



Benemérita Universidad Autónoma de Puebla



Facultad de Ciencias de la Electrónica

CONTROL DE POSICIÓN DE UN CILINDRO HIDRÁULICO CON INTERFAZ ENTRE MATLAB Y ARDUINO.

Mario Geovani Mendoza González, Elio Moisés González Ruiz

Facultad de Ciencias de la Electrónica de la BUAP

geo_12men@hotmail.com, big_2g@hotmail.com

REVISIÓN: 15/Mayo/2012

RESUMEN

Para el control de posición del cilindro hidráulico son necesarias 3 partes importantes para llevarse a cabo, la primera parte lo conforma la etapa de alimentación hidráulica, la segunda parte está conformada por la electrónica de potencia para el control de las electroválvulas, y una tercera parte dada por el control de la posición a lazo cerrado mediante el sensor de posición (Cámara WEB) y la comunicación entre MatLab y la tarjeta ARDUINO.

El control de salida y entrada el vástago del cilindro hidráulico será implementado a través de un control ON-OFF, la señal de error generada mandará un pulso mediante la ARDUINO a los transistores de potencia configurados en modo de switcheo para alimentar con 24VCD a la electroválvula 5 vías 3 posiciones, la electroválvula cambiará de la posición 2 a la 1 y el vástago del cilindro saldrá hasta que llegue al valor de referencia, análogamente para el retroceso del vástago cambiando la electroválvula de la posición 2 a la 3.

PALABRAS CLAVE

Arduino, Matlab, Guide, Cilindro Hidráulico, Control, Procesamiento Digital de Imágenes.

INTRODUCCION

El procesamiento digital de imágenes es un conjunto de técnicas que se aplican a las imágenes digitales para obtener información de ellas o mejorar su calidad. En todo el desarrollo de la computación hasta hace unas décadas era imposible de realizar de una manera más compacta pues se requerían tarjetas especializadas para ello además de sistemas operativos específicos. Al paso de los años éste proceso ha evolucionado rápidamente, con el crecimiento de todas las fases de tecnología de computadoras digitales, la rapidez de transmisión de

información, el bajo costo de elementos computacionales, así como la gran capacidad y alta densidad de dispositivos de memoria han hecho posible el procesamiento, manipulación y visualización de grandes cantidades de imágenes digitales. Todas estas ventajas que actualmente nos brinda la tecnología sobre el procesamiento digital de imágenes están enfocadas en áreas como la robótica, la automatización, la medicina, la astronomía y un sinnúmero de aplicaciones.

El posicionamiento de distintos actuadores como cilindros y motores es un proceso utilizado muy a menudo en distintas aplicaciones, principalmente en

maquinarias para maquila de productos tal es el caso de máquinas como fresadoras, torneadoras, curvadoras, troqueladoras, etc.

SISTEMA FÍSICO

La planta cuenta con un cilindro hidráulico de doble efecto en el cual tenemos que regular la posición lineal del vástago, ya sea de salida o retroceso. El cilindro de doble efecto representa uno de los actuadores más comunes que se utilizan en máquinas donde se requiere gran fuerza de salida debido a la alta presión generada por líquidos. La fuerza de salida del vástago es directamente proporcional a la presión generada por la bomba hidráulica y el área del cilindro [1].

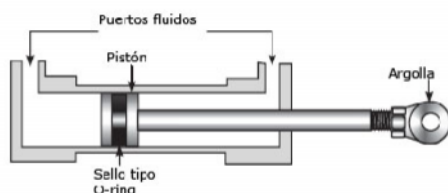


Figura 1: Cilindro Hidráulico de Doble Efecto.

Para el control de salida y retroceso del vástago del cilindro el sistema cuenta con una electroválvula de 4 vías 3 posiciones con retroalimentación a tanque como se muestra en la siguiente imagen.

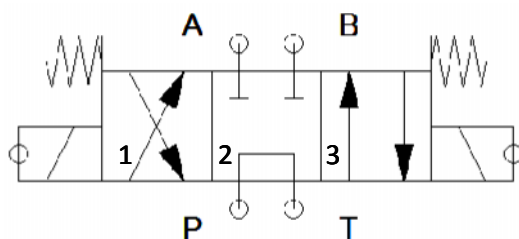


Figura 2: Electroválvula hidráulica 4/3.

Donde:

A,B=Salidas al cilindro.

P=Presión.

T=Tanque.

Se observa en la electroválvula los solenoides que accionan el cambio de posición cuando son alimentados con 24VCD.

Como posición de reposo, la electroválvula se encuentra en la 2 y la presión generada por la bomba hidráulica se va a tanque, mientras que si se energiza el solenoide de la derecha, habrá un

cambio de posición de 2-3, se conectarán P-A y T-B, el vástago saldrá mientras el solenoide se encuentre energizado, análogamente si se energiza el solenoide de la izquierda, el vástago retrocederá a su posición inicial mientras ésta se encuentre energizado.

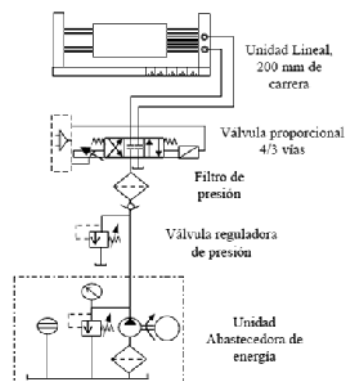


Figura 3: Sistema Hidráulico.

SISTEMA ELECTRÓNICO

Etapa de Potencia

Ya se ha mencionado anteriormente cómo se controla el cilindro de doble efecto, para ello es necesario acoplar los voltajes 5VCD de la tarjeta ARDUINO a 24 VCD de alimentación a la electroválvula. Los transistores de potencia BJT configurados en modo de switcheo o modo ON-OFF realizarán ésta tarea como se muestra a continuación [2]:

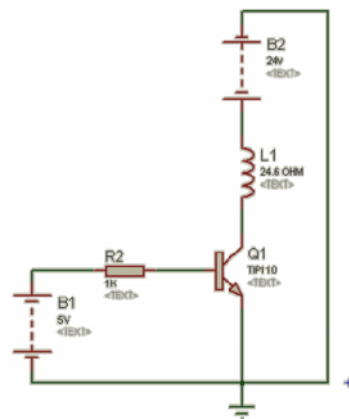


Figura 4: Etapa de potencia con un transistor Darlington TIP110.

INTERFAZ DE CONTROL

La señal de control mostrada en la figura anterior que activa y desactiva los transistores BJT es generada por el programa de control en MatLab y transmitida a través de la tarjeta de adquisición de datos ARDUINO. Dicha tarjeta consiste en una placa con un microcontrolador Atmel AVR y puertos de entradas analógicas/digitales y salidas digitales. La ARDUINO es utilizada como interfaz para la comunicación entre el Software y Hardware y cuenta con las siguientes especificaciones [3]:

- 14 Entradas digitales configurables i/o salidas que operan a 5V.
- Cada PIN puede proporcionar o recibir como máximo 40mA.
- Los pines 3, 5, 6, 8, 10 y 11 pueden proporcionar una salida PWM.
- 6 Entradas analógicas que proporcionan una resolución de 10 bits.

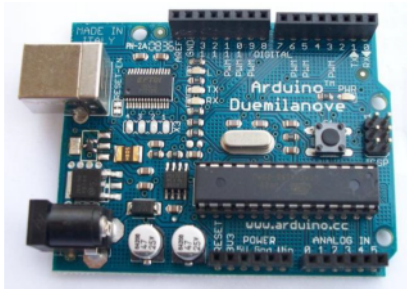


Figura 5: Tarjeta de Adquisición de Datos ARDUINO.

CAMARA WEB

Un elemento importante para calcular el error de nuestro sistema a lazo cerrado, es el sensor de retroalimentación, este caso se utilizará una cámara externa ActiveX. Ésta es una pequeña cámara digital que puede capturar imágenes y procesarlas de la manera que mejor nos convenga. Este se clasifica como dispositivo de entrada, ya que podemos transmitir imágenes hacia la computadora y cuenta con ciertas características importantes [4]:

- Resolución VGA 640X480.
- Código de colores RGB.
- Sensor CMOS integrado.

SOFTWARE (MATLAB)

El control de posición del cilindro de doble efecto será implementado en MatLab que es un Software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación M.

MatLab cuenta con distintas prestaciones básicas como: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuarios (GUI), y la comunicación con programas de otros lenguajes y con otros dispositivos hardware, además de que cuenta con 2 herramientas adicionales como Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario-GUI) [5].

PROGRAMACION

El usuario como se ha mencionado se encargará de posicionar el slider según la referencia deseada, el vástago del cilindro de doble efecto seguirá tal referencia, para ello hará uso del entorno GUIDE que es una prestación mediante la cual el usuario interactúa con el programa ya que contiene diferentes elementos gráficos tales como: botones, campos de texto, menús, gráficos, que le permiten manejarlo de una manera más amigable y fácil.

Una breve revisión para el control y la interfaz gráfica en Matlab se muestra a continuación [6]:

```
% Desconectar y borrar objetos de
adquisición de imágenes
imagreset

% Crear el objeto de video si la
cámara está conectada
try
handles.vid = videoinput('winvideo',
1,'YUY2_320x240');
catch % No continuar si no hay cámara
conectada
warndlg({'NO SE ENCUENTRA CÁMARA
CONECTADA'...
; 'POR FAVOR CONECTE
UNA WEBCAM'}, 'AVISO WEBCAM')
return
end
% Velocidad de adquisición de la trama de
video
handles.vid.FrameGrabInterval = 5;
% Número de ejecuciones de disparo de
adquisición
handles.vid.TriggerRepeat=Inf;
% Set value of a video source object
property.
sources = handles.vid.Source;
selectedsrc =
getselectedsource(handles.vid);
set(selectedsrc,'Tag','Intellicam setup');
get(selectedsrc);
%*****Iniciación del Serial
para ARDUINO*****
s=serial('COM7');
set(s,'BaudRate',9600);
set(s,'DataBits',8);
set(s,'Parity','none');
set(s,'StopBits',1);
set(s,'OutputBufferSize',1);
set(s,'InputBufferSize',1);
set(s,'timeout',1);
```

```

set(s,'FlowControl','none');
handles.ser=s;
guidata(hObject,handles)

% --- Executes on slider movement.
function nivel_Callback(hObject, eventdata, handles)
pos=get(handles.nivel,'Value');
set(handles.posicion,'String',pos);
% --- Executes on button press in INICIO_B.
function INICIO_B_Callback(hObject, eventdata, handles)
%Abre el puerto para la comunicación serial
s1=handles.ser;
fopen(s1);
%Deshabilitar el botón de Inicio
set(handles.INICIO_B,'Enable','off')
% Colocar a 0 el flag botón de parada
set(handles.PARAR_B,'UserData',0)
% Iniciar objeto de video
start(handles.vid);
%Ayuda a la camara ajustar las condiciones de iluminación
pause(2);
% Tamaño de la imagen: 352 x 288
ancho=320;
alto =240;
% Límites donde buscare el pixel más luminoso
lim_f1=ceil(alto/3);
lim_f2=ceil(alto-1);
lim_c1=ceil((ancho/3)-25);
lim_c2=ceil((ancho/1.3)+24);
limites=[lim_f1,lim_f2,lim_c1,lim_c2];

while (handles.vid.FramesAcquired<=1000)
% Si se presiona el botón PARAR detener el ciclo WHILE
if get(handles.PARAR_B,'UserData')
stop(handles.vid);
break
end
% Manda a mostrar el video de la camara
imgn =
ycbcr2rgb(getdata(handles.vid,1,'uint8'));
axes(handles.axes1);
image(imgn);
axis off
% Detección y cálculo de distancias
[X distancia]=detectar3(imgn, limites);
set(handles.x_coord,'String',X);
set(handles.distancia,'String',distancia);
% Control de posición
pos=get(handles.nivel,'Value');
referencia=pos;
histeresis=5;
error=referencia-distancia;
% Si el pistón sobrepasó la referencia deseada, se activa la señal de retroceso
if (error>histeresis)
fwrite(s1,'1');
% Si el pistón aún no alcanza referencia, se envía la señal de avance
elseif (error<-histeresis)
fwrite(s1,'2');
% Si se encuentra en la zona de histéresis, el sistema se mantiene en reposo

```

```

elseif (error<histeresis | error>-
histeresis)
fwrite(s1,'0');
end
end
handles.ser1=s1;
guidata(hObject,handles)
guidata(hObject, handles);
% --- Executes on button press in PARAR_B.
function PARAR_B_Callback(hObject, eventdata, handles)
% hObject handle to PARAR_B (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.PARAR_B,'UserData',1)
set(handles.INICIO_B,'Enable','on')
s1=handles.ser1;
fclose(s1);
delete(s1);
clear s1;
guidata(hObject,handles)

```

El programa arriba descrito llama a la función detectar 3, para marcar la posición del láser y medir la distancia a partir de nuestro "0". Este es el código utilizado en la función.

```

function [X distancia]=detectar3(I,lim)
hsv = rgb2hsv(I);
v = hsv(:, :, 3);
% max_v =
max(max(v(lim(1):lim(2),lim(3):lim(4)))));
max_v=0.95;
% busca el punto más luminoso dentro de los limites y guarda su ubicación
% en la matriz coord
coord=[];
for f=lim(1):lim(2)
for c=lim(3):lim(4)
if v(f,c)>=max_v
%v(f,c)==max_v
coord=[coord;[f,c]];
end
end
end

if isempty(coord)
X='';
Y='';
distancia='';
return
end
Y=round((min(coord(:,1))+max(coord(:,1)))/2);
X=round((min(coord(:,2))+max(coord(:,2)))/2);
%establecemos mediante calibración nuestro punto de referencia para medir la istancia
X1=105;
%dibujar los lineas de ubicación
hold on
plot(X,Y,'go','markerSize',10)
line([X X],[288 1],'Color','r')
line([352 1],[Y Y],'Color','r')
hold off
%la función regresa la distancia medida
distancia=round(abs(X1-X))*1.37;end

```

También se incluye el programa desarrollado para Arduino quien se comunica a través de puerto serial con Matlab y es el encargado de enviar la señal de control que excitará a los transistores y accionarán los solenoide [7].

```
int incomingByte = 0; // para el byte
leído

void setup() {
    Serial.begin(9600); // abre el
    puerto serie a 9600 bps
    //configure los pines digitales como
    salidas
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
}

void loop() {
    //Espera a que se reciba un dato
    if (Serial.available() > 0) {
        // lee el byte entrante:
        incomingByte = Serial.read();

        //Interpreta las señales de control y las
        ejecuta a través de los pines 8 y 9
        if (incomingByte == '1') {
            digitalWrite(8, LOW);
            digitalWrite(9, LOW);
            delay(10);
            digitalWrite(8, HIGH);
        }
        if (incomingByte == '2') {
            digitalWrite(8, LOW);
            digitalWrite(9, LOW);
            delay(10);
            digitalWrite(9, HIGH);
        }
        if (incomingByte == '0') {
            digitalWrite(8, LOW);
            digitalWrite(9, LOW);
        }
    }
}
```

RESULTADOS

Una vez programado las funciones necesarias y el GUIDE generado en Matlab, además de la tarjeta ARDUINO, para el control del vástago del cilindro de doble efecto, se obtuvieron los siguientes resultados.

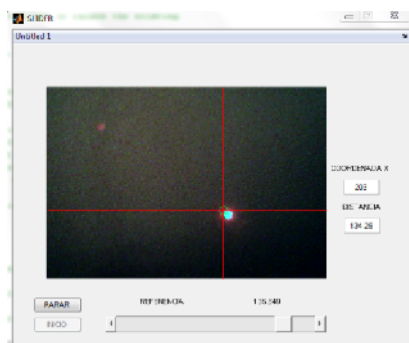


Figura 6: Interfaz gráfica GUI, detectando el punto más luminoso



Figura 7: Sistema implementado, cámara WEB, laser y sistema hidráulico

CONCLUSIONES

El procesamiento digital de imágenes es una herramienta muy poderosa, y de gran utilidad en la ingeniería. Las aplicaciones son bastas y en todos los ámbitos. En este artículo se muestra su uso en el control de actuadores como en este caso un cilindro hidráulico que podría ser de gran utilidad en la industria. La interacción entre usuario y proceso es muy importante, además la comunicación entre software y hardware, objetivos que se cumplieron exitosamente.

AGRADECIMIENTOS

Un agradecimiento especial para el laboratorio de Mecatrónica de la Facultad de Ciencias de la Electrónica por el equipo proporcionado para realizar este proyecto.

REFERENCIAS

- [1]<http://gemini.udistrital.edu.co/comunidad/profesores/drairan/documents/papers/fuzzytec19.pdf>.
- [2]Adrel S. Sedra, "Circuito Microelectrónicos", Quinta Edición, Mc. Graw Hill, 5ta. Edición.
- [3]<http://es.wikipedia.org/wiki/Arduino>.
- [4]http://es.wikipedia.org/wiki/C%C3%A1mara_web.
- [5] <http://es.wikipedia.org/wiki/MATLAB>.
- [6]<http://www.fimee.ugto.mx/profesores/rguzman/documentos/pie-guide.pdf>.
- [7] <http://www.arduino.cc/es/>