



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

DISEÑO Y CONSTRUCCION DE UN SISTEMA DE SERVO PARA EL LABORATORIO DE CONTROL AUTOMATICO

TESIS DE GRADUACION

Previa la obtención del Título de

**INGENIERO EN ELECTRICIDAD
ESPECIALIZACION ELECTRONICA**

Presentada por:

**ALFREDO XAVIER REINA LOGROÑO
JESUS VICENTE PARRA ZAMORA
LAURO NORBERTO VARGAS LOPEZ**

GUAYAQUIL - ECUADOR

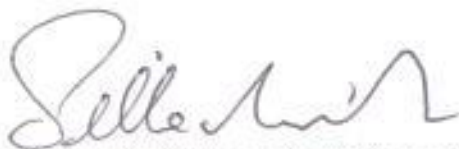
AÑO

1998



ING. ARMANDO ALTAMIRANO

Presidente del Tribunal



ING. HUGO VILLAVICENCIO

Director de Tesis



ING. NORMAN CHOOTONG

Miembro del Tribunal



ING. NELSON LAYEDRA

Miembro del Tribunal

DEDICATORIA

AL PADRE CELESTIAL

A NUESTROS PADRES

A NUESTROS HERMANOS

AGRADECIMIENTO

Al Ing. Hugo Villavicencio V., por su apoyo y guía en la culminación de esta obra.

Y a todos aquellos que colaboraron incondicionalmente en el éxito de este trabajo de tesis.

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(Reglamento de Exámenes y Títulos profesionales de la ESPOL).

.....
ALFREDO XAVIER REINA LOGROÑO

.....
LAURO NORBERTO VARGAS LOPEZ

.....
JESUS VICENTE PARRA ZAMORA

RESUMEN

El propósito de esta tesis es el de dar a conocer el diseño y construcción de un servomecanismo para controlar la velocidad y posición de un motor DC utilizando circuitos analógicos y diseño de un controlador digital basado en un microcomputador.

En el capítulo 1 se mencionan los objetivos propuestos y una breve descripción del funcionamiento de nuestro sistema, el cual se ha dividido en dos partes: parte analógica y parte digital. En la parte analógica se analizan los diferentes circuitos de control y amplificación. En la parte digital se hace mención del diseño de la interface de adquisición de datos y del controlador digital.

En el capítulo 2 se analiza las diferentes fuentes de poder requeridas para nuestros circuitos, como son una fuente de poder no regulada de ± 32 V para el motor, una fuente de ± 15 V, una fuente de 8 V y una fuente de 5 V.

En el capítulo 3, se analiza las etapas amplificadoras, como son un amplificador sumador inversor, una etapa pre-amplificadora y un servo-amplificador los cuales sirven para controlar el motor.

En el capítulo 4 se analiza las características de los 3 diferentes tipos de motores utilizados en el sistema , actuando uno de ellos como motor DC, otro como tacho-generador y el último como generador. Este generador manejará carga, la cual servirá para analizar la respuesta transitoria del motor.

En el capítulo 5 se analiza la diferencia entre sistemas de lazo abierto y sistemas de lazo cerrado; luego se hace un estudio de los diferentes modos de control como son el *control proporcional, proporcional - integral, proporcional - derivativo, proporcional integral - derivativo.*

En el capítulo 6 se analiza la estabilidad del sistema . Para ello se procederá a realizar los cálculos de estabilidad del sistema utilizando el método del lugar geométrico de las raíces.

En el capítulo 7 se hace el estudio del diseño de la tarjeta de interfaz de datos utilizando puertos de entrada y salida y convertidores analógico-digital y digital-analógico.

En el capítulo 8 se explica el funcionamiento del sistema de control digital de velocidad detallando el recorrido de la señal a través de cada uno de los circuitos del controlador

digital. También se analiza la respuesta transiente , en el cual se comprobará el correcto funcionamiento del sistema digital.

En el capítulo 9 se explica el funcionamiento del control digital de posición del sistema y el manejo de la señal de posición a través del controlador. Se hace mención de las características de funcionamiento y construcción del codificador de eje del motor.

En el capítulo 10 se muestra el diseño de la circuitería (hardware) de todos los bloques componentes del sistema de control digital y analógico ; así como también se detallan las características de diseño del sistema.

En el capítulo 11 se explica el funcionamiento del programa haciendo uso de los diagramas de flujo y algoritmos utilizados ya sea para el control de velocidad como para el control de posición.

En el capítulo 12 se detallan específicamente el diseño de cada uno de los medidores digitales utilizados en nuestro sistema.

En la parte final se anexa cierta información técnica de los circuitos integrados. Se expone además el manual de usuario (digital y analógico) del sistema de servos con

los diferentes experimentos didácticos que realizarán los estudiantes en el Laboratorio de Control Automático. Se concluye con el listado completo del programa .

INDICE GENERAL

	Pág.
RESUMEN	6
INDICE GENERAL	10
INDICE DE FIGURAS	16
INDICE DE TABLAS	28
INTRODUCCION	30
PARTE ANALOGICA	
CAPITULO I	
OBJETIVOS Y DESCRIPCION GENERAL DEL SISTEMA	32
1.1. OBJETIVOS	32
1.2. DESCRIPCION GENERAL DEL SISTEMA	33
CAPITULO II	
FUENTES DE ALIMENTACION	37
2.1. FUENTE DE PODER NO REGULADA	37
2.2. FUENTE DE PODER REGULADA BIPOLAR	38
2.3. FUENTES ADICIONALES	41
CAPITULO III	
AMPLIFICADORES DE SEÑAL	42

	Pág.
3.1. ETAPA PREAMPLIFICADORA	42
3.2. ETAPA AMPLIFICADORA	45
3.3. SERVOAMPLIFICADOR	46
CAPITULO IV	
SERVOS DC	48
4.1. GENERADOR: CARACTERISTICAS	48
4.2. TACOGENERADOR: CARACTERISTICAS	50
4.3. MOTOR: CARACTERISTICAS	51
4.3.1. CONTROL DE CAMPO	52
4.3.2. CONTROL DE ARMADURA	55
4.4. MEDICION DE LAS CARACTERISTICAS DEL MOTOR	59
4.5. RESPUESTA TRANSIENTE DEL MOTOR	60
4.6. CONTROL DE VELOCIDAD DEL MOTOR	62
4.7. CONTROL DE POSICION DEL MOTOR	65
CAPITULO V	
SISTEMAS REALIMENTADOS	69
5.1. SISTEMA DE LAZO ABIERTO VS. SISTEMA DE LAZO CERRADO	69
5.2. MODOS DE CONTROL EN SISTEMAS DE LAZO CERRADO	72
5.2.1. CONTROL PROPORCIONAL	73

	Pág.
5.2.2. CONTROL PROPORCIONAL-INTEGRAL	77
5.2.3. CONTROL PROPORCIONAL-DERIVATIVO	82
5.2.4. CONTROL PROPORCIONAL - INTEGRAL - DERIVATIVO	86
CAPITULO VI	
ANALISIS DE ESTABILIDAD DEL SISTEMA	90
6.1. METODOS DE ANALISIS DE ESTABILIDAD	90
6.2. ESTABILIDAD DEL SISTEMA	92
6.3. CONDICIONES EXTREMAS DE FUNCIONAMIENTO	98
PARTE DIGITAL	
CAPITULO VII	
DISEÑO DE LA TARJETA DE INTERFAZ DE DATOS	100
7.1. TRANSMISION Y ADQUISICION DE DATOS	100
7.1.1. INTERFACE PROGRAMABLE PPI 8255	101
7.1.2. PUERTO DE ENTRADA/SALIDA 8212	106
7.1.3. DECODIFICADOR DE DIRECCIONES	110
7.2. CONVERSION DE DATOS	117
7.2.1. CONVERTIDOR DIGITAL - ANALOGICO DAC 0808	118
7.2.2. CONVERTIDOR ANALOGICO - DIGITAL ADC 0808	119
CAPITULO VIII	
CONTROL DIGITAL DE VELOCIDAD DE UN MOTOR DC	122

	Pág.
8.1. OBJETIVOS	123
8.2. DIAGRAMA DE BLOQUES	123
8.3. FUNCIONAMIENTO DEL SISTEMA	125
8.3.1. INTERFACE ACONDICIONADORA DE ENTRADA	125
8.3.2. INTERFACE DE ENTRADA/SALIDA CON LA MICROCOMPUTADORA	126
8.3.3. INTERFACE ACONDICIONADORA DE SALIDA	131
8.4. ANALISIS DE RESPUESTA DEL SISTEMA	132
CAPITULO IX	
CONTROL DIGITAL DE POSICION DEL SISTEMA	149
9.1. OBJETIVOS	149
9.2. DIAGRAMA DE BLOQUES	149
9.3. CODIFICADOR DE EJE DEL MOTOR	151
9.4. FUNCIONAMIENTO DEL SISTEMA	158
CAPITULO X	
DETALLES DEL DISEÑO	175
CAPITULO XI	
IMPLEMENTACION DEL PROGRAMA DE CONTROL DIGITAL	200
11.1. DESCRIPCION DE LENGUAJE ENSAMBLADOR	200

	Pág.
11.2. DIAGRAMA DE FLUJO DE PROGRAMA PRINCIPAL	205
11.3. IMPLEMENTACION DE LAS MACROS	210
11.4. IMPLEMENTACION DEL ALGORITMO DE <i>MUESTREO</i>	226
11.5. IMPLEMENTACION DEL ALGORITMO DE CONTROL DE VELOCIDAD	230
11.5.1. ALGORITMO DE CONTROL <i>P</i>	239
11.5.2. ALGORITMO DE CONTROL <i>PI</i>	241
11.5.3. ALGORITMO DE CONTROL <i>PD</i>	244
11.5.4. ALGORITMO DE CONTROL <i>PID</i>	247
11.6. IMPLEMENTACION DEL ALGORITMO DE CONTROL DE POSICION	250
11.6.1. ALGORITMO <i>RECTIFICA</i>	253
CAPITULO XII	
MEDIDORES DIGITALES	258
12.1. TACOMETRO DIGITAL	258
12.1.1. DISEÑO Y CONSTRUCCION	259
12.2. AMPERIMETRO DIGITAL	261
12.2.1. DISEÑO Y CONSTRUCCION	262
CONCLUSIONES	265

INTRODUCCION

Todo proceso en la naturaleza se rige bajo un modelo de control. El hombre siempre ha buscado optimizar estos procesos y controlar parte del medio que lo rodea, usualmente denominado sistema, con la finalidad de proporcionar un producto económico y útil en beneficio de la sociedad.

El presente trabajo ilustra el diseño y construcción de un sistema de control utilizando métodos analógicos y digitales. Nuestro sistema está conformado en su estructura analógica por un circuito sumador, amplificador y uno de fuerza el cual se encarga de manejar el motor DC. La estructura digital está conformado por una tarjeta de interfaz de datos y un circuito de fuerza encargado de operar el motor de acuerdo a las órdenes enviadas por el microcomputador a través de la tarjeta controladora de datos.

Primeramente para el control analógico se presenta un estudio teórico y práctico del funcionamiento de los componentes del sistema. Para el control digital se presenta un análisis del diseño y construcción de la tarjeta de interfaz de datos. El control digital además incluye una algoritmo de control PID el cual es implementado en un programa escrito en lenguaje ensamblador. Mediante el programa se permite el ingreso de variables

de control de una manera fácil para el usuario. Todo el análisis y estudio se ha implementado en un banco de pruebas el cual representa un equipo didáctico para el Laboratorio de Control Automático.

CAPITULO I

OBJETIVOS Y DESCRIPCION GENERAL DEL SISTEMA

1.1. OBJETIVOS

Debido a la necesidad de equipos didácticos para complementar con experimentos prácticos las clases teóricas en el área de control automático, se diseñará un sistema de servos tomando en cuenta criterios didácticos para la enseñanza de los diferentes sistemas tales como: control de velocidad y control de posición e ilustrando las diferentes acciones de algoritmos generalizados PID. Se diseñará las interfaces necesarias para ejercer acciones de control mediante el computador. Todos los circuitos estarán ensamblados en un banco experimental en el laboratorio de control automático que será utilizado por los estudiantes de tal manera que lleguen a una comprobación de la teoría de control y que tengan una cierta

experiencia con la práctica.

1.2. DESCRIPCION GENERAL DEL SISTEMA

Un sistema de lazo cerrado genera una señal de error la cual opera el sistema. En la figura 1.1 se muestra un diagrama de bloques de un **sistema de lazo cerrado**.

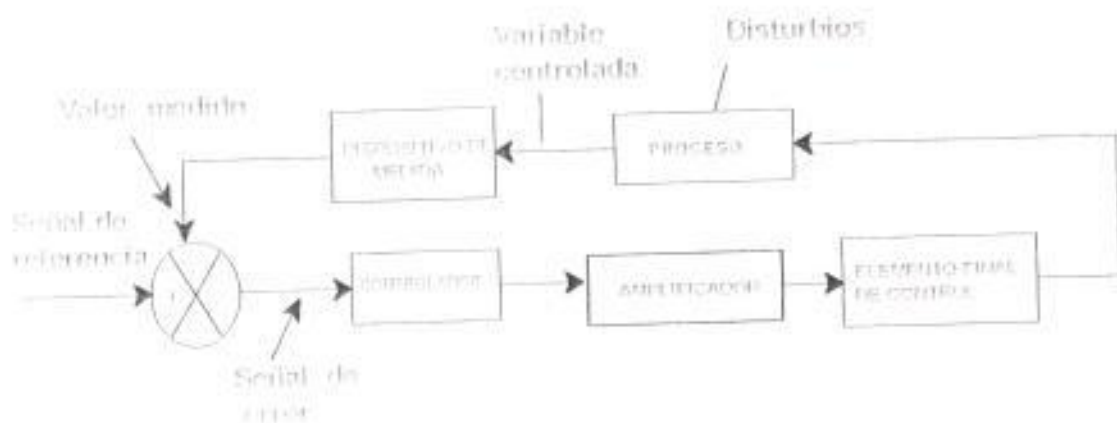


Figura 1.1 Diagrama de bloques de un sistema de lazo cerrado

En primer lugar una cierta variable de proceso que está siendo controlada (velocidad mecánica o posición mecánica) es medida y enviada a un comparador que en nuestro caso es un circuito sumador analógico implementado con opamps. Este sumador hace la comparación entre el valor de referencia, el cual representa el valor deseado de la variable, y el valor medido de la variable controlada. El comparador genera entonces una señal de error la cual representa la diferencia entre el valor deseado y el

valor medido. El controlador recibe la señal de error y genera un señal de salida. La relación existente entre la señal de salida del controlador y la señal de error depende del diseño y ajuste del controlador. Si se tiene un controlador proporcional el sistema opera siempre con una desviación o sea una señal de error apreciable.

Cuando se dispone de un controlador proporcional integral, en el cual la parte proporcional controla el sistema en proporción al error que exista en ese momento, la parte de control integral se da cuenta que existe un error (desbalance), por lo tanto a medida que pasa el tiempo, el control integral ayuda a reducir el desbalance, con lo que eventualmente el error se reduciría a cero.

Por otro lado cuando el control es proporcional derivativo, un rápido cambio en el error produce una acción correctora mayor que un cambio lento en el error; o sea la parte derivativa del controlador intenta mirar adelante y prevee que el proceso sufrirá un gran cambio basándose en las medidas actuales.

Finalmente un controlador PID resulta de la combinación de las 3 anteriores. En la figura 1.1 se muestra que la salida del controlador es enviada al elemento final de control, puede ser necesaria una amplificación si la señal de salida del controlador no tiene la suficiente potencia para manejar el dispositivo final de control que en nuestro caso es un motor DC. El dispositivo de medida es un tacómetro que genera una señal

de voltaje proporcional a la velocidad del motor, lo cual nos da el valor medido. En cuanto al sistema de control digital la señal de error es generada en el computador mediante la sustracción del valor deseado ingresado por el teclado y el valor actual de medición o variable del proceso(velocidad o posición). En la figura 1.2 se muestra el diagrama de bloques para este sistema.

El microcomputador mide la velocidad actual del motor (esta velocidad es suministrada por el tacogenerador cuando se trata de un control de velocidad) o mide la posición actual del motor la cual es suministrada por el codificador de eje.

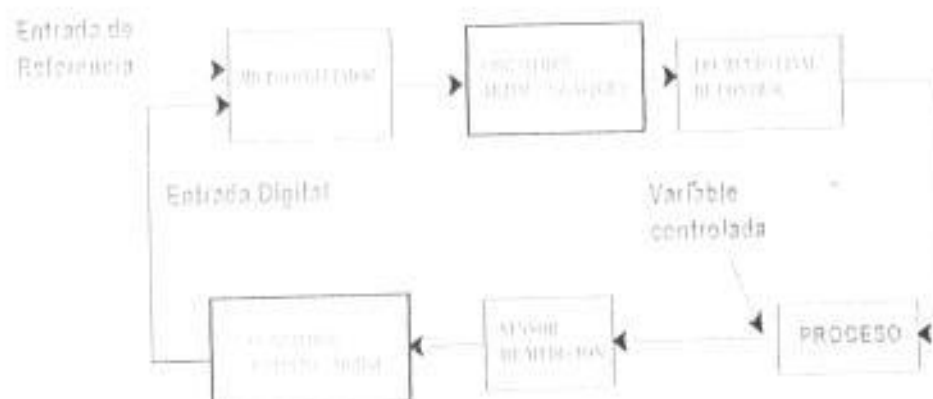


Figura 1.2 Diagrama de bloques de un sistema de control digital

cuando se trata de un control de posición. Luego compara la posición o velocidad actual con la posición o velocidad deseada produciendo un error. Con el error

producido se genera la señal de control de acuerdo al algoritmo aplicado por el microcomputador.

Esta señal de control es procesada por un circuito convertidor digital - analógico que luego es amplificada para operar el elemento final de control (motor DC). El sensor de medición mide la señal de salida la cual es procesada por un convertidor analógico - digital para ser enviada al microcomputador.

CAPITULO II

FUENTES DE ALIMENTACION

2.1 FUENTE DE PODER NO REGULADA

La mayor parte de los dispositivos electrónicos requieren voltajes de para poder operar. La fuente de alimentación disponible más fácil de obtener es de la red de 110V ac a 60 Hz. El circuito que convierte este voltaje ac en un voltaje de se denomina fuente de alimentación de,

Se requiere un transformador para reducir el voltaje de suministro que puede ser 115V AC a un voltaje más bajo. Los voltajes del transformador se dan en términos de valores *rms*. En la fig.2.1 se muestra el transformador utilizado en nuestro diseño. El transformador está especificado como de 115 V a 40 V. Con 115 V rms

conectado al primario, se desarrolla 40 V rms en el secundario, en nuestro caso tenemos un transformador con derivación central. El máximo voltaje instantáneo E_m está relacionado con el valor E_{rms} por: $E_m = 1.4 E_{rms}$

El siguiente paso consiste en convertir el voltaje más bajo de ca en corriente pulsante dc. Esto se consigue mediante la rectificación con diodos de silicio o germanio, en la fig. 2.1 se muestra esta rectificación. El voltaje dc pulsante, no es de pura, de modo que se coloca un capacitor de filtrado entre los terminales de salida de dc. Este capacitor suaviza los pulsos de salida y produce un voltaje dc de salida casi puro V_L . V_L es el voltaje no regulado que suministra potencia a la carga, en nuestro caso hacia el motor el cual opera con un voltaje de 27 V dc.

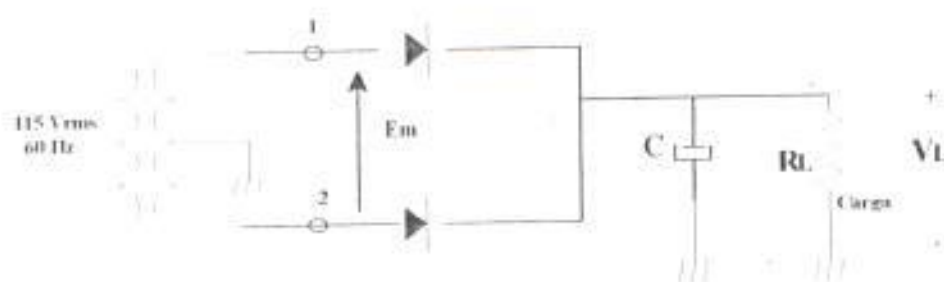


Figura 2.1 Fuente de Poder No Regulada

2.2 FUENTE DE PODER REGULADA BIPOLAR

La regulación de voltaje es necesario para evitar que el voltaje dc no regulado

disminuya conforme aumenta la corriente de carga y también para evitar que el voltaje de rizado aumente. Si la terminal central de la fuente de alimentación se pone a tierra en la figura 2.2, se tiene una fuente de alimentación bipolar no regulada.

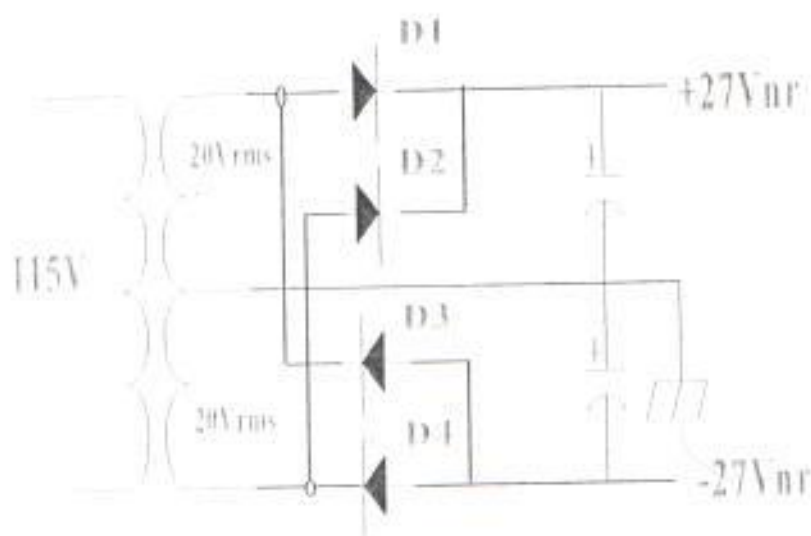


Figura 2.2 Fuente de Poder Bipolar No Regulada

Las desventajas de la fuente de poder no regulada se pueden minimizar al añadir una sección de regulación de voltaje como se muestra en la figura 2.3. La fuente de alimentación resultante se clasifica como fuente de voltaje o fuente de poder regulada, que para el presente estudio tiene una fuente de ± 15 V. En la figura 2.4 se muestra un circuito de regulación de +15V.

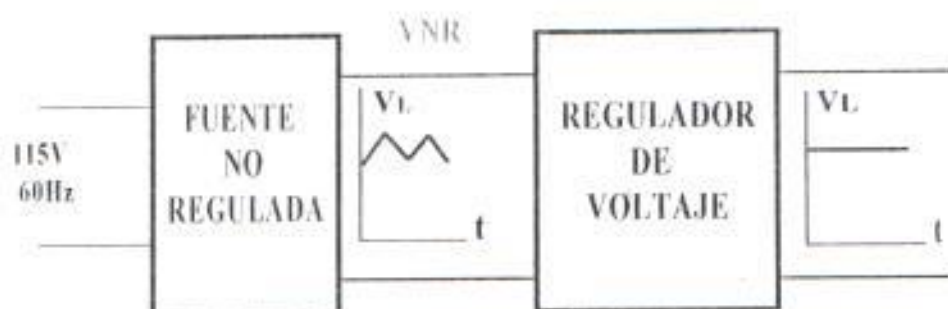


Figura 2.3 Fuente de Voltaje Regulada

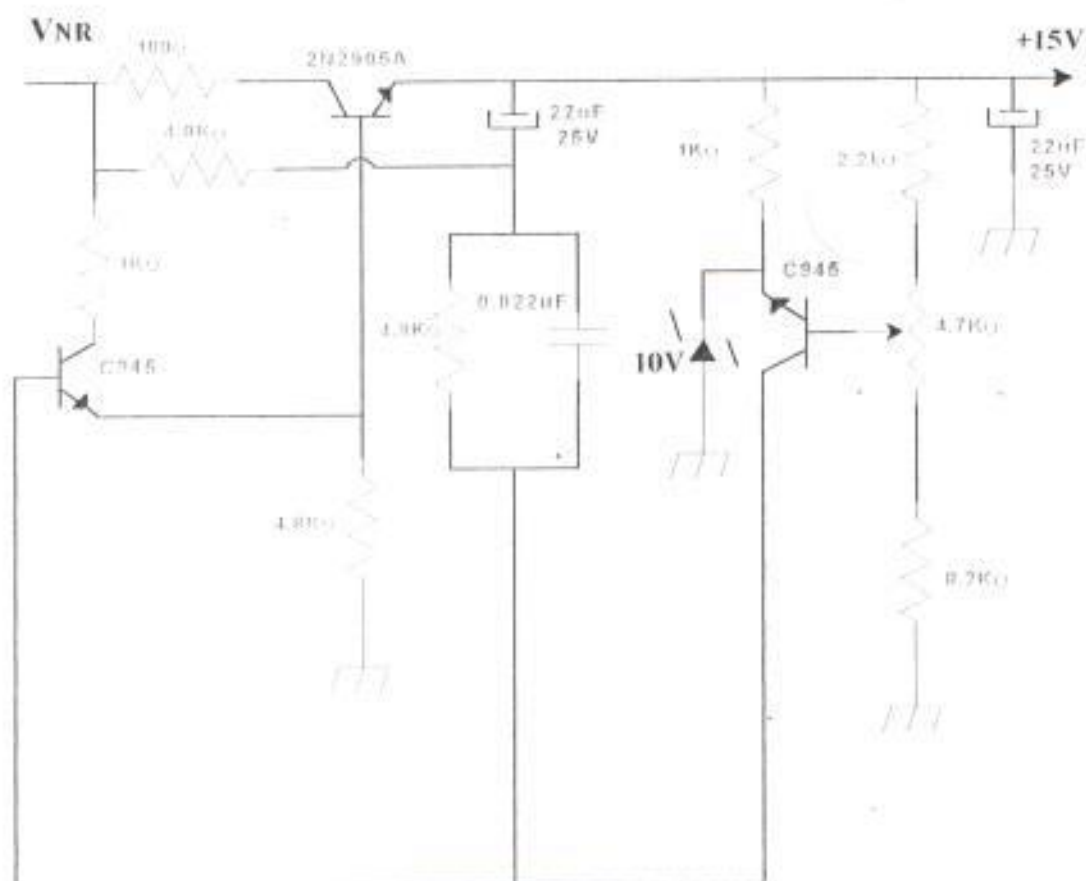


Figura 2.4 Etapa de Regulación positiva

2.3 FUENTES ADICIONALES

Las fuentes adicionales requeridas para nuestro proyecto son una de 8V (para el tacómetro) y otra de +5V (para el amperímetro). Estas fuentes están diseñadas con reguladores de voltaje tipo circuito integrado como son el LM7805 y el LM7808. En la fig.2. 5 se muestra estas fuentes:

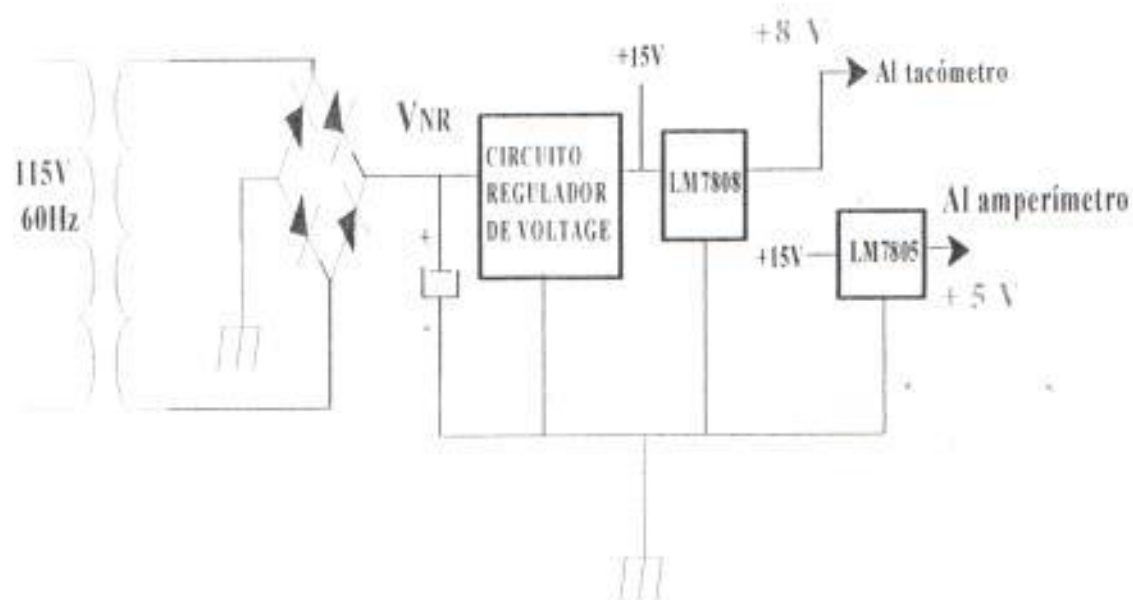


Figura 2. 5 Fuentes Adicionales de +8 V y +5 V

CAPITULO III

AMPLIFICADORES DE SEÑAL

3.1 ETAPA PREAMPLIFICADORA

Un sistema amplificador consiste en un transductor de entrada en el cual ingresa una señal, seguido por un amplificador de señales pequeñas, un amplificador de señales grandes y un dispositivo transductor de salida. La señal de entrada del transductor es generalmente pequeña y debe amplificarse suficientemente para operar algún dispositivo de salida. Los factores de principal interés en los amplificadores de señal pequeña es la linealidad y la ganancia. Puesto que la señal de voltaje y corriente dentro de su propia entrada es pequeña, la capacidad de manejo de potencia y la eficiencia de potencia son de poca preocupación. Estos amplificadores

de voltaje proporcionan una señal de voltaje suficientemente grande a las etapas amplificadoras de señal grande para operar dispositivos de salida tales como motores que en nuestro caso operará un motor de de 1/50 Hp.

Esta etapa amplificadora corresponde a un amplificador de señales pequeñas. En la figura 3.1 se muestra la representación esquemática para este preamplificador. La característica de esta etapa es su alta ganancia, por lo tanto es necesario un pequeño voltaje de entrada entre -1 y 1 V. Esta etapa tiene una ganancia de aproximadamente 20, con un voltaje de salida entre +13 V y -13 V. Además tiene la característica de poseer dos entradas y dos salidas. Para una entrada positiva en el pin 1 se tendrá un voltaje positivo en el pin 3 y un voltaje de 0 V en el pin 4, en cambio para una entrada negativa en el pin 1 se tendrá un voltaje de salida positivo en el pin 4 y un voltaje de 0 V en el pin 3, característica que nos sirve para manejar el motor en las dos direcciones. En la figura 3.2 se muestra las características tanto de ganancia como de linealidad de esta etapa, por lo tanto este amplificador se lo utilizará en su zona lineal que corresponde a un rango de voltaje de entrada entre -0.6 V y +0.6 V.

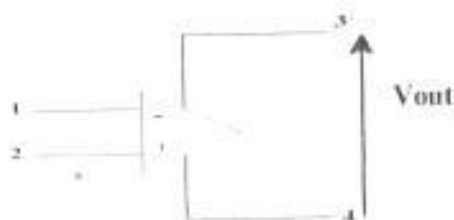


Figura 3.1 Etapa Pre - Amplificadora

La entrada 2 se conecta a tierra. En el capítulo 10 se muestra el diagrama esquemático del preamplificador usado en este proyecto.

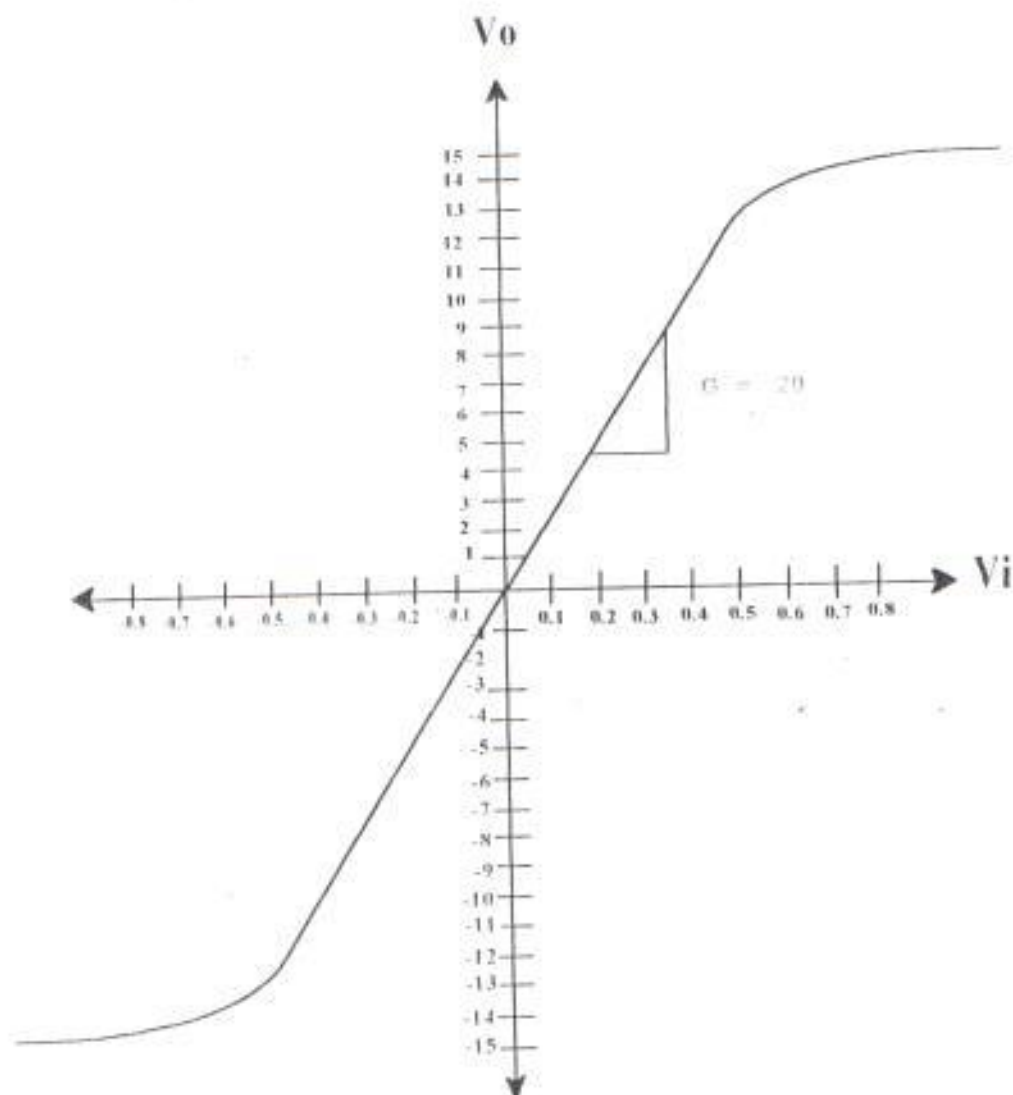


Figura 3.2 Característica de Ganancia del Preamplificador

3.2 ETAPA AMPLIFICADORA

Esta etapa consta de un amplificador operacional usado como un amplificador inversor - sumador implementado con CI LM741. En la figura 3.3 se muestra el circuito completo. La realimentación está abierta para proporcionarle ganancia variable o ganancia de 1 cerrando la realimentación. Si cerramos el lazo, el voltaje de salida es : $V_o = - (V_1 + V_2 + V_3)$, si tenemos resistencias iguales ; pero, si introducimos una ganancia variable con P_1 , entonces tendremos $V_o = - (V_1 + V_2 + V_3) / \alpha$, α es un porcentaje de P_1 .

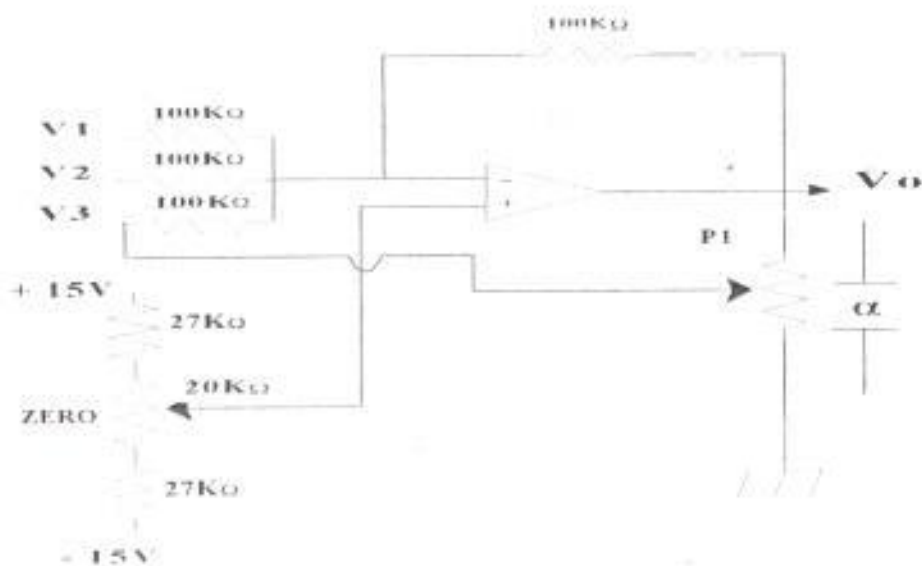


Figura 3.3 Etapa Amplificadora

Para nuestro proyecto este amplificador se denominará comparador o *detector de error*. Para utilizar este detector de error el primer paso será encerrarlo (ver manual del usuario), anulando de este modo el voltaje offset que se presenta en la salida.

3.3 SERVO AMPLIFICADOR

El servo es un amplificador que le da ganancia de potencia a la señal proveniente del preamplificador para manejar el motor. En este tipo de amplificadores se presentan problemas prácticos debido al acoplamiento directo, ya que cualquier cambio en el voltaje de polarización de los transistores debido a la variación de la fuente de alimentación, cambios de temperatura o envejecimiento de los componentes es tratado como una señal de entrada real. En la figura 3. 4 se muestra una representación esquemática de este servo amplificador.

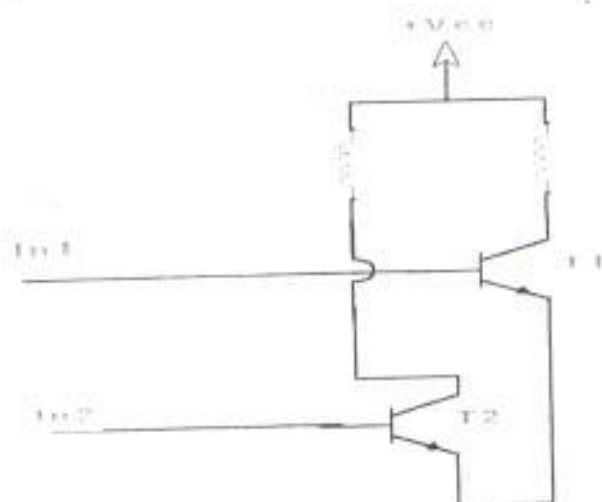


Figura 3. 4 Diagrama representativo para el Servo amplificador

En el capítulo 10 se muestra el diagrama esquemático del servo amplificador tanto para la conexión de armadura como de campo.

Este circuito amplificador tiene dos entradas positivas (externas), las cuales hacen girar al motor en dos sentidos. Este motor es de 1/50 Hp.

Si aplicamos una señal (voltaje positivo) a la entrada 1(ver diagrama esquemático del servo amplificador en el capítulo 10), este voltaje hace conducir a T2, entonces, el motor rotará a la derecha y con T4 (positivo) el motor rotará en sentido contrario. La armadura se conecta internamente entre el emisor de T2 o T4 y GND para control de armadura. Con este control el campo es conectado a los + 27 voltios a través de una resistencia de 0.5Ω . Los - 27 voltios sirven para asegurar que T2 o T4 conduzcan cuando el voltaje de entrada sea suficiente para poner en funcionamiento al motor. Los transistores T5 y T6 polarizan a T1 y T3 tanto para control de campo o armadura, respectivamente. Si queremos hacer girar el motor en sentido opuesto al anterior, lo único que se debe hacer es aplicar una señal de voltaje a la entrada 2 por medio de la cual entra a funcionar T3 y T4, e internamente el voltaje de alimentación se conecta a la otra mitad del devanado de campo; la entrada que no se usa se conectará a tierra.

CAPITULO IV

SERVOS DC

4.1. GENERADOR: Características

Cuando la variable controlada en un sistema de lazo cerrado es una función mecánica, este se denomina un servo sistema. En nuestro caso tenemos el sistema de control de posición, en el cual todos los elementos en conjunto se denominan servo mecanismos *dc* ya que básicamente funcionan con corriente *dc*. Los principales elementos son los potenciómetros tanto de entrada como de salida, el servo motor y el servo amplificador.

Los generadores *dc* son los mismos motores *dc*, con la diferencia en el sentido de flujo de potencia. En nuestro caso tenemos un generador *dc* con imán permanente, el cual produce el campo. La ecuación del voltaje generado en un generador *dc* es :

$$V_g = K \cdot B \cdot \omega \quad [V]$$

donde V_g representa el voltaje generado, K es alguna constante de proporcionalidad (que depende de la longitud y diámetro del rotor, etc) B es la fuerza del campo magnético (constante) y ω es la velocidad en rev/min. El voltaje generado es proporcional a la velocidad del eje hasta una velocidad de 440 rad/seg, como se muestra en la figura 4.1 ya que para velocidades mayores el generador entra a la zona de saturación es decir el voltaje generado no aumenta más. Este generador será acoplado al eje de alta del motor de pruebas para simular *carga* mediante un banco de resistencias que será conectado en los terminales del generador, en la práctica 1 se tendrá estas pruebas con carga. No se ha considerado la caída de tensión en las escobillas y armadura ya que su caída es mínima.

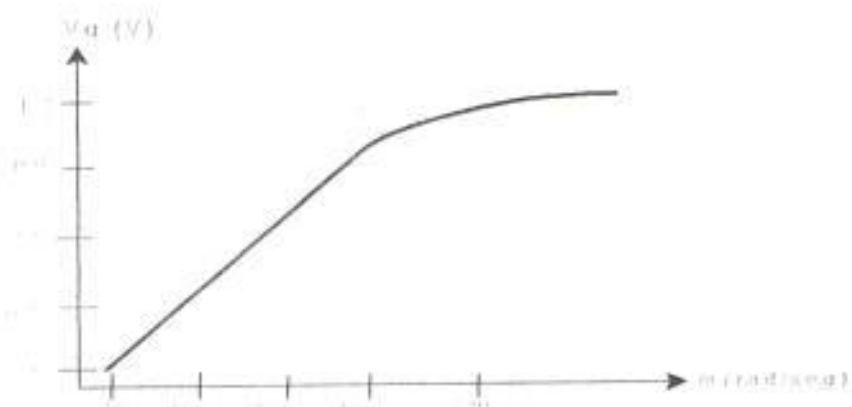


Figura 4.1 Voltaje generado vs. Velocidad para un Generador DC

4.2. TACOGENERADOR : Características

El tacogenerador a utilizar en los experimentos es un pequeño generador *dc* con imán permanente , el cual se lo usa como el elemento que transforma proporcionalmente la velocidad del motor en voltaje.

Este generador es el elemento de realimentación a nuestro sistema de control completando el sistema de lazo cerrado. Es necesario que el tacogenerador sea conectado con la polaridad correcta (Ver manual del usuario : apéndice A) ya que este se opondrá al voltaje de referencia en la entrada del servo amplificador.

Esto significa que si la velocidad disminuye debido a un incremento en la carga luego una señal más larga es habilitada para manejar el amplificador y mantener la velocidad constante.

Si la polaridad es incorrecta, la velocidad del motor se incrementará, ya que corresponde a una realimentación positiva que puede hacer que el sistema colapse. En el manual del usuario se presenta una tabla con valores obtenidos para voltaje y velocidad ; en el cual se obtuvo un máximo de 13 V para una velocidad de 6000 rev/min. En la figura 4.2 se muestra el simbolo esquemático del tacómetro para un giro de velocidad en favor de las manecillas de reloj.

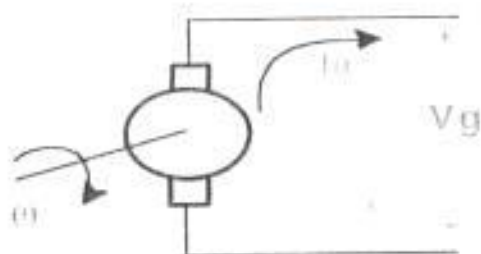


Figura 4.2 Símbolo Esquemático del Tacómetro

4.3. MOTOR : Características

Una característica esencial de cualquier sistema eléctrico de control de velocidad o posición es un *motor* eléctrico con una fuente de poder asociada y fase amplificadora para controlar el voltaje aplicado al motor en respuesta a una señal pequeña de control. Además, usualmente es necesario para el motor ser reversible. Las características del motor, esto es velocidad vs. voltaje y velocidad vs. torque tanto para conexión de campo como de armadura, se analizan a continuación.

4.3.1. CONTROL DE CAMPO

Colocando el motor como en la figura 4.3 la corriente en cada uno de los transistores es largamente determinada por cada una de las señales de entrada, es decir, es más sensitivo ya que para un pequeño incremento en la entrada, la velocidad aumenta rápidamente una vez que la rotación ha comenzado. Esto es

porque el torque suficiente ha sido obtenido para vencer la fricción de las escobillas, luego la corriente provista permanecerá constante en el colector ya que la fuerza contra electromotriz no afecta la corriente a través del transistor; el motor puede acelerar lentamente hasta una velocidad alta sino ha sido cargado. Luego si aplicamos una pequeña carga causa una gran reducción en velocidad porque la corriente no se incrementa con el decrecimiento de la velocidad como si ocurre con la conexión de armadura. Esto puede ser una desventaja, es decir, que no es aconsejable en un sistema de lazo cerrado ya que afectaría la estabilidad del sistema. En las figuras 4.4 y 4.5 se muestran las características para este tipo de conexión.

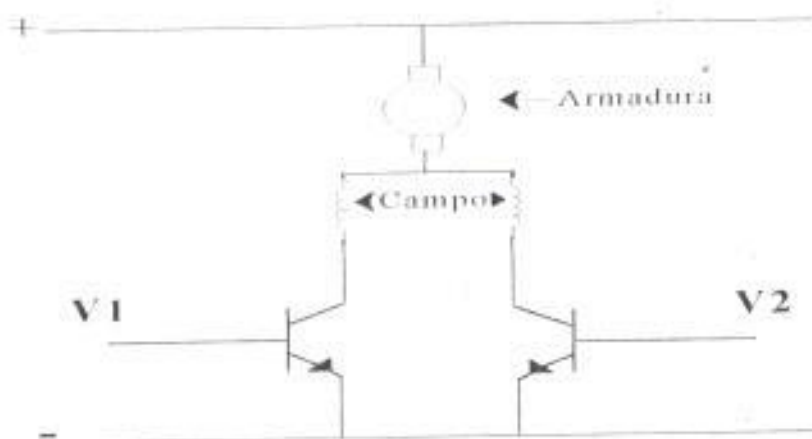


Figura 4.3 Servo Amplificador: Conexión de campo

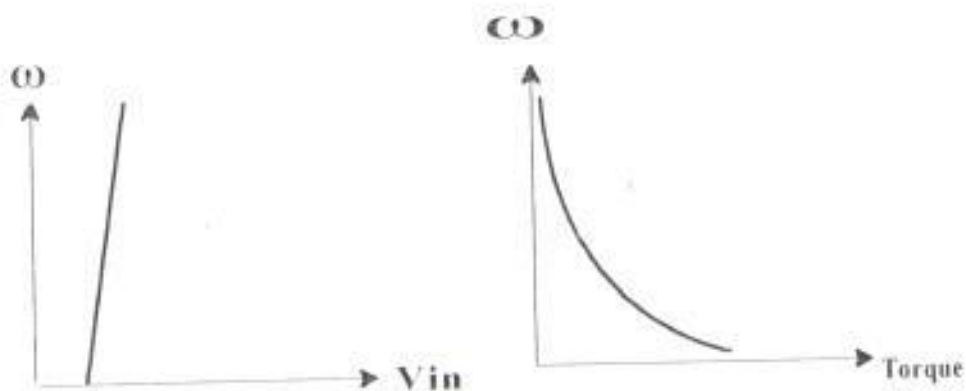


Figura 4. 4 Velocidad vs. voltaje de entrada Figura 4. 5 Velocidad vs. Torque

En la figura 4.6 se ilustra el modelo del diagrama de bloques de un motor controlado por campo, en el cual se muestra que no tiene realimentación, por lo tanto es un sistema de lazo abierto, es decir, no es controlable ya que para pequeños cambios en la entrada la velocidad sube o baja rápidamente.

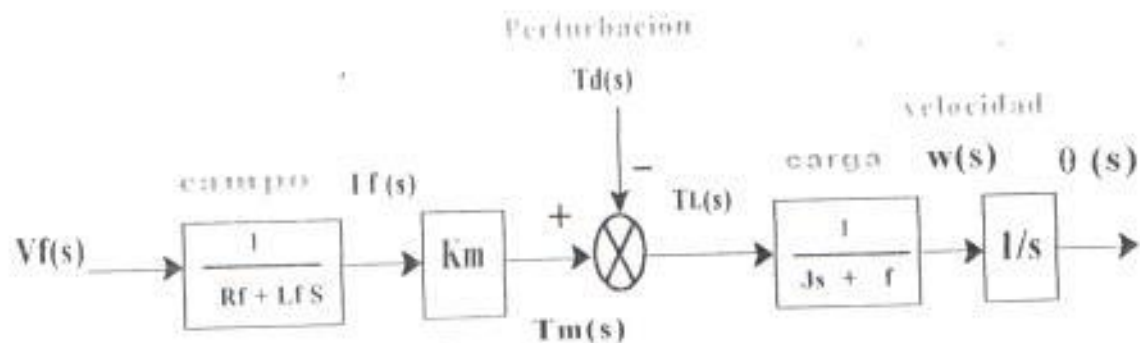


Figura 4. 6 Diagrama de bloques de un motor controlado por campo

en donde: $\tau_m = K_m I_f(s)$; $K_m = K_t K_f I_a$

$$I_f(s) = \frac{V_f(s)}{R_f + L_f S}$$

en donde K_m es la constante del motor. El momento de torsión del motor $T_m(s)$ es igual al proporcionado a la carga, de donde $\tau_m = T(s) + T_d(s)$, $T(s)$ es el momento de torsión de la carga y $T_d(s)$ es el momento de torsión de perturbación ≈ 0 y $T(s) = J S^2 \theta(s) + f S \theta(s) = \theta(s) S [J S + f]$.

Por tanto, la función de transferencia en términos de la combinación motor- carga es:

$$\frac{\theta(s)}{V_f(s)} = \frac{K_m}{S [J S + f] (R_f + L_f S)}$$

En forma alterna, la función de transferencia puede escribirse en términos de las constantes de tiempo del motor como:

$$\frac{\theta(s)}{V_f(s)} = G(s) = \frac{K_m f R_f}{S [\tau_f S + 1] (\tau_m S + 1)}$$

donde $\tau_f = L_f / R_f$ y $\tau_i = J / f$. Típicamente, encontramos que τ_i es mayor que τ_f y con frecuencia la constante de tiempo del campo puede despreciarse

4.3.2. CONTROL DE ARMADURA

En la figura 4.7 la armadura está conectada al circuito emisor con una vuelta de campo en cada colector. Aquí la f.e.m. de la armadura aparece entre el emisor y tierra. La elevación del voltaje V1 o V2 incrementa la velocidad del motor y si no hay carga, la velocidad es directamente controlada por la entrada. Si el motor es cargado la velocidad cae y la corriente se incrementa si V1 es mantenido constante.

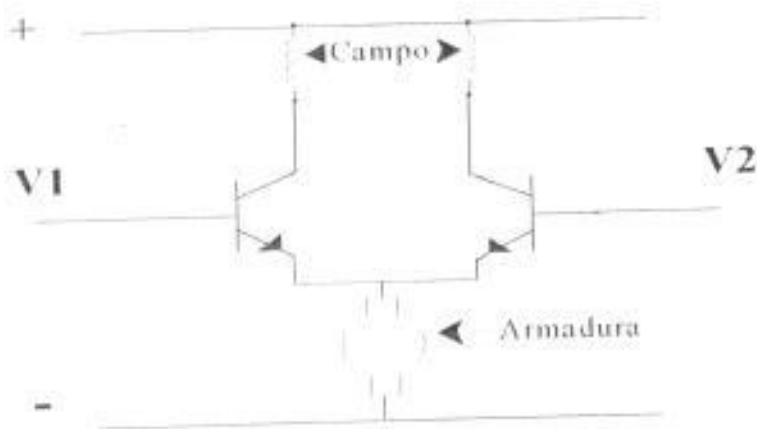


Figura 4.7 Servo Amplificador : Conexión de Armadura

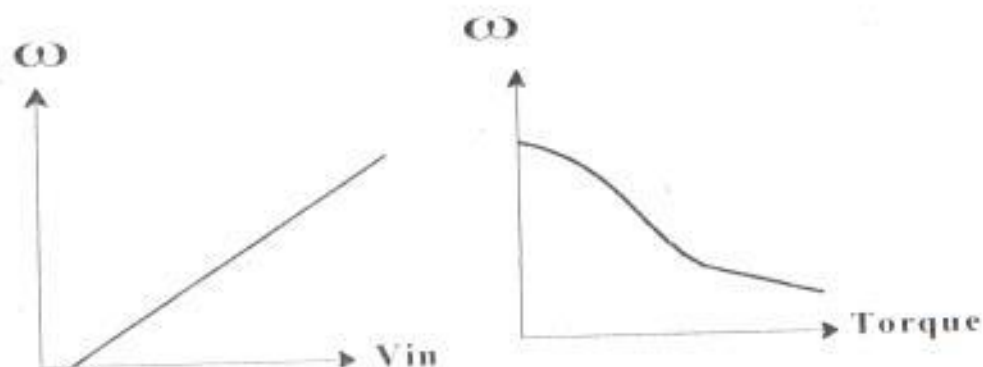


Figura 4. 8 Velocidad vs. Voltaje de entrada Figura 4. 9 Velocidad vs. Torque

Sin embargo el torque sube para guardar el movimiento de la carga. Este da las características de *velocidad / voltaje de entrada* y *velocidad / torque* de la forma general mostrada en la figuras 4.8 y 4.9 respectivamente. Además en la figura 4.10 se muestra el modelo del diagrama de bloques para el motor dc controlado por armadura.

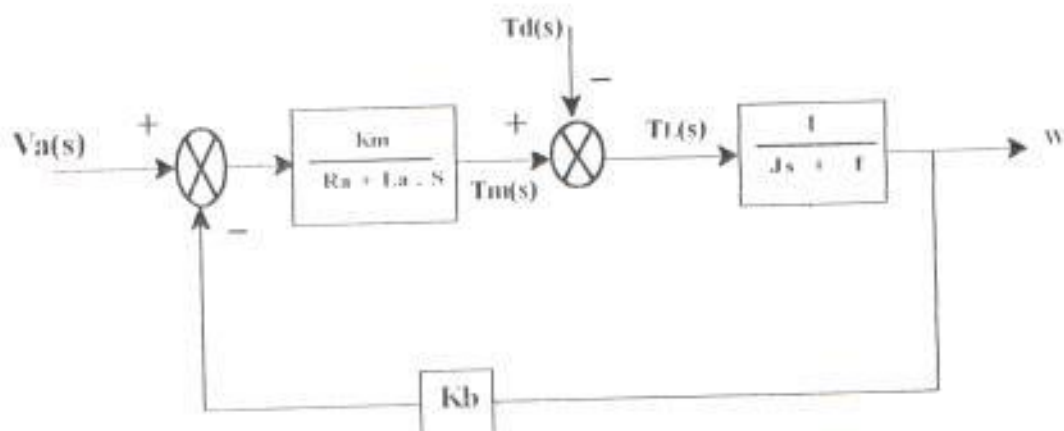


Figura 4. 10 Diagrama de bloques para el motor dc controlado por armadura

El motor de cd controlado por armadura utiliza una corriente de campo constante y, por tanto, el momento de torsión del motor es :

$$T_m(s) = (K_t K_f I_f) I_a(s) = K_m I_a(s)$$

La corriente de armadura se relaciona con el voltaje de entrada aplicado a la armadura por :

$$V_a(s) = (R_a + L_a S) I_a(s) + V_b(s)$$

donde $V_b(s)$ es el voltaje de la fuerza contra-electromotriz proporcional a la velocidad del motor donde

$$V_b(s) = K_b \omega(s)$$

y la corriente de armadura es :

$$V_a(s) = K_b L_a \omega(s)$$

$$I_a(s) = \frac{V_a(s) - K_b L_a \omega(s)}{R_a + L_a S}$$

$$R_a + L_a S$$

y el momento de torsión de la carga es :

$$Tl(s) = JS \cdot \theta(s) + f(s) \cdot \theta(s) = Tm(s) - Td(s)$$

Por lo tanto se determina que es un sistema realimentado lo que hace que el sistema sea estable. Utilizando las ecuaciones anteriores o en forma alternativa el diagrama de bloques, obtenemos su función de transferencia :

$$G(s) = \frac{\theta(s)}{Va(s)} = \frac{Km}{s [(Ra + LaS) (JS + f) + KbKm]}$$

Sin embargo, para muchos motores de cd es despreciable la constante de tiempo de la armadura, $\tau_a = La/Ra$, y por tanto

$$G(s) = \frac{\theta(s)}{Va(s)} = \frac{Km}{s [(Ra(JS + f) + KbKm)]} = \frac{[Km / (Ra f + KbKm)]}{s (\tau_l s + 1)}$$

Donde la constante de tiempo equivalente es $\tau_l = Ra J / (Ra f + KbKm)$.

4.4. MEDICION DE LAS CARACTERISTICAS DEL MOTOR

Para conexión de campo o armadura rotamos el switch correspondiente en el panel del banco de pruebas, de tal manera si colocamos el switch para conexión de campo, el motor girará a una velocidad alta para un pequeño incremento de voltaje, esto es porque el torque ha sido suficiente para vencer la fricción de las escobillas, ya que la corriente del transistor permanece constante debido a que la *f.e.m* no afecta esta corriente, luego si aplicamos una pequeña carga al motor notamos que la velocidad cae bruscamente debido a que la corriente no se incrementa con la elevación de la velocidad.

También para el caso en que el motor permaneciendo cargado para un voltaje de entrada grande, la velocidad aumenta rápidamente, si reducimos la carga. Estas diferencias, para cargas pequeñas son factores importantes en los sistemas de lazo cerrado. Luego se tendrá un gráfico como en la figura 4.4 y 4.5 tanto para Velocidad vs. Voltaje como Velocidad vs Torque respectivamente. Adaptando la conexión de armadura para cualquier valor de voltaje de entrada se tendrá la forma general de la figura. 4.6 y 4.7, alcanzando una velocidad máxima de alrededor 6000 r.p.m para una entrada de 13V. Se han aplicado diferentes valores de carga al motor, como se puede observar en el manual del usuario.

4.5. RESPUESTA TRANSIENTE DEL MOTOR

Teóricamente si la entrada del amplificador de control cambia, el motor debería responder inmediatamente a esta variación, pero en la práctica existe un atraso entre la aplicación de una entrada de paso de voltaje y la respuesta del motor.

Para un motor ideal controlado por armadura (uno en que la constante de fricción de la escobillas es insignificante comparado con el torque generado por el motor), con un campo energizado separadamente, la velocidad del motor responde exponencialmente hacia un cambio de paso en la entrada. La relación es de la misma forma como el de un circuito RC para un paso de voltaje.

En la figura 4.11 se muestra un circuito RC en el cual un paso inicial de voltaje causa una corriente V_s/R que fluye al capacitor cuya carga se eleva. Sin embargo, si el voltaje a través de R es reducida, el voltaje en el capacitor se incrementa lentamente. En la figura 4.12 existe una corriente inicial V_s/R_a , pero como la velocidad del motor se incrementa la *fem* que se genera reduce la corriente a través del motor y la velocidad se incrementa lentamente. Estas relaciones dan la siguiente ecuación para un circuito RC .

$$V_2 = V_1 (1 - e^{-t/\tau}) \quad \tau = R \cdot C$$

y para el motor se tiene la siguiente ecuación:

$$W = K_s \cdot V_s (1 - e^{-(t/\tau_m)})$$

$$K_s = \text{Velocidad} / \text{voltio aplicado}$$

$$\tau_m = \text{constante de tiempo del motor}$$

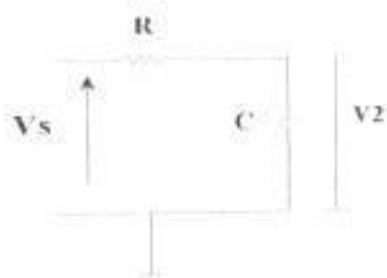
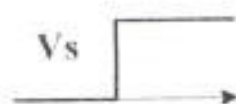


Figura 4. 11 Circuito RC

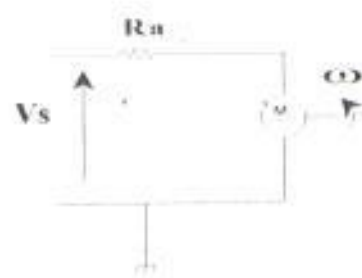
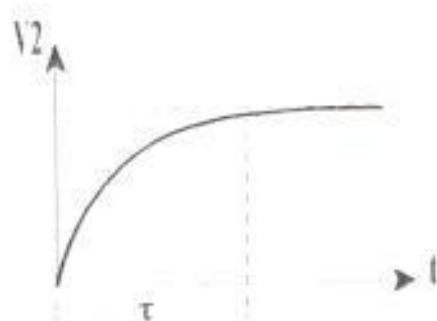
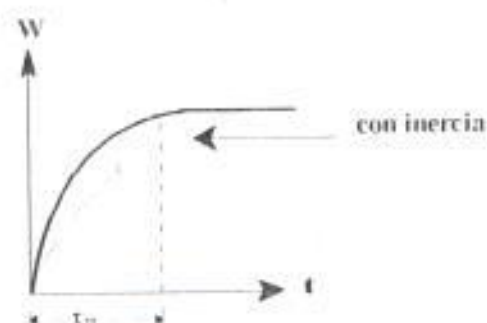


Figura 4. 12 Circuito del motor

Para una entrada tipo escalón se tendrá las respuestas como se muestra en la figura 4.13 y figura 4.14 tanto para el circuito RC como para el motor.



$$V_2 = V_1 (1 - e^{-\exp(-t/\tau)})$$



$$W = K_s \cdot V_s (1 - e^{-\exp(-t/\tau)})$$

Figura 4. 13 Respuesta de un circuito RC

Figura 4. 14 Respuesta del motor

Ambas relaciones son caracterizadas por una constante de tiempo que es igual al tiempo que sería tomado por la respuesta para alcanzar el valor final si fuese mantenida la pendiente de la salida inicial. Para el circuito RC, τ está en segundos, R en ohmios y C en faradios, para el motor τ depende de los parámetros mecánicos (como son fricción de inercia y viscosidad), y de los parámetros eléctricos (resistencia de armadura y fem generada). Agregando inercia al motor, la respuesta es más lenta como se muestra en la figura 4.13

4.6. CONTROL DE VELOCIDAD DEL MOTOR

En la fig 4.15 se muestra un sistema de control de velocidad del motor :

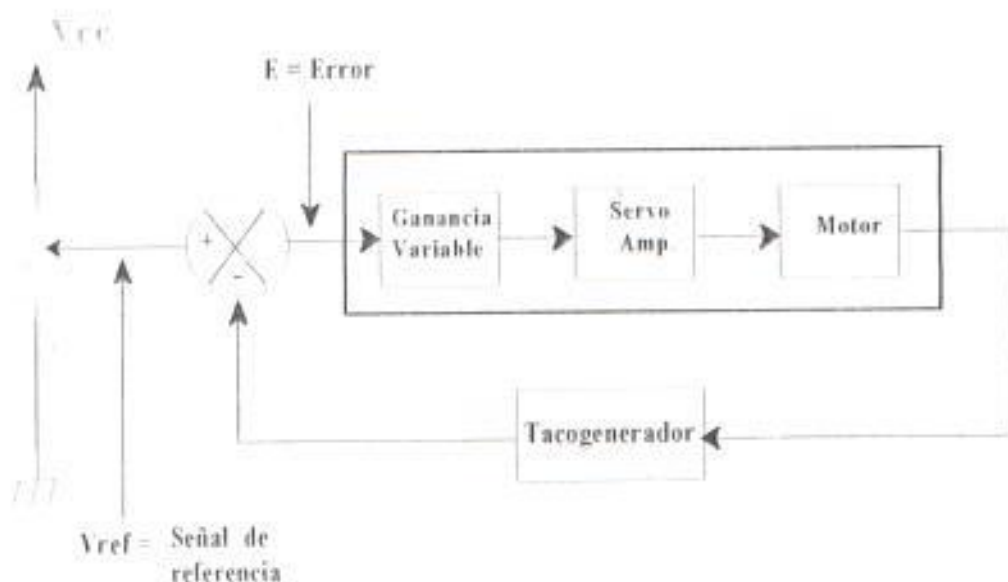


Figura 4. 15 Sistema de Control de Velocidad

Aquí se muestra que el voltaje del tacogenerador es restado del voltaje de referencia (valor deseado), para dar una señal de error que finalmente opera el sistema. Idealmente si la ganancia del amplificador es muy alta, una pequeña señal de error es requerida para operar el sistema y luego la velocidad del motor será tal que el voltaje del tacogenerador sea igual al voltaje de referencia, por lo que la velocidad es controlada por la señal de referencia. Para las condiciones de estado estable el sistema puede ser representado por la siguiente relación.

$$W = K.E$$

En donde W es la velocidad la cual es proporcional a la señal de error E , y K es la ganancia del sistema. También:

$$E = V_{ref} - K_g W$$

El error es la diferencia entre el voltaje de referencia y el voltaje generado, combinando estas dos relaciones tenemos:

$$W = K(V_{ref} - K_g W) \quad \text{de donde}$$

$$W = K V_{ref} / (1 + K K_g)$$

Por lo tanto si $K K_g \gg 1$, esta última relación será aproximadamente:

$$W = V_{ref} / K_g \quad \text{pero como } K_g \approx \text{cte}$$

$$\Rightarrow W \propto V_{ref}$$

y la velocidad es directamente proporcional al voltaje de referencia; la relación depende solamente de la constante del generador que no variará significativamente.

La relación de velocidad con el voltaje de referencia no será afectado por las

variaciones de ganancia dado que ésta permanece alta. Esta es la ventaja de usar un sistema de lazo cerrado. Una relación alternativa puede ser obtenida como:

$$E = Y_{ref} - KgKE$$

luego,

$$E = Y_{ref} / (1 + KgK)$$

que también muestra que el error disminuye según como la ganancia del sistema K se incrementa.

4.7. CONTROL DE POSICION DEL MOTOR

En un sistema de control de posición un requerimiento común para el motor es que el mismo rote un ángulo de salida igual al ángulo de entrada. La forma general de un sistema de control de posición es como se muestra en la figura 4.16. Para un control de posición los transductores de entrada y salida deben medir posiciones angulares y producir una señal de error proporcional al ángulo de desplazamiento.

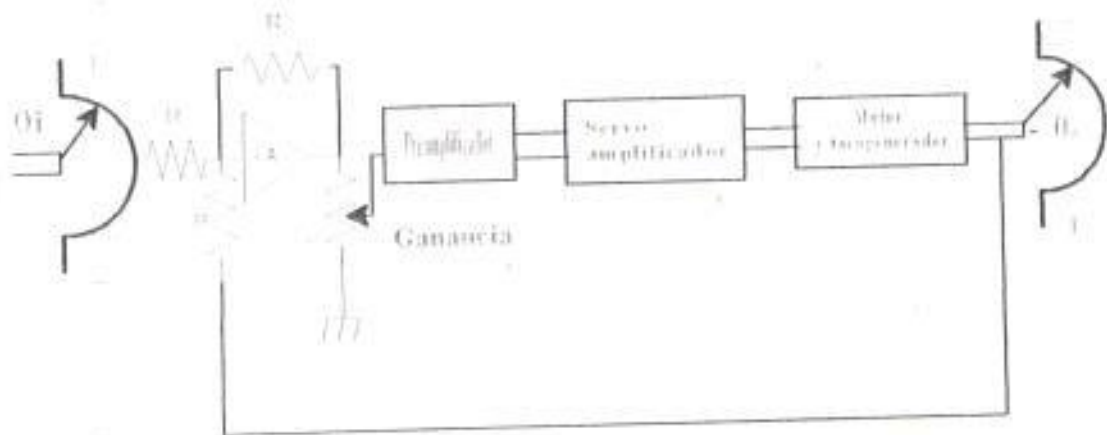
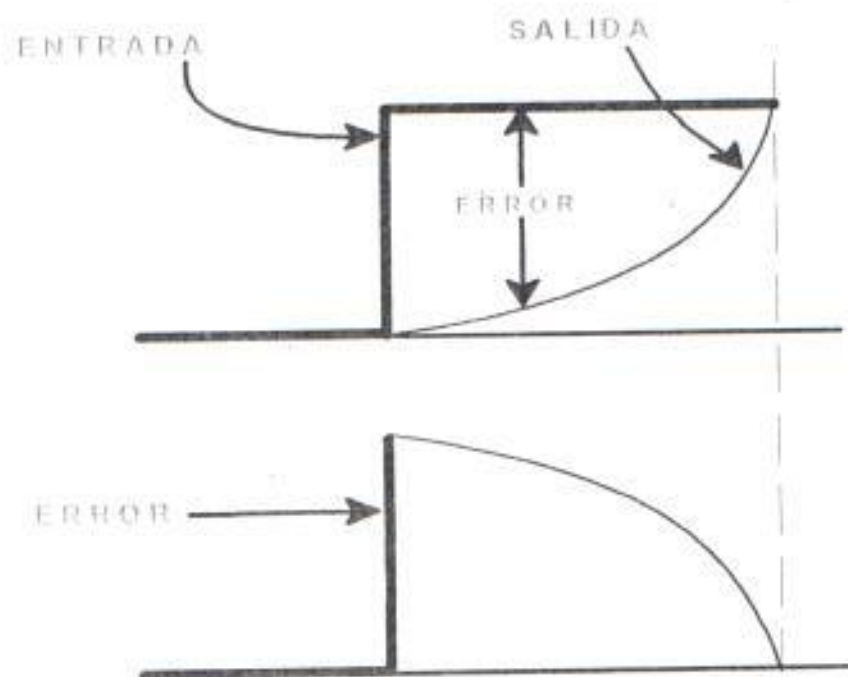


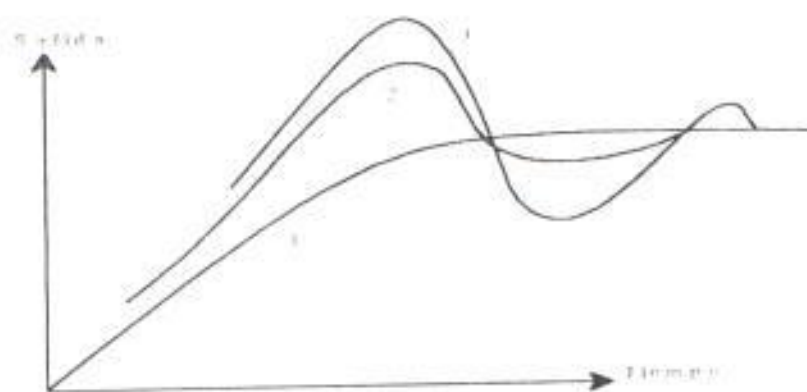
Figura 4.16 Sistema de Control de Posición

En un servo sistema como el presente caso uno de los factores de interés en el rendimiento es la *bandamuerta*, esto es la mínima señal que se necesita para que el sistema responda y el desarrollo transiente en términos de sobrenivel (*overshoot*) y en términos de tiempo que le toma en estabilizarse.

La *bandamuerta* crece porque una cierta señal mínima es requerida para mover el motor y la *bandamuerta* disminuye con el incremento de la ganancia. Un sobrenivel aumenta porque en respuesta a un cambio en la entrada existirá una señal de error la cual seguirá manejando el motor en el punto de alineamiento (donde idealmente el motor debería detenerse) y sin embargo atraviesa el alineamiento, en la figura 4.17 (a y b) se muestran estas características en el estado transiente.



(a)



(b)

Figura 4.17 Condición transiente

El error eventualmente revierte el sentido de giro del motor hacia el alineamiento pudiendo sobredispararse nuevamente hasta que el sistema se estabilice. Lo expresado anteriormente se ilustra en estas figuras. Para ganancias bajas, como se muestra en la figura 4.17 (b) (curva 1) no hay sobrenivel, pero en general incrementando la ganancia (curvas 2 y 3) tanto el sobrenivel como el tiempo de estabilización se incrementan. Esto se incrementa porque una ganancia mayor provee mas manejo al motor el cual estaria moviéndose rápidamente cuando el alineamiento ocurre.

CAPITULO V

SISTEMAS REALIMENTADOS

5.1. SISTEMAS DE LAZO ABIERTO vs. SISTEMAS DE LAZO CERRADO

Los sistemas realimentados son aquellos que tienden a mantener una relación preestablecida entre la salida y alguna entrada de referencia, comparándolas y utilizando la diferencia como medio de control.

A estos sistemas realimentados se les denomina *auto-correctivos*, es decir, tienen la habilidad de monitorcar automáticamente una cierta variable en un determinado proceso y sin intervención humana, corregirla si no es aceptable. Los sistemas de control en los que la salida no tiene efecto, la acción de control se denomina sistemas de *lazo abierto*. En cambio, en un sistema de *lazo cerrado* se produce un error el

cual tiene la facilidad de compensar cualquier desviación de la salida, ya que el error causa una señal de corrección para ser aplicada en el paso siguiente. Estas desviaciones pueden ser causadas por una carga externa, una perturbación interna o cambios en los parámetros del sistema. En la fig. 5.1 se muestra un sistema de control de lazo cerrado.

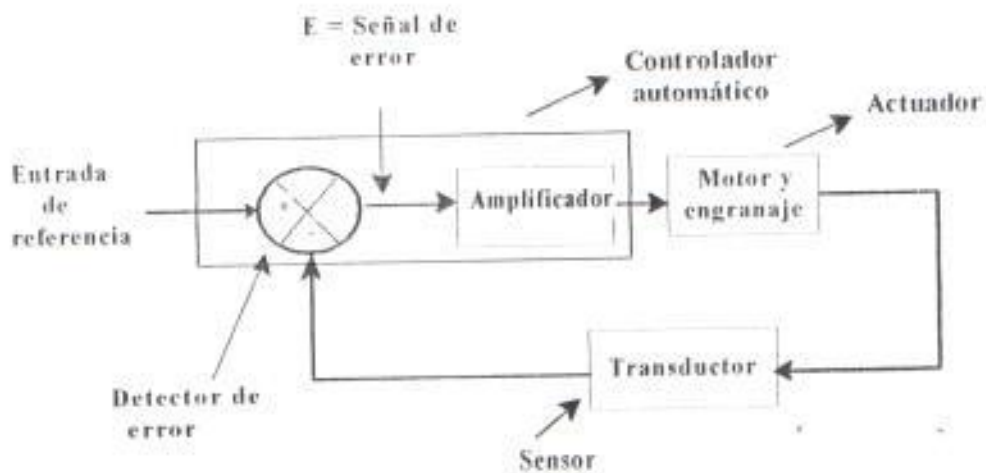


Figura 5.1 Sistema de control de lazo cerrado

El detector de error hace la comparación entre el valor medido de la variable controlada y el valor de referencia el cual representa el valor deseado de la variable. Luego genera una señal de error, el cual representa la diferencia entre el valor deseado y el valor medido. La señal de error se considera igual al valor deseado

menos el valor medido, de modo que si el valor medido es demasiado grande, la señal de error es negativa, y si el valor medido es demasiado pequeño es de polaridad positiva. Esto está expresado en la siguiente ecuación:

$$\text{ERROR} = \text{VALOR DESEADO} - \text{VALOR MEDIDO}$$

El detector de error detecta la señal que suele estar a un nivel de potencia muy bajo, el cual es amplificado a un nivel suficientemente alto. Así el controlador automático está constituido por un detector de error y un amplificador. También suele haber un circuito de retroalimentación adecuado y un bloque controlador que se utiliza para alterar la señal de error, amplificándola y a veces diferenciándola y/o integrándola para producir una mejor señal de control. La salida de un controlador automático alimenta a un actuador o accionador, que en nuestro caso es el motor de. El sensor o elemento de medición es un dispositivo que convierte la variable de salida en otra variable adecuada que en nuestro caso tenemos como sensor un tacogenerador de, que convierte la variable de velocidad en una variable de voltaje de. El punto de ajuste del controlador debe ser tal que la entrada de referencia tenga las mismas unidades que la señal de retroalimentación del sensor.

La desventaja de un sistema realimentado es que debido al paso de lazo cerrado el sistema puede tender a dar una respuesta oscilatoria para cualquier cambio en la

entrada, que puede durar algún tiempo en desaparecer, o que el sistema puede eventualmente llegar a ser inestable y mantener media oscilación, estos efectos son ilustrados en la fig. 5.2

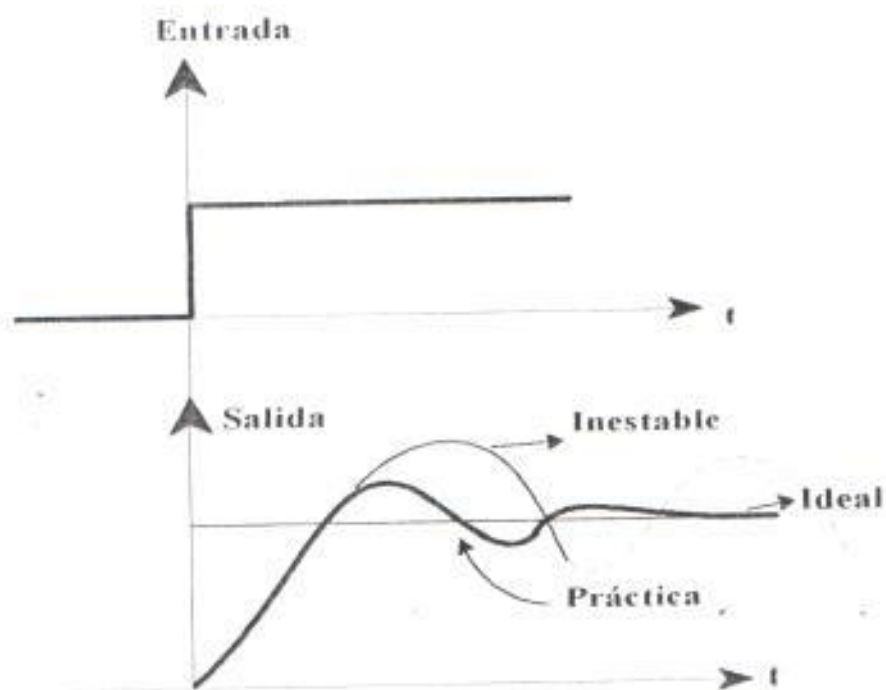


Figura 5.2 Efectos de un sistema de lazo cerrado

5.2. MODOS DE CONTROL EN SISTEMAS DE LAZO CERRADO

La manera como el controlador reacciona a una señal de error es una indicación del modo o leyes de control, generalmente se mencionan 5 leyes de control, como son :

- a) Encendido/Apagado(ON/OFF)
- b) Proporcional (P)
- c) Proporcional - Integral (PI)
- d) Proporcional - Derivativo (PD)
- e) Proporcional - Integral - Derivativo (PID)

La lista está en orden de menor a mayor complejidad de los mecanismos y circuitería involucrada. En general, entre más difícil es el problema de control se debe ir más abajo en la lista para encontrar al modo apropiado de control. Para nuestro proyecto se diseñó e implementó los 4 últimos modos de control utilizando op-amps, es decir se construyó controladores electrónicos.

5.2.1. CONTROL PROPORCIONAL

El tipo más elemental de control es el llamado "Encendido-Apagado" en el que la salida del proceso está siempre con 0 ó 100% de respuesta. Una extensión natural del control "ON-OFF" es el control Proporcional. En este tipo de acción existe una relación lineal entre la señal de error $e(t)$ de entrada y salida $m(t)$ del controlador, por lo que la salida del proceso tendrá una respuesta proporcional a la señal de mando del controlador. El controlador proporcional es esencialmente un amplificador con ganancia ajustable, como se muestra en la ecuación 5.1.

$$m(t) = m_0 + K_p \cdot e(t) \quad (5.1)$$

$$e(t) = [r(t) - c(t)] \quad (5.2)$$

en donde,

m_0 = valor base , generalmente se fija durante la calibración del controlador

K_p = constante de proporcionalidad

$r(t)$ = punto de control

$c(t)$ = variable que se controla

En las ecuaciones 5.1 y 5.2 se ve que la salida del controlador es proporcional al error entre el punto de control y la variable que se controla ; la proporcionalidad de la ganancia del controlador , K_p , con esta ganancia se determina cuanto se modifica la salida del controlador con un cierto cambio en el error, esto se muestra en la figura 5.3

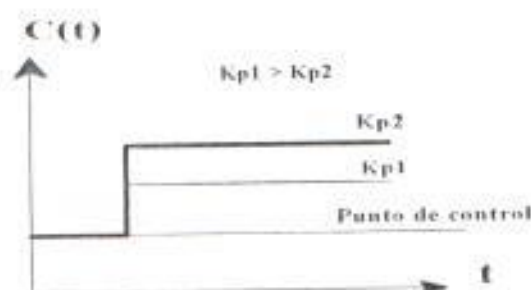


Figura 5.3 Respuesta del Control Proporcional

En la figura 5.4 se aprecia que cuanto mayor es la relación de K_p tanto menor es la desviación pero la respuesta del proceso se hace más oscilatoria, para la mayoría de los procesos existe un valor máximo más allá del cual el proceso se hace inestable

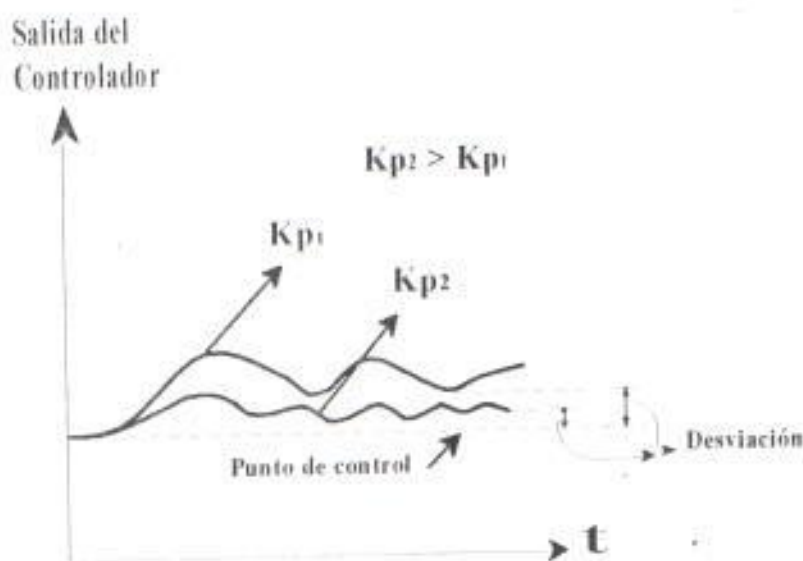


Figura 5.4 Salida del Controlador a diferentes valores de K_P

El término utilizado comúnmente para definir la ganancia o sensibilidad de un control proporcional es *banda proporcional*. Es la modificación expresada en porcentaje de variación de la entrada del controlador (señal de error) requerida para producir un cambio del 100% en la salida del actuador. Así, una banda

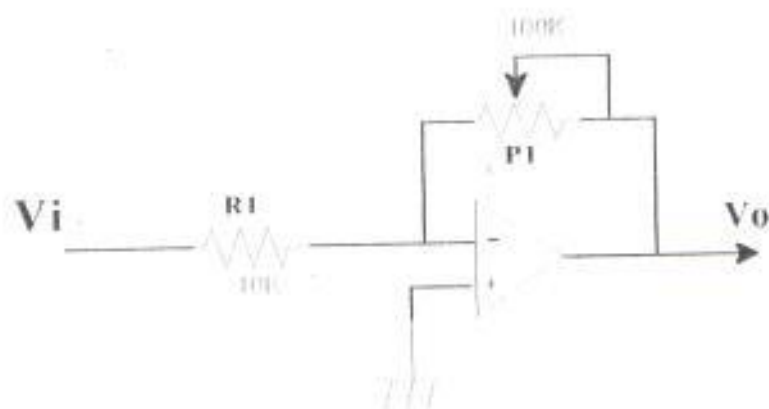
proporcional pequeña corresponde a una ganancia elevada o a una sensibilidad alta.

$$\text{BANDA PROPORCIONAL} = 100\% / K_p$$

Los controladores que son únicamente proporcionales tienen la ventaja de que solo cuentan con un parámetro de ajuste, K_p , sin embargo, adolecen de una gran desventaja, operan con un desviación o error de estado estacionario en la variable que se controla. A continuación se explica de manera simple el por qué existe la desviación. Supóngase que se tiene la siguiente ecuación:

$$m(t) = m_0 + K_p \cdot e(t)$$

Si m_0 es igual a 9 en un determinado proceso y si por alguna causa $m(t)$ es igual a 10, la única forma que en la ecuación anterior se cumpla la igualdad es que $K_p \cdot e(t) = 1$ y para que esto se cumpla el término de error $e(t)$, no puede ser cero en el estado estacionario. En la figura 5.5 se muestra la realización práctica de un circuito con opamps (LM348), cuyo funcionamiento se adecua a la ecuación 5.1, en el que P1 permite el ajuste de la constante de proporcionalidad (K_p).



$$V_o = (PI/R1) \cdot V_i$$

$$V_o = K_p \cdot V_i$$

Figura 5.5 Diagrama del Controlador Proporcional

donde:

V_o = señal de salida

V_i = señal de error

$PI/R1$ = Constante de proporcionalidad K_p

5.2.2. CONTROL PROPORCIONAL - INTEGRAL

En muchos procesos no se puede controlar con una desviación, entonces se debe añadir inteligencia al controlador proporcional, este nuevo modo de control es la

acción integral o de reajuste y en consecuencia, el controlador se convierte en controlador proporcional- integral, la siguiente es su ecuación descriptiva.

$$m(t) = K_p [r(t) - c(t)] + \frac{K_p}{T_i} \int [r(t) - c(t)] dt \quad (5.3)$$

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt \quad (5.4)$$

- $m(t)$ = salida del controlador
- $r(t)$ = punto de control
- $c(t)$ = variable que se controla
- K_p = ganancia proporcional
- T_i = tiempo integral

Utilizando la transformada de Laplace se tiene:

$$\begin{aligned} M(s) &= K_p E(s) + E(s) K_p / (T_i s) \\ &= K_p E(s) [1 + 1/(T_i s)] \end{aligned}$$

en consecuencia la función de transferencia es :

$$\frac{M(s)}{E(s)} = K_p \left[1 + 1/(T_i s) \right]$$

Ambos valores, K_p y T_i son ajustables. El tiempo integral regula la acción de control integral mientras que una modificación en K_p afecta tanto a la parte integral como a la proporcional de la acción de control. El recíproco del tiempo integral T_i recibe el nombre de frecuencia de reposición. La frecuencia de reposición es la cantidad de veces por minuto en que se repite la acción proporcional. La frecuencia de reposición se mide en términos de repeticiones por minuto para entender el sentido físico del tiempo de reajuste T_i .

Observando la figura 5.8, se deduce que tanto menor es el valor de T_i cuanto más pronunciada es la curva de respuesta, lo que significa que la respuesta del controlador se hace más rápida. Otra forma de analizar es mediante la observación de la ecuación 5.4, tanto menor es el valor de T_i , cuanto mayor es el término delante de la integral, K_p/T_i , y, en consecuencia, se le da mayor peso a la acción integral o de reajuste. La figura 5.6 muestra el diagrama de bloques de un controlador proporcional integral. Si la señal de error $e(t)$ es una función escalón unitaria como se ve en la figura 5.7, la salida del controlador $m(t)$ pasa a ser indicada en la figura 5.8.

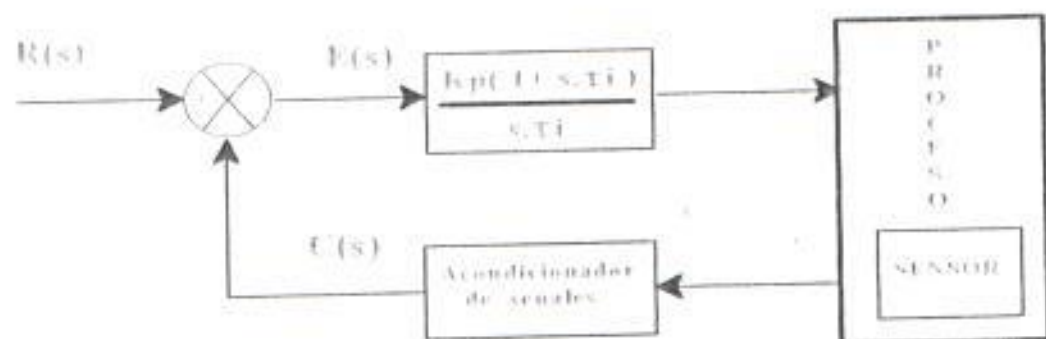


Figura 5.6 Controlador PI

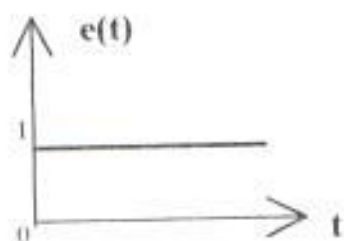


Figura 5.7 Entrada al Controlador - función escalón unitario

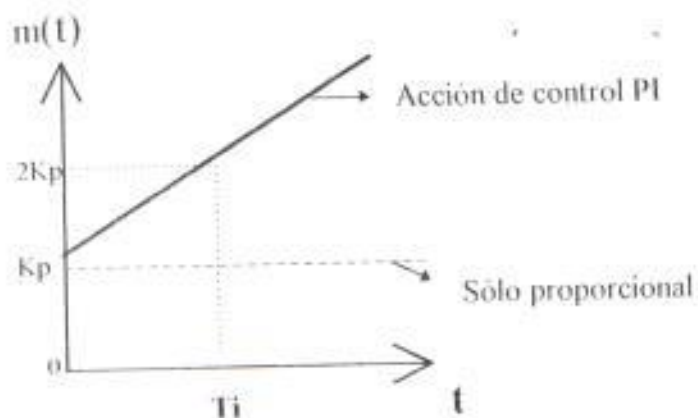


Figura 5.8 Salida del Controlador PI

para el presente proyecto en la figura 5.9 se muestra la implementación de un controlador proporcional - integral con circuitos integrados(LM348), en el cual se cumple la ecuación 5.3. En el circuito, R2 , tiene como objetivo estabilizar la ganancia del integrador en baja frecuencia, P2 es el encargado de dar diferentes valores al integrador y mediante P1 variamos el control proporcional.

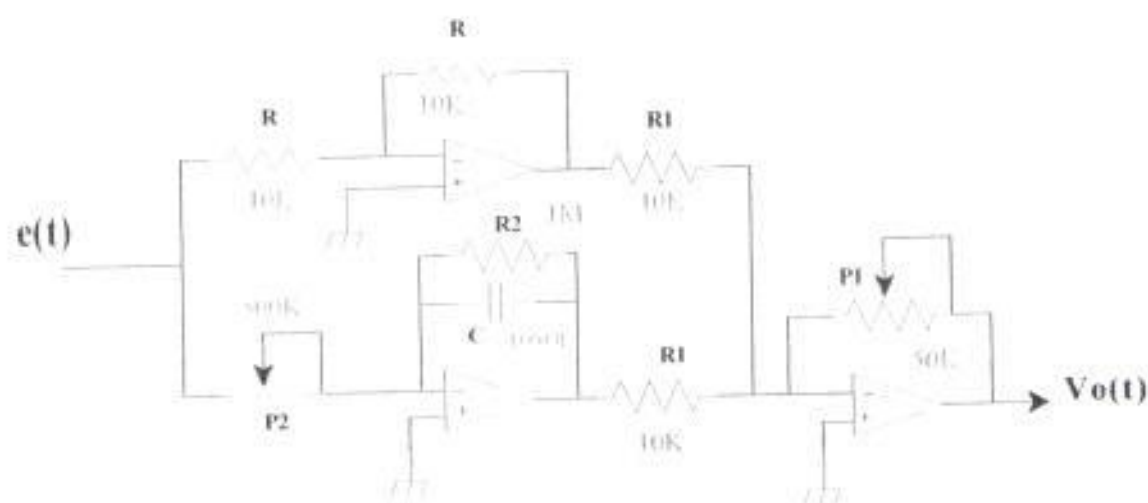


Figura 5.9 Diagrama esquemático del Controlador PI

$$V_o(t) = \frac{P1}{R1} e(t) + \frac{P1}{R1} \times \frac{1}{P2 \cdot C} \int_0^t e(t) dt \quad (5.5)$$

la ecuación 5.5 corresponde a un controlador PI , en donde $T_i = P2 \cdot C$, es el tiempo integral , el cual viene dado en segundos , también $K1 = 1/T_i$, es la ganancia

integral o razón de reposición. Cuando el T_i es grande (P_2 grande), la parte integral es menos efectiva (lenta para producir el efecto buscado). Cuando la constante de tiempo es pequeña (P_2 pequeña), la parte integral es más efectiva. De donde tenemos:

$V_o(t)$ corresponde a la señal de salida $m(t)$

$e(t)$ corresponde a la señal de error

$P_2 \cdot C$ corresponde al tiempo de integración = T_i (seg.)

$K_I = 1/T_i$ corresponde a la ganancia integral

$P_1/R_1 = K_p$

5.2.3. CONTROLADOR PROPORCIONAL - DERIVATIVO

La acción de un control proporcional derivativo se define por la siguiente ecuación

$$m(t) = K_p \cdot e(t) + K_p \cdot T_d \cdot de(t)/dt$$

donde K_p es la ganancia proporcional y T_d es la ganancia denominada *tiempo derivativo o tiempo de adelanto*. Tanto K_p como T_d son regulables. La acción derivativa se presenta cuando el valor de salida del controlador es proporcional a

la velocidad de variación de la señal de error. El tiempo derivativo T_d es el intervalo de tiempo en el que la acción derivativa se adelanta al efecto de la acción proporcional.

En la fig. 5.10 se puede ver un diagrama de bloques de un controlador proporcional - derivativo. Si la señal de error $e(t)$ es una función rampa como se muestra en la figura 5.11, la salida del controlador $m(t)$ es la que se ve en la figura 5.12. Como puede verse en esta figura, la acción derivativa tiene una característica *anticipatoria*. Por supuesto una acción derivativa nunca puede anticipar una acción que aun no acontece.

En un sentido intuitivo, la parte derivativa de un controlador intenta "mirar adelante" y prevee que el proceso sufrirá un gran cambio basándose en las medidas actuales. Es decir, si la variable medida está cambiando muy rápidamente, es seguro que tratará de cambiar en una gran cantidad. Siendo este el caso, el controlador tratará de "anticiparse" al proceso aplicando más acción correctora. En tanto acontece la acción derivativa tiene una ventaja al anticiparse al error, sus desventajas son que amplifica las señales de ruido y produce un efecto de saturación en el actuador. Nótese que nunca se usará una sola acción de control derivativo, ya que este control es efectivo ~~solo~~ durante periodos transitorios.

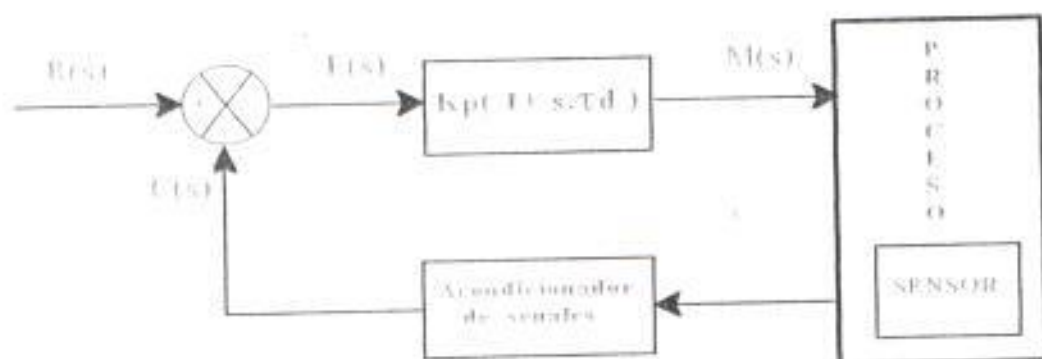


Figura 5.10 Controlador Proporcional - Derivativo

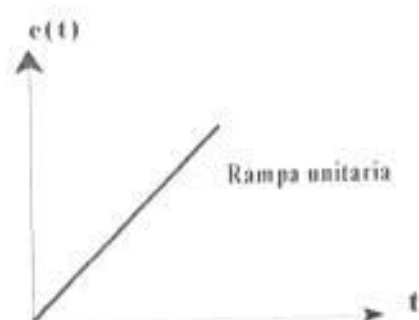


Figura 5.11 Entrada al Controlador PD

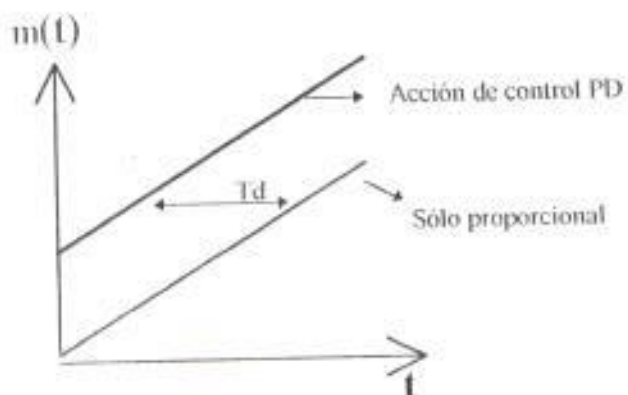


Figura 5.12 Salida del Controlador PD

De la ecuación 5.5 tenemos el siguiente circuito implementado con circuitos integrados (LM348), para un controlador proporcional - derivativo, como se muestra en la figura 5.13.

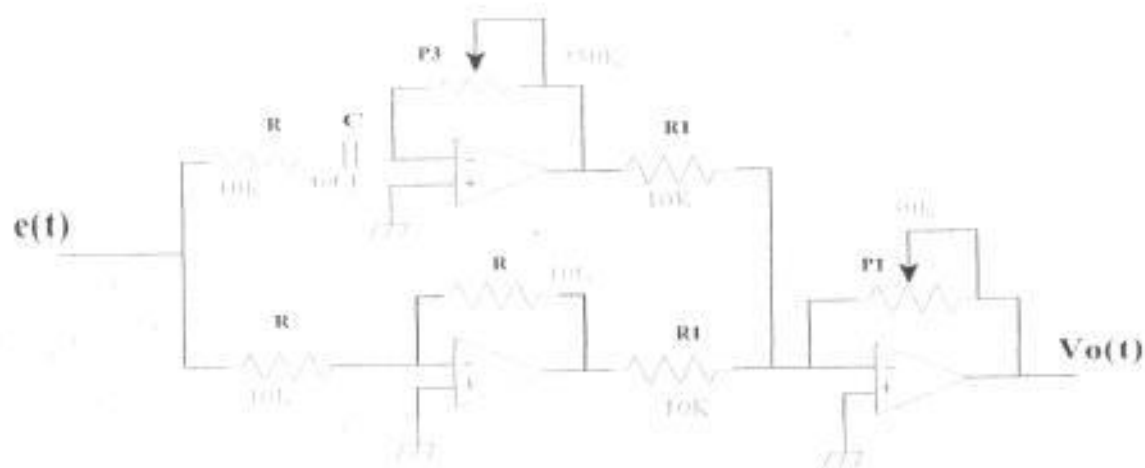


Figura 5.13 Diagrama esquemático del Controlador PD

$$V_o(t) = \frac{P1}{R1} e(t) + \frac{P1}{R1} P3 C \cdot de(t)/dt$$

$e(t)$ = señal de error

$V_o(t)$ = señal de salida

$P3 C$ = ganancia derivativa = T_d

$P1/R1$ = Ganancia proporcional = K_P

5.2.4. CONTROL PROPORCIONAL - INTEGRAL - DERIVATIVO

La combinación de los efectos de acción proporcional, integral y derivativa, se denomina acción de control proporcional - integral - derivativo. Esta combinación tiene las ventajas de cada una de las tres acciones de control individuales. El control PI se ajusta a la mayoría de las situaciones de control, pero hay un límite de que tan alta puede ajustarse la ganancia integral. Si se hace muy grande, el proceso puede caer en una oscilación prolongada después de un disturbio, por lo que si hay cambios muy rápidos en la carga y además hay retardos de tiempo entre la aplicación de la acción correctora y el apareamiento de los resultados de dicha acción en la variable medida, la solución puede ser un controlador PID.

Por ejemplo, si tenemos un error inicial grande y suprimiendo el control derivativo, consecuentemente al proceso le toma un tiempo largo en corregirse pero si aplicamos el control derivativo con un tiempo derivativo pequeño, el controlador introduce una sobrecorrección porque ha reconocido que la rápida tasa de cambio inicial presagia un gran cambio total si no se introducen pasos especiales de corrección. Debido a que el error inicial es pequeño tempranamente se recobra el valor de referencia, si se aumenta la contribución derivativa aumentando la constante de tiempo de derivación, entonces el error inicial es aún más pequeño debido a que se ha proporcionado más sobrecorrección inicial, de donde el

tiempo para recuperar y estabilizarse al valor de referencia es también más pequeño.

Así como hay un límite para la ganancia integral, hay un límite en el incremento para la ganancia derivativa. Pueden ocurrir oscilaciones prolongadas alrededor del valor de referencia si se introduce mucho control derivativo. La ecuación que gobierna a un controlador PID es como se muestra:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \cdot T_d \cdot d e(t) / dt$$

y la función de transferencia en términos de Laplace es:

$$\frac{M(s)}{E(s)} = K_p [1 + 1/T_i s + T_d s]$$

donde K_p es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo. En la figura 5.14 se puede ver el diagrama de bloques de un controlador proporcional, integral y derivativo. Si $e(t)$ es una función rampa unitaria, como se ve en la figura 5.15, la salida del controlador $m(t)$ resulta como se muestra en la figura 5.16.

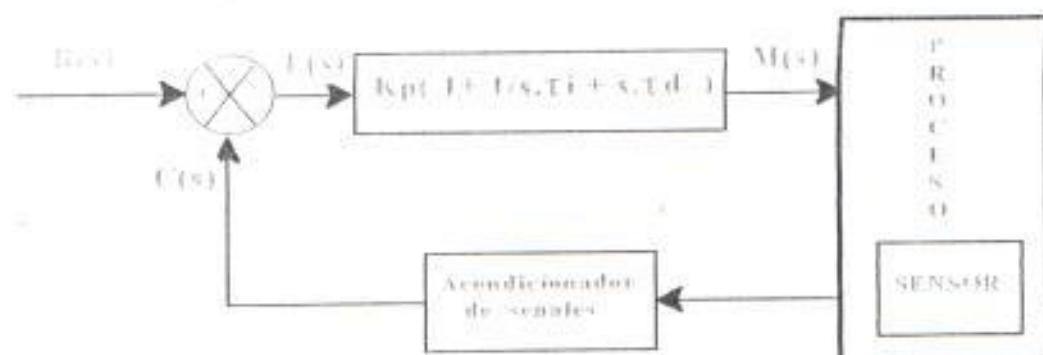


Figura 5. 14 Controlador PID

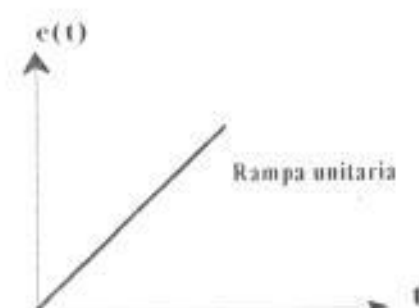


Figura 5. 15 Entrada al controlador PID

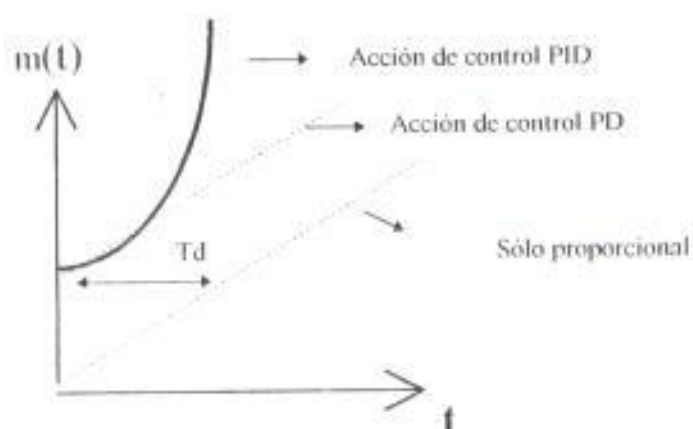


Figura 5. 16 Salida del controlador PID

$$V_o(t) = \frac{P1 \cdot e(t)}{R1} + \frac{P1}{R1} \cdot \frac{1}{C1 \cdot P2} \int_0^t e(t) \cdot dt + \frac{P1}{R1} \cdot C2 \cdot P3 \cdot de(t) / dt$$

en donde:

$V_o(t)$ = señal de salida de $m(t)$

$e(t)$ = señal de error

$P1 / R1 = K_P =$ ganancia proporcional

$C1 \cdot P2 = T_i =$ tiempo integral

$C2 \cdot P3 = T_d =$ tiempo derivativo

$K_I = 1/T_i =$ ganancia integral

entonces de la ecuación anterior tenemos:

$$V_o(t) = K_p \left[1 + \frac{1}{T_i} \int_0^t e(t) \cdot dt + T_d \cdot d e(t) / dt \right]$$

Con esta ecuación podemos implementar un controlador PID, con circuitos integrados (LM348) como se muestra en el capítulo diez

CAPITULO VI

ANALISIS DE ESTABILIDAD DEL SISTEMA

6.1. MÉTODOS DE ANÁLISIS DE ESTABILIDAD

Una característica muy importante del funcionamiento transitorio de cualquier sistema es la estabilidad. Un sistema de control está en equilibrio si la salida permanece en el mismo estado en ausencia de cualquier perturbación de entrada.

Un sistema de control lineal invariante en el tiempo, es estable si la salida regresa eventualmente a su estado de equilibrio cuando el sistema se somete a alguna perturbación, es inestable si, la salida oscila indefinidamente, o si la salida diverge sin límite de su estado de equilibrio, cuando el sistema sufre alguna perturbación. La respuesta a un desplazamiento, o condición inicial, dará como resultado, una

respuesta creciente, decreciente o neutral. Por la definición de estabilidad se deduce que un sistema lineal es estable si y solo si el valor absoluto de su respuesta a un impulso $\delta(t)$, integrada sobre un rango finito, es *finito*. La localización de los polos de la función de transferencia de un sistema en el plano s indica la respuesta transitoria resultante.

Los polos en la parte izquierda del plano s dan como resultado una respuesta decreciente para entradas de perturbación. Cuando los polos se encuentran en el eje $j\omega$ la respuesta es neutral y cuando se encuentran en el plano de la derecha la respuesta es creciente para una entrada de perturbación.

Existen varios métodos para analizar la estabilidad de un sistema:

- Método de la respuesta de frecuencia. Criterio de estabilidad de Nyquist
- Criterio de estabilidad de Routh Hurwitz
- Método del lugar geométrico de las raíces

El método utilizado en la presente tesis es el método del lugar geométrico de las raíces ya que la estabilidad y respuesta transitoria de un sistema de control de lazo cerrado estarán directamente relacionados con la localización en el plano s de las raíces de la ecuación característica del sistema. Además, a veces es necesario

ajustar uno o más parámetros del sistema para obtener las localizaciones adecuadas de las raíces. También mediante esta técnica gráfica se proporciona una buena medida de la sensibilidad de las raíces del sistema para una variación en el parámetro que se considera.

6.2 ESTABILIDAD DEL SISTEMA

En esta sección se procederá a analizar la estabilidad del sistema de control de posición. Para ello nos valdremos del diagrama de bloques de la figura 6.1.

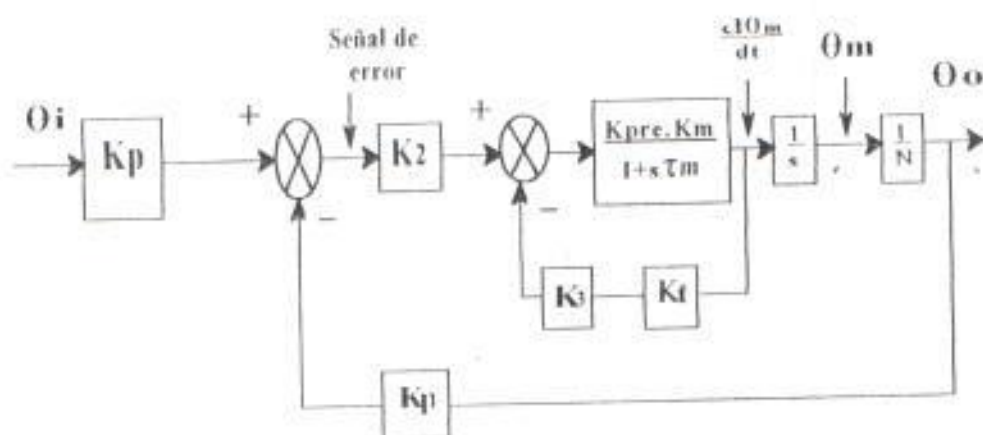


Figura 6.1 Diagrama de bloques del sistema de Control de Posición

donde se tienen los siguientes datos

K_{pre}	=	20
K_m	=	$16.3 \frac{\text{rad/seg}}{\text{volt}}$
τ_m	=	0.11 seg
K_t	=	$0.0179 \frac{\text{volt/rad}}{\text{seg}}$
N	=	40
K_p	=	5.7 volt/rad

Reduciendo el gráfico anterior tenemos:

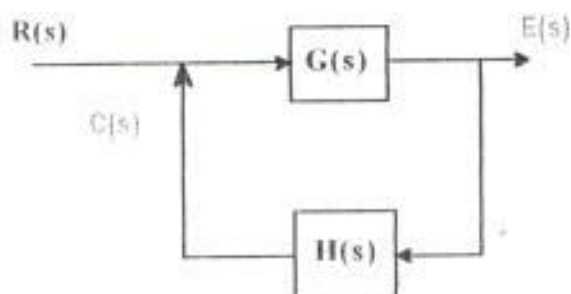


Figura 6. 2 Diagrama de bloques reducido

$$G(s) = \frac{K_2 K_{pre} K_m}{sN + s^2 N \tau_m + K_2 K_3 K_{pre} K_t s N}$$

$$H(s) = 1$$

dé donde la ecuación característica es $1 + G(s)H(s) = 0$

$$1 + G(s)H(s) = 1 + \frac{K_2 K_{pre} K_m K_p}{sN + s^2 N \tau_m + K_2 K_3 K_{pre} K_t s N} = 0$$

$$1 + G(s)H(s) = 1 + \frac{K_2 K_{pre} K_m K_p / N \tau_m}{s^2 + s \left[\frac{K_3 K_t K_{pre} K_m}{\tau_m} \right]} = 0$$

$$1 + G(s)H(s) = 1 + \frac{K_2 K_{pre} K_m K_p / N \tau_m}{s \left\{ s + \left[\frac{1 + K_3 K_t K_{pre} K_m}{\tau_m} \right] \right\}} = 0$$

Reemplazando valores tenemos:

$$1 + G(s)H(s) = 1 + \frac{K_2 (20 \times 16.3 \times 5.7)}{40 \times 0.11} = 0$$

$$s \left\{ s + \left[\frac{1}{0.11} + \frac{K_3 \times 0.0179 \times 20 \times 16.3}{0.11} \right] \right\}$$

$$= 1 + \frac{K_2 (422.3)}{s \left\{ s + [9.1 + K_3 (53)] \right\}} = 0$$

Si $K_3 = 0$ % y $K = K_2(422.3)$, tenemos:

Considerando a K_2 como parámetro tenemos:

$$1 + G(s)H(s) = 1 + \frac{K}{s[s + 9.1]} = 0$$

de donde tenemos las siguientes raíces:

$$\begin{aligned} s &= 0 \\ s &= -9.1 \end{aligned}$$

entonces para diferentes valores de K_2 existirá siempre un polo en el origen y otro en -9.1 y buscarán sus ceros en el infinito, saliendo por -4.55 .

Si $K_2 = 1$

$$s^2 + s(9.1) + 422.3 = 0$$

$$\Rightarrow s_1 = -4.55 + j20$$

$$s_2 = -4.55 - j20$$

Si $K_2 = 10$

$$\Rightarrow s_1 = -4.55 + j65$$

$$s_2 = -4.55 - j65$$

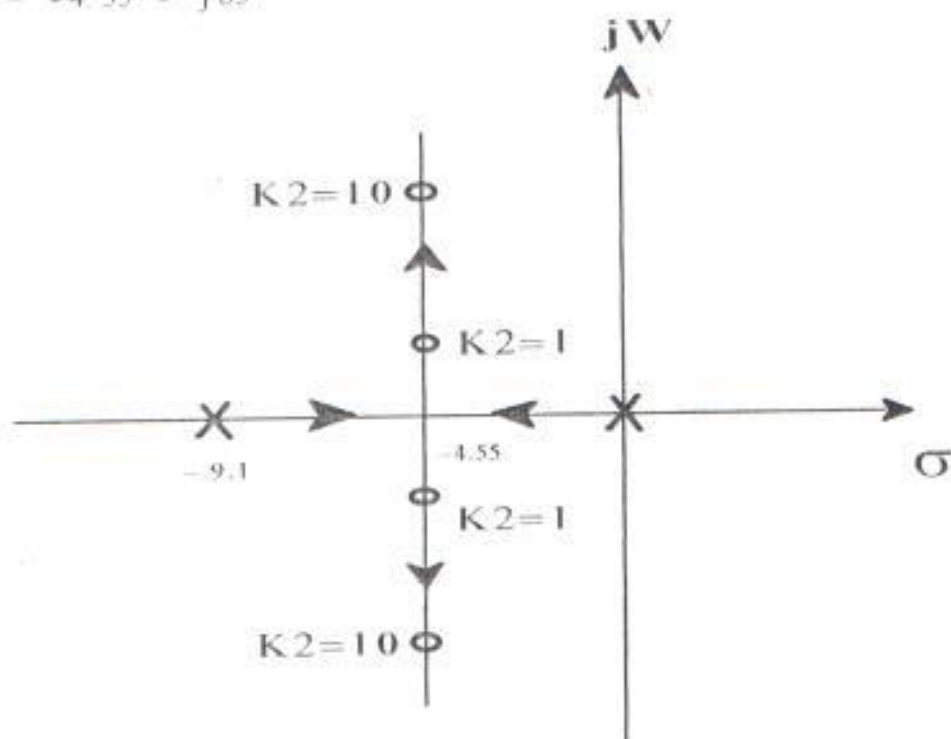


Figura 6.3 Ubicación de las raíces para variaciones de K_2

Ahora si variamos K_3 manteniendo constante K_2 entonces tendremos:

Si $K_3 = 1$

$$\Rightarrow 1 + \frac{K}{s[s + (9.1 + K_3, 53)]} = 0$$

$$1 + \frac{K}{s(s + 62.1)} = 0$$

⇒ las raíces serán:

$$s = 0$$

$$s = -62.1$$

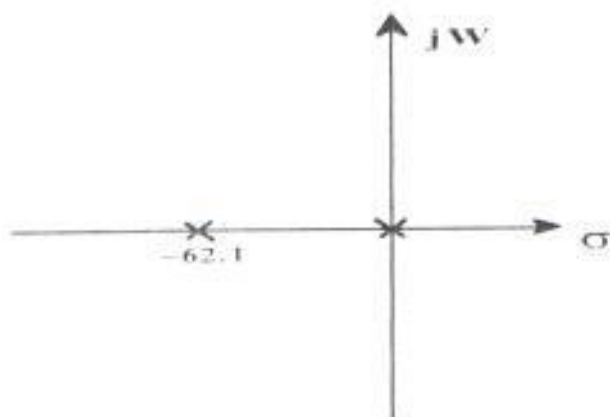


Figura 6.4 Ubicación de las raíces con $K_3=1$, K_2 constante

Si $K_3 = 5$

$$\Rightarrow 1 + \frac{K}{s[s + (9.1 + 265)]} = 0$$

$$1 + \frac{K}{s(s + 274.1)} = 0$$

y las raíces son:

$$s = 0$$

$$s = -274.1$$

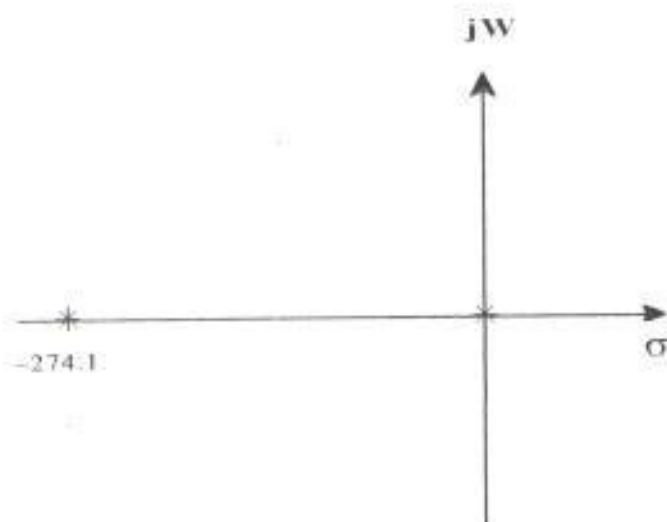


Figura 6.5 Ubicación de las raíces con $K_3=5$, K_2 constante

6.3. CONDICIONES EXTREMAS DE FUNCIONAMIENTO

A partir del análisis de estabilidad que hemos realizado podemos deducir lo siguiente:

En el sistema de control de posición de lazo cerrado notamos que tomando a K_2 y K_3 como parámetros, siempre va a existir un polo en el origen, por lo tanto el sistema es neutral, claro que si aumentamos K_2 el sistema es oscilante.

Cuando aumentamos K_3 el polo diferente de cero se desplaza cada vez más hacia la izquierda, por lo tanto el sistema se hace más estable, ya que para $K_3 = 0$, tenemos un polo en -9.1 pero para $K_3 = 5$, tenemos un polo en -274.1 , por lo tanto es relativamente más estable. Con el aumento de K_2 el sistema es más oscilante por lo tanto es relativamente menos estable; mientras mayor es el valor de K_3 el sistema es menos oscilante, es decir K_3 elimina las oscilaciones producidas con el aumento de K_2 . Claro que existirá valores de K_2 y K_3 que hagan que el sistema sea el más estable, ya que en un control de posición se requiere que el sistema no tenga oscilaciones, para el presente caso valores de K_3 grandes y K_2 pequeños, son los adecuados.

CAPITULO VII

DISEÑO DE LA TARJETA DE INTERFAZ DE DATOS

7.1 TRANSMISION Y ADQUISICION DE DATOS

Antes de iniciar la explicación de cada uno de los componentes del circuito de interfaz de datos se debe tener claro el concepto de *interface*. Una *interface* es la interconexión de dos dispositivos, circuitos o sistemas, eléctricamente incompatibles de modo que entre ellos pueda transferirse información de manera eficiente. En este capítulo nos referiremos a una interface constituida por puertos que van a interactuar con cada uno de los convertidores, etapas acondicionadoras y demás dispositivos que manejarán las señales de control digital ya sea de velocidad o de posición.

El sistema computador está diseñado para incluir dispositivos de I/O programables. Un *puerto programable* de I/O es un circuito que puede funcionar ya sea como un

puerto de entrada o un puerto de salida, con o sin "handshaking". Por consiguiente los dispositivos programables hacen que el sistema sea flexible y adaptable a los diferentes requerimientos.

Los dispositivos utilizados en nuestro diseño son la *Interface Programable 8255* y el *puerto IO 8212*.

7.1.1. INTERFACE PROGRAMABLE 8255A

El 8255A es una interfaz programable de periféricos (PPI). Su función es la de un componente de entrada y salida de propósito general para interfacear equipos periféricos al bus del sistema microcomputador.

La configuración funcional del 8255A es programada por software de tal forma que ninguna lógica externa es necesaria para interfacear los dispositivos periféricos. Esta interfaz es utilizada debido a su gran utilidad en sistemas en los cuales se requiere del manejo de varias líneas de entrada y salida. En la figura 7.1 se muestra la distribución de pines de este PPI, el cual contiene 3 puertos programables de entrada y salida de 8 bits cada uno. En el Apéndice B se puede hallar mayor información técnica del 8255A. Podemos observar que 24 de los 40 pines del 8255A son líneas de entrada y salida. Específicamente, PA0-PA7

corresponden al puerto A, PBO-PB7 al puerto B y PCO-PC7 corresponden al puerto C. Las 8 líneas del puerto C se dividen en dos grupos iguales, cada uno de cuatro líneas (PC7-PC4 y PC3-PC0). Estos grupos de líneas pueden trabajar en combinación con los puertos A y B o como dos puertos individuales de 4 bits.

En su estructura interna los registros se encuentran divididos en dos grandes grupos: el A y el B, cada uno de los cuales tiene su propio elemento de control.

El grupo A está compuesto por el puerto A y los cuatro bits más significativos del puerto C, mientras que el grupo B por el puerto B y los cuatro bits menos significativos del puerto C.

El sentido (entrada, salida o bidireccional) y la función (datos o control) de las líneas de todos los puertos se programan durante la operación normal del dispositivo mediante una palabra de control de 8 bits que envía la unidad central de proceso (CPU) al PPI.

Inicialmente, el 8255A dispone de 4 registros, 3 de los cuales se dedican a almacenar la información que entra o sale de los puertos A, B y C. El cuarto registro se denomina registro de control, y, como su nombre lo indica, se dedica a realizar las funciones de control, programando este registro se configuran los

grupos A y B, definiéndose el comportamiento de los puertos y , por consiguiente, el funcionamiento general del dispositivo.

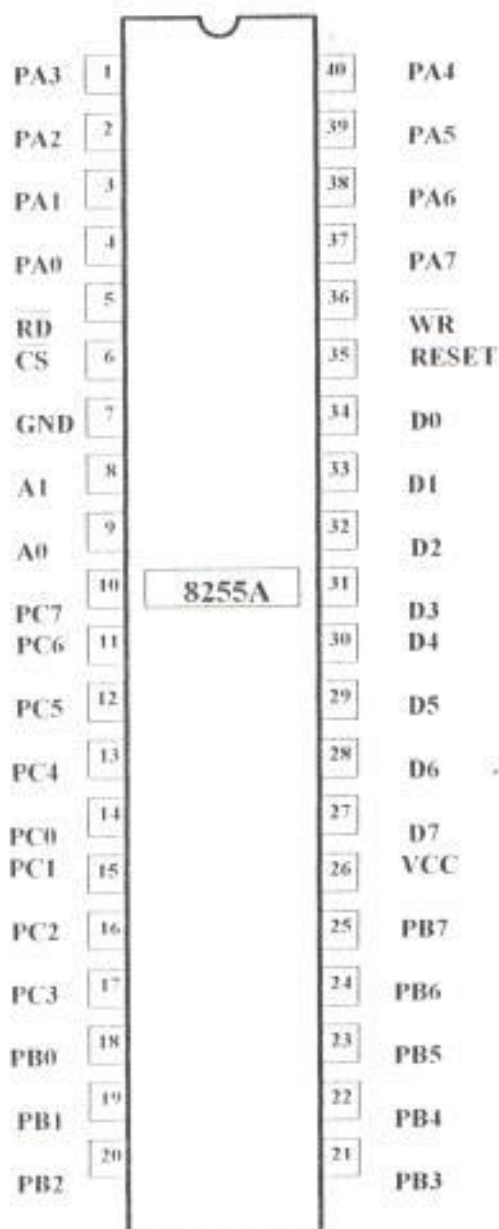


Figura 7.1 Distribución de pines del 8255A

FUNCIONES DE LOS PINES

Además de las 24 líneas de entrada/salida asociadas a los puertos A,B y C , la función de las líneas restantes del 8255 se puede resumir en los siguientes términos:

D7 A D0 - Líneas de datos. Comunican el 8255 con el bus bidireccional de datos del sistema microprocesador.

—
WR (WRITE).- Activa en bajo. Permite que la CPU realice una operación de escritura sobre el PPI.

—
RD (READ).- Activa en bajo. Permite que la CPU realice una operación o ciclo de lectura sobre el PPI.

—
CS (CHIP SELECT) - Activa en bajo se destina para la selección del integrado. El 8255 se comunica con la CPU cuando CS=0 y se aísla cuando CS=1.

RESET.- Activa en alto. Inicializa el PPI y borra todos los registros internos, incluyendo el de control, configurando como entradas todas las líneas de los puertos.

A0 y A1.- Estas señales en combinación con RD y WR, seleccionan el puerto o registro del 8255 sobre el cual se va a realizar una operación de lectura o escritura, normalmente, se conectan a las líneas A0 y A1 del bus de direcciones del sistema.

Para el diseño de la tarjeta de interfaz de datos se ha visto conveniente utilizar el funcionamiento del PPI en modo 0 (sin handshaking) ya que se acomoda a nuestro requerimiento de que cualquiera de los puertos A, B ó C puede trabajar como entrada ó como salida. La configuración utilizada para este modo 0 es la siguiente:

PUERTO A = SALIDA
PUERTO B = ENTRADA
PUERTO C = ENTRADA

Debido a que en la interfaz programable 8255 no existe una configuración en la cual 6 líneas de un puerto (5 del decodificador de eje y una señal de EOC proveniente del ADC0808) funcionen como entradas y una línea del mismo puerto (START) funcione como salida, se ha utilizado todo el puerto C como entrada en el cual ingresen los 5 bits del codificador de eje y el bit EOC del ADC,

sin utilizar los dos bits más significativos del puerto C. Aunque las 6 líneas de entrada fueron conectadas acertadamente, todavía persistía el problema de la señal START.

Por tal motivo se utilizó un puerto de entrada/salida para enviar la señal START hacia el ADC0808. Se escogió el puerto I/O 8212 para tal propósito.

7.1.2. PUERTO DE ENTRADA / SALIDA 8212

El 8212 es un registro de tres estados de ocho bits con lógica de control y selección que puede utilizarse como puerto de entrada o de salida en sistemas basados en el microprocesador 8086. En la figura 7.2 se muestra la distribución de pines del 8212. Este registro de almacenamiento proporciona un método sencillo de implementar un puerto I/O por hardware. En el apéndice B correspondiente al Manual Técnico se puede encontrar información más detallada.

FUNCIONES DE LOS PINES

El 8212 consta de ocho flip-flops tipo D (entradas) cada uno conectado a un registro de almacenamiento de 3 estados (buffer tri-state) no inversor de salida. En adición, el dispositivo tiene dos entradas de selección (DS1 y DS2), una

entrada de Modo (MD) , una entrada estroboscópica o de habilitación (STB) y una entrada de Clear o de borrado (CLR) .

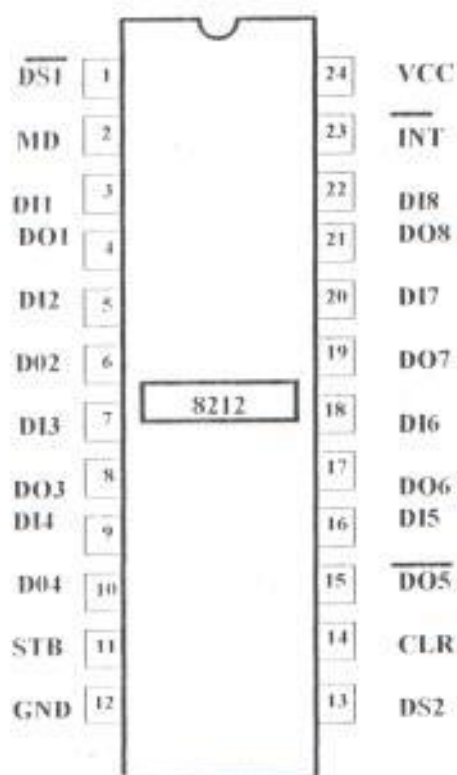


Figura 7. 2 Distribución de pines del 8212

El fabricante recomienda el esquema expuesto en la figura 7.3 para usar el circuito 8212 como un puerto de salida, que es como lo utilizaremos en el presente trabajo de tesis.

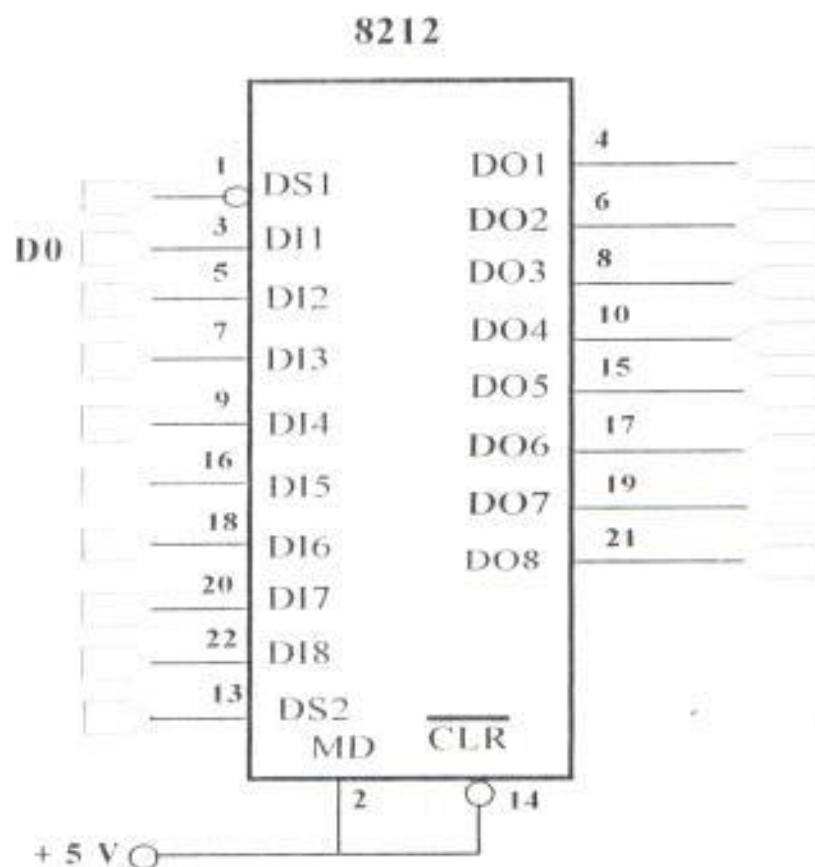


Figura 7.3 Aplicación como puerto de salida del 8212

La función de cada uno de los 24 pines puede resumirse en los siguientes términos

Las líneas **DI1** (pin 3) hasta **DI8** (pin 22) son las entradas de datos.

Las líneas **DO1** (pin4) hasta **DO8** (pin 21) son las salidas de datos.

La entrada **CLR** (clear , pin 14) es la línea de inicialización . Cuando recibe un bajo, borra todos los datos de los flip-flops , llevándolos al estado de reset .

Las entradas **DS1** y **DS2** en nuestro caso son usadas para la selección del dispositivo. Por ejemplo cuando DS1 está en nivel lógico '0' y DS2 está en nivel lógico '1' (DS1, DS2) el dispositivo es seleccionado. En este estado los buffers de salida del circuito son habilitados y actualizados con los niveles lógicos presentes en la entrada.

La entrada **MD** (pin2) es la línea de modo de entrada/salida. Controla el estado del buffer de salida y determina la procedencia de la señal de reloj de entrada para el enclavamiento de los datos. Cuando MD se encuentra en nivel lógico '1' (modo de salida) los buffers de salida son habilitados y la fuente de reloj para el enclavamiento de datos proviene del dispositivo lógico de selección (DS1 , DS2).

En resumen, cuando (DS1, DS2) sea igual a un nivel lógico '0' y MD sea igual a un nivel lógico 1 se producirá el enclavamiento de los datos en la salida. Debido a que la señal IOW realiza una acción de escritura sobre los dispositivos

de entrada y salida, es decir, produce el dato válido D0 proveniente del microprocesador hacia el 8212 y además aprovechando los niveles lógicos compatibles que ambas señales tienen, se procedió a conectar la señal de control de la barra IOW a la entrada DS1.

También como el puerto de I/O solamente enviará una señal de inicio de conversión START al ADC0808, se conectó el bit D0 de la barra de datos a la entrada DI1 del 8212. Como se justificará más adelante, la salida del decodificador de direcciones Y5, se la conectó a un inversor 74LS04, y la salida de éste, a la entrada DS2. El diagrama de conexión final del 8212 es el que se muestra en la figura 7.4.

7.1.3. DECODIFICADOR DE DIRECCIONES

Antes de entrar a analizar el circuito decodificador de direcciones es conveniente mencionar algunos detalles del espacio de direcciones del microcomputador que ayudarán a comprender mejor nuestro circuito.

La arquitectura del microprocesador 8086 posee un espacio de direcciones de puertos de 65536 direcciones. El computador no utiliza todo este rango de

direcciones sino solamente los 10 bits menos significativos (0 a 9) que son usados para decodificar las direcciones de un dispositivo.

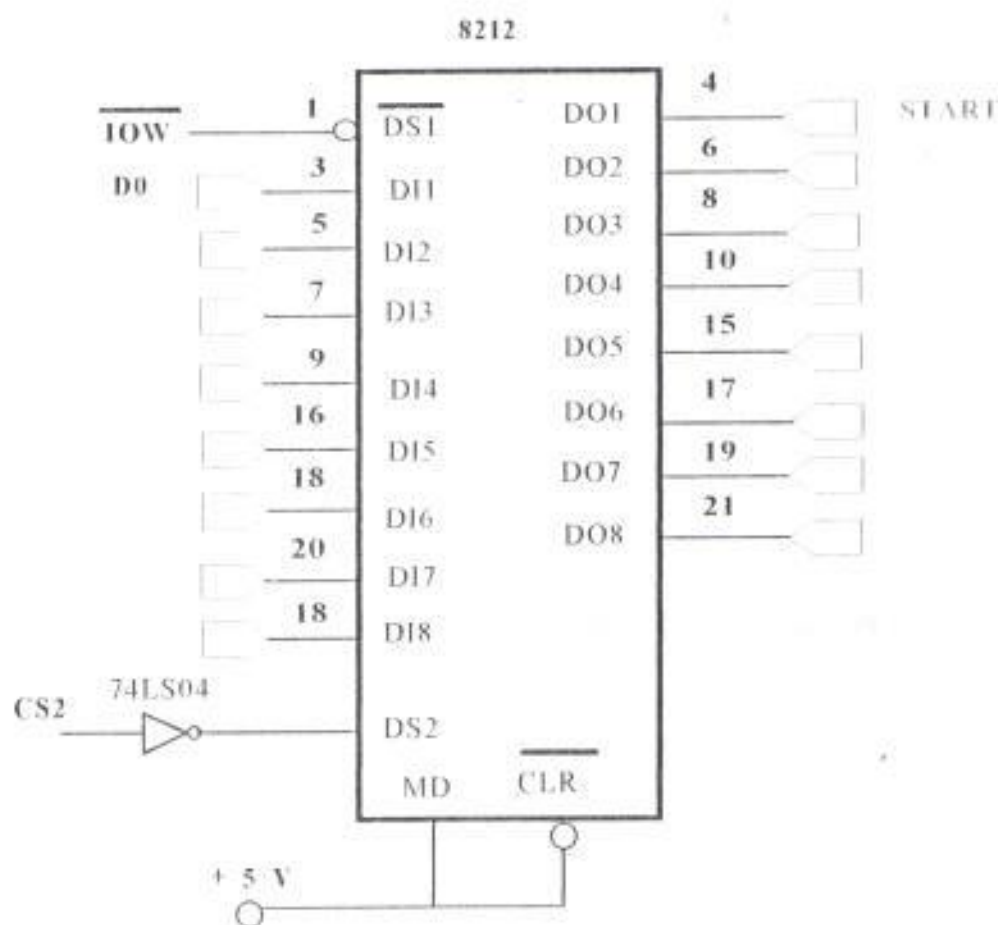


Figura 7.4 Puerto de Salida final

En la computadora personal PC se utiliza una parte del mapa de I/O para funciones específicas. Se suele reservar para el sistema de la computadora el

espacio de direcciones entre los puertos 0000H y 3FFH. Los puertos de entrada/salida desde 0400H hasta FFFFH suelen estar disponibles para las aplicaciones del usuario. Este es el rango de direcciones al cual hemos recurrido para direccionar la tarjeta de interfaz de datos. La figura 7.5 ilustra el uso del campo de direcciones de I/O en el PC.

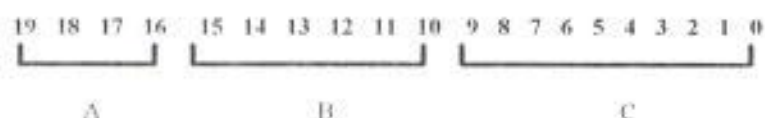


Figura 7.5 Campo de Direcciones de I/O de una PC

- A : rango no utilizado por la arquitectura de direccionamiento de puertos del 8086
- B : rango de operación de instrucciones IN y OUT, pero no es usado por el PC
- C : rango de dirección reservado para la computadora

Por lo tanto, se ha seleccionado el rango a partir de la dirección 500H (en hexadecimal) , con el cual se accederá a la dirección principal o base de la tarjeta

de interfaz, o sea la correspondiente al puerto A del 8255A y la dirección 540H (en hexadecimal) con el cual se accederá a la dirección del puerto de salida 8212. La siguiente tabla ilustra lo expresado:

Tabla 7.1
Dispositivos de I/O

Dirección	Nombre de señal	Dispositivo Seleccionado
500H	CS1	Puerto de I/O 8255A
540H	CS2	Puerto de Salida 8212

Como se mencionó anteriormente, era necesario tener una tarjeta de interfaz de datos para poder seleccionar las direcciones de los dispositivos utilizados en el sistema. Por lo que dentro de ese diseño cumple una función muy importante un circuito que nos ayuda a seleccionar debidamente tales dispositivos. Este circuito es el Decodificador de Direcciones 74ALS138.

El Decodificador de Direcciones es un circuito muy común que provee una sola salida activa de ocho posibles activadas en bajo. Al recibir un código binario de

tres bits se activa su entrada, generando una salida única equivalente por cada valor de código de entrada. La distribución de pines del circuito integrado 74ALS138 se muestra en la figura 7.6. Este circuito es utilizado para decodificar las líneas de dirección A8, A7 y A6 de tal forma que pueda seleccionar los dispositivos 8255A y 8212. En la tabla 7.2 se muestra el funcionamiento de selección y los valores de verdad del decodificador 74ALS138.

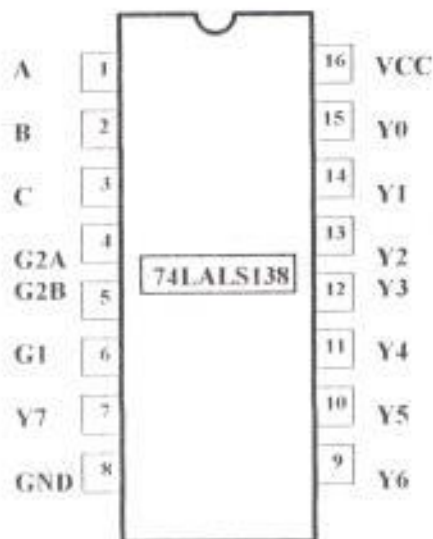


Figura 7.6 Distribución de pines del 74ALS138

Analizando la tabla de verdad nos damos cuenta que la salida del circuito 74ALS138 es una salida activa en estado bajo. Pero para que esto ocurra deben estar activas las tres entradas de habilitación, que corresponden a G2A, G2B y G1; para ello, las entradas G2A y G2B

Las líneas de dirección A6 y A8 se conectaron a las entradas de selección A y C respectivamente. La línea de dirección A7 se conectó primeramente a un juego de puertas NOR junto con las líneas A2, A3, A4 y A5, con el propósito de asegurar un normal desenvolvimiento de la tarjeta de interfaz de datos y el computador en relación al acceso de una dirección de memoria. Por lo que a la entrada de selección C del decodificador se ha ingresado la señal AIN la cual es la resultante del par de puertas NOR. En la siguiente tabla se ilustra lo expresado anteriormente:

Tabla 7.3

**Estados del bus de dirección para las salidas de
habilitación del 74ALS138**

A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Salida Habilitada	Dirección	Dispositivo Seleccionado
0	1	0	1	0	0	0	0	0	0	0	0	Y4	500H	8255A
0	1	0	1	0	1	0	0	0	0	0	0	Y5	540H	8212

Podemos decir entonces que cuando se realice un ciclo de escritura a la dirección 500H, solamente la salida Y4 se encontrará activa en estado bajo, de la misma forma, cuando se realice una acción de escritura a la dirección 540H, solamente la salida Y5 será habilitada en bajo. Cabe mencionar que con todas las

expresiones verdadas se ha podido justificar el diagrama de conexión final del 8212 mostrado en la sección anterior. Mediante el siguiente gráfico se visualiza el circuito decodificador de direcciones final:

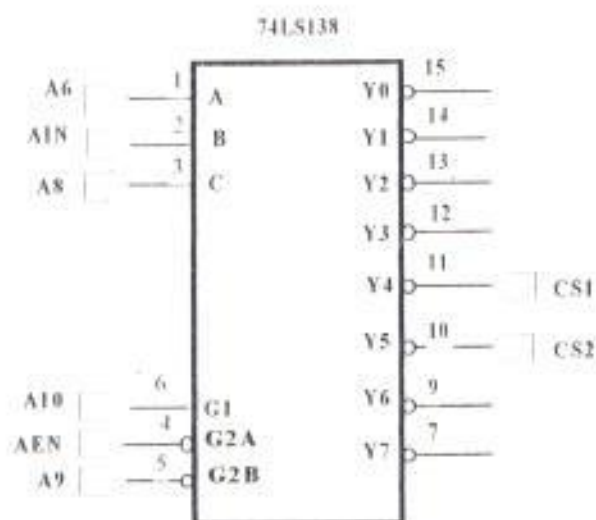


Figura 7.7 Circuito decodificador de direcciones para los dispositivos de I/O

7.2. CONVERSION DE DATOS

Los convertidores se utilizan para interconectar el microprocesador con un sistema analógico. En nuestro sistema éstos serán utilizados para convertir la velocidad y posición analógicos del motor.

7.2.1. CONVERTIDOR DIGITAL - ANALÓGICO DAC 0808

Uno de los convertidores D/A (digital - analógico) más utilizados es el de 8 bits DAC0808. Este es un convertidor de 8 bits cuyo tiempo de estabilización es de 150 nseg. La disipación de potencia es algo menor a 33 mW cuando se trabaja el circuito con alimentación de $\pm 5V$.

En la figura 7. 8 se muestra la distribución de pines de este chip. Observe los siguientes puntos destacados con respecto a este convertidor:

- Está compuesto por una red R-2R
- El voltaje de referencia aplicado a esta referencia se puede alterar externamente por medio de los terminales V_{ref+} (pin14) y V_{ref-} (pin 15)
- El DAC 0808 entrega por el pin 4 una corriente I_o . Para convertir esta corriente a voltaje hay que colocar a la salida un sistema como un amplificador operacional.
- Este convertidor puede alimentarse con fuentes duales desde $\pm 5v$ hasta 18 V.

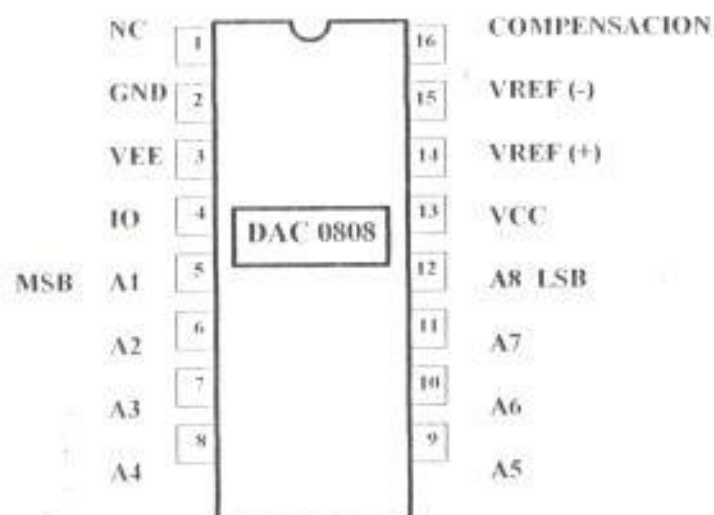


Figura 7.8 Distribución de pines del DAC 0808

7.2.2. CONVERTIDOR ANALOGICO - DIGITAL ADC 0808

El ADC 0808 es un convertidor A/D de 8bits (la señal analoga se convierte en una palabra de 8 bits), que tiene 8 entradas analógicas. El ADC 0808 contiene 28 pines de los cuales 8 corresponden a las señales de entrada; este dispositivo tiene un selector (multiplexor) de tres líneas digitales, que permite escoger la señal de entrada a convertir. La distribución de pines del integrado ADC0808 se ilustra en la figura 7.9.

Para cada una de las 8 combinaciones se utiliza el canal correspondiente a la salida del selector.

Esta salida está conectada a la entrada del convertidor analógico propiamente dicho. El ADC 0808 utiliza la técnica de las aproximaciones sucesivas para realizar la conversión. Después de seleccionar el canal (3 bits) y dar la señal de START, el circuito emplea 100 microsegundos para completar el proceso, cuando esto ocurre, coloca los 8 bits de la palabra digital resultante en un registro tri-state de almacenamiento en la salida y emite la señal EOC (fin de conversión) para que se entere el sistema encargado de recoger los datos.

La referencia VREF+ y VREF- permiten calibrar el rango de conversión. Por ejemplo. Si VREF+ es 5 V y VREF- es igual a 0V, la palabra digital de salida variará entre 00(Hex) para 0 voltios y FF(Hex) para 5 voltios de señal de entrada.

La conversión de un dato analógico a digital ocurre de la siguiente manera. El dato es admitido por el ADC a través de algunos de sus 8 canales de entrada siempre y cuando el circuito esté adecuadamente conectado y reciba por su pin 10 la señal de reloj.

Esta señal se recomienda que no supere los 500KHz para evitar un calentamiento indeseable en el circuito integrado al exigirle operar a una frecuencia mayor.

La selección del canal a través del cual se admite el dato se logra colocando en los pines 25,24 y 23 del ADC (ADDA, ADDB, ADDC respectivamente) la señales con el nivel lógico conveniente.

La conversión del dato se inicia cuando se activan las señales ALE (Habilita dirección del canal) y START (Iniciar Conversión) lo cual se hará por programa.

Al cabo de 100 microsegundos (tiempo de conversión del ADC), el proceso termina y el convertidor analógico- digital lo indica colocando en alto la señal EOC (pin 7).

La palabra digital de 8 bits aparece en el registro de salida, pines (D0 -D7) , si la entrada OE (salida habilitada) está activa en nivel alto.

La frecuencia del reloj con el cual trabaja el convertidor ADC 0808 se encuentra dentro del rango permitido en las especificaciones técnicas del circuito.

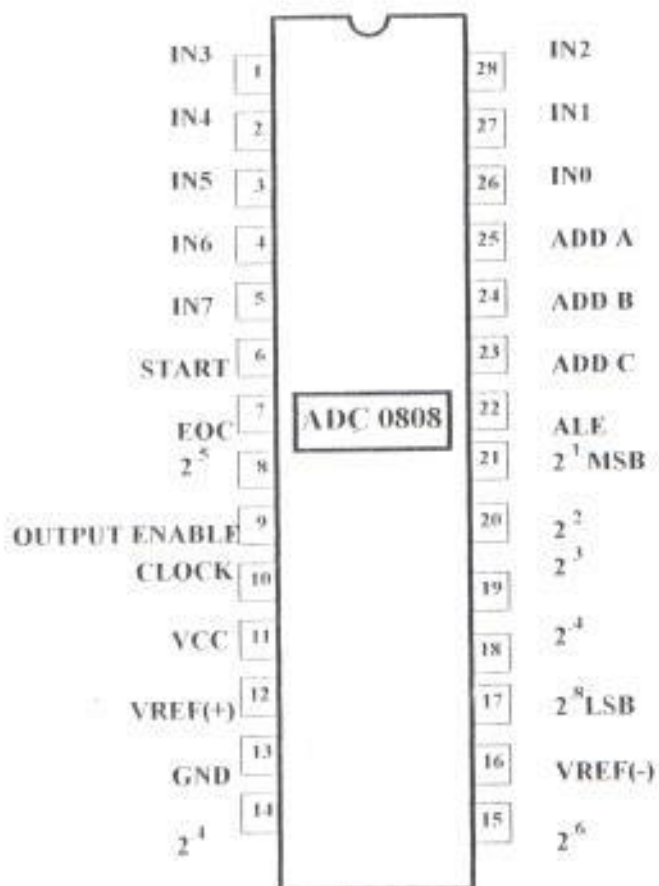


Figura 7.9 Distribución de pines del ADC 0808

CAPITULO VIII

CONTROL DIGITAL DE VELOCIDAD DE UN MOTOR DC

8.1. OBJETIVOS

En este capítulo se analizará el control de velocidad utilizando un sistema digital diseñado con puertos de entrada/salida, convertidores ADC y DAC y sensor de velocidad. Para esta aplicación el uso del microcomputador está orientado a controlar la realimentación mediante algoritmos apropiados. El sistema digital supera el funcionamiento de controles analógicos convencionales, ya que este es más flexible y adaptable a varios requerimientos.

8.2. DIAGRAMA DE BLOQUES

Para una mejor ilustración, el diseño del **sistema de control de velocidad** se lo ha

dividido en bloques, cuyo diagrama se presenta en la figura 8.1. El control se lo realiza mediante los siguientes pasos:

1. Medir la velocidad actual del motor (esta es suministrada por el tacómetro el cual nos da una señal de voltaje proporcional a la velocidad del motor).
2. Comparar la velocidad actual con la velocidad deseada para producir el error.
3. Con el error producido se genera la señal de control de acuerdo al algoritmo aplicado en el microcomputador.
4. La señal de control es transmitida al puerto I/O y al convertidor digital analógico, cuya salida es amplificada para manejar el servo-motor.

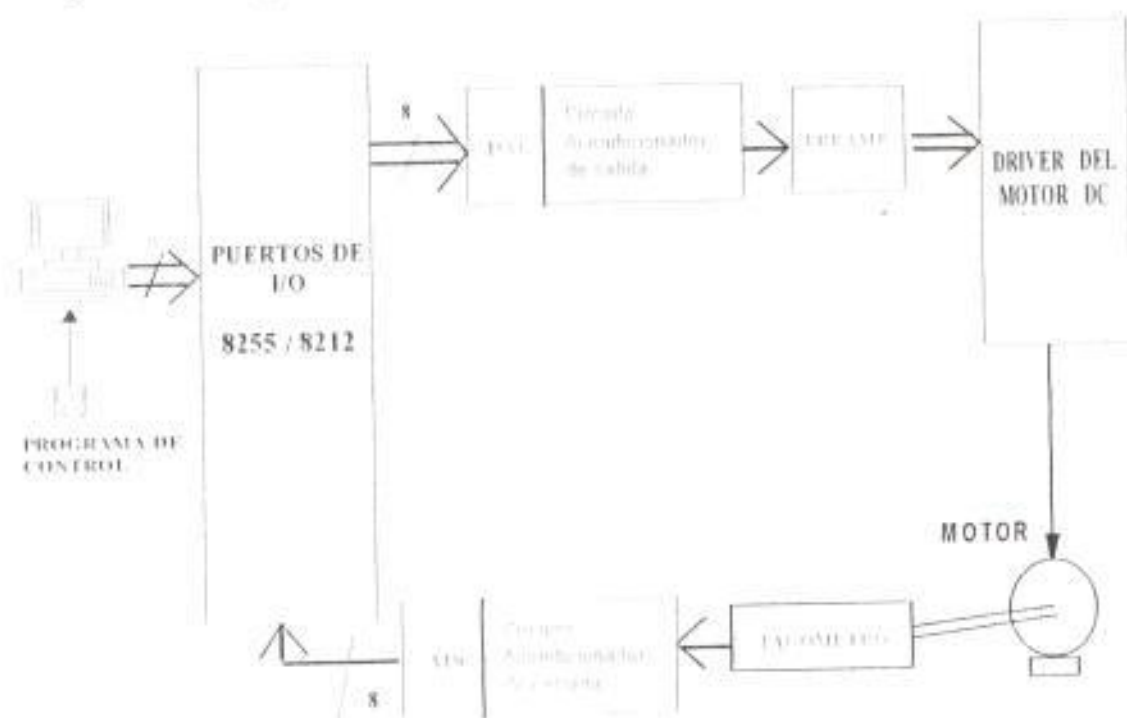


Figura 8.1 Sistema de Control Digital de Velocidad

8.3. FUNCIONAMIENTO DEL SISTEMA

El sistema se lo ha dividido en 3 partes para su mejor entendimiento:

1. **Interface Acondicionadora de Entrada**, la cual permite el ingreso de la variable controlada del proceso (velocidad) hacia el sistema controlador.
2. **Interface de Entrada/Salida con la Microcomputadora**, la cual controla y convierte los datos de I/O.
3. **Interface Acondicionadora de Salida**, la cual permite la utilización de la señal de salida de la microcomputadora (*variable manipulada*), en el proceso.

8.3.1. INTERFACE ACONDICIONADORA DE ENTRADA

Esta interface comprende un convertidor analógico - digital y un circuito acondicionador de entrada.

El circuito acondicionador de entrada, el cual se muestra en la figura 8.2, recibe la señal del tacómetro la cual genera una señal de voltaje proporcional a la velocidad en el eje de alta del motor. Esta señal es acondicionada mediante un circuito amplificador no inversor. Su funcionamiento es como sigue: La señal proveniente del tacómetro es reducida mediante R1 y R2 para luego ser amplificada a través de RF a un nivel de

voltaje compatible con la entrada del ADC0808. Además en este circuito mediante el potenciómetro de *zero* eliminamos el offset de tal manera que para una velocidad cero tengamos cero voltios a la entrada del ADC. En la siguiente sección se explica el funcionamiento del ADC en nuestro sistema.

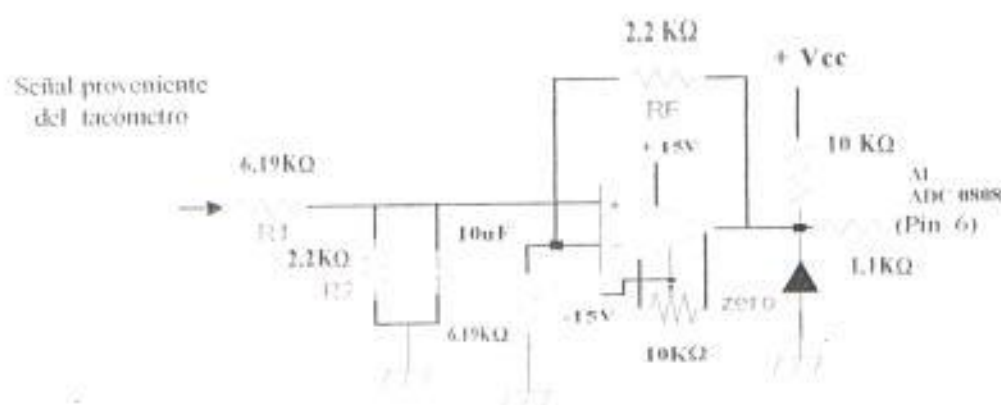


Figura 8.2 Circuito Acondicionador de Entrada

8.3.2. INTERFACE DE ENTRADA/SALIDA CON LA MICROCOMPUTADORA

En el diseño de esta *interface* se debe tener en cuenta que las señales provenientes del medio externo son de tipo analógico y que la misma permitirá entradas analógicas en un rango de 0-5 V. Bajo estas condiciones, la señal proveniente de la interface acondicionadora de entrada ingresa al multiplexor de 8 canales (IN6) interno del ADC 0808, tal como lo muestra la figura 8.3. La

selección del canal del mux a través del cual se admite el dato, se logra colocando en los pines 23,24 y 25 del ADC (A2,A1,A0) las señales con niveles lógicos alto, alto y bajo respectivamente, siendo A2 el bit más significativo.

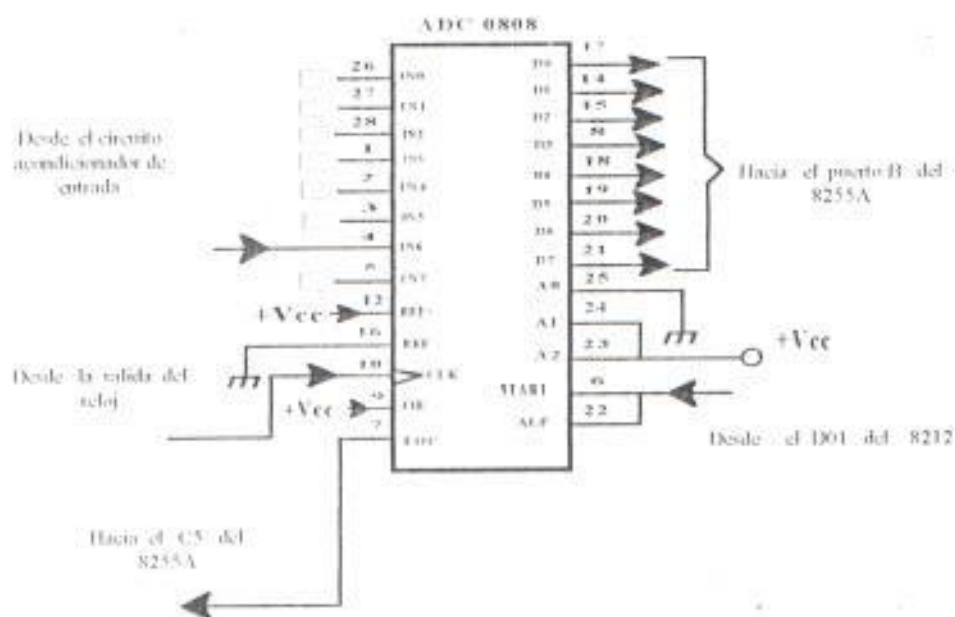


Figura 8.3 Circuito convertidor analógico - digital

La conversión del dato se inicia cuando se activan las señales *ALE* (habilita dirección del canal) y *START* (iniciar conversión), la cual se hará por programa.

Al cabo de 100 μ seg (tiempo de conversión del ADC) el proceso termina y el convertidor A/D lo indica colocando en *alto* la señal EOC (pin7). La palabra

digital de 8 bits aparece en el registro de salida (OUT0-OUT7 del ADC) si la entrada OE (output enable) está activa en nivel alto.

La señal **START** proviene de la salida *DOI* (pin 4) del registro de almacenamiento de entrada/salida del **8212**. Esta salida resulta de la captura del dato en la entrada *DI* (pin 3), cuya señal proviene del bus de datos del microcomputador (*D0*), el cual es manejado por software.

A la entrada de **DS1** se ha conectado la señal **IOW** (la cual se activa en bajo) proveniente del bus de control del microcomputador. La justificación de dicha conexión fue explicada en el capítulo anterior.

El código de entrada a los pines de selección del **74LS138** proviene de las líneas de dirección **A6**, **A7**, **A8**, siendo esta última el bit más significativo. Este código selecciona la salida **Y4** (habilitada en bajo), la cual corresponde a la señal **CS1** (pin 6 del **PPI**). Esta señal **CS1** es necesaria para la selección y operación del integrado **8255**. Este mismo código activa la salida **Y5** en estado alto, que corresponde a la señal **CS2**. Esta señal mediante una simple inversión es enviada a la entrada de selección **DS2** del circuito integrado **8212**. La señal **EOC** proveniente del **ADC 0808**, ingresa al pin 11 (**C5**) del **8255**, indicándole el

fin de conversión del dato. El 8255 permite entonces el ingreso de información al buffer de entrada, a través del puerto B (BO-B7, pines 18-25). Las conexiones del 8255A y del 8212 se muestran en el capítulo 10.

El ingreso de datos al puerto B del PPI se lo realiza mediante programación en lenguaje ensamblador utilizando el siguiente conjunto de instrucciones:

```
MOV DX,PORTB
MOV AL,DATO
IN DX,AL
```

Donde AL y DX son registros internos del CPU. DATO contiene la palabra digital de 8 bits convertida proveniente del ADC y PORTB contiene la dirección del puerto B (501H).

Este puerto B es seleccionado a través de las líneas de dirección A1 y A0 (que en este caso tendrían valores $A1 = 0$ y $A0 = 1$). A su vez estos datos se deben transferir a las líneas D0-D7 del bus del microcomputador a través de los pines (27-34) del 8255 con la siguiente línea de instrucción:

```
MOV AL,DATO
```

Se mencionó anteriormente que la señal EOC ingresaba a una línea del puerto C (C5). Las 5 líneas (C0-C4) del puerto C de la interface programable 8255 serán utilizadas para manejar el codificador en el control de posición. Mientras que las líneas C6 y C7 no serán usadas.

Las líneas IOR, IOW y GND son señales que van conectadas directamente al bus de control del microcomputador. Las líneas A0-A7 del puerto A son programadas como salidas por el registro de control del 8255. Estas líneas van conectadas a la entrada del DAC 0808 para su conversión. Esto se muestra en la figura 8.4.

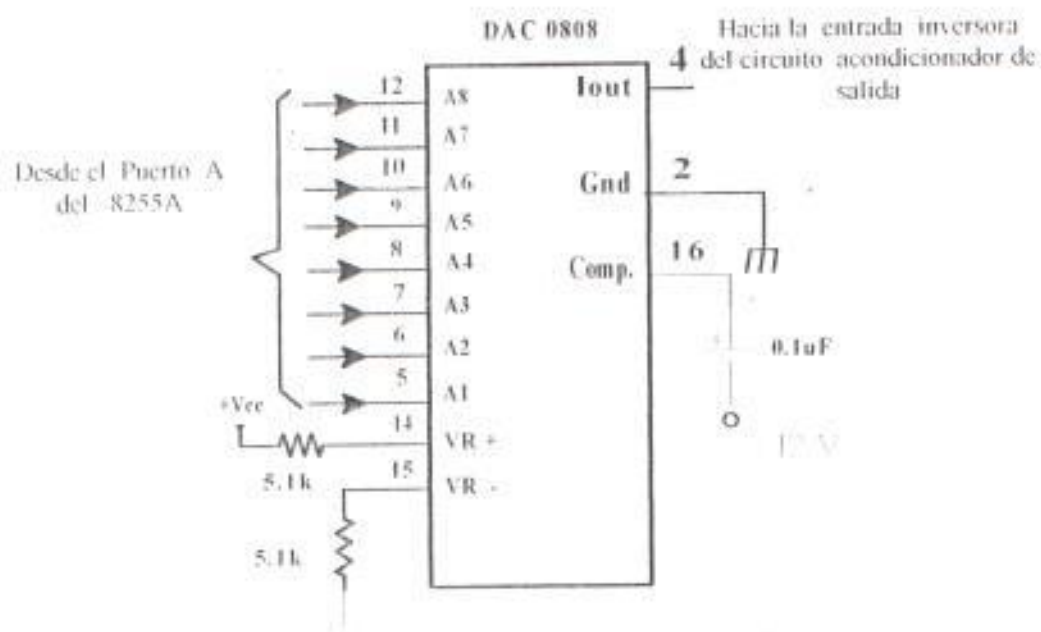


Figura 8.4 Circuito convertidor digital - analógico

El DAC 0808 posee además una entrada de referencia que está limitada a $V_{cc} = 5V$. La salida del DAC es una señal de corriente, que representa la información digital de posición o de velocidad, según sea la aplicación del controlador digital.

8.3.3. INTERFACE ACONDICIONADORA DE SALIDA

Esta interfaz comprende un convertidor digital - analógico y un circuito acondicionador de salida. El circuito acondicionador de salida recibe la señal del DAC (entre 0 y -5 V) cuyo voltaje es proporcional a la entrada digital del convertidor.

Este circuito se lo ha diseñado de tal manera que en su salida tengamos voltajes V_o (entre -1 y 1 V), los cuales son los adecuados para manejar el preamplificador. En la figura 8.5 se muestra este circuito. La ecuación que gobierna este circuito es:

$$V_o = - \left[\frac{R_F}{R_i} V_i + \frac{R_F}{R_x} (5 V) \right]$$

en donde R_i es una resistencia interna del DAC0808 la cual genera V_i .

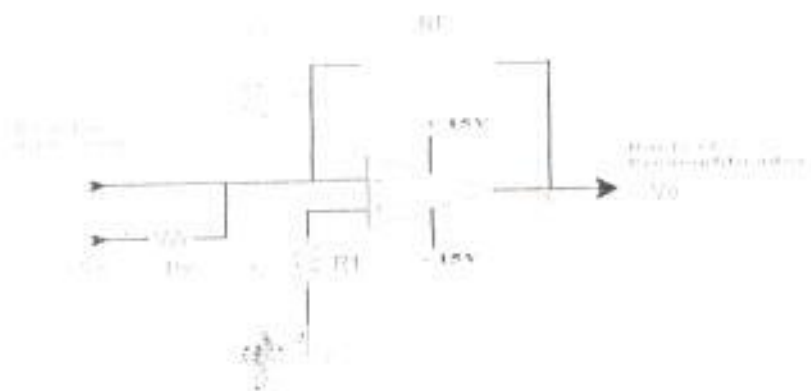


Figura 8.5 Circuito Acondicionador de Salida

8.4. ANALISIS DE RESPUESTA DEL SISTEMA

Nuestro sistema de control digital de velocidad es discreto en el tiempo, ya que los datos cambian solo a intervalos de tiempo T . Dicho intervalo está especificado por el tiempo durante el cual se realiza la medición de la variable controlada, la cual en este caso es la velocidad del motor. Siendo T iguala a 0.25mseg, correspondiente al periodo del reloj, que maneja el convertidor analógico - digital.

Para el motor de nuestro proyecto se determinó en base a un analisis de su respuesta de frecuencia en lazo cerrado, que el ancho de banda es 1.18 Hz correspondiente a un periodo de 0.85 mseg, por lo que trabajando con un T de 0.25 mseg, tendremos un buen margen de seguridad de obtener una información de velocidad muy fidedigna.

Para nuestro sistema la velocidad actual del motor (variable controlada) viene dado por la velocidad actual digital presente a la salida del ADC multiplicado por un factor de conversión igual a 30, por lo que en la pantalla mostrará un valor de velocidad múltiplo de 30.

Este factor de conversión se llegó a determinar experimentalmente, ya que como máximo podemos obtener 256 valores de velocidad, entonces si consideramos una velocidad máxima de operación de 8000 r.p.m en vacío, si dividimos este valor para 255 obtendremos exactamente 31.37 pero para mayor facilidad y agilidad en el cálculo determinamos el factor de 30 como el mejor valor, también hay que tomar en cuenta que la velocidad nominal del motor es 6000 r.p.m en vacío, entonces el valor máximo de velocidad mostrado en pantalla será el nominal el cual tendríamos un valor máximo en el DAC de 200 en estado estable. En el caso que tengamos por ejemplo un control PI, éste opera con un sobrenivel en el estado transiente, entonces la velocidad del motor sobrepasa los 600 r.p.m, pero como el DAC no se encuentra saturado (255), el sistema responderá eficientemente.

En la figura 8.6 se muestra las conexiones del banco para el control de velocidad con carga. Luego de esta explicación pasaremos directamente a realizar un análisis de la respuesta del sistema.

Análisis de la respuesta bajo un control proporcional:

Como se muestra en la figura 8.7, para una demanda de 3000 r.p.m y $K_P = 5$ se tiene un voltaje de salida en el tacogenerador de 2.3 voltios, equivalente a una velocidad actual de 1300 r.p.m, determinándose un error de velocidad en estado estacionario de 1700 r.p.m.

Si tenemos igual demanda pero $K_P = 8$ se tiene un voltaje en el tacogenerador de 4.4 voltios, equivalente a una velocidad actual de 2450 r.p.m, resultando un error en estado estacionario de 550 r.p.m estacionario de 480 r.p.m como se muestra en la figura 8.8.

En conclusión el control proporcional opera siempre con una desviación o señal de error actuante sea con carga o en vacío, que es lo que hace girar el motor en estado estable.

Cabe anotar que a mayor ganancia proporcional se obtiene menor error de velocidad en estado estable y mayor rapidez de respuesta, debido a que la corrección aplicada es mayor para una señal de error dada, pero el sistema se hace oscilante, como se analizó en el capítulo 5.

Análisis de la respuesta bajo un control proporcional – integral:

Análisis de la respuesta bajo un control proporcional - integral:

Ahora se analizará la respuesta del sistema para un controlador proporcional - integral. La figura 8.9 ilustra el caso de un controlador PI, siendo $K_P = 5$ y $K_I = 1/5$. De esta figura se observa que para una demanda de 3000 r.p.m se tiene un voltaje máximo en el tacogenerador de 6.4 voltios, consiguiéndose un error cero de velocidad en estado estable. La figura 8.10 muestra la respuesta del sistema para el caso de un controlador PI donde $K_P = 10$ y $K_I = 1/5$ con un sobrenivel en el tacogenerador de 6.2 voltios. De estas dos figuras deducimos entonces que un mayor K_P contribuye a disminuir el sobrenivel de la respuesta. Esto se debe a que el término proporcional también contribuye a disminuir el valor de salida del controlador un a vez que el valor de velocidad actual sobrepasa su respectivo valor de referencia.

En la figura 8.11 si mantenemos $K_P = 5$, con un aumento de $K_I = 1/2$, la respuesta del sistema se hace mas rápida, con un tiempo de estabilización mayor y con una sobrecorrección tanto positiva como negativa, o sea que si aumentamos K_I con un K_P constante el sistema es oscilante. En general se observa una disminución en magnitud tanto para la sobrecorrección positiva como negativa, con un K_P mayor lo que hace que el sistema se estabilice en menor tiempo. Además el

Como conclusión final este tipo de controlador opera con error en estado estacionario de cero.

Análisis de la respuesta bajo un control proporcional- derivativo:

Analizando un controlador proporcional – derivativo como se muestra en la figura 8.12 para un $K_P = 5$ y $K_D = 1/10$, se deduce que su comportamiento es similar a un control proporcional. Así para una demanda de 3000 r.p.m, en vacío tendremos un voltaje en el generador de 4 voltios correspondiente a una velocidad actual de 1860 r.p.m.

En la figura 8.13, $K_P = 5$ y $K_D = 10$, tendremos una velocidad actual de 1980 r.p.m con un tiempo de estabilización menor, con lo que deducimos que a mayor valor de K_D se tendrá una desviación y un tiempo de estabilización menor.

Este controlador no es eficiente como el control PI ya que el control diferencial se lo aplica a un controlador PID con el objeto de disminuir las oscilaciones producidas por la contribución integral.

Análisis de la respuesta bajo un control proporcional – integral – derivativo:

En la figura 8.14 se muestra la respuesta del sistema para un controlador proporcional – integral – derivativo, con $K_P = 5$, $K_I = 1/5$ y $K_D = 1/10$ comparado con la figura 8.9 se ve claramente que la contribución derivativa ayuda a disminuir el sobrenivel producido por la parte integral en el control PI, y además contribuye a disminuir la sobrecorrección tanto positiva como negativa, así como también a disminuir el tiempo de estabilización en la velocidad deseada de 3000 r.p.m, en vacío.

También en la figura 8.15 con $K_P = 3$, $K_I = 1/5$ y $K_D = 1/10$ tenemos una respuesta oscilante del sistema y en la figura 8.16 con $K_D = 20$ el tiempo de estabilización del sistema es menor, es decir el sistema es estable luego de una sobrecorrección producida por el aumento de K_D , este tipo de controlador también opera con una desviación de cero en el estado estable.

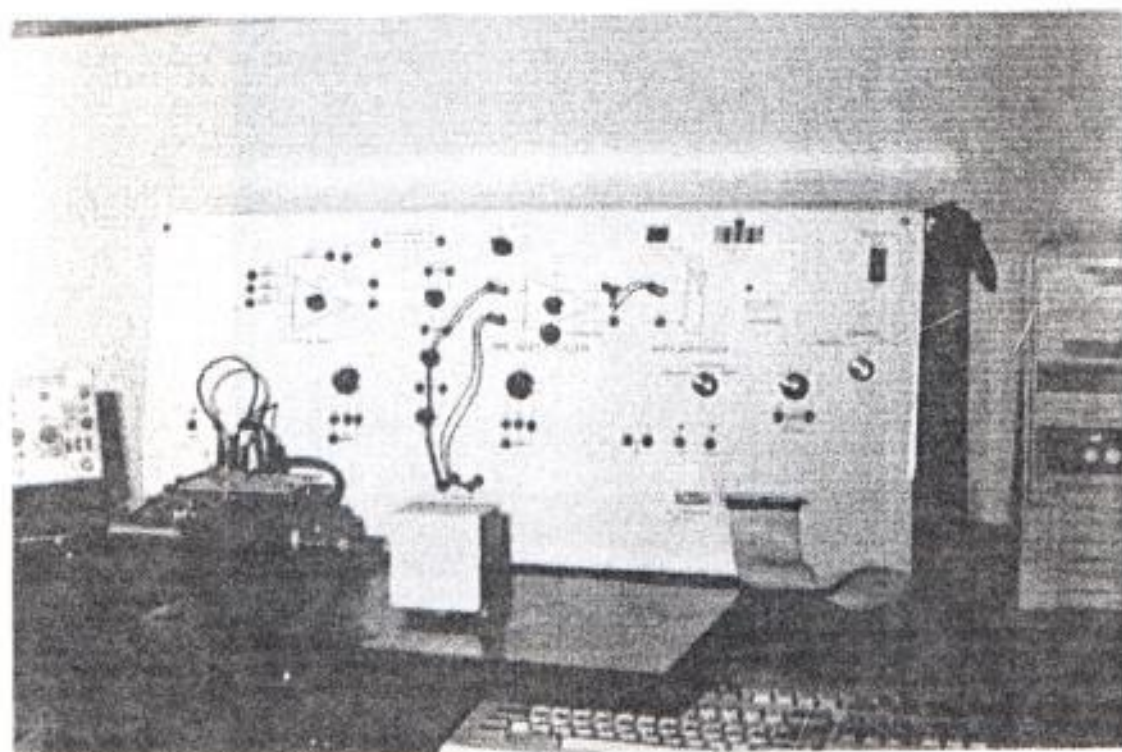


Figura 8.6 Conexiones para control digital de velocidad

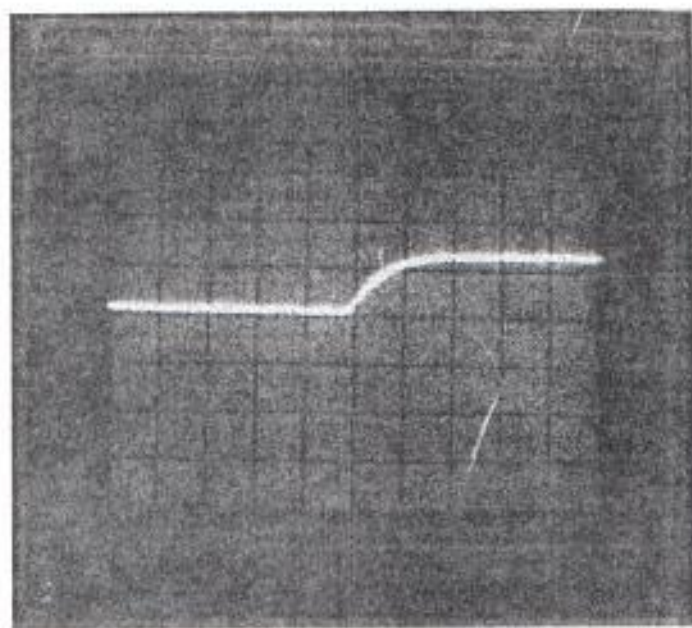


Figura 8. 7 Respuesta del sistema bajo la acción de un control proporcional
con $K_P = 5$

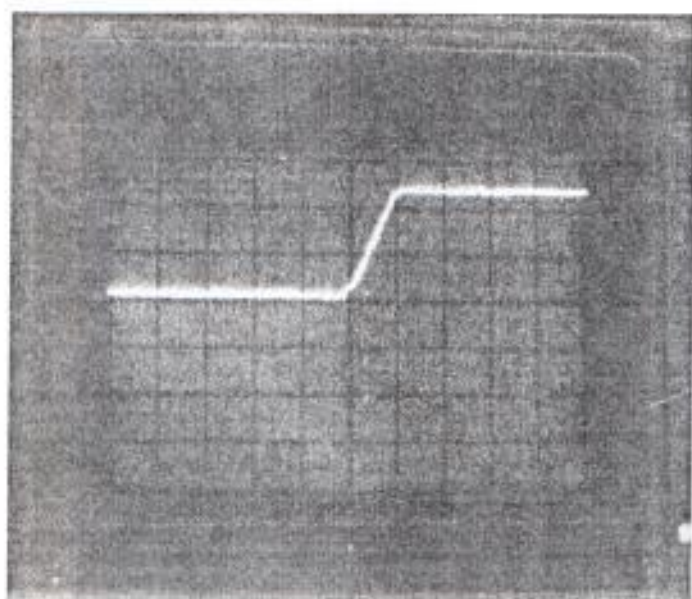


Figura 8. 8. Respuesta del sistema bajo la acción de un control proporcional

$$K_P = 8$$

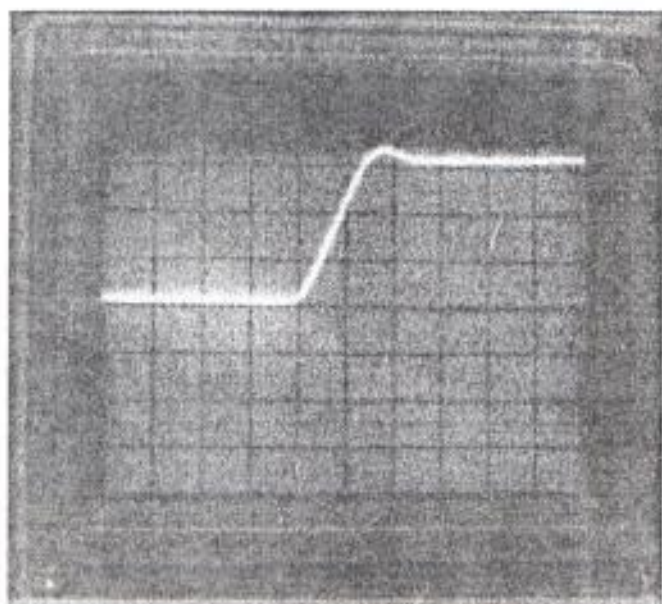


Figura 8.9 Respuesta del sistema bajo la acción de un control proporcional -
integral con $K_P=5$, $K_I=1/5$

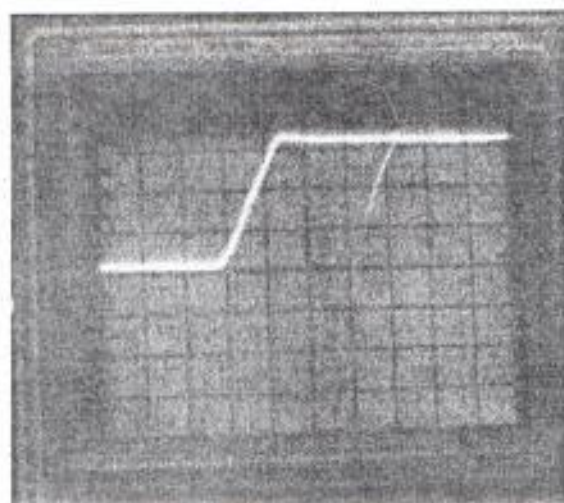


Figura 8. 10 Respuesta del sistema bajo la acción de un control proporcional-integral con $K_P = 10$, $K_I = 1/5$.

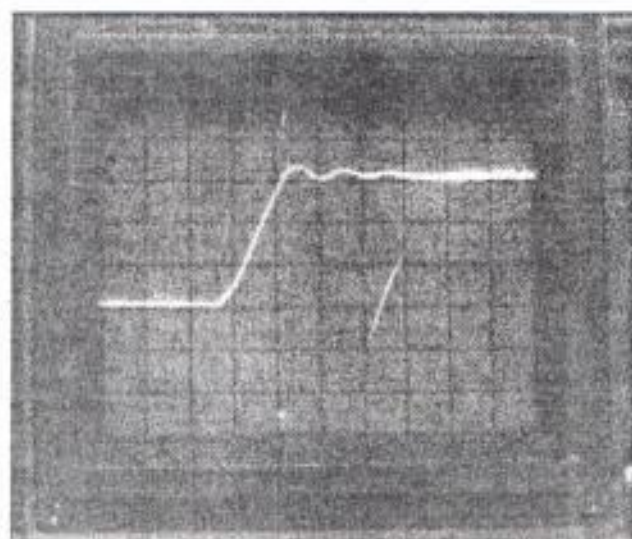


Figura 8.11 Respuesta del sistema bajo la acción de un control proporcional

- integral con $K_P = 5$, $K_I = 1/2$.

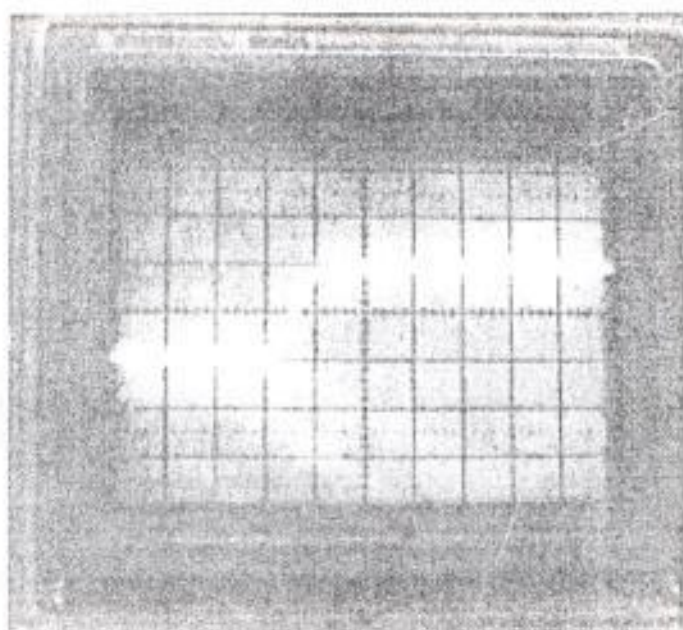


Figura 8.12 Respuesta del sistema bajo la acción de un control proporcional
- diferencial con $K_P = 5$, $K_D = 1/10$.

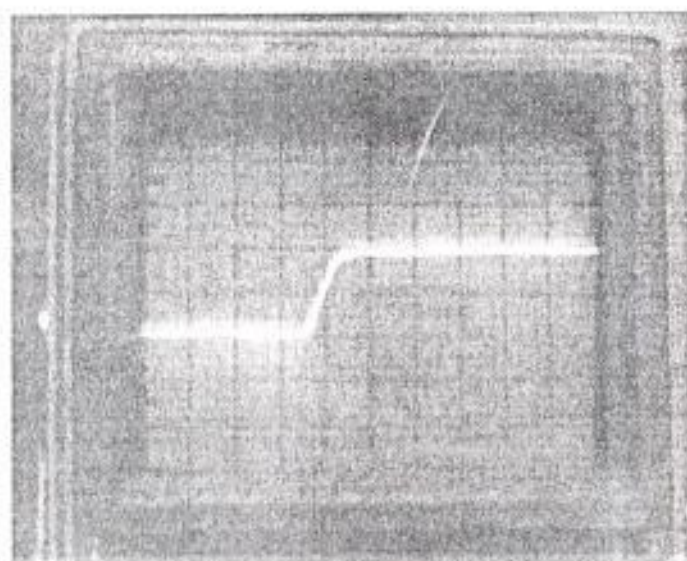


Figura 8.13 Respuesta del sistema bajo la acción de un control proporcional
-diferencial con $K_P = 5$, $K_D = 10$.

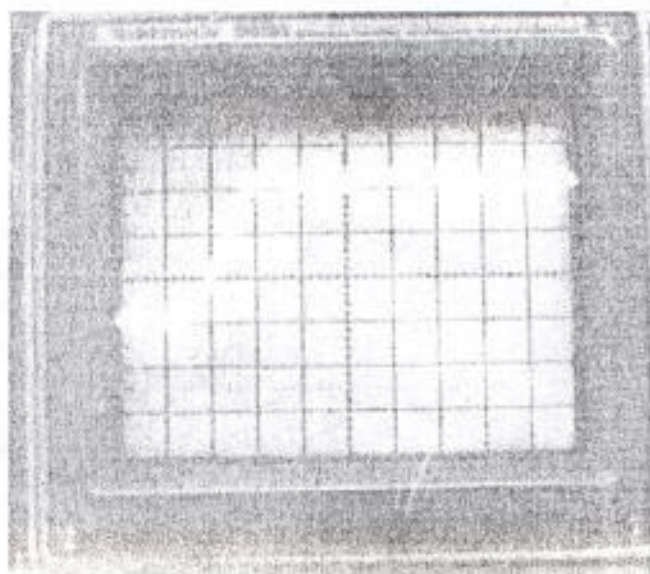


Figura 8.14 Respuesta del sistema bajo la acción de un control proporcional - integral - diferencial con $K_P = 5$, $K_I = 1/5$, $K_D = 1/10$.

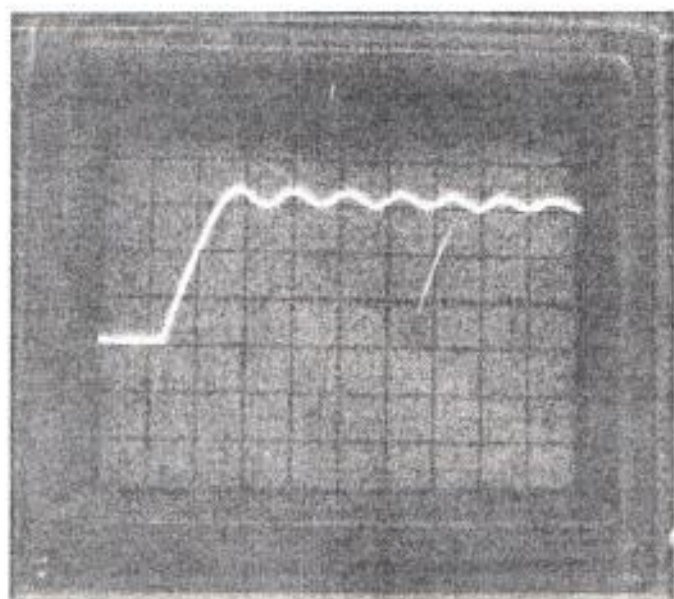


Figura 3.15 Respuesta del sistema bajo la acción de un control proporcional - integral - diferencial con $K_P=3$, $K_I=1/5$, $K_D=1/10$.

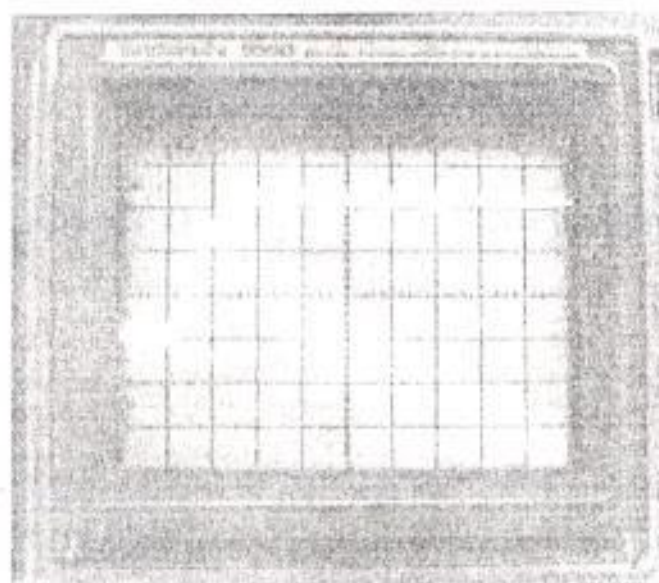


Figura 8.16 Respuesta del sistema bajo la acción de un control proporcional
-integral- diferencial con $K_P=3$, $K_I=1/5$, $K_D=20$.

CAPITULO IX

CONTROL DIGITAL DE POSICION DEL SISTEMA

9.1. OBJETIVOS

El controlador digital también ha sido diseñado para controlar la posición del motor DC, utilizando como medio transductor de posición, un codificador de eje.

Con el uso del computador, se aprovechan sus características de flexibilidad, las cuales brindan al sistema de control de posición un mejor manejo de la señal utilizando la misma circuitería con algunas variaciones en software.

9.2. DIAGRAMA DE BLOQUES

La figura 9.1, muestra el diagrama de bloques del sistema básico de control de

posición digital. Este sistema utiliza el mismo esquema del control de velocidad. La diferencia radica en que mientras en el sistema de control de velocidad la realimentación de la señal se lo hacía a través del taco-generador, ahora en el sistema de control de posición se lo realiza a través del codificador acoplado al eje de baja velocidad del motor. Este codificador de eje realiza una conversión analógica a digital, de la posición actual del eje de baja velocidad del motor. Dicha posición actual será comparada con la posición requerida del eje del motor, y el resultado de esa comparación será procesado por el algoritmo de control de posición, implementado en el microcomputador.

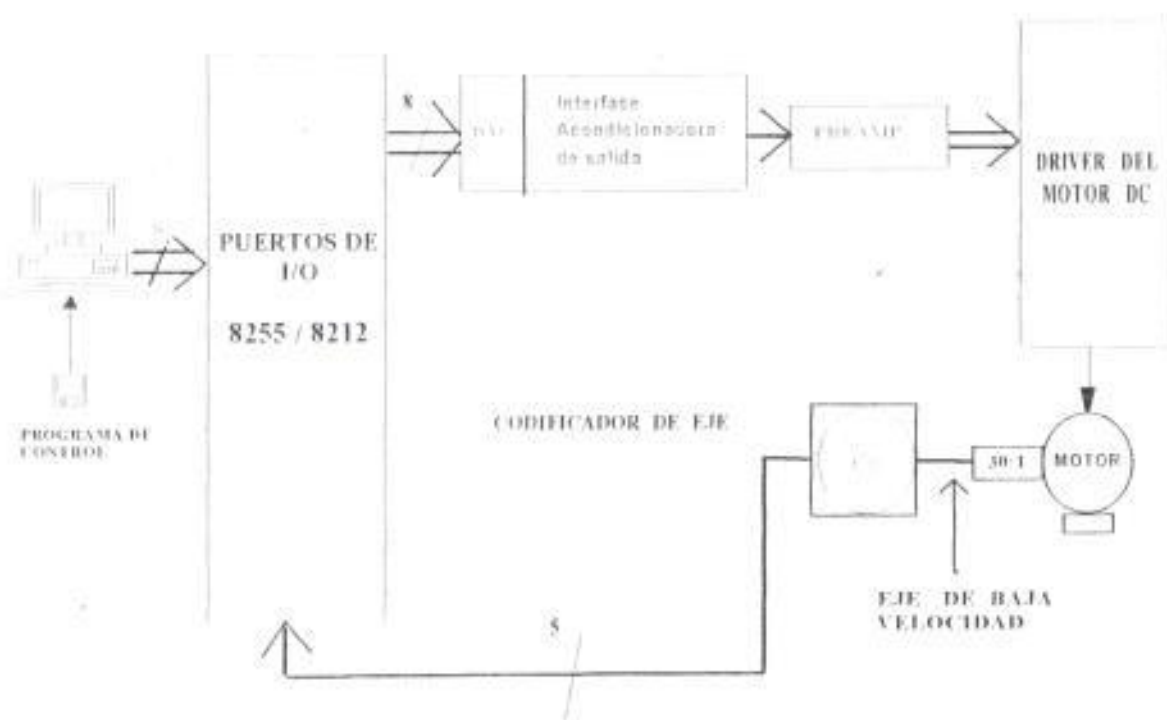


Figura 9.1 Diagrama de bloques del sistema de Control de Posición

9.3. CODIFICADOR DE EJE DEL MOTOR

El codificador de eje del motor DC aplicado al control de posición es utilizado como un elemento convertidor, ya que convierte la posición del eje de baja velocidad de una forma analógica a una digital. El codificador en mención provee una salida de cinco bits codificados en código GRAY.

Existen aplicaciones en las cuales es deseable representar números y otra clase de información con códigos que cambien en sólo una posición de bit desde una palabra de código a la siguiente. Esta clase de códigos digitales se denomina Código de Distancia Unitaria (CDU) y uno de los que pertenecen a este grupo es el código GRAY.

En la tabla 9.1 se puede observar la representación del código GRAY para los números decimales del 0 al 31 con su respectivo equivalente en binario puro.

En esta tabla podemos apreciar la diferencia que existe entre un código GRAY y un código binario puro. En el código binario puro, del paso de una palabra a la siguiente puede haber cambios desde un solo bit hasta todos sus bits, mientras que desde cualquier número decimal al siguiente sólo un bit del código GRAY cambia.

El código GRAY tiene algunas aplicaciones interesantes en un tipo de convertidor especial denominado *codificador posicional o rotacional*. El codificador rotacional o codificador de eje utilizado en el sistema es usado para convertir la posición angular del eje en formato digital.

La técnica utilizada en su estructura emplea un disco óptico acoplado al eje, el cual posee ventanas estratégicamente localizadas en patrones circulares a través de las cuales la luz puede pasar y activar un fotosensor, que para el tipo de codificador utilizado, es un fototransistor.

En el capítulo 10 se puede observar el esquema del codificador de posición del eje del motor. Si las ventanas mencionadas son colocadas sobre las circunferencias de círculos concéntricos alrededor del centro del eje, un código digital relacionado a su posición angular es generado. Por lo tanto debido a imperfecciones mecánicas (un inadecuado alineamiento óptico), si es que utilizamos un codificador que genere un código binario puro esta situación puede llevar a ciertos errores.

Por ejemplo, en una transición de 00011 a 00100, un mal alineamiento causaría un código transiente de 00101 que aparecería como salida del codificador, lo cual sería recibido como un dato errado, esto es, $00011 \rightarrow 00101 \rightarrow 00100$. Esto

ocurre cuando un mal alineamiento óptico se produce y ocasiona que no todos los bits que deben cambiar, lo hagan al mismo tiempo. Esto trae como consecuencia un dato o transición no válidos. El disco que produce un código gray elimina esto, ya que como se mencionó anteriormente en este código solo un bit cambia en una transición.

Es importante exponer el método para convertir cualquier secuencia de números binarios a su representación en código GRAY. Esto se realiza bajo el siguiente proceso:

Coloque un cero antes del bit más significativo de la secuencia de números binarios. Realice la función lógica OR-exclusivo entre el cero colocado y el primer bit (MSB) del número binario: esto es, será 1 si tales bits son diferentes, y será 0 si son iguales. El resultado de esta operación representará el primer bit del código GRAY. El segundo bit del código GRAY es igual al OR-exclusivo entre el primer y segundo bit del número binario, y así sucesivamente.

Este proceso puede ser ilustrado convirtiendo el número binario 11001 a gray en el siguiente ejemplo:

1 0 1 1 0 binario

Tabla 9.1

Equivalentes en código binario y gray para los números de 0 a 31

número decimal	número binario	número gray
0	00000	00000
1	00001	00001
2	00010	00011
3	00011	00010
4	00100	00110
5	00101	00111
6	00110	00101
7	00111	00100
8	01000	01100
9	01001	01101
10	01010	01111
11	01011	01110
12	01100	01010
13	01101	01011
14	01110	01000
15	01111	11000
16	10000	11001
17	10001	11001
18	10010	11011
19	10011	11010
20	10100	11110
21	10101	11111
22	10110	11101
23	10111	11100
24	11000	10100
25	11001	10101
26	11010	10111
27	11011	10110
28	11100	10010
29	11101	10011
30	11110	10001
31	11111	10000

Se coloca un cero antes del bit más significativo y se procede a realizar un OR-exclusivo.



Siendo este último 11101, la representación en código GRAY del número binario dado.

Este codificador de eje es del tipo de vuelta única, es decir que los códigos para las posiciones extremas (00H y 1FH) son los equivalentes binarios expresados en hexadecimal para las posiciones extremas de un codificador en GRAY de 5 bits) son adyacentes el uno al otro.

Como para una vuelta del disco se tiene 32 posiciones distintas, la resolución del codificador es de 11.25 grados por posición diferente. Es decir cada 11.25 grados de rotación del eje del motor, se verá un cambio en la salida del codificador. En la figura 9.2 y en la tabla 9.2 se ilustra de una mejor manera lo expresado anteriormente.

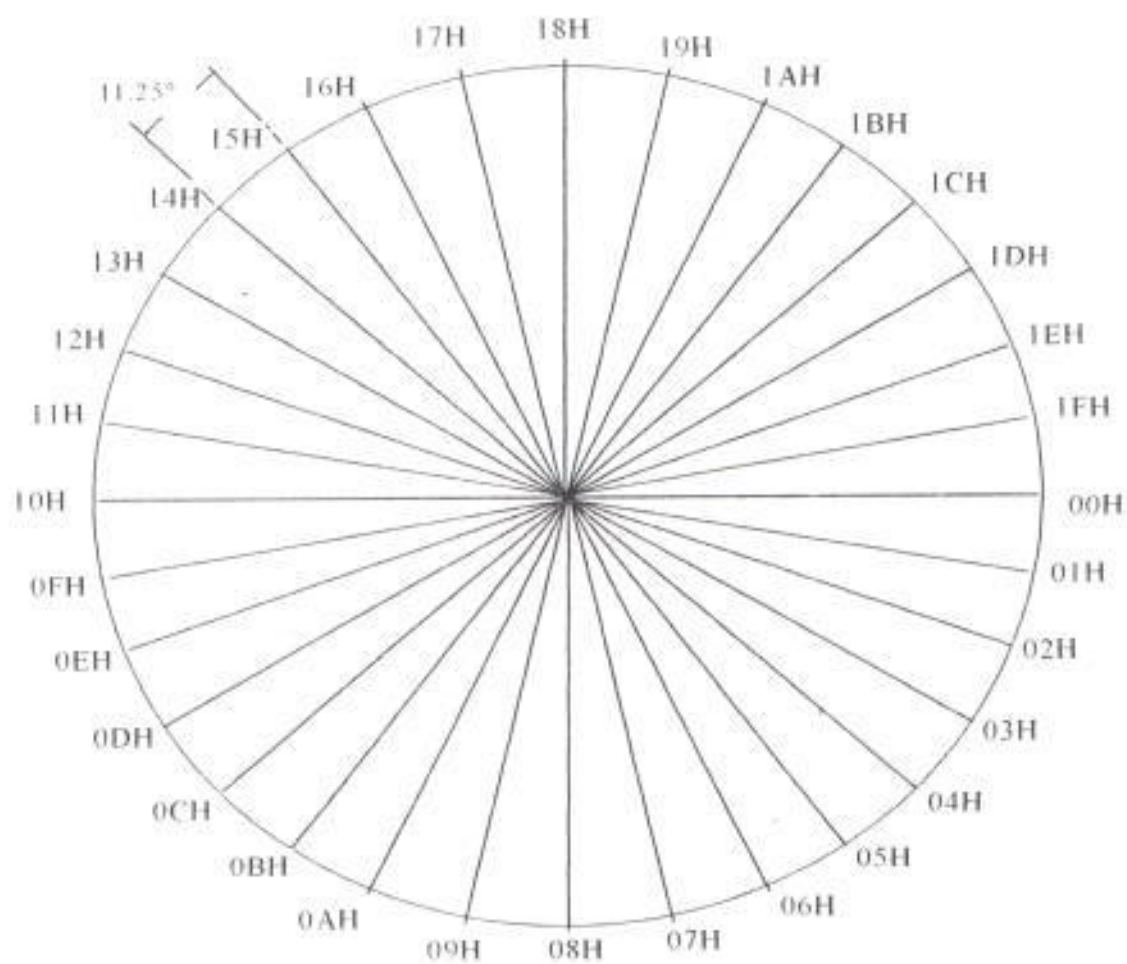


Figura 9.2 Representación del Disco Posicional

Tabla 9. 2
Correspondencia entre la posición hexadecimal del disco
y los grados de circunferencia

POSICION HEXADECIMAL	GRADOS
00	0
01	11.25
02	22.50
03	33.75
04	45.00
05	56.25
06	67.50
07	78.75
08	90.00
09	101.25
0A	112.50
0B	123.75
0C	135.00
0D	146.25
0E	157.50
0F	168.75
10	180.00
11	191.25
12	202.50
13	213.75
14	225.00
15	236.25
16	247.50
17	258.75
18	270.00
19	281.25
1A	292.50
1B	303.75
1C	315.00
1D	326.25
1E	337.50
1F	348.75

9.4. FUNCIONAMIENTO DEL SISTEMA

Observando la figura 9.1 que representa el diagrama de bloques del sistema de control de posición se deduce que es el mismo esquema del control de velocidad excepto que ahora la realimentación del circuito de lazo cerrado se lo hace mediante la variable controlada de *posición*.

A nivel de hardware en el sistema de control de posición las 5 líneas que componen el código GRAY y que representan la posición actual del motor son conectadas directamente a los 5 bits menos significativos del puerto C de la interface 8255. Este PPI se encuentra interactuando directamente con la barra de direcciones, de control y de datos del microcomputador.

El soporte principal del funcionamiento del sistema de control de posición se encuentra en el desarrollo del software. Por tanto explicaremos paso a paso desde el inicio del programa hasta cómo lograr una posición deseada a partir de una posición actual, el cual es el objetivo principal de este capítulo.

1. Ingresamos al programa de control de posición del sistema pulsando la tecla 2. El ingreso se lo realiza por intermedio de una subrutina denominada CONTROLPOS. Dicha subrutina cuando es llamada escribe en el campo de

dirección de la pantalla los mensajes correspondientes de posición actual y posición requerida. Posteriormente se hace una lectura del puerto C del 8255 y el dato codificado en GRAY proveniente del codificador se procede a convertirlo a código binario puro por medio de una tabla de conversión de GRAY a binario. Esta tabla será analizada más adelante. Después de ser convertido a binario puro, este valor se lo transforma a grados mediante un factor de relación $360/32$, ya que el codificador de eje tiene 32 posiciones diferentes, por lo tanto del factor de relación se obtiene 11.25. Entonces cada 11.25 grados tendríamos un valor de posición diferente. Despejando $360/32$ se tendrá $45/4$ y este es el factor final de conversión. Una vez obtenido el valor en grados el cual consta de un máximo de 3 dígitos (0 hasta 359), se lo almacena en una variable llamada POSACTUAL que me indica la posición actual del motor. Todas estas conversiones y operaciones son realizadas por la macro LEEPOSICION la cual es invocada dentro de la subrutina CONTROLPOS.

2. Para realizar la comparación entre la posición requerida y la posición actual de realimentación es conveniente que ambos valores se encuentren en compatibilidad de códigos, es decir que ambos sean expresados en código binario puro, por lo que se hace necesario la conversión de la posición actual codificada en GRAY a código binario. Para desarrollar el proceso de conversión hemos optado por

utilizar una tabla de conversión la cual resulta conveniente debido a la baja resolución de 5 bits que tiene el codificador. El proceso consiste en que el número en Gray proveniente del codificador será utilizado para fijar las direcciones de la tabla de conversión. En primer lugar, obtenemos una tabla con los números gray correspondientes a los números decimales 0 a 31 (que representan las 32 posibles posiciones del codificador). Luego realizamos la tabla inversa de dicho arreglo de 32 códigos gray y obtenemos un arreglo de 32 posiciones cuyas direcciones en la tabla serán determinadas por cada código Gray. Esta tabla final es la correspondiente a la tabla Gray a Binario que se necesita en la macro LEEPOSICION.

3. La entrada de la posición requerida al sistema se lo realiza cuando el procedimiento de control de posición verifica el contenido del buffer del teclado haciendo uso de la interrupción INT 21H del DOS. El programa compara el contenido del registro AL con el valor presente en el buffer que en el caso de requerir una posición del eje de baja velocidad del motor sería el código ASCII de la tecla 1 que vendría a ser 31H. Haciendo uso de la macro INGRESARNUM, la cual permite ingresar un número positivo desde el teclado e INGRESARNUM2 que permite ingresar un número negativo desde el teclado, entramos la posición requerida que debe tener una longitud máxima de 3 dígitos y valor en un rango

entre (-359 y 359). Cabe indicar que el usuario puede ingresar cualquier número dentro de este rango, pero como el programa de control de posición solamente reconoce valores que sean múltiplos de 11.25 grados (considerando que el computador toma la parte entera del valor ingresado), el algoritmo de control siempre hará que el motor se pare en el valor de posición múltiplo de 11.25 inmediatamente menor. Esto se explica mejor de la siguiente manera: Si el usuario ingresa un ángulo (valor de posición) de 30 grados, entonces el motor se parará en 22 grados y si ingresa -20 grados el motor se detendrá en -22 grados. Obviamente si el usuario ingresa ángulos múltiplos de 11.25 el motor va a detenerse en esos valores de posición. Para lograr estas posiciones se realiza un llamado a la subrutinas RECTIFICA o RECTIFICA2 según sea el sentido de giro.

4 Las subrutinas RECTIFICA y RECTIFICA2 son aquellas que cambian y corrigen la posición del eje del motor. En esta subrutina se recurre a una fórmula obtenida en pruebas experimentales, la cual corresponde al valor de posición requerida que se almacena en una localidad de memoria cuya variable asignada es REFERENCIA. La razón de la fórmula de REFERENCIA es principalmente para que el número ingresado sea compatible con la variaciones de posición reconocidas por el codificador del eje de baja velocidad del motor, esto es que el número ingresado correspondiente al valor de posición requerida sea múltiplo

de 11.25 grados. Para justificar esta fórmula se ha recurrido al siguiente ejemplo demostrativo:

Primero veamos la fórmula de REFERENCIA:

$$\text{REFERENCIA} = (\text{REFERENCIA} \cdot 4 + 3) / 45 \cdot (45/4)$$

donde REFERENCIA viene dado en grados y el factor de conversión a grados es $45/4 = 11.25$. Supongamos que se desea ingresar un valor de posición de 24 grados (no es múltiplo de 11.25 grados), entonces reemplazando en la fórmula tenemos:

$$\text{REFERENCIA} = (24 \cdot 4 + 3) / 45 \cdot (45/4)$$

que da como resultado:

$$\text{REFERENCIA} = 22.5 \text{ grados, cuyo valor sí es múltiplo de 11.25}$$

Cabe mencionar que el computador realiza las operaciones tomando la parte entera de sus resultados parciales, por lo que el valor final mostrado en pantalla será:

$$\text{REFERENCIA} = 22 \text{ grados}$$

Este si es un valor que puede ser operado y aceptado como posición en el programa de control. Si el usuario quiere comprobar la veracidad de ésta fórmula basta con ir probando con cualquier valor de 0 a 359 y se llegará a la conclusión que siempre se obtendrá un valor múltiplo de 45/4. Debido a que el sistema de control de posición debe hacer girar el motor en los dos sentidos (ángulos positivos y negativos). Esto se logra mediante la interface acondicionadora de salida el cual consta de un convertidor digital - analógico y un circuito amplificador sumador-inversor. En la salida del DAC tenemos un rango de 0 a -5 V el cual es sumado a un voltaje de 5 V como se muestra en la figura 9. 3.

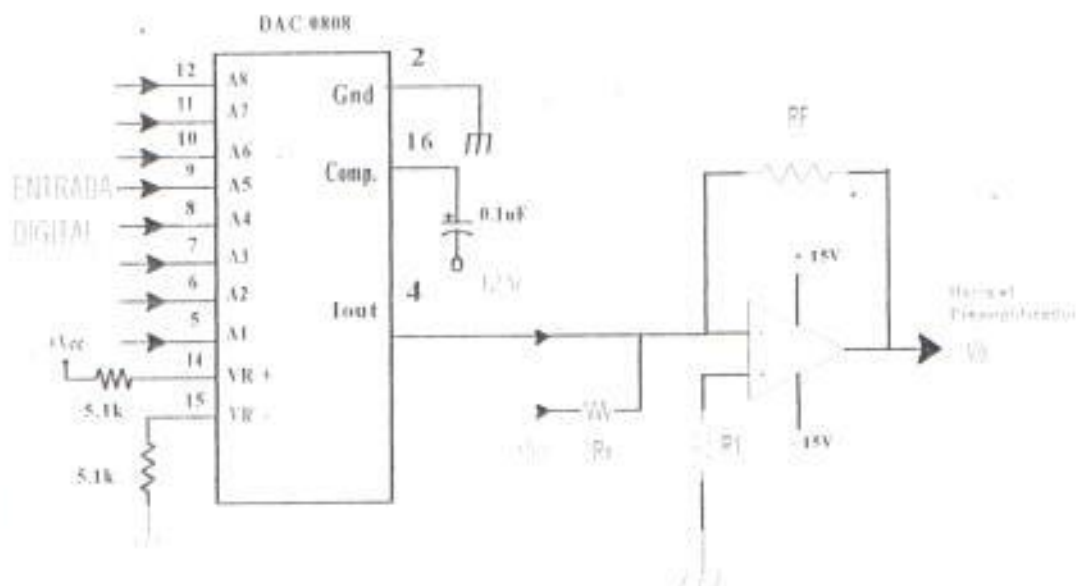


Figura 9. 3 Diagrama completo de la Interface Acondicionadora de Salida

Como se puede observar, al pin 2 del amplificador operacional se ha conectado 2 ramales: la salida unipolar del DAC 0808 (rango entre 0 y -5 V) junto con la resistencia interna del convertidor que es de 5.1 K y la fuente de 5 V con una resistencia R_x (de 8.77 K). La razón de este segundo ramal es para obtener también salidas con voltajes positivos, retroalimentándose a la salida a través de una resistencia R_f (de 2.41 K). En el pin 3 del op amp se ha dispuesto una resistencia conectada a tierra de $(5.1K // R_x // R_f)$ para compensar las resistencias conectadas a través de la entrada inversora.

5. En la subrutina de control de posición CONTROLPOS, se compara el valor de REFERENCIA y de POSACTUAL; si ambos son iguales se realiza un retorno de subrutina y el programa de control de posición sigue revisando el buffer de entrada del teclado en espera de una tecla oprimida para ingresar ya sea una nueva posición con la tecla 1 o definitivamente salir del programa hacia el menú principal mediante la tecla 2. En esta misma subrutina se verifica mediante comparación de bit de signo si el valor de posición ingresado es positivo o negativo. Además se analiza si el valor de REFERENCIA (posición requerida) es menor que el valor de POSACTUAL (posición deseada). En el caso que el valor ingresado sea negativo o menor que la posición actual del motor el algoritmo llamará al procedimiento RECTIFICA. En los demás casos llamará al procedimiento RECTIFICA 2.

6. El procedimiento **RECTIFICA** tiene dos alternativas: cuando el valor ingresado es negativo pero **mayor** que **POSACTUAL** (por ejemplo si la demanda es -20 y la posición actual es -90) directamente se compara la diferencia entre ambos con 180 y como el error de desviación en valor absoluto es menor, el programa me envía a través de la directiva **RECT 3** a un lazo en el cual la acción de control proporcional está en el sentido de giro positivo, es decir el camino más corto para el desplazamiento hacia ese ángulo. Mientras que cuando el valor ingresado es **menor** que la posición actual, el programa me envía a un lazo en el cual la acción de control se lo realiza en sentido de giro de las manecillas de reloj (sentido negativo), de tal forma que el codificador sea comandado en un sentido tal que se reduzca **POSACTUAL** hasta que alcance el valor de posición determinado por **REFERENCIA** (por ejemplo de 90 a 20 grados o de -10 a -90 grados).

El procedimiento **RECTIFICA2** es invocado cuando se tiene una **REFERENCIA** mayor que **POSACTUAL** siempre y cuando éstos valores sean positivos. De esta forma el sentido de giro del motor será contrario al de las manecillas de reloj (sentido positivo).

7. En los procedimientos **RECTIFICA** y **RECTIFICA 2** el movimiento del motor está comandado por la acción de un controlador proporcionalmente 'puro'. Esta

acción se ve reflejada en una constante de proporcionalidad la cual influye sobre el error mediante la siguiente relación:

$$DAC = ERROR \times KP$$

en donde DAC es una variable de memoria que contiene el valor a ser enviado al convertidor. La señal proveniente del computador (dato de 8 bits) representado en $ERROR \times KP$ es introducida al convertidor digital-analógico mediante el envío del byte actuante hacia la localidad de memoria direccionada por PORTA (puerto A del 8255).

Debido a la operación de la interface acondicionadora de salida en forma bipolar, se entiende que el motor se encuentra estacionado cuando la entrada del preamplificador tenga 0V.

Por tanto para la bipolaridad planteada anteriormente se tiene que 0V analógicos equivale a 128 en formato digital binario puro. Esto significa que al determinar que el motor se detenga es necesario enviar un byte de valor 128 a través del DAC. Por lo tanto para tener voltajes negativos a la entrada del preamplificador se requiere enviar en el programa valores digitales **menores** a 128 y para tener

voltajes positivos a la entrada del preamplificador se requiere enviar en el programa valores digitales mayores a 128.

En el diseño se ha conectado una de las entradas del preamplificador a tierra y la otra directamente al circuito convertidor digital-analógico. Las dos salidas del preamplificador están conectadas una de ellas a la entrada 1 del servo y la otra a la entrada 2 del servo.

Por lo tanto si en la entrada del preamplificador se tiene voltajes positivos, la entrada 1 del servo tendrá niveles de voltaje lo suficientemente altos para hacer mover el motor en el sentido de giro positivo, mientras que la entrada 2 tendrá niveles de voltaje bajos.

Y viceversa, si hay voltajes negativos en 1, la entrada 2 del servo tendrá niveles de voltaje lo suficientemente altos para hacer mover el motor en el sentido contrario, (giro en favor de las manecillas del reloj). Debemos tener presente que es necesario un voltaje mínimo de 6.2 voltios en cualquiera de las entradas del servo para vencer la inercia del motor.

Un aspecto esencial en el control de posición es la velocidad que debe tener el motor en el intento por alcanzar la posición deseada. La señal de ERROR está

determinada por la siguiente relación:

$$\text{ERROR} = \text{REFERENCIA} - \text{POSACTUAL}$$

donde **ERROR** es una variable que contiene el valor de la desviación producida, **REFERENCIA** es una variable que almacena el valor de la posición deseada y **POSACTUAL** es una variable que guarda el valor de la posición actual. Por lo tanto la velocidad será proporcional al error que se produzca en ese momento. Al inicio el error (valor de **ERROR**) es grande y por lo tanto la velocidad será grande y a medida que se va acercando al valor de posición deseada ésta irá disminuyendo hasta que se detenga en el ángulo final determinado.

En la figura 9.4 se muestra la fotografía de las conexiones realizadas en el banco para un sistema de control digital de posición. Aquí se hará un análisis de la respuesta del sistema bajo un control proporcional, las pruebas serán realizadas en base a un cambio desde una posición actual del eje a una posición deseada. Esto se lo hace con la finalidad de poder comparar para una misma condición de demanda, diferentes ganancias del controlador.

Como se ve en la figura 9.5, para una demanda de 45 grados y $K_P = 1, 1/2, 1/5$ y $1/10$ en ese orden. Del análisis de esta figura se concluye que el modo de control

proporcional es muy eficiente para sistemas donde no es necesario una fuerza de control de estado estable ya que con este se logra un error cero en la posición en estado estable. También se puede observar de la misma figura, que a mayores valores de la ganancia proporcional se tendrá una respuesta más rápida y al mismo tiempo será mayor el voltaje de salida en el tacogenerador, lo que concuerda con el sistema de conexión de este tipo de control.

También se puede observar que a mayor demanda la respuesta es más rápida, con una sobrecorrección tanto positiva como negativa, esto se muestra en la figura 9.6, en términos generales mientras mayor es la ganancia o la demanda la respuesta del sistema tiene sobrecorrección. En el caso de mantener constante K_P y variar la demanda a intervalos de 90 grados vemos que el sistema responde de la misma forma para demandas iguales. También en el caso de que la demanda fueran ángulos negativos en intervalos también de 90 grados, el sistema responde de la misma forma que el anterior con la diferencia que las gráficas resultantes serán invertidas debido a que el voltaje de salida en el tacogenerador es negativo. Esto se muestra en la figura 9.7.

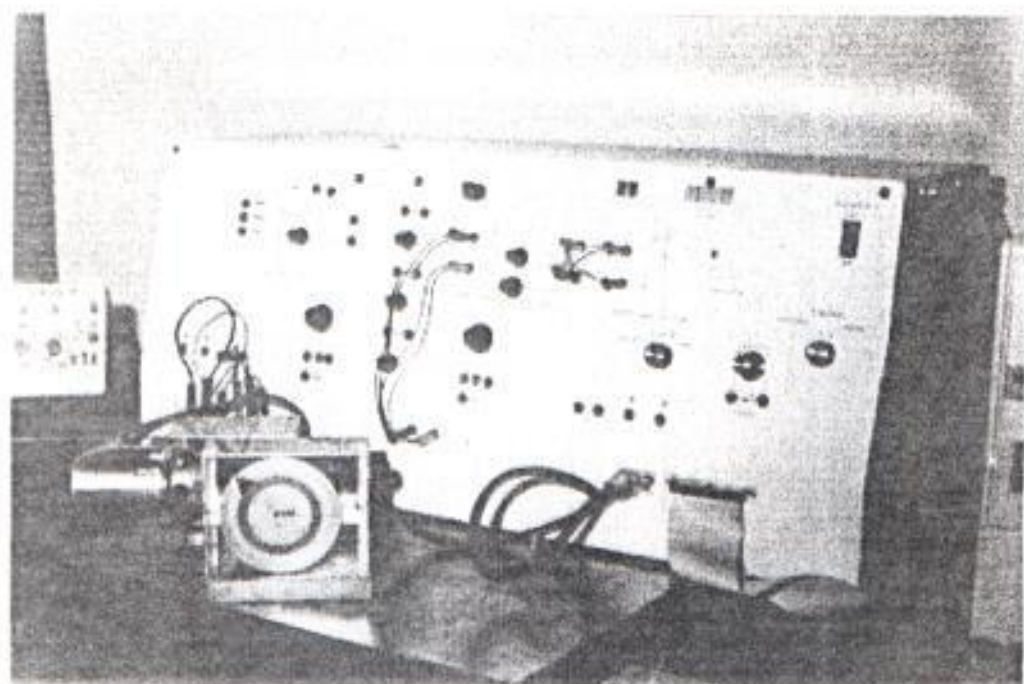


Figura 9.4 Diagrama de conexiones para el control digital de posición

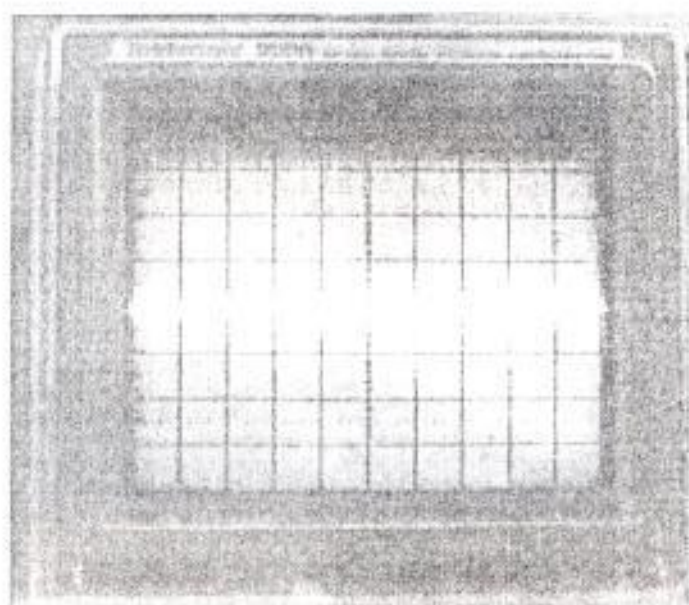


Figura 9. 5 Respuesta del sistema bajo la acción de un control proporcional con $K_P = 1, 1/2, 1/5$ y $1/10$ para una demanda de 45 grados.

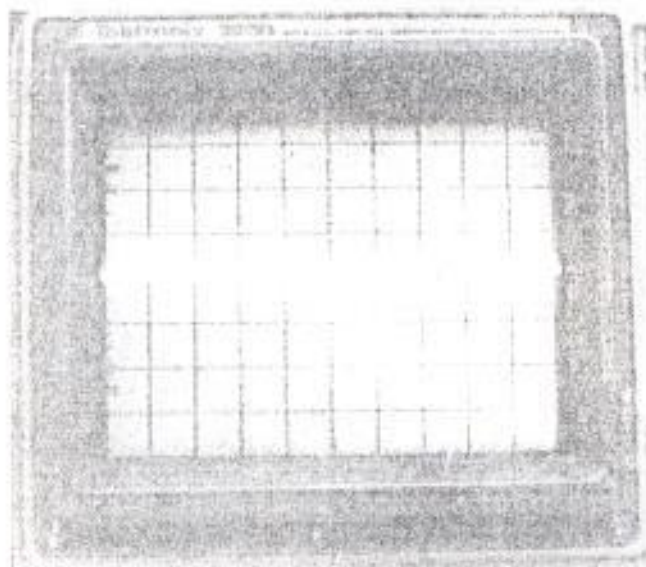


Figura 9. 6 Respuesta del sistema bajo la acción de un control proporcional con $K_P = 1/2$ para una demanda de 0 - 45 - 135 y 315 grados.

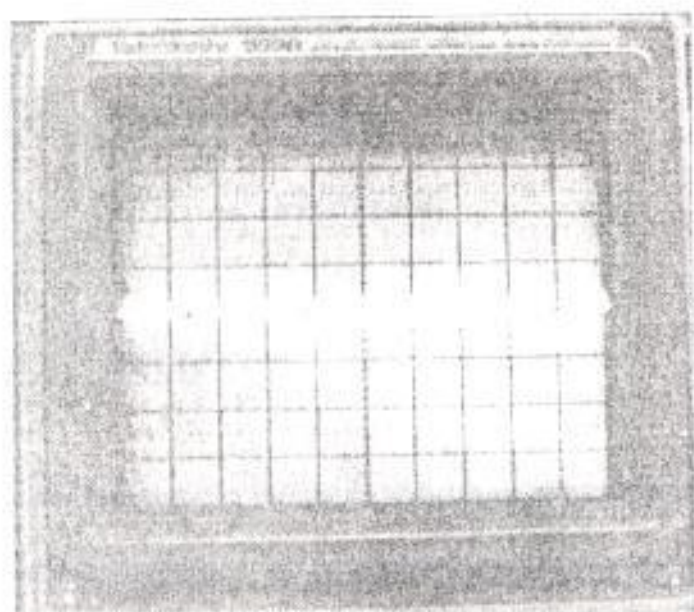


Figura 9.7 Respuesta del sistema bajo la acción de un control proporcional con $k_P = 1.5$ para una demanda de $0 - 90 - 0$ grados.

CAPITULO X

DETALLES DEL DISEÑO

En este capítulo se enumeran los diferentes diagramas esquemáticos de los circuitos analógicos y digitales utilizados en la presente tesis, los cuales se enumeran a continuación

- ◆ Circuito de la fuente de poder.
- ◆ Circuito del pre-amplificador.
- ◆ Circuito del servo amplificador : control por armadura.
- ◆ Circuito del servo amplificador : control por campo.
- ◆ Circuito del controlador P.I.D.
- ◆ Circuito de conversión analógico - digital.
- ◆ Circuito de conversión digital - analógico.

- ◆ Circuito acondicionador de entrada.
- ◆ Circuito acondicionador de salida.
- ◆ Circuito decodificador de direcciones.
- ◆ Circuito del reloj para el ADC 0808.
- ◆ Circuito de la tarjeta de interfaz de datos.
- ◆ Circuito del codificador de posición.
- ◆ Circuito del amperímetro digital.
- ◆ Circuito del tacómetro digital.

En las siguientes figuras se muestran la posición de los elementos de las diferentes tarjetas.

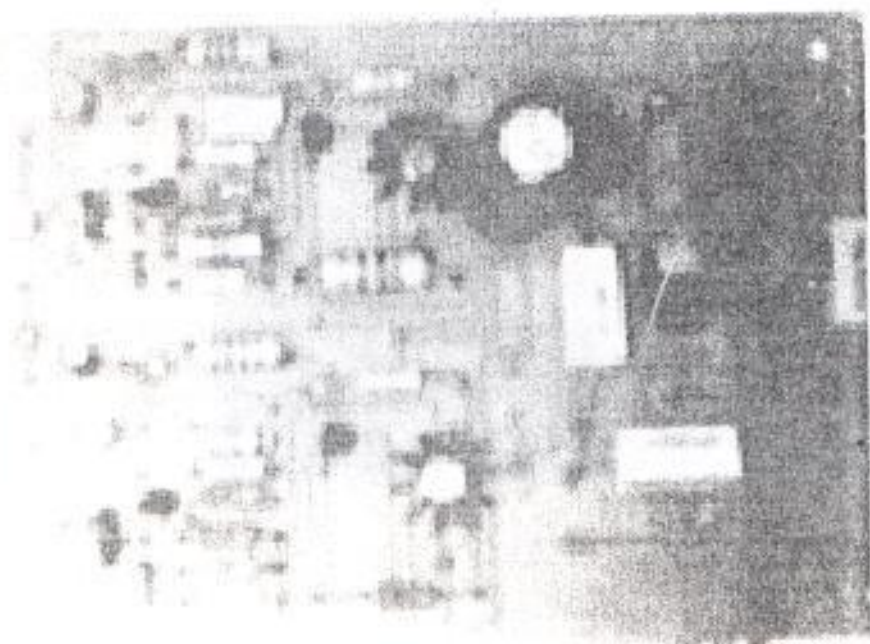


Figura 10.1 Fuente de poder $\pm 15V$, $\pm 27V$

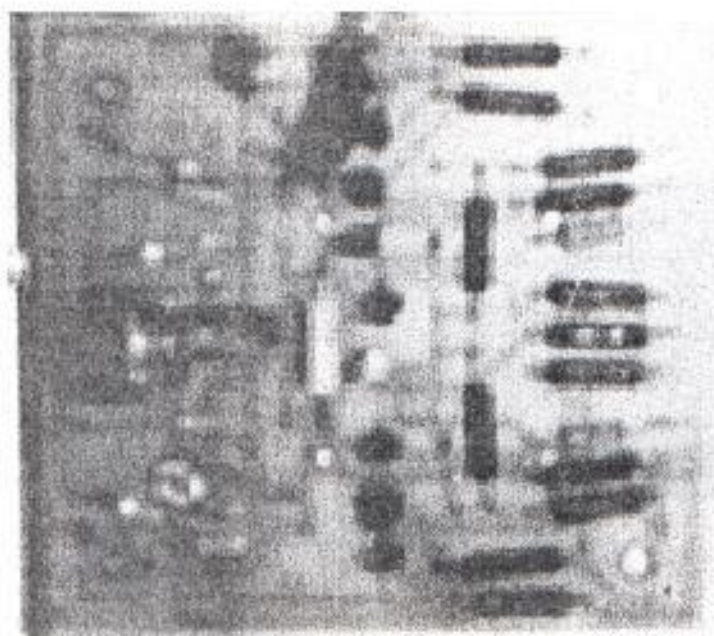


Figura 10. 2 Preamplificador

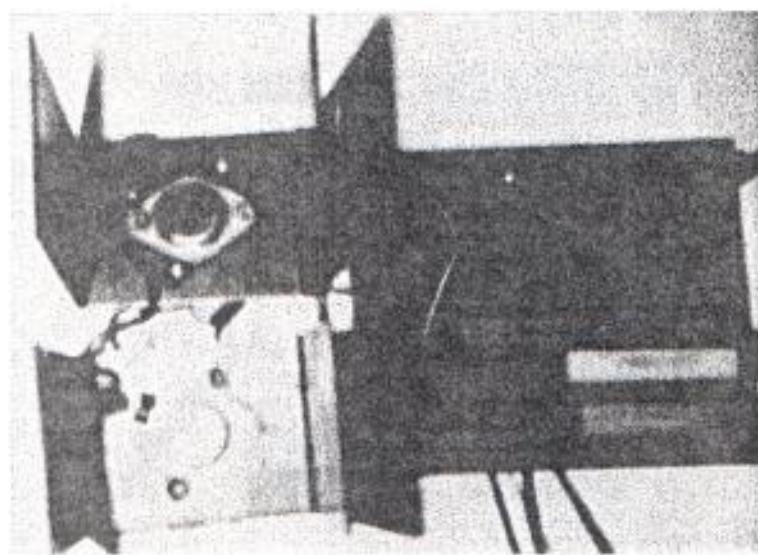


Figura 10.3 Servo amplificador



Figura 10.5 Tarjeta de interfaz de datos



Figura 10. 6 Tacómetro digital y fuente de 8V

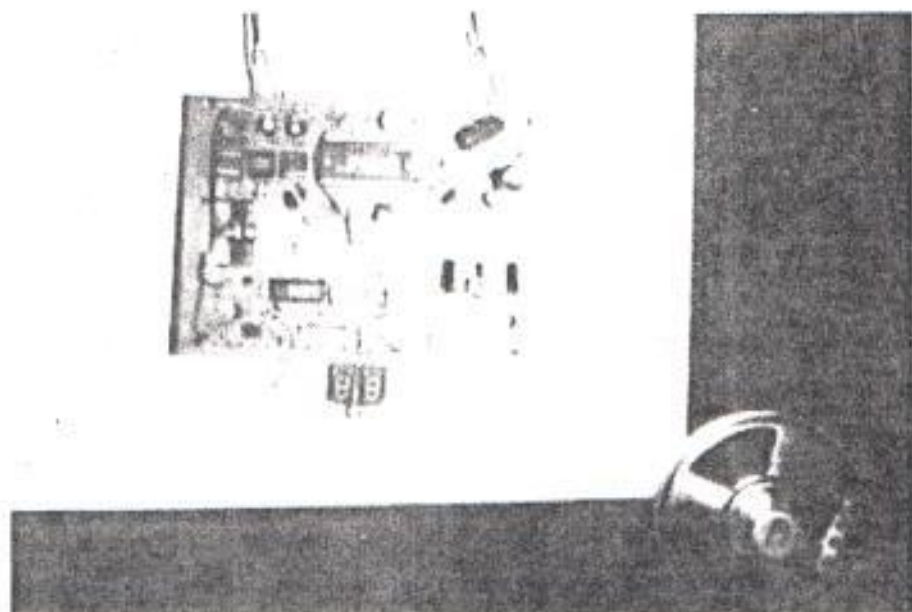


Figura 10.7 Amperimetro digital y circuito de alarma (sobrecarga)

CAPITULO XI

IMPLEMENTACION DEL PROGRAMA DE CONTROL DIGITAL

11.1. DESCRIPCION DE LENGUAJE ENSAMBLADOR

El ensamblador es un programa de computadora que traduce grupos de instrucciones escritos en lenguaje ensamblador en instrucciones en lenguaje de máquina. Este es un programa que guarda estrecha relación con la arquitectura del hardware del sistema computador.

A continuación mencionamos algunas ventajas del lenguaje ensamblador:

- Mayor control del hardware del computador.
- Desarrollo de programas de rápida ejecución.

- Acceso optimo y eficiente al coprocesador

El lenguaje ensamblador fue utilizado para implementar el programa de control, tanto de velocidad como de posición, del sistema digital. En el programa se hace uso de varias instrucciones, entre las cuales tenemos las siguientes:

La instrucción MOV coloca el contenido del operando fuente en el operando destino:

```
MOV [ OPERANDO DESTINO], [ OPERANDO FUENTE ]
```

Por ejemplo,

```
MOV AH, 4CH
```

En este caso el ensamblador hace uso del direccionamiento inmediato para colocar la constante 4CH (operando fuente en hexadecimal) en el registro AH (operando destino). El lenguaje ensamblador emplea la instrucción ADD para suma binaria de 8, 16 y 32 bits.

Por ejemplo:

```
ADD DAC, AX
```

En este caso se suma el contenido de la variable DAC con el contenido del registro AX y el resultado de la suma se coloca en el primer operando, es decir en la variable DAC. Por lo tanto el valor actual de DAC es igual al valor anterior de DAC más AX. Igual sucede en la operación de sustracción donde se usa la instrucción SUB.

Por ejemplo:

```
SUB AX, BX
```

Aquí el contenido de AX se resta de BX y el resultado se guarda en AX.

Una de las instrucciones utilizadas para efectos de comparación es la instrucción CMP. La instrucción CMP es una resta que sólo afecta los bits de bandera. La comparación es útil cuando se quiere comparar el contenido de un registro o localidad de memoria con otro valor. La instrucción CMP suele ir acompañada con otra instrucción de salto, por ejemplo JNE (condicional) que prueba los bits de bandera:

```
CMP AL, 31H
```

```
JNE POS2
```

El ensamblador compara el contenido de AL con 31H y si no son iguales el flujo del programa se traslada a POS2. POS2 es una directiva que me especifica una línea del programa. Además de las instrucciones aritméticas de suma y resta existen las instrucciones de multiplicación y división. La multiplicación se efectúa en bytes, palabras o doble palabras y pueden ser enteros con signo (IMUL) o sin signo (MUL). Si se multiplican dos números de 8 bits se genera un producto de 16 bits, si se multiplican dos de 16 bits se genera un producto de 32 bits. En la multiplicación de palabras (la más común) el multiplicando está siempre en el registro AX, el multiplicador puede ser cualquier registro (en el caso que se verá a continuación será CX) y el producto aparece en DX y AX. El registro DX siempre contiene los 16 bits más significativos del producto y AX los 16 menos significativos.

```
MOV AX, VELACTUAL
```

```
MOV CX, FACTCONV
```

```
MUL CX
```

En AX se encuentra el multiplicando (VELACTUAL) y en el registro CX se encuentra el multiplicador (FACTCONV). Donde VELACTUAL y FACTCONV son variables del programa. La división, igual que la multiplicación, se efectúa con números de 8, 16, 32 bits. El dividendo siempre es uno de doble

ancho que se divide para el operando, por ejemplo, en una división de 16 bits se divide un número de 32 bits para uno de 16 bits. Una división de 16 bits utiliza el DX y AX para almacenar el dividendo, que se divide para el contenido de cualquier registro o localidad de memoria de 16 bits. El cociente se transfiere a AX después de la división y DX contiene un residuo de número entero. Por ejemplo:

```
MOV AX, 255
MOV CX, KIDEN
MUL CX
MOV CX, KINUM
DIV CX
```

MACRO Una macro es un grupo de instrucciones que efectúan una tarea, igual que el procedimiento ejecuta una sola tarea. La diferencia radica en que a un procedimiento se accede con una instrucción CALL, mientras que el macro se inserta en el programa como nuevo código que contiene una secuencia de instrucciones. Una macro en resumen es un nuevo código que el programador produce. Las macrosecuencias se ejecutan con mayor rapidez que los procedimientos porque no necesitan ejecutar instrucciones RET y CALL. Las

instrucciones de una macro se colocan en el programa principal en el punto que se les invoca.

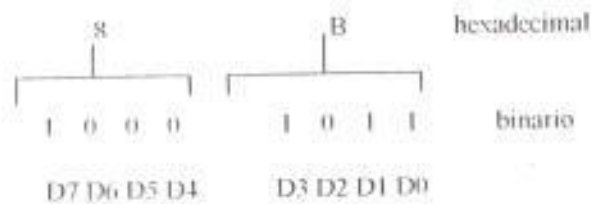
Las directivas `MACRO` y `ENDM` se emplean para delinear una macrosecuencia. El primer enunciado es el enunciado `MACRO` que contiene el nombre de la macro y cualesquiera de los parámetros relacionados con ello. A veces, los macros contienen variables locales. Una *variable local* es la que aparece en el macro, pero no está disponible fuera de él. Para definir una variable local, se utiliza el directivo `LOCAL`.

Cabe mencionar que existen también otras instrucciones tales como de llamado a subrutina (`CALL`), de retorno de subrutina (`RET`), instrucciones lógicas básicas (`AND,OR`), instrucciones de corrimiento (`SHL,SHR`), grupo de instrucciones de entrada y salida I/O (`IN, OUT`),etc. Para una descripción más detallada de otras y cada una de las instrucciones vistas anteriormente, el lector puede recurrir a cualquier texto de lenguaje ensamblador.

11.2. DIAGRAMA DE FLUJO DE PROGRAMA PRINCIPAL

La figura 11.1 muestra el diagrama de flujo del programa principal.

La interface programable de periféricos 8255 es el puerto principal de entrada y salida de datos en el diseño del controlador digital. En el momento de aplicar potencia por primera vez al circuito, el PPI 8255 se encuentra en estado de *reset*, es decir con todas las líneas de los puertos y bus de datos flotantes o en estado de alta impedancia. Por esta razón, antes de utilizar los puertos de entrada y salida, el *circuito integrado* debe ser inicializado *cargando* el registro de control con una palabra de 8 bits. La palabra cargada a la variable REGISTRO que representa el registro de control (dirección de memoria : 503H) es 8BH (en hexadecimal). Esta palabra llevada a formato binario da como resultado:



Esta palabra de control configura el funcionamiento del PPI. Cabe mencionar que este registro solamente puede ser escrito, es decir no es posible leer su contenido desde el microprocesador. La justificación de cada uno de los bits es la siguiente:

D7 = 1 : Indica el modo de trabajo de los 3 puertos A,B,C según los restantes bits

$D6 = 0$ Y $D5 = 0$ Indica que se ha seleccionado el modo de trabajo 0. Si hubiéramos escogido el modo 1, entonces $D6 = 0$ y $D5 = 1$ o el modo 2, donde $D6 = 1$ y $D5 = X$.

$D4 = 0$: Indica que el puerto se A ha programado como salida.

$D3 = 1$: Indica que los 4 bits más significativos del puerto C PCH (PC4 - PC7) están trabajando como entradas.

$D2 = 0$: Indica que el modo de trabajo del grupo B (PB y PCL) es 0.

$D1 = 1$: Indica que el puerto B será manejado como entrada.

$D0 = 1$: Indica que los 4 bits menos significativos del puerto C PCL (PC0 - PC3) están trabajando como entradas.

Para que la velocidad inicial del motor sea cero se debe enviar un *cero digital* al DAC0808 y esto se logra colocando 128 (número equivalente a 0 V analógicos por la configuración externa del DAC) en el puerto A. Este puerto A se encuentra determinado por la variable PORTA cuya dirección en el mapa de memoria es 500H. Es necesario que el convertidor analógico-digital ADC0808

no empiece a convertir en el momento de aplicar potencia por primera vez, ya que al inicio puede tener cualquier valor en las entradas y ese valor no deseamos manipular. Por este motivo se coloca un nivel lógico bajo en el bit de entrada SOC (Start of Conversion, la cual se activa en alto) del ADC.

Posteriormente el programa llama a la subrutina *APAGARCURSOR* la cual apaga el cursor en la pantalla. Se hace uso de las macro *WINDOW* que dibuja una ventana con atributo, de la macro *MARCO3* que dibuja un marco y de las macro *PUTMSG* que coloca en pantalla mensajes establecidos anteriormente. Pulsando cualquier tecla podemos avanzar en la pantalla, la cual nos conduce a una ventana de 2 opciones. Aquí el programa espera realizar una lectura del buffer de entrada desde el teclado y empieza a comparar. Si se pulsa la tecla 1 el programa de control hace un llamado al procedimiento de control de velocidad llamado *CONTROLVEJ*. Si se pulsa la tecla 2 el programa de control implementado nos lleva a la ejecución del procedimiento de control de posición llamado *CONTROLPOS*. Si se pulsa la tecla ESC el programa automáticamente limpia la pantalla a través de la macro *CLRSCR* y llama a la subrutina *ENCENDERCURSOR*. La subrutina *ENCENDERCURSOR* coloca el tamaño del cursor y lo hace visible saliendo inmediatamente al prompt del sistema operativo.

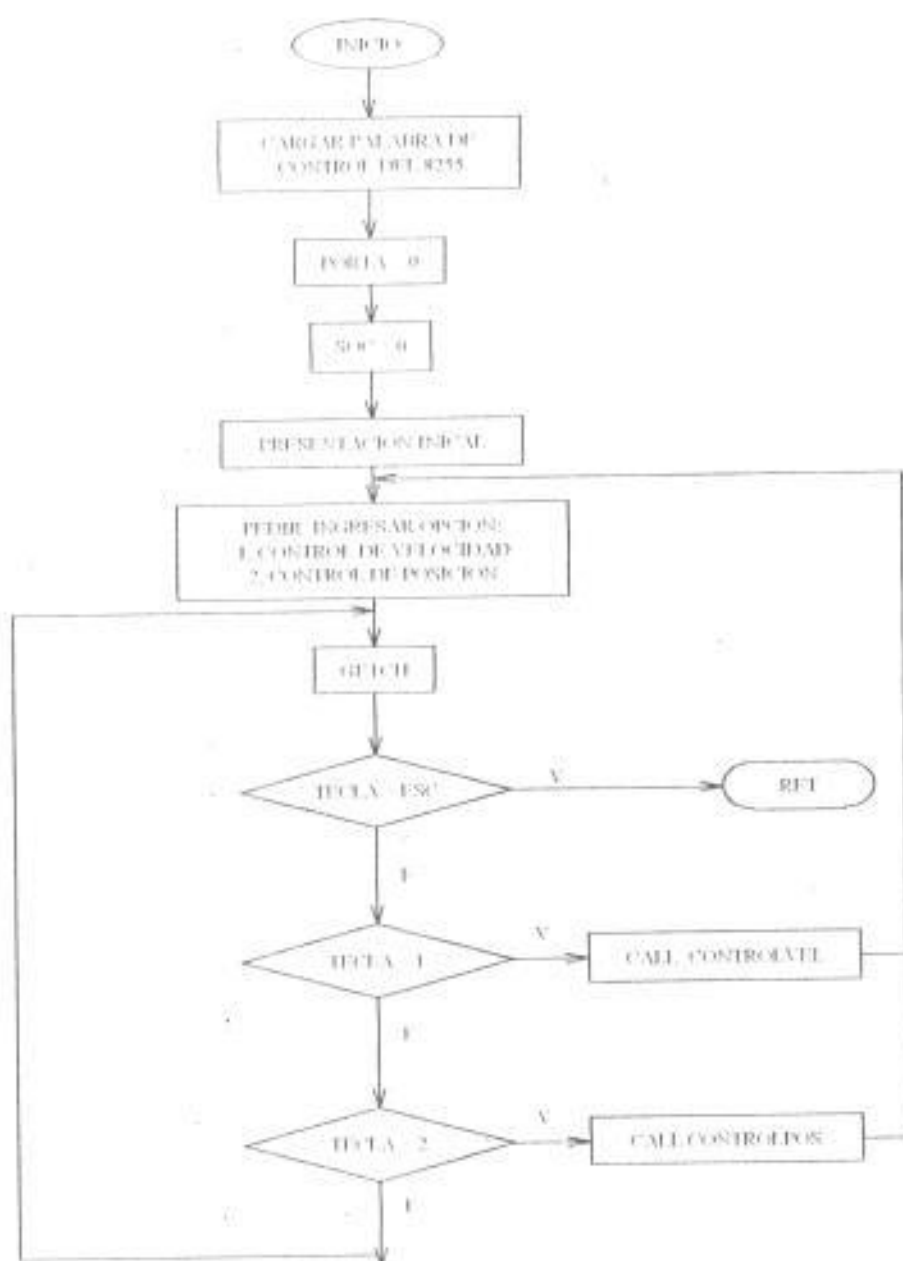


Figura 11.1 Diagrama de flujo del programa principal

11.3. IMPLEMENTACION DE LAS MACROS

En la implementación del programa de control es necesario recurrir a diferentes tipos de macros, las cuales nos facilitaran el trabajo en la estructuración del programa. Algunas de estas macros también hacen llamados a subrutinas específicas así como también a otras macros internas. Las macros utilizadas en el programa son las siguientes:

IMPRIMIRNUM

PUTCH

PUTCHH

CLRSCR

WINDOW

PUTMSG

GETCH

GOTOXY

MARCO

MARCO1

MARCO2

MARCO3

INGRESARNUM

INGRESARNUM2

INGRESARFRAC

INGRESARFRACI

CHRATRIB

LEEPOSICION

IMPRIMIRNUM : Es una macro que permite imprimir un número en pantalla de hasta cinco cifras, cuyo valor máximo es $2^{\text{exp}(16)} - 1 = 65535$. Esta macro hace uso de la subrutina *PIMPRIMIRNUM*. El número a mostrar en pantalla se lo designa como NUM y se lo coloca en la variable NUMIMPR. NUMDIG es la variable que contiene el número de dígitos de la variable NUM. El siguiente conjunto de instrucciones componen la macro analizada:

```

MOV AX, NUM
MOV NUMIMPR, AX
MOV BX, NUMDIG
CALL PIMPRIMIRNUM

```

PUTCH : Esta macro escribe un carácter con atributo en la posición actual del cursor. En el registro AL se coloca el código ASCII (variable C) del carácter a imprimir.

En esta macro aplicamos la interrupción 10H BIOS que se utiliza para los servicios de video porque controla la pantalla de video en el sistema. Esta instrucción 10H utiliza el registro AH para seleccionar el servicio de video. El número de este servicio de video es 09H con el cual se escribe un carácter con atributo en la posición actual del cursor. En el registro BH se coloca el número de la página, por eso ponemos un 0 que indica el seteo de la página activa. Esta página va con atributo blanco sobre negro (0FH si éste fuera el caso) o negro sobre blanco (70H si este fuera el caso). En el registro CX se coloca el número de caracteres a escribir, es decir el número de repeticiones que en este caso es 1.

```
MOV AL,C
MOV AH,9
MOV BH,0
MOV BL,ATRIBUTO
MOV CX,1
INT 10H
```

PUTCHI : Esta macro realiza la misma función que la anterior, con la salvedad que después de imprimir un carácter con atributo en pantalla ubica el cursor en la

posición siguiente. En esta macro se utilizan dos servicios de video: el 02H y el 03H. Con el 03H leemos la posición del cursor actual, incrementamos la columna actual que se encuentra en el registro DL para seleccionar la posición del cursor con el servicio de video 02H.

```

PUTCH C
MOV AH,3
INT 10H
INC DL
MOV BH,0
MOV AH,2
INT 10H
PUTCH 32

```

CLRSCR: Es una macro cuya función principal es la de limpiar pantalla. Hace uso de la macro **WINDOW** y **GOTOXY**. Realiza también un llamado a la subrutina *ENCENDERCURSOR*.

```

WINDOW 0,0,79,24, BLACK, WHITE
GOTOXY 0,0

```

CALL ENCENDERCURSOR

WINDOW : Esta macro dibuja una ventana con atributo, colocando las coordenadas de los extremos de la ventana (C1,F1 Y C2,F2) en los registros correspondientes. Se coloca el atributo del carácter en BH.

```
CHRATTRIB BACK,FORE
```

```
MOV CL,C1
```

```
MOV CH,F1
```

```
MOV DL,C2
```

```
MOV DH,F2
```

```
MOV AX,0600H
```

```
MOV BH,ATRIBUTO
```

```
INT 10H
```

PUTMSG : Esta macro pone mensaje carácter por carácter. C= número de columna, F= número de fila y MSG= el número de mensaje (A0,A1,etc.)

```
MOV DL,C
```

```

MOV D1,F
LEA SI,MSG
CALL PPMUTSG

```

GETCH: Esta macro me permite ingresar por teclado sin eco, es decir ingreso por teclado pero no muestra en pantalla. Lee el buffer de entrada del teclado y verifica si es funcional o no la tecla pulsada. El diagrama de flujo se muestra en la siguiente figura:

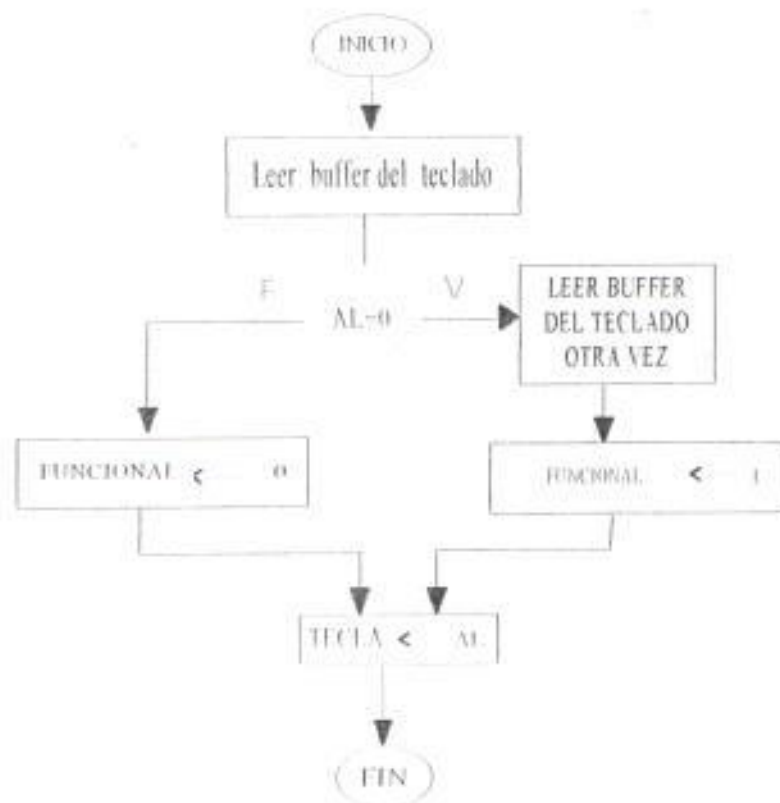


Figura 11.2 Diagrama de flujo de la macro 'GETCH'

```

LOCAL ENDI
MOV AH,7
INT 21H
MOV FUNCIONAL,0
CMP AL,0
JNZ ENDI
MOV AH,7
INT 21H
MOV FUNCIONAL,1
ENDI MOV TECLA,AL

```

GOTOXY : Esta macro posiciona el cursor en la pantalla. Se coloca la **COLUMNA** y **FILA** a la cual queremos enviar al cursor. **FILA** debe estar en registro **DH** y **COLUMNA** debe estar en registro **DL**.

```

MOV CX,FILA
MOV DX,COLUMNA
MOV AH,02H

```



```

MOV BH,00H
MOV DH,CL
INT 10H

```

MARCO : Esta macro solamente dibuja el marco de las ventanas de las presentaciones de pantalla del programa, donde (X1,Y1) es la esquina superior izquierda del marco, (X2,Y2) es la esquina inferior derecha y (T1,T2,T3,T4,T5,T6,T7,T8) representan los códigos ASCII de los caracteres utilizados como marco.

```

LOCAL HORTZ, VERTI, LIMIT1, LIMIT2
MOV AX,X1
MOV SI,AX
HORTZ: NOP
GOTOXY SI,Y1
PUTCH T5
GOTOXY SI,Y2
PUTCH T6
MOV AX,X2
INC SI

```

```
CMP    SLAX  
JLE    HORTZ  
MOV    AX,Y1  
MOV    SLAX
```

```
VERTI: NOP
```

```
GOTOXY X1,S1  
PUTCH  T7  
GOTOXY X2,S1  
PUTCH  T8  
MOV    AX,Y2  
INC    SI  
CMP    SLAX  
JLE    VERTI  
GOTOXY X1,Y1  
PUTCH  T1  
GOTOXY X1,Y2  
PUTCH  T4  
GOTOXY X2,Y1  
PUTCH  T2  
GOTOXY X2,Y2
```

```
PUTCH T3
MOV AX,X1
CMP AX,X2
JNE LIMIT1
GOTOXY X1,Y1
PUTCH T7
GOTOXY X1,Y2
PUTCH T7
JMP LIMIT2
```

```
LIMIT1: NOP
MOV AX,Y1
CMP AX,Y2
JNE LIMIT2
GOTOXY X1,Y1
PUTCH T5
GOTOXY X2,Y1
PUTCH T5
```

```
LIMIT2: NOP
```

MARCO1 : Esta macro cumple la misma función que la macro anterior. Utiliza un marco simple. Z, ?, Y, @, D, 3, son los códigos ASCII de los caracteres puestos como marco. A, B, C, D, son las coordenadas de las esquinas del marco:

```
MARCO A,B,C,D,'Z','?','Y','@','D','D','3','3'
```

MARCO2 : Este macro es la misma que MARCO1 solamente que utiliza otro tipo de marco.

```
MARCO A,B,C,D,'T','<','<','H','M','M','>','>
```

MARCO3 : Esta macro utiliza las mismas funciones que MARCO1, excepto que utiliza diferentes tipos de caracteres gráficos.

```
MARCO A,B,C,,D,219,219,219,219,223,220,219,219
```

INGRESARNUM : Esta macro me permite ingresar un número NUM por teclado con NUMDIG dígitos. Llama a las subrutinas ENCENDERCURSOR, PINGRESARNUM y APAGARCURSOR.

```

MOV AX, NUMDIG
MOV CONT, AX
CALL ENCENDERCURSOR
CALL PINGRESARNUM
CALL APAGARCURSOR
MOV AX, NUMIMPR
MOV NUM, AX

```

INGRESARNUM2. Esta macro realiza las mismas funciones que la anterior. La diferencia radica en que ésta es utilizada para manejar números negativos. La macro reconoce el signo (-) y lo guarda en la variable BITSIGNO, la cual será utilizada en el procedimiento CONTROLPOS para determinar si el número ingresado es negativo o no. La macro permite el ingreso del dato que le precede al signo “-”. Además llama a las subrutinas ENCENDERCURSOR, PINGRESARNUM2 y APAGARCURSOR.

```

MOV AX, NUMDIG
MOV CONT, AX
CALL ENCENDERCURSOR
CALL PINGRESARNUM2

```

```
CALL APAGARCUSOR
MOV AX, NUMIMPR
MOV NUM, AX
```

INGRESARFRAC : La función de esta macro es la de ingresar una fracción. NUM1 representa el numerador y NUM2 representa el denominador.

```
CALL ENCENDERCURSOR
CALL PINGRESARFRAC
CALL APAGARCUSOR
MOV AX, NUMIMPR
MOV NUM1, AX
MOV AX, NUMIMPR2
MOV NUM2, AX
```

INGRESARFRAC1: Esta macro cumple el mismo propósito que INGRESARFRAC, con la salvedad que INGRESARFRAC1 permite el ingreso de la fracción KP en una posición de pantalla cuyas coordenadas son : columna : 60 y fila : 14, dentro de la subrutina de control de posición CONTROLPOS.

```
CALL ENCENDERCURSOR
MOV COORDX,60
MOV COORDY,14
CALL PINGRESARFRAC
CALL APAGARCURSOR
MOV AX,NUMIMPR
MOV NUM1,AX
MOV AX,NUMIMPR2
MOV NUM2,AX
```

CHRTRIB : La función de esta macro es la de cargar el atributo. La variable **ATRIBUTO** contiene los atributos de todo lo que se vaya a 'imprimir'. El término 'imprimir' se refiere a mostrarlo en pantalla.

Utiliza los parámetros **BACK** y **FORE**. **BACK** es el fondo de la pantalla, y **FORE** es el relieve de la misma.

El diagrama de flujo se muestra en la siguiente figura.

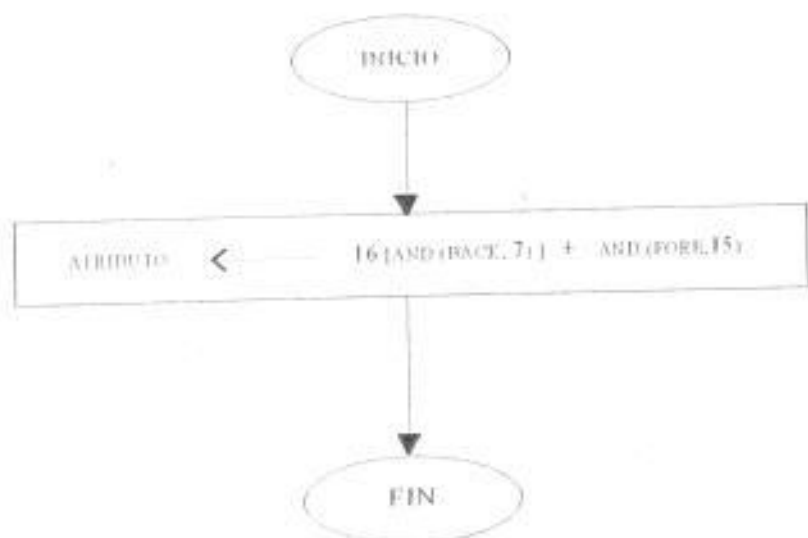


Figura 11.3 Diagrama de flujo de la macro "CHRATRIB"

```

MOV AX,BACK
AND AX,7
MOV CL,4
SHL AX,CL
MOV BX,FORE
AND BX,0FH
ADD AX,BX
MOV ATRIBUTO,AL
  
```


LEEPOSICION : Esta macro lee el puerto C de la interface PPI 8255; es decir lee la información proveniente del codificador de eje. Esta información es la posición actual del motor, la cual es convertida de código GRAY a código BINARIO por medio de una tabla de conversión de 32 posiciones ya que existen 5 líneas de entrada provenientes del codificador. Esta tabla de conversión fue obtenida a partir de datos experimentales realizados con el codificador de eje y el computador. POSACTUAL es una variable que almacena la posición actual del motor en grados. Esta variable es la que va actualizándose a medida que avanza el motor hacia su posición final por medio de la tabla de conversión como muestra el diagrama de flujo en la siguiente figura:

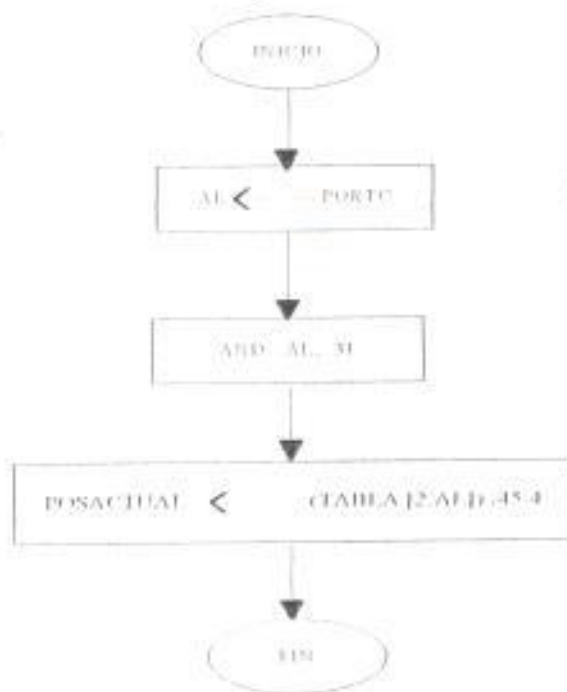


Figura 11.4 Diagrama de flujo de la macro "LEEPOSICION"

```
MOV DX,PORTC
IN AL,DX
AND AL,31
MOV AH,0
ADD AX,AX
MOV SI,AX
MOV AX,TABLA[SI]
MOV CX,45
MUL CX
MOV CX,4
DIV CX
MOV POSACTUAL,AX
```

11.4. IMPLEMENTACION DEL ALGORITMO DE *MUESTREO*

La figura 11.6 muestra el diagrama de flujo del algoritmo *MUESTREO*. El algoritmo de muestreo implementado en el computador se basa en *promedios* realizados a un número determinado de muestras con el propósito de que la velocidad actual se acerque lo más posible al valor deseado. Luego de

determinar el número de muestras necesario se carga este valor en una variable que se llama REG5

Posteriormente se obtiene el valor de una muestra (variable MUESTRA) mediante un llamado a la subrutina OBTIENE. Mas adelante se explicará con mayor detalle la subrutina OBTIENE. Este llamado es continuo hasta que se alcance el valor de REG5. Cada vez que se obtiene una muestra ésta se va acumulando y en el momento que se alcanza el valor contenido en REG5 se realiza un promedio de las muestras acumuladas y este promedio es almacenado en una variable denominada VELACTUAL.

En este algoritmo de muestreo existe una variable TOTPROM que acumula el total de las últimas 32 mediciones de la variable VELACTUAL. Estas 32 mediciones se almacenan en un arreglo llamado "PROMED" direccionado por la variable REG3 que solo puede adquirir valores entre 0 y 31 (índices del arreglo). Cuando se obtiene una nueva VELACTUAL después del muestreo se reemplaza en la tabla "PROMED", el valor actualmente direccionado por REG3 y se almacena en este lugar del arreglo el nuevo VELACTUAL. Luego se incrementa REG3 para que en el siguiente proceso de muestreo se reemplace el siguiente elemento del arreglo "PROMED" (reemplazo uniforme).

Finalmente se hace un promedio de los valores acumulados en la variable TOTPROM y su resultado se lo almacena en la variable VELACTUAL. El valor de esta VELACTUAL representa la velocidad actual digital, la cual será manejada por el programa de control de velocidad CONTROLVEL.

SUBROUTINA OBTIENE:

La subrutina OBTIENE envía un pulso de determinada duración a la entrada SOC (Start of Conversion) del convertidor analógico-digital para que convierta la muestra analógica en muestra digital.

El programa espera al ADC para que mande un nivel de voltaje alto (1 en binario) por la salida EOC que me indica que el dato muestreado ha sido convertido. Este dato (muestra de velocidad actual) en formato digital es ingresado al puerto B del PPI 8255 mediante la instrucción IN y almacenado posteriormente en una variable de memoria llamada MUESTRA.

La figura 11.7 muestra el diagrama de flujo de la subrutina OBTIENE.

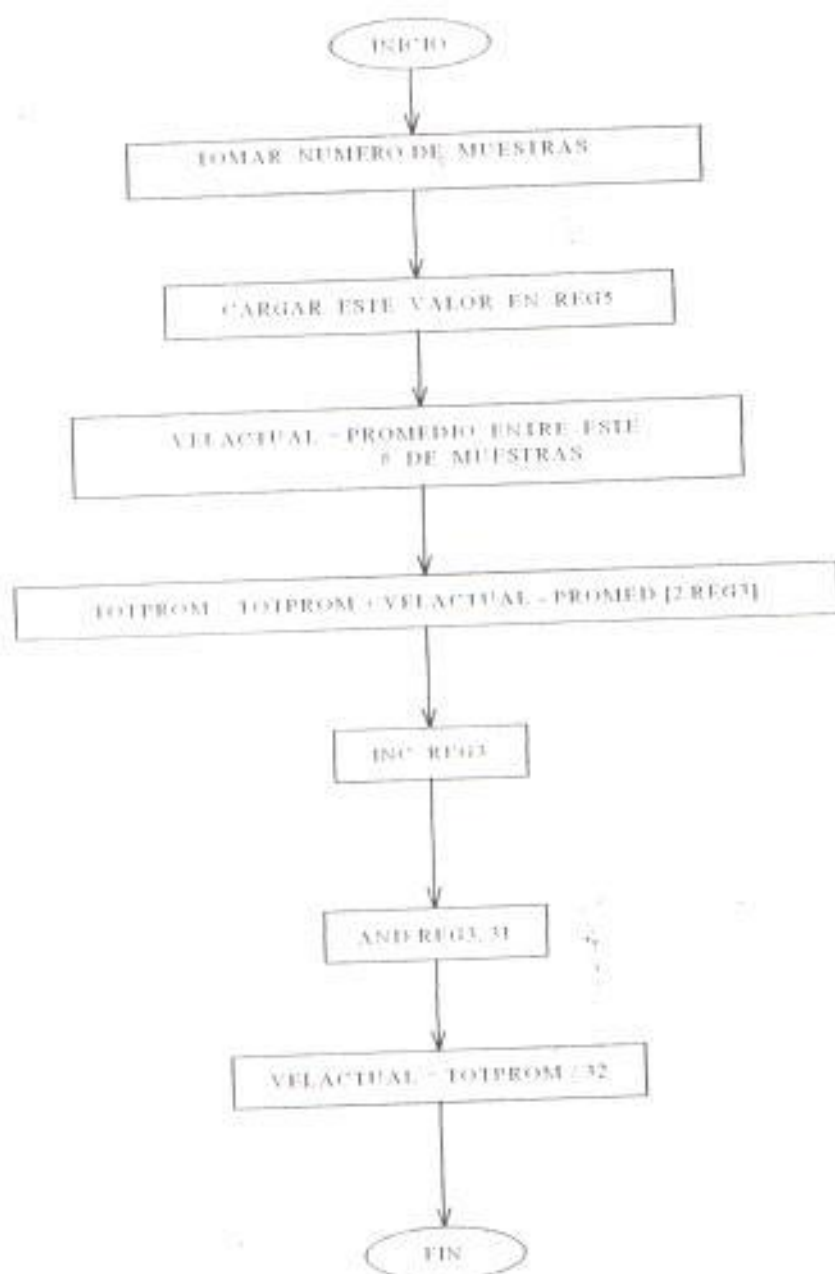


Figura 11.5 Diagrama de flujo del algoritmo "MUESTREO"

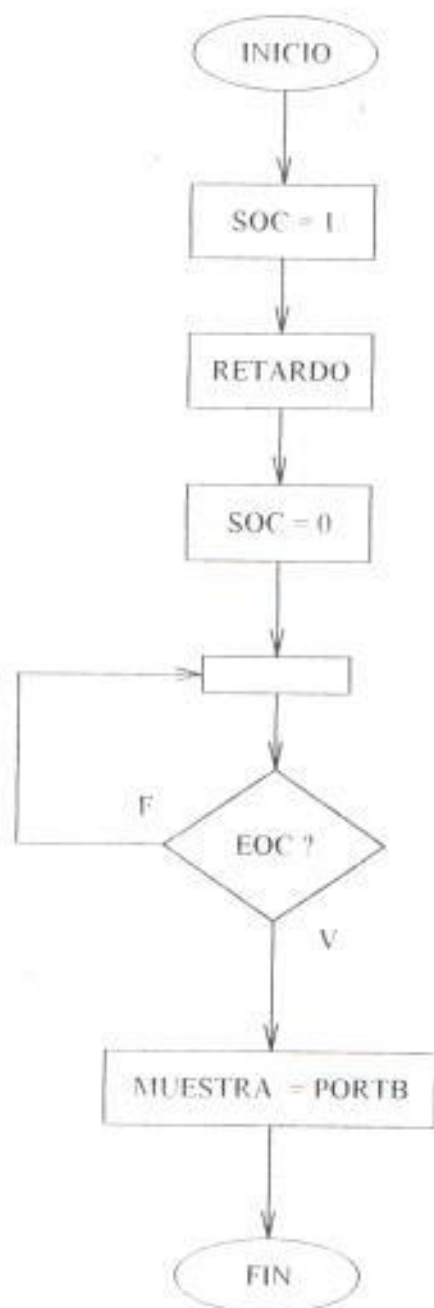


Figura 11.6 Diagrama de flujo de la subrutina "OBTIENE"

11.5. IMPLEMENTACION DEL ALGORITMO DE CONTROL DE VELOCIDAD

Inicialmente el programa pide ingresar una de 4 opciones del controlador: controlador proporcional puro (Control P), controlador proporcional-integral (Control PI), controlador proporcional-derivativo (Control PD) y controlador proporcional-integral-derivativo (Control PID).

En el diseño del programa se ha recurrido al uso de banderas de activación de controladores. Si la opción seleccionada es 1, es decir se elige un control P, el programa encera las banderas FLAGINTEG y FLAGDERIV. Si se elige un control PI, el programa activa la bandera FLAGINTEG y encera la bandera FLAGDERIV. Si se elige un control PD, el programa encera la bandera FLAGINTEG y activa la bandera FLAGDERIV.

En el caso de elegir un control PID, el programa activa la banderas FLAGINTEG y FLAGDERIV. Si se activa la bandera FLAGINTEG (FLAGINTEG=1), la contribución integral del controlador existe, de igual forma para la bandera FLAGDERIV. Se imprimen las pantallas respectivas de presentación del control de velocidad con los valores iniciales de velocidad de

referencia, constantes KP, KI, KD. A continuación se enceran todas las variables que se relacionan con este control de velocidad.

El primer paso para obtener la señal de retroalimentación hacia la interface digital es llamar a una subrutina de muestreo mediante la instrucción CALL. Este procedimiento de muestreo se explicó en la sección anterior.

La velocidad actual digital se la compara con la referencia digital produciendo un error. La señal de error es la desviación del valor actual de la variable del proceso (VELACTUAL) con respecto al valor deseado (REFERENCIA). Esta señal de error es la información de entrada al controlador. Una vez que el error ha sido determinado se lo almacena en una variable llamada ERROR. Para el siguiente valor de error, el valor anterior (ERROR) es guardado en otra variable llamada ERRORANT. Cada una de las contribuciones del controlador, que son la proporcional, integral y derivativa, se irán calculando por separado y adicionando en la variable DAC que representa el valor final del controlador y que ingresa al convertidor digital-analógico, por lo que antes de iniciar el cálculo de cada uno de los términos de dicho controlador, dicha variable debe ser encerada. A esta variable DAC (salida del controlador) se le ha adicionado un valor de 20 para vencer más rápidamente la inercia del motor. Si el contenido de DAC es mayor a

255, entonces $DAC=255$ (máximo valor de entrada al convertidor digital analógico), caso contrario DAC es menor a 255. Este valor de DAC que se encuentra entre 0 y 255 es luego convertido a un valor entre 128 y 255, el cual es requerido por la interface acondicionadora de salida que genera un voltaje positivo cuando la entrada digital se encuentra entre 128 y 255, pues en control de velocidad se trabaja solamente con voltajes positivos. Esto se logra mediante las siguientes dos líneas de instrucciones:

SHR AX, 1

ADD AX, 128

Con la instrucción SHR se desplaza el valor presente en AX un bit a la derecha logrando con esto que el número sea dividido para dos, además se adiciona al valor resultante del desplazamiento en AX el 128 (equivalente a 0 V analógicos y 0 de velocidad). Esta es la señal neta de control que será enviada al elemento final de control. El valor almacenado en AX se lo guarda finalmente en la variable DAC y se direcciona al puerto A para su posterior envío. El programa empieza a revisar el buffer del teclado, pudiendo cambiar el operador del sistema los valores de REFERENCIA, de KP, KI y KD, estos tres últimos ingresados en forma fraccionaria. De otro modo se puede salir del procedimiento de velocidad pulsando

la tecla M o directamente salir al sistema operativo mediante la tecla ESC.

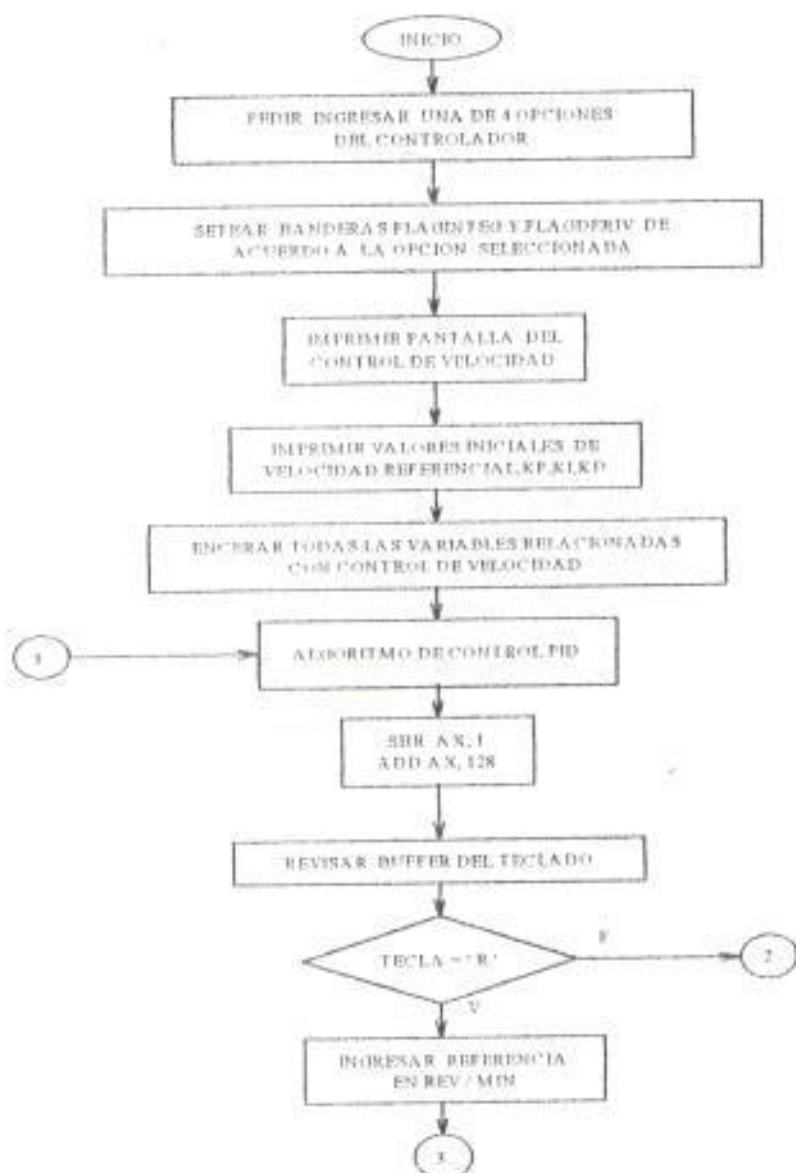


Figura 11.7 Diagrama de flujo del algoritmo de Control de Velocidad

“CONTROLVEL” (PARTE A)

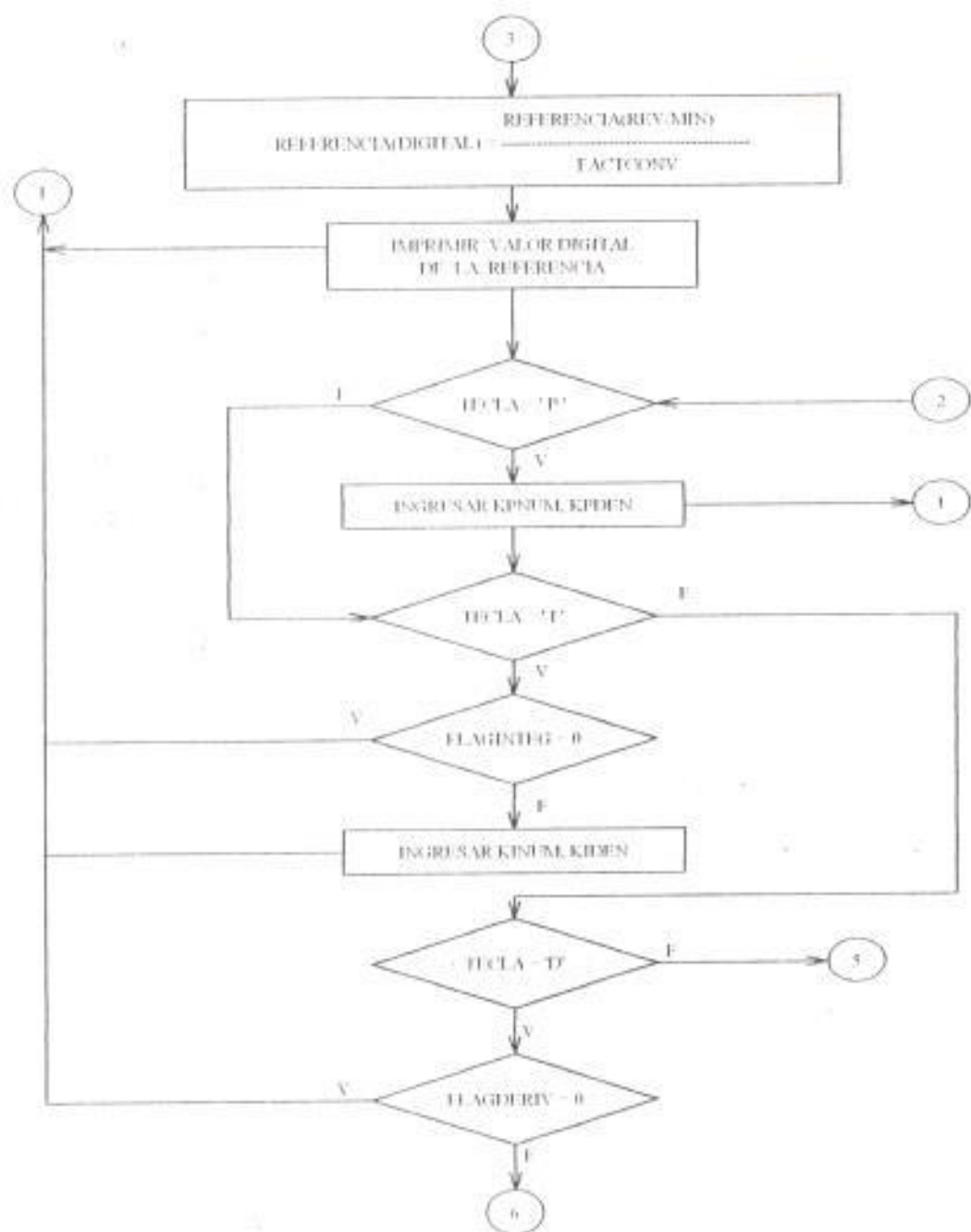
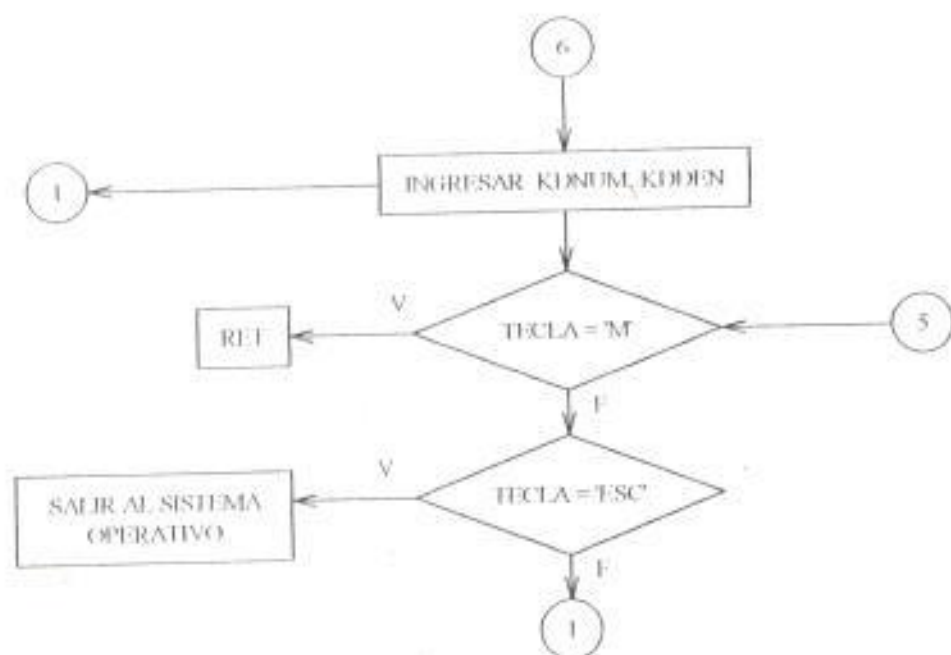


Figura 11.8 Diagrama de flujo del algoritmo de Control de Velocidad
"CONTROLVEL" (PARTE B)



**Figura 11. 9 Diagrama de flujo del algoritmo de Control de Velocidad
"CONTROLVEL" (PARTE C)**

En este control digital de velocidad se ha implementado un controlador PID. La expresión analítica del algoritmo en el dominio del tiempo es:

$$VEL(t) = Ke(t) + \frac{K}{T_I} \int e(t) dt + K T_D \frac{de(t)}{dt}$$

Donde

VEL(t)	=	salida de velocidad del controlador
e(t)	=	r(t) - c(t) = señal de error
r(t)	=	valor de referencia = valor deseado de velocidad
c(t)	=	valor actual de velocidad
K	=	ganancia proporcional
T _i	=	tiempo integral
T _d	=	tiempo derivativo

Podemos colocar también la anterior ecuación diferencial continua de la siguiente forma:

$$VEL(t) = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t)$$

Donde

$$K_p = K$$

$$K_i = K/T_i$$

$$K_d = K T_d$$

En un microcomputador no podemos ingresar ecuaciones de tipo continua ya que solo observa acciones del mundo exterior una vez cada ciclo. Por esta razón la ecuación diferencial continua debe ser convertida a su forma discreta:

$$VEL_n = K_p e_n + K_i \sum e_n \Delta t + K_d \frac{\Delta e}{\Delta t}$$

Donde:

Δt = intervalo de muestreo

Por lo tanto, el microcomputador calculará VEL_n de acuerdo a la siguiente fórmula:

$$VEL_n = K_p e_n + \sum K_i e_n + K_D [e_n - e_{n-1}]$$

$$e_n = r_n - c_n$$

r_n = valor de referencia

c_n = salida digitalizada del sistema controlado

K_p = coeficiente o ganancia proporcional

K_i = coeficiente o ganancia integral = $K_p \Delta t / T_i = K_i \Delta t$

K_D = coeficiente o ganancia derivativa = $K_p T_D / \Delta t = K_2 / \Delta t$

Cabe mencionar que el algoritmo implementado en el programa de control es un PID. En la presentación de la pantalla de control de velocidad existen 4 opciones de controlar el motor, por eso nos hemos visto en la necesidad de dividir dicho controlador en flujogramas P, PI, PD, PID, siendo éste último el resultado de la combinación de los anteriores. A continuación se describen los algoritmos utilizados en el diseño de los controladores para el sistema digital.

11.5.1. ALGORITMO DE CONTROL P

La función del control proporcional es proveer a la salida un término de la forma:

$$VEL_i = K_p e_i$$

Aquí e_i es igual a la REFERENCIA menos VELACTUAL en el muestreo i , esta señal de error es multiplicada en cada muestra i por una constante definida convenientemente K_p . El microcomputador seguirá muestreando la señal de velocidad y obteniendo una señal correctora proporcional para el motor, hasta que la referencia sea igual a la velocidad actual (claro que esto nunca sucede en control proporcional, ya que siempre operará con una desviación). En la siguiente figura se muestra el diagrama de flujo del algoritmo de control P.

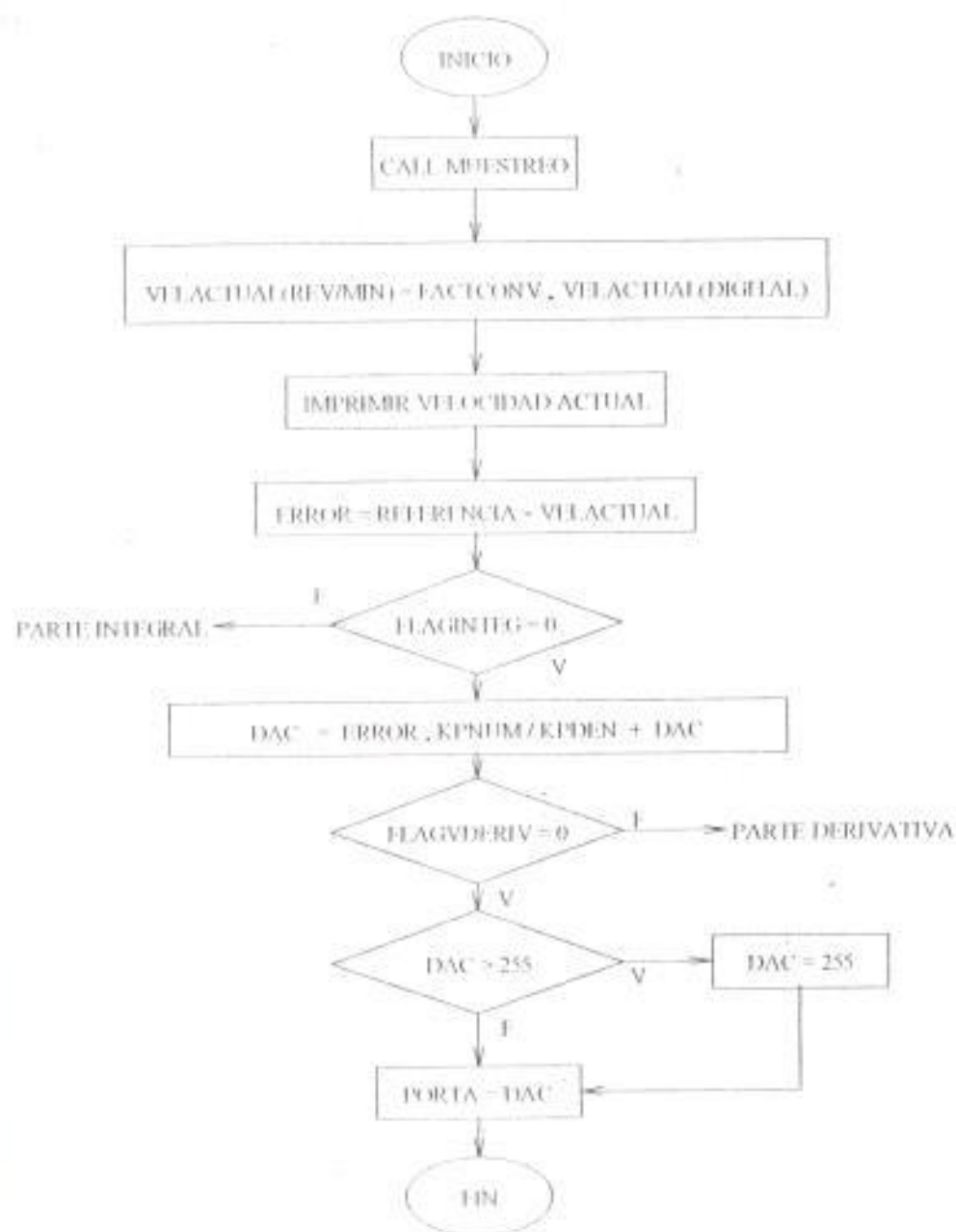


Figura 11.10 Diagrama de flujo del algoritmo "P"

11.5.2. ALGORITMO DE CONTROL PI

El modo de acción *integral* tiene por función realizar un reajuste de la salida del sistema controlador. La contribución integral se analiza de la siguiente manera:

$$V_{0i} = K_I e_i + V_{0i-1} \quad (1)$$

Esta ecuación nos indica que cuando la información de error es obtenida a partir de la comparación de la referencia con la velocidad actual, ésta se multiplica por una ganancia integral K_I y se la va acumulando en una variable denominada DAC que representaría a la salida V_{0i} . Esta variable DAC que está determinada en el programa de control de velocidad, cambia su contenido en base al nuevo valor calculado $K_I e_i$ en el muestreo i y en base al contenido de la información previa V_{0i-1} . En el algoritmo PI la fórmula asignada a INTEGRAL es: $INTEGRAL = INTEGRAL + ERROR$.

En nuestro controlador digital al inicio del programa la variable INTEGRAL es igual a 0 por lo que INTEGRAL será igual al contenido de la variable ERROR: $INTEGRAL = ERROR$. Siendo esta variable INTEGRAL(el error e_i en la ecuación 1) multiplicada por la constante K_I . El término $INTEGRAL \cdot K_I$ será almacenado en la variable DAC. Esta ganancia K_I está definida como un valor

fraccionario $K_I NUM / K_I DEN$, donde $K_I NUM$ es el numerador del valor fraccionario de la ganancia integral, mientras que $K_I DEN$ es el denominador. Haciendo una analogía el término $ERROR \cdot K_I$ vendría a ser igual a Vo_{i-1} . Vo_{i-1} es lo mismo que $K_I \cdot e_{i-1}$ por lo que $ERROR$ es e_{i-1} , con lo cual en el siguiente muestreo de la señal, la variable $INTEGRAL$ tomará el siguiente valor: (nuevo valor de $ERROR +$ el valor anterior de $ERROR$), por lo que la fórmula $INTEGRAL \cdot K_I$ en el algoritmo PI equivaldría a $(e_i + e_{i-1}) \cdot K_I$ y por consiguiente a $Vo_i - DAC$. Esta contribución integral debe ser adicionada al término proporcional analizado en la sección anterior para formar un controlador proporcional-integral (PI). El resultado de esta adición será almacenada en la misma variable DAC. La ecuación resultante del controlador PI tendría la siguiente forma: $VEL_i = K_p e_i + K_I e_i + Vo_{i-1}$. Haciendo una analogía con la ecuación implementada en el microcomputador:

$DAC = K_p NUM / K_p DEN \cdot ERROR + DAC(anterior)$, tendríamos que:

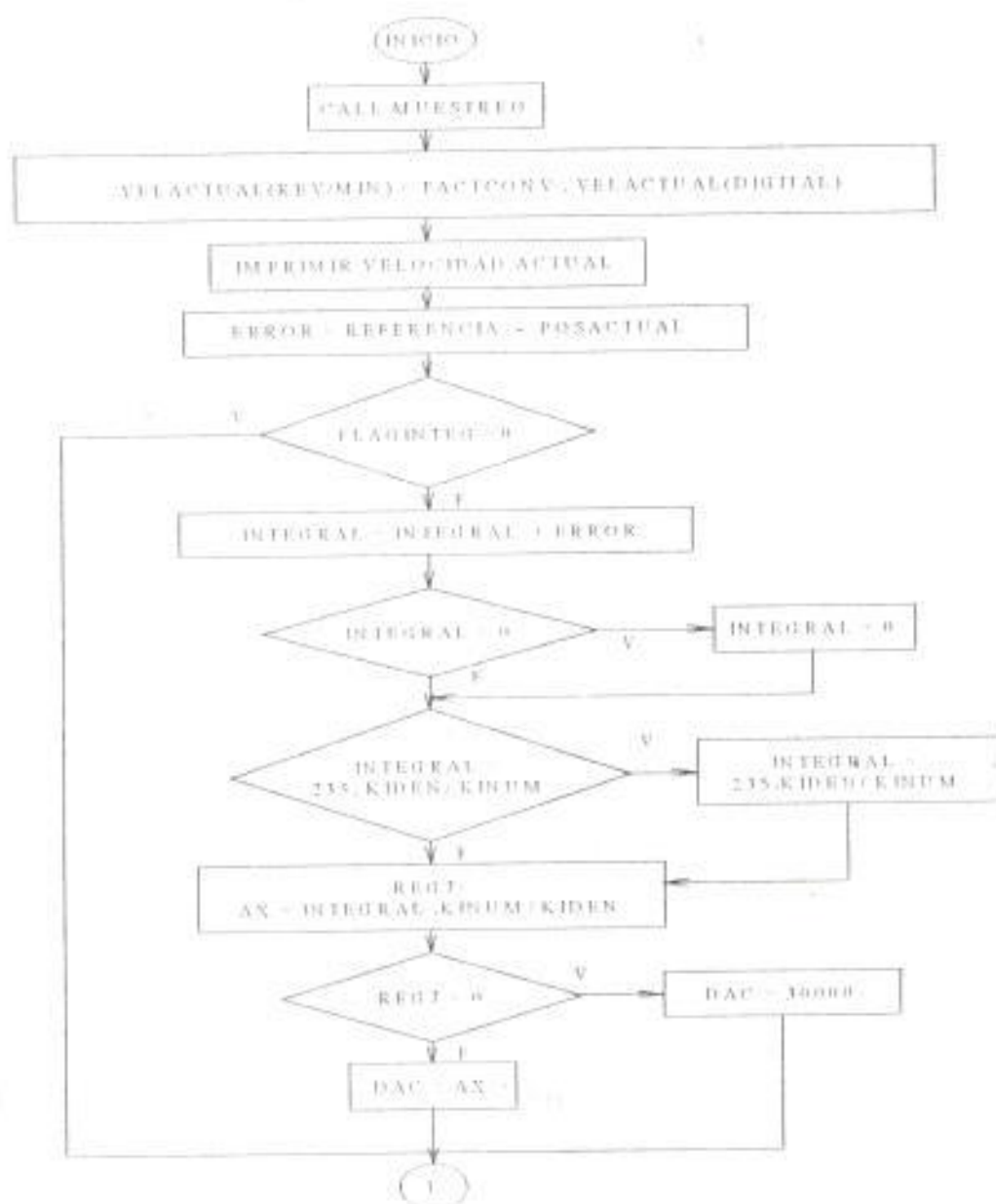
$$VEL_i = DAC$$

$$K_p = K_p NUM / K_p DEN$$

$$e_i = ERROR$$

$$K_I e_i + Vo_{i-1} = DAC(anterior)$$

En la siguiente figura se presenta el diagrama de flujo del algoritmo PI:



11.11 Diagrama de flujo del algoritmo "PI" (PARTE A)

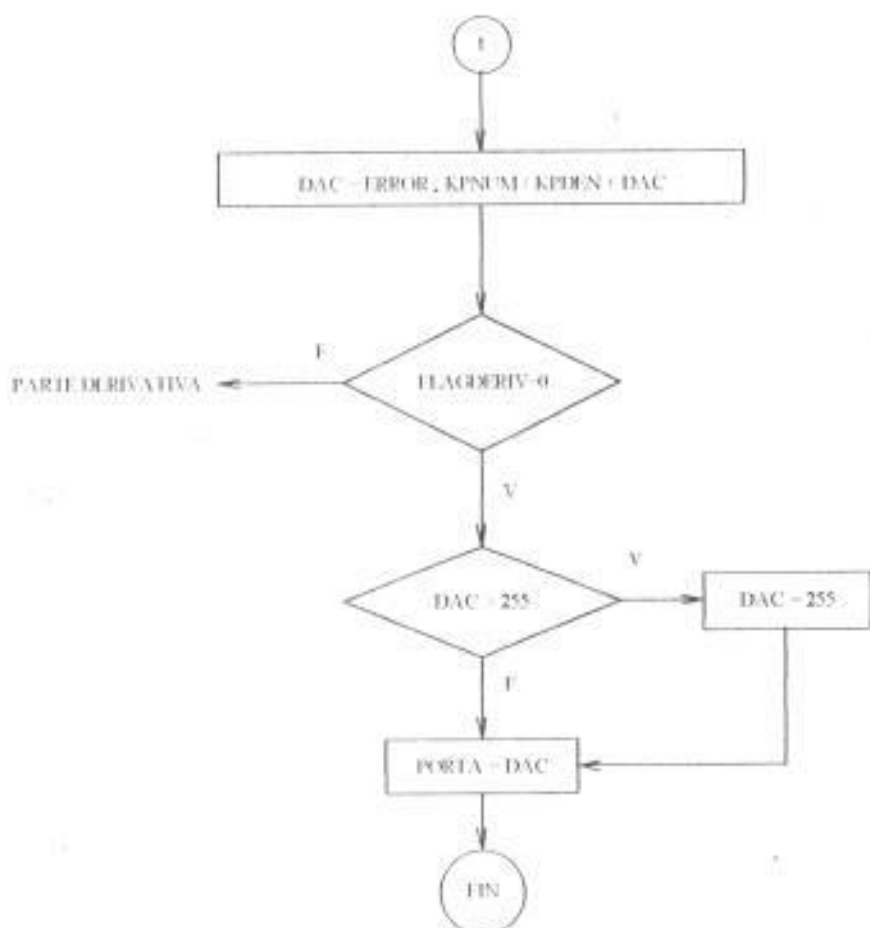


Figura 11.12 Diagrama de flujo del algoritmo "PI" (PARTE B)

11.5.3. ALGORITMO DE CONTROL PD

En este modo de control, la parte *derivativa* tiene por función proporcionar un

término a la salida del controlador de la forma:

$$V_{0i} = K_D(e_i - e_{i-1})$$

Esta ecuación implica que el valor del error correspondiente al muestreo anterior se resta del valor del error del muestreo presente, cuyo resultado se multiplica por la ganancia derivativa. Antes de realizar el cálculo de la contribución derivativa en el algoritmo PD se verifica el estado de la bandera FLAGDERIV que debe estar en 1. Luego el programa de control procede a calcular la parte derivativa y lo adiciona a la parte proporcional almacenando su resultado en DAC. La ecuación resultante de un controlador PD sería de la forma:

$$VEL_i = K_P e_i + K_D(e_i - e_{i-1})$$

De la misma forma que en el controlador PI, haciendo una analogía con la ecuación implementada en el microcomputador: $DAC = DAC + K_D NUM / K_D DEN \cdot (ERROR - ERRORANT)$, se tiene que:

$$\begin{aligned} VEL_i &= DAC \text{ (salida del controlador)} \\ K_P e_i &= DAC \text{ (parte proporcional)} \\ K_D &= K_D NUM / K_D DEN \\ (e_i - e_{i-1}) &= (ERROR - ERRORANT) \end{aligned}$$

La siguiente figura ilustra el flujograma del algoritmo PD.

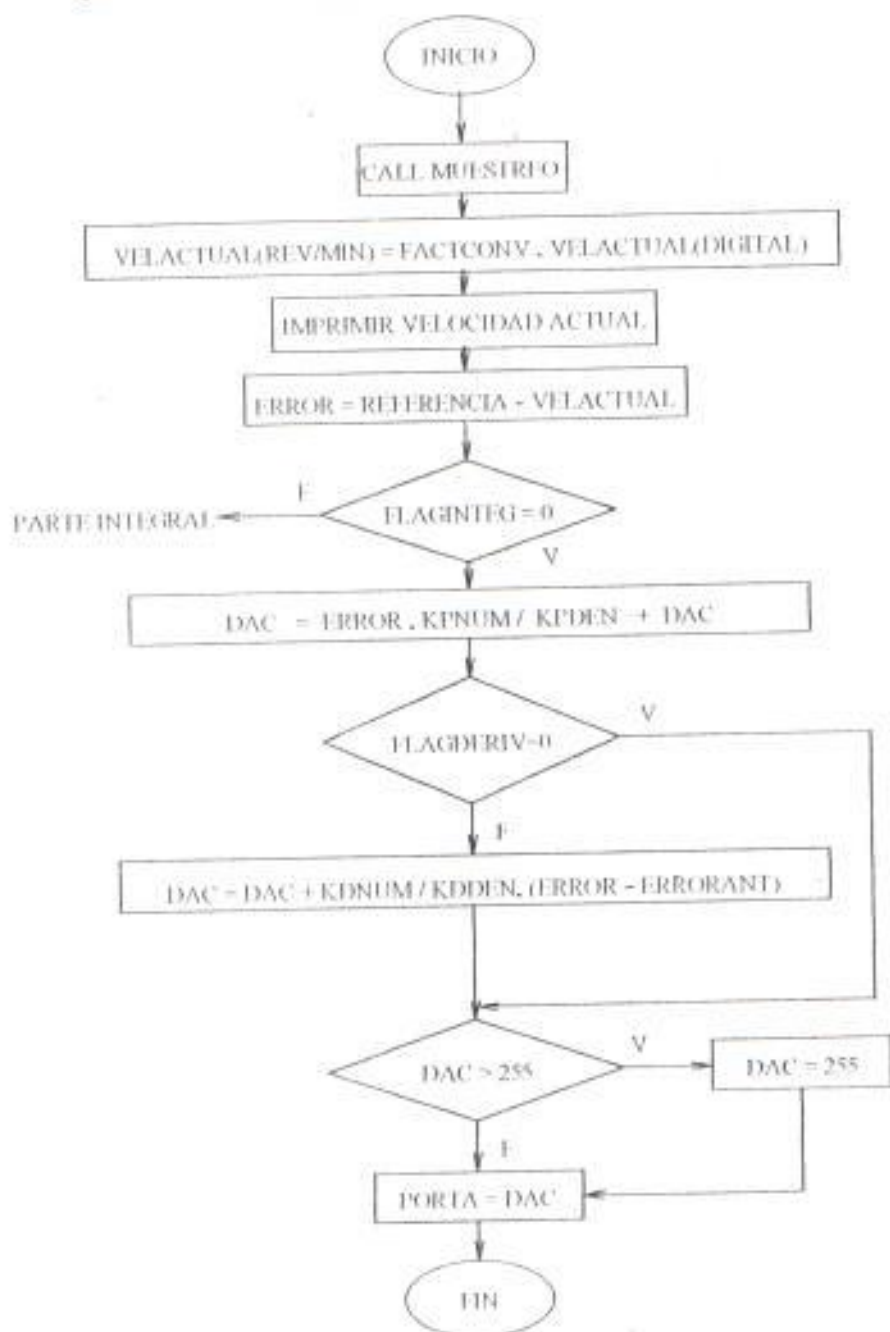


Figura 11.13 Diagrama de flujo del algoritmo "PD"

11.5.4. ALGORITMO DE CONTROL PID

En este modo de control, las contribuciones proporcional, integral y derivativa se adicionan para proporcionar la ecuación resultante del controlador PID:

$$VEL_i = K_p e_i + K_i(e_i + e_{i-1}) + K_D(e_i - e_{i-1})$$

Esta ecuación es implementada en el programa mediante una serie de combinaciones de fórmulas y variables, las cuales dan forma al diagrama de flujo del algoritmo de control PID que se muestra en la figura 11.14. El primer paso en el diagrama de flujo del algoritmo PID es obtener la variable controlada, variable de velocidad (señal del tacómetro), mediante el proceso de muestreo que fue explicado en secciones anteriores. Esta señal del tacómetro es una señal de voltaje cuyo valor es proporcional a la velocidad que desarrolla el motor. Este voltaje es luego utilizado por el ADC 0808 para suministrar el valor digital correspondiente a dicha velocidad que para efectos de mostrar en pantalla se multiplica por el factor de conversión determinado en el programa. Con este valor de velocidad se obtiene el error que manejará el algoritmo de control de la misma forma que los casos anteriores con la diferencia que ahora se encuentran activas las banderas de control integral y derivativo, con lo cual se utilizará en este tipo de control todos los flujogramas correspondientes.

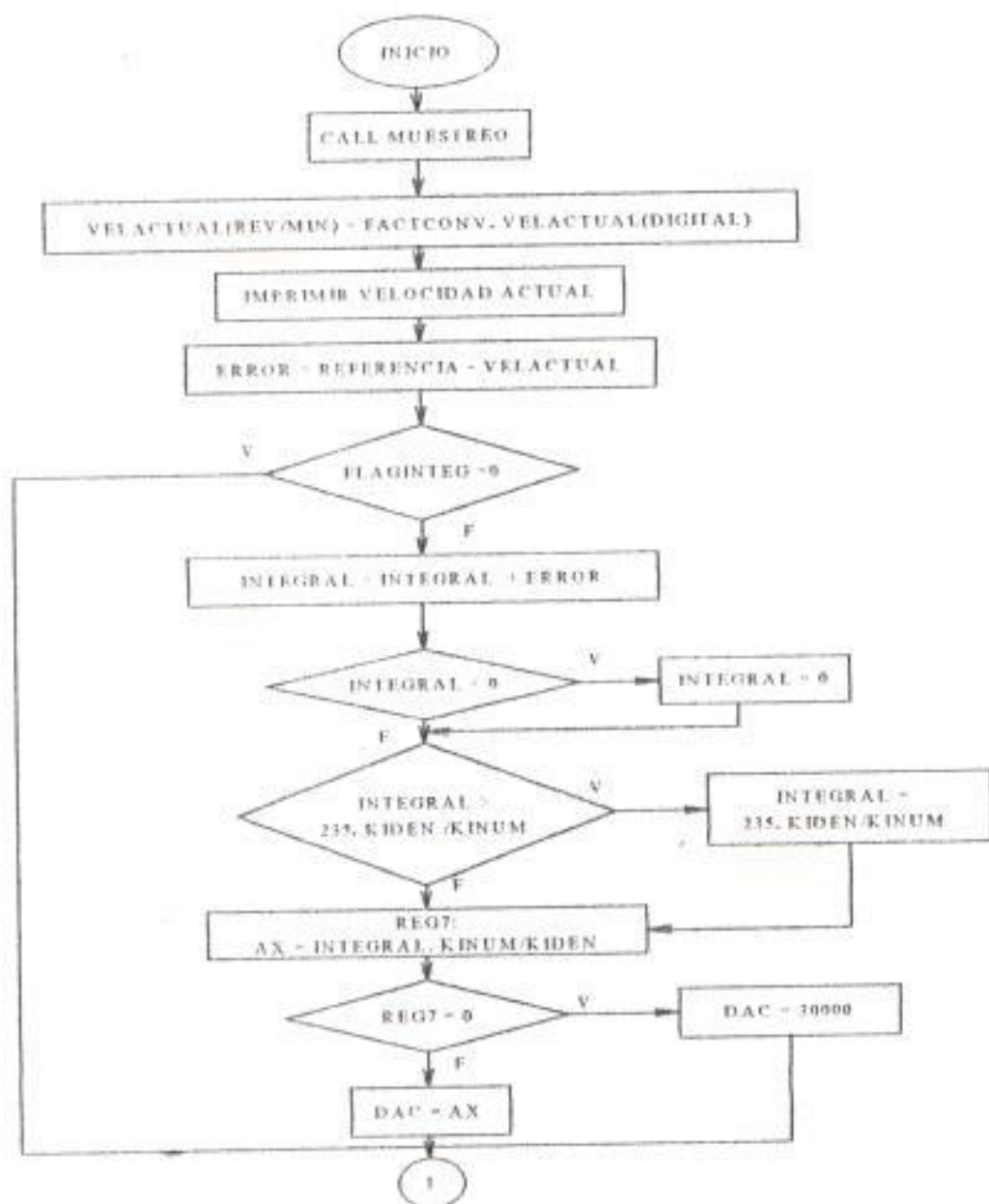


Figura 11.14 Diagrama de flujo del algoritmo "PID" (PARTE A)

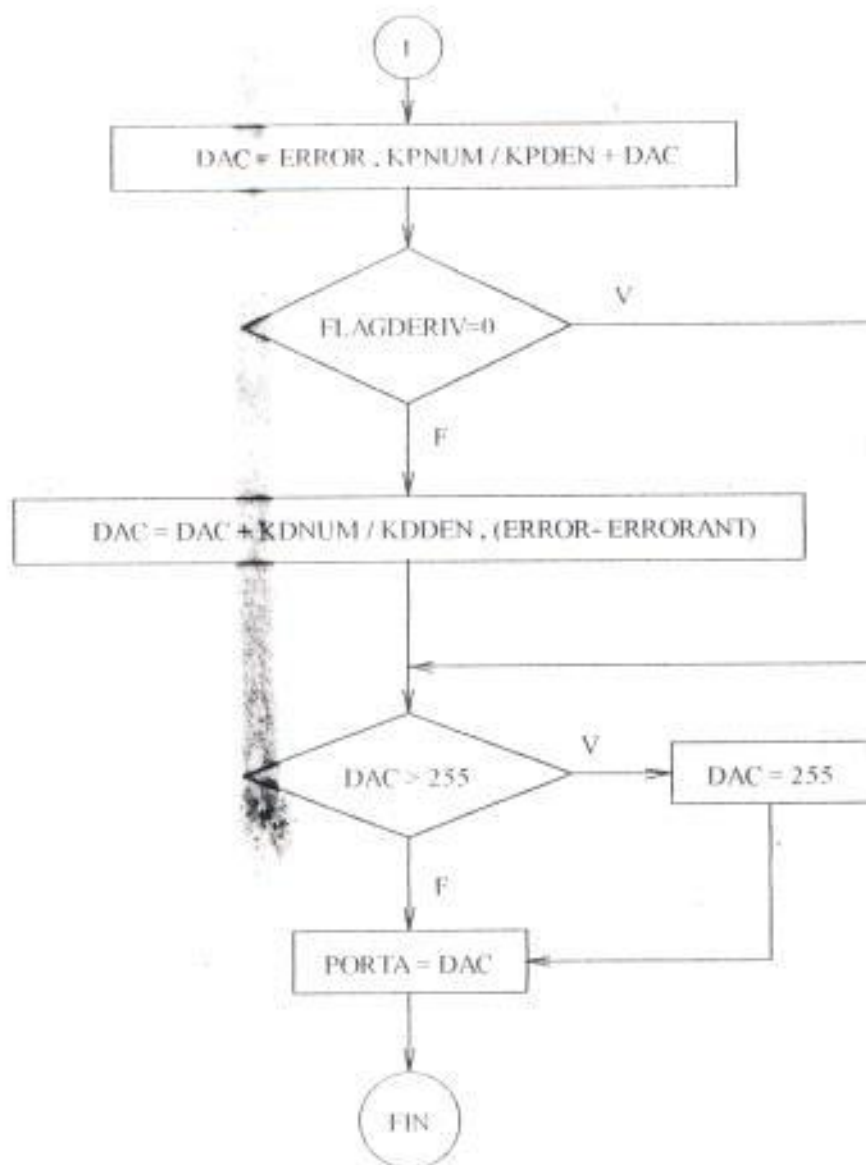


Figura 11. 15 Diagrama de flujo del algoritmo "PID" (PARTE B)

11.6. IMPLEMENTACION DEL ALGORITMO DE CONTROL DE POSICION

La figura 11.16 muestra el diagrama de flujo del algoritmo de control de posición.

El programa se inicia con la pantalla de presentación del control de posición. Una diferencia con el sistema de control de velocidad es que en lugar de leer la información de realimentación de velocidad presente en el puerto B (proveniente del tacómetro), el programa lee el puerto C donde ingresa la información proveniente de las líneas del codificador de eje (posición actual del motor) por medio de la macro LEEPOSICION.

La macro LEEPOSICION convierte el dato gray en binario y posteriormente en grados (variable POSACTUAL) para su impresión en pantalla. Para tal efecto se utiliza la tabla de conversión de 32 elementos que representan los 32 posibles cambios de posición. Cabe mencionar que el programa solo acepta valores de posición entre -359 y +359, por consiguiente solamente utilizará 3 dígitos.

La subrutina CONTROLPOS (Control de Posición) revisa el buffer del teclado para verificar el dato ingresado por el usuario. Si éste pulsa la tecla correspondiente al número "1", el programa pide ingreso del valor de posición requerida o REFERENCIA y llama a la subrutina RECTIFICA o RECTIFICA2

según sea el sentido de giro y el signo del error. Cuando el número ingresado es negativo (señala que la variable $\text{BITSIGNO} = 1$, lo que quiere decir que se ha ingresado por teclado el signo $^+ - ^-$) el programa salta al procedimiento **RECTIFICA**. De la misma forma salta al procedimiento **RECTIFICA** cuando el valor de **REFERENCIA** (variable que almacena el valor de posición requerida) es menor o igual a **POSACTUAL** (variable que almacena el valor de posición actual) lo que quiere decir que el error es negativo. En el caso de que el número ingresado **REFERENCIA** es mayor a **POSACTUAL**, el error es positivo, por lo tanto el programa salta al procedimiento **RECTIFICA2**. Estas dos subrutinas son las que en esencia hacen cambiar de posición al motor. Ambas serán explicadas con mayor detalle en la siguiente sección.

Si el usuario pulsa la tecla correspondiente al número "2", entonces el programa sale del procedimiento de control de posición y retorna al menú principal. Si no pulsa nada imprime la misma posición actual en la línea de posición requerida y revisa constantemente el buffer del teclado para verificar si existe algún valor ingresado. Si el usuario pulsa la tecla correspondiente al número "3" el programa pide ingreso de ganancia proporcional K_P , de esta manera se puede variar la constante de proporcionalidad y por consiguiente la ganancia del controlador. En consecuencia se podrá cambiar la respuesta del sistema de control de posición.

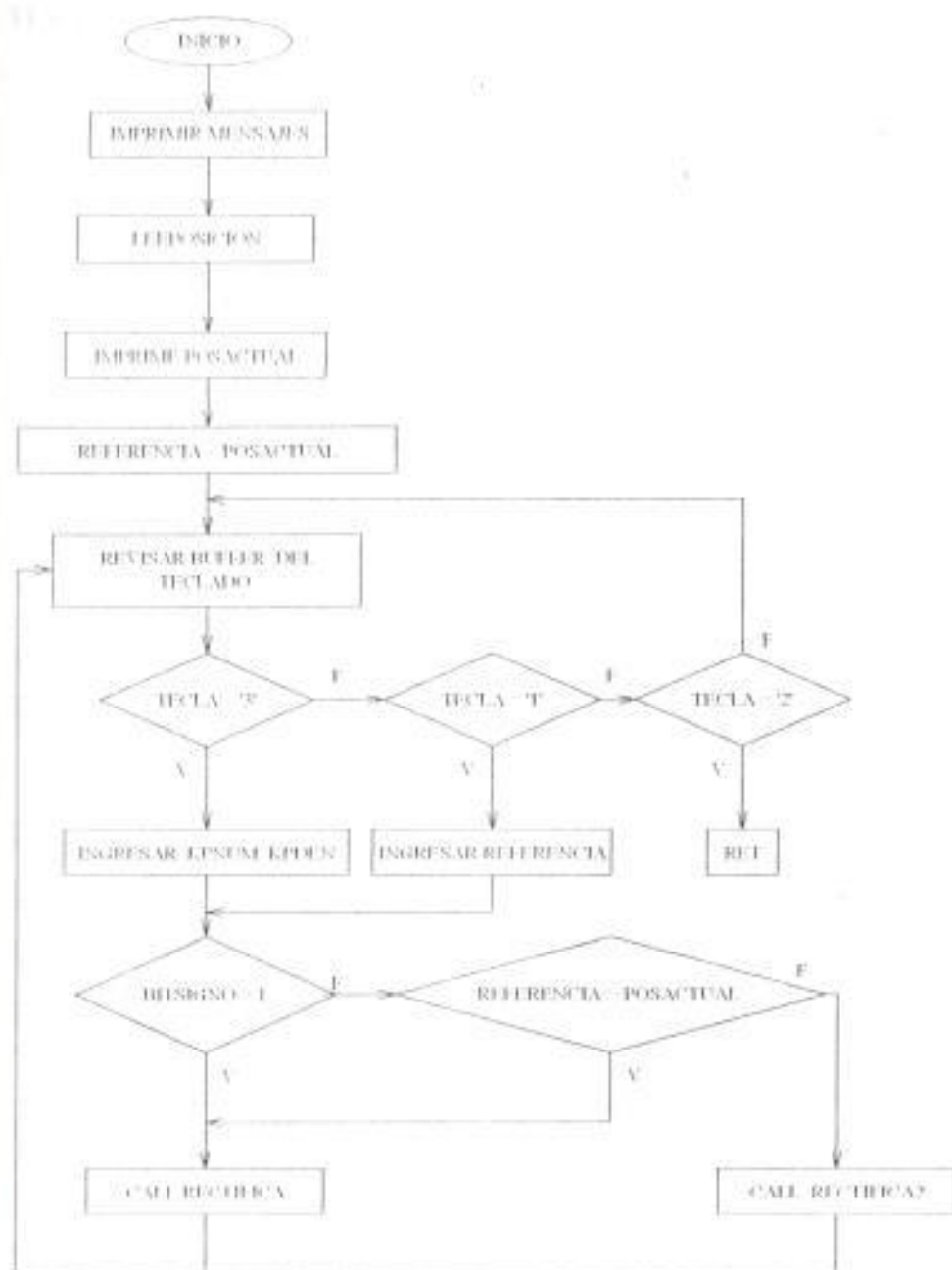


Figura 11. 16 Diagrama de flujo del algoritmo de Control de Posición

“CONTROLPOS”

11.6.1. ALGORITMO *RECTIFICA*

La figura 11.17 muestra el diagrama de flujo del algoritmo *RECTIFICA* y la figura 11.18 muestra el diagrama de flujo del algoritmo *RECTIFICA2*.

La función del algoritmo *RECTIFICA* es la de mover si fuera necesario (siempre y cuando se ingrese un valor de posición requerida diferente a la actual) la posición actual del motor a un valor deseado.

El criterio que se utilizó para la fórmula de *REFERENCIA* es que éste valor debe ser múltiplo de 11.25 (el computador realiza los cálculos con números enteros). Se toma este valor en grados porque el codificador de posición de eje del motor tiene 5 líneas de salidas de código GRAY con lo cual podría tener $2^5 = 32$ combinaciones de valores de posición dentro de un disco de 360 grados. Por lo tanto la resolución del codificador es de $360/32 = 11.25$ grados. Es decir el motor reconocerá solamente valores de posición múltiplos de 11.25 grados.

En el procedimiento *RECTIFICA*, cualquier valor ingresado en el rango 0-359 (en grados) será *rectificado*. Dentro del algoritmo se compara el valor de *REFERENCIA* con la posición actual del motor (*POSACTUAL*), si son

iguales hay un retorno de subrutina, si no lo son se compara la diferencia entre los valores de las dos variables con 180, de tal forma que el desplazamiento del ángulo sea siempre por el camino más corto (camino más óptimo).

Cuando la diferencia entre REFERENCIA y POSACTUAL es **menor** a 180, el programa salta a través de la directiva RECT3 a un lazo en el cual el desplazamiento del ángulo es positivo.

Caso contrario ocurre si la diferencia entre REFERENCIA y POSACTUAL es **mayor o igual** a 180, entonces el programa salta a través de la directiva RECT6 a un lazo en el cual el desplazamiento del ángulo es negativo.

Como se explicó en el capítulo 9, para valores digitales mayores a 128 el sentido de giro del motor será comandado positivamente, mientras que para valores digitales menores a 128 el sentido de giro del motor será comandado negativamente.

Se puede entonces deducir para producir un desplazamiento negativo del ángulo es necesario restar de 128 (velocidad igual a cero) el valor de la variable DAC que es igual a $ERROR \cdot KP$. También se ha dispuesto para

vencer rápidamente la inercia del motor restar un valor adecuado de 80 a la variable DAC. Hay que tomar en cuenta los valores de KP que el usuario coloca al sistema porque puede producir un sistema relativamente oscilante a medida que aumenta el valor de KP. Valores altos de KP pueden causar grandes oscilaciones que producirían que el sistema de posición demore en estabilizarse. La variable DAC tiene al comienzo un valor grande y a medida que avanza el motor éste va disminuyendo (debido a que el error va disminuyendo) hasta el punto que llega a valer cero cuando el motor se ha posicionado en el ángulo requerido. De la misma manera para obtener un desplazamiento positivo del ángulo es necesario sumar a 128 el valor de la variable DAC. Igualmente en este caso hay que considerar en gran medida el valor de ganancia proporcional KP. A medida que avanza el motor hacia su destino final, el programa va leyendo la posición actual del motor a través de la macro LEEPOSICION e imprimiéndolo en pantalla.

El algoritmo RECTIFICA2 cumple exactamente la misma función que el procedimiento RECTIFICA, con la salvedad de que este procedimiento sólo actúa cuando los ángulos referidos son positivos y además el valor de REFERENCIA es mayor al valor de POSACTUAL. Por lo que el sentido de giro siempre será positivo. Aquí no hay comparación con 180 para obtener el

desplazamiento mas corto ya que el giro es en un solo sentido.

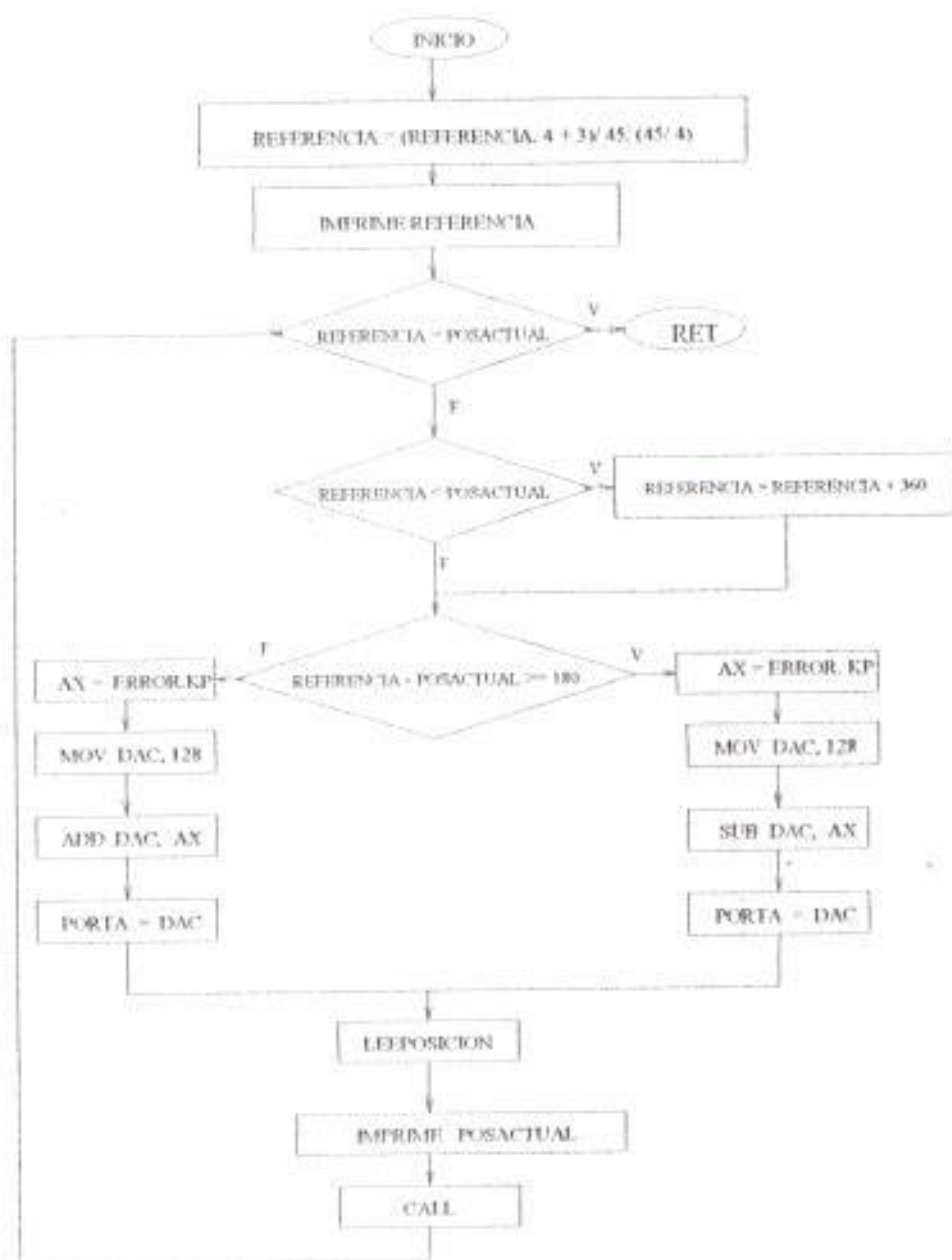


Figura 11. 17 Diagrama de flujo del algoritmo "RECTIFICA"

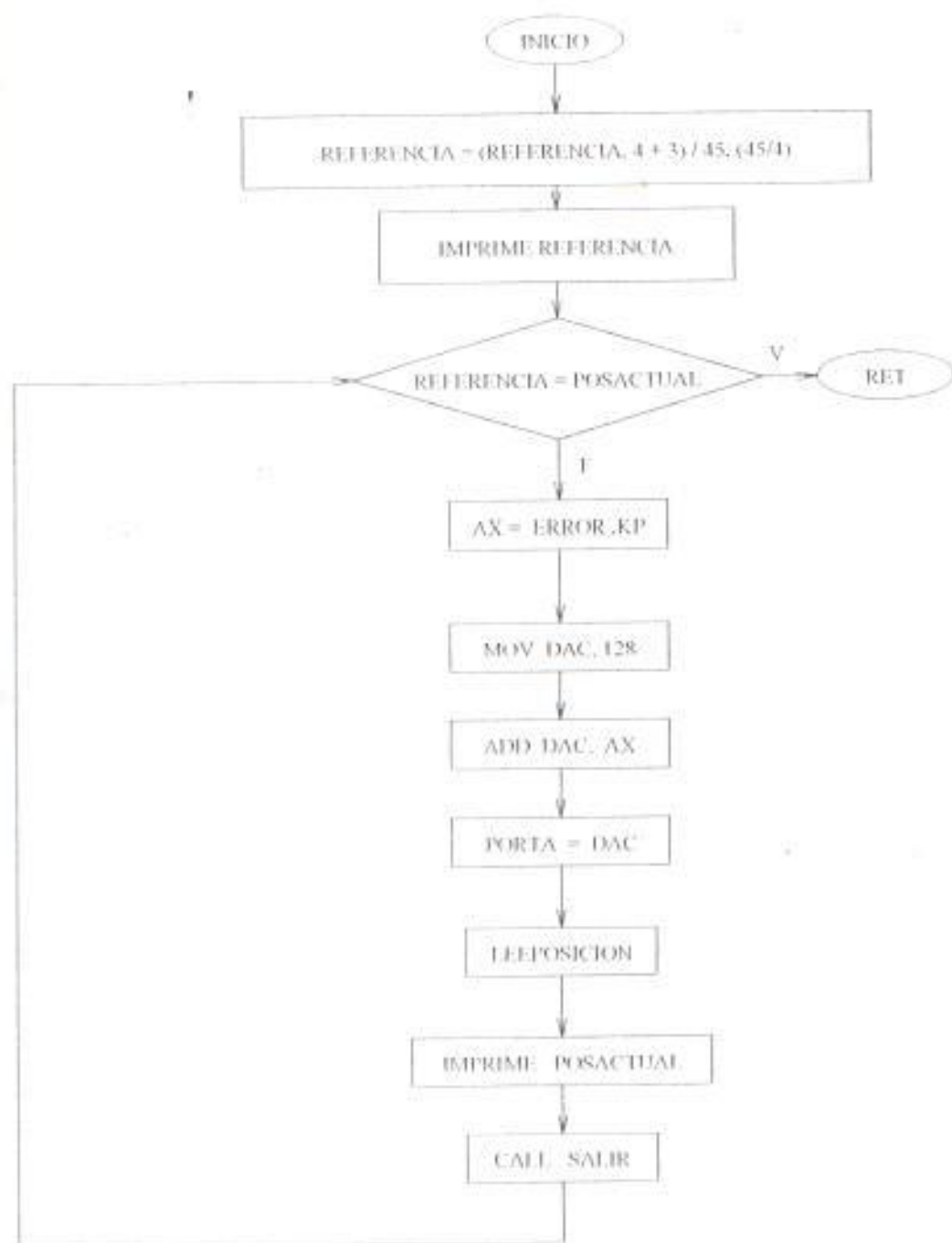


Figura 11. 18 Diagrama de flujo del algoritmo "RECTIFICA2"

CAPITULO XII

MEDIDORES DIGITALES

12.1. TACOMETRO DIGITAL

La construcción de un tacómetro digital para este proyecto nos servirá para confirmar la velocidad del motor en las pruebas ha realizarse tanto en el control digital como analógico. Este circuito consiste en un contador de revoluciones por detección óptica. Este tacómetro permite visualizar con buena precisión las revoluciones del eje de alta del motor de pruebas, ofreciendo directamente la medida de velocidad en r.p.m. La visualización se realiza mediante cuatro dígitos, lo que nos ofrece un margen de medida de hasta 9990 r.p.m. Entre las características principales de este circuito, tenemos una apreciación de 10 r.p.m., para un rango de medida calibrado entre 10-6000 r.p.m. para el presente caso, en la

figura 12.1 se muestra una representación de este tacómetro.

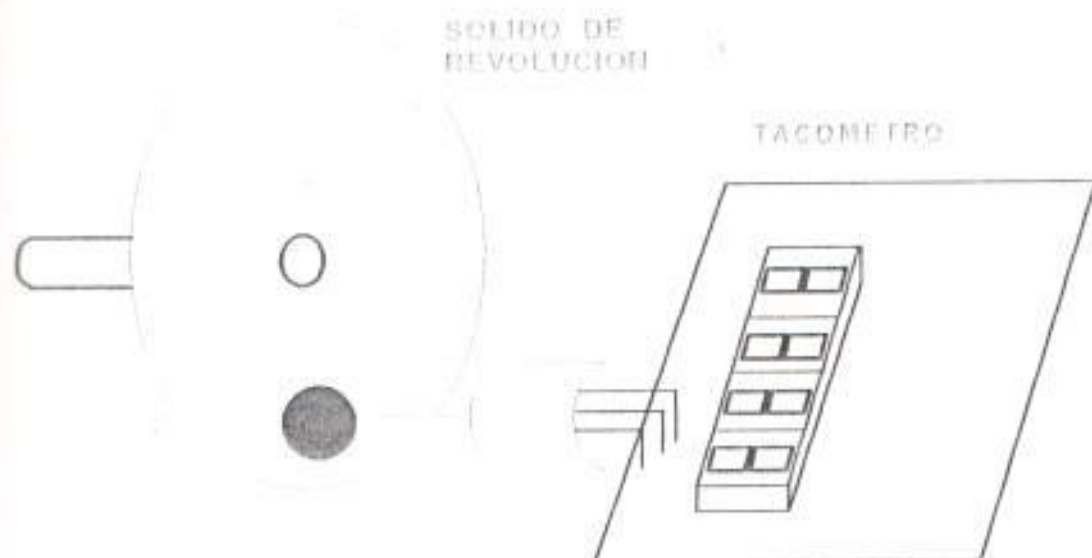


Figura 12.1 Representación esquemática del tacómetro digital

12.1.1. DISEÑO Y CONSTRUCCION

Para el diseño y construcción de este tacómetro hemos utilizado, el circuito integrado 4553 que es la base de este diseño ya que contiene tres contadores en código BCD de salidas multiplexores. Sus salidas codificadas se encuentran disponibles en los pines 5, 6, 7, y 9, los pines correspondientes a los encendidos de los dígitos son los pines 1, 2, y 15. Además este circuito integrado contiene memorias o latch de salida que se controlan a través del pin 10 y el pin 13 es el

reset o puesta a cero del contador. Otro componente utilizado es el chip 4543, el cual es un decodificador de siete segmentos con salida directa para conexión a displays; sus salidas son los pines del 9 al 15 siendo sus entradas en BCD los pines 2, 3, 4, y 5 mientras que el pin número 6 nos permite elegir la lógica de salida, por lo se utilizó displays de ánodo común, y el pin número 1 es el de reset. También tenemos un contador con registro de desplazamiento del tipo 4017, el cual es manejado por un oscilador basado en un 555. Su funcionamiento general es como sigue: el circuito está compuesto por un contador y una base de tiempos, el contador periódicamente es reajustado, obteniendo de esta forma el conteo en un tiempo determinado. Primeramente disponemos de un acoplador óptico por deflexión que forma una barrera óptica, ésta al ser interrumpida, proporciona impulsos los cuales son configurados (onda cuadrada) mediante Q1 que luego es enviado a IC2 (CI 4543), el cual multiplexa sus salidas (estando esta frecuencia fijada por C3, hacia Q2, Q3, Y Q4), las cuales se encargan del encendido sucesivo de los displays. Por otra parte, la base del tacómetro es el circuito que comanda el contador, proporcionándole las órdenes del reset, puesta a cero y memorización del número contado al final de cada periodo. Esto se lo hace mediante el contador con registro de desplazamiento (CI 4017), manejado por un oscilador basado en un circuito integrado 555. Para un correcto funcionamiento disponemos de un potenciómetro variable (R13), el cual nos

permite calibrar de manera que nos de una lectura correcta, debemos conocer alguna fuente de velocidad conocida, y calibrando con R13, el cual determina la frecuencia del reloj, que es el que maneja al contador (CI 4017), hasta obtener la velocidad de dicha fuente. En el capítulo 10 se encuentra el diagrama esquemático de este tacómetro.

12.2. AMPERIMETRO DIGITAL

Este instrumento de medición en nuestro proyecto, es el encargado de verificar el consumo de corriente del motor DC. La corriente nominal del motor es 1.5 amperios, por lo que en la mayoría de los casos esta será la máxima corriente de trabajo, una corriente mayor a ésta (1.9 amperios, corriente máxima que muestra el amperímetro), mantenida por un tiempo considerable quemaría el motor. En la figura 12.2 se muestra una representación de este amperímetro conectado al motor.

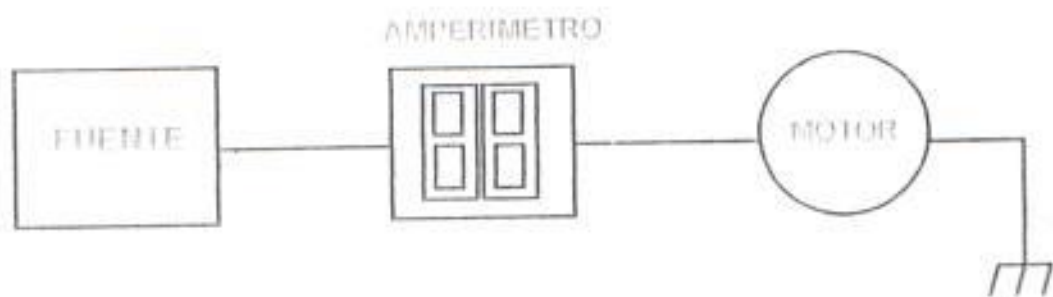


Figura 12.2 Representación del amperímetro digital

12.2.1. DISEÑO Y CONSTRUCCION

La base del diseño de este amperímetro, es un convertidor analógico-digital (ADC 0808), el cual se lo analiza en detalle en el capítulo siete. Además se utilizan circuitos integrados como un comparador (74LS85), un sumador (74LS83), un decodificador de siete segmentos (CD4543), con salida directa para displays, el chip (74LS32), que contiene cuatro puertas OR y dos displays del tipo ánodo común.

El elemento principal de este amperímetro es el convertidor analógico digital 0808, para la presente aplicación este convertidor digitalizará voltajes analógicos entre 0.4 y 1.5 voltios que equivalen a 0.4 y 1.5 amperios respectivamente.

Hemos utilizado los cuatro pines menos significativos de este convertidor (18, 8, 15, 14) como salidas digitales, además utilizamos el pin 26 (IN0), como entrada analógica, para habilitar esta entrada, los pines 23, 24, y 25 van conectadas a tierra ya que estos seleccionan este pin.

Los pines 6 (START), 22 (ALE) y 7 (EOC) van unidos entre sí ya que de esta manera el ADC siempre va a estar convirtiendo.

El reloj para este convertidor está conformado por el chip 555, el cual tiene una frecuencia de 4KHz, de acuerdo a los elementos que lo configuran. La señal de salida de este reloj es aplicada al pin 10 del convertidor (entrada del reloj).

Las salidas de este convertidor se las aplica a un sumador (74LS83) y a un comparador (74LS85) el cual compara el valor digital que le entrega el convertidor con nueve digital (1001). Si las salidas del convertidor es menor a este valor entonces la salida en el sumador (pines 9,6,2,15) es igual al valor del convertidor, este valor se lo aplica a un decodificador de siete segmentos (CD4543) el cual maneja displays de ánodo común que mostrará valores entre 0 y 0.9 amperios.

Si la salida del convertidor ADC es mayor a 1001 entonces la salida del comparador ($A > B$) habilita la otra entrada del sumador a través de dos puertas OR (74LS32) con un valor de seis digital (0110) luego del cual tendremos valores de 1.0 a 1.9 amperios (máxima corriente que muestra este amperímetro).

También hemos visto necesario diseñar un tipo de alarma audible, cuando la corriente en el motor sea mayor o igual a 1.5 amperios (corriente nominal del motor). De esta manera se alertará a las personas que están trabajando con este

motor para así evitar algún daño por sobrecalentamiento. Esta alarma tipo audible, se lo ha diseñado utilizando dos chips 555. Primeramente, uno de estos chips (numero uno), configurado como reloj manejará el pin 4 (reset) del otro chip (numero dos), de esta manera se tendrá un sonido no continuo determinado por la frecuencia de oscilación del primer chip. La alarma se activará a partir de un comparador diseñado con un circuito integrado LM741, la señal proveniente del motor (1.5 V) se lo compara con 1.49 V, de dicha comparación la salida del comparador (pin 6) se activará con un valor alto el cual habilita al chip 555 (numero 1) luego de lo cual la salida del chip 555 (numero 2) activa un amplificador de audio, el cual excita un parlante de 8 ohmios de medio vatio. El amplificador de audio consta de un transistor tipo NPN (D1313) como se muestra en el diagrama esquemático del capítulo 10 de este amperímetro.

CONCLUSIONES

En términos generales la presente tesis constituye una base muy importante para demostrar la teoría de control automático, teniendo presente que esta puede ser aplicada en otras ciencias para demostrar algunos de los principios que estas involucran. Se ha logrado demostrar que las respuestas obtenidas experimentalmente justifican el análisis teórico, es por este motivo que el *banco de pruebas* se ha convertido en una herramienta muy importante para cada uno de los estudiantes que desean comprobar lo que en la teoría se les ha instruido.

La construcción del banco de pruebas se lo ha realizado de una forma muy didáctica y ordenada, el cual está respaldado por varios experimentos que se ponen a consideración de todos los estudiantes de la ESPOL. El grupo humano que hemos trabajado estamos seguros que entregamos a nuestra Universidad un material de alta calidad de enseñanza para que las futuras generaciones se presenten a la sociedad con soluciones que sean respaldadas de una forma analítica y experimental.

En el aspecto económico el proyecto resultaría muy conveniente ya que el costo total es muy inferior con respecto a los valores que exponen compañías extranjeras dedicadas a la

venta de proyectos didácticos para estudiantes. Los integrantes de la presente tesis nos ponemos a disposición de toda Institución u Organismo que solicite la construcción individual o serie de cualquier equipo electrónico didáctico o de trabajo, garantizando de antemano un correcto funcionamiento y un respaldo científico del proyecto en cuestión.

RECOMENDACIONES

En los actuales momentos el país requiere de profesionales altamente capacitados en todos sus áreas y muy especialmente en el área técnica, los cuales deben ser capaces de resolver diferentes tipos de situaciones que a diario se presentan en la industria.

La ESPOE debe incrementar sus programas de estudios, la formación de grupos de trabajo que hagan que cada uno de sus integrantes sean participes directos en la investigación de cada uno de sus proyectos y dar todo el apoyo necesario a aquellos que realizan proyectos de tesis debido a lo difícil que resulta realizar una tesis para un estudiante debido principalmente al factor económico, ya que generalmente para realizar una tesis, se necesita mucha investigación, elementos que no existen en el mercado nacional, además de una falta de manuales, libros y equipos de computación.

Por lo tanto, la ESPOE debe relacionarse más con la industria, para que de esta manera los estudiantes puedan realizar proyectos de tesis de acuerdo a los problemas que existen en las empresas; con esto se cumpliría con un doble objetivo: conocer a fondo los problemas que existen en las industrias, con lo que los estudiantes ganarían experiencia con la práctica, condición fundamental para conseguir trabajo y el factor económico correría por cuenta de las empresas.

Con lo que respecta a la presente tesis se debería construir tres bancos adicionales, para que funcione el laboratorio de control automático, ya que es una materia de experimentación y práctica, y además el control automático se lo utiliza en todos los procesos sean eléctricos electrónicos, mecánicos, económicos, etc., por lo que es necesario conocer la teoría y práctica de esta materia.

BIBLIOGRAFIA

1. JACOB M., Industrial Control Electronics (applications and design), New York, Prentice Hall, 1989.
2. NATIONAL SEMICONDUCTOR CORPORATION , Linear databook 2, Santa Clara, California, National Semiconductor, 1994.
3. PUENTE J., " Diseño y construcción de un controlador digital basado en un microprocesador 8085A de Intel, para uso en el Laboratorio de Control Automático de la ESPOL", Escuela Superior Politécnica del Litoral, Guayaquil, 1983.
4. OGATA K. , Ingeniería de Control Moderna, México, Prentice Hall, segunda edición 1993.
5. CHESMOND C., Control System Technology , Australia , Edward Arnold, first edition 1982.

6. PERTENCE Jr., Amplificadores Operacionales , España, McGraw - Hill, 1991.
7. DRISCOLL, F., Amplificadores Operacionales y Circuitos Integrados Lineales, México, Prentice Hall, primera edición 1993.
8. SMITH C. Y CORRIPIO A., Control Automático de Procesos , México, Limusa, primera edición 1991.
9. VELEZ F., Diseño y construcción de un controlador digital basado en un microcomputador ", Escuela Superior Politécnica del Litoral , Guayaquil, 1995.
10. FLETCHER W., An engineering approach to digital design, New Jersey, Prentice Hall, 1980.