

# CONTROL DE PÉNDULO INVERTIDO UTILIZANDO TEORÍAS DE CONTROL CLÁSICO Y TÉCNICAS DE VISIÓN

***Bruno González Sánchez***

Instituto Tecnológico de Celaya  
*12030373@itcelaya.edu.mx*

***Martín Laguna Estrada***

Instituto Tecnológico de Celaya  
*Martin.laguna@itcelaya.edu.mx*

## **Resumen**

Actualmente bastantes de los mecanismos industriales se operan de manera manual o se pueden programar para que realicen tareas repetitivas o siempre realicen una misma función, muchos de ellos no se adaptan a ciertos cambios como el ser detectados a través de captura de imágenes y en base a esta captura, mandar señales que manipulen el mecanismo. Al utilizar medios visuales de monitoreo, se puede expandir su uso a aumentar la velocidad de producción en las líneas de ensamble y selección, todo esto solo con la captura de imágenes. Aunado a las teorías de control clásico aplicadas para controlar el movimiento de un péndulo invertido, se presenta adicionalmente el diseño y desarrollo de un prototipo de esta naturaleza. Para el logro de tal fin, se utilizó el software LabVIEW y una placa de adquisición y escritura de datos para el control de dicho péndulo, así como módulos de visión para estabilizar su movimiento.

**Palabra(s) Clave(s):** control, escritura de datos, LabVIEW, módulos de visión, péndulo invertido, placa de adquisición.

## **1. Introducción**

Ciertos procesos industriales requieren de un monitoreo no invasivo, el monitoreo por visión es una alternativa que ofrece cero contacto, a diferencia de

otros sensores que tienen que hacerlo directamente con la materia dentro del proceso, lo que puede ocasionalmente contaminar el producto. Con lo anteriormente mencionado, se pretende realizar un prototipo de péndulo invertido del que no estén sujetos sensores invasivos como puede ser el monitoreo con una cámara de video. Un péndulo invertido es un sistema que puede desestabilizarse de manera muy brusca al ser perturbado externamente. Para poder controlar esta perturbación, se debe generar un control que permita mandar señales lo más rápido posible.

El procesamiento de imágenes es más tardado que el procesamiento de una señal analógica de voltaje por ejemplo, aun así al analizar una imagen podemos monitorear más variables como distancias colores, luz, posición y formas. Este retraso en el procesamiento de la información representa un verdadero reto ya que el sistema debe responder rápidamente para lograr mantener el péndulo siempre en posición vertical.

## **2. Justificación**

Actualmente ya existen cámaras capaces de monitorear temperatura y colores que están fuera del espectro de frecuencia del ojo humano, lo que amplía exponencialmente las aplicaciones de técnicas de visión. Mediante este proyecto se pretende manipular la posición un sistema por medios visuales y monitorear otras características del mismo a largo plazo, como posiciones variables e incluso toma de decisiones propias, por lo que resulta un buen experimento dentro del campo de la visión artificial.

Los sistemas de visión artificial comprenden una tecnología cuyo desarrollo ha crecido significativamente en los últimos años, fomentando el reemplazo de múltiples sensores hacia la adquisición de información a través del procesamiento online de imágenes. El creciente interés en esta tecnología reside en la elección y determinación de la misma como un recurso estratégico para desarrollos asociados al control y monitoreo de dispositivos mecánicos sub-actuados y robóticos. Con el objetivo de analizar la posibilidad de reemplazar sensores tradicionales por técnicas de procesamientos de imágenes en sistemas que

requieran la aplicación de control de manera continua, precisa y veloz, se propone un sistema de medición basado en visión artificial para el monitoreo dinámico de un péndulo invertido. Debido a sus no linealidades, su inestabilidad intrínseca y su modelo sencillo, el péndulo invertido suele ser utilizado como modelo de referencia para analizar el desempeño de los sistemas de control.

### **3. Objetivos**

El objetivo principal de este proyecto es realizar el control de un péndulo invertido utilizando como sensor de retroalimentación una cámara para adquirir imágenes y procesarlas en un software. Mediante técnicas de instrumentación manipular la información obtenida de cada imagen y procesarla para mandar señales que controlen la posición del péndulo. El propósito es que el péndulo sea capaz de mantenerse vertical en todo momento y responder de manera rápida ante las perturbaciones externas. Se pretende de igual manera, Implementar y diseñar un control (PI, PD, PID) que permita que el sistema sea rápido y estable.

### **4. Desarrollo**

Los elementos de un péndulo invertido lo constituyen un elemento móvil en la base y una varilla sujeta al elemento móvil como ya se ha mencionado anteriormente. Se ha optado por un diseño que se adapte al mecanismo de cremallera de una impresora. En la figura 1 se observa la estructura que contiene la cremallera y un motor a pasos el cual originalmente ocasionaba el movimiento del cartucho de tinta. Se puede observar que el mecanismo se ha sujetado a una base rectangular de madera para evitar que los movimientos bruscos del motor se presenten en la estructura.

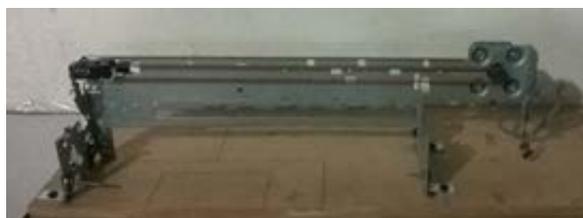


Figura 1 Mecanismo de cremallera.

La cremallera que se encontraba fija en el cartucho de tinta se ha modificado con ayuda de un mototool para que a éste se pueda sujetar una varilla de madera que gira libremente. En la figura 2 se puede observar el resultado de la modificación del cartucho.



Figura 2 Cartucho modificado.

Al colocar la base junto con la cremallera se observó que por el peso de algunos elementos no se sostenía de manera recta, para compensar esta situación, se le agregaron unas pequeñas guías en la parte trasera para que la base se pueda sujetar de la propia estructura y de esta manera la estructura no presente perturbaciones que pudieran impedir el movimiento de la base. En la figura 3 se observa la base desde la parte de atrás para poder apreciar las guías que se han colocado. En la figura 4 se muestra la base colocada en la cremallera.



Figura 3 Base con perspectiva.

El elemento de visión es quizá el más importante en el proyecto, debido a que será el elemento de retroalimentación, fundamental para un buen control, LabVIEW

será el software utilizado para la adquisición y el procesamiento de la imagen con ayuda del módulo de visión.



Figura 4 Base en cremallera-visión.

El primer paso es obtener la señal visual a través de una webcam para posteriormente procesarla. En esta etapa del proceso hay que considerar no solo la resolución sino también las características propias del equipo de cómputo, debido a que si se adquiere una webcam con una resolución muy alta y el equipo con él se cuenta tiene características de desempeño bajas entonces al utilizar la webcam en la computadora observaremos lentitud al momento de capturar el video. Debido a los altos tiempos de procesamiento que necesita el equipo para mostrar la imagen de video, una cámara de mediana resolución fue la elegida para este fin. Una vez adquirida la imagen es necesario instalar previamente los controladores del dispositivo. Finalizado este proceso, se puede trabajar con la cámara web sin mayor problema dado que se tiene control sobre las acciones que se realizan hasta este punto.

El objetivo principal del elemento de visión es transformar los datos de visión en datos numéricos y obtener una medición o una posición que se traduzca en alguna característica del sistema monitoreado, de esta manera dicho dato numérico se puede utilizar posteriormente como un dato para programación.

La cámara se posicionó a una distancia adecuada de la estructura del péndulo tratando de ubicar la webcam lo más centrada posible. En la figura 5 se observa el posicionamiento de la cámara frente al péndulo.



Figura 5 Posicionamiento de la cámara.

### **Configuración del asistente de adquisición de imagen**

Mediante el uso del software LabVIEW y con la cámara conectada a uno de los puertos USB de la computadora, se colocó el bloque “vision acquisition”, que es el asistente para inicializar una cámara disponible en nuestro equipo (figura 6).



Figura 6 Vision acquisition.

Instantáneamente después de colocar el bloque, nos aparece una ventana donde se muestran los dispositivos de video disponibles y se elige el dispositivo correcto. En la figura 7 se observa lo anteriormente mencionado. Continuando con la configuración, se selecciona el tipo de adquisición continua, el cual permite que el proceso programado trabaje en un ciclo que se repite indeterminadamente. Aunado a esto, cada vez que el ciclo se repita, se adquiere una nueva imagen del dispositivo de video. En la figura 8 se observa la selección de este tipo de adquisición. Al seleccionarlo se continúa con el asistente de adquisición de imagen.

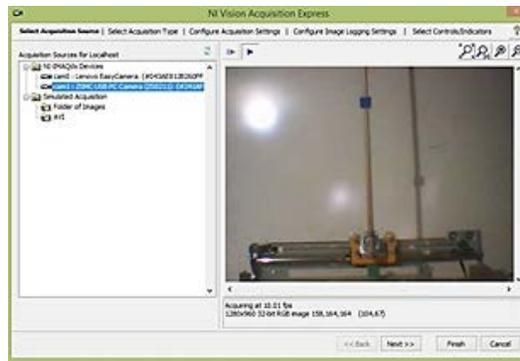


Figura 7 Selección de dispositivo de video.

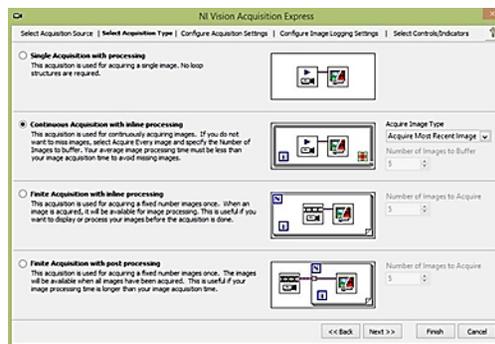


Figura 8 Tipo de adquisición.

En el siguiente paso se selecciona la resolución de video en la ventana de configuración de adquisición de imagen, los fps (por sus siglas en ingles frames per second), son una medida de velocidad para la captura de fotogramas de video. Se ha seleccionado una resolución de 800x600 a 30 fps, se prueba la adquisición de la imagen y se finaliza el asistente. En la figura 9 se observa la ventana de configuración de adquisición de imagen.

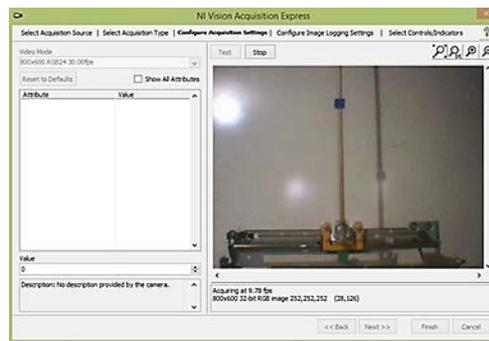


Figura 9 Ventana de configuración de adquisición de imagen.

Al finalizar el asistente, automáticamente se genera un Instrumento virtual en el diagrama de bloques, el cual ya está configurado con las opciones seleccionadas anteriormente y en el panel de control se observa una ventana con la imagen que se captura a través de la cámara. En la figura 10 se observa el VI (Virtual Instrument) generado y en la figura 11 se observa el panel de control.

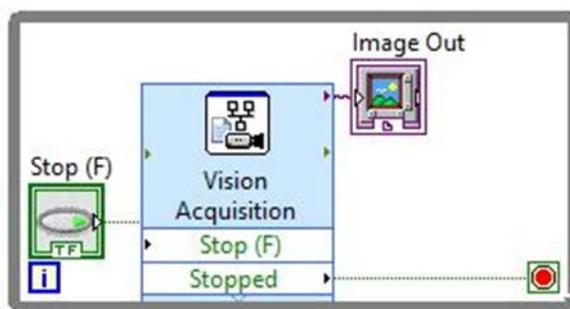


Figura 10 VI generado.

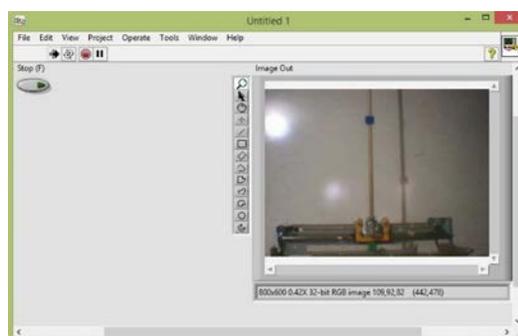


Figura 11 Panel de control.

### Método de identificación de color

Si se observa la figura 5, notaremos la presencia de dos pequeños elementos (referencias), uno de color azul, colocado en la varilla y otro color verde ubicado en la base, el elemento verde únicamente se mueve de forma horizontal siguiendo una trayectoria recta, mientras que el elemento azul girará con respecto a la base pudiendo ésta colocarse en puntos diferentes.

Este método consiste principalmente en ubicar colores en el plano de la imagen. El bloque "Vision Assistant" permite manipular la imagen para obtener información que después puede ser utilizada con otros fines. Este bloque requiere un dato de

entrada, el cual se trata de una imagen, cuando se coloca el bloque en el diagrama de bloques, automáticamente se abre el asistente de visión, en la figura 12 se muestra el bloque del asistente de visión mientras que en la figura 13 se aprecia la ventana que se abre al colocar el asistente de visión.

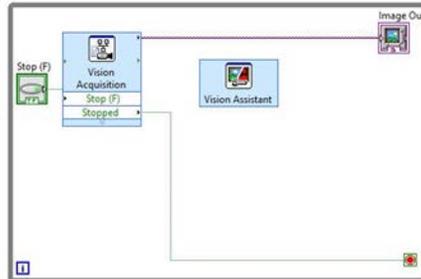


Figura 12 Bloque Vision Assitant.

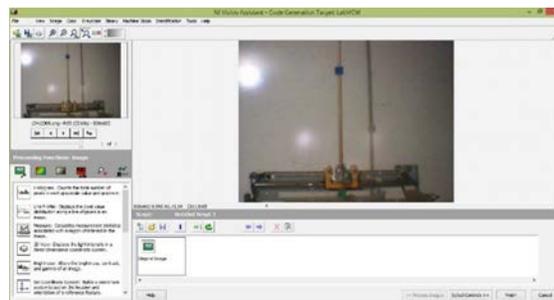


Figura 13 Asistente de visión.

Dentro de las funciones de procesamiento se busca aquella dedicada al color, que se puede identificar igualmente por titularse "Color". En la figura 14 se observan las herramientas de color. En la figura 15 se observa la herramienta a utilizar titulada "Color Pattern Matching" y de acuerdo a la descripción de la herramienta, ésta realiza un chequeo para buscar un modelo o muestra de un color en una imagen completa o una región de ella. Una de las ventajas es que se puede elegir la muestra de color directamente de la imagen de video, por lo que no es necesario introducir el color en un código RGB o CMYK. Esta herramienta seleccionada permite crear una nueva muestra de color y como se ha mencionado anteriormente, se ubicaron dos referencias de color, una azul en la parte alta de la

varilla y otra verde ubicada en la base. De estas dos referencias se crean muestras de color con la idea de que se puedan ubicar los colores en la imagen de video.

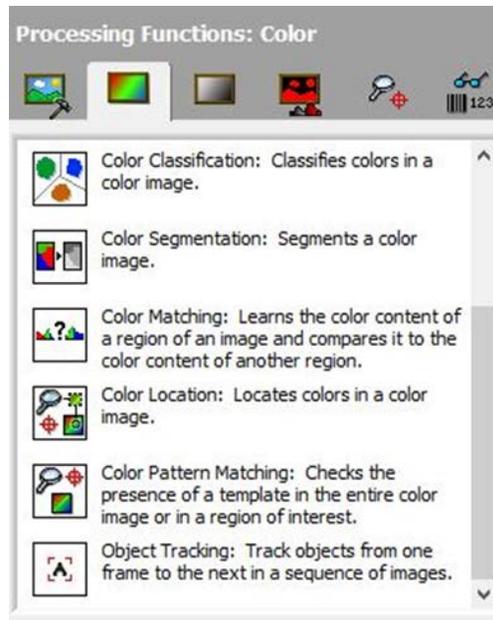


Figura 14 Funciones de procesamiento de color.

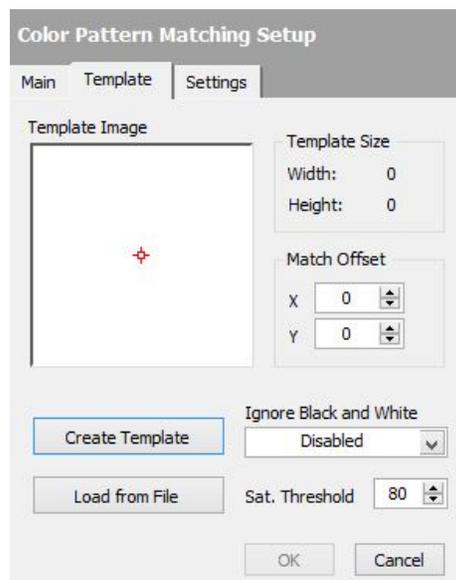


Figura 15 Configuraciones de Color Pattern Matching.

Para crear una muestra, basta con hacer clic en el botón "Create Template", después en la imagen seleccionamos la muestra de color encerrándolo en un

rectángulo de color rojo, al mismo tiempo se muestra un rectángulo verde, la región dentro de este rectángulo es en donde se realiza la búsqueda y ubicación de la muestra. Se pueden modificar las dimensiones de este rectángulo aumentando o disminuyendo la región predeterminada. En la figura 16 se observa la selección de la muestra de color y la región de búsqueda de la muestra. Al generar la muestra se observan también una ampliación de la misma y la ubicación del centro de ésta (figura 17).

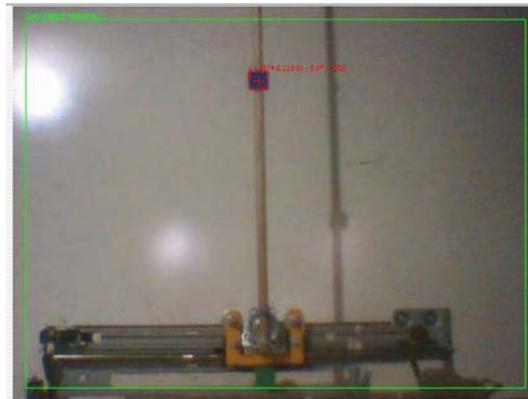


Figura 16 Creando muestra de color.

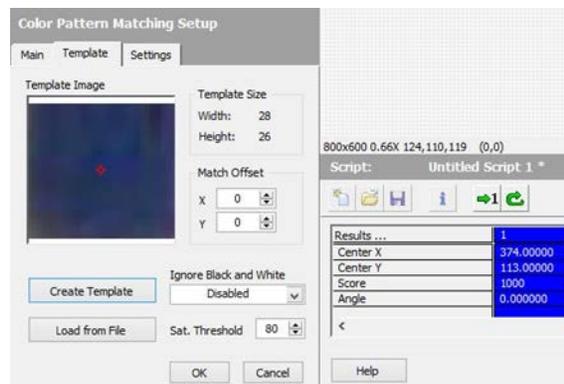


Figura 17 Datos de la muestra generada.

Ahora se selecciona dentro del grupo de funciones de procesamiento de imagen aquella que se identifica por el título "Imagen" (figura 18). La función que se utiliza lleva por título "Set Coordinate System", la cual de acuerdo a la descripción dada, se trata de una herramienta que permite establecer un origen de un sistema de

coordenadas basado en la localización y orientación de alguna referencia, al seleccionar la herramienta, se puede establecer un nombre al sistema de coordenadas y también nos permite seleccionar diferentes modos de movimiento que se pueden monitorear. El objetivo principal es establecer un origen de coordenadas, para que en base a este origen se realice el monitoreo de la muestra que se ha generado anteriormente. En la figura 19 se observa que se ha establecido el origen de coordenadas justo en la parte donde se encuentra la referencia azul.

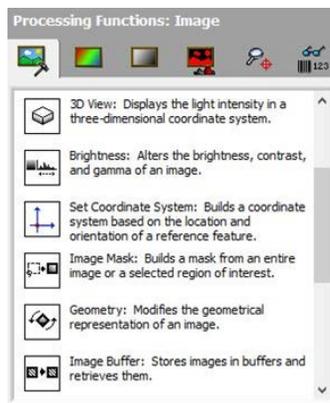


Figura 18 Funciones de procesamiento de imagen.

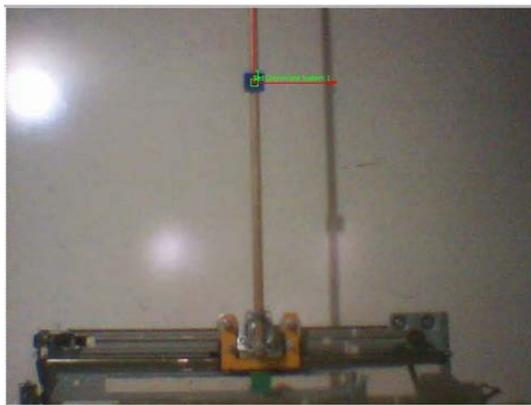


Figura 19 Origen de sistema de coordenadas.

Ahora se realiza el mismo método para la referencia verde ubicada en la base del péndulo, después se podrá visualizar una cadena (figura 20), en la cual se observan las configuraciones que se han realizado. Es importante mencionar que

las muestras de color no serán siempre las mismas a pesar de que físicamente no se hayan cambiado las referencias, se trata más que nada de la iluminación del ambiente que ocasiona que el color sea más brillante o más opaco. Esto no resulta un problema tan grave, ya que se puede modificar la muestra de color en cualquier momento así como cualquier otro atributo que hasta ahora se ha configurado, para ello solo basta hacer doble clic en alguno de los iconos encadenados, para poder modificar los atributos del mismo.

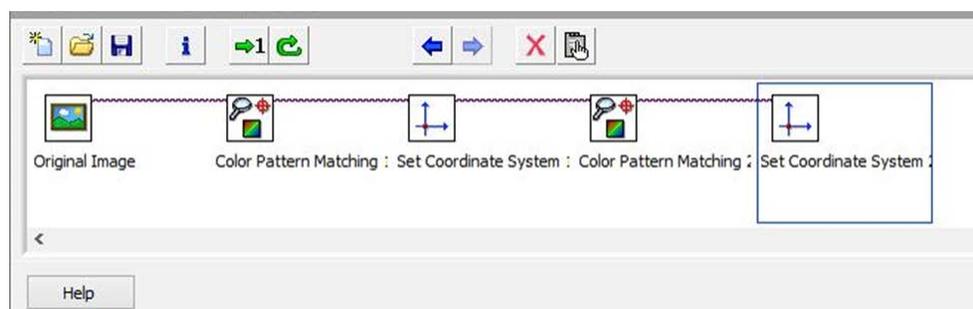


Figura 20 Procesos.

### Método de identificación de bordes

Este método consiste en identificar los bordes de una figura y en base a ellos realizar mediciones. Esto se puede realizar dentro del asistente de visión. Antes de comenzar se prepara un ambiente necesario ya que ahora el método no se basará en una imagen a color sino en una imagen con tonalidades grises. Para un mejor rendimiento, se ha trabajado sobre un fondo blanco y se han colocado máscaras blancas a ciertas partes de la estructura del péndulo, para cambiar el color de la vara a blanco (figura 21). Ahora las referencias son 4 y todas de la misma tonalidad.

Para obtener la imagen en tonalidades grises, dentro del asistente de visión, se seleccionan las funciones de procesamiento tituladas "Color" que anteriormente fueron utilizadas para generar muestras de color, en esta ocasión nos dirigimos a la herramienta "Color Plane Extraction" la cual permite extraer hasta 3 tipos de planos de una imagen a color, lo importante al elegir alguno de los planos es asegurarse de que el color azul de las referencias no se pierda y que se distinga

lo más posible, de igual manera se busca que las sombras y brillos se pierdan para que no afecten en el trabajo posterior.

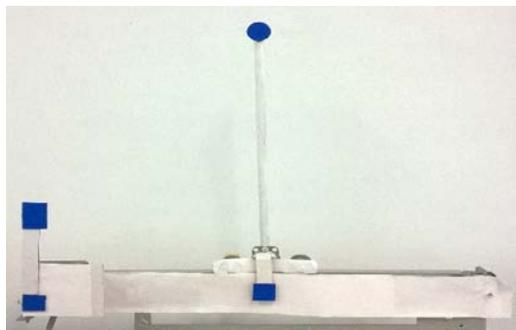


Figura 21 Estructura del péndulo modificada.

A continuación en las funciones de procesamiento buscamos la que se titula “Machine Vision” en la que encontraremos la herramienta Clamp o Rake, la cual realiza una medición en pixeles de la distancia entre dos bordes separados, el objetivo al utilizar esta herramienta es realizar una medición horizontal de los bordes exteriores de la distancia que hay entre la referencia superior izquierda y la referencia en la vara, y otra también de bordes exteriores entre la referencia inferior izquierda y la referencia en la base, en figura 22 se observa la captura de video y se puede apreciar cómo se marca con líneas rojas los bordes que ha detectado y de los cuales se está realizando lo medición.

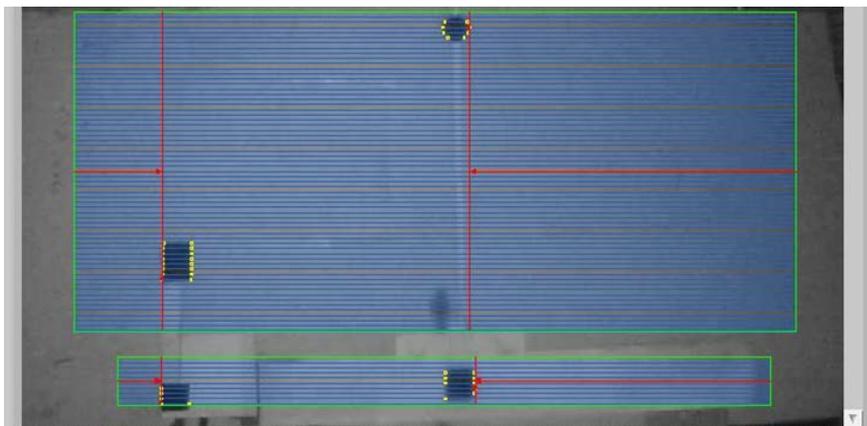


Figura 22 Mediciones de bordes.

En la figura 23 se observa el momento en que se ha inclinado el péndulo para observar si las mediciones arrojadas son congruentes y que no presenten problemas.

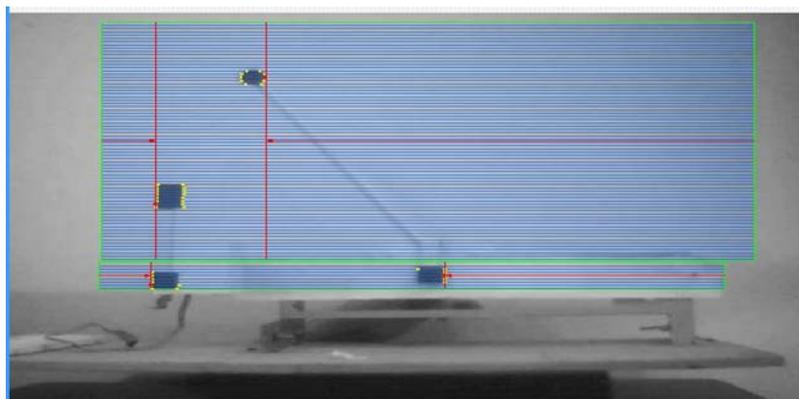


Figura 23 Adquisición de la imagen en el panel frontal.

### Motor a pasos bipolar

En primera instancia se utilizó el motor integrado en el mecanismo de la impresora, este se trata de un motor a pasos bipolar en la figura 24 se puede observar el esquema de un motor a pasos bipolar.

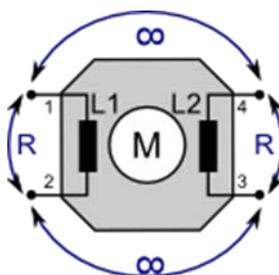


Figura 24 Esquema motor a pasos.

Para hacer funcionar un motor a pasos bipolar hay que alimentar las bobinas en una secuencia específica. En una primera prueba se utilizó la plataforma Arduino y el driver L293D. Se utilizó LabVIEW y se programó un VI para hacer funcionar el mismo motor a pasos pero esta vez utilizando una tarjeta DAQmx.

Para este caso se utilizó un contador y una máquina de estados, además de un control para variar el tiempo en que salta de estado en estado.

## **Motor de corriente continúa**

En un motor de CC mientras más grande sea el potencial eléctrico obtendremos más velocidad y torque, de esta forma lograremos variar la velocidad y torque del motor para las aplicaciones necesarias. En muchas ocasiones no se cuenta con un sistema que varíe el voltaje entre los rangos necesarios, para ello existe otra opción para variar la velocidad y torque sin la necesidad de variar el potencial eléctrico. En este proyecto se utilizó un pequeño motor de imanes permanentes. La modulación por ancho de pulsos o PWM (Pulse Width Modulation) por sus siglas en inglés, es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, su construcción se basa en la utilización de comparadores, y su aplicación puede ser para el control de motores CC o para convertidores de voltaje. Una ventaja importante del control de PWM sobre el control por variación de voltaje es que se ha demostrado que es más eficiente en términos de calentamiento ya que para hacer girar el motor no utiliza el 100% de la señal si no cantidades que pueden variar de 0 a 100% lo que hace que haya momentos de fracciones de tiempo muy pequeños (en el orden de los mili segundos o inferiores) en que el motor se encuentre girando pero la señal se encuentre en un estado "Off", esto ocasiona que no se suministre toda la potencia para hacer girar el motor lo que se traduce en menor calentamiento, entonces se entiende que mientras menor sea el ciclo de trabajo menos girará el motor, así que si se tiene un ciclo de trabajo del 100% el motor se encontrará en su límite máximo de velocidad.

La plataforma de desarrollo electrónico Arduino® ofrece en sus tarjetas salidas digitales y PWM, las cuales sustituyen en cierta medida a las analógicas por lo mencionado anteriormente, Arduino ofrece una señal PWM a 16 MHz, segmentado en 255 partes aunado a esto dentro del software LABView se puede programar utilizando el bloque de Arduino. En tres sencillos pasos se puede describir la escritura de una señal PWM en una plataforma Arduino desde LABView:

- Inicializar la placa Arduino con el bloque "INIT", únicamente se debe de colocar fuera del ciclo "While" y de preferencia en la parte izquierda, por defecto está asignado para una placa Arduino Uno. Si no es la placa con la

que trabajamos, generar una constante en la terminal “Board Type” y seleccionar la placa correcta.

- Colocar el bloque de escritura PWM, asignar el pin de escritura, éste tiene que ser un pin habilitado para escritura PWM que varía entre placa y placa, también asignar un valor entre 0 a 255 para la escritura del PWM.
- Colocar un bloque “CLOSE” fuera del ciclo “While” en la parte derecha y conectar todas las terminales “Arduino Resource” y “error”.

Para el control del motor se utilizaron 2 salidas digitales y una PWM, las salidas digitales indicarán el sentido del motor, mientras que la salida PWM se encargará del control de la velocidad. El integrado L293D se alimenta de 5 volts por lo que para ponerlo en funcionamiento se ha conectado a la pequeña fuente de 5 volts que está incluida en la placa Arduino, este voltaje no es el mismo con el que se alimenta el motor, el actuador en cambio se alimenta del voltaje que se conecta al pin Vmotor, al cual se ha conectado 9 volts.

## Control

Un sistema de control está formado por subsistemas y procesos unidos con el fin de controlar las salidas de los procesos, en este caso la posición de un péndulo es controlada por el movimiento de un motor (figura 25).

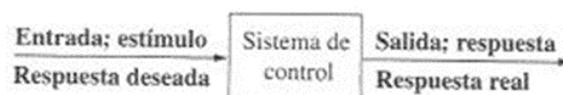


Figura 25 Descripción simplificada de un sistema de control.

Un sistema de control produce una salida o respuesta para una entrada o estímulo dado. La entrada representa una respuesta deseada mientras que la salida es la respuesta real. Dentro de los sistemas de control, éstos se pueden definir de dos tipos, los sistemas de lazo abierto y los de lazo cerrado.

Los sistemas de lazo abierto no pueden compensar ninguna perturbación y simplemente se comandan por la entrada, mientras que los sistemas en lazo cerrado reciben información en retroalimentación que permite corregir las perturbaciones, el objetivo del proyecto es que el sistema logre balancear la barra en todo momento, aunque se presenten perturbaciones, por lo que nuestro sistema se define como de lazo cerrado, con un elemento de retroalimentación visual.

### El control PID

Existen muchas técnicas de control que se pueden aplicar a múltiples plantas dependiendo del tipo de respuesta que se desee, pero el control PID tiene una ventaja, la cual es que no se necesita conocer el modelo matemático de la planta para aplicar el control, debido a esto, se ha optado por la aplicación de un control PID ya que no se contempla el modelo matemático de nuestro sistema.

El control Proporcional, Integral y Derivativa (PID) es la estructura de control más usada en el medio industrial (figura 26).

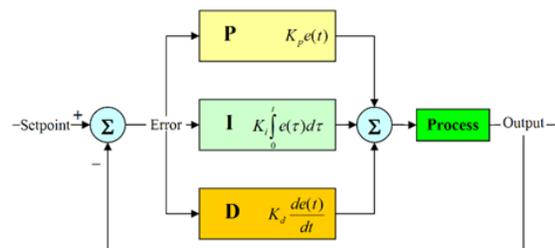


Figura 26 Estructura de control PID.

Ahora que se ha definido un sistema de control y un control PID, se puede definir nuestro sistema con el control PID implementado (figura 27).

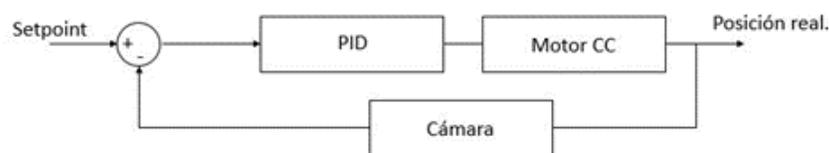


Figura 27 Sistema de control péndulo.

## **Asociación de los elementos**

Hasta este momento se han probado dos métodos para obtener información numérica de las propiedades del sistema (péndulo), pero mediante la técnica identificación de color, se ha notado mayor lentitud para obtener la imagen del dispositivo debido a las características del equipo con el que se cuenta, además de que debido a esta lentitud, el asistente de visión pierde la referencia del origen del sistema de coordenadas, por lo que detiene el programa e impide continuar momentáneamente su ejecución, en cambio la técnica identificación de bordes trabaja muy bien con la imagen de video y al usarla en este caso no presentó errores que detuvieran el programa, debido a esto se optó por continuar utilizando la técnica de identificación de bordes.

De igual forma se probó utilizando dos tipos de motores, el primero un motor a pasos bipolar, el cual es más difícil de trabajar debido a sus características y el segundo un motor CC mucho más simple. Para continuar, se ha elegido trabajar con el motor CC debido a sus características.

Antes de continuar se explica el procedimiento de toma y procesamiento de imágenes:

- La cámara obtiene la imagen del péndulo en el cual se han colocado 4 referencias.
- El asistente de visión cambiará la imagen de color a una en escala de grises, de donde hará 2 mediciones horizontales.
- La primera medición será desde una referencia fija colocada en el extremo izquierdo de la estructura hasta una referencia colocada en el extremo del péndulo, a dicha distancia la llamaremos d1.
- La segunda medición será desde una referencia fija colocada en el extremo izquierdo del péndulo, hasta una referencia colocada en la base del péndulo, a dicha distancia la llamaremos d2.
- d1 y d2 serán entregados por el asistente de visión para poder utilizarlos como datos de programación.
- Si la diferencia entre estas dos distancias ( $d1-d2$ ) es cero, entonces significa que el péndulo está en equilibrio, si por el contrario la diferencia entre las

distancias es diferente de cero, entonces significa que se ha provocado una perturbación en el sistema y el péndulo está a punto de caer.

- A la diferencia de distancias la llamaremos “x”, este dato es importante ya que mientras mayor sea el valor absoluto de x, el péndulo se encontrara más cerca de caer.
- El dato x será la variable de control, debido a ello esta variable se introducirá a un bloque PID dentro del mismo LABView, este bloque se encargara de el control del motor para que decida si este debe de ir más o menos rápido para tratar de equilibrar el péndulo.
- El motor mueve la cremallera y equilibra el péndulo en posición vertical.

Todos los pasos mencionados anteriormente, están en un bucle infinito, que se repite indefinidamente hasta que se detenga la ejecución del programa.

El asistente de visión entregará las dos distancias, como ya se había mencionado. Las distancias entregadas están en pixeles, al ser éstos de magnitud muy pequeña, al tener una mínima variación en la inclinación del péndulo las distancias variaban a pesar de que el péndulo se encontrara en equilibrio, para darle un poco más de margen a la inclinación del péndulo, las distancias se dividieron entre 5 y las cantidades se convirtieron a enteros para no considerar los decimales, luego de esto se ha realizado la diferencia, en la figura 28 se observa lo anteriormente mencionado.

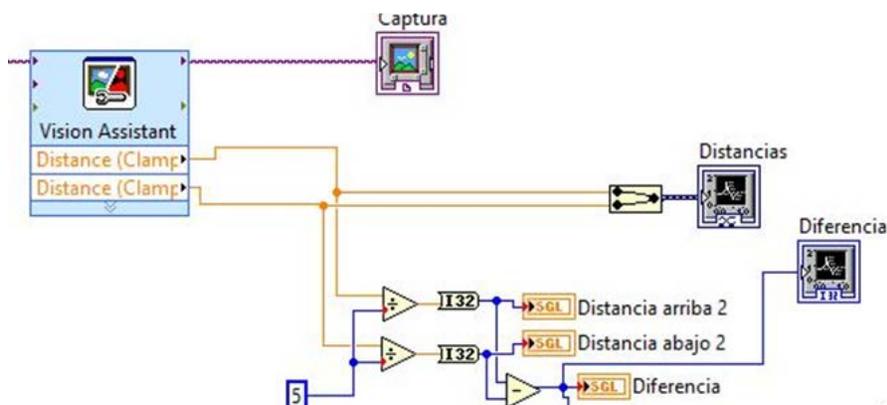


Figura 28 Ajuste del Sistema de visión.

Ahora la diferencia entre las distancias se ha cambiado para obtener su valor absoluto, ya que esta diferencia puede cambiar de signo dependiendo de hacia dónde esté cayendo el péndulo con respecto a la base, esta cantidad se cambiará de signo para obtener siempre valores negativos, debido a que el setpoint (valor deseado) es 0, es decir la distancia  $d_1$  y  $d_2$  deben ser iguales y que es más fácil trabajar con un PID cuando el setpoint es la cantidad más grande que puede alcanzar el sistema.

En el software LABView se coloca un bloque PID, al que se le crea un control para modificar las constantes ( $K_p, T_d$  y  $T_i$ ), también se colocan los límites inferior y superior, (0-255) de la señal de control que posteriormente serán dirigidos a una salida PWM de la placa Arduino (figura 29).

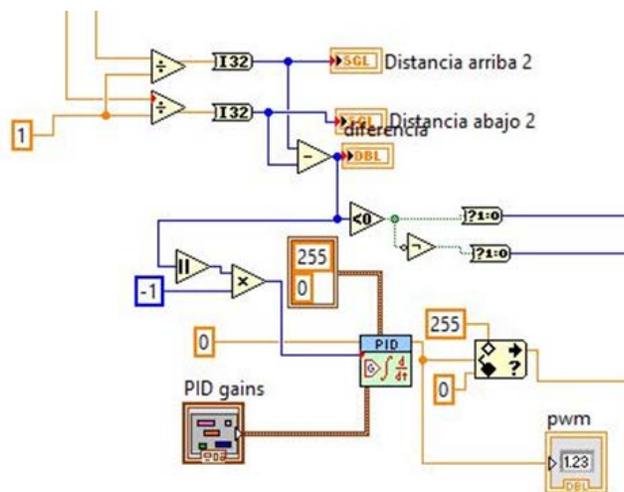


Figura 29 PID en software.

En el panel frontal se han colocado los controles para modificar los valores de las constantes del PID, un botón de paro, un indicador para la distancia  $d_1$ , otro para la distancia  $d_2$ , así como uno que indica que valor de PWM se escribe en la placa Arduino, también se muestran dos gráficas en la parte derecha, que muestran las distancias  $d_1$  y  $d_2$  a través del tiempo y también la diferencia entre éstas (figura 30).



Figura 30 Panel frontal del VI.

## 5. Resultados

Al realizar las pruebas se comprobó el correcto funcionamiento del control del péndulo invertido y aunque la velocidad de respuesta del mismo no fue tan rápida como la esperada, varios factores pudieron influir en este resultado como la velocidad del motor, la altura la barra o el lento procesamiento de la imagen, aun así y ante pequeñas perturbaciones, se logró observar como el sistema trataba de equilibrarse. Durante el desarrollo se cumplió con todos los objetivos secundarios propuestos.

En la figura 31 se observa la gráfica al momento de iniciar el programa, se nota un inicio abrupto donde el sistema se tambalea un poco al no tener datos aun de imagen, posteriormente se estabiliza.



Figura 31 Gráfica al momento de iniciar el programa.

En la figura 32 se observa como ante una perturbación pequeña (línea azul) el sistema se estabiliza después de unos momentos moviendo la base (línea roja).



Figura 32 Estabilización del sistema.

## 6. Resultados

A pesar de que el desarrollo de un dispositivo controlado por visión en este caso un péndulo invertido resultó más complejo de lo previsto, el objetivo principal de este proyecto se cumplió al poder realizar el control del péndulo invertido utilizando como sensor de retroalimentación una cámara para adquirir imágenes y procesarlas en un software. Se rescataron puntos notables como el hecho de poder monitorear variables así como no tener que montar sobre el dispositivo un sensor que añadiría peso y que tal vez no hubiera sido lo más conveniente para el sistema o proceso, sin embargo se puede concluir en base a los resultados obtenidos que es posible mejorar el desempeño del dispositivo siempre que el equipo de cómputo y el elemento actuador (en este caso un motor) sean los adecuados.

Otro punto de interés a mencionar es el hecho de que mediante técnicas de instrumentación fue posible manipular la información obtenida de cada imagen y procesarla para mandar señales que controlen la posición del péndulo con el propósito de que el péndulo sea capaz de mantenerse vertical en todo momento y responder de manera rápida ante las perturbaciones externas. Aunque se implementó y diseñó un control PI, PD y PID para permitir que el sistema fuera más rápido y estable, sin embargo, en este sentido la rapidez de respuesta no fue la que se esperaba. Esto presenta pues, un área de oportunidad de mejora para tener un dispositivo con mejor y más rápida capacidad de respuesta.

## **7. Bibliografía**

- [1] D. Domínguez Álvarez, R. Ulloa Tegoma, D. Valencia García, D. Zaragoza Fernández, Control de un motor por medio de PWM para procesos de velocidad, Tesis de grado en ingeniero en comunicaciones y electrónica, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México D.F, 2007.
- [2] N. S. Nise, Sistemas de control para ingeniería, 3ª Ed. 2006, Universidad Politécnica del estado de California.
- [3] K. Ogata, Ingeniería de control moderna, 5ª Ed., 2010, Pearson Educación S.A., Madrid.