

# ANÁLISIS BÁSICO DE LOCOMOCIÓN PARA UN ROBOT NAO

Abraham Pérez Trujillo (1), Horacio Rostro González (2)

1 [Ingeniería en Sistemas Computacionales, Universidad de Guanajuato] | [Dirección de correo electrónico: [maxhacher@hotmail.com](mailto:maxhacher@hotmail.com)]

2 [Departamento de Ingeniería Electrónica, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato] | Dirección de correo electrónico: [hrostrom@ugto.mx]

## Resumen

El presente documento es reporte de lo realizado en la estancia de verano en la universidad de Guanajuato el cual consistió en el estudio del robot humanoide nao de Aldebaran Robotics así como la plataforma de desarrollo de aplicaciones Python NAOqi SDK, para posteriormente implementar rutinas básicas de locomoción.

## Abstract

This document is report what was done in the summer stay at the University of Guanajuato which consisted in studying the humanoid robot Nao Aldebaran Robotics well as the development platform Python Naoqi SDK, later to implement basic routines of locomotion .

## Palabras Clave

Python;Articulacion; Bio-Inspirado, locomoción, humanoide.

## INTRODUCCIÓN

El robot Nao tiene una semejanza con el cuerpo humano, cuenta con 25 grados de libertad de los cuales 10 se encuentran en las piernas y cada uno de ellos cuenta con un sensor de posición [1]; se realizó la instalación de software en los equipos de computo y la implementación de rutinas de locomoción además del sensado de la misma y a partir de ahí se analizaron datos y se generaron graficas para la búsqueda de patrones y su posterior uso en algoritmos bio-inspirados entre ellos SNN.

El Naoqi Framework es la plataforma usada para programar el robot Nao, cuenta con la flexibilidad de poder desarrollar módulos tanto en Python como en C++ además de trabajar en distintos sistemas operativos y con la ventaja de cubrir las necesidades de paralelismo, manejo de recursos, sincronización y eventos[2].

## MATERIALES Y MÉTODOS

### Materiales.

- Computadora
- Switch
- Robot Humanoide Nao modelo H25.

### Software Utilizado.

- C++ Naoqi SDK.
- Python Naoqi SDK
- Ubuntu 14.04 .

### Metodología.

#### Instalación del software

##### C++ SDK.

Antes de empezar la instalación del software se verifico que la maquina contara con GCC, una versión reciente de Qt Creator y que contara con Python 2.7.

Para la instalación se descargo el SDK de C++ y se extrajo después se ingreso a la carpeta extraida.

Se verifico que `$HOME/.local/bin` estuviera en el path y después se ejecuto la instrucción `“pip install qibuild —user”` y al final se ejecuto la instrucción `“qibuild config —wizard”` en la que se estableció el compilador instalado y el IDE a utilizar.

##### Python SDK.

Se descargo el archivo `“pynaoqi-python-2.7-naoqi-x.x-linux32.tar.gz”` y se extrajo en una carpeta, después se agrego una instrucción `“export PYTHONPATH=${PYTHONPATH}:/path/to/python-sdk”` en la última línea del archivo `~/.profile`.

#### Conocimiento del robot.

En la página de Aldebaran [3] se muestran varios esquemas con la posición de cada articulación en este caso nos interesan principalmente las que componen los pies del robot como se muestra en la imagen 1

#### Uso de las funciones del Robot.

El robot cuenta con varias funciones disponibles, entre ellas se encuentran la de caminar la cual puede ser accedida desde un script en Python.

Utilizamos la función `MoveTo` incluida en la SDK de python.

#### Lectura de los Sensores.

Para la lectura de sensores se utilizaron las funciones de `ALMemory` el cual nos permitia un acceso directo a la memoria ram del robot donde se almacenan los valores de cada articulación y se tomo lectura cada 50ms la cual se almaceno en una tabla y después se grafico.

#### Análisis de información de sensores.

A partir de los datos obtenidos los sensores se grafican los ángulos con respecto al tiempo para poder comprobar visualmente cuando empieza y cuando termina un movimiento para poder encontrar patrones en el movimiento y después aplicarlos a una red pulsante.

## RESULTADOS Y DISCUSIÓN

Se obtuvieron las siguientes graficas (Figura 2,3) en las cuales se puede notar que existe un patrón entre las señales, por ejemplo podemos destacar que LHIPROLL es igual a RHIPROLL y que a su vez estas señales son inversas a LANCLEROLL y RANCKEROLL y que además las señales presentan cierto patrón que facilitara implementaciones futuras.

## CONCLUSIONES

A partir de los datos se obtuvo un patrón que en trabajos futuros puede ser usada para entrenar distintos algoritmos entre ellos ANN y SNN,

## REFERENCIAS

Ejemplo: Haner, R. L., Llanos, W. & Mueller, L. (2000). Small volume flow probe for automated direct injection NMR analysis: design and performance. *Journal of Magnetic Resonance*, 143(8), 69-78.

Una página web deberá incluir la fecha de consulta número de referencia proporcionado por lapágina electrónica:

Aldebaran Robotics. (2015). NAO Robot: Intelligent and friendly companion Disponible en: <https://www.aldebaran.com/en/humanoid-robot/nao-robot> Julio 2015

Aldebaran Robotics. (2015) NAOqi Framework Disponible en: <http://doc.aldebaran.com/1-14/dev/naoqi/index.html> Julio 2015.

Aldebaran Robotics. (2015) H25 - Joints Disponible en: [http://doc.aldebaran.com/1-14/family/nao\\_h25/joints\\_h25.html](http://doc.aldebaran.com/1-14/family/nao_h25/joints_h25.html) Julio 2015.

Aldebaran Robotics. (2015) H25 - Joints Disponible en: [http://doc.aldebaran.com/1-14/family/nao\\_h25/joints\\_h25.html](http://doc.aldebaran.com/1-14/family/nao_h25/joints_h25.html) Julio 2015.

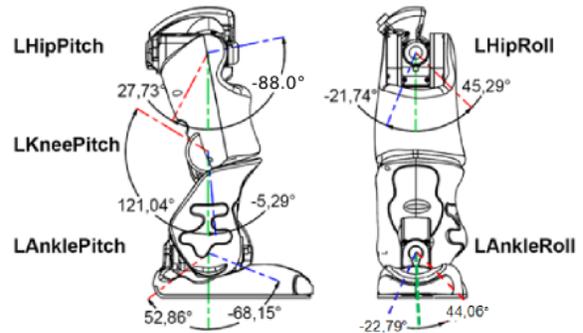


IMAGEN 1: Pie izquierdo con todas sus articulaciones y nombres usados [4].

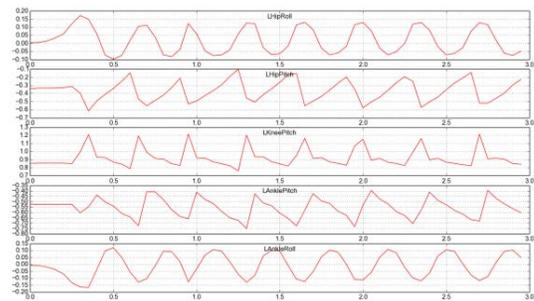


IMAGEN 2: Grafica de cada articulación, tomando el Angulo en radianes y el tiempo en milisegundos

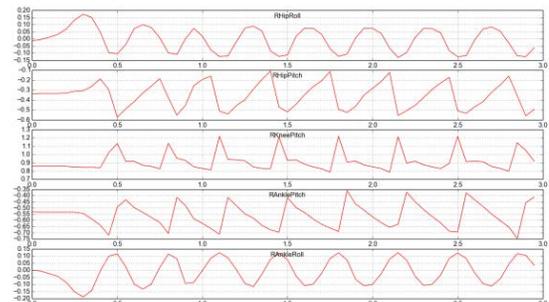


IMAGEN 3: Grafica de cada articulación del p, tomando el Angulo en radianes y el tiempo en milisegundos