



**UNIVERSIDAD DE COSTA RICA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS DE LA  
COMPUTACIÓN E INFORMÁTICA**

**CI2657- ROBÓTICA  
Prof. PhD. Kryscia Daviana Ramírez Benavides**

## **Tarea #3**

**Elaborado por:**

**Ricardo Aguilar Vargas B10141**

**[ricardoaguilar0@gmail.com](mailto:ricardoaguilar0@gmail.com)**

**Gabriel Bermúdez Mora B10954**

**[gbermudezmora@gmail.com](mailto:gbermudezmora@gmail.com)**

**Kevin Delgado Sandí B22214**

**[kefdelgado@gmail.com](mailto:kefdelgado@gmail.com)**

**Andrey Pérez Pérez B25076**

**[robinperez38@gmail.com](mailto:robinperez38@gmail.com)**

**05 de septiembre del 2016**

# Tabla de contenidos

[Tema](#)  
[Objetivos](#)  
[Enunciado](#)  
[Desarrollo](#)  
[Referencias](#)

# *Tarea #3*

## **1. Tema**

Odometría y PID.

## **2. Objetivos**

Familiarizarse y realizar diferentes actividades con el robot Q.bo.

## **3. Enunciado**

A cada grupo se le asignará un robot Q.bo. Luego cada grupo debe buscar información sobre el tema asignado y realizar lo siguiente:

### 01. Temas

- a. OpenQbo (software).
- b. Q.bo (hardware).
- c. Odometría y PID.
- d. SLAM.
- e. Detección, aprendizaje y reconocimiento de personas y objetos.
- f. Reconocimiento de voz.

### 02. Cada grupo de proyecto tiene asignado un tema:

- a. Grupo 1.
- b. Grupo 1.
- c. Grupo 2.
- d. Grupo 3.
- e. Grupo 4.
- f. Grupo 5.

Deben generar un archivo donde expliquen y resuman el tema. Además, deben programar el robot con una función asociada al tema. Luego deben realizar una presentación a la clase.

Algunos URL de interés:

- <http://thecorpora.com/blog/>
- <http://openqbo.org/wiki/doku.php>

## 4. Desarrollo

El Qbo dispone entre múltiples componentes de dos motores de 170 rpm. Ambos motores son monitoreados por un “**encoder**”. Un encoder o codificador es básicamente un dispositivo electromecánico que toma la posición angular de un eje y lo convierte en una medida digital. De esta manera el codificador permite saber datos como cuántas vueltas dan las ruedas del Qbo y de esta manera, utilizando el algoritmo **PID**, lograr que el robot vaya en línea recta.

El algoritmo PID es el encargado de corregir el error que hay entre el valor leído por el codificador y el que se quiere lograr, es por esto que es posible para el Qbo moverse en línea recta, por ejemplo.

Por otra parte los valores leídos por el codificador pueden ser procesados para obtener información, sobre la posición del robot en un espacio. A esto se le conoce como **odometría**.

Esta tecnología si bien es cierto es más cómoda económicamente hablando, suele ser poco precisa cuanto más se desplace el robot en el espacio. Esto debido a imperfecciones en la simetría de las ruedas, al nivel de fricción de la superficie o por el alineamiento de las ruedas.

Por otro lado tenemos varios controladores para el movimiento y ubicación del robot, el Base controller y el IMU controller, el primero lo que permite es activar el movimiento de la base del robot y el sistema de posicionamiento, este utiliza sensores en las ruedas para poder calcular el lugar en el espacio donde se encuentra con respecto a su posición inicial. El Qbo cuenta con un sensor IMU, este está compuesto de acelerómetros y giroscopios que logran proporcionar datos de velocidad, posición y fuerzas gravitacionales del robot. El IMU controller se encarga

del envío de mensajes del sensor IMU (Inertial Measure Unit) al sistema ROS. Básicamente son tres tópicos: envía el mensaje del sensor, realiza un chequeo de si el sensor IMU está calibrado, y calibra el sensor. Es importante que el sensor IMU esté calibrado ya que el giroscopio se desfasa con el tiempo.

En el documento se habla de Slam, pero el tema solo pide PID y Odometría, no estoy seguro de que debamos hablar de Slam.

Uno de los links sugeridos es este, pero no sé qué extraer de ahí.

## 5. Referencias

1. [Odometría - PID](#)
2. [Odometría en Qbo \(ver BaseController e IMUController\)](#)