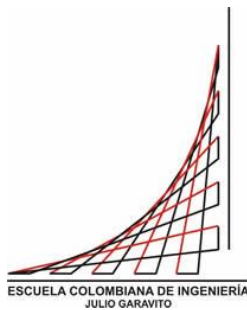


**Maestría en Ingeniería Electrónica**

**IMPLEMENTACIÓN DE UN SENSOR CON VISIÓN ARTIFICIAL  
COMO RETROALIMENTACIÓN EN UN SISTEMA BARRA Y BOLA  
QUE PERMITE EL APRENDIZAJE DE LAS LEYES DE CONTROL**

**Gustavo Alonso Virviescas Toledo**



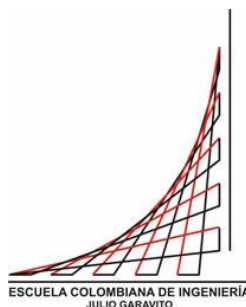
**ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO  
MAestría EN INGENIERÍA ELECTRÓNICA  
BOGOTÁ D.C.  
2016**

**IMPLEMENTACIÓN DE UN SENSOR CON VISIÓN ARTIFICIAL  
COMO RETROALIMENTACIÓN EN UN SISTEMA BARRA Y BOLA  
QUE PERMITE EL APRENDIZAJE DE LAS LEYES DE CONTROL.**

**Gustavo Alonso Virviescas Toledo**

**Trabajo de grado para optar al título de magister en ingeniería  
electrónica con énfasis en automatización industrial**

**Director:  
Ing. Enrique Estupiñán**



**ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO  
MAESTRÍA EN INGENIERÍA ELECTRÓNICA  
BOGOTÁ  
2016**

El trabajo de grado de maestría titulada “IMPLEMENTACIÓN DE UN SENSOR CON VISIÓN ARTIFICIAL COMO RETROALIMENTACIÓN EN UN SISTEMABARRA Y BOLA QUE PERMITE EL APRENDIZAJE DE LAS LEYES DE CONTROL”, presentada por Gustavo Alonso Virviescas Toledo cumple con los requisitos establecidos para optar al título de Magister en Ingeniería Electrónica con énfasis en Automatización Industrial.

---

Firma Jurado 1

---

Firma Jurado 2

---

Firma Jurado 3

---

Director de Trabajo de Grado

## **DEDICATORIA**

A mi familia, por ser el apoyo incondicional y ser el principal motor que me impulsa a construir y conseguir los sueños, planes y metas propuestas. Por permanecer siempre unida y tomar mis fracasos y logros como propios.

## **AGRADECIMIENTOS**

Expreso agradecimiento a mi director de trabajo de grado Ing. Enrique Estupiñán por motivar el desarrollo de la idea de este trabajo y su buena disposición para guiar y aconsejar durante todo el proceso. Agradezco también al programa de maestría por ofrecer las asignaturas: Control no Lineal y Visión de Maquina, razón por la cual surge este trabajo.

## RESUMEN

En el estudio de temas y asignaturas en relación a la ingeniería electrónica y sus ramas de desarrollo profesional, es necesario llevar la teoría al campo práctico. En el ámbito de la academia, es necesario desarrollar las ciencias de estudio en el laboratorio.

En la teoría del control automático, todas sus leyes se comprueban en el desarrollo de laboratorios prácticos, que en general, incluyen el moldeamiento de un sistema de primer orden u orden superior, como un motor o un horno para posteriormente calcular teóricamente controladores que permiten examinar las variables de salida, ya sea velocidad y posición o temperatura respectivamente.

La Escuela Colombiana de Ingeniería, en sus laboratorios de ingeniería electrónica cuenta con dos plantas didácticas marca Quanser, que son utilizadas para el desarrollo de prácticas en el área de control; una de ellas un modelo Barra y Bola que consiste en dirigir la posición de una bola, que se desplaza por una barra siguiendo una señal de referencia.

Estos modelos didácticos permiten aplicar los conocimientos adquiridos, para un completo desarrollo del aprendizaje de las leyes de control, y es un modelo particularmente especial para el estudio de sistemas dada su naturaleza no lineal.

Por otra parte, existe una rama de la ingeniería electrónica conocida como “La visión artificial”; visión de máquina o visión por computador, que permite entre otras muchas cosas, hacer mediciones como lo hace el sistema de visión del ser humano. Para el estudio de esta rama de la ingeniería también es necesario realizar prácticas de laboratorio una vez se comprendan todas las teorías y conceptos. Dichas prácticas pueden limitarse a la aplicación de algoritmos que permitan extraer las características deseadas de una imagen, como la silueta humana, los gestos de una mano humana o aplicaciones industriales como determinar el tamaño de una pieza en una banda transportadora, o el grado de suciedad en una botella de una línea de envase de gaseosas, cervezas o licores.

Este trabajo une las dos ramas de la ingeniería electrónica mencionadas anteriormente, el control y la visión de máquina en una herramienta didáctica, modificando un sistema existente para motivar su uso en el desarrollo de prácticas de laboratorio.

Particularmente se generan alternativas de medición de variables físicas de un sistema de control creando un sensor de visión artificial, en este caso particular, es una herramienta para determinar una posición.

A continuación, se describe el desarrollo de la idea central de cambiar la forma de determinar la posición de la bola en el sistema barra y bola mediante visión artificial y acoplar este procedimiento a un sistema de control ya existente en los laboratorios. El documento describe la motivación inicial para el desarrollo del proyecto, enfatizando las diferentes problemáticas del sensor y su incorporación a la plataforma existente.

Se presenta un marco teórico de la forma más sencilla para junto con el registro de la metodología implementada este trabajo sea replicable; finalmente se presentan los resultados y las correspondientes conclusiones.

## ÍNDICE GENERAL

1. Introducción.....	12
1.1 Motivación .....	12
1.2 Planteamiento del problema.....	13
1.3 Justificación.....	14
1.4 Pregunta de investigación .....	14
1.5 Objetivos .....	15
1.5.1 Objetivo General.....	15
1.5.2 Objetivos Específicos .....	15
1.6 Alcance y Limitaciones.....	15
2. Marco Teórico .....	17
2.1 Barra y Bola .....	17
2.1.1 Versión del Sistema Barra y Bola .....	20
2.2 Sistema de Visión Artificial .....	23
2.3 Representación .....	24
2.3.1 Imágenes.....	24
2.3.2 Secuencias Animadas .....	24
2.4 Elementos del Sistema.....	25
2.5 Configuración Hardware y Software.....	25
2.5.1 Adquisición y representación .....	25
2.5.2 Dispositivos de captura.....	26
2.5.3 Formatos más frecuentes .....	26
2.5.4 Cámara digital.....	29
2.5.5 Modelos de color .....	31
2.5.6 Modelo RGB .....	32
2.5.7 Procesamiento en MATLAB .....	33
3. Metodología .....	34
3.1 Hipótesis .....	34
3.2 Variable Principal .....	34



3.3	Metodología de Trabajo .....	35
3.4	Instrumentos .....	36
3.4.1	Cámara Utilizada .....	36
3.4.2	Implementación del Sensor .....	39
3.5	Diagrama de Bloques.....	43
3.6	Mecanismo para la Integración del Sensor con la Planta Quanser .....	45
3.7	Interfaz HMI y Determinación de Posición .....	50
3.8	Documentación de las Pruebas de Posición .....	51
4.	Resultados y Contribución .....	54
5.	Conclusiones.....	58
6.	Bibliografía .....	59

## ÍNDICE DE FIGURAS

Figura 1. Modelos barra y bola según el punto de inclinación de la barra.....	18
Figura 2. Vectores de fuerza newtoniana en el sistema barra bola.....	18
Figura 3. Prototipo barra y bola.....	21
Figura 4. Diagrama de bloques maestro-esclavo sistema barra y bola.....	22
Figura 5. Modelo barra y bola laboratorio,.....	23
Figura 6. Representación de un objeto como una superficie.....	24
Figura 7. Etapas de un sistema de visión artificial.....	25
Figura 8. Formación de una imagen Digital.....	28
Figura 9. Modelo cámara digital.....	29
Figura 10. Muestreo de una imagen.....	30
Figura 11. “Cuantización” de una imagen.....	30
Figura 12. Mezcla aditiva de colores.....	31
Figura 13. Modelo RGB.....	32
Figura 14. Imagen binaria <i>versus</i> imagen RGB.....	33
Figura 15. Especificaciones cámara utilizada.....	37
Figura 16. Resta de <i>frames</i> .....	39
Figura 17. “Umbralización” y multiplicación por máscara.....	40
Figura 18. Marcación del objeto y obtención de coordenadas (centroide).....	41
Figura 19. Ejemplo real, marcación y obtención de coordenadas (centroide).....	41
Figura 20. Zoom región encerrada.....	41
Figura 21. Prueba real con cámara DMK 23U618.....	42
Figura 22. Detección de movimiento y determinación de la posición.....	45
Figura 23. Diagrama de bloques en Simulink.....	46
Figura 24. Datos experimentales sensor original.....	46
Figura 25. q_bb01_pos_vision/sensores.....	47
Figura 26. q_bb01_pos_vision/sensores/señal.....	47
Figura 27. Datos longitud de la barra versus valores del sensor.....	48
Figura 28. q_bb01_pos_vision/sensores/señal/sensor.....	49
Figura 29. Modulo soporte cámara.....	51
Figura 30. Toma de datos para comparar resultados.....	53
Figura 31. Ejemplos medición y comparación entre valores de los sensores.....	55

## LISTA DE TABLAS Y ANEXOS

Tabla 1. Relación valores sensor con píxeles a lo largo de la barra .....	48
Tabla 2. Datos obtenidos entre los dos sensores.....	55
<b>Anexo 1</b> .....	<b>61</b>
<b>Anexo 2</b> .....	<b>62</b>
<b>Anexo 3</b> .....	<b>65</b>
<b>Anexo 4</b> .....	<b>66</b>
<b>Anexo 5</b> .....	<b>67</b>

# 1. Introducción

## 1.1 Motivación

El uso de modelos didácticos para el aprendizaje de las leyes de control, posibilita en el estudiante, la mejor comprensión y entendimiento de los modelos teóricos y matemáticos en el área de control y automatización.

La experiencia en el laboratorio es muy útil, por esta razón existe un gran número de herramientas de simulación, sin embargo, es necesario tener entornos de trabajos prácticos donde el estudiante pueda desarrollar los conceptos aprendidos teóricamente (Padula & Visioli, 2013).

La idea principal detrás de este trabajo, es la medición de una variable física y su integración, a un sistema de control. La Escuela Colombiana de Ingeniería, cuenta en sus laboratorios de electrónica con dos sistemas mecánicos marca Quanser, para el desarrollo de prácticas de laboratorio de control y “modelamiento” de sistemas dinámicos. El primer sistema consiste en un péndulo invertido de un grado de libertad. El segundo sistema es un modelo barra y bola que se describe en la sección 2.1.

El uso del sistema barra y bola es poco frecuente en estudiantes de pregrado, por otra parte esta herramienta también puede ser usada en prácticas de asignaturas en el área de visión artificial porque la posición de la bola es una variable que se puede determinar mediante técnicas de procesamiento de imágenes y video.

Este tipo de herramientas contribuyen al desarrollo del aprendizaje. “Las prácticas de laboratorio son una estrategia didáctica de lograr un proceso de enseñanza-aprendizaje” (Cardona Buitrago, 2013). La autora afirma que al realizar prácticas de laboratorio existen dificultades conceptuales, procedimentales o actitudinales de los estudiantes, sumado a la falta de equipos y espacios para el desarrollo de las mismas, implicando que se requiere más trabajo en el desarrollo de prácticas como en equipos para las mismas.

## 1.2 Planteamiento del problema

Las técnicas de análisis no lineales son convenientes para sistemas inestables, a pesar que su formulación teórica no es simple. Los sistemas industriales inestables más comunes como calderas, cargadores de batería, sistemas de generación de energía, procesos químicos, presentan dificultades de escala y de mucho riesgo para ser llevados al laboratorio. Existen alternativas más adecuadas como el sistema barra y bola que se utilizan y se desarrollan para estudiar este tipo de sistemas.(Moriano & Naranjo, 2012)

Este sistema presenta una configuración mecánica simple y puede ser usado para estudiar métodos clásicos y modernos de control, dado que, por su construcción, la inestabilidad se reduce al introducir restricciones espaciales. Se pueden diseñar controladores sencillos y controladores no lineales que permitan el seguimiento de referencia de este sistema y realizar la comparación en el diseño y su implementación, mientras se mantengan las restricciones de tiempo real.

En el sistema se busca posicionar la bola en la barra, modificando el desplazamiento angular del eje del motor, la dificultad radica en que la bola no permanece quieta, por el contrario, se mueve al mismo tiempo que se mueve la barra debido a la poca fricción.

Este es un sistema inestable en lazo abierto, debido a que la salida del sistema (posición de la bola), se incrementa sin límite (podría caer de la barra), como respuesta a una entrada constante (el ángulo de inclinación de la barra), si no existiera la señal que retroalimenta la posición exacta de la bola.

Al retroalimentar la posición, se hace que el sistema sea oscilante, se requiere entonces que el sistema de control varíe la posición angular del motor en función de la posición de la bola en la barra y vaya disminuyendo la amplitud de las oscilaciones. La planta Quanser tiene sensores resistivos y magnéticos que permiten detectar la posición de la bola, los cuales necesitan tener contacto físico con la bola.

La visión artificial permite mediante la utilización de diversas técnicas, la obtención, el procesamiento y análisis de información obtenida a través de imágenes digitales. En este caso se debe captar una secuencia de imágenes mediante una cámara para posteriormente ser tratadas y procesadas, permitiendo intervenir en el proceso, en este caso la retroalimentación de la variable posición de la bola.

Este proceso se debe realizar en su totalidad antes, de que el sistema pierda la posibilidad de realizar acción de control, a esto se le denomina “tiempo real”. En otras palabras, el sistema de visión artificial, capta una imagen de un objeto real, la convierte en un formato digital, la procesa mediante un sistema computarizado y obtiene la variable de interés procesada y entrega su valor al sistema de control en un tiempo restringido.

### **1.3 Justificación**

El sistema barra y bola del laboratorio de control de la Escuela Colombiana de Ingeniería Julio Garavito; utiliza un sistema de referencia una barra con un sensor magnético, que, debido al uso y al peso de la bola, puede generar errores en la definición de la referencia deseada.

Con el sistema de visión artificial también podría generarse la referencia de manera virtual lo que reduce el error de medición, sin embargo, esto no está incluido como parte de este proyecto.

En general un sensor de la posición de la bola que no requiere contacto físico, tiene ventajas debido a que los contactos físicos son susceptibles a perturbaciones generadas por ruido e interferencias en la medida. De igual manera se elimina el mantenimiento, que requieren los sensores eléctricos y magnéticos.

El proyecto contribuye con un sistema alternativo de retroalimentación para las prácticas de laboratorio, adicionalmente se puede usar en proyectos de grado, pregrado y posgrado; posibilitando diferentes herramientas tanto a estudiantes como profesores, para generar destrezas en conocimientos físicos, que sean aplicados en temas de control automático y visión artificial.

### **1.4 Pregunta de investigación**

¿Es posible diseñar e implementar un sensor con visión artificial para determinar la posición de una bola que se desplaza sobre una barra y usarlo como señal de retroalimentación en el lazo de control del sistema barra y bola existente en el laboratorio de electrónica de la Escuela Colombiana de Ingeniería?

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Implementar un sensor con visión artificial, que permita determinar la posición de la bola en un sistema Quanser de barra y bola para cerrar el lazo de control en simulink.

### **1.5.2 Objetivos Específicos**

- Determinar la cámara adecuada para implementar el sensor utilizando visión artificial, con base en los tiempos y resolución requeridos para realizar el procesamiento en tiempo real.
- Implementar sistema de visión artificial, que permita capturar, procesar, y segmentar la imagen de la bola para generar un solo dato de posición.
- Establecer un mecanismo para la integración de la planta barra y bola con el sensor implementado.
- Diseñar una HMI (Interfaz Hombre Maquina), en Matlab para manipular el sistema.

## **1.6 Alcance y Limitaciones**

El modelo de barra y bola de Quanser, existente en el laboratorio de ingeniería electrónica de la Escuela Colombiana de Ingeniería Julio Garavito, es una herramienta que se opera a través de Matlab y su entorno de trabajo Simulink.

Este trabajo, no incluye identificar el sistema para obtener modelos matemáticos del sistema eléctrico, ni mecánico, ni dinámico, dado que se incluye en los laboratorios ya existentes.

Para el desarrollo de este trabajo se utilizan los ejemplos propios de Quanser, para verificar el sensor de visión artificial y determinar la posición de la bola en la retroalimentación del sistema de control. De igual manera, no se pretende evaluar ni diseñar, ni implementar una ley de control.

Este trabajo aunque permite determinar la posición de un objeto no está enmarcado en un proyecto a cerca de medidas y métricas ni pretende hacer un análisis estadístico. Se pretende adicionar un método diferente de encontrar la posición en un sistema que

originalmente no lo tiene y se pueda integrar fácil y eficientemente.

Por ser un módulo, un programa diseñado en Matlab, el sensor puede ser modificado para ser utilizado para otros fines como ejemplo de un sistema que permite medir la posición de un objeto al detectar su movimiento.



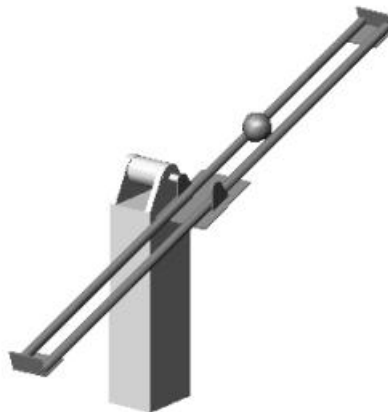
## 2. Marco Teórico

### 2.1 Barra y Bola

Equipo para estudiar la estabilización de un sistema naturalmente inestable y control de posición de la bola sobre la barra.

#### Características

- El sistema permite posicionar una bola que se desplaza por una guía, que puede oscilar en torno a una posición de referencia, en cualquier punto deseado de la guía.
- Es un equipo compacto con conexión directa a un computador, en el que la interface con otros equipos se realiza por medio de borneras, que permiten conectar las entradas/salidas. Toda la electrónica de potencia y de medida se encuentra internamente en el equipo y oculta al usuario.
- El equipo lleva una fuente de alimentación como módulo adicional adaptada al mismo.



El propósito de este mecanismo es controlar la posición de la bola cambiando el ángulo de inclinación de la barra. Esta tarea requiere del conocimiento constante de la posición de la bola, para que la barra se incline de acuerdo a la posición deseada de la bola. Convencionalmente el sistema consta de un sensor de posición eléctrico, magnético o resistivo que lleva la información hasta el controlador y un encoder que permita determinar el ángulo de inclinación de la barra sujeta al motor.

Existen diferentes modelos de barra y bola, el cambio radica principalmente en la ubicación del punto, donde se conecta la barra con el motor, puede ser en el centro o en uno de los extremos de la barra. Esto cambia la dinámica del modelo según sea el caso. La Figura 1 muestra los modelos de barra y bola según la posición donde se conecta la barra con el eje del motor.

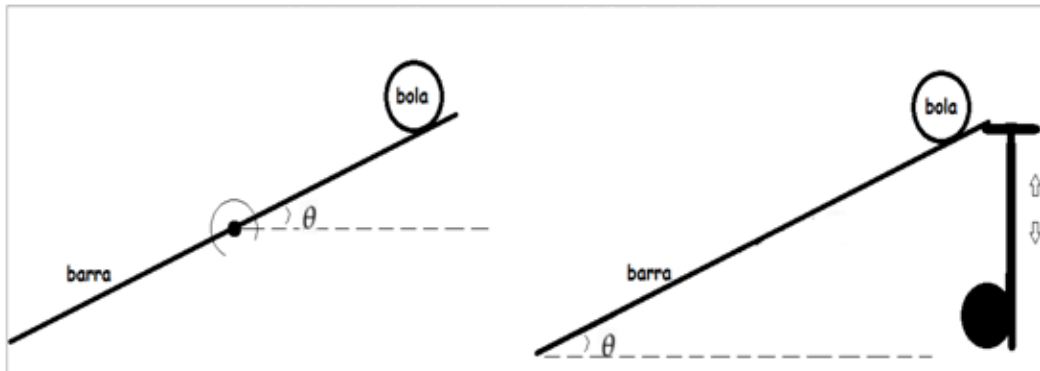


Figura 1. Modelos barra y bola según el punto de inclinación de la barra  
Fuente: El autor

### Modelado matemático del sistema

Para modelar el mecanismo bola y barra, se debe aplicar la segunda ley de Newton a la bola del sistema:

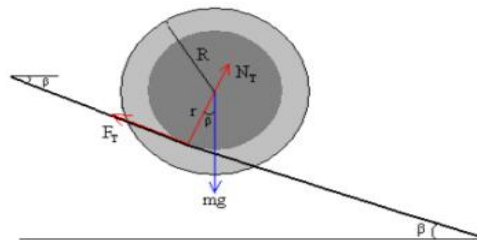


Figura 2. Vectores de fuerza newtoniana en el sistema barra bola

En la Figura 2 se observan las fuerzas que actúan en la bola: el peso debido a la fuerza gravitatoria terrestre, la normal y la fuerza de rozamiento entre la bola y la barra, tomando el eje de referencia en el centro de la barra y el eje x paralelo a esta tendremos las ecuaciones de movimiento tomadas de las guías de Quanser:

$$\text{Eje x: } m * g + \text{sen}\beta - Fr = m * a \quad (1)$$

$$\text{Eje y: } m * g \cos\beta = N \quad (2)$$

Suponiendo que la bola rueda sin deslizar por la barra, obtenemos la condición de “rodadura” que nos relaciona el ángulo de la barra con la posición de la bola:

$$\text{Condición de rodadura: } x = R_g * \beta \quad (3)$$

La fuerza de rozamiento es la causante de que la bola ruede sobre la barra, es decir, el momento de la bola es causado por la fuerza de rozamiento, aplicando la segunda ley de Newton de la rotación:

$$\text{Rotación: } Fr * R_g = J_{bola} * \frac{d^2\beta(t)}{dt^2} \quad (4)$$

La bola es una esfera perfecta por lo tanto el momento de inercia de la misma será:

$$\text{Momento de inercia de la bola: } J_b = \frac{2}{5} * m * R^2 \quad (5)$$

Y la aceleración angular de la bola y la velocidad lineal de la misma se relacionan a través de la condición de rodadura de la siguiente forma:

$$\frac{d^2\beta(t)}{dt^2} = \frac{1}{R_g} * \frac{d^2x(t)}{dt^2} \quad (6)$$

Con esta ecuación y la anterior sustituyéndolas en la ecuación de rotación obtenemos la relación de la fuerza de rozamiento que será proporcional a la aceleración:

$$Fr = \frac{2}{5} * m * a + m * \left(\frac{R}{R_g}\right)^2 * \frac{d^2x(t)}{dt^2} \quad (7)$$

Y sustituyéndola en la ecuación (1):

$$m * g * \text{sen}\beta - \frac{2}{5} * m * \left(\frac{R}{R_g}\right)^2 * \frac{d^2x(t)}{dt^2} = m * a \quad (8)$$

Y operando se obtiene:

$$\frac{d^2x(t)}{dt^2} = \frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g}\right)^2\right]} * \text{sen}\beta \quad (9)$$

Por trigonometría se obtiene el radio de giro de la bola y sustituyendo el radio de la bola y la gravedad ( $g=9.8\text{m/s}^2$ ) se obtiene la función de transferencia:

$$\frac{X(s)}{\beta(s)} = \frac{6.675}{s^2} \quad (10)$$

### 2.1.1 Versión del Sistema Barra y Bola

En el laboratorio de la Escuela Colombiana de Ingeniería Julio Garavito, el sistema del fabricante Quanser, consiste en una esfera de acero colocada sobre un riel, acoplado en uno de sus extremos a un motor mediante un sistema de engranajes. La posición de la bola se mide mediante un sensor resistivo, que tiene integrado la barra del sistema. El ángulo de inclinación de la barra es medido por medio de un potenciómetro propio del sistema unido al motor. Para que el sistema de engranajes no sufra daño alguno, el ángulo de desplazamiento debe ser entre  $-45^\circ$  y  $+45^\circ$ .

Para definir la posición de referencia de la esfera, se usa una barra gemela que se localiza cerca y por medio de un sensor magnético se detecta la posición de la bola de referencia deseada como se muestra en la Figura 34.

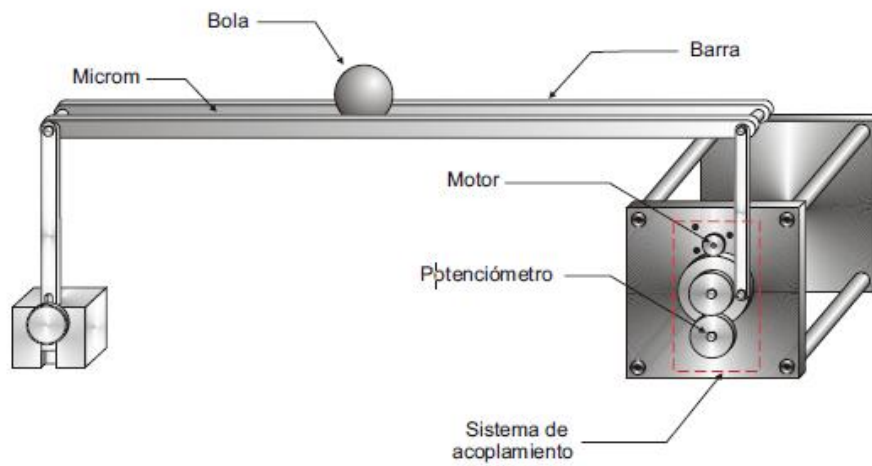


Figura 3. Prototipo barra y bola  
(Cuevas López, Mora Reyes, & Olivera Mera, 2008)

El esquema de control utilizado, es el esquema maestro-esclavo mostrado en la Figura 4; consta de dos sistemas modulares:

- El esclavo o parte a controlar.
- El maestro que genera la señal de referencia. La barra del esclavo tiene un sensor resistivo y la barra del maestro tiene un sensor magnético.

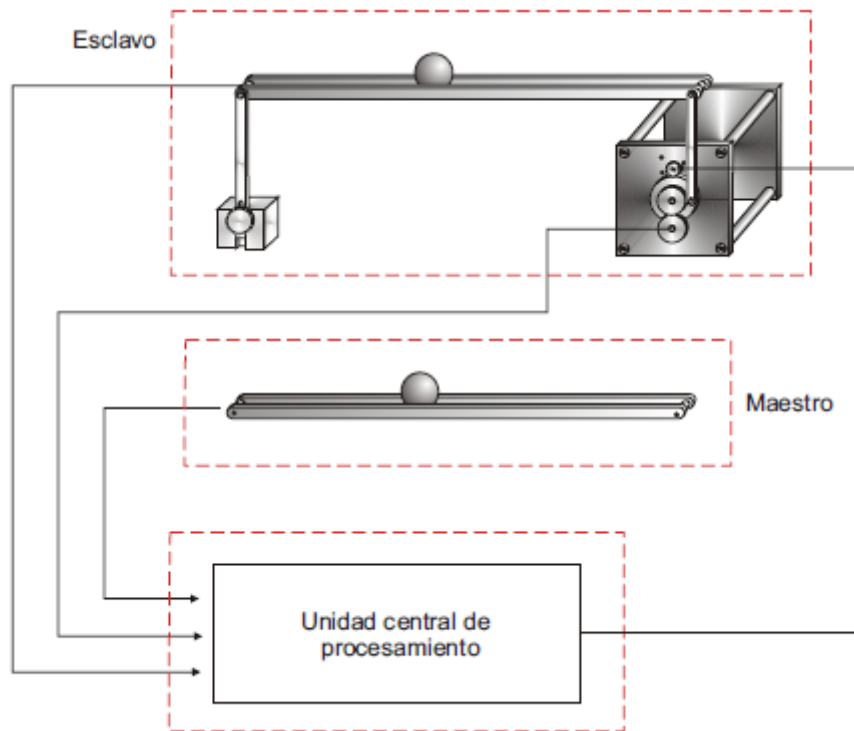


Figura 4. Diagrama de bloques maestro-esclavo sistema barra y bola  
(Cuevas López, Mora Reyes, & Olivera Mera, 2008)

El sistema cuenta con una tarjeta de adquisición de datos, donde están conectados los sensores de esclavo y maestro y las señales de potenciómetro y encoder propios del sistema.

La Figura 5 es una foto tomada por el autor de la planta Quanser (maestro/esclavo), existente en el laboratorio de automatización y control de la Escuela Colombiana de Ingeniería Julio Garavito.

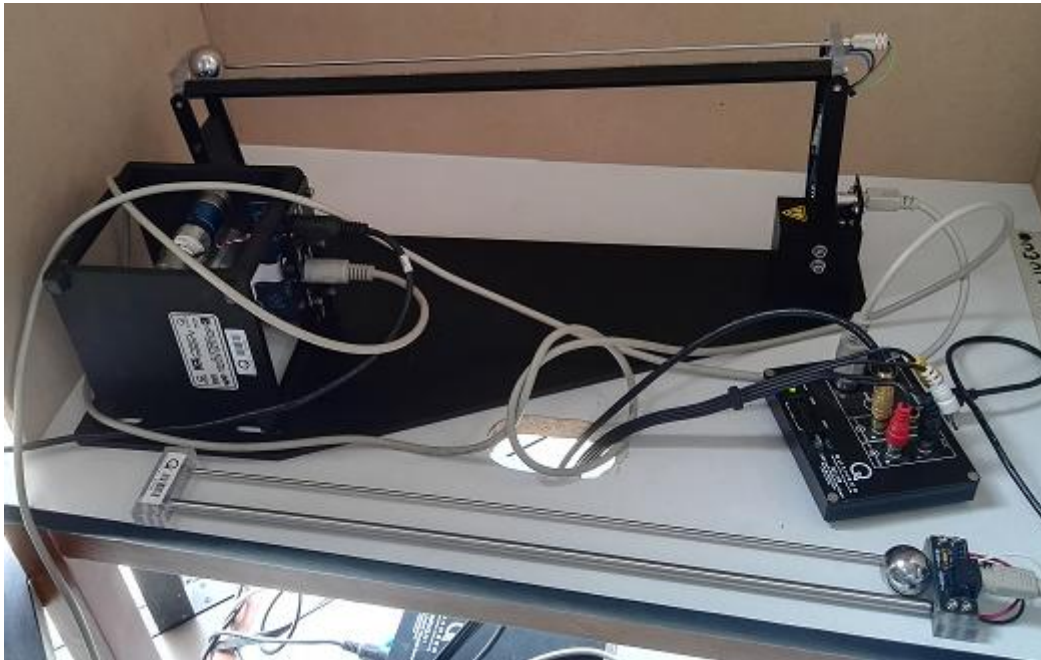


Figura 5. Modelo barra y bola laboratorio,  
Escuela Colombiana de Ingeniería Julio Garavito, Fuente: el autor

## 2.2 Sistema de Visión Artificial

La visión artificial es un campo del procesamiento de señales que mediante la utilización de técnicas adecuadas, permite obtener, procesar y analizar cualquier tipo de información especial obtenida a través de imágenes digitales. (Velez Serrano, Moreno Díaz, Sánchez Calle, & Sánchez Marín, 2003)

El proceso a realizar en visión artificial es: captar la imagen, almacenar la información, procesar la información e interpretar los resultados.

La visión artificial da facultades a una máquina para tomar decisiones en el mundo real, basándose en la información que provee el procesamiento de las imágenes provenientes de un dispositivo de captura. El objetivo principal es que la máquina interprete la información visual como lo haría un ser humano.

La visión artificial permite hacer tareas como el seguimiento de objetos o personas como por ejemplo en cálculo de personas que transitan por un determinado lugar o el conteo de objetos que transporta una banda transportadora. Clasificación de figuras, colores, tamaños y texturas en el caso de un sistema de control de calidad.

La dificultad existente en sistemas con visión de máquina radica en la incertidumbre de la información que necesita la máquina para tomar una decisión apropiada en contraposición al sistema de visión humana.

## 2.3 Representación

El objetivo de la visión artificial es extraer información de algo físico, a partir de imágenes tomadas y procesadas en un computador. La representación de la realidad extraída de las imágenes proporciona información sobre brillo, colores, formas. La representación puede ser de imágenes estáticas, escenas tridimensionales o imágenes en movimiento.

### 2.3.1 Imágenes

Una imagen bidimensional, es una función en coordenadas  $(x, y)$ , en la cual cada punto tiene asociado un valor que representa su brillo o intensidad. Una imagen sin color donde cada punto representa un valor de brillo se puede representar como una superficie. El valor del brillo indica la representación de la altura, como se observa en la Figura 6.

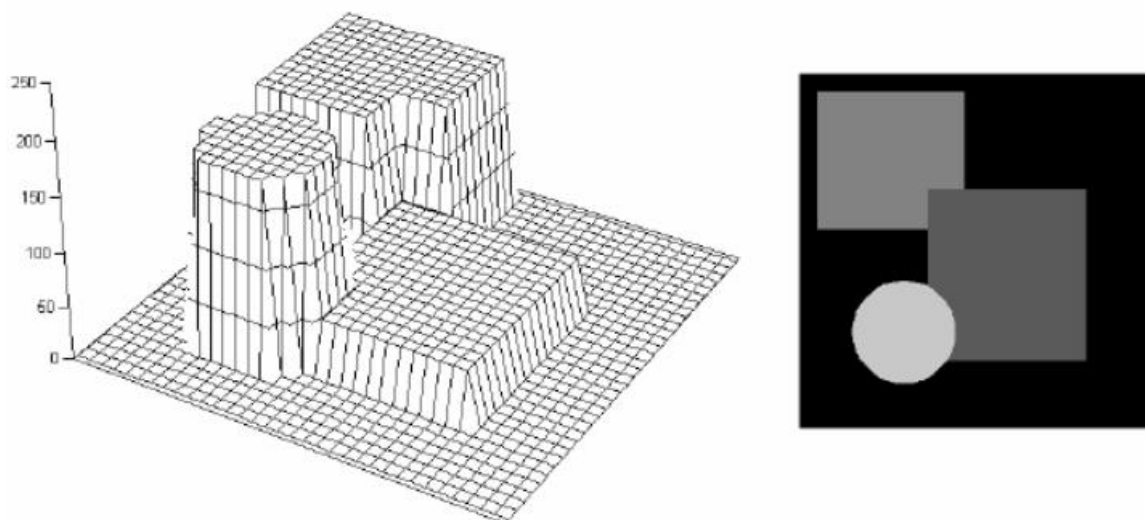


Figura 6. Representación de un objeto como una superficie  
Tomada de: (Velez Serrano, Moreno Díaz, Sánchez Calle, & Sánchez Marín, 2003)

### 2.3.2 Secuencias Animadas

Una secuencia de imágenes estáticas, da como resultado un video (secuencias animadas). El ejemplo más claro es una cámara cinematográfica que toma imágenes estáticas sucesivamente a una frecuencia determinada. Si la frecuencia es mayor a 25 imágenes por segundo, el ojo humano no es capaz de distinguir el cambio e interpreta la sucesión como



movimiento. El video entonces se compone de una secuencia de imágenes, cada imagen se denomina cuadro (*frame*), siendo la resolución del video el número de *frames* por segundo (*fps*). Cada *frame* contiene una resolución espacial la cual hace referencia a la cantidad de píxeles que representa la imagen.

## 2.4 Elementos del Sistema

Un sistema de visión artificial busca en algunos aspectos emular el sistema de visión humano, el cual captura luz a través de los ojos y envía información al cerebro por medio del nervio óptico. El proceso se puede dividir en cuatro fases.

- Captura o adquisición de imágenes digitales mediante algún sensor
- Procesamiento, tratamiento digital de las imágenes (filtros y transformaciones geométricas)
- Segmentación, que consiste en aislar la información que es relevante estudiar o comprender.
- Reconocimiento o clasificación, consiste en obtener información de los elementos segmentados analizando las características obtenidas.

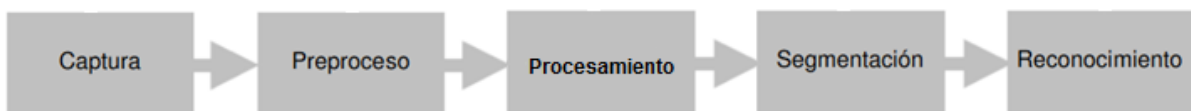


Figura 7. Etapas de un sistema de visión artificial  
Tomada de: (Velez Serrano, Moreno Díaz, Sánchez Calle, & Sánchez Marín, 2003)

## 2.5 Configuración Hardware y Software

### 2.5.1 Adquisición y representación

Como se describió anteriormente, el primer paso es capturar una imagen la cual debe ser digitalizada para poder ser procesada en un computador. Es decir que en el proceso de obtener una imagen digital existen 2 etapas las cuales son la captura y la digitalización.

Una vez digitalizada una imagen bidimensional, está constituida por un conjunto de elementos llamados píxeles. Cada píxel ofrece información sobre una región elemental de la imagen. En una imagen en niveles de gris esta información es el brillo. En imágenes a

color, la información corresponde a la intensidad de cada una de las componentes de una base de color (por ejemplo, RGB) (Velez Serrano, Moreno Díaz, Sánchez Calle, & Sánchez Marín, 2003)

### **2.5.2 Dispositivos de captura**

Los principales dispositivos de captura de una imagen son:

- Cámara fotográfica analógica, tecnología antigua, es común en la actualidad en el uso artístico en el proceso fotográfico tradicional.
- Cámara de video analógica, tecnología antigua, en la actualidad aún se usa en menor escala en procesos de televisión y cine.
- Cámara fotográfica digital, tecnología moderna, se usa en el procesamiento de imágenes, también se usa en el ámbito artístico debido a su procesamiento después de la captura.
- Cámara de video digital, tecnología moderna, se usa en el procesamiento de video y en las nuevas tecnologías de televisión y cine.
- Escáner, dispositivo netamente para digitalizar un documento o imagen física.

Para el tratamiento de imágenes en procesos de visión artificial y visión de máquina se requiere trabajar en formato digital, por esta razón si la captura se realiza con dispositivo análogos, es necesario hacer una conversión análogo-digital para su posterior procesamiento.

### **2.5.3 Formatos más frecuentes**

Formatos de imágenes digitales:

- BMP. Imagen de mapa de bits, se usa para imágenes de baja calidad. Es un formato estándar para almacenar imágenes independientes del dispositivo e independientes de la aplicación. Los archivos BMP no se comprimen, por tal motivo no son apropiados para ser transferidos en internet.
- GIF. Formato gráfico usado en la web, tanto para imágenes como para animaciones. Se suelen comprimir sin que se pierda información.
- JPG. Es el formato más común y está presente en la mayoría de cámaras digitales. Al

comprimir el archivo se degrada la calidad de la imagen.

- PNG. Muy parecido al formato GIF, se comprimen sin perder información.

Formatos de video:

- AVI. Creado por Microsoft, para usar en los computadores. Los archivos AVI se pueden ver en cualquier reproductor dependiendo de los códec instalados.
- MPEG. Es de los primeros formatos de video digital, es compatible con casi todos los reproductores. En la actualidad es el estándar de codificación de video utilizado en la TDT como también en los discos DVD.
- MOV. Es el formato usado por QuickTime, sistema de Apple. Muchas cámaras digitales usan este formato.
- WMV. Formato Windows Media Video, desarrollado por Microsoft, es un formato propio de esta marca.
- FLV. Flash Video, creado para transmitir video en formato flash por internet.

A continuación, se ilustra cómo es la formación de una imagen digital. Muestra un sensor (cámara digital), que gracias a la luz natural captura un plano de un elemento físico y obtiene como resultado una imagen digitalizada.

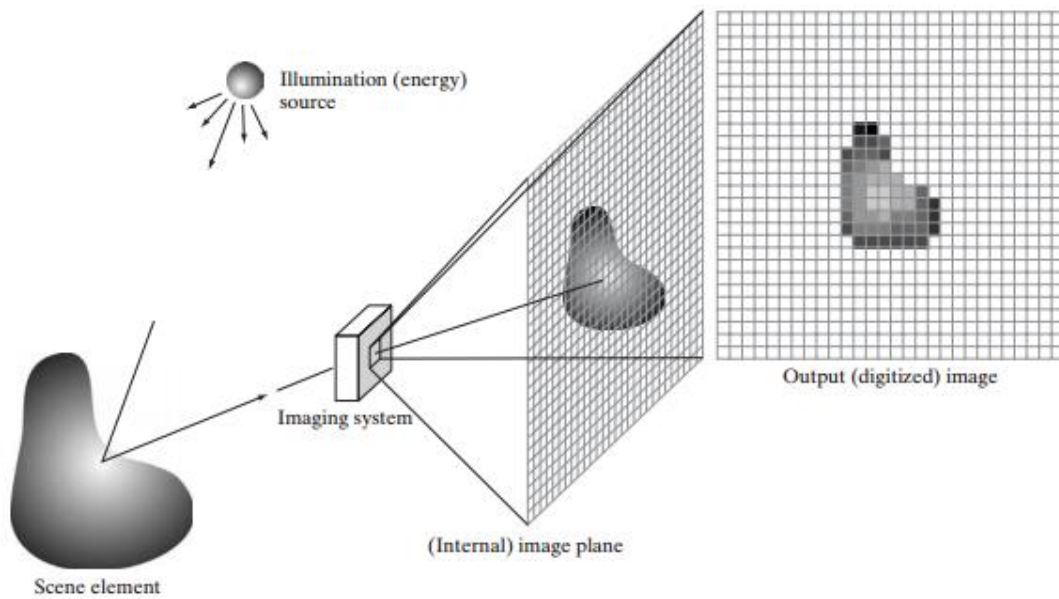


Figura 8. Formación de una imagen Digital.  
Tomada de: (Gonzalez & Woods, 2007)

La imagen digitalizada para efectos de procesamiento, es una matriz de ( $m \times n$  pixeles), donde cada pixel representa, en el caso de una imagen monocromática, un nivel de gris asociado al brillo. Si la imagen es una imagen RGB (Rojo, verde, Azul por sus siglas en inglés); existen 3 matrices que representan los tres colores respectivamente, y cada pixel representa un nivel de intensidad de color en cada matriz.

## 2.5.4 Cámara digital

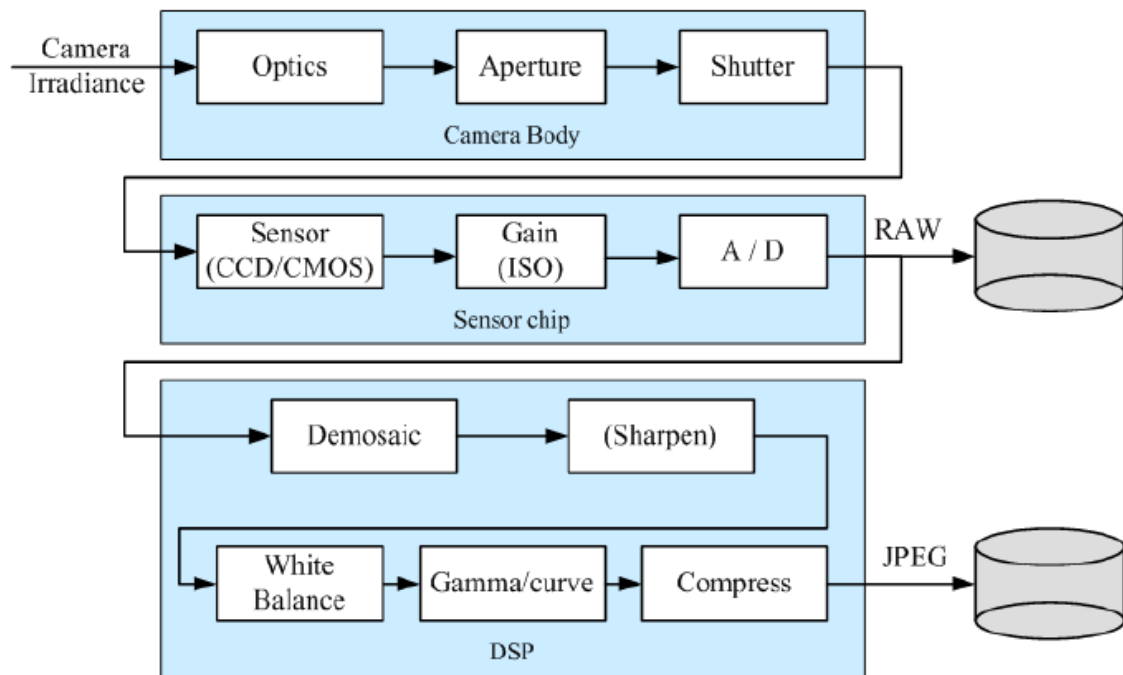


Figura 9. Modelo cámara digital  
Tomada de (Szelski, 2010)

En la Figura 109 se diagrama el chip sensor, que es donde se forma la imagen. Los más usados son los CMOS (Complementary Metal Oxide On Silicon), o los sensores CCD (Charged Coupled Device). Estos convierten la luz que llega en una señal eléctrica (por medio de celdas fotoeléctricas), y contienen filtros de color (para ajustar la imagen al contenido de colores deseado). Diferencias del chip sensor:

- Sensor CCD: Almacena la carga eléctrica de las celdas y la agrupa para transmitirla como una sola señal que se amplifica y digitaliza.
- Sensor CMOS: La carga eléctrica de cada foto-celda se almacena, se amplifica y se digitaliza por separado.

El sensor CMOS es más rápido que el sensor CCD, porque no requiere de agrupar la señal por filas y columnas. También es más económico debido a que contiene menos componentes. Los sensores CMOS en la actualidad, son los más usados, aunque tienen ciertas desventajas debido a que es más sensible a captar ruido. El módulo A/D conversor

análogo-digital, que se muestra en la figura 9 cumple con dos procesos: muestreo y “cuantización”. El muestreo consiste en determinar la resolución espacial de la imagen (detalle más pequeño discernible)

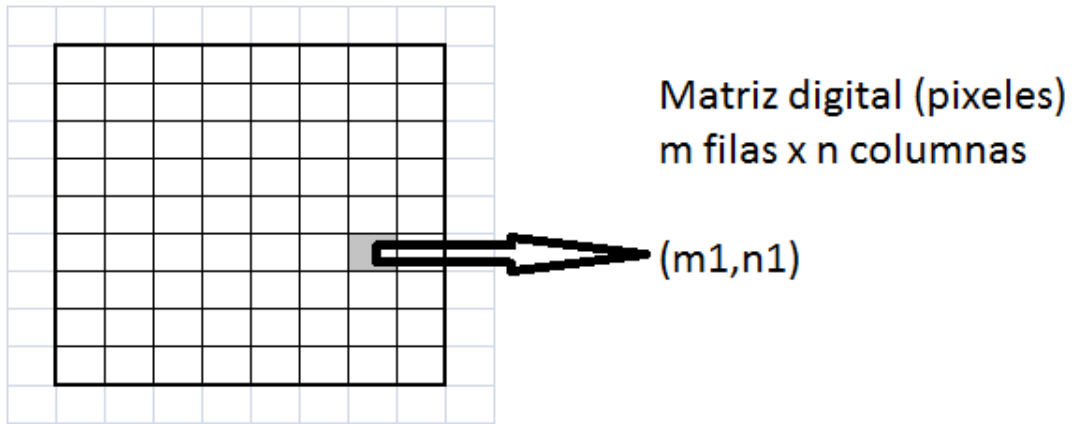


Figura 10. Muestreo de una imagen  
Fuente: el autor.

La “cuantización” consiste en determinar la resolución de niveles de gris (profundidad de píxel), de la imagen. Corresponde al cambio más pequeño discernible en los niveles de gris.

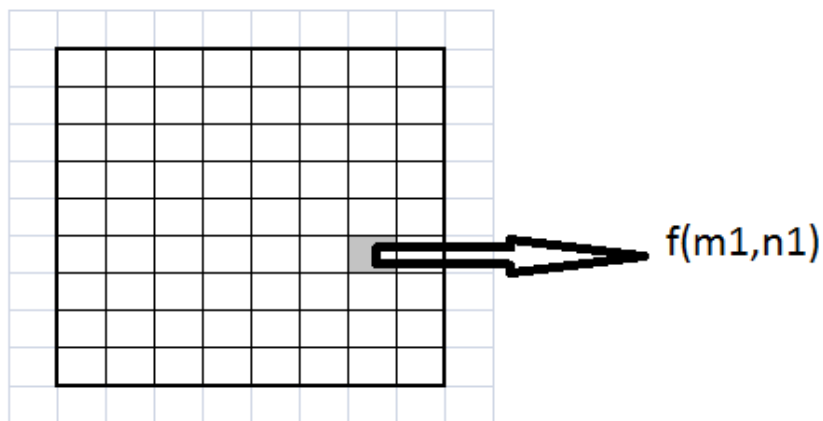


Figura 11. “Cuantización” de una imagen  
Fuente: el autor.

Como se muestra en la figura 11,  $f(m_1, n_1)$ ; es una función de la cantidad de luz que incide en el objeto y la cantidad de luz que éste refleja. Es un nivel de gris definido en un número de bits ( $k$ -bits). Por ejemplo 8 bits = 256 posibles niveles de gris donde, 0 es negro y 255 es blanco.

La resolución total de la imagen sería igual a  $m \times n \times k$ . Si M y N fueran 100, la imagen sería de  $100 \times 100 \times 8$  (100x100 píxeles con 256 niveles de gris). El módulo *Demosaic*, identificado en la figura 9 es el módulo de la representación del color. Es una matriz de filtros rojos, verdes y azules, que se sitúa sobre el sensor digital para hacer llegar a cada fotodiodo información de luminosidad a cada sección de los distintos colores primarios. La luz es una mezcla aditiva de tres colores.



Figura 12. Mezcla aditiva de colores  
Tomada de (Szelski, 2010)

### 2.5.5 Modelos de color

Los modelos sirven para la representación de cada color en una imagen digital usando un sistema de coordenadas y estos son:

- Modelo RGB (Red Green Blue)
- Modelo HSV (Hue, Saturation, Value)
- Modelo YCBCR (Y: Luminancia, Cb: Crominancia Azul, Cr: Crominancia roja)

Para efectos del trabajo vamos a ver en detalle el modelo RGB, ya que en visión artificial y visión de máquina, es un modelo muy usado, y los resultados son buenos. En el procesamiento se descomponen 3 matrices, cada una de ellas representando cada uno de los colores RGB.

## 2.5.6 Modelo RGB

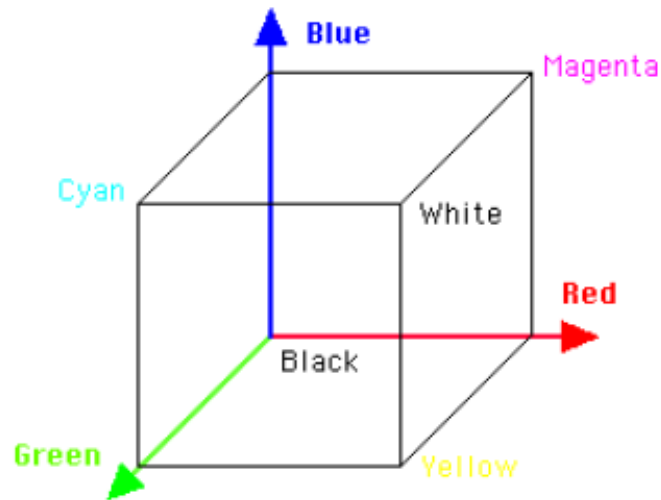


Figura 13. Modelo RGB  
Tomada de:(Coelho Simplicio, 2011)

El modelo RGB, se representa, como un cubo en el plano cartesiano, donde el azul, el verde y el rojo son sus componentes principales. Tienen una escala desde el centro hacia afuera, como lo indican sus flechas desde el nivel más oscuro de los colores al más claro. De acuerdo a la mezcla aditiva de colores se obtiene por sus diferentes combinaciones los colores Amarillo, Magenta y Cyan. Como también el blanco y el negro.(Coelho Simplicio, 2011)

La representación normalizada del color blanco sería  $(1, 1, 1)$ ; y la representación del color negro sería  $(0, 0, 0)$ . Si fuese una imagen con resolución de 8 bits el color blanco estaría representado por el valor  $(255, 255, 255)$  y el color negro por  $(0, 0, 0)$ ; donde estos valores hacen referencia al nivel que representa cada uno de los colores (R, G, B) en cada pixel. Un pixel  $(m1, n1)$  tiene un valor (R, G, B), lo que significa que una imagen se puede descomponer en 3 matrices diferentes, matriz de R, matriz de G y matriz de B.

La siguiente figura muestra cómo se representa una imagen monocromática binaria en una sola matriz, donde los píxeles pueden tomar valores de 0 o 1. Cada matriz representa un color y cada píxel toma un valor en la escala de colores de sí mismo.



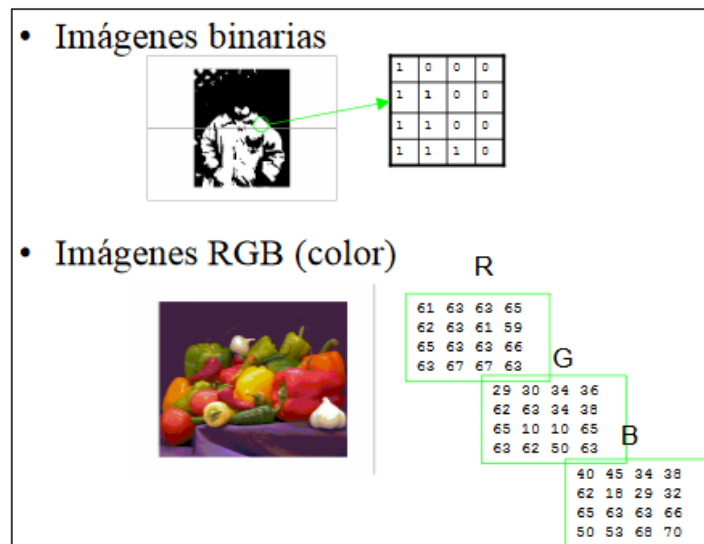


Figura 14. Imagen binaria *versus* imagen RGB

Tomada de: <http://informatica.uv.es/iiguia/VC/tutorial.pdf>, agosto 25 de 2015

### 2.5.7 Procesamiento en MATLAB

Teniendo la representación de una imagen como una matriz, se puede realizar cualquier operación necesaria para manipular la información y procesarla de acuerdo, a lo que se necesite. En una imagen se puede manipular los píxeles, los valores de los píxeles, se pueden hacer operaciones en las 3 matrices. Para el caso de un video, el procesamiento y la manipulación es la misma ya que un video se compone de imágenes sucesivas, es decir que para el procesamiento, es necesario extraer las matrices sucesivas y adicionalmente se debe tener en cuenta la frecuencia en que aparecen estas.

En el software se puede manejar matrices, representar datos, funciones, ya que permite implementar algoritmos propios e incluso con otros lenguajes de programación. Tiene muchas cajas de herramientas (Toolbox), para el procesamiento de imágenes posee (Toolbox Image Processing).

### 3. Metodología

La cimentación del trabajo, está enmarcado en una investigación aplicada, dado que se busca diseñar el sensor con visión artificial, dando solución a un reto, y una problemática de índole tecnológico.

Se pretende utilizar todos los conceptos existentes en el proceso y tratamiento de señales digitales, como también las diferentes funciones y algoritmos que Matlab permite utilizar de tal forma que se optimicen todos los recursos.

En cuanto a la implementación del sensor diseñado, se desarrolla un trabajo de tipo experimental, ya que se está manipulando un modelo existente de barra y bola. Se manipula un sistema creado, mediante la utilización del sensor integrado a un sistema diseñado en Matlab, software en el cual está desarrollado el modelo barra y bola de Quanser.

#### 3.1 Hipótesis

Para diseñar el sensor con visión artificial, se parte de la idea y el principio que cualquier imagen digitalizada se puede tratar y procesar como una matriz en un computador, utilizando todos los conceptos y teorías existentes en cuanto al tratamiento y procesamiento de imágenes.

Al tener una cámara de video digital adecuada se puede adquirir información del mundo real para procesar su información. El procesamiento del video se hace de la misma forma a como se tratan las imágenes digitales trabajando sobre cada *Frame* de video. MATLAB es un software adecuado para el tratamiento de imágenes y video digital permitiendo generar algoritmos con funciones especializadas y propias del programa.

#### 3.2 Variable Principal

La variable principal es la posición de la bola respecto a la barra. El sensor con visión artificial debe determinar la posición mediante el procesamiento del video adquirido y entregar un valor de forma tal, que el modelo de barra y bola de Quanser pueda interpretarlo. La cámara toma información de la variable en este caso la bola para ser procesada y entregar un valor, este valor es la posición.

### 3.3 Metodología de Trabajo

Existen trabajos realizados con visión artificial que permiten el reconocimiento facial, reconocimiento de placas de vehículos, reconocimientos de objetos en cuanto a formas y texturas, detección de movimiento, detección de colores, conteo de personas que transitan por algún lugar específico, entre otros. Trabajos realizados de detección de movimiento tales como “Detección y Análisis de Movimiento Usando Visión Artificial” (Sanabria S. & Archila D., 2011) y “Sistema de Visión Artificial para el Registro de Densidad Peatonal en Tiempo Real” (Abril Cañas, Valencia L., Restrepo, & Holguín, 2007) permitieron un conocimiento y experiencia previa para desarrollar el proyecto.

El estudio de asignaturas como, control no lineal y visión de máquina, durante el período de maestría; proporcionaron un conocimiento amplio del sistema modelo “Barra y Bola”, y las técnicas de procesamiento de imágenes y video digital.

En la asignatura electiva, “Control no Lineal”, se consideró como proyecto final diseñar un controlador para el sistema “Barra y Bola” de Quanser; utilizando las herramientas e información necesaria para llevar a cabo dicho propósito.

Como proyecto final de la electiva visión de máquina, se ejecutó un algoritmo para determinar el tráfico en una avenida, contando el número de vehículos que transitaban por un punto en un determinado instante de tiempo. El análisis no realizaba el conteo de vehículos en tiempo real, se capturaba un video durante un periodo de tiempo, para luego ser procesado en Matlab mediante algoritmos diseñados para tal fin. Como resultado se obtuvo la detección de movimiento segmentado a un solo carril (centro del video); pero con muchas dificultades para determinar el número de carros detectados.

Los resultados contribuyeron en la metodología del trabajo, debido a la experiencia y conocimiento previo de sistemas de visión artificial. En primer lugar, se determina la cámara adecuada a utilizar, posteriormente se diseña un sensor con visión artificial mediante algoritmos registrados en un código de Matlab. Se integra la información generada por el sensor con la planta Quanser, para que mediante experimentos obtener la posición de la bola; todo esto se hace en la interfaz de Simulink de Matlab.

A continuación, se describen los parámetros que se determinaron para la selección de la

cámara y en el algoritmo implementado. Se hacen pruebas para determinar la correcta medición y determinación de la posición de la bola, como la verificación del código desarrollado para que el sensor arroje la señal y el valor correcto. Pruebas de comunicación y tiempo de respuesta. Se documentan los respectivos experimentos y procesos.

### **3.4 Instrumentos**

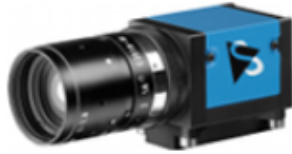
Como todo sistema de visión artificial, los instrumentos utilizados dentro de las fases del proyecto fueron la cámara, un soporte especial construido y diseñado para adquirir la información de la bola sobre la barra, mediante los datos procesados del computador; para automatización y el registro que contiene las licencias de Matlab y de Quanser; herramientas necesarias para poder desarrollar el código de procesamiento de video, para obtención del sensor y asociarlo con Quanser, que funciona sobre la plataforma de Matlab.

#### **3.4.1 Cámara Utilizada**



Para la adquisición de la cámara, las principales características que se tuvieron en cuenta fueron la resolución del sensor, la frecuencia de cuadros (Frame Rate fps); que se pudieran conectar vía USB, y que tuviera compatibilidad con Matlab o LabVIEW, que en lo posible fuera una cámara monocromática, con el fin de procesar una sola matriz; esto debido a que el objetivo principal era detectar movimiento y no color, y disminuir la carga para el procesamiento y memoria de almacenamiento.

Se buscaron cámaras especializadas para el procesamiento de imágenes, con sus características propias e integración de software especializados para estos procesos, como Matlab y su facilidad de manipular en un computador, se adquirió la cámara DMK 23U618 de la marca (The ImagineSource), es una cámara monocromática con resolución de 640X480 pixeles y una frecuencia de hasta 120 fps.

## DMK 23U618



Lens not included

- **USB 3.0 Monochrome Industrial Camera**
- 1/4 " Sony CCD
- 640x480 pixel
- Trigger input and I/O
- Windows  and Linux  software included

Video formats @ Frame rate  
640x480 Y800 @ 120, 90, 60, 30, 15, 7.5, 3.75 fps  
640x480 Y16 @ 120, 90, 60, 30, 15, 7.5, 3.75 fps

Figura 15. Especificaciones cámara utilizada  
Tomada de: <http://www.theimagingsource.com>

Es una cámara que se conecta por medio de un puerto USB, que garantiza un Frame Rate de hasta 120 fps, conectado a un puerto USB3.0 y hasta 60 fps si se conecta a un puerto USB convencional. Tiene dos resoluciones: 640x480 píxeles *y800*, que significa que tiene 8 bits de resolución y 640x480 píxeles; *y16* que significa que tiene 16 bits de resolución. Para efectos prácticos, se utiliza resolución de 8 bits.

Para comprobar las características de frecuencia de toma de cuadros de la cámara, se diseñó un código en Matlab, ver anexo 1. Los resultados obtenidos probando una cámara web y la cámara DMK 23U618 de The ImagingSource fueron:

DMK 23U618:

```
Command Window
frameRates = |
    '120.0005'
    '110.0001'
    '100.0000'
    '90.0001'
    '80.0000'
    '70.0001'
    '60.0002'
    '50.0000'
    '40.0000'
    '35.0000'
    '30.0000'
    '25.0000'
    '20.0000'
    '15.0000'
    '10.0000'
    '7.5000'
    '5.0000'
    '3.7500'

actualRate =
    120.0005
```

Como se observa, usando el código se encontró que el máximo Frame Rate de la cámara DMK 23U618 fue: 120fps.

Web Cam:

```
Command Window
frameRates =
    '10.0000'
    '5.0000'

actualRate =
    10
```

El máximo Frame Rate de la cámara webcam USB fue: 10fps.

La cámara DMK23U618, experimentalmente tiene mayor Frame Rate, y además es monocromática (una sola matriz para temas de procesamiento digital). Por lo tanto se elige esta cámara para el desarrollo del proyecto.

Se comparan resultados con diferentes cámaras web convencionales existentes en el laboratorio de la universidad, arrojando resultados similares a los encontrados con la cámara web registrada anteriormente.

### 3.4.2 Implementación del Sensor

Para determinar el movimiento de la bola que se desplaza sobre la barra se requiere hacer un procesamiento de *frames*, en este caso se usa la resta de *frames* consecutivos de manera que el fondo queda negro y lo que queda diferente se está moviendo en el primer plano.

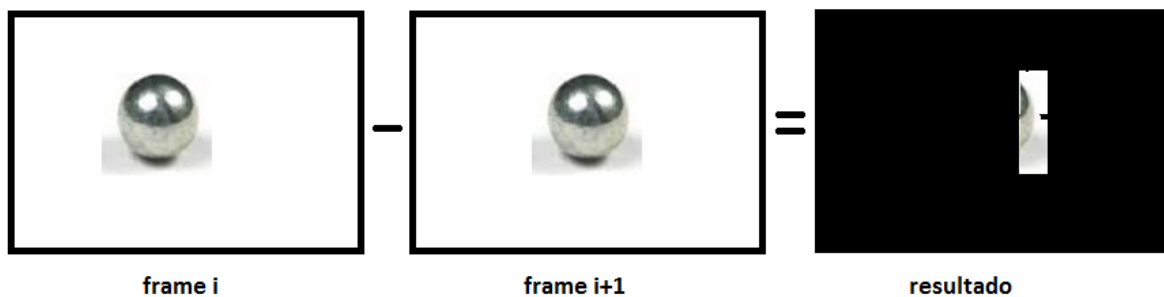


Figura 16. Resta de *frames*

La anterior figura se representa como al restar un frame actual (frame i+1) de un frame anterior (frame i) da como resultado una región diferente de 0 (negro), al presentarse un desplazamiento de la bola. Esta región posee una información en cuanto a coordenadas equivalentes a píxeles, y es donde un sistema de visión de máquina puede tomar una decisión. Para tomar una decisión de la región obtenida respecto al resultado de la resta entre frames se debe hacer un procesamiento especial. En primer lugar, se hace una “umbralización”; que consiste en volver binaria la imagen obtenida a partir de un umbral.

La “umbralización” es uno de los más importantes métodos de segmentación. El objetivo es convertir una imagen en escala de grises a una nueva con sólo dos niveles, de manera que los objetos queden separados del fondo. Es decir, tener una imagen negra y blanca, valores 0 y 1 respectivamente. Sin embargo, en la imagen resultante se pudo haber captado movimientos en regiones que no son de interés, se puede obtener regiones en blanco en un espacio donde no se mueve la bola. Para eliminar regiones no importantes, para el estudio se utiliza una máscara en este caso un valor de 1 (blanco), para la región de interés y valores de 0 (negro) para lo que se quiere desechar, ejemplo Figura 17.

Una máscara es una imagen con la misma resolución, que tienen los frames del video con una información normalmente binaria que representan los colores: negro (0) y blanco (1), donde la región blanca es la región que se desea estudiar. Generalmente, esta máscara se multiplica a otra imagen, para obtener como resultado la región de interés, puesto que lo que se multiplique por cero es negro y todo lo que se multiplique por uno es el valor original. Esta máscara se multiplica a la imagen “umbralizada” anteriormente, recordemos que en el procesamiento de imágenes estamos haciendo operaciones matriciales. Para efectos del proyecto la máscara selecciona la región de la barra por donde se desplaza la bola. La figura 18 ilustra este procedimiento. La razón por la cual se utiliza esta máscara, es porque la región de interés para detectar movimiento es única y exclusivamente la barra por donde se desplaza la bola.

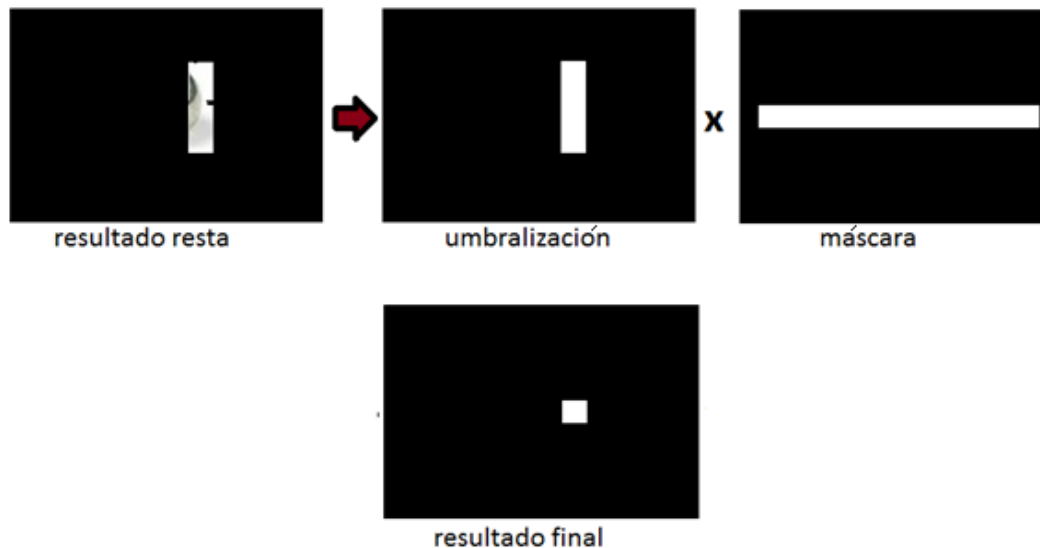
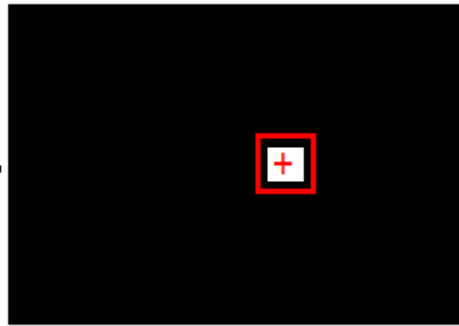


Figura 17. “Umbralización” y multiplicación por máscara

El resultado de multiplicar la región que determina un movimiento, la máscara nos da la información exacta que la bola se movió a una posición  $x$  sobre la barra y es necesario saber el valor de  $x$ , en cuanto a las coordenadas de la matriz (píxeles). Para esto se enmarca la región y se calcula el centro. Podemos decir que esta es la posición (resultado final).





resultado final

Figura 18. Marcación del objeto y obtención de coordenadas (centroide)

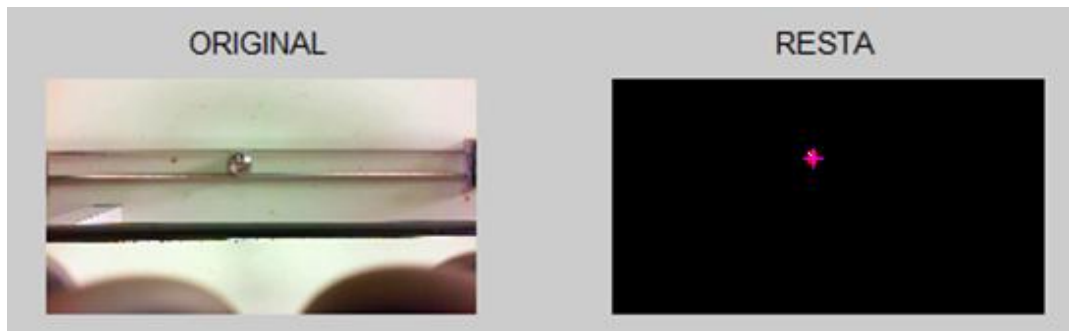


Figura 19. Ejemplo real, marcación y obtención de coordenadas (centroide)

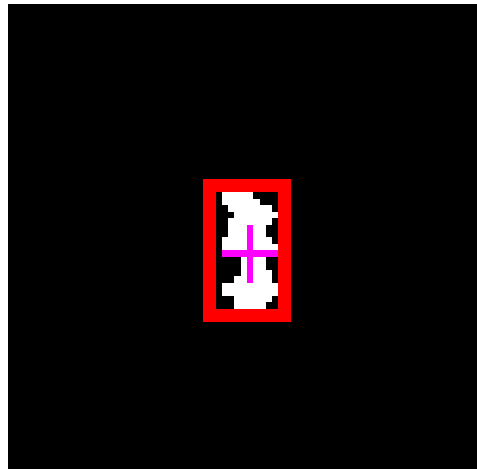


Figura 20. Zoom región encerrada.

Dependiendo de las condiciones de luz con las que se trabaje en la adquisición de imágenes y la forma de ubicar la cámara, la calidad de la región obtenida al captar un movimiento mediante la resta de frames cambia. En la sección 3.7 se toca el tema de cómo

solucionar estos inconvenientes para mejorar la medición.

En la Figura 21 se muestra como el sistema capta el movimiento, pero no se tiene certeza de que la información equivalga al centro de la región, para corregir esto, es necesario determinar la dirección del movimiento y el número de píxeles al cual equivale el tamaño de la bola para sumar o restar píxeles al resultado final. Para efectos prácticos del proyecto, la adquisición de datos se toma con la cámara localizada perpendicularmente a la barra donde se desplaza la bola a una distancia de 150 cm. Mayor especificación en sección 3.7.

El sistema mediante la resta de frames y el procesamiento descrito anteriormente obtiene una región cuyo centro determina la posición. El procesamiento es simple para calcular la variable posición, y enviarla con una mayor rapidez utilizando los mínimos recursos de máquina.

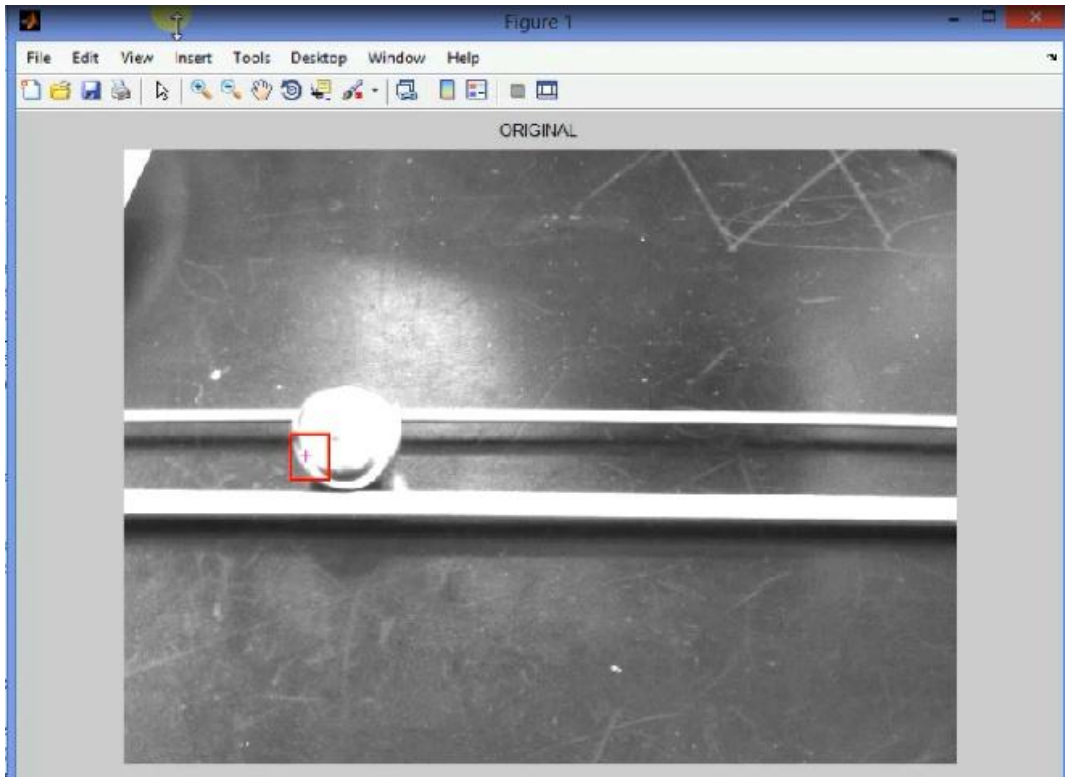
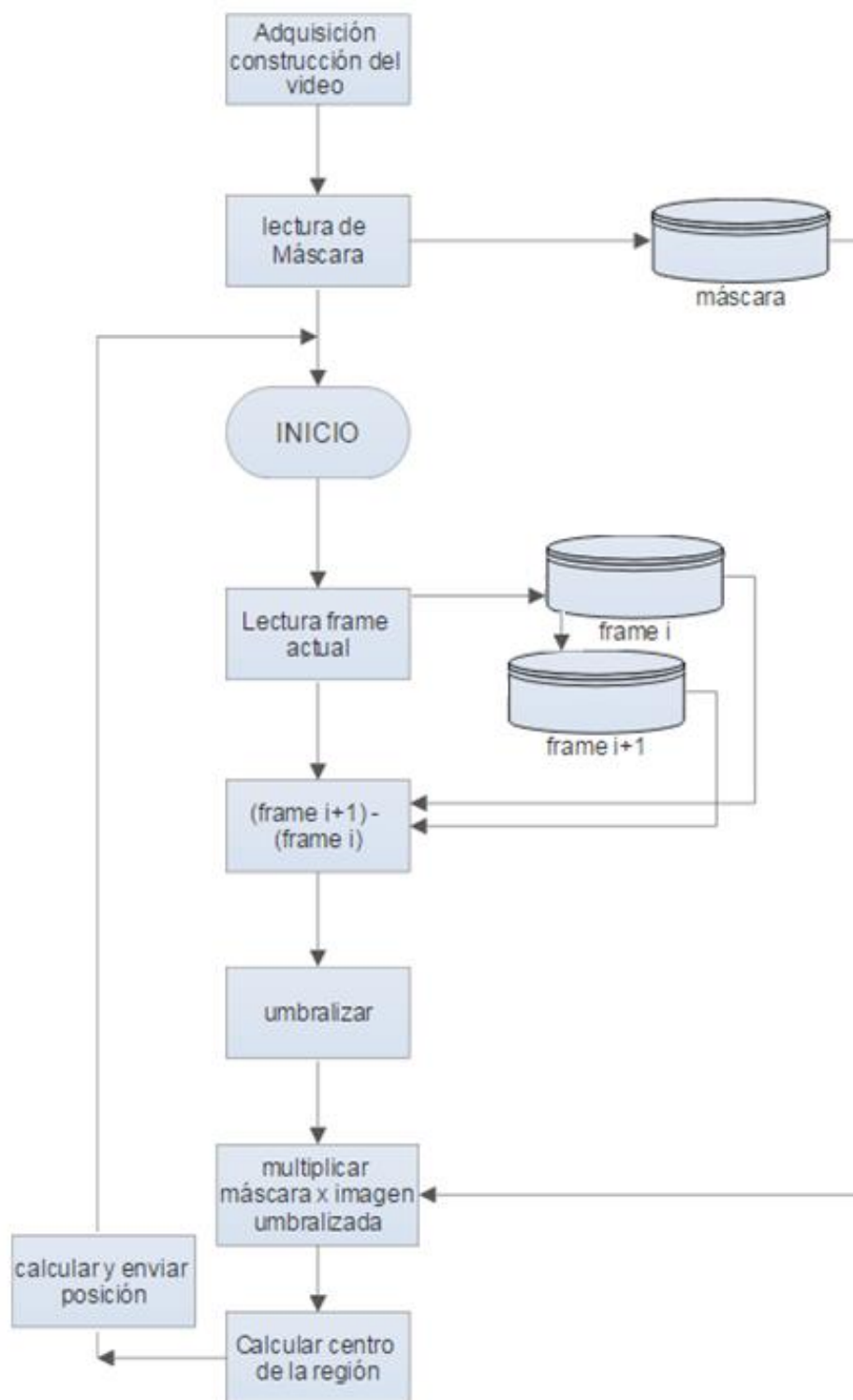


Figura 21. Prueba real con cámara DMK 23U618.

### 3.5 Diagrama de Bloques

El código completo diseñado e implementado para determinar la posición de la bola que se desplaza sobre la barra se puede ver en el Anexo 2.

Con el anterior código de Matlab se obtiene la posición de la bola cuando se mueve. Al detectar un movimiento después de restar los frames constantemente; arroja un valor que determina la posición y este valor es enviado al modelo Quanser. Con este código en Matlab, diseña e implementa el sensor con visión artificial del sistema. El valor obtenido por el algoritmo, no hace referencia precisamente al centro de la bola ya que la región que determina, por ser la resta de dos frames, indica es el borde de la bola dependiendo el sentido en que se mueva. Al saber un valor de posición actual y un valor de posición anterior y realizar una resta entre valores, se puede determinar la dirección de movimiento. La Figura 21, muestra este procedimiento con la región encontrada.



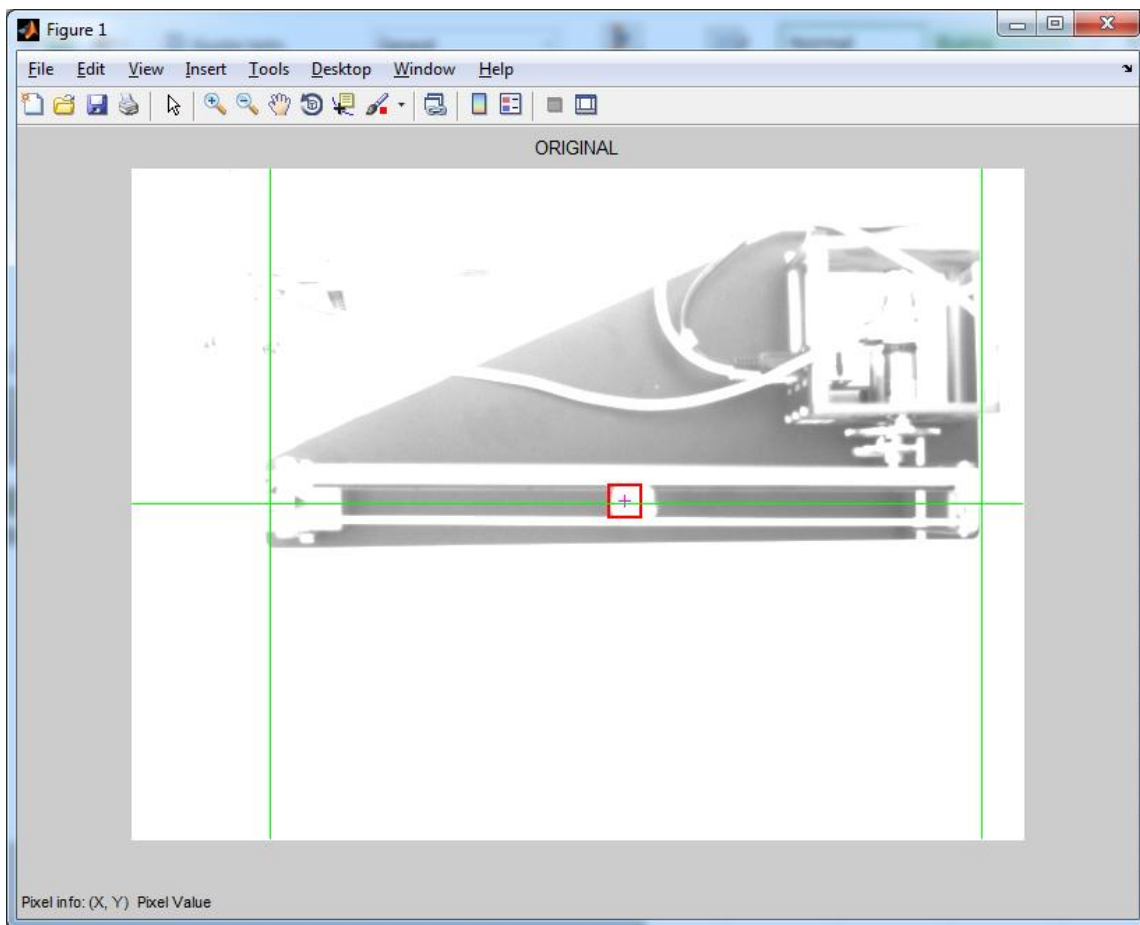


Figura 22. Detección de movimiento y determinación de la posición  
Con el sensor diseñado e implementado.

### 3.6 Mecanismo para la Integración del Sensor con la Planta Quanser

Para probar e integrar el sensor con la planta de barra y bola de Quanser, se utilizaron los ejemplos didácticos propios del fabricante.

Este experimento ejemplo de Matlab es ideal ya que el archivo *setup\_srv02\_exp04\_bb01.m* trabaja en conjunto con el modelo de Simulink, al calcular automáticamente los parámetros de control para hacer la práctica de laboratorio. El modelo de Quanser en Simulink *q\_bb01\_pos.mdl* recibe automáticamente parámetros de un controlador (PD), al ejecutar el script *setup\_srv02\_exp04\_bb01.m*, permitiendo manipular la planta barra y bola de la forma Maestro/Esclavo. Es decir, la bola del esclavo sigue la referencia de la posición de la bola del maestro. El modelo Simulink presenta en la Figura 23.



Esta señal se encuentra en el módulo de sensores del modelo *q\_bb01\_pos.mdl* y está variable se llama  $x$  (m) como se ve en el bloque encerrado de la Figura 21.

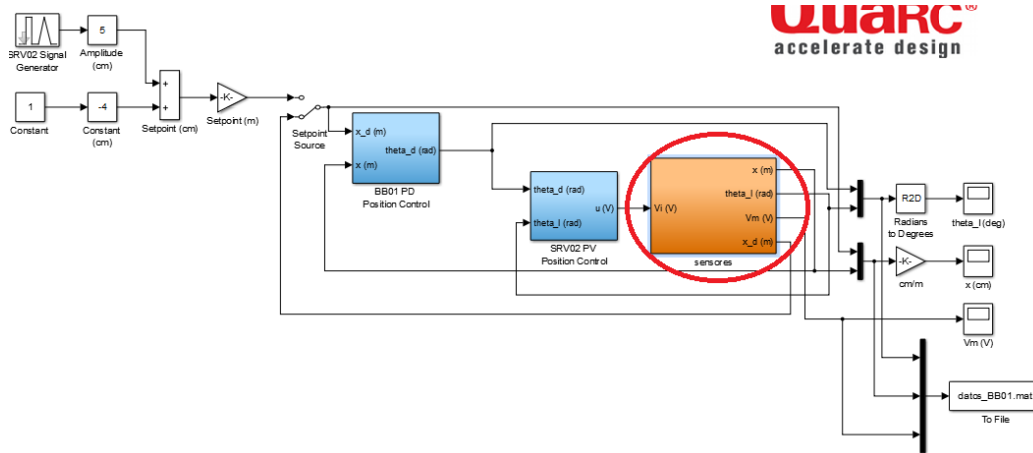


Figura 25. *q\_bb01\_pos\_vision/sensores*

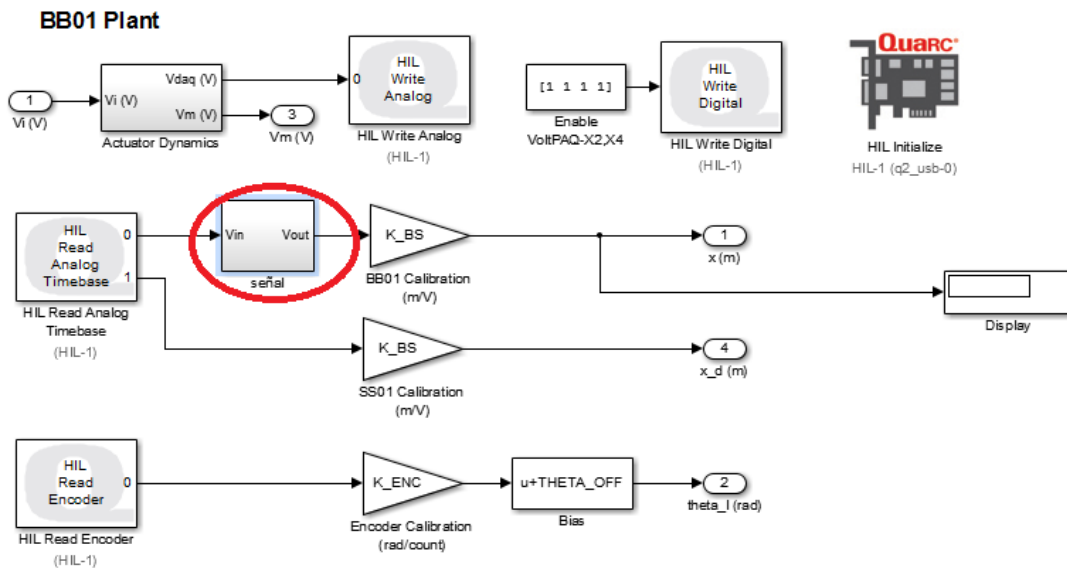


Figura 26. *q\_bb01\_pos\_vision/sensores/señal*

En el módulo señal, contenido en el módulo sensores de modelo *q\_bb01\_pos.mdl* se implementó un análisis, para introducir la señal del sensor diseñado con visión artificial.

Esta señal cumple con el mismo escalado para que el mínimo valor sea -4.5 y el máximo valor sea 4.5. Para determinar a qué valor de coordenada en píxeles corresponden estos valores mínimo y máximo, se determinó mediante la función de Matlab *Impixelinfo* desplazando el cursor del mouse sobre la imagen que estos píxeles en coordenadas eran: (100,240) y (600,240); es decir en el eje x los valores 100 (mínimo) y 600 (máximo).

Mediante marcas de cada 4 centímetros (4 cm), sobre la barra se tomaron los siguientes datos. (Marcas realizada con la referencia de un metro flexible). La bola tiene un diámetro de 2.5 cm, es decir, equivale a 31 píxeles.

Tabla 1. Relación valores sensor con píxeles a lo largo de la barra

Cm	Valor Sensor	pixel X
0	-4,5	100
4	-3,6	150
8	-2,7	200
12	-1,8	250
16	-0,9	300
20	0	350
24	0,9	400
28	1,8	450
32	2,7	500
36	3,6	550
40	4,5	600

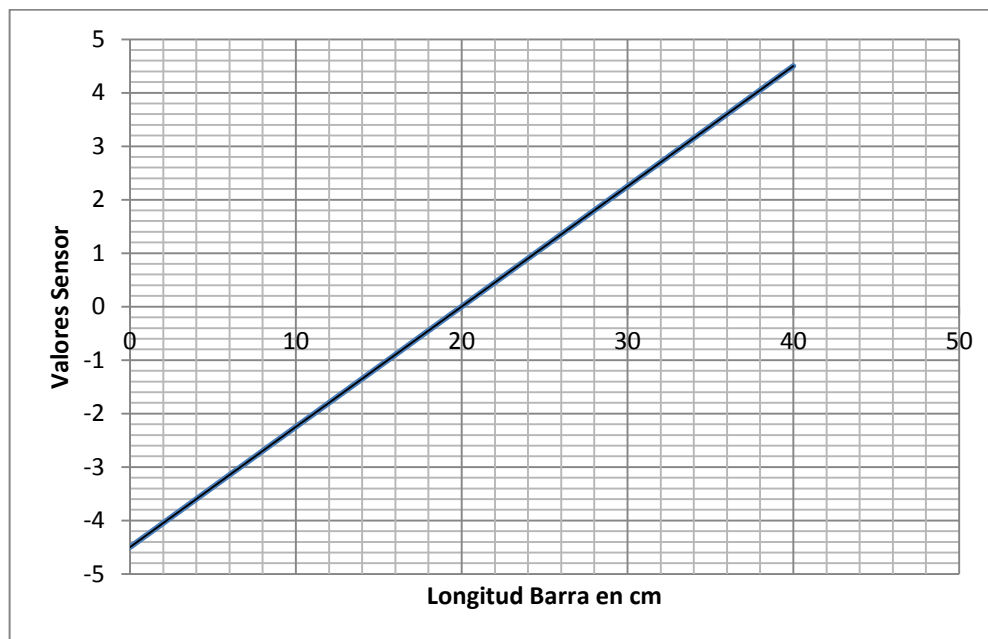


Figura 27. Datos longitud de la barra versus valores del sensor



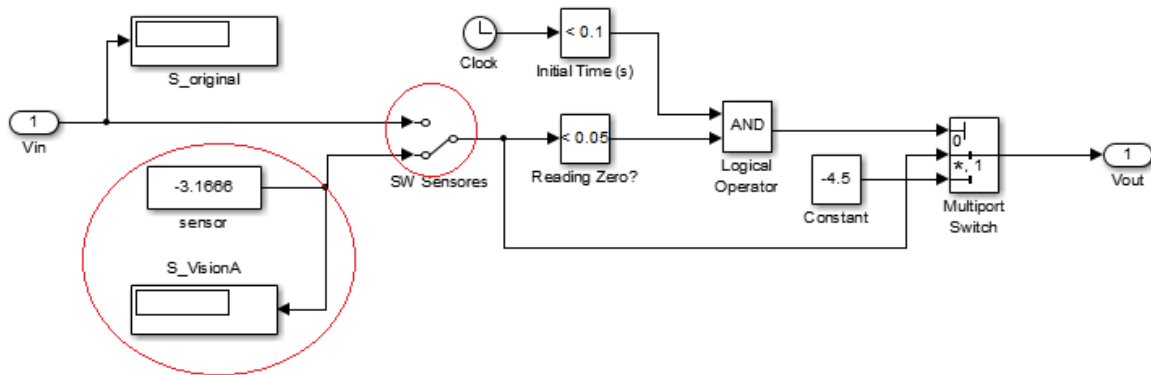


Figura 28. q\_bb01\_pos\_vision/sensores/señal/sensor

Lo que se encierra en círculos rojos en la Figura 28, son los módulos implementados para la lectura del valor generado por el código de visión artificial. La constante llamada “sensor” es el valor “instantáneo”, de la señal que envía el script con el código diseñado e implementado para el sensor de visión artificial. El SW sensores, se implementa para escoger si trabajar con el sensor original o con el sensor de visión artificial.

Desde Script de Matlab que contiene el código del sensor de visión artificial se envían los datos constantemente al sensor con el comando descrito en el Anexo 3.

Donde ‘y’ es el valor que varía entre -4.5 y 4.5, que se calcula mediante una ecuación lineal que se obtuvo de medir experimentalmente cada extremo de la barra y su relación con el pixel (coordenada en X pertenecía). En la ecuación ‘x’ es el pixel en X.

La ecuación que describe el cálculo de los valores que debe enviar el sensor de acuerdo al pixel es:

$$y = ((9/500)*x) - 6.3, \text{ ecuación lineal de la forma } y = mx + b.$$

La figura representa los el comportamiento de la ecuación anteriormente descrita.

‘Sv’ se llama a la variable que recibe el sensor en el modelo de Simulink q\_bb01\_pos que para efectos prácticos para no modificar el ejemplo de Quanser se renombró q\_bb01\_pos\_vision. El constante sensor está en la extensión: q\_bb01\_pos\_vision/sensores/señal/sensor. Nombres que también se modificaron del modelo original.

Para lograr el mecanismo para la integración del sensor con la planta Quanser se hicieron muchas pruebas utilizando todas las herramientas que ofrecen Simulink y Matlab. Se evaluaron muchas maneras de modificar el parámetro utilizando los bloques y funciones de datos como los módulos *From Workspace*, *From File*. Se intentó modificar el valor de la variable desde el workspace, incluso se creó una subsistema y un modelo en Simulink pero todas las pruebas experimentales fueron fallidas debido a que los datos no se modificaban “en línea” de acuerdo a las características propias y sus funcionalidades, es decir, estos sistemas y módulos entregaban un resultado al finalizar su ejecución y lo que se requiere es que constantemente en un ciclo infinito se esté procesando y mostrando el valor calculado. Finalmente, después de muchas pruebas se encuentra la forma de modificar una variable de simulink desde comandos ejecutados en un script con el comando *set\_param*. El anexo 3 muestra la solución implementada cuyos resultados fueron satisfactorios.

### **3.7 Interfaz HMI y Determinación de Posición**

En la interfaz HMI (Interfaz Hombre Maquina), se determinó que es el entorno de Matlab, ya que el módulo implementado es muy sencillo, básicamente es un código de Matlab registrado en un script y en un módulo de Simulink; es un bloque de una constante. Lo anterior, garantiza la generación del valor de la variable en tiempo real, lo que no se garantiza al realizar una interfaz gráfica. Se agrega una cámara conectada por USB al mismo computador que contiene la licencia de Quanser. La planta barra y bola se manipula de la misma forma como se trabajaría si no existiera este sensor. Se adiciona una base para montar la cámara que fue diseñada y construida con el fin de adquirir el video perpendicularmente a la barra y evitar la influencia de sombras y reflejos generados por la luz ambiente.



Figura 29. Modulo soporte cámara.

La Figura 29 muestra el soporte diseñado y construido, tiene 150 cm de alto, 80 cm de ancho y 45 cm de profundidad. Para determinar la altura se tuvo en cuenta la distancia a la que la imagen capturada abarcara el total de la barra. Con este soporte se logra eliminar fuentes de luz que modifiquen la intensidad reflejada que recibe la cámara. Se consiguen buenos resultados.

Con lo anterior se permite una interacción entre la persona que manipule el sistema y el actuador tal como se haría si no se tuviera este modulo adicional. No es necesario crear una interfaz grafica adicional a la ya existente de Quanser.

### **3.8 Documentación de las Pruebas de Posición**

Se realiza la conexión pertinente de Quanser, para realizar el experimento  $q\_bb01\_pos$  de acuerdo a la documentación de Quanser. Se conecta al puerto USB la tarjeta de adquisición de datos, procedimiento normal, como lo estipulan las guías del modelo barra y bola (Escuela Colombiana de Ingeniería, 2013). Se implementa el modulo con el bloque constante en Simulink para recibir la variable enviada por el sensor implementado en el código de Matlab.

Se ejecuta el script *setup\_srv02\_exp04\_bb01.m*, para que el modelo *q\_bb01\_pos* tome esos parámetros. Posteriormente se ejecuta el modelo y se verifica funcionamiento del sistema esclavo/maestro.

Una vez este corriendo el modelo, se debe ejecutar el script de Matlab con el código del sensor, estando previamente la cámara conectada a un puerto USB. (Tener en cuenta que la cámara está centrada sobre la barra de acuerdo a las cuadrículas que muestra el sistema de visión implementado). Cuando el sensor detecte movimiento, empieza a calcular la posición en la que se encuentra la bola. Cambiando el switch, se puede poner en funcionamiento el sensor de visión artificial. Cabe resaltar que todos los documentos deben estar incluidos en una misma carpeta.

Se tomaron datos experimentales de muchas pruebas moviendo la referencia (maestro), permitiendo que el esclavo rodara por toda la barra. Se tomaron medidas de ambos sensores en archivos *.mat*; que posteriormente se observaron en Excel. También se pudo visualizar con los bloques de “display”, las dos variables de los sensores (sensor original y sensor con visión artificial).

La Figura 30, muestra el modelo usado para extraer y comparar los datos del sensor original y el sensor implementado con visión artificial y analizarlos posteriormente en Excel. Visualmente mediante los display llamados *S\_original* y *S\_VisionA*, se puede comparar los datos registrados por los dos sensores.

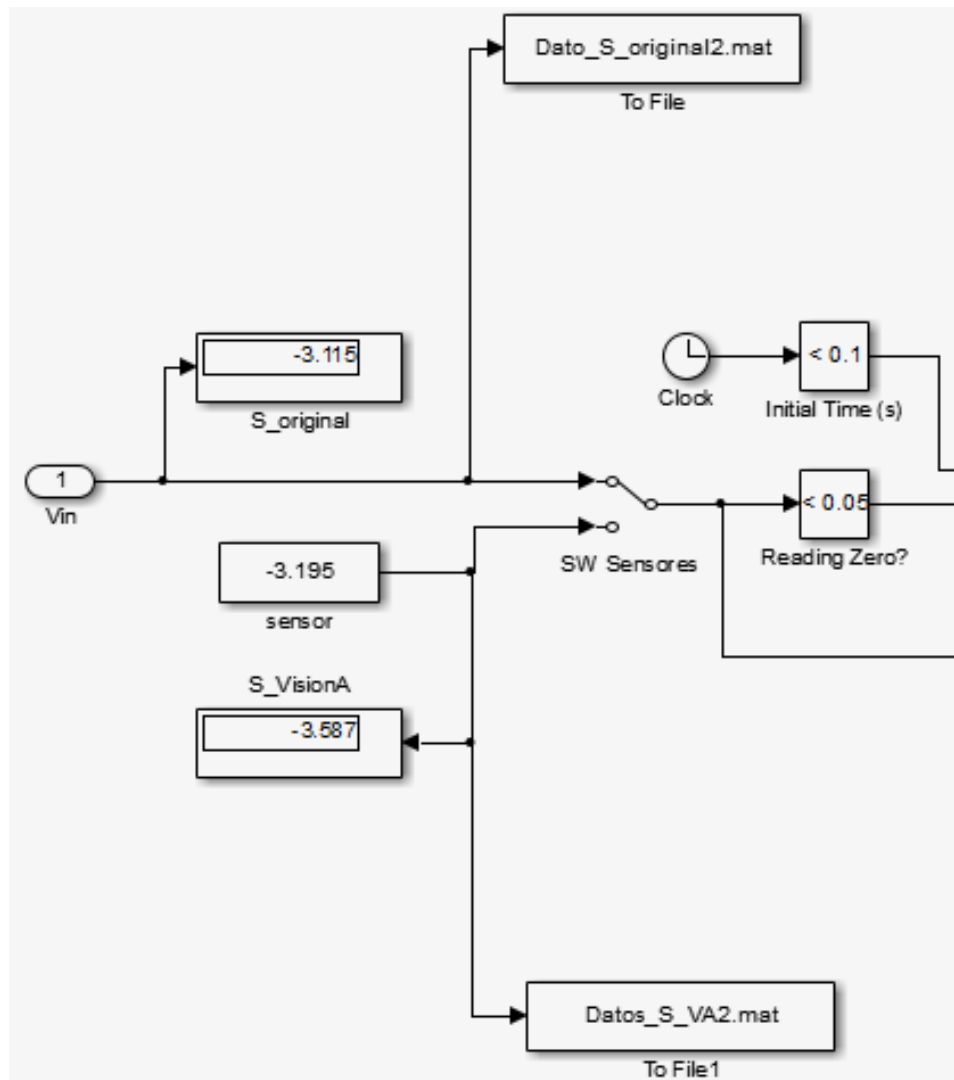


Figura 30. Toma de datos para comparar resultados

#### 4. Resultados y Contribución

Al realizar las pruebas para determinar la posición de la bola el sistema barra y bola de Quanser mediante el sensor de visión artificial implementado se puede determinar que el frame rate que uso la cámara mientras el sensor se ejecutó solo sin el modelo de Quanser fue de 60 fps debido a que los puertos USB que tiene el equipo que tiene la licencia del modelo barra y bola son convencionales y no son USB 3. De ser USB 3 sería de 120 fps.

Cuando se ejecuta el sensor en conjunto con el modelo de Quanser el la tasa de cuadros por segundo cae a la mitad. El procesamiento del equipo es mayor y la tarjeta de adquisición de datos de Quanser también se conecta a otro puerto USB del equipo.

El código del sensor de visión artificial ejecuta un ciclo en promedio cada 90 milisegundos. Promedio obtenido de los registros de tiempo que arroja la función *tic toc* de Matlab cada vez que se ejecuta un ciclo. Siendo esto coherente con el hecho de que la cámara tiene una tasa de 30fps. Esto da como resultado que si la cámara tiene una frecuencia de 30 cuadros por segundo en 90 milisegundos se pueden procesar 2 cuadros. Estos cuadros son los que se necesitan para hacer una resta en cada ciclo. Ver anexo 4.

Analizando los datos experimentales ya sea observando su comportamiento o con los resultados registrados en el archivo .mat y posteriormente interpretados en Excel, ver anexo 5. El sensor de visión artificial determina la posición de la bola con un error respecto al sensor original del 3,9% teniendo en cuenta que el sensor implementado no puede determinar la posición sino hasta después de que exista movimiento. El código implementado determina la posición mediante un algoritmo de detección de movimiento. El sistema está en constante movimiento puesto que al no concordar el valor de referencia en el maestro con el valor del esclavo el sistema oscila.

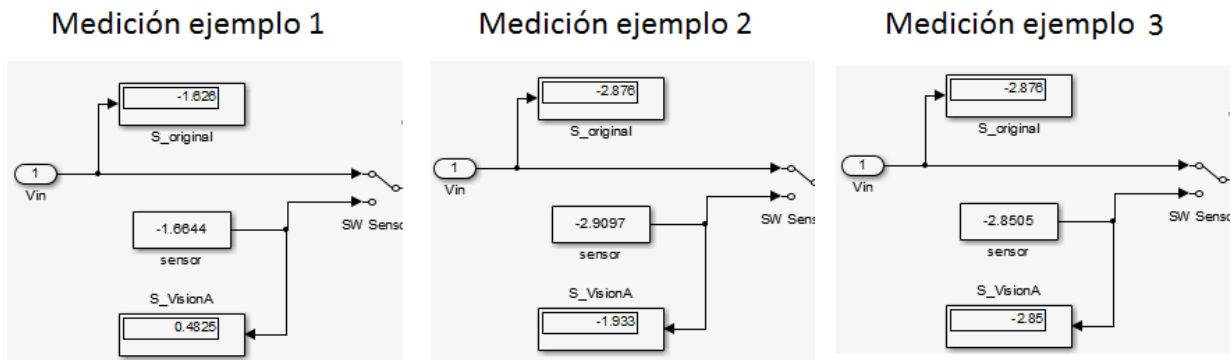


Figura 31. Ejemplos medición y comparación entre valores de los sensores

La Figura 31, muestra los resultados de medición de 3 ejemplos, donde se puede constatar los valores capturados por el sensor original y el sensor implementado mediante visión artificial. La constante (“*sensor*”), hace referencia al valor que envía el código implementado y el *displayS\_VisionA*, es la señal que toma el sistema del bloque constante mencionado anteriormente.

Se realizaron cuatro pruebas las cuales se obtuvieron los siguientes resultados registrados en la Tabla 2 después de extraer los valores de los archivos *mat*, para registrarlos y analizarlos (Anexo 5).

Tabla 2. Datos obtenidos entre los dos sensores

	Vt	Vpo	Vpa	e1	e2	e3	e4	e5	e6
Posición de la bola en la barra (cm)	valor sensor teórico	Valor promedio sensor Original	Valor promedio sensor visión artificial	Error absoluto del sensor original respecto a valor teórico	Error absoluto del sensor visión artificial respecto a valor teórico	Error relativo del sensor original respecto a valor teórico	Error relativo del sensor visión artificial respecto a valor teórico	Error absoluto entre sensores	Error relativo entre sensores
0	-4,5	-4,5281	-4,48775	0,0281	-0,01225	-0,00620569	0,00272965	-0,04035	0,00899114
4	-3,6	-3,60875	-3,6535	0,00875	0,0535	-0,00242466	-0,01464349	0,04475	-0,01224853
8	-2,7	-2,73625	-2,67725	0,03625	-0,02275	-0,01324806	0,00849753	-0,059	0,02203754
12	-1,8	-1,83475	-1,76225	0,03475	-0,03775	-0,01893991	0,02142148	-0,0725	0,04114059
16	-0,9	-0,91425	-0,8825	0,01425	-0,0175	-0,01558655	0,01983003	-0,03175	0,03597734
20	0	0,03125	0,0775	-0,03125	-0,0775	-1	-1	-0,04625	-0,59677419
24	0,9	0,909675	0,8983	-0,009675	0,0017	-0,01063567	0,00189246	0,011375	0,01266281

28	1,8	1,83775	1,7255	-0,03775	0,0745	-0,02054142	0,04317589	0,11225	0,06505361
32	2,7	2,711675	2,71625	-0,011675	-0,01625	-0,00430546	-0,00598251	-0,004575	-0,00168431
36	3,6	3,62475	3,65375	-0,02475	-0,05375	-0,00682806	-0,01471091	-0,029	-0,00793705
40	4,5	4,5065	4,49825	-0,0065	0,00175	-0,00144236	0,00038904	0,00825	0,00183405

Donde:

$$e1=V_{po}-V_t$$

$$e2=V_{pa}-V_t$$

$$e3=e1/V_{po}$$

$$e4=e3/V_{pa}$$

$$e5=V_{po}-V_{pa}$$

$$e6=e5/V_{pa}$$

El promedio del error relativo entre sensores (última columna), es de 0,039, es decir del 3,9%.

Por otra parte, existe la dificultad de la velocidad de transmisión al enviar el dato, esta es muy lenta, por tal motivo si la esfera se mueve muy rápido (más de 1 cm/seg), el sistema se pierde, debido a la respuesta que se envía al actuador es tardía. Aunque determine la posición relativamente bien, y el sistema detecte el movimiento para determinar dicha posición.

El sensor implementado en este punto del desarrollo no es óptimo, se requiere una comunicación más veloz y efectiva. Utilizar un puerto USB 3.0 para conectar la cámara permite sacar mayor provecho de este dispositivo. Sin embargo ninguno de los equipos de laboratorio a los que se puede conectar el sistema barra y bola tiene USB 3.0.

Se contribuye con un trabajo que implementa un sistema de visión artificial permitiendo determinar la posición de la bola en un sistema didáctico. Queda un antecedente con un código desarrollado como punto de referencia para comenzar trabajos futuros para determinar mediante visión artificial el movimiento o posición de una variable. Es adecuado, como base esencial para seguir desarrollando este proyecto.



El laboratorio de electrónica Escuela Colombiana de Ingeniería Julio Garavito; contará con una cámara especializada para desarrollar proyectos de visión artificial y prácticas de laboratorio, y una estructura física para el soporte de la cámara y proseguir en la consecución de un sensor óptimo partiendo como referencia la elaboración del proyecto planteado de tesis.

Como recomendación a estudios y trabajos futuros, se recomienda realizar y probar técnicas de reconocimiento de colores o formas, para determinar la posición de la bola, sin la necesidad que exista movimiento para poder tomar una decisión. También, se propone la idea de utilizar y manipular el modelo barra y bola de Quanser mediante Labview. Posiblemente utilizar el sensor diseñado en este software mejore considerablemente la velocidad de comunicación. En Simulink no se logró la comunicación por problemas en los bloques de adquisición de imágenes y su integración con Quanser.

## 5. Conclusiones

Se puede determinar la posición de la bola que se desliza sobre una barra, mediante un algoritmo en Matlab, que detecta el movimiento mediante la manipulación de los *frames* adquiridos por una cámara de video digital.

La comunicación entre el algoritmo del sensor y el modelo barra y bola, debe ser más rápida, para que el sistema funcione adecuadamente. El código implementado envía datos cada 90 milisegundos y el tiempo de muestra de Quanser es de 1 milisegundo.

Se pueden aplicar muchas más técnicas para mejorar el algoritmo implementado y desarrollar un sensor con visión artificial, el reto radica en usarlo como dispositivo de retroalimentación en otro sistema, con las restricciones de comunicación del equipo y las del tiempo real para la acción de control.

Las condiciones de luz natural y/o artificial alteran la medición de la posición de la bola incluso imposibilitando la detección completa del movimiento. Aunque en el diseño y construcción del soporte para la ubicación de la cámara se contempló este tema ya que es una estructura recubierta (tapas a los lados) si el lugar donde se realicen las pruebas y practicas tiene mucha iluminación los resultados son afectados negativamente.

Además de esta visión general del proyecto, se pretende también hacer una valoración de las dificultades encontradas durante el desarrollo de los mecanismos implementados en el trabajo de grado, así como del aprendizaje obtenido, para concluir con una exposición de futuros desarrollos, estudios y aplicaciones a efectuar en el sistema de barra y bola.

Este trabajo es un punto de partida para continuar con el reto de utilizar visión artificial en la forma de determinar la posición en un modelo barra y bola de Quanser como señal de retroalimentación.

Podemos concluir que la iluminación se considera el punto más importante para el desarrollo de proyectos de visión artificial con adquisición de imágenes, ya que de la intensidad y ubicación de esta va a depender el tipo de datos que recoja la cámara, variando el procesamiento de los mismos y por ende el resultado que arroje el sistema.

## 6. Bibliografía

- Abril Cañas, M., Valencia L., M., Restrepo, B. J., & Holguín, G. A. (2007). *Stesma De Visión Artificial Para El Registro de Densidad Peatonal En Tiempo Real*. Pereira: Recuperado de: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/5361> 6 de abril de 2015.
- Cardona Buitrago, F. E. (2013). *Las prácticas de laboratorio como estrategia didáctica*. Santiago de Cali, Colombia: Recuperado de: <http://bibliotecadigital.univalle.edu.co/bitstream/10893/6772/1/CD-0395428.pdf> 24 de abril de 2016.
- Coelho Simplício, R. J. (2011). *SIMBOLOGIA EM REALIDADE AUMENTADA*. Recuperado de: [http://repositorio.ul.pt/bitstream/10451/8839/1/ulfc104265\\_tm\\_Raul\\_Simplicio.pdf](http://repositorio.ul.pt/bitstream/10451/8839/1/ulfc104265_tm_Raul_Simplicio.pdf). Junio 6 de 2015.
- Cuevas López, F., Mora Reyes, M. A., & Olivera Mera, C. (2008). *Diseño, Construcción y control de un "Ball and Beam"*. Pachuca, México: Recuperado de: <http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/handle/231104/1832> Marzo 13 de 2015.
- Escuela Colombiana de Ingeniería. (2013). *Guías de uso para plantas Quanser*. Bogotá D.C., Colombia: Escuela Colombiana de Ingeniería, Julio Garavito.
- Gonzalez, R. C., & Woods, R. E. (2007). *Digital Image Processing*. Pearson Prentice Hall - Recuperado de: [http://www.imageprocessingplace.com/downloads\\_V3/dip3e\\_downloads/dip3e\\_sample\\_book\\_material/dip3e\\_chapter\\_02.pdf](http://www.imageprocessingplace.com/downloads_V3/dip3e_downloads/dip3e_sample_book_material/dip3e_chapter_02.pdf) julio 4 de 2015.
- K. S., F., R. C., G., & C. S. G., L. (s.f.). *Robotica: Control, Detección, Visión e Inteligencia*. MacGraw-Hill, Recuperado de: [http://www.itsbasicas.com/davila/Libro\\_Robotica.pdf](http://www.itsbasicas.com/davila/Libro_Robotica.pdf) 15 de septiembre de 2015.
- Lichtenberg, G., & Kwiatkowski, A. (s.f.). *Teaching Tool Based Controller Desing - The Control Challenge*. Institute of Control Systems Hambury University Technology.
- Moore, H. (s.f.). *Matlab Para Ingenieros*. Pearson Prentice Hall, Recuperado de: <https://drive.google.com/folderview?id=0B3OI02r6d7e8a0VXYzIZVWxfd3M&usp=sharing> 18 de octubre de 2015.
- Moriano, P., & Naranjo, F. (2012). *Modelado y Control de un nuevo sistema bola viga con levitación magnética*. Santiago de Cali, Colombia: Recuperado de: [http://apps.elsevier.es/watermark/ctl\\_servlet?\\_f=10&pident\\_articulo=90147406&pident\\_usuario=0&pcontactid=&pident\\_revista=331&ty=139&accion=L&origen=zonadelectura&](http://apps.elsevier.es/watermark/ctl_servlet?_f=10&pident_articulo=90147406&pident_usuario=0&pcontactid=&pident_revista=331&ty=139&accion=L&origen=zonadelectura&)

web=www.elsevier.es&lan=es&fichero=331v09n03a90147406pdf001.pdf Mayo 1 de 2015.

Padula, F., & Visioli, A. (2013). *An Approach for Teaching Automatic Control in a Laboratory of Mechatronics*. Brescia, Italia: University of Brescia.

Pajares, G., & de la Cruz, J. M. (2008). *Visión por computador, imágenes digitales y aplicaciones*. Ra-Ma.

Peña Baeza, A. (2010). *Módulo de Visión Artificial del Robot Humanoide HOAP3. Aplicación al Seguimiento de Objetos Móviles*. Madrid, España.: Recuperado de: [http://robots.uc3m.es/images/c/c4/Modulo\\_de\\_vision\\_artificial\\_del\\_robot\\_humanoide\\_HOAP3\\_Aplicacion\\_al\\_seguimiento\\_de\\_objetivos\\_moviles.pdf](http://robots.uc3m.es/images/c/c4/Modulo_de_vision_artificial_del_robot_humanoide_HOAP3_Aplicacion_al_seguimiento_de_objetivos_moviles.pdf) 6 de Abril de 2015.

Pérez Cisneros, M. A., & Wellstead, P. (s.f.). *Sistema de Balancin y Pelota:Principios básicos*. Guadalajara, México: Recuperado de: <http://www.control-systems-principles.co.uk/whitepapers/spanishwp/04BallandBeam1SP.pdf> 25 de Abril de 2015.

Sanabria S., J. J., & Archila D., J. F. (2011). *Detección y Análisis de movimiento usando Visión Artificial*. Bucaramanga, Colombia: Recuperado de: <http://www.redalyc.org/pdf/849/84922625031.pdf> 24 de junio de 2015.

Sanchez Escribano, M. G. (s.f.). *SISTEMA DE VISIÓN PARA UN ROBOT SOCIAL*. Recuperado de: [http://tierra.aslab.upm.es/documents/MThesis/MThesis\\_Guadalupe\\_Sanchez.pdf](http://tierra.aslab.upm.es/documents/MThesis/MThesis_Guadalupe_Sanchez.pdf) 21 de abril de 2015.

Szeliski, R. (2010). *Computer Vision:Algorithms and Applications*. Recuperado de: [http://szeliski.org/Book/drafts/SzeliskiBook\\_20100903\\_draft.pdf](http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf) febrero 3 de 2016.

Velez Serrano, J. F., Moreno Díaz, A. B., Sánchez Calle, Á., & Sánchez Marín, J. L. (2003). *Visión por Computador*. Recuperado de: <http://www.visionporcomputador.es/libroVision/libro.html> junio 1 de 2015.

## Anexo 1

Código en Matlab para comprobar las características de frecuencia de toma de cuadros de la cámara.

```
%CÁLCULO FRAME RATE DE LA CÁMARA
% se accede al dispositivo de adquisición de video
%vidobj=videoinput('winvideo',1); %cámara web
vidobj=videoinput('tisimaq_r2013', 1,'RGB24 (640x480)')% DMK23U618

% se configura el número de frames a cargar
vidobj.FramesPerTrigger = 50;

% se asigna delay para saltar los primeros frames
vidobj.TriggerFrameDelay = 5;

% se accede a la fuente de video del dispositivo
src = getselectedsource(vidobj);

% s determina los fps disponibles por la cámara.
frameRates = set(src, 'FrameRate')

%FRAME RATE ACUAL:
% se configura la cámara con la configuración más alta.
src.FrameRate = frameRates{1};
actualRate = str2num( frameRates{1} )
```

## Anexo 2

Código diseñado e implementado para determinar la posición de la bola que se desplaza sobre la barra.

```
%#####  
%Escuela Colombiana de Ingeniería  
%Maestría en Ingeniería Electrónica  
%Gustavo Virviescas Toledo  
%SENSOR VISIÓN ARTIFICIAL - POSICION BOLA SISTEMA BARRA Y BOLA QUANSER  
%#####  
  
%Construct a video input object.  
%vid=videoinput('winvideo',1); %como si fuera web cam  
vid=videoinput('winvideo',2); %con cámara imagingsource DMK 23U618 como si fuera webcam  
%640x480  
  
%#####  
% Access the device's video source.  
src = getselectedsource(vid);  
  
% Determine the device specific frame rates (frames per second) available.  
frameRates = set(src, 'FrameRate');  
  
%FRAME RATE ACUAL:  
% Configure the device's frame rate to the highest available setting.  
src.FrameRate = frameRates{1};  
actualRate = str2num( frameRates{1} )  
%#####  
  
%Se configura las opciones de adquisicion de video  
set(vid, 'FramesPerTrigger', inf);  
set(vid, 'ReturnedColorspace', 'rgb')  
vid.FrameGrabInterval = 1;  
%framegrabinterval significa que tomara cada X frame del stream de video adquirida  
%con start(vid) se activa la adquisicion, pero todavía se toma la primera foto  
start(vid)  
  
%MASCARA  
masc = imread('mascara.png');  
mascara = masc(:,:,1);  
  
% creamos un bucle que puede ser while  
%después de X frames adquiridos se sale del bucle para evitar  
%cargar memoria  
  
while(vid.FramesAcquired<=4000)  
tic %%inicio medición tiempo de ejecución por ciclo.  
%Adquirir y guardar frame actual
```

```

%framecta1 = getsnapshot(obj);
framecta1 = getsnapshot(vid);

%multiplicar imagenpor mascara
%framecta1 = inmultiply(framecta1,mascara);

for j=1:2
end
% Adquirir y guardar frame siguiente
%framecta2 = getsnapshot(obj);
framecta2 = getsnapshot(vid);

%multiplicar imagenpor mascara
%framecta2 = inmultiply(framecta2,mascara);
% Convertir frames de color a escala de grises NO ES NECESARIO!
%framecta1g = rgb2gray(framecta1);
%framecta2g = rgb2gray(framecta2);

%Obtener el valor absoluto de la diferencia entre frame actual y siguiente
imadif1= abs(framecta1-framecta2);

%Submuestrear imagen de diferencia a la mitad de la resolución espacial
%640x480
res1=imresize(imadif1,1); % substraer x

%Umbralizar la imagen de diferencias asignando un umbral estático
resbin=im2bw(res1,0.1);
%Multiplicar por la mascara
resbin=immultiply(resbin,mascara);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
resbin = bwareaopen(resbin,50);
bw = bwlabel(resbin, 8);
stats = regionprops(bw, 'BoundingBox', 'Centroid');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% guardo el primer valor umbralizado para posteriormente ir sumando y
% obtener el acumulado
%if i==1
%resbin1= resbin;
%end

%resbin1= resbin1 + resbin;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mascara%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%multiplicación x
mascara%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%x=resbin1*('1');
%x=immultiply(resbin1,MASC);
%x=immultiply(resbin1,masc2);

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Visualizar la imagen de diferencias umbralizada acumulada
% subplot(2,1,1);imshow(framecta1);title('ORIGINAL')
imshow(framecta1);title('ORIGINAL')%%visualiza el movimiento
%%subplot(2,3,2);imshow(framecta1g);title('ESCALA GRISES')
%subplot(2,1,2);imshow(resbin);title('huella ball')
%%subplot(2,3,4);imshow(masc2);title('mascara')
%impixelinfo
```

```
hold on
%rectangle('position',[20,150,280,75])
%maracar objeto
%centroide
for object = 1:length(stats)
bb = stats(object).BoundingBox;
bc = stats(object).Centroid;
rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
plot(bc(1),bc(2), '-m+')
%a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));
%set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
x=bc(1);
y=((9/500)*x)-6.3; %EL VALOR DE LA POSICIÓN
%%Enviar dato al modelo q_bb01_pos de quanser a la constante llamada sensor
Sv=num2str(y);
set_param('q_bb01_pos_vision/sensores/señal/sensor','Value',Sv)
pause(0.002)
```

```
end
%%%linea centro
v1=[0 240];
v2=[640 240];
plot([v1(1) v2(1)],[v1(2) v2(2)],'g')
```

```
%%%lineas bordes
p1=[100 0];
p2=[100 480];
plot([p1(1) p2(1)],[p1(2) p2(2)],'g')
```

```
p3=[610 0];
p4=[610 480];
plot([p3(1) p4(1)],[p3(2) p4(2)],'g')
```

```
hold off
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
flushdata(vid);% borra buffer/removes all buffered image frames from memory.
toc %%medir tiempo de ejecucion por ciclo
end
%Remove video input object from memory and close preview
delete(vid);
```



### Anexo 3

Comando para enviar los datos al constante *sensor*, desde el script de Matlab a Quanser en Simulink.

```
y=((9/500)*x)-6.3; %EL VALOR DE LA POSICIÓN
%%Enviar dato al modelo q_bb01_pos de quanser a la constante llamada sensor
Sv=num2str(y);
set_param('q_bb01_pos_vision/sensores/señal/sensor','Value',Sv)
```

## Anexo 4

Datos con el tiempo empleado en cada ciclo del código del sensor.

CICLO	RESULTADO FUNCIÓN tic toc	TIEMPO (Segundos)
1	Elapsed time is 0.090157 seconds.	0,090157
2	Elapsed time is 0.078567 seconds.	0,078567
3	Elapsed time is 0.096393 seconds.	0,096393
4	Elapsed time is 0.097640 seconds.	0,097640
5	Elapsed time is 0.089099 seconds.	0,089099
6	Elapsed time is 0.088721 seconds.	0,088721
7	Elapsed time is 0.079735 seconds.	0,079735
8	Elapsed time is 0.098039 seconds.	0,098039
9	Elapsed time is 0.088190 seconds.	0,088190
10	Elapsed time is 0.088628 seconds.	0,088628
11	Elapsed time is 0.077413 seconds.	0,077413
12	Elapsed time is 0.094466 seconds.	0,094466
13	Elapsed time is 0.081870 seconds.	0,081870
14	Elapsed time is 0.116975 seconds.	0,116975
15	Elapsed time is 0.082996 seconds.	0,082996
16	Elapsed time is 0.103835 seconds.	0,103835
17	Elapsed time is 0.105692 seconds.	0,105692
18	Elapsed time is 0.093483 seconds.	0,093483
19	Elapsed time is 0.084778 seconds.	0,084778
20	Elapsed time is 0.115112 seconds.	0,115112
21	Elapsed time is 0.084606 seconds.	0,084606
22	Elapsed time is 0.116595 seconds.	0,116595
23	Elapsed time is 0.083712 seconds.	0,083712
24	Elapsed time is 0.093392 seconds.	0,093392
25	Elapsed time is 0.116999 seconds.	0,116999
26	Elapsed time is 0.108154 seconds.	0,108154
27	Elapsed time is 0.076123 seconds.	0,076123
28	Elapsed time is 0.092346 seconds.	0,092346
29	Elapsed time is 0.088377 seconds.	0,088377
30	Elapsed time is 0.078207 seconds.	0,078207
31	Elapsed time is 0.097428 seconds.	0,097428
32	Elapsed time is 0.090441 seconds.	0,090441
33	Elapsed time is 0.077458 seconds.	0,077458
34	Elapsed time is 0.095869 seconds.	0,095869
35	Elapsed time is 0.076812 seconds.	0,076812
36	Elapsed time is 0.088453 seconds.	0,088453
37	Elapsed time is 0.078470 seconds.	0,078470
38	Elapsed time is 0.097303 seconds.	0,097303
39	Elapsed time is 0.089680 seconds.	0,089680
40	Elapsed time is 0.079275 seconds.	0,079275
41	Elapsed time is 0.097640 seconds.	0,097640
42	Elapsed time is 0.088258 seconds.	0,088258
43	Elapsed time is 0.077879 seconds.	0,077879
44	Elapsed time is 0.097182 seconds.	0,097182
45	Elapsed time is 0.087543 seconds.	0,087543
PROMEDIO		0,090906

## Anexo 5

Datos obtenidos del sensor original y el sensor con visión artificial.

	Medición 1	Medición 2	Medición 3	Medición 4	
Posición de la bola en la barra (cm)	Sensor Original	Sensor Original	Sensor Original	Sensor Original	Promedio
0	-4,5014	-4,509	-4,505	-4,597	-4,5281
4	-3,615	-3,605	-3,61	-3,605	-3,60875
8	-2,795	-2,715	-2,705	-2,73	-2,73625
12	-1,887	-1,825	-1,81	-1,817	-1,83475
16	-0,925	-0,916	-0,905	-0,911	-0,91425
20	0,015	-0,005	0,09	0,025	0,03125
24	0,9097	0,909	0,915	0,905	0,909675
28	1,876	1,835	1,81	1,83	1,83775
32	2,7097	2,715	2,705	2,717	2,711675
36	3,644	3,625	3,615	3,615	3,62475
40	4,525	4,495	4,501	4,505	4,5065

	Medición 1	Medición 2	Medición 3	Medición 4	
Posición de la bola en la barra (cm)	Sensor Visión Artificial	Sensor Visión Artificial	Sensor Visión Artificial	Sensor Visión Artificial	Promedio
0	-4,459	-4,48	-4,525	-4,487	-4,48775
4	-3,689	-3,705	-3,595	-3,625	-3,6535
8	-2,715	-2,695	-2,615	-2,684	-2,67725
12	-1,815	-1,791	-1,699	-1,744	-1,76225
16	-0,895	-0,995	-0,825	-0,815	-0,8825
20	-0,185	0,325	0,215	-0,045	0,0775
24	0,9015	0,885	0,897	0,9097	0,8983
28	1,795	1,687	1,685	1,735	1,7255
32	2,702	2,693	2,685	2,785	2,71625
36	3,585	3,695	3,699	3,636	3,65375
40	4,45	4,515	4,495	4,533	4,49825

## Sensor Original Vs. Sensor V. Artivicial

