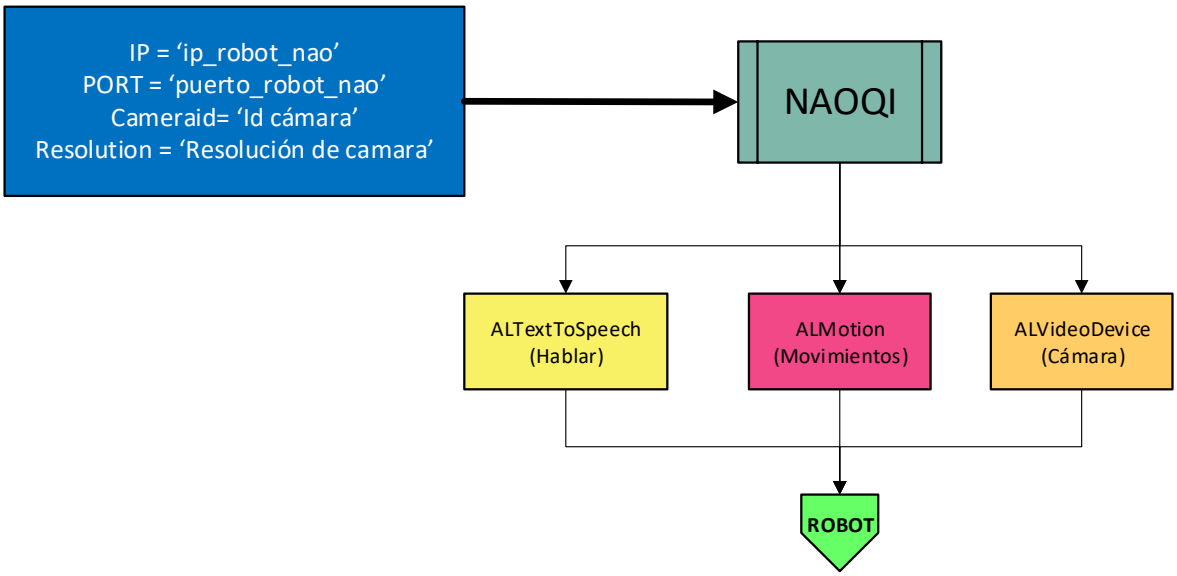


Figura 26-4: Estructura de datos con el NAOQI

Realizado por: CARRANCO, Carlos Alberto, 2017

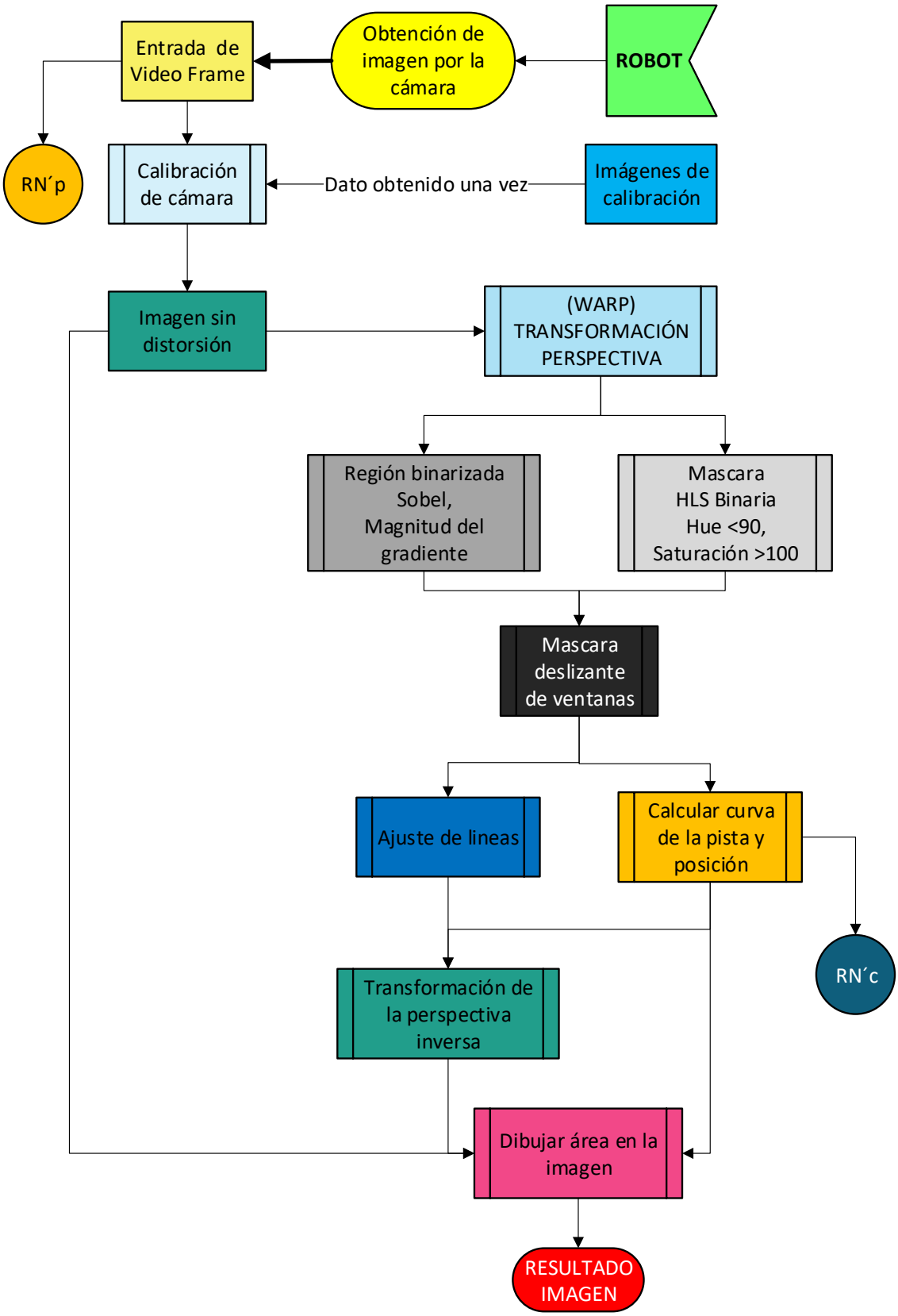


Figura 27-4: Estructura procesamiento imagen para realizar acciones de corrección del coche

Realizado por: CARRANCO, Carlos Alberto, 2017

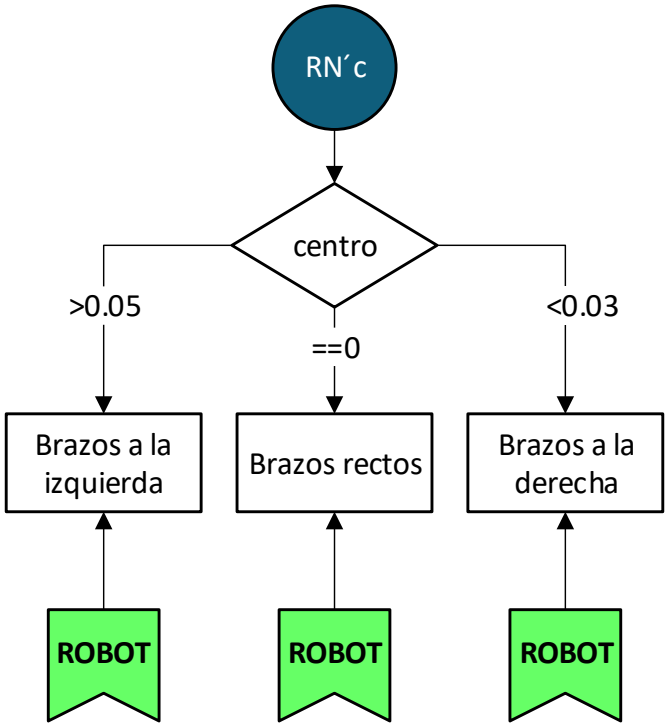


Figura 28-4: Comparación datos con procesamiento imagen y movimientos del robot NAO

Realizado por: CARRANCO, Carlos Alberto, 2017

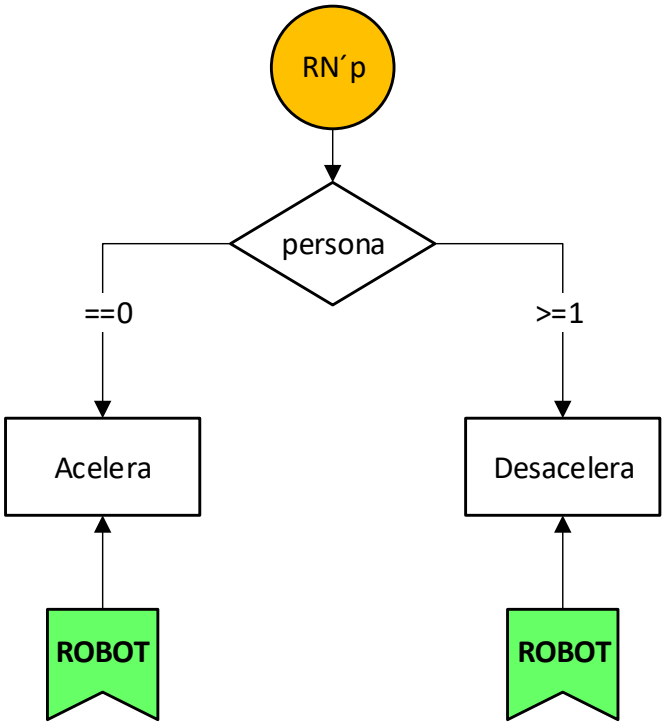


Figura 29-4: Comparación dato al detectar una persona, en la acción del pedal del coche

Realizado por: CARRANCO, Carlos Alberto, 2017

En la **Tabla** -3, se muestra los datos obtenidos en las diferentes posiciones del robot NAO de cada articulación, dependiendo de la maniobra que el robot realice , siendo que se tome una recta Figura 31-4, girar a la derecha o izquierda Figura 32-4, así mismo la posición del pie derecho en la posición de aceleración o desaceleración Figura 33-4, en esta última siendo que se ha detectado a una persona para evitar que el vehículo se desplace.

Los datos se pueden ver graficados en Figura 31-4, Figura 32-4, Figura 33-4, donde los puntos indican la posición de las distintas articulaciones que el robot NAO posee.

Tabla 1-4: Datos de los sensores por articulación (motores encendidos)

	Stiffness 1					
	CUERPO	PIE DERECHO		BRAZOS		
BodyName	Sentado	Pie Acelerado	Pie Desacelerado	Giro derecha	Recta	Giro Izquierda
HeadYaw	0,000	0,000	0,200	0,000	0,000	0,000
HeadPitch	-0,170	-0,170	-0,200	-0,170	-0,170	-0,170
LShoulderPitch	1,493	1,301	1,472	0,598	1,110	1,301
LShoulderRoll	0,216	-0,144	0,185	0,092	0,000	-0,144
LElbowYaw	-1,198	-1,801	-1,194	-1,128	-1,563	-1,801
LElbowRoll	-0,394	-1,540	-0,410	-0,900	-1,400	-1,540
LWristYaw	0,100	1,500	0,100	1,201	1,465	1,500
LHipYawPitch	0,057	0,057	-0,170	0,057	0,057	0,057
LHipRoll	-0,001	-0,001	0,100	-0,001	-0,001	-0,001
LHipPitch	-0,996	-0,996	0,130	-0,996	-0,996	-0,996
LKneePitch	1,114	1,114	-0,090	1,114	1,114	1,114
LAnklePitch	0,032	0,032	0,090	0,032	0,032	0,032
LAnkleRoll	0,018	0,018	-0,130	0,018	0,018	0,018
RHipYawPitch	0,057	0,057	-0,170	0,057	0,057	0,057
RHipRoll	-0,092	-0,092	-0,100	-0,092	-0,092	-0,092
RHipPitch	-0,804	-0,804	0,130	-0,804	-0,804	-0,804
RKneePitch	0,216	0,216	-0,090	0,216	0,216	0,216
RAnklePitch	0,428	0,562	0,090	0,436	0,436	0,562
RAnkleRoll	0,104	0,104	0,130	0,104	0,104	0,104
RShoulderPitch	1,484	0,675	1,472	1,449	1,053	0,675
RShoulderRoll	-0,215	-0,206	-0,185	0,067	0,047	-0,206
RElbowYaw	1,199	1,015	1,194	1,750	1,511	1,015
RElbowRoll	0,398	0,850	0,410	1,519	1,135	0,850
RWristYaw	0,100	1,720	0,100	1,519	1,557	1,720
LHand	0,000	0,432	0,300	0,414	0,394	0,432
RHand	0,300	0,283	0,300	0,231	0,173	0,283

Realizado por: CARRANCO, Carlos Alberto, 2017

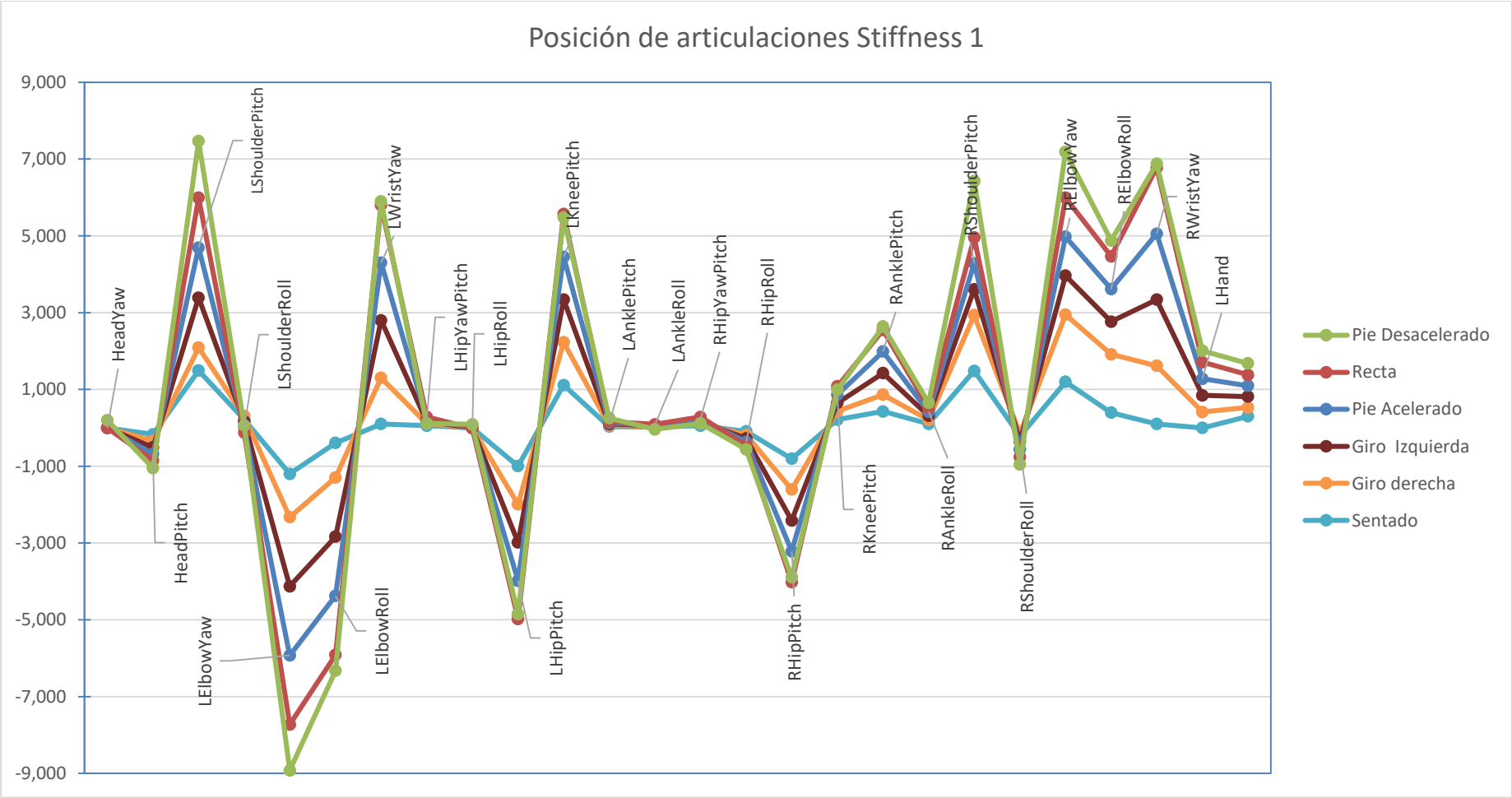


Figura 30-4: Datos de los sensores por articulación (motores encendidos)

Realizado por: CARRANCO, Carlos Alberto, 2017

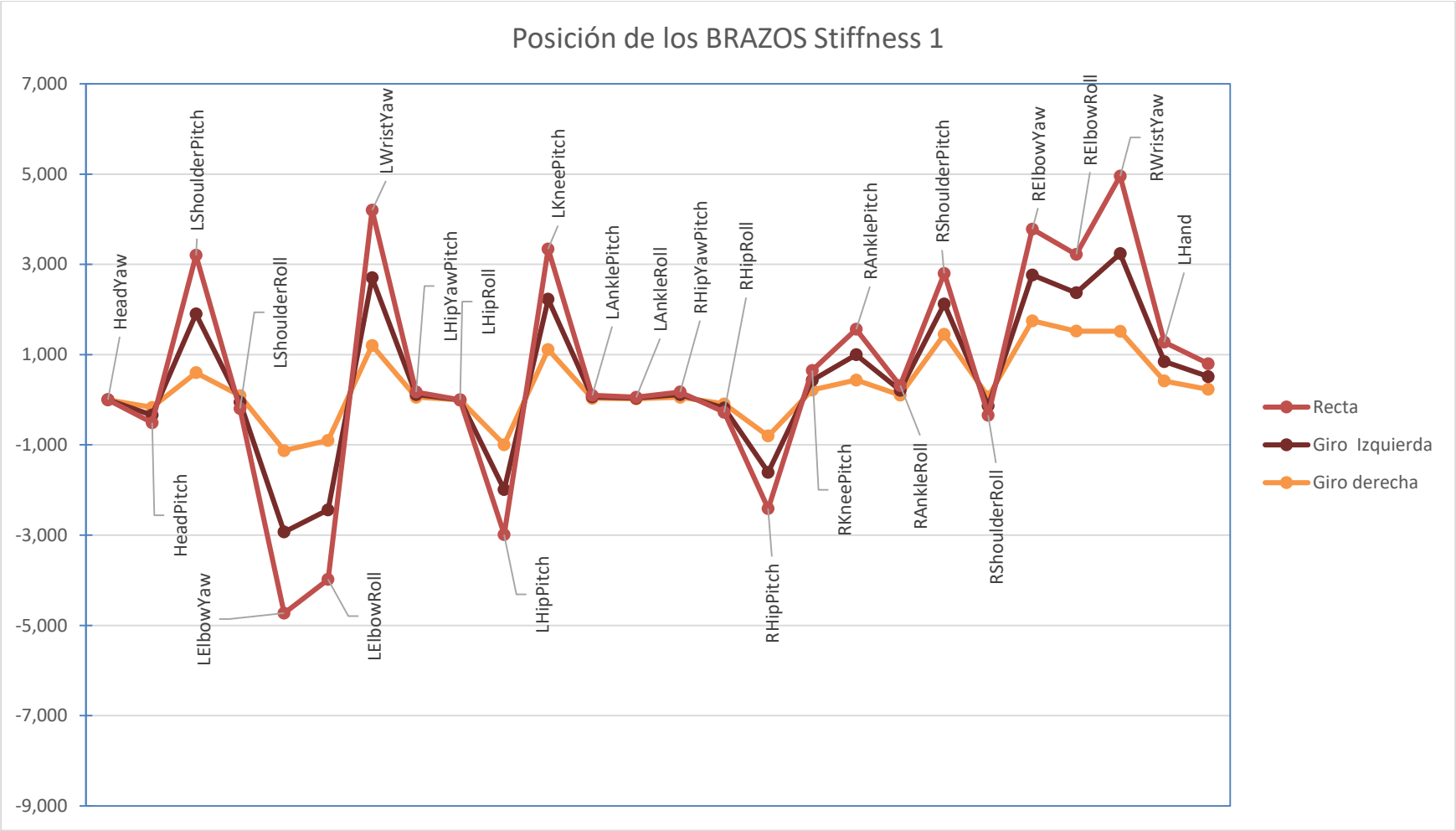


Figura 31-4: Datos de los sensores articulación (BRAZOS)

Realizado por: CARRANCO, Carlos Alberto, 2017

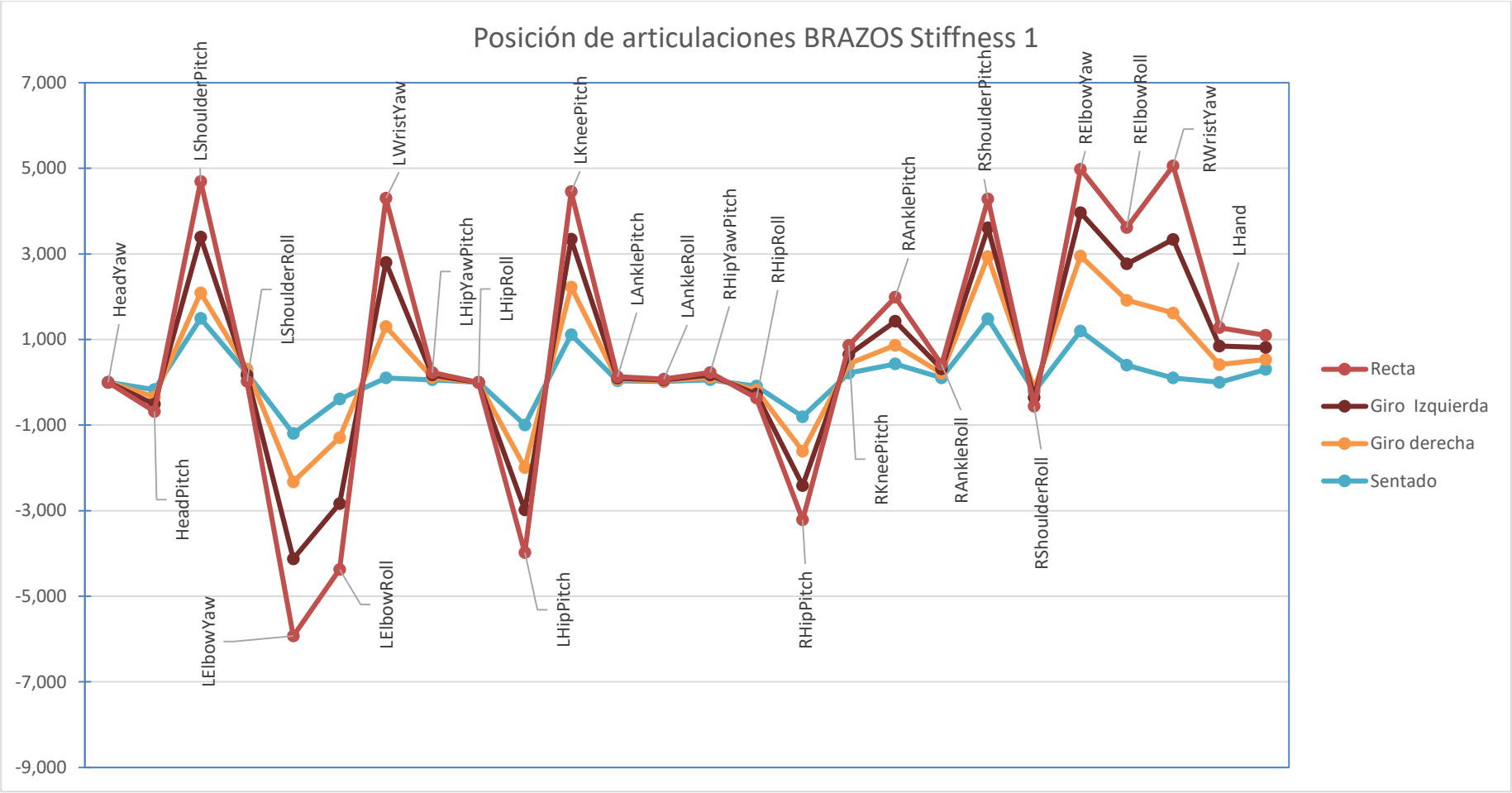


Figura 32-4: Datos de los sensores articulación (BRAZOS y CUERPO)

Realizado por: CARRANCO, Carlos Alberto, 2017

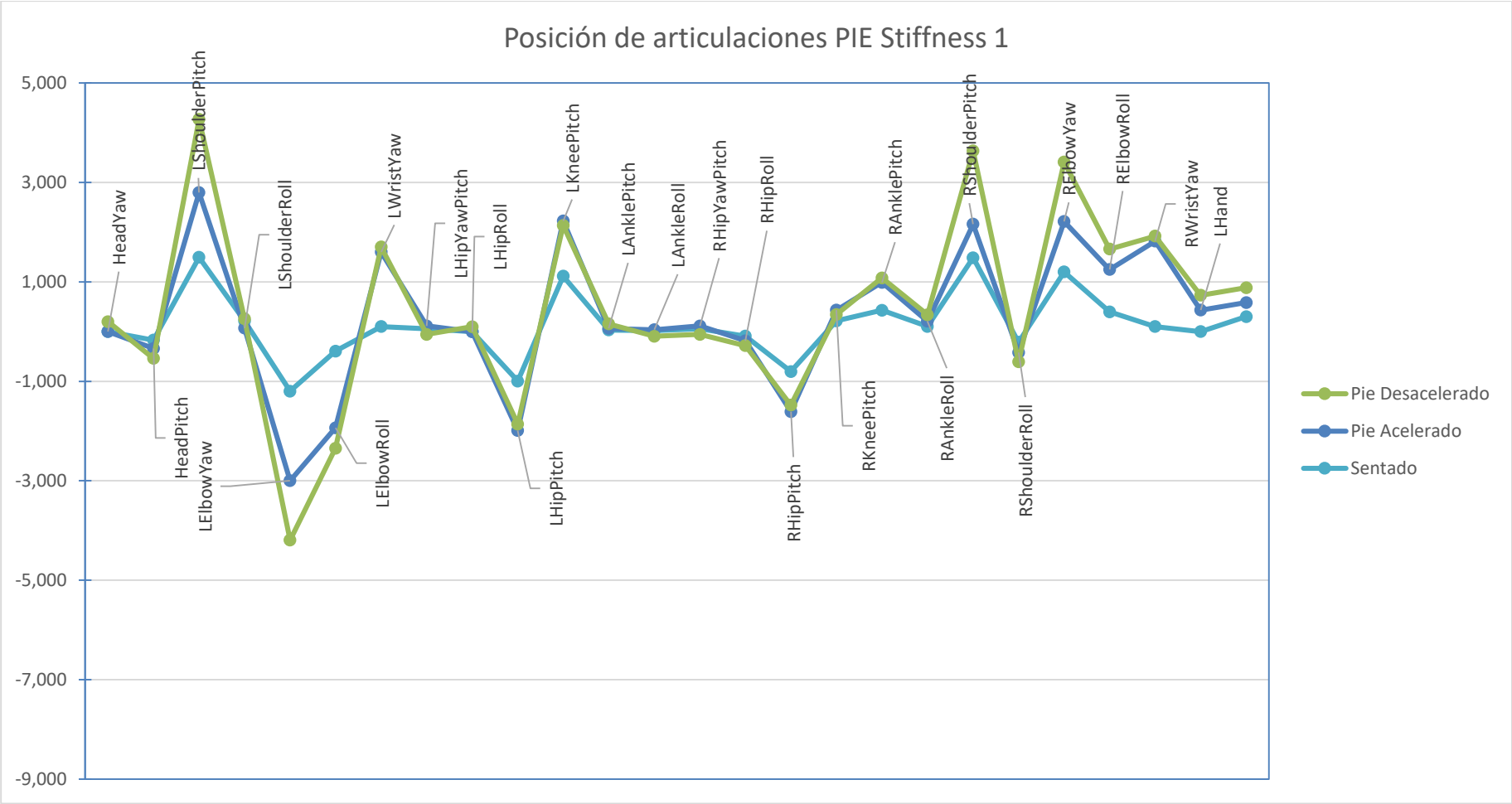


Figura 33-4: Datos de los sensores articulación (PIE)

Realizado por: CARRANCO, Carlos Alberto, 2017

CONCLUSIONES

La hipótesis que se planteó al iniciar el proyecto, de diseñar e implementar funciones de conducción de un coche por el robot humanoide NAO con el uso de los dispositivos propios, brindan un aporte para trabajos futuros con el uso de visión artificial. Utilizando las distintas aplicaciones con las que se maneja el robot humanoide NAO, tal como: el procesamiento de imagen, y el control de movimientos de las articulaciones en los mandos del coche.

El código base implementado en la detección de líneas, resulto favorable en su ejecución con el robot humanoide NAO, confirmando así la capacidad de realizar la tarea de conducción de un coche eléctrico para niños, concluyendo así que se puede dar más funcionalidad en la navegación autónoma, ya que determina el camino por donde el coche debe circular y con la detección de personas.

En la elaboración del proyecto se alcanzó los objetivos planteados sobre:

El manejo de los movimientos del robot NAO, corrigiendo su trayectoria o las maniobras que debe realizar mediante el uso de visión artificial, en el momento de la conducción.

El tratamiento de imágenes que se obtuvieron mediante la cámara del robot humanoide NAO, es un factor indispensable, por lo que los procesos que se ejecutaron en tiempo real, son bastante favorables en las pruebas que se establecieron.

El reconocimiento de la pista mediante la cámara del robot humanoide NAO, fue satisfactorio con el algoritmo que se implementó, en conjunto con los movimientos del robot para el control del coche en el momento del desplazamiento.

Se hace el uso de los dispositivos propios del robot, siendo uno de ellos, la cámara superior del robot humanoide NAO, para identificación de la pista, identificación de personas que se encuentren en la trayectoria del vehículo. Seguido de la parte mecánica del robot para las maniobras que este debe realizar.

Siendo Python unos de los softwares más propicios en la programación de visión artificial, así como, la relación con la parte del mecanismo del robot humanoide NAO para el control ha sido un punto para el manejo en conjunto de ambos sistemas.

RECOMENDACIONES

Al finalizar la investigación se considera interesante, plantear nuevas aplicaciones relacionados con el robot NAO en la parte mecánica (articulaciones) y visión artificial.

Se recomienda ampliar la investigación con el mejoramiento del control del vehículo, en lo posible el mando sea más manipulable, debido a la estructura mecánica que es un poco compleja, el llevar acabo movimientos que limitan a ciertas articulaciones del robot NAO.

La implementación de tarjetas, para un rápido procesamiento de imágenes, serían de gran ayuda para que el tiempo de reacción sea minimizado, en los movimientos propios del robot, en el desplazamiento con el coche.

En el desarrollo, se evidencia que se puede realizar una aplicación muy interesante con el robot NAO, siendo conductor de un coche, principalmente, para niños con capacidades especiales, dependiendo del nivel de discapacidad sea: intelectual, motriz, sensorial, múltiples y otras.

Ubicarlo en una plataforma móvil, para el traslado de objetos en distancias bastante extensas, ahorrando energía con el uso del robot humanoide NAO, donde este sea situado para trabajar, así como guía de museo, en aeropuertos, hospitales, son bastante llamativos para interactuar con los seres humanos.

Mejorar el sistema de visión, para la identificación de señales de tránsito, características de vehículos que se encuentren en el entorno, con el robot NAO y diversas situaciones que a la mayoría de personas les ocurre tras el volante de un automotor.

BIBLIOGRAFÍA

Aldebaran. (2017). Joints. Recuperado de http://doc.aldebaran.com/2-1/family/robots/joints_robot.html

Ariagna. (2015). Modelo HSL. Recuperado de https://www.ecured.cu/Modelo_HSL

Ariffin, I. M., Rasidi, A. I. H. M., Mohamed, Z., Miskam, M. A., Amin, A. T. M., Omar, A. R., & Omar, A. R. (2015). Sensor Based Mobile Navigation Using Humanoid Robot Nao. *Procedia Computer Science*, *76*, 474–479. https://doi.org/10.1016/j.procs.2015.12.319

Artificial, V. (2012). Aplicación práctica de la visión artificial en el control de procesos industriales, 26. Recuperado de http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf

Centro Zaragoza. (2016). Vehículo autónomo, 4. Recuperado de http://www.centro-zaragoza.com:8080/web/sala_prensa/revista_tecnica/hemeroteca/articulos/R70_A7.pdf

Cordoba, C. (2011). Introducción a Python. Recuperado de http://ccordoba12.github.io/modelacion/Texto/Introduccion.html#importar-librerias

DAF Trucks N.V. (2018). Sistema de advertencia de salida de carril. Recuperado de http://www.daf.es/~/media/files/daf trucks/trucks/euro 6/general/safety/ldws/daf-lane-departure-warning-system-66033-es.pdf

E Fierro, J., Pamanes, A., A Santibáñez, V., Ruiz, G., & Ollervides, J. (2016). *AMRob Journal, Robotics: Theory and Applications Condiciones para una marcha elemental del robot NAO Regular Paper*. *AMRob Journal, Robotics: Theory and Applications* (Vol. 4).

Enrique, C. (2016). Aprendiendo Arduino. Recuperado de https://aprendiendoarduino.wordpress.com/category/pwm/

FITSA. (2007). Descripción del Sistema de Alerta de Cambio Involuntario de Carril y evidencias científicas de su eficacia. Recuperado de http://www.nextautomotivesafety.com/pdf/cambio_de_carril.pdf

Florencia, Lanzaro; Fuentes, M. (2014). CALIBRACIÓN Y POSICIONAMIENTO 3D. Recuperado de http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2014/candombe/calibracion.html

Furfaro, A. (2010). Manejo de Bibliotecas Opencv. Recuperado de http://www2.electron.frba.utn.edu.ar/~afurfaro/Info1/Opencv/opencv.pdf

GeekMag. (2012). Nao car: le petit robot d'Aldebaran conduit une voiture. Recuperado de http://www.geekmag.fr/nao-car-le-petit-robot-daldebaran-conduit-une-voiture/

González, J. C. (2015). Yamaha está desarrollando un espectacular robot capaz de conducir motos de alta cilindrada. Recuperado de http://www.xataka.com/robotica-e-ia/yamaha-esta-desarrollando-un-espectacular-robot-capaz-de-conducir-motos-de-alta-cilindrada

González, J. I. (2003). *Estudio experimental de métodos de calibración y autocalibración de cámaras*. Recuperado de http://mozart.dis.ulpgc.es/Gias/josep/TesisJosep.pdf

Gurjashan Singh Pannu, Pritha Gupta, M. D. (2015). Design and Implementation of Autonomous Car using Raspberry Pi. Recuperado de http://research.ijcaonline.org/volume113/number9/pxc3901789.pdf

Gutiérrez González, L., Ribacoba Menoyo Tutores, I., & Etxebarria Ecenarro Alfredo García Arribas, V. (2007). *Montaje y puesta a punto de un robot humanoide*.

hahnsang. (2016a). Direction of the Gradient. Recuperado de https://inspro9.wordpress.com/2016/12/15/code-direction-of-the-gradient/

hahnsang. (2016b). Lane Curvature. Recuperado de https://inspro9.wordpress.com/2016/12/14/lane-curvature/

hahnsang. (2016c). Magnitude of the Gradient. Recuperado de https://inspro9.wordpress.com/2016/12/15/magnitude-of-the-gradient/

Herrera, A. (2014). MODELOS DE COLOR (RGB, CMYK, HSV/HSL). Recuperado de https://ahenav.com/2014/04/09/modelos-de-color/

Kofinas, N. (2012). *Forward and Inverse Kinematics for the NAO Humanoid Robot*. Technical University of Crete, Greece. Recuperado de https://www.cs.umd.edu/~nkofinas/Projects/KofinasThesis.pdf

Lakshmi, Rajya, Mounika, J. (2010). Image Processing Lane Departure System With Edge Detection Technique Using Hough Transform – a . Rajya Lakshmi & J . Mounika. Recuperado de http://www.yuvaengineers.com/image-processing-lane-departure-system-with-edge-detection-technique-using-hough-transform-a-rajya-lakshmi-j-mounika/

Lluch, Z. F. (2016). Implementación del juego de las damas en un robot NAO. Recuperado de https://riunet.upv.es/bitstream/handle/10251/59555/FERRER - Implementación del juego de las damas en un robot NAO.pdf?sequence=1

MAPFRE. (s/f). Campo visual y conducción. Recuperado de https://www.fundacionmapfre.org/fundacion/es_es/programas/seguridad-vial/medicos/temas-

conduccion-segura/vista-oido-piel/campo-visual.jsp

Moreno Wilfrido. (2016). ISTEC y SoftBank Robotics Americas lanzan un programa en América Latina, para que universidades y empresas, accedan con mayor facilidad a robots humanoides NAO – Blog de los miembros de ISTEC. Recuperado de http://www.istec.org/blog-miembros/2016/07/01/istec-y-softbank-robotics-americas-lanzan-un-programa-en-america-latina-para-que-universidades-y-empresas-accedan-con-mayor-facilidad-a-robots-humanoides-nao/

NAO Educación - Aliverobots. (s/f). Recuperado de http://aliverobots.com/nao-educacion/

Navarro, C. R. (2016). procesing « Soloelectronicos. Recuperado de https://soloelectronicos.com/category/procesing/

Navas, Gabriel, Rivera, C. (2017). *Desarrollo de un sistema para administrar medicamentos a adultos mayores a través del reconocimiento de codigo de barras y su implementación en robot NAO*. Universidad Politecnica Salesiana. Recuperado de https://dspace.ups.edu.ec/bitstream/123456789/14534/1/UPS - ST003189.pdf

Ocampo, A. S. (2014). *Desarrollo de funciones de movimiento y control de los sensores para una plataforma robótica NAO*. Recuperado de http://eie.ucr.ac.cr/uploads/file/proybach/pb0856t.pdf

Pérez Cisneros, Marco Antonio ; Cuevas Jiménez, Erik Valdemar ; Zaldívar Navarro, D. (2009). Visión artificial aplicada a vehículos autónomos. *Gaceta*, 15. Recuperado de http://www.gaceta.udg.mx/Hemeroteca/paginas/578/G578_COT 15.pdf

Priyanka Dwivedi. (2017). CarND-Advanced Lane Finder-P4. Recuperado de https://github.com/priya-dwivedi/CarND/tree/master/CarND-Advanced Lane Finder-P4

Rafael, C. L. (2006). *Robótica*. Universidad de las Américas Puebla.

Rosebrock, A. (2015a). HOG detectMultiScale parameters explained. Recuperado de http://www.pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained/

Rosebrock, A. (2015b). Pedestrian Detection OpenCV. Recuperado de http://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/

Santillan, D. (2014). Ideas Roboticas Avanzadas. Recuperado de http://ideasroboticasavanzadas.blogspot.com/2014/01/vision-artificial-ii.html

Siemens S.A. (2007). 5 NIVELES PARA ALCANZAR LA CONDUCCIÓN AUTÓNOMA. Recuperado de http://www.ciudadesdelfuturo.es/5-niveles-conduccion-autonoma.php

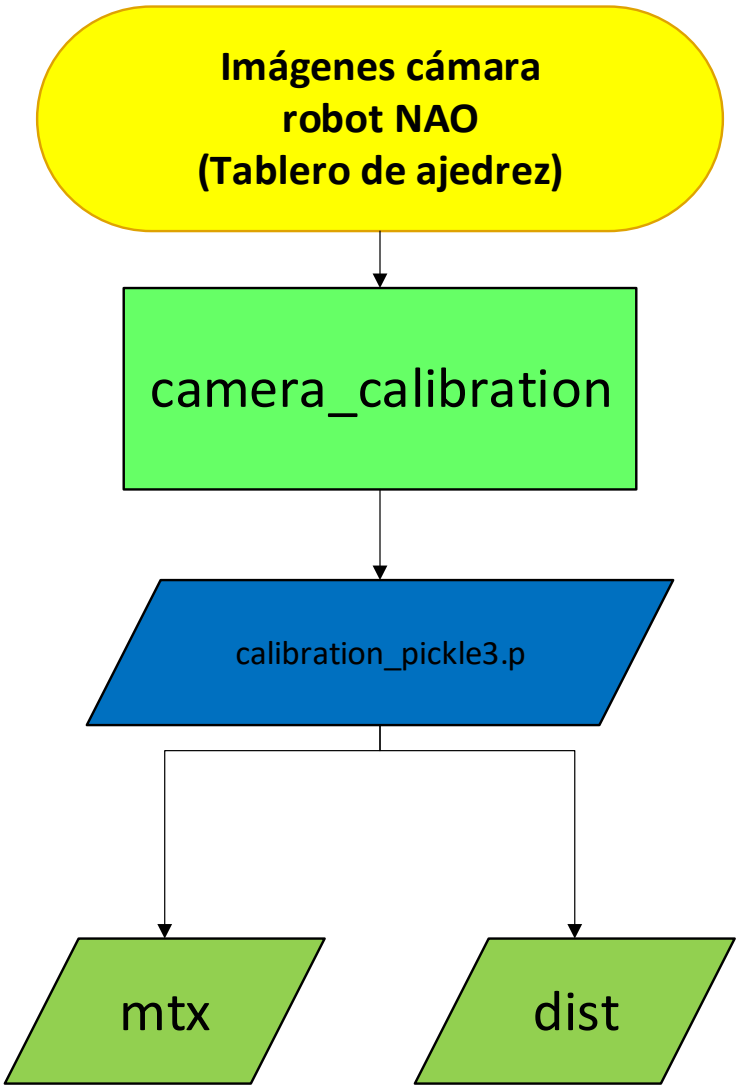
Sim, R. (2012). El campo de visión. Recuperado de http://www.racesimonline.com/articulos/El_campo_de_vision.php

SqalliFollow, M. (2017). Advanced Lane detection. Recuperado de https://medium.com/@MSqalli/advanced-lane-detection-6a769de0d581

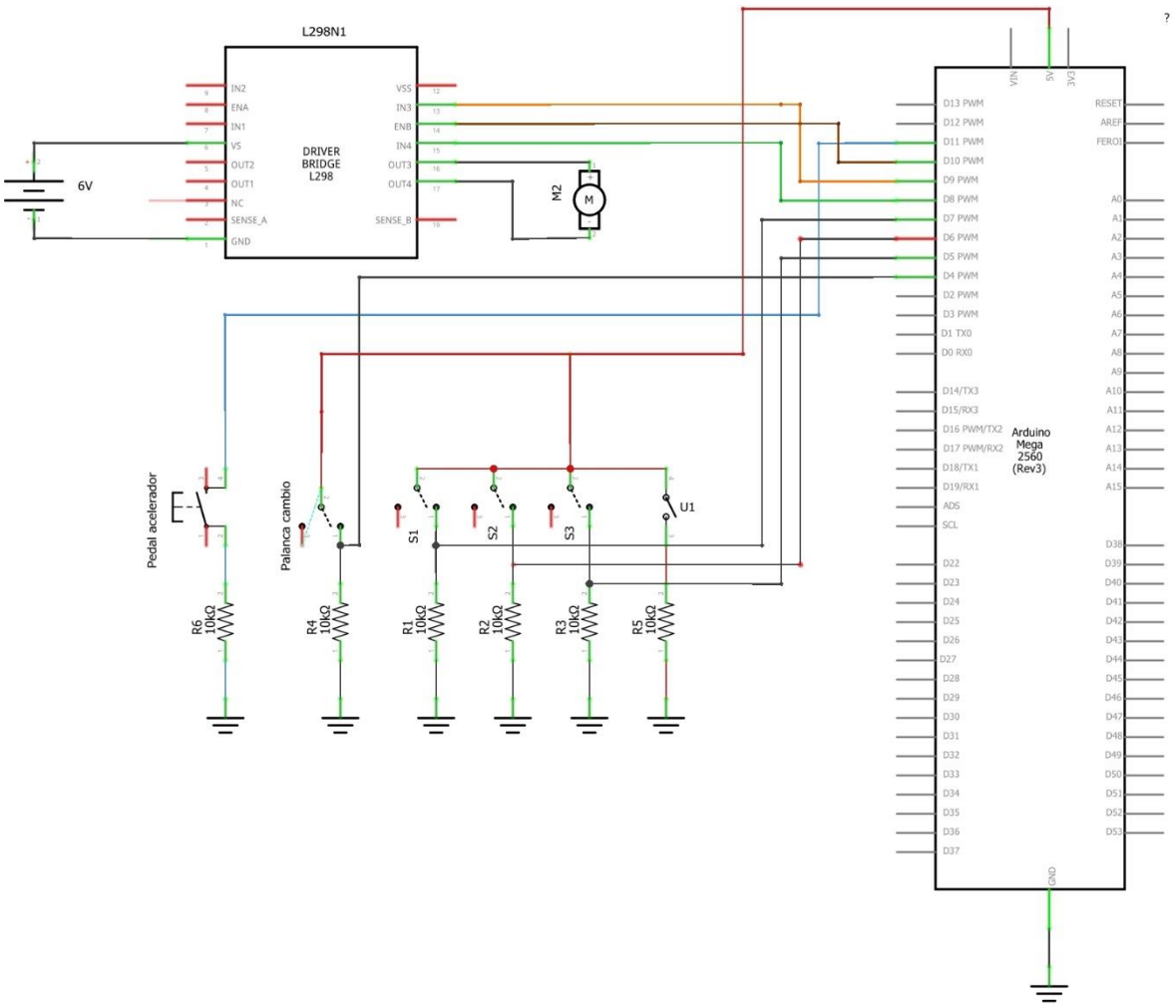
Suddrey, G., Cunningham-Nelson, S., Richards, D., Gariano, P., & Maire, F. (2014). A Simple Vehicle for the Transportation of a Humanoid Nao Robot, *2*.

Will Knight. (2015). Una cámara basta para crear un coche casi autónomo. Recuperado de https://www.technologyreview.es/s/5065/una-camara-basta-para-crear-un-coche-casi-autonomo

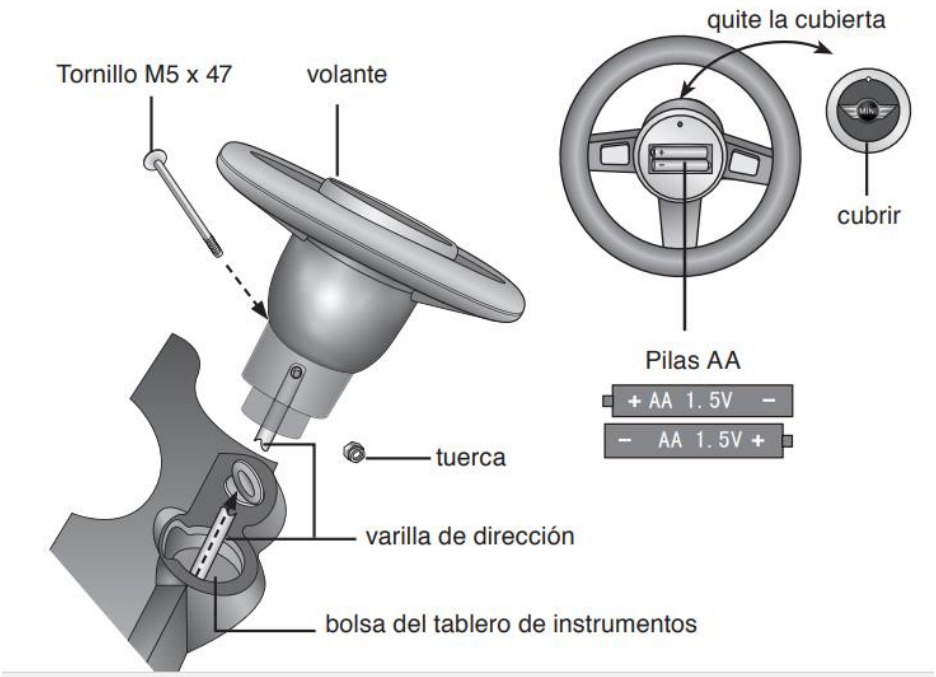
ANEXOS



Anexo A. Calibración de cámara

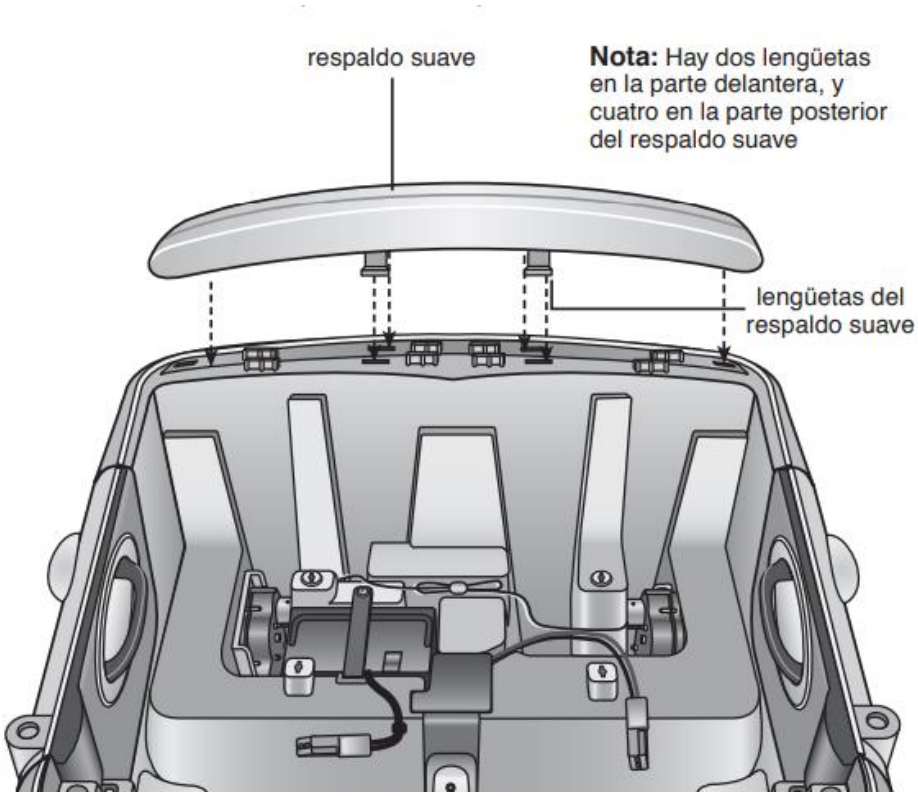


Anexo B. Diagrama de conexión Arduino, mandos y motor del coche



Anexo C. Volante original del coche.

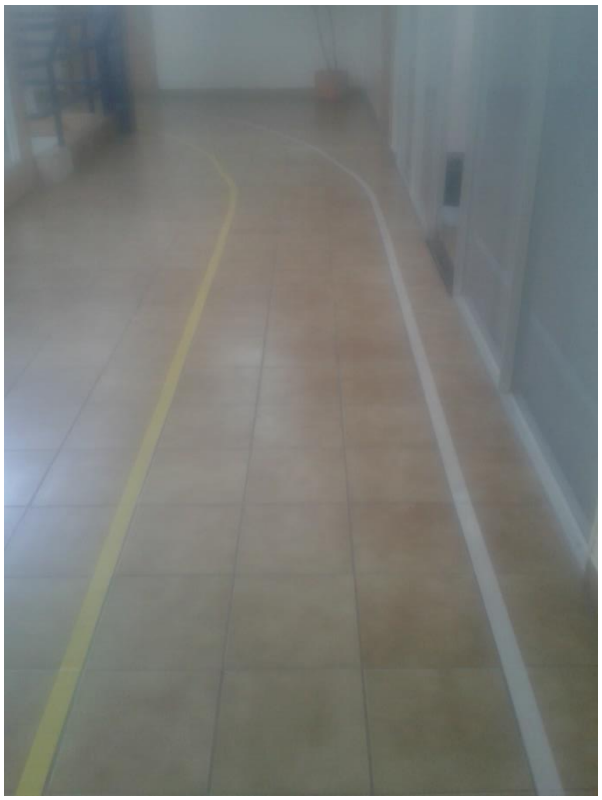
Anexo D. Modificación del volante.



Anexo E. Parte posterior donde se coloca un soporte para la laptop

Anexo F. Soporte para laptop, colocado en el coche

Anexo G. Cinturón para sujetar al robot NAO.



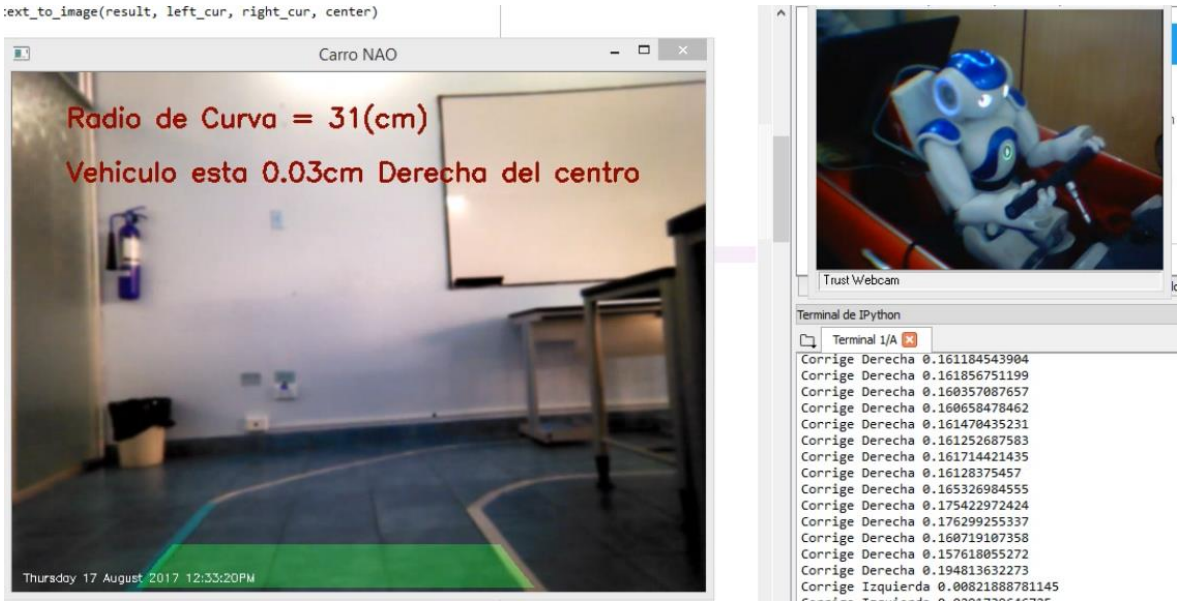
Anexo H. Pista para conducción

Anexo I. Robot NAO: sentado y manos al volante

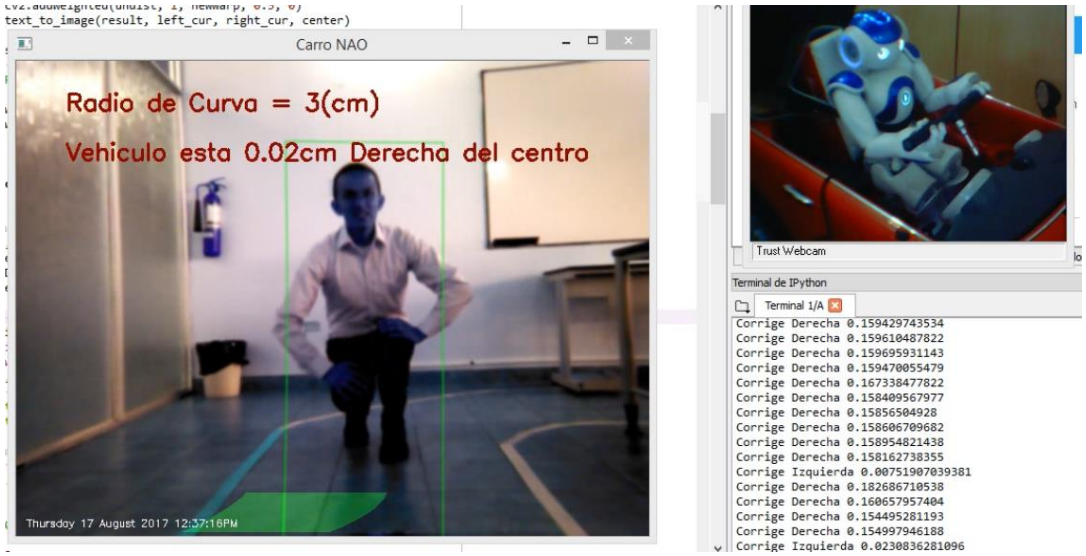
Anexo J. Robot NAO sentado en el coche y sujetado con el cinturón

Anexo K. Desplazamiento del coche y las maniobras del robot NAO

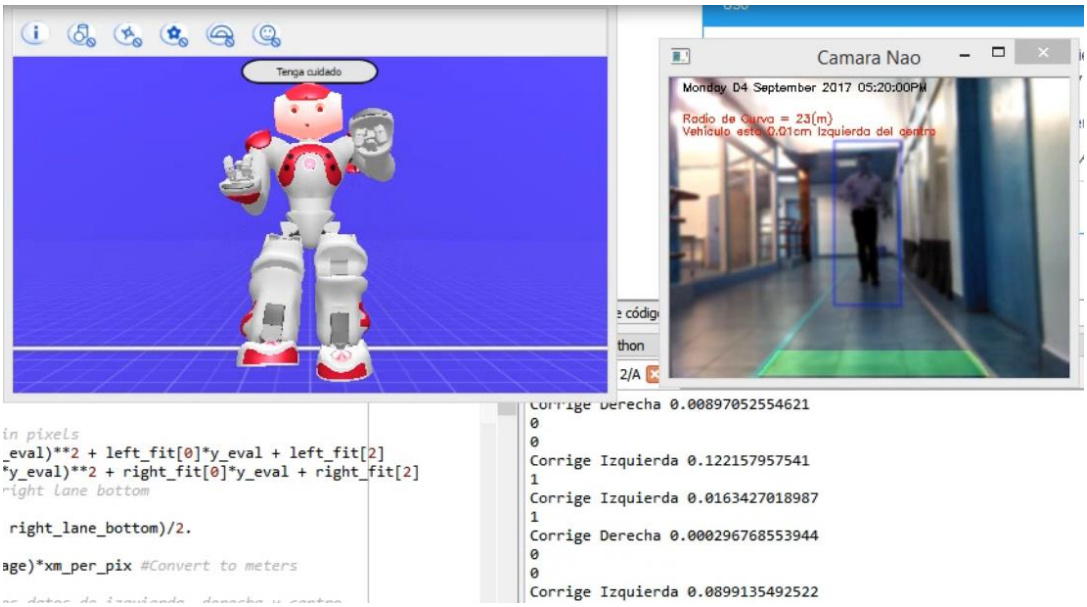
Anexo L. Giro de volante a la izquierda por el robot NAO



Anexo M. Pantalla de visualización del robot, posición sentado y datos de corrección.



Anexo N. Corrección de la trayectoria, detección de líneas y persona



Anexo O. Cámara del robot NAO y simulador enlazados, datos de corrección

PRESUPUESTO

Costos directos:

Materiales	Cantidad	$ Precio Unitario	$ Precio Cantidad
Robot Humanoide NAO (Disponible en la Universidad Politécnica Salesiana campus sur)	1	17000.00	17000.00
Licencia Choregraphe 2.1.4 (Disponible en la Universidad Politécnica Salesiana campus sur)	1	3000.00	3000.00
Laptop	1	800.00	800.00
Coche eléctrico para niños	1	300.00	300.00
Router	1	100.00	100.00
Modificaciones en coche eléctrico	-	100.00	100.00
Cinta adhesiva color amarilla	1	10.00	10.00
Cinta adhesiva color blanco	1	10.00	10.00
Total, costos Directos			**21.320,00 $**

Costos Indirectos:

Gasto	Precio
Transporte	200.00
Impresiones, copias, internet	100.00
Otros	100.00
Total, costos Indirectos	**400,00 $**