

19
24

Algoritmos de
CONTROL DIGITAL DIRECTO
aplicados al control de velocidad de un Motor de CD



TESIS

que presentan:

Francisco Emigdio Torres García
Carlos Cortés Martínez

para obtener el título de:

Ingeniero Mecánico Electricista.

Escuela Nacional de Estudios Profesionales, Plantel Aragón.
Universidad Nacional Autónoma de México.

San Juan de Aragón, Estado de México, abril de 1991.

TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

CONTENIDO

GENERALIDADES

PREFACIO	1
DOCUMENTACION DEL PROYECTO	1
PREPARACION DE LA TESIS	1
OBJETIVOS GENERALES DE LA TESIS	2
CONTENIDO DE LA TESIS	3
INTRODUCCION	5
MECANIZACION Y AUTOMATIZACION	5
HISTORIA DEL CONTROL AUTOMATICO	6
EL SIGLO XVII	6
EL SIGLO XVIII	6
EL SIGLO XIX	7
EL SIGLO XX	7
CONTROL CLASICO Y CONTROL MODERNO	9
SISTEMAS DE CONTROL COMPUTARIZADOS	10
TIPOS DE CONTROL DIGITAL	10
CONTROL SUPERVISOR	10
CONTROL DISTRIBUIDO	10
CONTROL DIGITAL DIRECTO	12
RESUMEN	13

DESARROLLO DE LOS PROGRAMAS

HARDWARE	15
SOFTWARE	16
OPERACION	16
ENFOQUE ESTRUCTURADO	16
ESPECIFICACIONES DEL SISTEMA	18
ESQUEMA GENERAL	19
ARCHIVOS DE DATOS	23
SEGURIDAD DEL SISTEMA	24
FORMATOS DE ENTRADA/SALIDA	24

DESCRIPCION DE PROGRAMAS Y PROCEDIMIENTOS	25
PROGRAMA Menú	26
PROGRAMA Captura	30
PROGRAMA Traza	35
PROGRAMA Transfo	39
PROGRAMA Control	42
PROGRAMA Calcula	46
RESUMEN	50
LISTADO DEL PROGRAMA MENU	51
LISTADO DEL PROGRAMA CAPTURA	55
LISTADO DEL PROGRAMA TRAZA	60
LISTADO DEL PROGRAMA TRANSFO	66
LISTADO DEL PROGRAMA CONTROL	69
LISTADO DEL PROGRAMA CALCULA	73

DESCRIPCION DEL EQUIPO Y ACCESORIOS

DESARROLLO DE LA MICROBBC	79
DESCRIPCION DE LA MICROBBC	80
CONVERTIDOR ANALOGICO DIGITAL	82
FUERTO PARALELO	83
UNIDAD DE DISCO	83
MONITOR DE COLOR RGB	83
OTROS CIRCUITOS	84
DESCRIPCION DE LOS ACCESORIOS	85
FUENTE DE PODER	85
TACOMETRO DIGITAL	85
AMPLIFICADOR DE POTENCIA	90
CONVERTIDOR DIGITAL — ANALOGICO	91
MOTOR DE CORRIENTE DIRECTA	92
TACOGENERADOR DE CD	93
AMPLIFICADORES DE INSTRUMENTACION	94
RESUMEN	102

OBTENCION EXPERIMENTAL Y ANALISIS DEL MODELO

FUNCION DE TRANSFERENCIA	103
FUNCION TEORICA	104
FUNCION PRACTICA	104
METODO LOGARITMICO	105
METODO DE LIPKA	107
METODO DE ZIEGLER Y NICHOLS	111
OBTENCION DEL MODELO DE LA PLANTA	113
CONEXION DEL EQUIPO	113
CALENTAMIENTO DEL MOTOR	116
VELOCIDAD MAXIMA	116
CALIBRACION DE SENSORES	118
SIMULACION DE ENTRADA ESCALON	117
GRAFICACION DE LA RESPUESTA TRANSITORIA	118
NORMALIZACION DE ABCISAS	119
PROMEDIO DE ARCHIVOS	121
ANALISIS DEL MODELO DE LA PLANTA	122
RESPUESTA TRANSITORIA	123
METODO LOGARITMICO	124
APROXIMACION DE UN POLO	124
APROXIMACION CON DOS POLOS	127
APROXIMACION CON TRES POLOS	132
ANALISIS EN LAZO ABIERTO (3 POLOS)	136
METODO DE LIPKA	138
METODO DE ZIEGLER Y NICHOLS	141
RESUMEN DE FUNCIONES	143

RESUMEN	146
---------------	-----

CONTROLADORES DIGITALES P, PI Y PID

TIEMPO DE MUESTREO	147
TRANSFORMADA Z	148
DESCRIPCION DE LOS ALGORITMOS	152
Acción Proporcional	153
Acción Integral	154
Acción Derivativa	154
Acción Proporcional, Integral y Derivativa (PID)	155
DISEÑO DEL CONTROLADOR CONTINUO	156
DISEÑO DEL CONTROLADOR DISCRETO	157
DISCRETIZACION	159
DISEÑO EN EL PLANO Z	161
CALCULO DE LAS CONSTANTES PID	161
ACCION PROPORCIONAL	163
ACCION INTEGRAL	165
ACCION PROPORCIONAL INTEGRAL	167
ACCION PROPORCIONAL INTEGRAL DERIVATIVA	170
APLICACION DEL CONTROLADOR	173
ACCION INTEGRAL (I)	174
ACCION PROPORCIONAL INTEGRAL (PI)	175
ACCION INTEGRAL DERIVATIVA (ID)	176
ACCION PROPORCIONAL, INTEGRAL Y DERIVATIVA (PID)	177
ANALISIS EN LAZO CERRADO	178
RESUMEN	178

CONCLUSIONES

EVALUACION DEL SISTEMA	179
ERRORES Y LIMITACIONES	180
ERRORES DE ALGORITMO	180
DESARROLLOS POTENCIALES	181

APENDICES

SISTEMA OPERATIVO Y HARDWARE	183
FUNCIONES ESPECIALES	185
PARTICULARIDADES DEL LENGUAJE BASIC-BBC	185
METODO DE MINIMOS CUADRADOS	192
METODO DE INTERPOLACION DE LAGRANGE	193
TRANSFORMADA DE LAPLACE	194
FUNCION DE TRANSFERENCIA	196
METODO DEL LUGAR DE LAS RAICES	196
GLOSARIO	199

BIBLIOGRAFÍA

FIGURAS

Fig. 1 Sistema de Control con Computadora Digital.	8
Fig. 2 Hardware del sistema.	16
Fig. 3 Sistema de Control Digital Directo	20
Fig. 4 Subsistema de calibración	20
Fig. 5 Subsistema de Simulación.	21
Fig. 6 Subsistema de Análisis y Transformación.	21
Fig. 7 Subsistema de Control.	22
Fig. 8 Pseudocódigo del Programa Menú.	27
Fig. 9 Pseudocódigo del Programa Captura.	31
Fig. 10 Pseudocódigo del Programa Traza.	36
Fig. 11 Pseudocódigo del Programa Tranfo.	40
Fig. 12 Pseudocódigo del Programa Control.	43
Fig. 13 Pseudocódigo del Programa Calcula.	47
Fig. 14 Computadora MicroBBC	80
Fig. 15 Conector y cable del puerto analógico digital.	82
Fig. 16 Conector y cable paralelo (salida digital).	83
Fig. 17 Asignación de pines del cable paralelo.	84
Fig. 18 Fuente de poder para planta y circuitos.	86
Fig. 19 Motor y optosensor.	87
Fig. 20 Tacómetro Digital conectado a la planta.	88
Fig. 21 Diagrama de tiempos del tacómetro.	90
Fig. 22 Amplificador de potencia.	91
Fig. 23 Convertidor Digital Analógico.	91
Fig. 24 Motor de corriente directa.	92
Fig. 25 Circuito LM324	93
Fig. 26 Amplificador de instrumentación.	94
Fig. 27 Cable de los amplificadores a la computadora.	95
Fig. 28 Variación de ganancia A/Ao en el amplificador.	97
Fig. 29. Conexión de los amplificadores a la planta.	100
Fig. 30 Polos y ceros de la planta real.	104
Fig. 31 Simplificación de polos y ceros de la planta.	105
Fig. 32 Respuesta experimental de un sistema.	106
Fig. 33 Ajuste de recta sobre LN(w).	106
Fig. 34 Curva de reacción de Ziegler y Nichols.	112
Fig. 35 Respuesta ideal de un sistema usando control PID.	113
Fig. 36 Diagrama a bloques. Obtención de modelo de la planta	114
Fig. 37 Conexión de la planta.	115
Fig. 38 Lugar de raíces. un solo polo.	126
Fig. 39 Lugar de raíces. Dos polos.	131
Fig. 40 Lugar de raíces. Tres polos.	138
Fig. 41 Lugar de raíces. Dos polos por el método de Lipka.	141
Fig. 42 Lugar de raíces. Un polo y retraso.	143
Fig. 43 Lugar de raíces discreto. Un solo polo.	151
Fig. 44 Lugar de raíces discreto. Dos polos.	151
Fig. 45 Lugar de raíces discreto. Tres polos.	152
Fig. 46 Planta continua con controlador D(s).	153
Fig. 47 Respuesta óptima de un sistema con controlador PID.	155
Fig. 48 Curva de reacción de Ziegler y Nichols.	156
Fig. 49 Lugar de raíces. Acción proporcional. Un solo polo.	164
Fig. 50 Acción Integral. Dos raíces reales iguales.	165
Fig. 51 Acción Integral. Dos raíces complejas conjugadas.	166
Fig. 52 Acción Proporcional Integral. Dos polos reales.	168
Fig. 53 Acción Proporcional Integral. Dos polos complejos.	170
Fig. 54 Posibles polos de lazo cerrado.	170
Fig. 55 Tres polos, acción PID.	172
Fig. 58 Acción proporcional integral.	175
Fig. 59 Acción Integral y derivativa.	178
Fig. 60 Acción proporcional, integral y derivativa.	177
Fig. 61 Dispersión de datos y ajuste de recta.	182
Fig. 62 Sistema de control.	196

TABLAS

Tabla 1. Variación de la ganancia A/A_0.	98
Tabla 2. Tres polos. Resumen de valores	135
Tabla 3. Método de Ziegler - Nichols (continuo)	157
Tabla 4. Método de Ziegler - Nichols, controlador discreto.	159
Tabla 5. Discretización de potencias.	160
Tabla 6. Transformadas de Laplace	195

PREFACIO

DOCUMENTACION DEL PROYECTO

La documentación que se prepara para cualquier proyecto de Ingeniería, cualquiera que sea la naturaleza de éste, es más importante que la programación, el equipo, o cualquier otra parte individual del mismo. De hecho, ningún proyecto puede considerarse enteramente válido si no está debidamente apoyado en una documentación profusa y adecuada.

De esta forma, la documentación del proyecto, es decir esta tesis, es la integración de los objetivos buscados, los métodos empleados, los experimentos llevados a cabo, así como las conclusiones a las que se llegó en tal empresa, además de ser la guía y la llave maestra de la reproductibilidad de los experimentos que en ella se describen.

En esta reproductibilidad se basa fundamentalmente la veracidad científica del quehacer del ingeniero.

Asimismo, en estos documentos deben encontrarse los puntos de referencia, las premisas de que se ha partido y los alcances y limitaciones esperados del proyecto en su conjunto.

Nota: Este trabajo está debidamente registrado ante la Dirección General de Derechos de Autor, de la Secretaría de Educación Pública.

PREPARACION DE LA TESIS

La generación de los textos que conforman esta tesis se hizo directamente en el procesador de palabra de la MicroBBC, llamado Vista, donde se hicieron la mayoría de las correcciones iniciales, así como las primeras copias de prueba, en una impresora de matriz Delta 10/130 conectada al puerto paralelo (donde también se conecta el convertidor D/A); luego los textos fueron transferidos a una computadora personal marca Sprint 386SX de 16 MHz, con sistema operativo MS-DOS versión 3.3, a través de un cable que conectó el puerto serial RS232 de la PC (COM1:) al puerto serial RS423 de la MicroBBC.

La transferencia de archivos se hizo con 8 bits de datos, un bit de parada, y sin comparación de paridad, utilizando el modo full duplex a 9600 baudios, considerando los listados como archivos ASCII sin caracteres especiales de formato. La corrección de estilo y de formato inicial se hizo con el ambiente de operación Microsoft Windows 3.0, así como con el procesador de palabra Microsoft Word 5.0.

De allí se exportaron los textos al programa de formación editorial Xerox Ventura Publisher, Extensión Profesional, donde fueron formados en el tamaño adecuado a la tesis (medio tamaño oficio) y con la tipografía correcta. Las ecuaciones y fórmulas también se editaron directamente en Xerox Ventura Publisher 2.0 con la extensión profesional.

Los diagramas provienen de varias fuentes: los de tipo eléctrico y electrónico se hicieron con el programa OrCAD, los diagramas de bloques y la representación del motor y los de lugar de raíces se hicieron con el programa Corel DRAW! 2.0 y se exportaron a Ventura en el formato TIFF (Tag Image File Format).

Algunas tablas se hicieron en Aldus PageMaker 3.0 y fueron impresas por separado. La fotografía de la MicroBBC y algunas otras figuras fueron transformadas a imágenes digitales con un digitalizador Hewlett Packard ScanJet plus, con el programa Scanning Gallery 5.0 de HP. Algunas de estas imágenes fueron luego trazadas con Corel Trace.

Los programas fueron desarrollados en el lenguaje (intérprete) BASIC incluido en la Micro BBC, y depurados directamente en ella, junto con sus descripciones y listados de pseudocódigo, para luego ser exportados también a Ventura Publisher.

La impresión final del original mecánico para la reproducción de la tesis se hizo en una impresora Hewlett Packard LaserJet III.

Nota: Las marcas mencionadas están registradas por sus respectivos dueños.

OBJETIVOS GENERALES DE LA TESIS

1. Mostrar un método de obtención experimental de una función de transferencia. Hacer hincapié en que la generalidad de este método lo hace transportable a otras plantas y computadoras.
2. Elaborar un conjunto de programas para una microcomputadora digital, orientados a la obtención del modelo experimental de la planta, así como al cálculo y aplicación de los controladores digitales, haciendo uso del método estructurado para desarrollo de sistemas de software.
3. Mostrar la aplicación en una microcomputadora digital de los principales algoritmos de control: Proporcional (P), Proporcional + Integral (PI), y Proporcional + Integral + Derivativo (PID), y mencionar sus ventajas y desventajas comparados con controladores analógicos.

Para alcanzar estos objetivos se utilizará como planta un motor de corriente directa, alimentado por un amplificador de potencia excitado por la salida de un convertidor D/A, bajo control de la computadora. Por otro lado, la velocidad del motor es sensada por un tacogenerador de CD conectado a un amplificador operacional de instrumentación y usando un convertidor A/D como interfase de regreso a la computadora, cerrando así el lazo de realimentación.

Del sistema en su totalidad se obtendrá y analizará la función de transferencia en forma experimental utilizando la computadora, y con ello su transformada Z, y otras características importantes.

Con esta información se calcularán los elementos del algoritmo de control (PID) para su aplicación en la microcomputadora.

En cada caso se hará la experimentación correspondiente para mostrar los efectos y las características de cada tipo de control, es decir se aumentarán a la planta los controladores calculados y se aplicará un escalón de entrada, para observar su respuesta.

Con base en los resultados, se expondrán conclusiones.

CONTENIDO DE LA TESIS

INTRODUCCION

Aquí se hace un breve relato histórico del Control Automático, desde la antigüedad hasta nuestros días. Se hace referencia a la Teoría de Control Clásica, versus la Teoría Moderna, sus diferencias y puntos de contacto. Se hace referencia al control digital directo.

DESARROLLO DEL SISTEMA DE SOFTWARE

Se describe el método estructurado de desarrollo de sistemas, la elaboración de la documentación y de los programas de computadora.

Se hace notar el uso de un lenguaje de alto nivel y de pseudocódigo, para asegurar la portabilidad de los conceptos y los algoritmos a otras computadoras.

Se presentan las descripciones y los listados de los programas desarrollados.

DESCRIPCION DEL EQUIPO Y ACCESORIOS

Aquí se describe la historia y la arquitectura de la microcomputadora MicroBBC, sus características y su lenguaje de programación estructurado BASIC.

Se describe la planta en mayor detalle: El motor de CD, tacogenerador y los otros implementos usados para el desarrollo de la tesis: amplificador de potencia, amplificadores de instrumentación, fuente de poder, convertidores A/D y D/A, multímetro, etc.

Se hacen los cálculos pertinentes para el diseño de las interfaces entre la computadora y la planta (amplificadores de instrumentación y tacómetro digital).

MODELADO Y ANALISIS DE LA PLANTA

Se describen las premisas, las bases teóricas, así como los métodos, algoritmos y programas usados para la obtención experimental de la función de transferencia de la planta en su conjunto.

Se hace notar que los métodos son transportables a otras plantas y computadoras, dada su generalidad.

Se hace una comparación de resultados analíticos con los datos originales (experimentales) para dar validez a los modelos obtenidos.

A partir del modelo hallado, se obtienen las características y ecuaciones de la planta: la transformada en Z, la gráfica del lugar de las raíces, etc.

Se discuten los programas y algoritmos usados para llevar a cabo el análisis.

Se hace un resumen de las funciones de transferencia halladas, usando tanto la transformada de Laplace (variable compleja s) como en su correspondiente forma usando la transformada Z.

CONTROLADORES P, PI Y PID

Se discuten los algoritmos de control P, PI y PID, y se diseñan los controladores digitales como funciones dentro de un programa de cómputo, que será usado sobre la planta.

En cada caso, se obtienen los datos, gráficas, etcétera, para evaluar los resultados, comparándolos con la planta sin controlador.

Se aplica el controlador digital a la planta, a través del algoritmo PID programado en la computadora. Se analiza cada acción de control en particular, observando sus características.

RESUMEN Y CONCLUSIONES

Se resumen los procedimientos y los datos obtenidos a lo largo de este trabajo, utilizándose los resultados para determinar las ventajas y desventajas de cada método, y su aplicación en una microcomputadora. Finalmente se exponen las conclusiones.

APENDICES

Aquí se reúnen las herramientas matemáticas o de la Teoría de Control que se hayan utilizado, y que no hayan sido descritas en detalle en otro lugar.

GLOSARIO.

Se anotan algunos de los términos más importantes usados a lo largo de la tesis, junto con su significado.

BIBLIOGRAFIA

Se listan las obras consultadas, por orden alfabético de apellido de autor principal, agrupadas por área de conocimiento. En el texto se hace referencia a ellas por autor principal y página consultada.

INTRODUCCION

MECANIZACION Y AUTOMATIZACION

Desde la antigüedad, el hombre debió confiar en su fuerza o la de los animales con el fin de obtener la energía necesaria para trabajar. Después aprendió a utilizar dispositivos mecánicos, tales como palancas y ruedas, o a aprovechar las fuentes naturales como el agua y el viento, para amplificar sus recursos energéticos, aunque no podía obtener su energía útil a voluntad. Este proceso de reemplazo de fuentes de energía es lo que se conoce como mecanización.

Para lograr el avance a la edad de la automatización, el hombre buscó la manera de inventar máquinas que pudieran independizarse de la intervención humana, incluso cuando realizaran tareas de considerable complejidad, que requerían habilidades que físicamente no podía igualar el hombre.

En consecuencia, la diferencia principal entre la mecanización y la automatización es el proceso de control: en la primera, se trata con máquinas, sencillas o complejas, controladas por operadores humanos, de manera que la mecanización, por compleja que sea, resulta un proceso de reemplazar los músculos por trabajo mecánico.

En la automatización, se trata de una máquina que se controla a sí misma mediante sucesiones de tareas sin intervención humana; y todo el proceso está ajustado a requerimientos predeterminados, que pueden haber estado o no ajustados externamente por el operador humano; es decir, la máquina automatizada puede comportarse como un cuerpo vivo capaz de controlar sus acciones y reaccionar a cualquier cambio de circunstancias, corrigiendo desviaciones.

Esta característica, conocida como control de lazo cerrado o de realimentación, es la base de cualquier automatización y es lo que permite a la máquina controlar su funcionamiento en cualquier momento por medio de las instrucciones dadas por una unidad de control.

La unidad de control actúa sobre los datos que le entregan los órganos sensoriales, y la máquina ejecuta las instrucciones por medio de sistemas actuadores.

HISTORIA DEL CONTROL AUTOMATICO

Las primeras aplicaciones al control automático por realimentación se hallan en Grecia, antes de nuestra era, en los mecanismos reguladores de flotación. El reloj de agua de Ktesibios usó un regulador de flujo; una lámpara de aceite diseñada por Phylon aproximadamente en 250 a.n.e. también usó un regulador de flotación para mantener un nivel constante de aceite combustible.

El griego Herón de Alejandría, quien vivió en el siglo 1 de nuestra era, escribió un tratado titulado "Pneumática", que indicaba varios mecanismos de nivel de agua usando reguladores de flotación; él también diseñó uno de los primeros sistemas de control de lazo abierto, usando mecanismos hidráulicos, en este caso para abrir las puertas de un templo de manera automática al encender una hoguera.

Aparentemente, también hubo ciertas invenciones importantes en la China antigua, descritas por el investigador Needham (Mayr, 49).

EL SIGLO XVII

El primero sistema de realimentación inventado en la Europa Moderna fué el incubador (regulador de temperatura) desarrollado por el holandés Cornelis Drebbel en 1624.

El regulador de presión para calentadores de vapor de Dennis Papin, que data de 1681, el cual era una forma de regulador de seguridad similar a una válvula de olla de presión, también se considera importante.

Sin embargo, hasta el siglo XVIII, el diseño de sistemas de control se caracterizó exclusivamente por la invención intuitiva.

EL SIGLO XVIII

La aparición del motor de vapor y la subsecuente revolución industrial dieron a los humanos cantidades de potencia controlada más grandes de las que habían tenido hasta ese momento. Es decir que se creó la necesidad de manejar grandes cantidades de energía por medio de señales de baja potencia, preparando el camino para el desarrollo del análisis de sistemas realimentados.

Generalmente se acepta que el primer controlador automático de realimentación (de lazo cerrado) usado en un proceso industrial fué el gobernador centrífugo de James Watt, desarrollado por él en 1765. Sin embargo, por razones tecnológicas y económicas, no fue implantado sino hasta 1788. Este controlador tenía la función de regular la velocidad de un motor de vapor.

Este sistema de J. Watt utilizaba realimentación para el control del sistema y no precisaba de la intervención de ningún ser humano para su operación; este dispositivo era totalmente mecánico y medía la velocidad de la flecha de salida y utilizaba el movimiento de un par de contrapesos para regular el flujo de la válvula y por lo tanto la cantidad de vapor que entraba al motor.

El primer sistema histórico de realimentación reclamado por la Unión Soviética es el regulador de flotación inventado por I. Polzunov en 1765. El flotador detecta el nivel de agua y controla la válvula que cubre la entrada a un calentador.

Los esfuerzos para aumentar la exactitud de estos sistemas llevaba a la atenuación más lenta de las oscilaciones transitorias e inclusive a sistemas inestables. Entonces se hizo imperativo el desarrollar una teoría del control automático.

EL SIGLO XIX

El conocido científico James Maxwell formuló en 1868 una teoría matemática relacionada a la teoría de control usando un modelo de ecuaciones diferenciales de un gobernador. Dicho estudio ("On Governors", Proceedings of the Royal Society, London) se ocupó del efecto que varios parámetros del sistema tenían en la ejecución del mismo.

Más adelante, en 1876, ese estudio fué seguido por una solución más detallada de la estabilidad de un gobernador por el ingeniero ruso I.A. Vyshnegradskii, quien también formuló una teoría matemática de los reguladores. {Dorf,5}.

John M. Keynes propuso un modelo sencillo de la economía de la sociedad que abarcaba la respuesta de realimentación de las fuerzas económicas.

EL SIGLO XX

Si bien Max Tolle escribió uno de los primeros libros formales sobre control automático ("Regelung der Kraftmaschinen", Berlin, 1921 {Takahashi et al., 3 a 26}), hay una fecha más significativa en la historia de los sistemas de control automáticos, en 1934, cuando Hazen publicó su "Teoría de Servomecanismos", marcando el inicio de un intenso y moderno interés en este nuevo campo.

Fué en este documento donde se originó la palabra "servomecanismo", de las palabras siervo (o esclavo) y mecanismo, lo cual significa que la palabra servomecanismo implica mecanismo esclavo o controlado.

Sin embargo, la teoría y práctica del control se desarrollaron en los Estados Unidos y Europa Occidental de una manera distinta que en la Unión Soviética y Europa Oriental.

El principal ímpetu del uso de realimentación en los E.U.A. fué el desarrollo del sistema telefónico y los amplificadores de realimentación, en la década de los 30's por Bode, Nyquist y Black en los laboratorios telefónicos Bell. Durante este lapso, casi todos los controladores eran de tipo proporcional, los de tipo PID aparecen hasta la siguiente década.

El estudio de los sistemas de control por medio de métodos del dominio de la frecuencia se usó preeminentemente para describir la operación de los amplificadores de realimentación en términos del ancho de banda y otras variables de frecuencia, como el margen de ganancia y el margen de fase.

En contraste, los eminentes matemáticos y mecánicos aplicados de Rusia inspiraban y dominaban el campo de la Teoría del Control, por lo tanto, la teoría rusa tendió a utilizar la formulación en el dominio del tiempo usando ecuaciones diferenciales.

Por otro lado, el desarrollo moderno de la automatización se inició al principio de la década de 1940. Entonces, un gran impulso de la teoría y la práctica del control automático de sistemas realimentados se dió en la Segunda Guerra Mundial, cuando fué necesario diseñar y construir

sistemas de posicionamiento de armas, pilotos automáticos, sistemas de control de radares y otros ingenios militares usando el control por realimentación.

La exactitud y el desempeño esperados de estos sistemas forzó una extensión a las técnicas de control disponibles y despertó el interés en sistemas de control y en la creación de nuevas ideas y métodos de análisis y diseño.

Norbert Wiener y sus colegas, Bigelow y Rosenblueth, se reunieron para estudiar los problemas

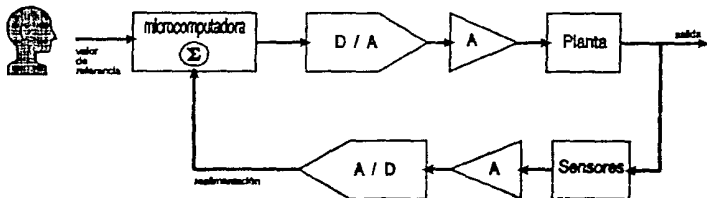


Fig. 1 Sistema de Control con Computadora Digital.

de los instrumentos de cómputo y su utilización en la fisiología y la guerra. Fueron ellos quienes acuñaron el término *cibernética* (del vocablo griego κυβερνήτης = timonel) en 1948.

Evans presentó su teoría del Lugar de Raíces también en este año, la cual suministra una exhibición gráfica de las propiedades de estabilidad de un sistema y permite la evaluación gráfica de la respuesta a la frecuencia del mismo. Debido a la guerra, estos descubrimientos fueron considerados secretos hasta la terminación de la misma, tras lo cual esta rama de la ingeniería se convirtió en una disciplina por derecho propio.

Las técnicas del dominio de la frecuencia continuaron dominando el campo de la ingeniería de control después de la Segunda Guerra Mundial con el uso de la transformada de Laplace y el plano de frecuencia compleja, principalmente usados para el análisis y diseño de sistemas de un solo lazo de realimentación.

También en 1948, Mason tradujo del alemán y publicó el libro "The dynamics of Automatic Controls", escrito por Oidenbourg y Sartorius (Dynamic Selbsttätiger Regelungen, Munich, 1944), donde se exponían los métodos analíticos en el dominio del tiempo, además de otros temas como ecuaciones en diferencias y retrasos. Este libro pasó casi inadvertido hasta que se popularizó el uso de sistemas muestreados, utilizando el dominio Z.

Otro caso notable fué la creación de tres importantes laboratorios en el Massachusetts Institute of Technology (MIT): los laboratorios de Servomecanismos, Instrumentación y Radiación.

Estos laboratorios contribuyeron mucho al avance del campo del control automático, aunque también muchas compañías privadas hicieron investigaciones en este campo que ayudaron a fortalecer la fundación de esta nueva ciencia.

Es en esta década de 1940 a 1950, cuando los métodos matemáticos y analíticos crecieron más en número y utilidad, y desde entonces muchos libros, artículos y documentos técnicos se han escrito, y la aplicación de sistemas de control en los campos militar e industrial ha sido extensa.

Con la llegada del Sputnik y la era del espacio otro nuevo ímpetu fue impartido a la ingeniería de control. Se hizo luego necesario el diseñar sistemas de control complejos y altamente exactos para sistemas de misiles y sondas espaciales.

En esta época nació también el campo del Control Óptimo, cuyas propuestas más importantes se deben a L.S. Pontryagin de la U.R.S.S. y R. Bellman de los E.U.A.

Debido a estos requerimientos, los métodos del dominio del tiempo han encontrado gran interés en las últimas décadas. Estos avances se han debido a estudiosos como Lyapunov y Minorsky.

Minorsky (quien fué el primero en hacer aplicaciones deliberadas de elementos no lineales en sistemas de lazo cerrado), se centró en el estudio del direccionamiento automático de una nave (servosistema de posición), y estableció las bases para la automatización industrial.

Ahora parece que la ingeniería de control del futuro deber considerar tanto el dominio del tiempo como el dominio de la frecuencia simultáneamente en el análisis y diseño de sistemas de control.

Es importante hacer notar que hay muchas otras fechas significativas y personas que tuvieron gran importancia en el desarrollo de la Ingeniería de Control. Una lista mas completa se halla en {Franklin et al., 4,11}.

CONTROL CLASICO Y CONTROL MODERNO

Durante los cincuentas el énfasis en la Teoría del Control se fortaleció en el desarrollo y uso de los métodos del plano s , y sobre todo en el uso del método del lugar de las raíces.

La teoría de control que floreció hasta esos años podría ser definida como teoría "convencional", la cual es todavía aplicada con gran eficacia a muchos problemas de diseño de control, especialmente a sistemas con una sola entrada y una sola salida.

La teoría de control que ha sido desarrollada desde entonces para el diseño de sistemas mas complicados y sistemas de entradas múltiples y salidas múltiples es llamada teoría del control "moderna", con una fuerte utilización de álgebra matricial y funciones de variable compleja. Los avances que se han logrado en los viajes espaciales sólo han sido posibles gracias a esta moderna teoría del control y el uso de computadores.

Algunos de los contribuyentes más importantes a la moderna Teoría de Control sobre todo en el campo de sistemas muestreados, son Jury, Kalman, Kuo, Takahashi y otros.

Referencias: {Dorf; Takahashi et al.}

SISTEMAS DE CONTROL COMPUTARIZADOS

Durante esta época de los 50's también empezó a ser posible la utilización de computadores analógicos y digitales en sistemas de control. Estos nuevos elementos de regulación poseían una habilidad para calcular rápida y exactamente.

Es decir, los computadores se usaron como elementos de adquisición de datos o bien de control, ya no solo dedicados a tareas de cómputo, lo cual ha tenido un enorme impacto en el diseño de los sistemas realimentados.

En 1958 se implantó el primer sistema de control de procesos industriales por computadora en una planta de la Texaco Oil Co., el cual era un sistema de control supervisor conectado a un conjunto de controladores analógicos ya existentes. {Alvarez y Alvarez, 1.3}

Debido a la marcada reducción de precios y aumento en la confiabilidad de estos dispositivos, para 1977 había ya más de treinta mil computadoras digitales para el control de procesos en los E.U.A., sobre todo en lugares donde muchas variables son medidas y controladas simultáneamente.

En la actualidad se estima que hay ya en las industrias alrededor de un millón de computadoras en Sistemas de Control, desde los más sencillos hasta los más complejos, siendo la naturaleza de la planta lo que dicta el tipo y potencia de la computadora a usar.

En un sistema como el de la figura 1, la computadora recibe y opera con señales digitales, ya que los datos que miden los sensores, tras ser amplificados y acondicionados, son convertidos de forma analógica a digital, y después la salida producida por la computadora es convertida de regreso a forma analógica y luego amplificada. {Cadzow, 72}.

TIPOS DE CONTROL DIGITAL

Hay varias formas en que la computadora puede ser utilizada en el control de un proceso. Las más importantes son:

CONTROL SUPERVISOR

La computadora sólo realiza una supervisión del proceso, fijando para ello los valores de referencia para los controladores analógicos (tradicionales) de lazos individuales. {Auslander, 276}

CONTROL DISTRIBUIDO

Se utiliza una red de computadoras, cada una de las cuales lleva a cabo el control a nivel local de una fracción del proceso. La información proporcionada o medida por cada computadora puede ser enviada a otra computadora en un nivel jerárquico superior. Esta arquitectura de controladores ofrece una gran flexibilidad. {Chesmond, 188}.

Hay varios tipos de arquitecturas de comunicaciones en red de área local para sistemas de control, siendo los más comunes los determinísticos, como el Token Ring y el Token Bus, si bien es posible utilizar en ciertos casos los modelos de tipo probabilístico (Ethernet y similares). En configuraciones sencillas, la comunicación entre computadoras de control también podría hacerse mediante señales analógicas o a través de puertos seriales. {Chesmond; 188, 411}.

**Esta página fue dejada
intencionalmente en blanco.**

CONTROL DIGITAL DIRECTO

Aquí la computadora está propiamente en el lazo cerrado de control. Los algoritmos que utilizan las acciones P, PI y PID son básicos también en este tipo de controladores. {Auslander, 276}. En este tipo de control las funciones de la computadora son principalmente cuatro:

- 1 Generar datos de la variable de referencia, fijados externamente por el operador a través de un dispositivo de entrada (normalmente un teclado).
- 2 Medir a través de transductores y convertidores adecuados las variables del proceso a controlar, y finalmente:
- 3 Generar el valor del error con 1) y 2).
- 4 Sintetizar (calcular) mediante un algoritmo adecuado el valor de salida del controlador con base en el valor de referencia y los valores medidos. {Chesmond, 403}. Estos cálculos se llevan a cabo mediante programación, en vez de equipo, por lo que nos existen las restricciones normalmente a éste último. {Auslander, 276}.

Es decir, se utiliza la computadora para manejar todos los datos de la planta, y con base en ellos se generan las acciones de control necesarias. {Chesmond, 402}.

En este tipo de control, la computadora reemplaza los controladores analógicos de un sistema convencional, la ecuación del controlador puede ser ejecutada por medio de un programa en la computadora, el cual sustituye al bloque controlador tradicional, y así producir una salida que controle la planta. Una gran ventaja de un sistema controlador digital es que puede sintetizar con facilidad funciones no lineales o de otro tipo. {Chesmond, 189}. Otra ventaja es que con la computadora se puede centralizar el proceso de control de muchos lazos (hasta 200 en sistemas grandes). {Auslander et al., (ICS) 280}.

Esta salida es convertida de forma digital a analógica, para luego ser llevada a un actuador, mientras que las variables medidas por transductores son convertidas de forma analógica a digital para que la computadora las pueda manejar.

Normalmente, la velocidad de la computadora le permite controlar varios lazos en forma secuencial, siendo las principales limitantes la constante de tiempo del proceso y la cantidad de datos que deban ser almacenados en la memoria.

En un principio, este tipo de instalaciones de Control Digital Directo resultaban muy costosas, ya que siempre se tenían controladores analógicos convencionales de respaldo, además de que se requerían varias computadoras para controlar la gran cantidad de variables de un proceso industrial.

Sin embargo, el uso de controladores digitales directos, utilizando algún lenguaje de alto nivel en tiempo real, y las altas velocidades en sistemas multiprocesadores son la orden del día, ya que se facilita el cómputo, monitoreo, almacenamiento y análisis de señales. Por otra parte, es fácil modificar los algoritmos de control.

Esta aplicación de las computadoras se vuelve muy eficiente y flexible, particularmente cuando el sistema es distribuido, es decir cuando el proceso utiliza un número de computadores enlazados entre sí. {Rao, 283,373}.

A la larga, el control digital directo en micro o minicomputadoras como elemento de realimentación y de control reemplazarán con mayor confiabilidad y eficiencia al gran número de dispositivos de control convencional que involucran una multiplicidad de lazos de realimentación para regular temperaturas, velocidades, etc., basándose en algoritmos del tipo PID. {Rao, 302}.

Estos sistemas de control digital directo, aún utilizan en su mayor parte los algoritmos de control PID o sus variantes. {Auslander et al., 1}.

RESUMEN

En este capítulo, se mencionaron algunas de las fechas y sucesos más importantes en la historia de la Ingeniería de Control, tanto clásica como moderna, así como los principales tipos de sistemas con control digital (con computadora), particularmente aquellos que usan los algoritmos de control digital directo, en su forma proporcional integral y derivativa.

**Esta página fué dejada
intencionalmente en blanco.**

1 DESARROLLO DE LOS PROGRAMAS

A continuación se describen brevemente las partes necesarias para que el sistema de control digital directo funcione correctamente y se mencionan los procedimientos que debe seguir el usuario para aprovechar este sistema de forma adecuada. Después se procede al desarrollo del sistema de software.

HARDWARE

El conjunto de programas que forman este sistema estará diseñado para usarse en una microcomputadora MicroBBC, con interfase para disco flexible, puerto paralelo Centronics, puerto para cassette, monitor de color, unidad impulsora de discos flexibles de 5.25"; una impresora es opcional.

También se necesita la planta a controlar, en este caso un motor de corriente continua, de 12 volts, acoplado a un tacogenerador.

Este motor es accionado por un amplificador de potencia (transistor NPN), que a su vez recibe su excitación de un convertidor digital analógico (DAC0831), el cual es controlado por la microcomputadora.

La computadora recibe la información de velocidad del motor mediante un conjunto de cuatro amplificadores operacionales (LM324), conectados como amplificadores de instrumentación, que alimentan a un convertidor analógico digital (interno), lo cual cierra el lazo de realimentación.

Para dar energía a la planta y los varios circuitos asociados es necesaria una fuente de poder que entregue 15V a 2A y -15V a 1A, los cuales serán regulados a +12V y -12V.

Todos estos bloques de hardware se ven en la fig. 2 y son descritos con mayor detalle en la sección dedicada a la planta.

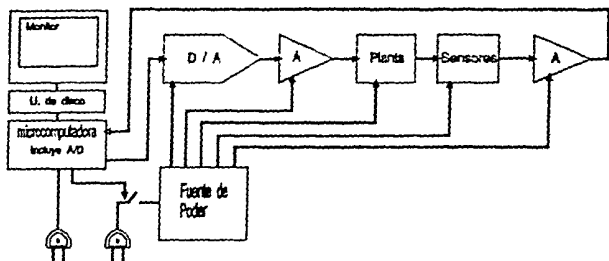


Fig. 2 Hardware del sistema.

SOFTWARE

El software necesario para el sistema es el conjunto de programas de análisis y control de la planta, todos ellos escritos en BASIC, junto con las utilerías para calcular ganancias, etc. Este software está contenido en un disco flexible de 5.25", con formato propio de la MicroBBC (Disk Filing System DFS).

OPERACION

Para operar el software, es necesario cerciorarse de que la planta y los circuitos asociados están correctamente conectados a la computadora, luego se enciende ésta y se introduce el disco de programas en la unidad de disco, se cierra la puerta, y en el teclado se tecldea la instrucción *EXEC !Boot <Enter> con lo cual automáticamente se carga el menú del sistema.

A continuación, se selecciona una de las opciones de dicho menú, oprimiendo el número deseado.

De presentarse algún error, el sistema se detendrá y aparecerá en la pantalla un mensaje de error, debiendo hacerse referencia al manual de la MicroBBC para su solución. En todo caso, es conveniente revisar que todo el equipo esté correctamente conectado.

ENFOQUE ESTRUCTURADO

La documentación del proyecto se elabora utilizando el enfoque estructurado de sistemas y se divide en varias secciones importantes:

- 1 La documentación del Sistema, acerca del Sistema en su totalidad. Aquí se incluye:
 - a) La especificación del usuario. Donde se describen los propósitos y objetivos finales del Sistema, tanto generales como particulares, desde el punto de vista del usuario.

- b) El esquema general de funcionamiento. Una breve referencia visual, en forma de diagramas de flujo o de bloques, mostrando el flujo y tratamiento de la información a través del Sistema.
 - c) La especificación para los programas y archivos de datos. Se describen los datos, estructuras y archivos usados por el Sistema o sus interfaces, con sus nombres, tipos, formatos y longitudes, así como los programas que manipulan dichos datos o archivos.
 - d) La seguridad del Sistema. Como resguardar la integridad de los datos almacenados, y en su caso, su respaldo y/o restauración, así como definir quiénes y de qué manera tendrán acceso al Sistema y sus archivos de datos.
 - e) Diseño de formas de entrada y salida de datos. Esbozos de cómo se hará la presentación visual de los datos, textos, gráficas u otra información a lo largo del sistema.
- 2 La documentación de la Programación, acerca de los programas y sus archivos de datos. Esto incluye:
- a) Una descripción general del conjunto de programas que forman el sistema, y para cada programa:
 - b) Una descripción breve del procesamiento realizado, sea en diagramas de flujo, pseudocódigo, etc.
 - c) Una descripción de los algoritmos particulares usados, con énfasis en cómo se está tratando la información.
 - d) Una lista o descripción de las subrutinas usadas.
 - e) Una explicación de las estructuras de datos usadas.
 - f) Un listado comentado, que incluya la descripción de las variables, procedimientos y funciones usadas.
- 3 La documentación del usuario, acerca de la manera de hacer uso del Sistema en su conjunto, incluyendo algunos ejemplos. Esto abarca:
- a) Una descripción del hardware necesario.
 - b) Una guía general de la operación y características del software.
 - c) Una breve guía de referencia del software.
 - d) Una sección de ayuda.
 - e) Una descripción de los posibles mensajes de error.
 - f) Un índice general y de referencias cruzadas.
- 4 Una evaluación del Sistema, que incluya:
- a) Una descripción de la interfase hombre — máquina.
 - b) Una relación de errores o limitaciones.
 - c) Una descripción de desarrollos potenciales.

ESPECIFICACIONES DEL SISTEMA

En forma general, los objetivos del sistema de control digital directo que se desea implantar en la microcomputadora, son el facilitar al usuario la captura, análisis y transformación de datos del comportamiento dinámico de una planta; y con ello, el cálculo de algoritmos específicos que le permitan controlarla automáticamente mediante la misma computadora.

En particular, se podrían desglosar los objetivos del sistema de la siguiente manera:

- 1 El sistema deberá estar integrado por un conjunto de programas escritos en un lenguaje de alto nivel, siendo posible acceder a ellos mediante un menú, y que otorguen al usuario las siguientes posibilidades:
- 2 Almacenar en un archivo en disco hasta 100 datos de comportamiento en el tiempo de la variable a controlar en una planta; en nuestro caso la velocidad angular de un motor de corriente directa, los cuales pueden provenir de tres fuentes distintas:
 - a) Medición directa del funcionamiento de la planta (a través de un convertidor A/D), como respuesta a una función de excitación. En este caso deberá ser posible calibrar, escalar o transformar las mediciones de forma tal que se representen de forma fiel en la computadora las variables medidas.
 - b) Tabla de valores obtenida por otros medios, en cuyo caso deberá poderse indicar la cantidad de datos que se van a almacenar.
 - c) Función analítica (algebraica), en cuyo caso deberá poderse especificar el dominio de la función a calcular, así como la cantidad de intervalos en que debe subdividirse. Deberá poderse incluir también la función escalón como parte de la ecuación.

En todos los casos deberá poderse almacenar un comentario o descripción (p.ej. fecha) con el conjunto de datos, así como sus unidades (opcional). También deberá ser posible desplegar en pantalla un catálogo de los archivos almacenados.
- 3 Dichos datos, una vez almacenados, deberán poderse manipular de las siguientes maneras:
 - a) Lista de datos por pantalla, incluyendo el comentario o descripción de los datos, número de dato, valor mínimo, valor máximo, porcentaje del valor máximo, coordenadas de origen de la gráfica.
 - b) Gráfica cualitativa en pantalla del conjunto de datos. Preferiblemente a color, con puntos, línea punteada o trazo continuo.
 - c) Graficación sobre los mismos ejes y proporciones de hasta tres conjuntos de datos (para comparar).
 - d) Interpolación de valores por el método de Lagrange, de modo que aparezca en pantalla el nuevo valor, en forma numérica y gráfica.
 - e) Ajuste de una recta por el método de mínimos cuadrados a un subconjunto especificado de datos, trazando la recta sobre los datos originales y mostrando los valores calculados de pendiente e intercepto, así como la correlación del ajuste.
 - f) Deberá ser posible calcular nuevas rectas cercanas a la primera, haciéndolas pasar por un mismo punto, cuyas coordenadas son las medias aritméticas de los valores muestrales

(centro de gravedad de los datos). Esto es estadísticamente válido {Spiegel, 220,228} y se requiere para hacer pequeñas modificaciones a los valores calculados, de manera sistemática y sin que pierdan validez.

g) Aplicación a la recta ajustada antes a los algoritmos de cálculo de polo y coeficiente (método logarítmico), o bien par de polos (y par de coeficientes) (método de diferencias de Lipka), o bien método de Ziegler Nichols (retraso y polo simple).

- 4 También deberá poderse modificar el archivo de datos, de manera tal que sea posible afectarlos con todo tipo de manipulaciones algebraicas, sea con constantes o variables, incluyendo el número de dato, interpolación de valores, traslación de ejes, etc., o bien combinación con otro conjunto del mismo tipo y con el mismo número de datos, siendo posible almacenar en disco el conjunto ya transformado, sin alterar el original.
- 5 El sistema deberá incluir también los siguientes programas que interactúan con la planta:
 - a) Programa para ejercitar (precalentar) el motor, en forma cíclica y al azar, sea antes de calibrar los sensores, o bien antes de usar uno de los siguientes programas:
 - b) Programa para ejercer manualmente (en lazo abierto) el control de velocidad del motor (mediante el teclado), desde su velocidad mínima hasta la máxima, desplegando en forma numérica y gráfica dichos cambios.
 - c) Programa para ejercer automáticamente el control digital de velocidad del motor, mediante un algoritmo Proporcional, Proporcional + Integral, Proporcional + Integral + Derivativo, para tabular y luego mostrar su respuesta al escalón en lazo cerrado.
- 6 Dado que una de las interfases de la planta con la computadora (la de salida) se basa en el uso de amplificadores de instrumentación, deben incluirse programas de utilidad para realizar los siguientes cálculos:
 - a) Cálculo de un par de resistencias comerciales en paralelo para obtener un valor deseado.
 - b) Cálculo de resistencias comerciales para obtener una ganancia deseada en un amplificador operacional.
 - c) Cálculo de sensibilidad de ganancia de un amplificador operacional dados los valores de resistencias, sus tolerancias de error y el voltaje de entrada al amplificador.
- 7 Dado que el sistema de control se basará primordialmente en los algoritmos PID, también deberán poderse calcular los parámetros de estos, en forma continua o discreta, a partir de los parámetros R, L y τ de las reglas de Ziegler y Nichols, así como el tiempo de muestreo T.

ESQUEMA GENERAL

El esquema general de funcionamiento del Sistema de Control digital Directo se ve en las figuras 3 a la 7.

En la figura 3 se aprecian los cuatro bloques esenciales del sistema. El primero de ellos es el de calibración de sensores, de donde se pasa al subsistema de simulación de respuesta al escalón o a otras funciones excitadoras, tras de lo cual se hace un análisis de los datos obtenidos, transformándolos en varias ocasiones, para llegar al modelo de la planta y al diseño de

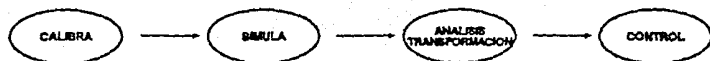


Fig. 3 Sistema de Control Digital Directo

controladores digitales. Finalmente está el bloque de control, donde se aplican dichos controladores a la planta.

En el Subsistema de Calibración (fig. 4) lo que hace el usuario es aplicar a la planta una excitación en lazo abierto, de valor constante y conocido mediante un medidor externo (en este caso un multímetro digital). A la vez, la computadora mide el valor aplicado tanto de salida como de entrada, representándolos en forma digital como una cantidad hexadecimal sin dimensiones, en el rango hex 00 a hex FF. Para ello la computadora realiza 100 muestras de cada valor y hace luego un promedio, y al comparar con los valores proporcionados por el usuario, calcula los factores de

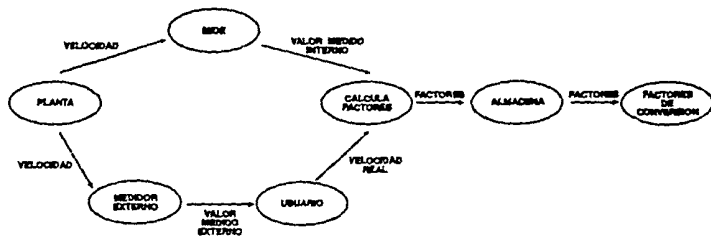


Fig. 4 Subsistema de calibración

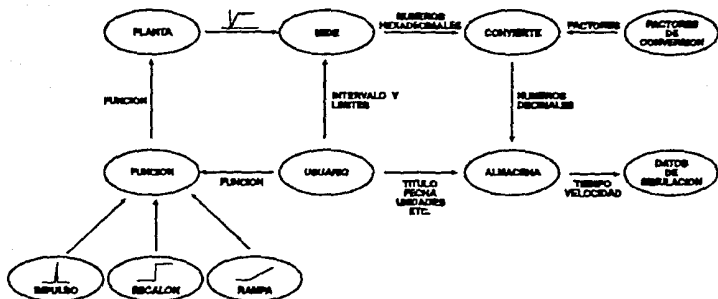


Fig. 5 Subsistema de Simulación.

conversión respectivos para cada canal de conversión. Dichos factores son almacenados en disco con el nombre de archivo D.Sensor, los cuales se utilizarán en los siguientes subsistemas.

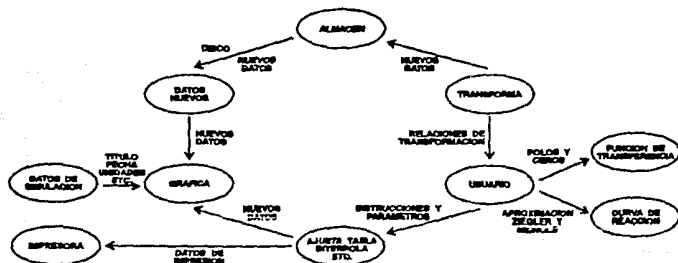


Fig. 6 Subsistema de Análisis y Transformación.

En este subsistema de control (fig. 7), se aplican los controladores digitales PID calculados, y se obtienen las respuestas del sistema en lazo cerrado. Para ello, el usuario fija la variable de referencia deseada, las constantes del algoritmo PID y el número de ciclos que ha de aplicarse. La computadora verifica la validez que estos datos antes de aplicar la excitación a la planta. Los datos obtenidos se almacenan para después graficarse o analizarse.

ARCHIVOS DE DATOS

Dadas las modestas necesidades impuestas por las especificaciones anteriores, en lo que toca a archivos de datos, se necesitará un formato único para todos ellos, ya que todos serán del mismo tipo.

Los datos se almacenarán en un estructura de archivo serial, cuyos tres primeros elementos serán del tipo alfanumérico (textual), de longitud variable L ($0 \leq L \leq 40$ caracteres), el primero para el comentario o descripción del conjunto de datos, el segundo para las unidades de la variable independiente y el tercero para las unidades de la variable dependiente; posteriormente n ($1 \leq n \leq 100$) pares de datos con formato de punto flotante de precisión sencilla (4 bytes mantisa y 1 byte exponente), para las variables independiente (usualmente tiempo) y dependiente (usualmente velocidad).

De acuerdo a lo anterior, y considerando que en el caso de la MicroBBC cada variable de texto cuya longitud sea L ocupa en disco $2+L$ bytes, y cada variable de punto flotante usa 6 bytes, tendremos que la longitud máxima de un archivo de datos de este tipo será $3x(2+40) + 100x2x6 = 1326$ bytes, y la mínima $3x(2+0) + 1x2x6 = 18$ bytes. {Coll et al. vol II, 70}.

De esta manera, el flujo de información entre los distintos programas del sistema será a través de archivos de datos de esta naturaleza.

El primer programa en generar un archivo de datos será el programa de calibración de sensores, que almacenará cuatro pares de datos, a saber: el número de sensor (puerto del convertidor analógico digital) y el correspondiente factor para convertir la cifra adimensional a un valor con unidades que represente a la variable medida. El nombre del archivo será D.sensor, y el texto descriptivo que se almacenará con él será: "Factores de conversión". Ninguno de los elementos tendrá unidades.

El factor de conversión será calculado al dividir el valor observado por el operador en un medidor externo al sistema (multímetro o tacómetro), entre el valor medido por el computador en el puerto de conversión A/D.

El siguiente programa que genere un archivo de datos será el que realice la captura de datos de la planta, cuyo texto descriptivo variará según los datos en cuestión. Este programa generará hasta 100 pares de datos con unidades, provenientes de alguna de las tres fuentes descritas en la especificación 2: experimental, tabular o analítica.

En el caso experimental, el programa leerá el archivo preexistente de factores de conversión para dar significación y unidades a las cifras medidas, que provendrán de los canales de conversión analógico digital.

En el segundo caso, los datos serán introducidos por el operador a través del teclado uno por uno y en el último caso, serán calculados usando una ecuación que el operador proporcione (en BASIC).

Otro programa generador de un archivo de datos será el programa de transformación, que tomará uno o dos archivos de datos preexistentes, y los transformará o combinará utilizando relaciones dadas por el operador, para formar un solo conjunto de datos transformado, el cual se almacenará. Aquí las variables también podrán tener unidades, y el conjunto resultante un texto descriptivo.

Finalmente, el programa de control, tanto en lazo abierto como en lazo cerrado, utilizan el archivo de datos de factores de conversión. En particular, el control de lazo cerrado crea también un archivo de datos similar al producido por el programa de captura.

Los programas que utilicen archivos de datos preexistentes serán:

El programa de captura de datos y el programa de transformación descritos arriba, además del programa de graficación, el cual leerá el archivo de datos para listar sus valores y graficarlos en la pantalla.

El resto de los programas requeridos por las especificaciones no tendrán interacción con ningún archivo de datos.

SEGURIDAD DEL SISTEMA

En lo que toca a la seguridad de acceso al sistema de software, no se implantará ningún tipo de dispositivo de restricción, ya que se pretende que pueda ser usado libremente por cualquier persona.

La única protección que asignará es el bloqueo de algunos archivos de datos y programas (atributo de sólo lectura) para evitar que se borren accidentalmente. Lo mismo se hará con el archivo D.sensor que almacenará los factores de conversión. Esto se logra con la instrucción del sistema operativo *ACCESO.

En cuanto a la recuperación del sistema en caso de pérdida accidental de datos, tampoco es indispensable tomar provisión alguna, ya que el proceso de simulación y análisis se puede repetir con relativa facilidad en cualquier momento. De cualquier forma, sí es recomendable contar con un respaldo en disco de los programas y de los archivos de datos, particularmente el archivo D.Sensor, además de sus respectivos listados.

En el caso de darse una pérdida de energía eléctrica o desconexión accidental de la computadora, el sistema contará (a través del hardware) con una protección para desconectar a su vez toda alimentación eléctrica a la planta, para evitar su acción y cualquier posible desperfecto o peligro para el operador. También es posible efectuar esta desconexión oprimiendo la tecla ROMPE.

FORMATOS DE ENTRADA/SALIDA

A continuación se describe el aspecto que se dará a la información, tanto para introducirla a la computadora como para extraerla de ella, en función de las posibilidades de la computadora y de los requerimientos de datos descritos antes.

Se utilizará el modo de pantalla 1, ya que éste ofrece cuatro colores, (blanco, negro, rojo, verde) además de una buena resolución para gráficas (320 x 200 píxeles). La longitud máxima de cada línea de texto en este modo es de 40 caracteres, con un máximo de 25 líneas en pantalla. Cada letra ocupa un espacio de 8x8 píxeles.

Todos los programas serán presentados con una carátula, con un título en la parte superior que indique cual programa se está usando. La última línea será una línea de comandos, donde se presentará un breve texto que indique al operador lo que debe teclear, las opciones de que dispone, o bien el resultado de su acción previa.

El centro de la pantalla será el área principal, y en ella se presentarán instrucciones, nombres de variables, relaciones de transformación, gráficas, tablas, etc.

Todos los programas se presentarán sobre fondo negro, con los textos y valores numéricos en blanco, reservándose el rojo para la carátula y gráficas, y el verde para opciones de programas y para gráficas.

Los valores numéricos se presentarán como cantidades de punto flotante con tres decimales, en notación científica de ser necesario.

Las gráficas tendrán un enfoque cualitativo más que cuantitativo, por lo que los ejes en que se dibujen no estarán subdivididos ni acotados, solamente se incluirá el texto que describe el conjunto de datos, y las unidades, de haberlas.

Las tablas se presentarán en columnas adyacentes, mostrando el número de dato, el valor de la variable independiente, el valor de la variable dependiente y el porcentaje de ésta con respecto al valor máximo. El despliegue en pantalla de la tabla se detendrá por páginas de 15 líneas, para poder examinar los datos. Para pasar al siguiente grupo de datos se oprime la tecla de CAMBIO.

DESCRIPCION DE PROGRAMAS Y PROCEDIMIENTOS.

En esta sección se describen los programas y procedimientos usados en el Sistema de Control Digital Directo, los listados de programas en BASIC se agregan al final de cada sección.

Algunas de las rutinas (p.ej. PROCara) se usan sin cambio en varias partes del Sistema, por lo cual sólo se describen una vez.

En su caso, se describe cuáles variables se utilizan y de qué manera dentro de cada procedimiento.

Para ayudar a la comprensión de los programas en BASIC, hágase referencia al resumen de comandos que se presentan en la descripción del lenguaje BASIC (en el apéndice).

PROGRAMA Menú.

Este es el programa inicial del sistema, en él se selecciona una opción del usuario, que lo lleva a realizar una función específica de las que se dispone en el Sistema de Control Digital Directo. Dichas funciones son:

- 1 Catálogo de archivos.
- 2 Calentamiento de motor.
- 3 Calibración de Sensores.
- 4 Entrada de Datos.
- 5 Graficación, Tabla, Ajuste, Interpolación.
- 6 Transformación de Datos.
- 7 Cálculos varios.
- 8 Control Digital Directo.

Dado que ocupan poco espacio y no necesitan traer archivos de datos del disco, las funciones 1, 2 y 3 (catálogo de archivos, calentamiento de motor y calibración de sensores), son sólo procedimientos dentro de este mismo programa Menú, en vez de programas independientes.

Las funciones de la 4 a la 8 están contenidas en programas separados, los cuales son llamados desde este menú.

Cada función se examinará en detalle mas adelante.

Segmento principal.

Aquí se prepara la pantalla en el modo 1, se borran todas las variables de la memoria, despliega la carátula de presentación del programa y luego las opciones disponibles, actuando posteriormente en base a la selección hecha.

No existen variables propias del programa principal.

PROCmenu

Despliega una lista de las opciones disponibles (PROCprogs), reduce la ventana de textos a una sola línea, pide al usuario que teclee el número de la función que desea realizar, y con él realiza una transferencia de control al procedimiento o programa en cuestión.

Maneja la variable local `opc%` para seleccionar una opción y efectuar la transferencia de control.

PROCprogs

Este procedimiento despliega las opciones del menú dentro de la carátula, remarcando con verde los números 1-8, es decir las opciones disponibles.

Maneja la variable local `i%` para remarcar los números en verde.

PROCCat (opción 1 del menú).

Este procedimiento despliega en pantalla los nombres de todos los archivos almacenados en el disco. Para organizarlos, los archivos de datos normalmente van precedidos de los identificadores D., 1., 2., etc. Los archivos de programas no llevan prefijos. Este procedimiento no maneja variables.

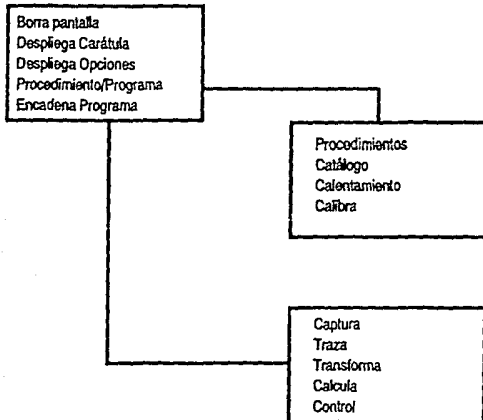


Fig. 8 Pseudocódigo del Programa Menú.

PROCCara

Este procedimiento despliega la carátula del programa, que consiste de un marco rojo (color 2) y letreos blancos (color 3). El título de la carátula se despliega en color verde (color 1). PROCCara se usa también en todos los otros programas del Sistema.

El procedimiento recibe el parámetro `txt$`, que es el título de la carátula.

Las variables locales que maneja el procedimiento son: `i%`, contador para desplegar el marco, `z%` y `d%` que sirven para centrar el título dentro del marco.

Aquí también se redefine la tecla de función número 10 (ROMPE) para reiniciar el sistema en cualquier momento.

PROCCalibra (opción 3 del menú).

Calibración por software de los medidores de instrumentación conectados al puerto de conversión analógico digital de la microcomputadora.

Físicamente integrado al Programa MENU, este procedimiento obtiene los valores adquiridos a través de los cuatro canales convertidores analógico digital (seleccionados con la variable local `j%`), pide al operador los valores equivalentes obtenidos en forma externa y calcula un factor de conversión de la cifra binaria del CAD a un valor de voltaje, velocidad, etc.

Primero restaura las ventanas de texto y gráficas a sus valores iniciales, despliega la carátula, prepara las variables (ver PROCprepara), inicia el contador de tiempo en cero y repite cuatro veces el procedimiento de medición (ver PROCmide).

Al terminar borra la pantalla, apaga el motor, guarda los valores obtenidos (PROCGuarsensor), bloquea el archivo contra escritura y vuelve al Menú Principal (PROCFin).

PROCprepara

Aquí se dimensionan las variables para recibir los valores del CAD (100 valores), declara (inicializa) cuatro canales de conversión (*FX16,4) y enciende el motor a velocidad media, esperando a que se oprima una tecla para continuar. La variable `@%` fija el formato de despliegue de los valores numéricos.

PROCmide

Recibe el parámetro `i%` que es el número de canal por medir. Define e inicializa las siguientes variables locales:

<code>valmin</code>	valor mínimo muestreado
<code>valmax</code>	valor máximo muestreado
<code>valmed</code>	valor medio calculado
<code>k%</code>	contador de muestreo (hasta 100)
<code>SUMval</code>	suma de valores muestreados
<code>valobs</code>	valor medido externamente por el operador
<code>dif</code>	diferencia máxima de valores muestreados
<code>pct</code>	diferencia en porcentaje

Prepara la planta (PROcInicio) y ejecuta un ciclo de 100 muestras, esperando cada vez a que termine el proceso de conversión en el canal en cuestión. Calcula mínimo, máximo y promedio de cada ciclo de muestreo, muestra estos valores y la máxima diferencia en porcentaje, y luego pide el valor observado por el operador (en volts) y calcula el factor de calibración de cada sensor.

PROccalcmax

Recibiendo el parámetro a, (cantidad a comparar), calcula los valores máximo y mínimo (adimensionales) con cada valor obtenido por muestreo del canal de conversión.

PROcguarsensor

Abre un archivo de salida con el nombre "D.sensor", escribe en el el título "Factores de conversión" y guarda los números de canal de conversión, cada uno seguido del valor (número real) de calibración encontrado, cerrando después el archivo. Las variables locales son el número de archivo y el número de canal de conversión.

PROcfin

Pide oprimir una tecla cualquiera, después de lo cual nos lleva al programa principal.

PROcbip

Hace un par de sonidos para alertar al operador, al momento de pedírsele su intervención. Se usa a lo largo del Sistema.

PROcsaldig

Recibe el parámetro val%, el valor digital de salida al puerto paralelo de la computadora. Selecciona dicho puerto e "imprime" un byte.

PROctstmot (opción 2 del menú).

Este procedimiento forma la rutina de calentamiento para el motor. Prepara la planta (PROcInicio), y al oprimir una tecla enciende y apaga el motor en forma consecutiva, por intervalos crecientes y decrecientes (PROctST), y luego al azar (PROcAzar), para finalmente dejarlo encendido en su valor máximo, pidiendo se oprima una tecla para regresar al menú principal.

La variable local i% es el contador de periodos de encendido y apagado. El valor $m\% = 2500 + \text{RND}(500)$ dá aproximadamente un minuto para ciclos crecientes y uno para ciclos decrecientes.

PROcInicia

Aquí se declaran los cuatro puertos de conversión (*FX16,4), se selecciona el puerto paralelo (*FX5,1 y *FX6) y se enciende el relevador de alimentación al motor.

PROctST

Este procedimiento enciende y apaga el motor de cero a plena escala (&FF), en intervalos crecientes y decrecientes.

PROCAzar

Aquí se envían al motor 500 valores al azar de alimentación entre cero y plena escala. Las variables locales son i% (contador de iteraciones) y num% (valor al azar).

PROCjoycnt

Este procedimiento cuenta las veces que haya sido apretado el botón de disparo asociado al canal de conversión no. 1, en un lapso de 49 centisegundos, es decir que puede contar pulsos aplicados en dicho punto. La variable local ant% detecta si se pulsó el interruptor de disparo del joystick.

Es una alternativa que sería lenta, —por lo cual realmente no se implantó— con respecto al tacómetro digital, que podría utilizarse en su lugar para la calibración de sensores. Es un posible desarrollo a futuro.

PROGRAMA Captura

El programa de captura de datos es llamado desde el Menú Principal, y nos permite almacenar datos en un archivo en disco, ya sea que estos datos provengan de una ecuación, tabla o bien directamente muestreados del canal 1 del puerto analógico digital (datos experimentales).

Estos datos se almacenan en un archivo cuyo formato está estandarizado con respecto a los otros programas, para que pueda ser leído en forma compatible por ellos.

Este formato es: la descripción de los datos (texto), las unidades de la variable independiente (texto), las unidades de la función (texto), y luego n% pares de datos (números reales de precisión simple —32 bits—) de la variable independiente seguida de la función.

Los otros programas leerán este archivo de la misma manera, y calcularán la cantidad de datos de acuerdo al tamaño del archivo.

Segmento principal

Pone el modo de pantalla 1, borra las variables de memoria, inicia el funcionamiento de la planta (PROcinicia) y despliega la carátula del programa. Pregunta cuantos datos serán almacenados (PROccant) y que origen tendrán dichos datos (PROcfuente). Guarda los datos y transfiere control al programa de graficación.

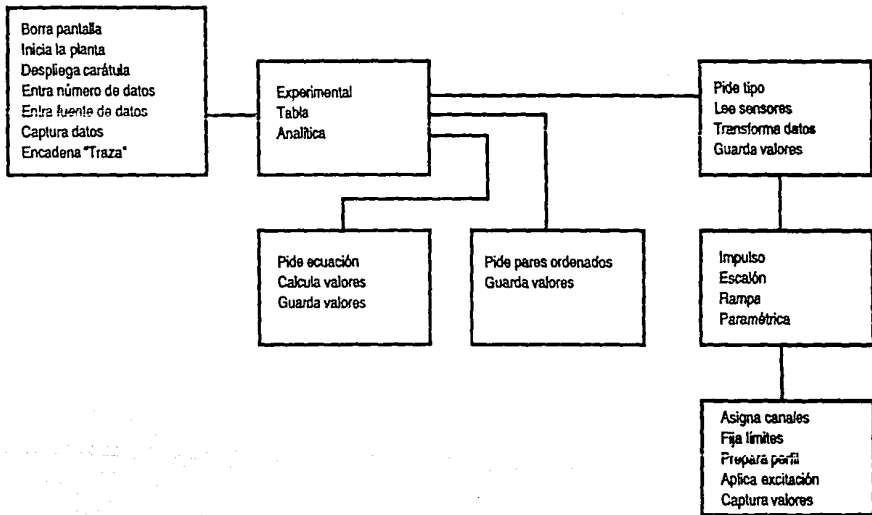
PROccant

Pide el número n% de datos a almacenar, donde n% es una variable entera (max 100), y reserva espacio en memoria para alojarlos. Es de notarse que n% es una variable global. El arreglo w(4,n%) reserva 5 columnas (0-4) una para el tiempo y 4 para los canales de conversión. El arreglo val%(n%) son los valores de la función excitadora.

PROcfuente

Propone las opciones de origen para los datos (Experimental, Tabular o Analítica), y en base a la elección que haga el operador, transfiere control al procedimiento adecuado, a través de la variable local org\$.

Fig. 9 Pseudocódigo del Programa Captura.



FNkey

Espera a que se oprima una tecla, y la valida al compararla contra la cadena `opcS`, que se dá como parámetro. Espera en un ciclo hasta que la tecla oprimida sea válida.

PROCanalit

Procedimiento para capturar datos, a partir de una función analítica, la cual se nos pide, así como el rango de la variable independiente bajo el cual debe calcularse la función. Los puntos donde se evalúa la función se ven discretizados en función del número de datos que se definió en `PROCcant`. Después los datos son guardados.

Estos datos no ameritan conversión con los factores de calibración mencionados antes, ya que no provienen del puerto de conversión analógico digital, antes bien, la función que se proporcione ya debe contemplar el escalamiento adecuado.

Las variables locales son:

<code>wS</code>	función de <code>t</code> , en forma de texto.
<code>:0, :1</code>	límites de la variable independiente
<code>dt</code>	incremento de la variable
<code>t</code>	valor de la variable en cada punto
<code>i%</code>	contador de iteraciones

PROCTabla

Un procedimiento alternativo para introducir datos a la computadora es a través de una tabla de valores, los cuales se teclean directamente como pares ordenados de variable y función, tantas veces como datos se hayan declarado en `PROCcant`. Estos datos tampoco ameritan conversión con los factores de calibración. Aquí la única variable local es el contador `i%`.

PROCexper

Selecciona el tipo de entrada que se dará a la planta (`PROCCentra`). Después lee los valores de conversión (ver `PROCCeensor`), transforma los datos recibidos y luego los guarda en disco.

PROCCentra

Despliega las opciones disponibles para entradas experimentales: Impulso, Escalón, Rampa (es decir, medición de respuesta transitoria) y mediciones paramétricas (de estado permanente). Según la que se seleccione se asigna una variable global (`funS`) para indicar el tipo de función. Si se desea una medición paramétrica transfiere el control al procedimiento `PROCCanal`, de otro modo al procedimiento `PROCCfza`.

PROCCanal

Este procedimiento se usa en el caso de mediciones paramétricas, siendo posible tener como variable independiente (variable global `ca%`) el valor digital de salida o bien el voltaje de entrada a la armadura del motor.

Como salida (variable global cb%) es posible medir el voltaje del tacogenerador, el contador de pulsos (disparos del joystick) o bien la corriente de entrada (calculada a partir de voltajes).

Una vez seleccionadas las variables se pasa al procedimiento PROCmide.

PROCmide

Se usa solamente en las mediciones paramétricas. Reestablece tres canales de conversión, despliega un aviso de captura de datos. Inicia un ciclo de n% mediciones, donde n% es el parámetro de entrada.

Para cada medición calcula el valor de entrada dividiendo la escala máxima entre n%, aplica este valor a la salida digital, y dependiendo de las elecciones hechas en cuanto a las variables, asigna los valores al arreglo w(ca%,i%) y w(cb%,i%).

Cuando hace conversiones analógico digitales, toma veinte mediciones y hace un promedio (ver FNconvierte). Al terminar apaga la planta, y calcula la corriente de armadura si ésta se solicitó como salida.

PROCespera

Espera a que expire un contador local i%, hasta el valor j%, antes de regresar el control al lugar de donde fué llamado.

FNconvierte

Tiene como parámetros el número de veces que hay que hacer la conversión, y el canal asignado para ello. La variable local k% es el contador de veces, y suma% es la suma de las conversiones efectuadas dividida entre el número de veces, es decir el promedio.

PROCcalcula

Con el contador local i%, calcula la caída (diferencia) de voltajes en la resistencia de entrada a la armadura ($R=3.3$ ohms), para después dividir entre este valor y obtener la corriente en amperes, asignándola como nuevo valor de la variable de salida.

PROCfza

Aquí se forzará la función de excitación sobre la planta, para lo cual se asignan los canales ca%=0 (tiempo) y cb%=1 (canal de conversión 1), se toman los límites entre los cuales se aplicará la función (PROClimites), se calculan los valores de la función excitadora (PROCimp, PROCesc o PROCram) y luego se aplica ésta (PROCaplica), para luego regresar a PROCentra y de allí a PROCexper.

PROClimites

Se piden los límites entre los cuales se aplicará la función excitadora, como un porcentaje entre cero y 100. Estos valores se transforman a escala digital de salida (0-255). Las variables locales a% y b% sirven para este propósito, mientras que wa% y wb% (calculadas) son globales. mag% es la diferencia entre estos límites.

PROCimp

Aquí se calculan los valores clave de la función impulso, para luego calcular el resto con PROCperfil. val(i%) son los valores de la función, mientras que inc es el incremento entre valores sucesivos.

PROCesc

Igual que PROCimp pero para función escalón.

PROCram

Igual que PROCimp pero para función rampa.

PROCperfil

Calcula el resto de los valores de la función excitadora, en función de los valores inicial y final, y el incremento. El parámetro j% es el primer valor que hay que calcular en el ciclo.

PROCaplica

Aquí finalmente se aplica la función excitadora, se reestablece un solo canal de conversión (el 1).

Despliega un aviso para oprimir una tecla, y entonces fuerza un inicio de conversión en el canal 1, asigna el tiempo inicial a t) y ejecuta n% ciclos, aplicando los valores de función calculados, muestreando en cada caso el canal de conversión, y asignando estos valores al arreglo w(0,i%) y w(1,i%) (tiempo y función respectivamente). Al terminar apaga el motor.

PROCaviso

Informa al operador que se están capturando datos.

PROCtransforma

Aquí se multiplica cada uno de los valores de la función por el valor de calibración de su canal de conversión, para pasar de una cifra sin dimensiones medida por el CAD a un valor equivalente con unidades de voltaje, velocidad, etc.

PROCguardatos

Este procedimiento pide el nombre de archivo que se asignará al conjunto de datos capturados, una descripción para los mismos, así como las unidades de la variable independiente y de la función.

Abre ese archivo para salida de datos, escribiendo en él las variables anteriores, seguidas por los pares de datos: variable independiente y función. Al terminar cierra el archivo.

PROCleesensor

Aquí se abre el archivo D.Sensor como entrada, y se leen los cuatro factores de calibración de sensores, los cuales se utilizan para modificar los valores de la función cuando ésta se obtiene experimentalmente. Al terminar se cierra el archivo.

PROCprompt

Alerta al operador para que oprima una tecla.

FNu(t)

Aquí se efectúa la función escalón, introducida por Oliver Heaviside, tal que $u=0$ para $t \leq 0$ y $u=1$ para $t > 0$. El parámetro t puede ser una expresión algebraica.

PROCbip, PROCsaldig, FNjoyent, PROCinicia y PROCcaratula ya fueron descritos antes.

PROGRAMA Traza

Este programa es el más complejo y extenso de todos, su función es el presentar tablas de datos y tres o más gráficas en la pantalla, además de poder ajustar una recta por el método de mínimos cuadrados a un subconjunto de los datos desplegados, interpolar valores por el método de Lagrange, calcular coeficientes y constantes de tiempo por el método de Lipka, el de Ziegler y Nichols, y por el método logarítmico.

Segmento principal.

Aquí se selecciona el modo de pantalla 1, se limpia la memoria y se dimensionan las variables a leer $t(200)$, $w(200)$, además de dar espacio para coeficientes de Lagrange $C(100)$.

Se inician variables y se enciende la fuente de poder del motor, se despliega la carátula del programa, se pide el nombre del archivo, se leen sus datos, se calculan máximos y mínimos y se entra a un ciclo (PROCciclo) de traza de gráficas, listado de datos, etc. Al terminar se regresa el control al menú principal.

PROCnombre

Aquí se lee el nombre del archivo cuyos datos se van a listar o a graficar. Se lee el título de los datos, las unidades y se calcula la longitud del archivo para determinar el número de datos.

PROCleedat

Aquí se leen del archivo en disco los n pares de datos a graficar.

PROCmaxmin

Se calculan los máximos y mínimos de los datos.

PROCcalctmax

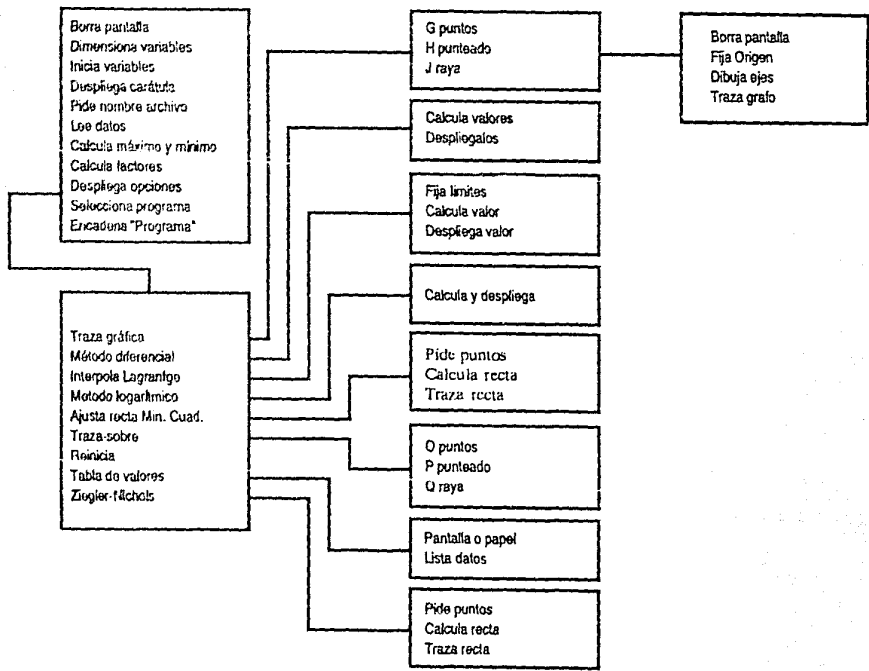
Usado por PROCmaxmin para calcular los valores máximo y mínimo de la variable independiente.

PROCcalcwmax hace lo mismo para los valores de la función.

PROCcalcfac

Este procedimiento calcula el origen para las gráficas, a partir de los valores de entrada de las variables, considerando que toda la pantalla tiene coordenadas gráficas de (0,0) a (1280,800), le resta el espacio que ocupa el marco y deja un espacio libre, quedando un área para gráficas de 480

Fig. 10 Pseudocódigo del Programa Traza



x 960 (120 x 240 pixeles), también calcula a partir de los valores máximos el factor de conversión para las variables para que el total de datos quepa en pantalla.

PROCCiclo

Es un ciclo continuo en el cual la opresión de una tecla selecciona alguna de las funciones disponibles, regresando a este ciclo una vez que la función se ejecutó. Antes que nada se despliegan las opciones (Gráfica, otra gráfica, tabla de datos, interpolación, ajuste de mínimos cuadrados, método logarítmico y método de diferencias) y se selecciona el color inicial para las gráficas. Cuando se selecciona la opción de fin, se retorna al segmento principal y de allí al menú del sistema.

PROCmasgraf

Este procedimiento se usa para trazar la segunda o tercera gráficas sobre la anterior, utilizando los mismos factores de despliegue, se lee el nombre del archivo cuyos datos se desea graficar, se leen sus datos y se grafica con PROCtraza.

PROCtraza

Aquí se traza la gráfica principal. Primero se define una ventana para textos, se imprime el título, se fija el origen de la gráfica, se trazan los ejes, se selecciona un color y luego se trazan los puntos de datos, de acuerdo a los factores de despliegue calculados por PROCcalcfac.

PROCFijorg

Aquí se fija el origen de las gráficas, se define la ventana para gráficos y el cursor gráfico se mueve a las coordenadas (0,0).

PROCdibeje

Aquí se dibujan un eje, moviéndose a un extremo dado y trazando una recta al otro extremo.

PROCTabla

Este procedimiento define una ventana de texto, fija el formato de despliegue de variables (@%), pregunta si descamos la tabla por impresora o por pantalla, en base a lo cual selecciona el puerto adecuado, para después imprimir el nombre del archivo de datos, el título de los datos, el origen de la gráfica y los valores mínimos y máximos de las variables.

Se imprime un encabezado con las variables y sus unidades y a continuación los n pares de datos, con su número secuencial y el porcentaje del valor de la función con respecto al valor máximo. Al terminar se borra la pantalla y regresa a pedir otra opción.

PROCAjusta

Este es el encargado de utilizar el método de mínimos cuadrados para hacer el ajuste de una recta a un subconjunto de datos. Primero define una ventana de textos, pide el intervalo en el cual hará el ajuste, calcula todas las sumatorias, la media, varianza, desviación estándar y con ello el intercepto y la pendiente de la recta, además de su correlación estadística (bondad del ajuste). Selecciona un color y traza la recta, además de desplegar los valores calculados.

PROCziegler

Este es el encargado de calcular la pendiente (máxima) de la recta que pasa por un subconjunto de puntos dados, dando además el coeficiente, el retardo de transporte y la constante de tiempo de la ecuación equivalente con el método de Ziegler y Nichols. Es posible ajustar el valor del coeficiente A para apearse a condiciones iniciales cero, recalculando la constante de tiempo y el retraso L.

PROCCraya

Este paso es el encargado de trazar una recta recién calculada, usando su pendiente e intercepto.

PROCdata

Aquí se despliegan los datos calculados por PROCZiegler.

PROCdife

Esta es la realización del método de diferencias de Lipka, para calcular el par de constantes de tiempo o par de coeficientes, a partir de la pendiente e intercepto de una recta ajustada. Los valores se imprimen después de pedir el incremento h entre valores sucesivos de la variable independiente.

PROCloga

Implantación del método logarítmico para el cálculo de un coeficiente y una constante de tiempo, a partir de la recta recién ajustada.

PROCnueA

Aquí es posible modificar el valor del coeficiente A recién calculado con el método logarítmico, para refinar el ajuste de la ecuación a los datos experimentales, se calcula la nueva pendiente e intercepto de acuerdo al nuevo valor de A, haciendo girar la recta alrededor del punto cuyas coordenadas son las medias muestrales de los puntos experimentales.

PROCinterpola

Este es el procedimiento encargado de ejecutar la interpolación de Lagrange. Se le dan los puntos extremos a considerar para calcular coeficientes de Lagrange, y luego pide el valor de la abscisa a interpolar, trazando luego el punto y dando sus coordenadas.

PROCmulti

Este segmento se encarga de calcular los productos de diferencia de términos generados al momento de hacer una interpolación de Lagrange.

PROCcoef

Aquí se calculan los coeficientes de Lagrange propiamente dichos, utilizando el procedimiento PROCmulti y las reglas de interpolación.

FNLagrange

Aquí se calcula la ordenada correspondiente al valor dado de la abscisa, utilizando los coeficientes calculados antes.

PROCOpc

Lista las opciones disponibles en el programa TRAZA.

PROCSubind

Aquí se generan los pequeños subíndices que aparecen en el encabezado de las tablas de datos.

PROCTecla

Espera a que una tecla sea oprimida para con ella seleccionar una opción de funcionamiento del programa.

PROCIncia

Enciende la fuente de poder y prepara variables para el funcionamiento del programa.

PROCFinal

Borra la pantalla y muestra los programas a los que se puede ir al terminar el programa TrazA.

PROCCara y PROCBip ya fueron descritos antes.

PROGRAMA Transfo

Este programa nos permite modificar los archivos de datos obtenidos con el programa de captura, de manera que a partir de éste se obtenga un nuevo archivo de datos con características que definiremos en base a los datos originales.

En particular, podemos hacer una manipulación matemática de los valores originales, junto con sus valores máximo y mínimo así como el número de dato (subíndice) combinándolos en funciones algebraicas que nosotros elijamos, para obtener resultados específicos.

Por ejemplo, podemos obtener a partir de un conjunto de datos originales las primeras o segundas diferencias de los valores de la función, el logaritmo natural de los datos originales, etc.

También es posible interpolar nuevos valores entre los valores originales, obtenidos por el método de Lagrange o por otros medios.

Otra aplicación posible es combinar dos archivos de datos para obtener su suma algebraica o cualquier otra función.

Segmento principal

Aquí se pone el modo de pantalla 1, se limpia la memoria y se dimensionan las variables para admitir hasta dos archivos con 100 datos cada uno.

Se elige entre trabajar con uno o dos archivos, para finalmente guardar los datos resultantes, pasando el control luego al programa de graficación.

PROCUodos

Aquí se nos pregunta si trabajaremos con uno o dos archivos. Se pide el nombre del primer archivo y se leen sus datos. Si se eligió trabajar con un solo archivo, se tiene la opción de interpolar

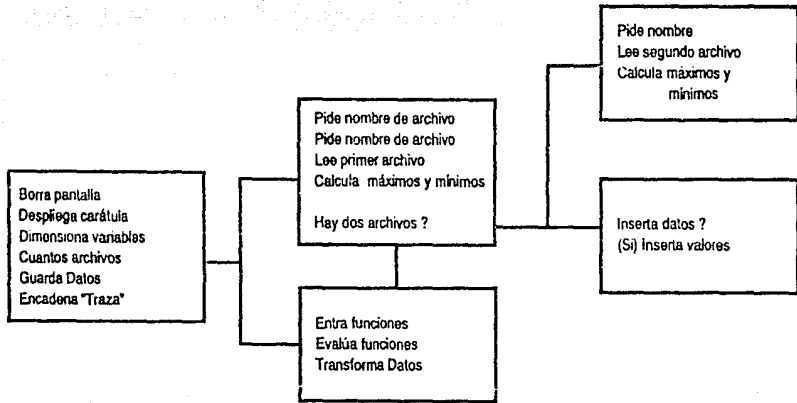


Fig. 11 Pseudocódigo del Programa Transito.

nuevos valores, de otra manera se procede a leer el nombre y los datos del segundo archivo. Es importante notar que de mezclarse dos archivos, deben tener el mismo número de datos.

Luego se piden las relaciones de transformación con que afectaremos el (los) archivos, para proceder con ellas sobre los datos originales.

PROCdos

Aquí se lee el nombre y los valores del segundo (si existe) archivo de datos.

PROCnom_arch(nf%)

Este procedimiento pide el nombre de archivo cuyos datos se van a leer, siendo nf% 0 ó 1 según sea el primer o segundo archivo que se lee. Se abre el archivo para entrada, se lee la descripción de los datos y las unidades de la variable independiente y de la función. Con el tamaño del archivo se calcula cuántos datos contiene.

PROCleedatos(nf%)

El número entero nf% es el número de archivo cuyos datos se van a leer, pudiendo valer 0 ó 1. Pone en ceros los valores iniciales, iniciando un ciclo de lectura de n% pares de datos, calculando para cada par los valores máximos y mínimos. Finalmente cierra el archivo.

PROCdar_fun

Una vez leídos los datos de el (los) archivos, se nos piden las relaciones de transformación para convertir los datos originales, con nf%=0 ó 1 según se hayan leído 1 ó 2 archivos. Estas funciones se darán usando las variables originales, y opcionalmente sus máximos y mínimos y el número de datos.

PROCtransf

Aquí se da el proceso de transformación de los datos originales a los nuevos datos, tras preguntar en qué subconjunto de datos se desea efectuar.

PROCguardatos

Aquí se pide el nuevo nombre de archivo para los datos modificados, junto con una nueva descripción y nuevas unidades, de ser aplicable. En ese momento se abre el archivo de salida, y se escriben en él esos datos para después escribir los pares de valores transformados. Al terminar se cierra el archivo.

PROCcalc_tmax(a)

Se calculan el valor mínimo y el máximo para la variable t.

PROCcalc_wmax(b)

Hace lo mismo para la variable w(t), es decir la función.

PROCinserta(nf%)

En el caso de haber seleccionado un solo archivo para modificar, se presenta la opción de insertar nuevos datos interpolados. De aceptarse esta opción, se pregunta tras de cuál dato se desea

interpolarse un valor, en cuyo caso se transfieren íntegramente los datos anteriores a él, se inserta el dato nuevo y se recorre la tabla. Se copia el resto de los datos originales, y se incrementa en uno el contador de datos. Se nos pregunta si deseamos repetir el proceso, en cuyo caso esta secuencia se realiza de nuevo.

FNu(t), PROCcarátula y PROCbip ya fueron descritos antes.

PROGRAMA Control

Este programa permite ejercer sobre la planta el Control Digital Directo, ya sea en lazo abierto o en lazo cerrado.

En lazo abierto el control se ejerce oprimiendo las teclas de flecha del cursor, lo cual hace variar el voltaje de entrada a la planta. El programa mide y despliega los valores equivalentes de voltaje de entrada y de salida, junto con el valor digital aplicado.

En el de lazo cerrado, se utiliza un algoritmo Proporcional + Integral + Derivativo. Esto se hace solicitando varios parámetros y aplicando a la planta los valores resultantes de dicho algoritmo controlador PID.

Segmento principal.

Se selecciona la opción de alto en caso de error, se pone el modo de pantalla 1 y se borra la memoria de datos. Se despliega la carátula y se inicia la planta y algunos parámetros de uso general, se leen los sensores de conversión del CAD.

Ahora se pide al operador seleccione entre lazo abierto y lazo cerrado, transfiriendo el control al procedimiento adecuado al oprimirse una tecla.

El proceso de control de lazo abierto regresa siempre al menú principal, mientras que el de lazo cerrado siempre lleva al programa "Traza".

PROClazabi

Control de lazo abierto. Despliega una nueva carátula, con una ventana especial para gráficos; al oprimir una tecla se ponen en la pantalla las marcas 0-255 y se inicia el ciclo de control, manipulado a través de las teclas de flecha del cursor. Si se oprime la tecla ROMPE se vuelve al menú principal.

PROClazcer

Cambia la carátula de la pantalla, calcula los valores mínimos y máximos aplicables a la planta, para luego pedir el valor deseado de velocidad y los valores de las constantes K_P , K_I y K_D para poder efectuar el control PID. Calcula los valores iniciales del algoritmo y entra de lleno al procedimiento de control digital al oprimir una tecla, tras lo cual se guardan los datos de dicho proceso, para pasar luego al segmento principal y de allí al programa de gráficos.

PROCCalcula

Se calculan como parámetros el valor de salida máximo esperado en el tacogenerador, el valor máximo de voltaje de entrada, y con ello los factores de conversión que correspondan, desplegándose todo ello en la pantalla.

Seudocódigo de control

Fig. 12 Pseudocódigo del Programa Control.

PROCpideK

Se pide el valor deseado de salida en el tacogenerador (setpoint) en el rango del valor mínimo esperado con la planta en acción, al valor máximo posible de acuerdo a PROCcalcula; se pide también el número de ciclos de muestreo durante los que se aplicará el algoritmo PID (máximo 100).

PROCvalini

Se asignan los dos primeros valores (0 y 1) de las variables a medir: tiempo, voltaje de salida en el tacogenerador, resultado del algoritmo controlador y valor digital a la salida del puerto paralelo.

PROCsetpoint

Aquí se fija el valor de referencia que se alimenta al controlador, pudiéndose variar el valor de la variable del punto de operación deseado, oprimiendo las teclas con flecha, pudiendo aumentar o disminuir el valor en 1 ó 5 veinteaos de volt de salida. El valor obtenido se limita a mayor que cero y menor que el valor máximo permisible (hex FF) según PROCcalcula. Su uso está reservado al Control de lazo cerrado de duración indefinida.

FNtecias

Utilizado en lazo abierto. Se definen las variables locales pica% y tecla%, la primera para almacenar el valor de la tecla oprimida, y la segunda para devolver el valor asignado de incremento al procedimiento que llamó a esta función.

De acuerdo a la flecha oprimida, se calcula un incremento (o decremento) de 1 ó 5 unidades. Si se oprime una tecla distinta de las flechas o no se oprime ninguna tecla, el valor devuelto es cero (sin cambio).

PROCbarra(u%)

Se traza la barra de despliegue de velocidad como un par de triángulos, con la instrucción PLOT85. La variable ancho% es el ancho de la barra definido en PROCinicia.

PROCoprime

Concita al operador a oprimir una tecla para continuar.

PROCmide

Es utilizado en el control de lazo abierto, inicia ciclos de espera para obtener los valores convertidos por el CAD, para luego multiplicarlos por sus respectivos factores de conversión para que queden en valores equivalentes de voltaje, tanto de salida como de entrada.

PROCmarabi

Borra la pantalla y pone las marcas adecuadas para la barra de despliegue de velocidad, así como los textos que aparecerán en la pantalla, junto con los voltajes de entrada y de salida, todo en lazo abierto. Incluye la definición del subprocedimiento PROCdisplay, que hace lo mismo, pero sin borrar la pantalla.

PROCdisplay

Redefine el área de textos al rectángulo de esquinas con coordenadas (24,15) y (29,10). Borra este segmento e imprime allí los valores de salida digital, voltaje de entrada y voltaje de salida.

PROCconman

Se inicia un ciclo sin término, en el cual se lee el valor de las teclas con flecha (FNteclas), el cual se suma al valor previo, para producir la nueva salida digital. Se miden y convierten los valores de voltaje de entrada y de salida, para luego ser desplegados con PROCdisplay.

PROCcondig

Este es el procedimiento mas importante del programa. Es el que efectivamente ejecuta el ciclo de control de lazo cerrado, que se inicia al oprimir una tecla.

Primero se asigna el valor del tiempo en cero. Luego se repite un lazo iniciando con un contador en 2 y hasta el valor de número de ciclos que se hayan dado. El inicio en 2 es debido a que el algoritmo utiliza el valor $w(1, i\% - 2)$, y el índice mínimo accesible siempre es 0.

Cada vez que reinicia el ciclo se envía el valor de salida $u(i\% - 1)$ al puerto paralelo, tras de lo cual se espera a que se termine una conversión en el canal 1, este valor adimensional medido por el CAD se multiplica por su respectivo valor de conversión, para equipararlo al valor real (físico) de salida. Asimismo se asigna el valor del tiempo en cada instante.

Se calcula el error en ese instante (Err), con lo cual se calculan las porciones proporcional, integral y derivativa del algoritmo (el cual es de tipo incremental o de velocidad), las cuales se suman al valor anterior del algoritmo para dar el nuevo valor de salida. Este es truncado y limitado al valor máximo admisible, convirtiéndose en el nuevo valor de la variable $u(i\%)$, el cual es mandado a la salida al reiniciarse el ciclo.

Se imprimen los valores finales de la variables del programa: punto deseado, valor final de salida y valor final de la acción de control, así como el tiempo de muestreo promediado para el ciclo.

Al terminar el ciclo de control la planta es apagada y desconectada.

PROCinicia

Se inician las condiciones para el funcionamiento de la planta: se conecta el relevador (interno) para encender la planta, se declaran dos canales de conversión y se selecciona el puerto paralelo, por otra parte, las teclas de flecha se declaran como teclas programadas (*FX4,1). También se dimensiona el arreglo para los valores de los sensores y se prepara la tecla ROMPE para transferir control al menú principal.

PROCventana

Genera una ventana donde se desplegarán los resultados, en el control de lazo abierto.

PROCsuma

Hace un sonido de frecuencia proporcional al voltaje aplicado a la planta, conserva el valor antiguo y determina el color con que ha de dibujar la barra de velocidad (verde si el incremento es

positivo y negro si es positivo). Calcula el nuevo valor de salida, limitándolo al máximo &FF (255). Mueve el cursor de gráficas a la nueva posición para dibujar la barra de velocidad con el procedimiento

PROCmarcas

Aquí se despliegan los textos informativos del control de lazo abierto.

PROCmuestra y PROCalgo no se utilizan, están reservados como ampliación futura para la aplicación del control de lazo cerrado en acción permanente.

PROCSaldig, PROCzara, PROCbip, PROCguardatos, PROCleesensor y PROCjoy ya fueron descritos antes.

PROGRAMA Calcula

En este programa se concentran los cálculos relacionados con resistencias, como son el cálculo de sensibilidad de un amplificador operacional, el cálculo de una ganancia o el cálculo de un paralelo de resistencias para obtener un valor deseado.

También se encuentra aquí el cálculo de los coeficientes de las funciones de transferencia de segundo y tercer órdenes dados los coeficientes y exponentes hallados antes. Por otra parte se incluyen las rutinas de cálculo de los coeficientes de un controlador PID por la regla de Ziegler-Nichols, tanto en forma continua como discreta.

Segmento principal.

Se pone la pantalla en el modo 1, se limpia la memoria y se despliega la carátula, se dimensionan los valores de resistencias (comerciales) y se declaran en una lista de datos.

Luego se despliega la lista de opciones, y se recibe el valor de una tecla, en función de la cual se hace una transferencia de control al procedimiento deseado. Al regresar aquí se insta al operador para oprimir una tecla, pudiendo seleccionar otra opción o volver al menú principal.

PROCsens

Aquí se calcula la sensibilidad (desviación del valor ideal) de un amplificador operacional en función de sus resistencias y tolerancias conocidas, así como los voltajes de entrada.

Se fija un formato de despliegue numérico con 3 decimales en un campo de 8 espacios, se borra la pantalla y se informa al operador de las premisas de cálculo tomadas ($A_v = R_2/R_1$, $e = 5\%$ y $V_{in} = 0$).

Recuérdese que los amplificadores de instrumentación están referidos a tierra y conectados como no-inversores.

Se piden los valores de las resistencias y de voltaje aplicado a la entrada no inversora del amplificador. Se calculan y despliegan el valor de la ganancia y voltaje de salida ideales. Se calculan las máximas desviaciones (positiva y negativa) del voltaje de salida, las cuales se muestran en valor absoluto y en porcentaje.

Seudocódigo de calcula

Fig. 13 Pseudocódigo del Programa Calcula.

PROCGana

Aquí se calcula una combinación de un par de resistencias comerciales para obtener una ganancia deseada en un amplificador operacional. La ganancia se establece entre 1 y 10, ya que es posible escalar la ganancia en factores de 10 elevados a una potencia, escalando en forma apropiada las resistencias que la determinan.

Se borra la pantalla y se leen los doce valores de resistencias comerciales, siendo igual la lista de valores de R_1 y R_2 . Se pide el valor de ganancia deseado y el máximo porcentaje admisible de error. Se despliega un encabezado de la tabla de valores que se producirán, se fija el formato de despliegue en tres decimales.

Ahora se inician dos ciclos anidados, que combinarán todos los pares posibles de resistencias comerciales, excepto cuando sean idénticas, se calcula la ganancia que producirían, escalándose al rango de 1 a 10, así como el porcentaje de desviación de la ganancia deseada. Si el valor absoluto de esta desviación es menor que el valor máximo fijado, los valores de las resistencias, ganancia y desviación son desplegados, de otro modo se continúa con el ciclo.

PROCPara

Aquí se calcula la combinación de un par de resistencias comerciales para producir un valor de equivalente deseado al conectarlas en paralelo.

Se limpia la pantalla, se recuperan (leen) los doce valores comerciales de resistencias, se pide el valor del paralelo de resistencias deseado y el máximo porcentaje de error permisible. Se muestra el encabezado de la lista de datos que serán calculados y se fija el formato de despliegue en tres decimales.

Se inician cuatro ciclos anidados: los dos más internos harán variar R_2 a través de sus doce valores comerciales, multiplicados desde por 1 hasta por 1000; los dos ciclos externos harán lo mismo con R_1 .

En cada caso se calculan las nuevas resistencias R_{n1} y R_{n2} , se multiplican y se suman para calcular su paralelo y el porcentaje de desviación del valor pedido, si el valor absoluto de éste es menor que el admisible, el par de resistencias se despliegan, de otro modo se continúa con los ciclos.

FNvalor

Esta función opcional se puede utilizar para calcular el valor de las resistencias afectadas por tolerancias "e", y luego los respectivos coeficientes para la ganancia del amplificador operacional, devolviendo el voltaje real de salida.

FNtecla

Esta función devuelve el valor de una tecla oprimida, la cual es limitada al rango indicado en la variable opcs\$.

PROCziegler

Aquí se aplicarán las reglas de Ziegler y Nichols al cálculo de los coeficientes de un controlador PID, ya sea en su versión continua o discreta.

Se fija el formato de despliegue en modo general, con tres decimales en un campo de 7 posiciones. Se borra la pantalla y se piden los valores de R y de L calculados a partir de la curva de reacción del sistema. Se pide el tipo de diseño que se desea (continuo o discreto), de ser la versión discreta se pedirá el tiempo de muestreo. En la versión continua este tiempo vale cero.

Se borra la pantalla y se despliegan los encabezados de los datos que serán calculados, para luego acceder a los procedimientos de cálculo propiamente dichos.

PROCcont

Aquí se llaman a los correspondientes procedimientos de cálculo de coeficientes en la versión continua, desplegando en cada caso los valores hallados.

PROCp, PROCpi, PROCpid

Cálculo de los coeficientes K_p , K_i , K_d del controlador PID en su versión continua.

PROCmuesT

Pide el valor del tiempo de muestreo para la versión discreta del controlador PID.

PROCdiscre

Aquí se llaman a los procedimientos de cálculo de coeficientes pero para la versión discreta, desplegando en cada caso los valores hallados. Ver PROCcont.

DEFPROCpD, PROCpiD, PROCpidD

Hacen los cálculos de los coeficientes K_p , K_i , K_d del controlador PID en su versión discreta, con las salvedades indicadas cuando el cálculo L/T tiene ciertos valores específicos (casi cero o cerca de un valor entero). {Alvarez, 7.1}.

PROCpinta(t\$)

Despliega los valores calculados de los coeficientes, en cada caso.

PROCvalcerseg

Aquí se calculan los coeficientes del numerador de la función de transferencia equivalente de segundo grado, y de allí la ganancia de la función y el valor del polo en el plano complejo s . Da también el error en condiciones iniciales.

PROCvalcerTer

Aquí se calcula el polinomio del numerador de la función de transferencia de tercer grado, dados los términos separados. Calcula la ganancia y los correspondientes polos en el plano complejo s , y luego los coeficientes para pasar a la función discreta equivalente $G(z)$, dado el tiempo de muestreo T .

PROCcuadra

Este procedimiento resuelve una ecuación cuadrática x^2+bx+c ($a=1$) con la fórmula general. Despliega las raíces resultantes, sean reales o complejas.

PROCcaratula y PROCbip ya fueron descritos antes.

RESUMEN

Se describió el equipo y los requisitos para poder diseñar y usar los programas para el control de una planta sencilla, de una sola entrada y una sola salida, a través de una microcomputadora de bajo costo; se describió el enfoque estructurado de desarrollo de software y su aplicación en este caso.

Se desarrollaron los bloques de pseudocódigo para la generación de los programas, así como algunos diagramas de bloques relacionados.

Se desarrollaron en total seis programas, escritos en el lenguaje estructurado de alto nivel BASIC de la BBC, los cuales se extienden en un total de 1,900 líneas de código, que ocupan un espacio total de casi 40 kB de memoria RAM, sin considerar el espacio requerido para los datos.

Estos programas abarcan los cuatro subsistemas descritos al inicio del capítulo: calibración, captura, análisis—transformación y control. En total se efectúan ocho funciones que se pueden llamar desde el menú principal.

Cada programa se describe en cierto detalle, anotando las funciones que realiza y los principales algoritmos que usa, en cada caso. También se presenta el listado completo de cada uno de ellos, justo antes de los apéndices.

Dada su generalidad, este conjunto de programas en BASIC sería fácilmente transportable o convertible a otras computadoras y plantas.

A continuación, se agregan los listados comentados de la totalidad de los programas.

§

LISTADO DEL PROGRAMA MENU

```

0010
0020 REM Menu Principal y otras
funciones
0030 REM revisado 130291 1000
0040
0050 ONERRORSTOP
0060 MODE1
0070 CLEAR
0080 PROCcara("Control Digital Directo
= Menú")
0090 PROCmenu
0100 END
0110

0120 DEFPROCmenu
0130 LOCALopc%
0140 PROCprogs
0150 REPEAT
0160 *FX15
0170 PROCbip
0180 opc%=GET-48
0190 UNTILOpc%>0 AND opc%<9
0200 PRINTTAB(0,0);"Espere un
momento por favor...";
0210 ONopc% GOTO 220, 230,
240, 250, 260, 270, 280, 290, 300
0220 PROCcat
0230 PROCismot
0240 PROCcalibra
0250 CHAIN"Captura"
0260 CHAIN"Traza"
0270 CHAIN"Transfo"
0280 CHAIN"Calcula"
0290 CHAIN"Control"
0300 RUN
0310 ENDPROC
0320

0330 DEFPROCprogs
0340 LOCALi%
0350 @%=&A
0360 PRINTSPC3;"Opciones de
operación:"
0370 PRINT"1: Catálogo de archivos"
0380 PRINT"2: Calentamiento de motor"
0390 PRINT"3: Calibración de
sensores"
0400 PRINT"4: Entrada de datos"
0410 PRINT"5: Gráfica, Tabla, Ajuste,
Interpola"
0420 PRINT"6: Transformación de datos"
0430 PRINT"7: Cálculos varios"
0440 PRINT"8: Control Digital"
0450 COLOR1
0460 FORi%=2 TO 16 STEP 2
0470 PRINTTAB(0,i%);i%DIV2;
0480 NEXT
0490 VDU28,0,24,39,24
0500 CLS
0510 COLOR2
0520 PRINTTAB(2,0);"Escoja (";
0530 COLOR1
0540 PRINT"1-8";
0550 COLOR2
0560 PRINT") ";
0570 ENDPROC
0580

0590 DEFPROCcat
0600 VDU26,12
0610 PROCcara
("Catalogo de Archivos")
0620 PROCbip
0630 *CAT
0640 PROCfin
0650 ENDPROC
0660

0670 DEFPROCcara(txt$)
0680 LOCALi%,z%,d%
0690 COLOR2
0700 VDU19,1,2,0;0;0;
0710 VDU19,2,1,0;0;0;
0720 PRINT "+";STRING$(38,"-")
;"+";"";
0730 z%=(38-LENtxt$)DIV2

```

```

0740 d%=37-LENtxt$-2%
0750 COLOR1
0760 PRINTSPCz%:txt$;SPCd%;
0770 COLOR2
0780 PRINT " ";";STRINGS$(38,"-");";
0790 FORi%=0TO19
0800 PRINTTAB(39,3+i%
;";TAB(0,3+i%);";
0810 NEXT
0820 PRINT"+";STRINGS$(38,"-");
"+";TAB(3,1);
0830 COLOR3
0840 VDU28,2,22,38,3
0850 *KEY10*E.!BOOT|M
0860 ENDPROC
0870

0880 DEFPROCcalibra
0890 LOCALj%
0900 VDU26,12
0910 PROCcra("Calibración de
sensores")
0920 PROCprepara
0930 TIME=0
0940 FORj%=1 TO 4
0950 CLS
0960 PROCmide(j%)
0970 NEXT
0980 CLS
0990 PRINT"Motor apagado"
1000 PROCsaldig(0)
1010 *MOTOR 0
1020 PROCguarsensor
1030 OSCLI("Acceso D.Sensor L")
1040 PROCfin
1050 ENDPROC
1060

1070 DEFPROCprepara
1080 DIMsensor(4),valor(100)
1090 PROCinicia
1100 @%=&40A
1110 PRINT"Oprima una tecla para
comenzar "
1120 PROCbip

```

```

1130 REPEAT
1140 UNTILGET
1150 PROCsaldig(&B0)
1160 PRINT"Motor encendido"
1170 PRINT"Oprime ESPACIO para
continuar"
1180 PROCbip
1190 REPEAT
1200 UNTILGET
1210 ENDPROC
1220

1230 DEFPROCmide(i%)
1240 LOCALvalmin,valmax,
valmed,k%
1250 LOCALSUMval,valobs,dif,pct
1260 PROCinicia
1270 PRINT"Sensor número ";i%"
1280 valmin=&FFFF
1290 valmax=0
1300 SUMval=0
1310 FORk%=1 TO 100
1320 REPEAT
1330 UNTILADVAL(0)
DIV256=i%
1340 valor(k%)=ADVAL(i%)
1350 PROCcalcmax(valor(k%))
1360 SUMval=SUMval+valor(k%)
1370 valmed=SUMval/k%
1380 PRINTTAB(0,VPOS);"Valor
medio: ";valmed;
1390 NEXT
1400 PRINT"Valor mínimo: ";valmin
1410 PRINT"Valor máximo: ";valmax
1420 dif=valmax-valmin
1430 pct=dif/valmin*100
1440 PRINT"diferencia: ";dif;"
( ";pct;"% )"
1450 PROCbip
1460 INPUT"Valor observado: ";
valobs
1470 sensor(i%)=valobs/valmed
1480 ENDPROC
1490

```

```

1500 DEFPROCcalcmax(a)
1510 IFvalmin<a valmin=a
1520 IFvalmax>a valmax=a
1530 ENDPROC
1540

1550 DEFPROCguarsensor
1560 LOCALarchnum%,i%,fe$
1570 INPUT"La fecha por favor "fe$
1580 fe$="Factores de Conversión,
    "+fe$
1590 PRINT"Guardando factores"
1600 archnum%=OPENOUT
    ("D.Sensor")
1610 PRINT#archnum%,fe$,"n","*"
1620 FORi%=1 TO 4
1630     n=i%
1640     PRINT"Factor(",i%;")=";
        sensor(i%)
1650     PRINT#archnum%,n,
        sensor(i%)
1660     NEXT
1670 CLOSE#archnum%
1680 ENDPROC
1690

1700 DEFPROCfin
1710 PRINT"Oprime ESPACIO para
    ir al menú";
1720 PROCbip
1730 REPEAT
1740     UNTILGET
1750 RUN
1760 ENDPROC
1770

1780 DEFPROCbip
1790 SOUND1,-13,80,1
1800 SOUND1,-15,200,1
1810 ENDPROC
1820

```

```

1830 DEFPROCscaldig(val%)
1840 VDU2
1850 VDU1,val%
1860 VDU3
1870 ENDPROC
1880

1890 DEFPROCtstmot
1900 LOCALi%
1910 VDU26,12
1920 PROCcara("Calentamiento de
    motor")
1930 PROCinicia
1940 PRINT"Oprime una tecla para
    comenzar"
1950 REPEAT
1960     UNTILGET
1970 CLS
1980 PRINT"Fase 1: Ciclo
    creciente";
1990 m%=2500+RND(500)
2000 TIME=0
2010 FORi%=1 TO m% STEP 2
2020     PROCTST(i%,m%)
2030     NEXT
2040 PRINT;TIME DIV100;" seg."
2050 PRINT"Fase 2: Ciclo
    decreciente ";
2060 TIME=0
2070 FORi%=m% TO 1 STEP -2
2080     PROCTST(i%,m%)
2090     NEXT
2100 PRINT;TIME DIV100;" seg."
2110 PRINT"Fase 3: Valores al
    azar";
2120 TIME=0
2130 PROCazar(3500)
2140 PRINT;TIME DIV100;" seg."
2150 PRINT"Fase 4: Valor máximo"
2160 PRINT"Oprime ROMPE para
    ir al menú"

```

```

2170 REPEAT
2180   PROCsaldig(&FF)
2190   UNTILO
2200 RUN
2210 ENDPROC
2220

2230 DEFPROCInicia
2240 *FX16,4
2250 *FX5,1
2260 *FX6
2270 *MOTOR 1
2280 ENDPROC
2290

2300 DEFPROCTST(i%,m%)
2310 LOCALj%
2320 PROCsaldig(&FF)
2330 FORj%=1 TO m%-i%*20
2340   NEXT
2350 PROCsaldig(0)
2360 FORj%=1 TO m%-i%*20
2370   NEXT
2380 ENDPROC
2390

2400 DEFPROCCazar(m%)
2410 LOCALi%,num%
2420 FORi%=1 TO m%
2430   PROCsaldig(0)
2440   num%=55+RND(200)

```

```

2450   PROCsaldig(num%)
2460   NEXT
2470 ENDPROC
2480

2490 DEFPROCJoyent
2500 REM Contador en el joystick
2510 REM conteo cada .49 seg
2520
2530 ant%=0
2540 REPEAT
2550   cnt%=0
2560   TIME=0
2570   REPEAT
2580     REPEAT
2590       ant%=ADVAL(0)
2600       AND 1
2610       UNTILant%=1 OR
2620       TIME=49
2630       cnt%=cnt%+1
2640       REPEAT
2650         ant%=ADVAL(0)
2660         AND 1
2670         UNTILant%=0
2680         UNTILTIME=49
2690         PRINTTAB(2,2)cnt%
2700         REPEAT
2710           UNTILTIME=100
2720           UNTILO
2730           ENDPROC

```

LISTADO DEL PROGRAMA CAPTUR A

```

0010
0020 REM Captura datos
0030 REM Revisada 130291 1000
0040
0050 ONERRORSTOP
0060 MODE1
0070 CLEAR
0080 PROCinicia
0090 PROCcara("Entrada de datos")
0100 PROCcant
0110 PROCfuente
0120 PROCbip
0130 PROCprompt("terminar")
0140 CHAIN"Traza"
0150 END
0160

0170 DEFPROCcant
0180 REPEAT
0190   PROCbip
0200   PRINTTAB(0,1)"¿Cuántos
      datos"
0210   INPUTTAB(0,2)
      "almacenará? "n%"
0220   UNTILn%>2 AND n%<150
0230   DIMw(4,n%),sensor(4),val(n%)
0240 ENDPROC
0250

0260 DEFPROCfuente
0270 LOCALorg$
0280 PRINT"Función:""1 Experimental"
0290 PRINT"2 Tabla""3 Analítica"
0300 PRINT"Seleccione un número:";
0310 PROCbip
0320 org$=FNkey("123")
0330 IForg$="1" PROCexper
0340 IForg$="2" PROCTabla
0350 IForg$="3" PROCAnalit
0360 ENDPROC
0370

0380 DEFFNkey(opcs$)
0390 LOCALkey$
0400 REPEAT
0410   key$=GET$
0420   UNTILINSTR(opcs$,key$)<>0
0430   =key$
0440

0450 DEFPROCAnalit
0460 LOCALw$,t0,t1,dt,t,i%
0470 CLS
0480 PROCbip
0490 PRINT"Ecuación a tabular"
0500 INPUT"en función de t"
      ""w(t)= "w$
0510 INPUT"desde t= "t0
0520 INPUT"hasta t= "t1
0530 dt=(t1-t0)/n%
0540 t=t0
0550 CLS
0560 PRINT"Calculando función"
0570 FOR i%=1 TO n%
0580   w(0,i%)=t
0590   w(1,i%)=EVALw$
0600   t=t+dt
0610   PLOT69,64+t/t1*1100,616
0620   NEXT
0630 PROCguardatos(n%,0,1)
0640 ENDPROC
0650

0660 DEFPROCTabla
0670 LOCALi%
0680 CLS
0690 PRINT""Digite los valores:"
0700 PROCbip
0710 FOR i%=1 TO n%
0720   PRINT""(:"i%;")="";
0730   INPUT""w(0,i%)
0740   PRINTTAB(17,VPOS-1);"
      w("i%;")="";
0750   INPUT""w(1,i%)
0760   NEXT

```

```

0770 PROCguardatos(n%,0,1)
0780 ENDPROC
0790

0800 DEFPROCexper
0810 PROCentra
0820 PROClecsensor
0830 IFca%=2 PROCtransforma
(n%,ca%)
0840 PROCtransforma(n%,cb%)
0850 PROCguardatos(n%,ca%,cb%)
0860 ENDPROC
0870

0880 DEFPROCentra
0890 CLS
0900 PRINT"Qué función de
entrada:"
0910 PRINT"0 Impulso"
0920 PRINT"1 Escalón"
0930 PRINT"2 Rampa"
0940 PRINT"3 Mediciones
paramétricas"
0950 PRINT"Entrada: ";
0960 fun$=FNkey("0123")
0970 IFfun$="0" fun$="imp"
0980 IFfun$="1" fun$="esc"
0990 IFfun$="2" fun$="ram"
1000 IFfun$="3" fun$="par"
1010 CLS
1020 IFfun$="par" PROCcanal ELSE
PROCfza(fun$)
1030 ENDPROC
1040

1050 DEFPROCcanal
1060 PRINT"Seleccione la variable de
entrada"
1070 PRINT"0 Valor digital 0-255"
1080 PRINT"2 Voltaje de armadura"
1090 ca%=VALFNkey("02")
1100 PRINT"Seleccione la salida:"
1110 PRINT"1 Voltaje del tacómetro"
1120 PRINT"2 Contador de pulsos"
1130 PRINT"3 Corriente de entrada"

```

```

1140 cb%=VALFNkey("123")
1150 PROCmide(n%)
1160 ENDPROC
1170

1180 DEFPROCmide(n%)
1190 LOCALi%.arg%
1200 *FX16,3
1210 PROCaviso
1220 FOR i%=1 TO n%
1230 arg%=i%*255 DIV n%
1240 PROCsaldig(arg%)
1250 PLOT69,64+i%/n%*1100,560
1260 PROCespera(500)
1270 IFca%=0 w(ca%,i%)=arg%
1280 IFca%=2 w(ca%,i%)=
FNconvierte(20,ca%)
1290 IFcb%=1 w(cb%,i%)=
FNconvierte(20,cb%)
1300 IFcb%=2 w(cb%,i%)=
FNjoyent
1310 IFcb%=3 w(cb%,i%)=
FNconvierte(20,cb%)
1320 NEXT
1330 IFcb%=3 PROCcalcula
1340 PROCsaldig(0)
1350 ENDPROC
1360

1370 DEFPROCespera(j%)
1380 LOCALi%
1390 FOR i%=1 TO j%
1400 NEXT
1410 ENDPROC
1420

1430 DEFFNconvierte(veces%,chan%)
1440 LOCALk%,suma%
1450 suma%=0
1460 FOR k%=1 TO veces%
1470 REPEAT
1480 UNTILADVAL(0)
DIV256=chan%
1490 suma%=suma%+
ADVAL(chan%)

```

```

1500 NEXT
1510 =suma% DIV veces%
1520

1530 DEFPROCcalculaI
1540 LOCALi%,dif
1550 CLS
1560 PRINT"Calculando corriente"
1570 FOR i%=1 TO n%
1580 dif=w(ca%,i%)-w(cb%,i%)
1590 w(cb%,i%)=dif/3.3 :REM
ohms
1600 PLOT69,64+i%/n%*1100,304
1610 NEXT
1620 ENDPROC
1630

1640 DEFPROCfza(tipo$)
1650 REM asign canales
1660 ca%=0
1670 cb%=1
1680 PROClimites
1690 IFfun$="imp" PROCimp
1700 IFfun$="esc" PROCesc
1710 IFfun$="ram" PROCram
1720 PROCaplica
1730 ENDPROC
1740

1750 DEFPROClimites
1760 LOCALa%,b%
1770 PRINT"En porcentaje
(0-100) dar"
1780 INPUT"Valor inicial= "a%
1790 INPUT"Valor final = "b%
1800 wa%=255*a% DIV 100
1810 wb%=255*b% DIV 100
1820 mag%=wb%-wa%
1830 PROCsaldig(wa%)
1840 ENDPROC
1850

```

```

1860 DEFPROCimp
1870 PRINT"Impulso de voltaje"
1880 val(0)=wb%
1890 val(1)=wb%
1900 val(2)=0
1910 inc=0
1920 PROCperfil(3)
1930 ENDPROC
1940

```

```

1950 DEFPROCesc
1960 PRINT"Escalón de voltaje"
1970 val(0)=wb%
1980 inc=0
1990 PROCperfil(1)
2000 ENDPROC
2010

```

```

2020 DEFPROCram
2030 PRINT"Rampa de voltaje"
2040 val(0)=wa%
2050 inc=mag%/n%
2060 PROCperfil(1)
2070 ENDPROC
2080

```

```

2090 DEFPROCperfil(j%)
2100 LOCALi%
2110 FORi%=j% TO n%
2120 val(i%)=val(i%-1)+inc
2130 NEXT
2140 ENDPROC
2150

```

```

2160 DEFPROCaplica
2170 LOCALi%
2180 REM establece 1 canal
2190 *FX16,1
2200 PROCprompt("aplicar función")
2210 PROCaviso
2220 REM Fuerza convers canl 1
2230 *FX17,1
2240 REPEAT
2250 TIME=0

```



```

2260 UNTILADVAL(0)DIV256=1
2270 FOR i%=1 TO n%
2280 PROCsaldig(val(i%))
2290 REPEAT
2300 UNTILADVAL(0)
      DIV256=1
2310 w(0,i%)=TIME
2320 w(1,i%)=ADVAL(1)
2330 NEXT
2340 PROCsaldig(0)
2350 ENDPROC
2360

2370 DEFPROCuviso
2380 CLS
2390 PRINT"Por favor espere"
2400 PRINT"Capturando datos"
2410 ENDPROC
2420

2430 DEFPROCtransforma(n%,j%)
2440 LOCALi%
2450 PRINT"Transformando datos"
2460 FOR i%=1 TO n%
2470 w(j%,i%)=w(j%,i%)*
      sensor(j%)
2480 NEXT
2490 ENDPROC
2500

2510 DEFPROCguardados(n%,j%,k%)
2520 LOCALArch$,tit$,ut$,uw$,
      archnum%,i%
2530 PROCbip
2540 PRINT"Nombre para"
2550 INPUT"este archivo: "arch$
2560 INPUT"Descripción de los
      datos:"tit$
2570 INPUT"Unidades de t:"ut$
2580 INPUT"Unidades de w(t):"uw$
2590 PRINT"Guardando datos"
2600 archnum%=OPENOUT(arch$)
2610 PRINT#archnum%,tit$,ut$,uw$
2620 FOR i%=1 TO n%

```

```

2630 PRINT#archnum%,
      w(j%,i%),w(k%,j%)
2640 NEXT
2650 CLOSE#archnum%
2660 ENDPROC
2670

2680 DEFPROCteensor
2690 LOCALArchnum%,i%,tit$,x$
2700 PRINT"Leyendo factores"
2710 sensor(0)=1
2720 archnum%=OPENIN
      ("D.Sensor")
2730 INPUT#archnum%,tit$,x$,x$
2740 FOR i%=1 TO 4
2750 INPUT#archnum%,n,
      sensor(i%)
2760 NEXT
2770 CLOSE#archnum%
2780 ENDPROC
2790

2800 DEFPROCbip
2810 SOUND1,-13,80,1
2820 SOUND1,-15,200,1
2830 ENDPROC
2840

2850 DEFPROCsaldig(byte%)
2860 REM salida pto paralelo
2870 VDU2
2880 VDU1,byte%
2890 VDU3
2900 ENDPROC
2910

2920 DEFNFjoyent
2930 REM Contador en el joystick
2940 REM max 120 Hz=7200
      seg
2950 ant%=0
2960 cnt%=0
2970 t=TIME
2980 REPEAT

```

```

2990 REPEAT
3000   ant%=ADVAL(0) AND 1
3010   UNTILant%=1 OR
      TIME-t=49
3020   cnt%=cnt%+1
3030 REPEAT
3040   ant%=ADVAL(0) AND 1
3050   UNTILant%=0
      OR TIME-t=49
3060   UNTILTIME-t=49
3070   =cnt%
3080

```

```

3090 DEFPROCincia
3100 *FX16,1
3110 *FX5,1
3120 *FX6
3130 *MOTOR 1
3140 *KEY10*FX3|MCLO.#0|M
      CH."!"|M
3150 PROCsaldig(0)
3160 ENDPROC
3170

```

```

3180 DEFPROCprompt(t$)
3190 PRINT"Oprima una tecla
      para ";t$
3200 PROCbip
3210 REPEAT
3220   UNTILGET
3230 ENDPROC
3240

```

```

3250 DEFPROCcara(txt$)
3260 LOCALi%,z%,d%
3270 COLOR2
3280 VDU19,1,2,0;0;0;
3290 VDU19,2,1,0;0;0;
3300 PRINT "+";STRING$(38,"-");"+";"";
3310 z%=(38-LEN(txt$))DIV2
3320 d%=(37-LEN(txt$)-z%
3330 COLOR1
3340 PRINTSPCz%;txt$;SPCd%;
3350 COLOR2
3360 PRINT" ";"";STRING$(38,"-");"";
3370 FORi%=0TO19
3380   PRINTTAB(39,3+i%);"";
      TAB(0,3+i%);"";
3390 NEXT
3400 PRINT"+";
      STRING$(38,"-");"+";TAB(3,1);
3410 COLOR3
3420 VDU28,2,22,38,3
3430 @%=&307
3440 *KEY10*E.!BOOT|M
3450 ENDPROC
3460

```

```

3470 DEFFNu(t)
3480 REM escalon unitario
3490 LOCALu
3500 IFt<=0 u=0 ELSE u=1
3510 =u

```

LISTADO DEL PROGRAMA TRAZA.

```

0010
0020 REM Grafica ajusta tabla etc
0030 REM Revisado 090191 1630
0040
0050 MODE1
0060 CLEAR
0070 DIMt(200),w(200),C(100)
0080 PROCinicia
0090 PROCcara
0100 PROCnombre
0110 PROCleedat(num%)
0120 PROCmaxmin(num%)
0130 PROCciclo
0140 PROCfinal
0150 END
0160

0170 DEFPROCnombre
0180 LOCALong%
0190 PROCbip
0200 PRINT"Nombre del"
0210 INPUT"archivo de datos "arch$
0220 archno%=OPENIN(arch$)
0230 INPUT#archno%,tit$,ut$,uw$
0240 PRINT"tit$"uw$;" vs ";"ut$
0250 long%=LENTit$+LENut$+
    LENuw$+6
0260 num%=(EXT#archno%-long%)
    DIV12
0270 PRINT"El archivo tiene ";"num%";
    " datos"
0280 ENDPROC
0290

```

```

0300 DEFPROCleedat(num%)
0310 LOCALi%
0320 PRINT"Leyendo ";"num%";
    " datos"
0330 FORi%=1 TO num%
0340 INPUT#archno%,t(i%),w(i%)
0350 NEXT
0360 CLOSE#archno%
0370 ENDPROC
0380

0390 DEFPROCmaxmin(num%)
0400 LOCALi%
0410 PRINT"Calculando máximos y
    mínimos"
0420 torg=0
0430 worg=0
0440 tmin=t(1)
0450 tmax=tmin
0460 wmin=w(1)
0470 wmax=wmin
0480 FORi%=1 TO num%
0490 PROCcaltmax(t(i%))
0500 PROCcaltwmax(w(i%))
0510 NEXT
0520 PROCcaltfac
0530 ENDPROC
0540

0550 DEFPROCcaltmax(a)
0560 IFtmin>a tmin=a
0570 IFtmax<a tmax=a
0580 ENDPROC
0590

0600 DEFPROCcaltwmax(b)
0610 IFwmin>b wmin=b
0620 IFwmax<b wmax=b
0630 ENDPROC
0640

```

0650 DEFPROCalfac
 0660 LOCALdif,t,difw
 0670 IFtmin<=0 torg=tmin
 0680 dift=imax-worg
 0690 IFwmin<=0 worg=wmin
 0700 difw=wmax-worg
 0710 REM 1280-2*160=960,
 800-160*2=480, 95%
 0720 IFdift<>0 ifac=912/dift ELSE
 tfac=1
 0730 IFdifw<>0 wfac=456/difw ELSE
 wfac=1
 0740 ENDPROC
 0750

 0760 DEFPROCciclo
 0770 LOCALf\$
 0780 PROCTeclea
 0790 PROCopc
 0800 nuco1%=1
 0810 REPEAT
 0820 PRINTTAB(0,20);"opción: ";
 0830 PROCBip
 0840 f\$=CHR\$(GET AND &5F)
 0850 VDU12,28,2,23,38,22,15
 0860 IFS="A" PROCnucA
 0870 IFS="G"
 CLG:PROCTraza(69,num%,1)
 088 0 IFS="H"
 CLG:PROCTraza(21,num%,1)
 0890 IFS="J"
 CLG:PROCTraza(5,num%,1)
 0900 IFS="D" PROCdife
 0910 IFS="I" PROCinterpola
 0920 IFS="L" PROCloga
 0930 IFS="M" PROCajusta
 0940 IFS="O" PROCmasgraf(69)
 0950 IFS="P" PROCmasgraf(21)
 0960 IFS="Q" PROCmasgraf(5)
 0970 IFS="R" RUN
 0980 IFS="T" PROCTabla(num%)
 0990 IFS="Z" PROCziegler
 1000 IFS=CHR\$31 OR f\$=
 CHR\$15 PROCopc
 1010 UNTILf\$="F"

1020 ENDPROC
 1030

 1040 DEFPROCmasgraf(tip%)
 1050 nuco1%=nuco1%+1
 1060 PROCnombre
 1070 PROCTcedat(num%)
 1080 PROCTraza
 (tip%,num%,nuco1%)
 1090 ENDPROC
 1100
 1110 DEFPROCTraza
 (tip%,num%,color%)
 1120 LOCALi%
 1130 VDU28,2,3,38,3,12
 1140 PRINTiii\$;
 1150 VDU28,2,23,38,22,12
 1160 PROCFijorg
 1170 PROCDibeje
 (torg*tfac,0,imax*tfac,0)
 1180 PROCDibeje
 (0,worg*wfac,0,wmax*wfac)
 1190 GCOL0,color%
 1200 MOVEt(1)*tfac,w(1)*wfac
 1210 FORi%=1 TO num%
 1220 PLOTtip%,t(i%)*ifac,
 w(i%)*wfac
 1230 NEXT
 1240 ENDPROC
 1250

 1260 DEFPROCFijorg
 1270 LOCALxorg%,yorg%
 1280 xorg%=160+ABS(torg)*tfac
 1290 yorg%=160+ABS(worg)*wfac
 1300 VDU29,xorg%,yorg%;
 1310 MOVE0,0
 1320 ENDPROC
 1330

 1340 DEFPROCDibeje(a,b,c,d)
 1350 GCOL0,3
 1360 MOVEa,b

```

1370 PLOT21,c,d
1380 MOVE0,0
1390 ENDPROC
1400

1410 DEFPROCtabla(num%)
1420 LOCALi%
1430 VDU28,2,23,38,3,14,12
1440 @i%=&30407
1450 PRINT"¿Impresora o Pantalla
(I/P)?"
1460 i$=GET$
1470 IFi$="I" VDU15,2 ELSE VDU12
1480 PRINT"Archivo de datos: ";
arch$
1490 PRINT"tit$
1500 PRINT"Listando ";num%;" datos"
1510 PRINT"torg=";torg," tmin=";tmin,"
tmax=";tmax
1520 PRINT"worq=";worq,"
wmin=";wmin," wmax=";wmax
1530 PRINT" i"SPCG"i";SPC8;"w,"
SPCG"wmmax"
1540 PRINTSPC8;ut$;SPC8;uw$
1550 IFwmax=0wmax=1E-4
1560 FORi%=1 TO num%
1570 @i%=&3
1580 PRINTi%;
1590 @i%=&2030A
1600 PRINTi(i%);
1610 PRINTw(i%);
1620 @i%=&2030A
1630 PRINTw(i%)/wmax*100
1640 NEXT
1650 PRINT"Fin de los datos"
1660 PRINT"Oprime ESPACIO"
1670 REPEAT
1680 UNTILGET
1690 VDU12,3
1700 ENDPROC
1710

```

```

1720 DEFPROCajusta
1730 LOCALi%,j%,k%,n%,St,Stt,
Sw,Sww,Sww,tden,p
1740 LOCALwden,vars,desstdt,
varw,desstdw,corr
1750 VDU28,2,23,38,22,15
1760 @i%=&30407
1770 Stt=0
1780 St=0
1790 Sw=0
1800 Sww=0
1810 Stw=0
1820 PRINT"
1830 REPEAT
1840 INPUTTAB(1,19);"De j%="
"j%
1850 INPUTTAB(1,19);" a k%="
"k%
1860 n%=k%-j%+1
1870 IFn%<2 PRINT
"2<= n% <=";num%;
1880 UNTILn%>=2 AND n%<=num%
1890 PRINT"Esperc.."
1900 FORi%=j% TO k%
1910 St =Stt+(i%)
1920 Stt=Stt+(i%)*i%
1930 Sw =Sw+w(i%)
1940 Sww=Sww+w(i%)*w(i%)
1950 Stw=Stw+(i%)*w(i%)
1960 NEXT
1970 medt=St/n%
1980 vars=(Stt/n%-medt*medt)
1990 desstdt=SQRvars
2000 medw=Sw/n%
2010 varw=(Sww/n%-medw*medw)
2020 desstdw=SQRvarw
2030 tden=n%*Stt-St*St
2040 wden=n%*Sww-Sw*Sw
2050 b=(Sw*Stt-St*Stw)/tden
2060 twnum=(n%*Stw-St*Sw)
2070 m=twnum/tden
2080 p=twnum/wden
2090 corr=100*twnum/SQR
(tden*wden)
2100 PROCraya

```

```

2110 PROCdata
2120 ENDPROC
2130

2140 DEFPROCzlegler
2150 LOCALT,i%
2160 PRINT"R=";m;" L=";-b/m;" ";
2170 PRINT"T=";wmax/m+b/m;"
      A=";wmax+b;
2180 PRINT" Tecla A para afinar";
2190 a$=CHR$(GET AND &5F)
2200 PRINT
2210 IFa$<>"A" ENDPROC
2220 INPUT"nueva T="T
2230 INPUT"punto i%"i%
2240 m=wmax/T
2250 L=1(i%)-w(i%)/m
2260 b=-m*L
2270 PRINT"R=";m;" L=";L;" ";
2280 PRINT"T=";T;" A=";wmax
2290 PROCraya
2300 ENDPROC
2310

2320 DEFPROCraya
2330 GCOL1,2
2340 MOVE0,b*wfac
2350 PLOT21,tmax*tfac,wfac*
      (m*tmax+b)
2360 ENDPROC
2370

2380 DEFPROCdata
2390 @%=&3
2400 PRINTTAB(0,20);"j=";j%;"
      k=";k%;
2410 @%=&30407
2420 PRINT" m=";m;" b=";b;" r=
      ";corr;" ";
2430 ENDPROC
2440

```

```

2450 DEFPROCdife
2460 LOCALh,detr,r1,r2
2470 INPUT"valor de h="h
2480 detr=SQR(m*m+4*b)
2490 r2=LN((m-detr)/2)/h
2500 r1=LN((m+detr)/2)/h
2510 PRINTTAB(0,20);"T=";-1/r1;" T=";
2520 PRINT-1/r2;" 1/T2--1/T1=";r2-r1;" ";
2530 ENDPROC
2540

2550 DEFPROCloga
2560 PRINTTAB(0,20);"A="
      ";EXP(b);
2570 PRINT" T=";-1/m;" T=";-m;" ";
2580 ENDPROC
2590

2600 DEFPROCnueA
2610 LOCALna
2620 INPUTTAB(1,19);"Nuevo
      coeficiente A="na
2630 b=LNna
2640 m=(medw-b)/medt
2650 PRINTTAB(0,20);"A=";na;
2660 PRINT" T=";-1/m;" T=";-m;" ";
2670 PROCraya
2680 ENDPROC
2690

2700 DEFPROCinterpolo
2710 LOCALx,z,j%,k%
2720 INPUTTAB(1,19);"De j%="j%
2730 INPUTTAB(1,19);" a k%="k%
2740 PROCcoefLag(j%,k%)
2750 @%=&30407
2760 INPUT"l="x
2770 z=FNlagrange(x,j%,k%)
2780 PRINT"w(x;")="z;" ";
2790 GCOL0,2
2800 PLOT69,x*tfac,z*wfac
2810 ENDPROC
2820

```

```

2830 DEFFNmulti(m%,x,j%,k%)
2840 LOCALi%,factor,prod
2850 prod=1
2860 FORi%=j% TO k%
2870   factor=x-1(i%)
2880   IFm%=i% factor=1
2890   prod=prod*factor
2900   NEXT
2910 =prod
2920

2930 DEFPROCcoefLag(j%,k%)
2940 LOCALi%
2950 REM
2960 PRINT"Calculando ";
2970 PRINTk%-j%+1;" coeficientes"
2980 FORi%=j% TO k%
2990   C(i%)=w(i%)/FNmulti
      (i%,i(i%),j%,k%)
3000   NEXT
3010 ENDPROC
3020

3030 DEFFNlagrange(x,j%,k%)
3040 LOCALi%,sum
3050 PRINT"Interpolando"
3060 sum=0
3070 FORi%=j% TO k%
3080   sum=sum+C(i%)*FNmulti
      (i%,x,j%,k%)
3090   NEXT
3100 =sum
3110

3120 DEFPROCopc
3130 VDU28,2,23,38,3,14
3140 CLS
3150 PRINT"Funciones:"
3160 PRINT"? lista funciones"
3170 PRINT"G H J gráfica original"
3180 PRINT"O P Q gráfica
      superpuesta"
3190 PRINT"Tabla de datos"
3200 PRINT"Interpolación de
      Lagrange"

```

```

3210 PRINT"Mínimos cuadrados (ajuste)"
3220 PRINT"A rejusta coeficiente A"
3230 PRINT"Logarítmico (método)"
3240 PRINT"Diferencias: método de
      Lipka"
3250 PRINT"Reinicia programa"
3260 PRINT"Ziegler Nichols""Fin"
3270 COLOR1
3280 PRINTTAB(POS,VPOS-16);"??"
3290 PRINT"G""O""T""I""M"
3300 PRINT"A""L""D""R""Z""F"
3310 COLOR3
3320 ENDPROC
3330

3340 DEFPROCcara
3350 LOCALi%
3360 VDU19,1,2,0;0;0;
3370 VDU19,2,1,0;0;0;
3380 COLOR2
3390 PRINTTAB(0,24);"+";
      STRING$(38,"-");"+";TAB(0,0);
3400 VDU11
3410 PRINT"+";STRING$(
      38,"-");"+";"";
3420 COLOR1
3430 PRINTSPC9;"Gráfica, Tabla,
      etc.";SPC8;
3440 COLOR2
3450 PRINT" ";"";STRING$(38,"-");"";
3460 FORi%=0TO20
3470   PRINTTAB(39,3+i%);"";
      TAB(0,3+i%);"";
3480   NEXT
3490 COLOR3
3500 @i%=&A
3510 VDU28,2,23,38,3
3520 VDU24,96;96;1140;672;
3530 ENDPROC
3540

```

3550 DEFPROCsubind
3560 *FX25
3570 VDU23,166,0,0,0,64,0,64,64,64
3580 VDU23,167,0,0,0,32,0,32,160,224
3590 VDU23,174,0,0,0,0,64,80,96,80
3600 ENDPROC
3610

3620 DEFPROCbip
3630 SOUND1,-13,80,1
3640 SOUND1,-15,200,1
3650 ENDPROC
3660

3670 DEFPROCtecla
3680 PRINT"Oprime una tecla";
3690 PROCbip
3700 REPEAT
3710 UNTILGET
3720 ENDPROC
3730

3740 DEFPROCinicia
3750 *FX5,1
3760 *FX6
3770 *MOTOR 0

3780 *KEY10*FX3|MCLO.
#0|MCH."1"|M
3790 PROCsubind
3800 ENDPROC
3810

3820 DEFPROCfinal
3830 VDU28,2,23,38,3
3840 CLS
3850 PRINT"1 Menú"
3860 PRINT"4 Captura"
3870 PRINT"6 Transforma"
3880 PRINT"7 Calculos"
3890 PRINT"8 Control digital"
3900 PRINT""Archivo: ";
3910 REPEAT
3920 g%=VALGET\$
3930 UNTILg%8
3940 IFg%=8 CHAIN"Control"
3950 IFg%=7 CHAIN"Calcula"
3960 IFg%=6 CHAIN"Transfo"
3970 IFg%=4 CHAIN"Captura"
3980 CHAIN"Menu"
3990 ENDPROC

LISTADO DEL PROGRAMA TRANSFO.

```

0010
0020 REM Rutina de Transformacion
      de datos
0030 REM Revisado 13/02/91 1000
0040
0050 ONERRORSTOP
0060 MODE1
0070 CLEAR
0080 PROCcara("Transformación de
      datos")
0090 DIMarch%(1),n%(1),t(101,1),
      w(101,1),u(101)
0100 DIMv(101),tmin(1),tmax(1),
      wmin(1),wmax(1)
0110 PROCunodos
0120 PROCguardatos
0130 PRINT"Oprime ESPACIO para
      continuar";
0140 PROCbip
0150 REPEAT
0160 UNTILGET
0170 CHAIN"Traza"
0180 END
0190

0200 DEFPROCunodos
0210 LOCALno%
0220 CLS
0230 PRINT"Opera con 1 o 2
      archivos ";
0240 PROCbip
0250 REPEAT
0260 no%=GET
0270 UNTILno%=49 AND no%=50
0280 PRINTCHR$no%;
0290 no%=no%-49
0300 PROCnom_arch(0)
0310 PROCcleedatos(0)
0320 IFno%=1 PROCdos ELSE
      PROCinserta(0)
0330 PROCdar_fun
0340 PROCtransf
0350 ENDPROC

```

```

0360
0370 DEFPROCdos
0380 PROCnom_arch(1)
0390 PROCcleedatos(1)
0400 ENDPROC
0410

0420 DEFPROCnom_arch(nf%)
0430 LOCALent%
0440 PRINT""Nombre de archivo ";
      nf%;
0450 PROCbip
0460 INPUT" = "arch$
0470 arch%(nf%)=OPENIN(arch$)
0480 INPUT#arch%(nf%),tit$,
      uni_t$,uni_w$
0490 lent%=LENtit$+LENuni_t$+
      LENuni_w$+6
0500 n%(nf%)=(EXT#arch%(nf%)-
      lent%)/DIV12
0510 PRINT"tit$"Unidades de t:"uni_t$
0520 PRINT"Unidades de w(t):"uni_w$
0530 ENDPROC
0540

0550 DEFPROCcleedatos(nf%)
0560 LOCALi%
0570 PRINT""Leyendo ":"n%(nf%);"
      datos"
0580 INPUT#arch%(nf%),t(1,nf%),
      w(1,nf%)
0590 tmin(nf%)=t(1,nf%)
0600 tmax(nf%)=tmin(nf%)
0610 wmin(nf%)=w(1,nf%)
0620 wmax(nf%)=wmin(nf%)
0630 FORi%=2 TO n%(nf%)
0640 INPUT#arch%(nf%),
      t(i%,nf%),w(i%,nf%)
0650 PROCcalc_tmax(t(i%,nf%))
0660 PROCcalc_wmax(w(i%,nf%))
0670 NEXT
0680 CLOSE#arch%(nf%)

```

```

0690 ENDPROC
0700

0710 DEFPROCdar_fun
0720 CLS
0730 PRINT"Dado nf%=0,1; dé las
relaciones de"
0740 PRINT"transformación para t'(i%),
w'(i%)"
0750 PRINT"en función de: i%,
tmin(nf%),"
0760 PRINT"tmax(nf%), wmin(nf%),
wmax(nf%),"
0770 PRINT"t(i%,nf%), w(i%,nf%)
n%=";n%(nf%)
0780 PROCbip
0790 OSCLI"KEY1t(i%,0)"
0800 OSCLI"KEY2w(i%,0)"
0810 REPEAT
0820 INPUTLINE"t'(i%)= "funt$
0830 IFfunt$=""
funt$="t(i%,0)":PRINTTAB
(8,VPOS-1);funt$
0840 UNTILfunt$=""
0850 REPEAT
0860 INPUTLINE"w'(i%)= "funw$
0870 IFfunw$=""
funw$="w(i%,0)":PRINTTAB
(8,VPOS-1);funw$
0880 UNTILfunw$<>""
0890 OSCLI"KEY1P.TOP-PA."
0900 OSCLI"KEY2SA."
0910 ENDPROC
0920

0930 DEFPROCtransf
0940 LOCALi%
0950 PROCbip
0960 INPUT"Transforma desde el dato
j%= "j%
0970 INPUT" Hasta el dato k%= "k%
0980 m%=k%-j%+1
0990 PRINT"Transformando ";m%;"
datos"
1000 FORi%=j% TO k%

```

```

1010 u(i%)=EVALfunt$
1020 v(i%)=EVALfunw$
1030 NEXT
1040 ENDPROC
1050

1060 DEFPROCguardatos
1070 LOCALnarch$,archnum$,
uni_t$,uni_w$,i%
1080 CLS
1090 PROCbip
1100 INPUT"Nombre para"el nuevo
archivo: "narch$
1110 INPUT"Nueva descripción:""tit$
1120 INPUT"Unidades de t:"uni_t$
1130 INPUT"Unidades de w:"uni_w$
1140 PRINT"Guardando ";m%;
1150 PRINT" datos""en ";narch$
1160 archnum%=OPENOUT(narch$)
1170 PRINT#archnum%,tit$,
uni_t$,uni_w$
1180 FORi%=j% TO k%
1190 PRINT#archnum%,u(i%),v(i%)
1200 NEXT
1210 CLOSE#archnum%
1220 ENDPROC
1230

1240 DEFPROCcalc_tmax(a)
1250 IFimin(nf%)>a tmin(nf%)=a
1260 IFtmax(nf%)<=a tmax(nf%)=a
1270 ENDPROC
1280

1290 DEFPROCcalc_wmax(b)
1300 IFwmin(nf%)>b wmin(nf%)=b
1310 IFwmax(nf%)<=b wmax(nf%)=b
1320 ENDPROC
1330

1340 DEFPROCinserta(nf%)
1350 LOCALx,z,i$,g$
1360 PROCbip
1370 PRINT"¿Insertar datos? (S/N) ";

```

```

1380 REPEAT
1390 f$=GET$
1400 UNTILINSTR("SsNn",f$)<>0
1410 IFf$="N" OR f$="n" ENDPROC
1420 REPEAT
1430 INPUT""Después del dato j="j%
1440 FORi%=1 TO j%
1450   u(i%)=t(i%,nf%)
1460   v(i%)=w(i%,nf%)
1470   NEXT
1480 PRINT""valores a insertar:"
1490 PRINT"t(";j%+1;","0)= ";
1500 INPUT""x
1510 PRINT"w(";j%+1;","0)= ";
1520 INPUT""z
1530 u(j%+1)=x
1540 v(j%+1)=z
1550 n%(nf%)=n%(nf%)+1
1560 PRINT""El resto de los datos se
recorre"
1570 PRINT""Ahora se tienen
";n%(nf%);" datos"
1580 FORi%=j%+2 TO n%(nf%)
1590   u(i%)=t(i%-1,nf%)
1600   v(i%)=w(i%-1,nf%)
1610   NEXT
1620 FORi%=1 TO n%(nf%)
1630   t(i%,nf%)=u(i%)
1640   w(i%,nf%)=v(i%)
1650   NEXT
1660 PRINT""¿Otra inserción? (S/N)"
1670 PROCbip
1680 REPEAT
1690   g$=GET$
1700   UNTILINSTR("SsNn",g$)0
1710   UNTILg$="n" OR g$="N"
1720   ENDPROC
1730

```

```

1740 DEFPROCcara($)
1750 LOCALi%
1760 COLOR2
1770 VDU19,1,2,0;0;0;
1780 VDU19,2,1,0;0;0;
1790 PRINTTAB(0,24);"+";
STRING$(38,"-");"+";TAB(0,0);
1800 VDU11
1810 PRINT"+";STRING$(38,"-");
"+";"";
1820 COLOR1
1830 PRINTSPC7;!$;SPC7;
1840 COLOR2
1850 PRINT " ";"";STRING$(38,"-");"";
1860 FORi%=0TO20
1870 PRINTTAB(39,3+i%);"";
TAB(0,3+i%);"";
1880 NEXT
1890 COLOR3
1900 @i%=&307
1910 VDU28,2,23,38,3
1920 *KEY10*FX3|MCLO.#0|M
CH."!"|M
1930 ENDPROC
1940
1950 DEFPROCbip
1960 SOUND1,-13,80,1
1970 SOUND1,-15,200,1
1980 ENDPROC
1990
2000 DEFFNu(t)
2010 REM escalon unitario
2020 LOCALu
2030 IFt u=0 ELSE u=1
2040 =u

```

LISTADO DEL PROGRAMA CONTROL

0010		0370	
0020	REM Control digital directo	0380	DEFPROCcalcula
0030	REM Lazo abierto y cerrado	0390	salmax=&FFFF*sensor(1)
0040	REM rev 130291 1000	0400	inmax=&FFFF*sensor(2)
0050		0410	PRINT"Viomax=";inmax,
0060	ONERRORSTOP		"Vsalmax=";salmax;" volts"
0070	MODE1	0420	ENDPROC
0080	CLEAR	0430	
0090	PROCCara("Control Digital Directo LA y LC")	0440	DEFPROCpideK
0100	PROCCinicia	0450	PRINT"Valor deseado a la salida"
0110	PROCCcesensor	0460	PRINT"válido de 0.3 a ";0.9*salmax;" volts""Setpoint= "
0120	PRINT"Lazo abierto o cerrado (A/C) ";	0470	PROCbip
0130	REPEAT	0480	REPEAT
0140	laz\$=GET\$	0490	INPUTTAB(10,8)set
0150	UNTILINSTR("AaCc",laz\$)<>0	0500	UNTILset>=0.3 AND set<=0.9*salmax
0160	PRINTlaz\$	0510	PRINT"Dar las ganancias PID (discretas)"
0170	IFINSTR("Aa",laz\$)<>0	0520	PROCbip
	PROClazabi ELSE PROClazcer	0530	INPUT"Kp= "Kp
0180	PROCCoprime	0540	INPUTTAB(12,VPOS-1)"Kj= "Kj
0190	CHAIN"Traza"	0550	INPUTTAB(24,VPOS-1)"Kd= "Kd
0200	END	0560	PRINT"Ciclos de control (2-100)"
0210		0570	INPUT"n= "ciclo%
		0580	ENDPROC
		0590	
0220	DEFPROClazabi	0600	DEFPROCvallini
0230	PROCCara("Control de lazo abierto")	0610	w(0,0)=0
0240	PROCCventana	0620	w(1,0)=0
0250	PROCCmarcas	0630	w(1,1)=0
0260	PROCCconman	0640	u(0)=set
0270	ENDPROC	0650	u(1)=u(0)
0280		0660	ENDPROC
		0670	
0290	DEFPROClazcer		
0300	PROCCara("Control de lazo cerrado")		
0310	PROCCcalcula		
0320	PROCCpideK		
0330	PROCCvallini		
0340	PROCCcondig		
0350	PROCCguardatos(ciclo%,0,1)		
0360	ENDPROC		

```

0680 DEFPROCsetpoint
0690 set=set+FNteclas/50
0700 IFset<=0 set=0
0710 IFset>=salmax set=salmax
0720 ENDPROC
0730

0740 DEFFNteclas
0750 pica%=INKEY(0)
0760 IFpica%=-1 tecla%=0
0770 IFpica%=136 tecla%=-1
0780 IFpica%=137 tecla%=1
0790 IFpica%=138 tecla%=-5
0800 IFpica%=139 tecla%=5
0810 =tecla%
0820

0830 DEFPROCsaldig(val%)
0840 VDU2
0850 VDU1,val%
0860 VDU3
0870 ENDPROC
0880

0890 DEFPROCbip
0900 SOUND1,-13,80,1
0910 SOUND1,-15,200,1
0920 ENDPROC
0930

0940 DEFPROCleesensor
0950 LOCALarchnum%,i%,tit$,x$
0960 PRINT"Leyendo factores"
0970 sensor(0)=1
0980 archnum%=OPENIN
("D.Sensor")
0990 INPUT#archnum%,tit$,x$,x$
1000 FORi%=1 TO 4
1010 INPUT#archnum%,n,
sensor(i%)
1020 NEXT
1030 CLOSE#archnum%
1040 ENDPROC
1050

```

```

1060 DEFPROCguardatos(n%,j%,k%)
1070 LOCALarch$,tit$,ut$,uw$,
archnum%,i%
1080 PROCbip
1090 PRINT"Nombre para"
1100 INPUT"este archivo: "arch$
1110 INPUT"Descripción de los
datos:"tit$
1120 INPUT"Unidades de t:"ut$
1130 INPUT"Unidades de w(t):"uw$
1140 PRINT"Guardando datos"
1150 archnum%=OPENOUT(arch$)
1160 PRINT#archnum%,tit$,ut$,uw$
1170 FORi%=1 TO n%
1180 PRINT#archnum%,w(j%,i%),
w(k%,i%)
1190 NEXT
1200 CLOSE#archnum%
1210 ENDPROC
1220

1230 DEFPROCbarra(u%)
1240 PLOT85,u%*4,0
1250 PLOT85,u%*4,ancho%
1260 ENDPROC
1270

1280 DEFPROCoprime
1290 PRINT"Oprime ESPACIO para
continuar"
1300 REPEAT
1310 UNTILGET
1320 ENDPROC
1330

1340 DEFPROCmide
1350 REPEAT
1360 UNTILADVAL(0)DIV256=1
1370 vsal=ADVAL(1)*sensor(1)
1380 REPEAT
1390 UNTILADVAL(0)DIV256=2
1400 vent=ADVAL(2)*sensor(2)
1410 ENDPROC
1420

```

```

1430 DEFPROCmarabl
1440 CLS

1450 DEFPROCdisplay
1460 VDU28,24,15,29,10
1470 CLS
1480 @%=&20206
1490 PRINTabohora%'vent'vsal;
1500 ENDPROC
1510

1520 DEFPROCconman
1530 *FX16,2
1540 REPEAT
1550 delta%=FNteclas
1560 PROCsuma(delta%)
1570 PROCsaldig(ahora%)
1580 PROCmide
1590 PROCdisplay
1600 UNTILFALSE
1610 ENDPROC
1620
1630 REM poner PROCsetpoint despues
de FOR
1640 REM cuando ciclos sean infinitos
1650

1660 DEFPROCcondig
1670 *FX16,1
1680 PROCoprime
1690 TIME=0
1700
1710 FORi%=2 TO ciclo%
1720 PROCsaldig(u(i%-1))
1730 REPEAT
1740 UNTILADVAL(0)DIV256=1
1750 w(1,i%)=ADVAL(1)*sensor(1)
1760 w(0,i%)=TIME
1770
1780 Err=w(1,i%-1)-w(1,i%)
1790 u(i%)=u(i%-1)+Kp*Err+
Ki*(set-w(1,i%))+
Kd*(w(1,i%-1)+Err-w(1,i%-2))
1800 IFu(i%)>&FF u(i%)=&FF
1810

```

```

1820 NEXT
1830 CLS
1840 PRINT"Ganancias"
1850 PRINT"Kp= ";Kp;" Ki= ";Ki;"
Kd= ";Kd
1860 PRINT"Entrada r(k)=";set
1870 PRINT"Salida w(k)=";w
(1,i%-1)
1880 PRINT"Control u(k)=";u(100)
1890 PRINT"Tiempo de muestreo T=
";TIME/ciclo%
1900 PROCsaldig(0)
1910 *MOTOR 0
1920 ENDPROC
1930

1940 DEFPROCinicia
1950 @%=&20308
1960 *MOTOR 1
1970 *FX16,2
1980 *FX5,1
1990 *FX6
2000 *FX4,1
2010 PROCsaldig(0)
2020 *KEY10*FX3|M*FX4|M
CLO.#0|MCH."!"|M
2030 DIMsensor(4),w(2,100),u(100)
2040 ENVELOPE1,0,100,0,0,0,0,
1,0,0,0,40,40
2050 ancho%=28
2060 digancho%=24
2070 antes%=0
2080 ahora%=0
2090 marco$="+ - + +
2100 ENDPROC
2110

2120 DEFPROCventana
2130 VDU29,128;610;
2140 VDU24,-610;-550;610;100;
2150 ENDPROC
2160

```

```

2170 DEFPROCcura(txt$)
2180 LOCAL i%,z%,d%
2190 COLOR2
2200 VDU26,12
2210 VDU19,1,2,0;0,0;
2220 VDU19,2,1,0;0,0;
2230 PRINT
      "+";STRING$(38,"-");"+";"";
2240 z%=(38-LEN(txt$))/DIV2
2250 d%=(37-LEN(txt$)-z%
2260 COLOR1
2270 PRINTSPCz%;txt$;SPCd%;
2280 COLOR2
2290 PRINT " ";"";STRING$(38,"-");"";
2300 FOR i%=0 TO 19
2310   PRINTTAB(39,3+i%);"";
      TAB(0,3+i%);"";
2320 NEXT
2330 PRINT "+";STRING$(38,"-");
      "+";TAB(3,1);
2340 COLOR3
2350 VDU28,2,22,38,3
2360 ENDPROC
2370

2380 DEFPROCJoy
2390 REPEAT
2400   cnt%=0
2410   TIME=0
2420 REPEAT
2430   REPEAT
2440     ant%=ADVAL(0) AND 1
2450     UNTIL ant%=1 OR
      TIME=49
2460     cnt%=cnt%+1
2470   REPEAT
2480     ant%=ADVAL(0) AND 1
2490     UNTIL ant%=0
      OR TIME=49
2500   UNTIL TIME=49
2510 PRINTTAB(2,2)cnt%
2520 REPEAT

```

```

2530 UNTIL TIME=100
2540 UNTILO
2550 ENDPROC
2560

2570 DEFPROCsuma(incres%)
2580 SOUND1,1,ahoras%/3,1
2590 antes%=ahoras%
2600 GCOL0,1
2610 IF incres%<>0 AND
      SGN(incres%)<0 GCOL0,0
2620 horas%=ahoras%+incres%
2630 horas%=ABS horas%
2640 IF horas%>=&FF horas%=&FF
2650 MOVE horas%*4,0
2660 MOVE horas%*4,zncho%
2670 PROC barra(antes%)
2680 ENDPROC
2690

2700 DEFPROCmarcas
2710 PRINTTAB(0,18);"ROMPE para
      volver al menú"
2720 PRINTTAB(1,1);"0";
2730 PRINTTAB(32,1);"255";
2740 PRINTTAB(8,7);"Valor digital="
2750 PRINTTAB(3,9);"Voltaje de
      entrada="
2760 PRINTTAB(4,11);"Voltaje de
      salida="
2770 ENDPROC
2780

2790 DEFPROCmuestra
2800 ENDPROC
2810

2820 DEFPROCcalgo
2830 ENDPROC

```

LISTADO DEL PROGRAMA CALCULA

```

0010
0020 REM Rutinas p calculos varios
0030 REM rev 130291 1000
0040
0050 ONERRORSTOP
0060 MODE1
0070 CLEAR
0080 PROCcara("Cálculos varios")
0090 DIMR1(13),R2(13)
0100 DATA1,1.2,1.5,1.8,2.2,2.7,3.3
0110 DATA3.9,4.7,5.6,6.8,8.2,8.2
0120 PRINT"Opciones:"
0130 PRINT"1. Resistencias en
paralelo"
0140 PRINT"2. Resistencias para
ganancia"
0150 PRINT"3. Sensibilidad de
amplificador"
0160 PRINT"4. Polos y ceros 2 orden"
0170 PRINT"5. Polos y ceros 3 orden"
0180 PRINT"6. Reglas de Ziegler y
Nichols"
0190 PRINT"7. Menú Principal"
0200 PRINT"Opción: ";
0210 op$=FNtecla("1234567")
0220 PRINTop$;
0230 IFop$="1" PROCpara
0240 IFop$="2" PROCgana
0250 IFop$="3" PROCsens
0260 IFop$="4" PROCvalcerSEG
0270 IFop$="5" PROCvalcerTER
0280 IFop$="6" PROCziegler
0290 IFop$="7" SOUND-16,116,0,0:
CHAIN"Menu"
0300 PROCbip
0310 PRINT""Oprime ESPACIO para
reiniciar"
0320 PRINT"o ROMPE para volver al
Menú"
0330 REPEAT
0340 UNTILGET
0350 RUN
0360 END
0370

0380 DEFPROCsens
0390 REM Calc sensibilidad de amplif
oper
0400 REM en funcion de R1,R2 y tol
+- 5%
0410 @%=&20308
0420 CLS
0430 PRINT"Dado Av = R/R, e=5%, V-
= 0"
0440 INPUT""Resistencias: ""R1="R1
0450 INPUT""R2="R2
0460 INPUT""Entrada al
operacional: ""V+=";V2
0470 Av=R2/R1
0480 PRINT"Ideal Av= ";Av
0490 idealv0=Av*V2
0500 PRINT"Ideal Vo= ";idealv0
0510 e=0.05
0520 max=(1+Av*(1+2*e))/
(Av+1-2*e)
0530 maxP=(max-1)*100
0540 PRINT" Vo max =
";idealv0*max, "(+";maxP;"%)
0550 min=(1+Av*(1-2*e))/
(Av+1+2*e)
0560 minP=(min-1)*100
0570 PRINT" Vo min =
";idealv0*min, "(;"minP;"%)
0580 ENDPROC
0590

0600 DEFPROCgana
0610 REM Calc Rs comercial p
ganancia de AmpOp
0620 CLS
0630 FORi%=1 TO 12
0640 READ R1(i%)
0650 R2(i%)=R1(i%)
0660 NEXT
0670 INPUT""Que valor de
ganancia ""deseas (1-10) ";Av
0680 INPUT""Máximo porcentaje de
error ";maxpct

```



```

0690 PRINT"SPC4;
      "R / R = Av"SPC7"%";
0700 @%=&20308
0710 FORi%=1 TO 12
0720 FORj%=2 TO 12
0730 IFj%=i% THEN 820
0740 r=R1(i%)/R2(j%)
0750 IFr<1 r=*10
0760 IFr>10 r=r/10
0770 pct=(r-Av)/Av*100
0780 IFABSpc<maxpc
ok=TRUE ELSE ok=FALSE
0790 IFpct>0 sgn$="+ "
ELSEsgn$=""
0800 IFABSpc>=10
sgn$=CHR$8+sgn$
0810 IFok PRINTR1(i%),R2(j%)
; "r;" ";sgn$;pct;"% "
0820 NEXT
0830 VDU13
0840 NEXT
0850 ENDPROC
0860
0870 DEFPROCpara
0880 REM Calc Rs en paralelo p valor
deseado
0890 CLS
0900 RESTORE
0910 FORi%=1 TO 12
0920 READ R1(i%)
0930 R2(i%)=R1(i%)
0940 NEXT
0950 INPUT"Que valor deseas ";par
0960 INPUT"Máximo porcentaje de
error ";maxpc
0970 PRINT"SPC4"R"SPC6"R"SPC5"R
%"
0980 @%=&20308
0990 FORi%=1 TO 12
1000 FORfact1=0 TO 3
1010 FORj%=i% TO 12
1020 FORfact2=0 TO 3
1030 Rn1=R1(i%)*10^fact1
1040 Rn2=R2(j%)*10^fact2

```

```

1050 mpy=Rn1*Rn2
1060 den=Rn1+Rn2
1070 r=mpy/den
1080 pct=(r-par)/par*100
1090 IFABSpc<maxpc
ok=TRUE
ELSE ok=FALSE
1100 IFpct>0 sgn$="+ "
ELSEsgn$=""
1110 IFABSpc>=10
sgn$=CHR$8+sgn$
1120 IFok PRINTRn1,Rn2;" "
;r;" ";sgn$;pct;"% "
NEXT
1130 NEXT
1140 NEXT
1150 NEXT
1160 VDU13
1170 NEXT
1180 ENDPROC
1190
1200 DEFFNvalor
(R1,R2,e1,e2,e3,e4,V2,V1)
1210 NR1=R1*(1+e1)
1220 NR2=R2*(1+e2)
1230 NR3=R1*(1+e3)
1240 NR4=R2*(1+e4)
1250 cv2=((NR1+NR2)/
(NR3+NR4))*NR4/NR1
1260 cv1=NR2/NR1
1270 v0=cv2*V2-cv1*V1
1280 =v0
1290
1300 DEFPROCcara(txt$)
1310 LOCALi%,z%,d%
1320 COLOR2
1330 VDU19,1,2,0;0;0;
1340 VDU19,2,1,0;0;0;
1350 PRINTTAB(0,24);"+";
STRING$(38,"-");"+";TAB(0,0);
VDU11
1360 PRINT"+";STRING$(38,"-");"
+";"";
1380 z%=(38-LENtxt$)DIV2

```

```

1390 d%=37-LENtxt%-z%
1400 COLOR1
1410 PRINTSPCz%;txt%;SPCd%;
1420 COLOR2
1430 PRINT " ";";STRING$(38,".-");";
1440 FORi%=0TO20
1450   PRINTTAB(39,3+i%);";
      TAB(0,3+i%);";
1460 NEXT
1470 COLOR3
1480 @%=&A
1490 VDU28,2,23,38,3
1500 *KEY10*FX3|MCLO.#0
      [MCH."!"|M
1510 ENDPROC
1520

1530 DEFPROCbip
1540 SOUND1,-13,80,1
1550 SOUND1,-15,200,1
1560 ENDPROC
1570

1580 DEFFNtecla(opcs$)
1590 REPEAT
1600   tecla$=GET$
1610   UNTILINSTR(opcs$,tecla$)0
1620   =tecla$
1630

1640 DEFPROCziegler
1650 @%=&30308
1660 CLS
1670 PRINT""Reglas de Ziegler
      Nichols"
1680 PRINT""Por favor las constantes
      R y L"
1690 INPUT""R= "R
1700 INPUT""L= "L
1710 T=0
1720 PRINT""Continuo o Discreto
      (C,D) ""
1730 REPEAT
1740   tip$=CHR$(GET AND &5F)
1750   UNTILtip$="C" OR tip$="D"

```

```

1760 PRINTtip$
1770 IFtip$="D" PROCmuesT
1780 CLS
1790 PRINT""Reglas de Ziegler
      Nichols"
1800 PRINT""R= ";R;TAB(16);"L=
      ";L;TAB(28);"T= "T
1810 PRINT""Ganancias independientes:"
1820 PRINT""TAB(5);"Kp";TAB(16)
      ";"Ki";TAB(28);"Kd"
1830 IFtip$="C" PROCcont ELSE
      PROCdiscre
1840 ENDPROC
1850

1860 DEFPROCcont
1870 PROCp
1880 PROCpinta("P")
1890 PROCpi
1900 PROCpinta("PI")
1910 PROCpid
1920 PROCpinta("PID")
1930 ENDPROC
1940

1950 DEFPROCp
1960 Kp=1/R/L
1970 Ki=0
1980 Kd=0
1990 ENDPROC
2000

2010 DEFPROCpi
2020 Kp=0.9/R/L
2030 Ki=Kp*0.3/L
2040 Kd=0
2050 ENDPROC
2060

```

```

2070 DEFPROCpid
2080 Kp=1.2/R/L
2090 Ki=Kp*0.5/L
2100 Kd=Kp*0.5*L
2110 ENDPROC
2120

2130 DEFPROCmuest
2140 PRINT"Por favor el tiempo de
muestreo"
2150 INPUT"T= "T
2160 ENDPROC
2170

2180 DEFPROCdiscre
2190 PROCpD
2200 PROCpinta("P")
2210 PROCpID
2220 PROCpinta("PI")
2230 PROCpidD
2240 PROCpinta("PID")
2250 ENDPROC
2260

2270 DEFPROCpD
2280 r1=R*(L+T)
2290 Kp=1/r1
2300 Ki=0
2310 Kd=0
2320 ENDPROC
2330

2340 DEFPROCpID
2350 IFL/T.2 PRINT"Cuidado: L/T
=":L/T;" puede oscilar"
2360 r1=R*(L+T/2)
2370 r12=r1*(L+T/2)
2380 Ki=.27*T/r12
2390 Kp=.9/r1-Ki/2
2400 Kd=0
2410 ENDPROC
2420

```

```

2430 DEFPROCpidD
2440 LOCALe,f0,g
2450 r1=R*(L+T)
2460 f0=1.*(L+T/2)*(L+T/2)
2470 Ki=.6*T/r12
2480 Kp=1.2/r1-Ki/2
2490 e=1
2500 f0=ABS(10*L/T-INT(10*L/T))
2510 IFF<=0 THEN g=0.5 ELSE
g=0.6
2520 IFF<=0.5 PRINT"Note que
L/T entero"
2530 Kd=g/(R*T)
2540 ENDPROC
2550

2560 DEFPROCpinta(5)
2570 PRINT"t5;TAB(5);Kp;TAB(10)
;Ki;TAB(28);Kd
2580 ENDPROC
2590

2600 DEFPROCvalcerSEG
2610 LOCALA0,A1,A2,A3,T1,T2
2620 LOCALtau1,tau2,dcn,err,c0
2630 @%=&20308
2640 CLS
2650 PRINT"Dado g(t)=""
2660 PRINT"A - Aexp(-t/T) - Aexp(-t/T)""
2670 PRINT"se calcularán B, a y c en:"
2680 PRINTSPC6"B(as+s+c)"
2690 PRINT"G(s)=""STRING$(16,")"
2700 PRINTSPC5"(s+1/T)(s+1/T)""
2710 PRINT"donde idealmente a=0""
2720 PRINT"Oprime ESPACIO para
proceder"
2730 REPEATUNTILGET
2740 CLS
2750 PRINT"OJO con los signos"
2760 PRINT"Coefficientes"
2770 INPUT"A="A0
2780 INPUT"A="A1
2790 INPUT"A="A2
2800 PRINT"Constantes de tiempo"
2810 INPUT"T="T1

```

```

2820 INPUT"T="T2
2830 tau1=1/T1
2840 tau2=1/T2
2850 den=A0*(tau1+tau2)+
A1*tau2+A2*tau1
2860 c0=A0*tau1*tau2/den
2870 err=(A0-A1-A2)/den
2880 PRINT"La ganancia es B= ";den
2890 PRINT"Los polos Ū, Ū son: ";
2900 PRINT"-";tau1"; -";tau2
2910 PRINT"El cero es c= ";-c0
2920 PRINT"El error es a= ";err
2930 ENDPROC
2940

2950 DEFPROCvalcerTER
2960 @%=&20408
2970 LOCALA0,A1,A2,A3,T1,T2,T3,
t1,t2,t3
2980 CLS
2990 PRINT"Dado g(t) = A -
Aexp(-t/T)"
3000 PRINT" - Aexp(-t/T) -
Aexp(-t/T)"
3010 PRINT"se calcularán B, C y C
en:"
3020 PRINTSPC11;"B(s+C)(s+C)"
3030 PRINT"G(s)= ";STRING$(24,".")
3040 PRINTSPC6"(s+1/T)(s+1/T)
(s+1/T)"
3050 PRINT"OJO con los signos"
3060 PRINT"Oprime ESPACIO para
proceder"
3070 REPEATUNTILGET
3080 CLS
3090 PRINT"Coefficientes"
3100 INPUT"A="A0
3110 INPUT"A="A1
3120 INPUT"A="A2
3130 INPUT"A="A3
3140 PRINT"Constantes de tiempo"
3150 INPUT"T="T1
3160 INPUT"T="T2
3170 INPUT"T="T3
3180 t1=1/T1

```

```

3190 t2=1/T2
3200 t3=1/T3
3210 CLS
3220 PRINT"En el plano complejo s"
3230 PRINT"los polos Ū, Ū, Ū son:"
3240 PRINT-t1"; -";t2"; -";t3
3250 a0=A0-A1-A2-A3
3260 a1=A0*(t1+t2+t3)-
A1*(t2+t3)-A2*(t1+t3)-
A3*(t1+t2)
3270 a2=t1*t2*(A0-A3)+
t1*t3*(A0-A2)+t2*t3*(A0-A1)
3280 a3=t1*t2*t3*A0
3290 IFa1<0.001 @%=&030408
3300 PRINT"En el plano complejo s"
3310 PRINT"los ceros C, C son:"
3320 PROCcuadra(a2/a1,a3/a1)
3330 c1=rcal+imag
3340 c2=rcal-imag
3350 PRINT"La ganancia es B= ";a1
3360 PRINT"El error es a= ";a0/a1
3370 REM Ojo Valido solo cuando
ceros en s son reales
3380 INPUT"tiempo de muestreo T=
";T
3390
3400 alfa1=a1*(c1-t1)*(c2-t1)/
(-t1+t2)*(-t1+t3)
3410 alfa2=a1*(c1-t2)*(c2-t2)/
(-t2+t1)*(-t2+t3)
3420 alfa3=a1*(c1-t3)*(c2-t3)/
(-t3+t1)*(-t3+t2)
3430 g1=EXP(-T*t1)
3440 g2=EXP(-T*t2)
3450 g3=EXP(-T*t3)
3460 b1=alfa1*(1-g1)
3470 b2=alfa2*(1-g2)
3480 b3=alfa3*(1-g3)
3490 d0=1/(b1+b2+b3)
3500 d1=-d0*(b1*(g2+g3)+
b2*(g1+g3)+b3*(g1+g2))
3510 d2=d0*(b1*g2*g3+
b2*g1*g3+b3*g1*g2)
3520 CLS

```

```
3530 PRINT"Las constantes calculadas  
son"  
3540 PRINT"(α's)";alfa1;" ";alfa2;" ";  
alfa3  
3550 PRINT"(β's)";b1;" ";b2;" ";b3  
3560 PRINT"En el plano Z los polos  
son:"  
3570 PRINT"(γ's)";g1;" ";g2;" ";g3  
3580 PRINT"(δ's)";d0;" ";d1;" ";d2  
3590 PRINT"En el plano Z los ceros son:"  
3600 PRINT"(ζ's)";  
3610 PROCcuadra(d1,d2)  
3620 ENDPROC  
3630
```

```
3640 DEFPROCcuadra(b,c)  
3650 real=-b/2  
3660 discr=(b*b-4*c)/4  
3670 IFdiscr>=0 imag=SQR(discr)  
ELSE imag=SQR(-discr)  
3680 IFdiscr cmplx=1  
3690 IFdiscr=0 PRINTreal+imag,"  
"real-imag'  
3700 IFdiscr PRINTreal;"  
3710 ENDPROC
```

2 DESCRIPCION DEL EQUIPO Y ACCESORIOS

DESARROLLO DE LA MICROBBC

El nacimiento de los microcircuitos dió también lugar al nacimiento de las microcomputadoras. Lamentablemente, la introducción de estos equipos al mercado fué principalmente orientada a juegos para pasar el tiempo, sin emplear al máximo las capacidades que tienen tales máquinas y desde luego, también el potencial del hombre para aprender y desarrollarse.

Aquella situación marca un desarrollo limitado del uso de las microcomputadoras, que se presentó como fenómeno mundial. Desde este punto de vista fué la Gran Bretaña uno de los primeros países que identificó el impacto que la computación tendría sobre la educación futura a todos los niveles. En particular, fué la British Broadcasting Corporation (BBC) la que tuvo un papel decisivo en difundir masivamente el potencial de la computación.

Al inicio de los ochentas, la BBC de Londres produjo una serie de programas televisivos en forma de documental, que analizaron el funcionamiento y la operación de las microcomputadoras así como las consecuencias de su uso en términos tecnológicos, socioeconómicos y culturales. Estos programas para la televisión obtuvieron audiencias muy altas, hasta entonces inesperadas para documentales.

Con éstos éxitos, la BBC tomó la alternativa de contratar a educadores prominentes de la Universidad de Cambridge así como a expertos en computación, a quienes les solicitó que formularan las especificaciones de una microcomputadora flexible y de costo bajo que ejemplificara los temas tratados en los programas mencionados.

Con ésto nació la BBC Micro, diseñada y fabricada por Acorn Computers Limited., de Cambridge, compañía bien conocida en Inglaterra por sus innovaciones tecnológicas. (Acorn hoy en día es subsidiaria del consorcio italiano Olivetti).

La MicroBBC incorporó una gama de posibilidades de aplicación y expansión, sobre todo en cuanto a su uso educativo, por lo que fue adoptada por el Gran Bretaña en un programa ambicioso

para dotar de microcomputadoras a todo el ámbito escolar desde primaria hasta profesional. Hoy día alrededor de 90% de las escuelas británicas están equipadas con computadoras, sea como máquinas independientes, en redes locales o bien enlazadas a redes públicas de información.

DESCRIPCION DE LA MICROBBC

La MicroBBC está contenida en un gabinete de plástico de color beige claro, de 43x 34x 7 cm., y pesa 3 kg. aproximadamente. La alimentación eléctrica adecuada para su funcionamiento es de 117 Volts \pm 10%, 60 Hertz \pm 1%, 50 watts, con conexión de fase, neutro y tierra física. Ver figura 14.

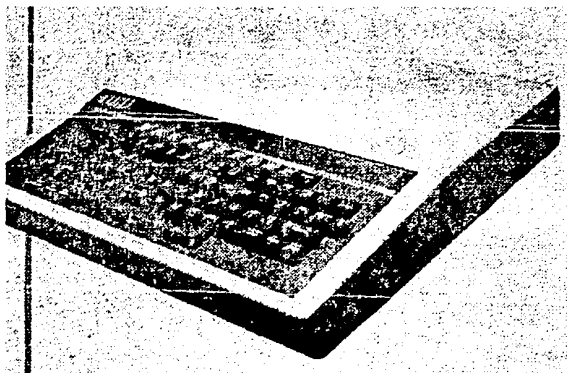


Fig. 14 Computadora MicroBBC

A continuación se resumen las características más importantes que distinguen a la MicroBBC:

Fuente de poder interna que genera los voltajes necesarios para el funcionamiento de la máquina (+5V, -5V, +12V), y salida auxiliar de poder para una unidad de disco externa.

Microprocesador 6502A de 2Mhz, con reloj de tiempo real, con período de 10 milisegundos.

Teclado de 73 teclas tipo QWERTY con auto-repetición, incluyendo 10 teclas de función definibles por el usuario, y las teclas de control del cursor.

Circuitos integrados de generación de sonido, en 3 canales de música y uno de ruido, con control completo de envolvente (ADSR) y bocina propia.

Salida integrada para televisor de color, norma NTSC, modulada en VHF canal 3.

Salida integrada de video compuesto para monitor PAL de color o blanco y negro con escala de grises.

Salida RGB con sincronía, con niveles de voltaje TTL, para monitor de color de alta resolución.

Entrada/Salida serial con el estándar RS423, interoperable con estándar RS232C, con velocidades seleccionables por programa, desde 95 hasta 9600 baudios.

Sistema Operativo de la máquina contenido en ROM de 32kB.

Lenguaje de programación intérprete de BASIC con lenguaje ensamblador 6502 incluido, en ROM de 32kB, con extenso soporte para gráficos y sonido.

Sistema de archivos en Disco/Red, en ROM de 32kB. Cada uno de ellos se selecciona con los comandos *DISCO o *RED.

Dos zapatas adicionales para ROMS/EPROMS de 16kB, para utilizarlas como procesador de palabras, lenguajes PASCAL, LOGO, LISP, etcétera.

Desplegado a colores teletext (MODO 7) de 40 caracteres por 25 líneas, con caracteres redondeados, y caracteres semigráficos, con bajo requerimiento de RAM (1 kB).

Siete modos adicionales de despliegue (0-6), con caracteres redefinibles por el usuario, en adición al estándar ASCII, alfabeto griego y símbolos científicos con 80, 40 o 20 caracteres por línea, hasta 16 efectos de color y gráficos de alta resolución, con ventanas de texto y de gráficos separadas o encimadas, definibles por el usuario.

Memoria RAM dinámica, de 32 kB, dividida para despliegado de gráficos de alta resolución, programa y variables del usuario.

Puerto de usuario: Entrada/Salida digital programable de 8 bits de datos y 2 bits de control, manejado por programa. Se usa para conectar periféricos, a saber: ratón, tabletas de gráficos, lector de barras, y otros.

Bus de extensión de 1MHz con acceso directo a memoria, para conexión de disco duro (Winchester) estándar SCSI, de 10 o 30 MB, interfase IEEE488, etc.

Interfase para discos flexibles de 5 1/4" doble lado-doble densidad, 80 pistas, con interfase estándar SASI.

Conexión TUBO, especialmente dedicada para la conexión de segundos procesadores (6502, Z80, 80186, etc).

Puerto Econet de entrada/salida, para la conexión de varias MicroBBC conectadas en Red de Area Local (LAN), con hasta 254 máquinas interconectadas, compartiendo uno o varios Servidores de Archivo donde se almacena la información, y uno o varios Servidores de Impresión para la salida de listados. Este puerto ya está integrado como estándar en la MicroBBC.

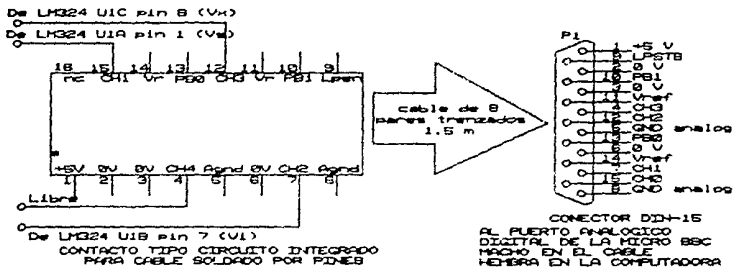


Fig. 15 Conector y cable del puerto analógico digital.

Entrada/Salida integrada para grabadora de cassettes estándar, con control automático de motor, para lectura/escritura de datos o programas, a 300 ó 1200 baudios (seleccionable por el usuario). El control de motor de este puerto (relevador) se utilizó como interruptor general de encendido del motor.

CONVERTIDOR ANALOGICO DIGITAL

Para convertir voltajes analógicos a valores digitales, la MicroBBC cuenta también con una entrada de conversión analógica digital de 4 canales, seleccionables por programa, integrados en un circuito mPD7002.

Este circuito es del tipo integrador, con voltaje de referencia de 1.8 volts, con tiempo de conversión de 10 milisegundos para resolución de 12 bits, o de 4 ms para 10 bits (más rápido con menor precisión). Por omisión se utilizan conversiones de 12 bits (dando un máximo aproximado de 100 muestras por segundo). En ambos casos el valor obtenido es presentado en 16 bits. {Holmes y Dickens, 369}.

Los dos bits menos significativos del valor devuelto por la instrucción ADVAL(0) de BASIC indican cual canal es el que más recientemente ha terminado su conversión.

A este puerto normalmente se conectarían las palancas de juego (joysticks); en nuestra aplicación se conectarán los amplificadores de instrumentación que miden el voltaje de

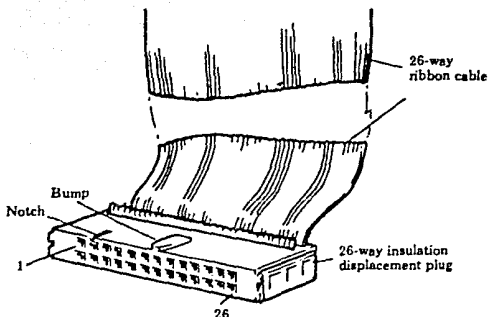


Fig. 16 Conector y cable paralelo (salida digital).

alimentación y la velocidad del motor. Ver diagrama del conector y cable usados con este puerto, en la fig. 15.

PUERTO PARALELO

La MicroBBC cuenta con un puerto de salida para impresora en paralelo, con interfase CENTRONICS de 8 bits, manejado por un circuito 6522 VIA (Versatile Interface Adaptor). Este puerto se ha conectado como salida de la computadora hacia el convertidor digital analógico, mismo que produce el voltaje de alimentación al amplificador de potencia, el cual mueve al motor. Ver figs. 16 y 17, para el diagrama de la asignación de pines del conector y del cable utilizados para este puerto.

UNIDAD DE DISCO

Esta unidad de disco es de doble cara, doble densidad, utilizando 80 pistas por cara (96tpi), y almacena información en 80 pistas de diez sectores de 256 bytes cada uno, es decir un total de 204,800 bytes (200 kB) por cara, dando un total de 409,600 bytes (400 kB) por unidad. El formato de grabación es Frecuencia Modulada.

El arranque del motor tarda menos de 250 milisegundos, haciendo girar al diskette a 300 rpm, con un tiempo de latencia de 100 ms, tiempo de acceso entre pistas de 4 ms y tiempo de asentamiento de 15 ms.

Esta unidad se conecta a cualquier MicroBBC. Requiere de alimentación eléctrica en forma de corriente directa: +5 volts y +12 volts, y consume menos de 20 watts.

MONITOR DE COLOR RGB

Monitor de alta resolución de 14 pulgadas, entrada RGB (Red, Green, Blue) compatible TTL de 1500 ohms de impedancia a través de conector DIN de 6 pines, con resolución de 452 x 550

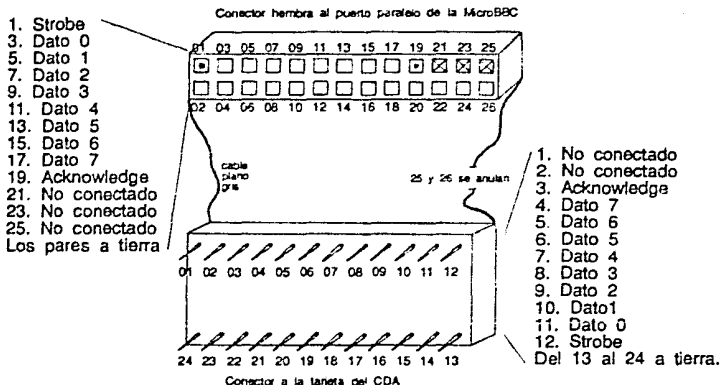


Fig. 17 Asignación de pines del cable paralelo.

pixeles, en 525 (NTSC) o 625 (PAL) líneas entrelazadas de 60 campos. Este monitor consume alrededor de 70 watts, alimentado con 117 V 60 Hz.

OTROS CIRCUITOS

A continuación también se describen algunos de los circuitos más importantes de la MicroBBC.

IC79 6502A

Microprocesador de 8 bits, de la familia NMOS 6500, que corre a 2 MegaHertz. Uno de los microprocesadores más populares diseñados para microcomputadoras personales y escolares de uso generalizado (Apple, Atari, Commodore). Controla todas las funciones de la MicroBBC.

IC16 8271

Controlador de discos flexibles, modo Frecuencia Modulada (FM), para unidades de discos de 5 1/4", de 40 u 80 pistas. También uno de los más comunmente usados.

IC17 6522

Versatile Interface Adaptor (VIA) del Sistema, se usa para el control del teclado, generadores de sonido y de habla, así como para el control y manejo de las señales del convertidor

analógico/digital (CAD), la sincronía vertical del despliegue de imagen, y otras funciones del procesador. (ver también IC11).

IC11 6522

Versatile Interface Adaptor (VIA) del usuario. La mitad "A" provee la interfase paralela Centronics a través de IC10, la mitad "B" se lleva directamente al conector PL10 para formar el puerto digital del usuario.

IC34 6505

Procesador serial de 8 bits, diseñado específicamente para controlar la transferencia de datos desde/hacia la grabadora de cassettes, así como para controlar el puerto serial de comunicaciones RS423.

DESCRIPCION DE LOS ACCESORIOS

A continuación se describen los accesorios y elementos que se han conectado a la microcomputadora MicroBBC para llevar a cabo este trabajo.

Como herramientas principales se utilizaron un multímetro marca Fluke modelo 8022A, desarmadores, pinzas de punta y de corte, dos tarjetas de conexión Protoboard, cautín y soldadura. Como herramientas de cálculo se utilizaron la misma computadora MicroBBC y una calculadora Texas Instruments TI-59.

FUENTE DE PODER

Como fuente de poder (ver fig. 18) para la planta (motor) y los demás circuitos asociados (amplificador de potencia, amplificadores de instrumentación, convertidor digital analógico y tacómetro digital) se utilizaron dos arreglos de transformadores y puentes de diodos efectuando rectificación de onda completa, con sus respectivos capacitores para eliminar el rizo. {Sladdin, 97; Millman, 349}.

Estas fuentes de poder entregan +15 V a 2 A y -15 V a 1 A, aproximadamente. La regulación de voltaje se hizo en forma local con reguladores de +12 V (7812), -12 V (7912) y +5 V (7805). {Fairchild, 14-89} Ver fig. 18.

TACOMETRO DIGITAL

Para medir la velocidad angular del motor en lazo abierto se le alimentó con corriente directa de 0 a 12 volts, y se obtuvieron medidas en revoluciones por minuto, en el rango de 50 a 6,800 rpm.

Sin embargo, el tacómetro mecánico manual que se utilizó originalmente, tenía una precisión inferior a $\pm 10\%$ (± 5 a ± 680 rpm), lo cual se evidenció como insuficiente para las necesidades del proyecto.

Por ese motivo se diseñó e implantó un sencillo tacómetro digital, con el único propósito de medir y desplegar la velocidad angular con una precisión mucho mayor de lo que era capaz de medir el tacómetro mecánico, que además resulta incómodo para usarse con regularidad.

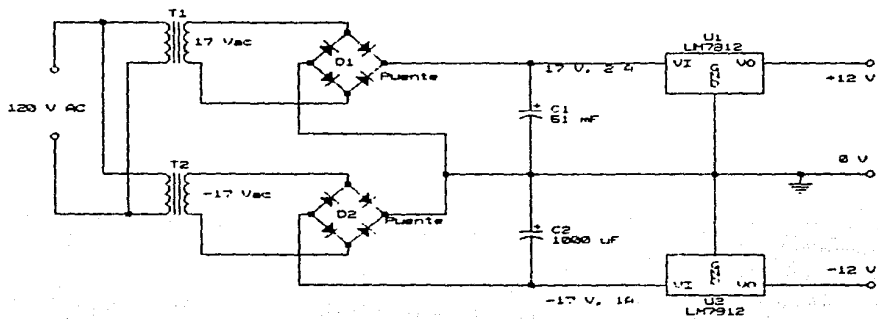


Fig. 18. Fuente de poder para planta y circuitos.

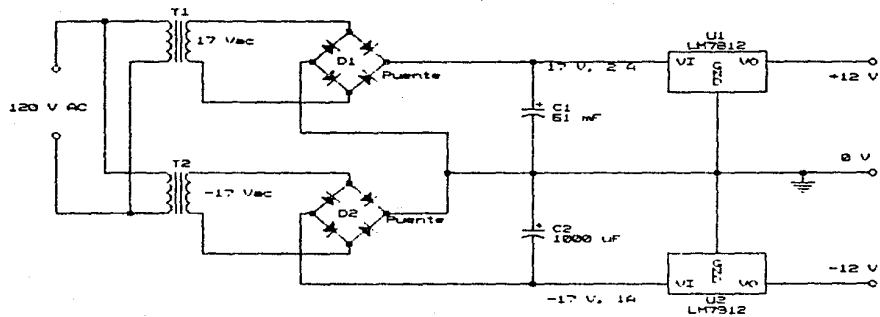


Fig. 18. Fuente de poder para planta y circuitos.

Cabe hacer notar que este tacómetro se diseñó exclusivamente para que el operador pueda visualizar la velocidad del motor de CD, si bien un posible desarrollo futuro de dicho tacómetro sería el proporcionar directamente un conteo digital a la computadora.

Este tacómetro se basa en un dispositivo sensor optoelectrónico (ST8339) formado por un par fotodiodo - fototransistor que detecta el paso de un disco cortado en sectores, adosado a la flecha del motor, de manera que se genera un pulso de voltaje a la salida del sensor cada vez que un sector del disco cruza el fotosensor. Este disco tiene dos sectores cubiertos y dos descubiertos, de modo que generará dos pulsos por cada revolución completa del rotor. Ver fig. 19.

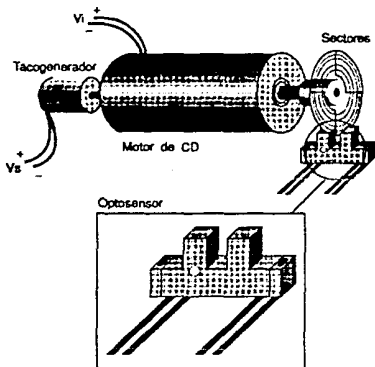


Fig. 19 Motor y optosensor.

A la salida de dicho sensor óptico se conecta un inversor digital de lógica TTL (74L04), con el propósito de acondicionar ese pulso, para luego llevarlo a un contador CMOS de 4 décadas cuya salida está ya acondicionada internamente para manejar 4 despliegues luminosos de 7 segmentos, en forma multiplexada, que mostrarán el conteo (en este caso, de revoluciones por segundo).

Este contador MM74C925 será alimentado con +5v; tiene una entrada asíncrona, activa alta para reestablecerlo (Reset); Habilita conteo (Latch Enable) alta para conteo continuo, y Reloj (Clock), que puede contar hasta 2 MHz. En sus salidas a-g se conectan resistores de 100 ohms a los despliegues, para reducir la disipación de calor y calentamiento del circuito.

El conjunto de transistores CA3046 son de propósito general, y se usan para seleccionar la salida multiplexada (A,B,C,D) hacia el despliegue luminoso.

El diagrama eléctrico a bloques del tacómetro digital completo se ve en la figura 20.

Mediante un oscilador externo (la mitad de un NE556), conectado en configuración monoestable, se generará un pulso exactamente de medio segundo de duración, que se usará como ventana de habilitación del contador. Ya que el sensor óptico generará dos pulsos por cada

revolución, tendremos una cuenta equivalente a la mitad de esa cantidad cada segundo, es decir, directamente en revoluciones por segundo.

De acuerdo al fabricante (Fairchild), el monoestable genera una salida alta durante un tiempo

$$t_1 = 1.1 R_1 C_1 \quad (1)$$

por lo cual, teniendo $t_1=0.5s$ y fijando $C_1=47\mu F$, queda:

$$R_1 = \frac{0.5}{1.1 \times 47 \times 10^{-6}} = 9.671 \text{ k}\Omega$$

por lo cual se usará un potenciómetro de 10 k Ω para ajustar este valor exactamente.

Además se generará un pulso de borrado y reinicio del contador, utilizando la otra mitad del circuito NE556, conectado como biestable. La secuencia de borrado y conteo se iniciará automáticamente ya que este pulso será generado por el oscilador conectado como astable. Este mismo pulso es el que dispara el monoestable descrito arriba.

De acuerdo al fabricante, el astable generará una onda cuadrada cíclica, de valor bajo por la duración de t_2 , y alto por el tiempo t_3 , cuyas ecuaciones son:

$$t_2 = 0.693 R_2 C_2 \quad (2)$$

$$t_3 = 0.693 (R_2 + R_3) C_2 \quad (3)$$

El pulso t_2 invertido provocará el reestablecimiento del contador y el disparo del monoestable. La duración mínima de este pulso debe ser de 250 ns para el contador, y de 150 ns para el monoestable (según sus respectivos fabricantes), por lo cual fijando $C_2=47\mu F$ y $R_2=1\Omega$, se tiene:

$$t_2 = 0.693(47 \times 10^{-6})(1) = 32.571 \mu s > 250 \text{ ns} > 150 \text{ ns}$$

por lo cual cumple muy bien con ambas condiciones.

El lapso t_3 será el tiempo que el contador mantenga fijo el conteo en los despliegues para que el operador pueda verlo, este tiempo se puede fijar alrededor de 2 segundos, por lo cual, dado $C_2=47\mu F$, tendremos:

$$R_3 = \frac{2}{(0.693 \times 47 \times 10^{-6})} - 1 = 61.403 \text{ k}\Omega$$

utilizando un valor de $R_3 = 68 \text{ k}\Omega$ tendremos un valor de:

$$t_3 = 0.693 \times 47 \times 10^{-6} \times 68.001 \times 10^3 = 2.215 \text{ seg}$$

el cual es suficiente para que el operador aprecie el valor desplegado. En la fig. 21 se aprecian los tiempos del tacómetro.

De esta manera, obtendremos una medición de velocidad angular en revoluciones por segundo, con una precisión ideal (sin considerar variaciones en las resistencias, por ejemplo) de ± 1 rps, es decir, $\pm 1/60$ (0.02) rpm, a todas luces muy superior al tacómetro mecánico original (ya que la precisión del tacómetro digital no depende de una escala).

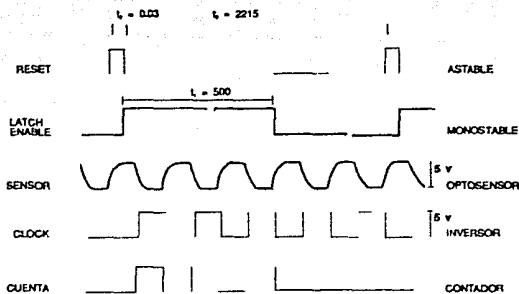


Fig. 21 Diagrama de tiempos del tacómetro.

En nuestro caso particular, se ha determinado experimentalmente que el motor nunca gira a más de alrededor de 6,000 rpm (al ser alimentado con 9 volts o menos), es decir 100 rps, por lo cual sólo se utilizarán 3 de los 4 despliegues de 7 segmentos, que podrían medir sin dificultad hasta 999 rps (999 Hz < 2MHz — límite fijado por el fabricante del chip), es decir casi hasta 60,000 rpm, cantidad muy superior a las que encontraremos en nuestro proyecto. Por otro lado, para una aplicación en tales frecuencias sería indispensable colocar un optosensor con mayor sensibilidad.

El módulo de circuito que contiene al tacómetro digital lleva su propio regulador de voltaje de +5v (7805), de modo que no depende de regulación externa.

AMPLIFICADOR DE POTENCIA

Como amplificador de potencia para impulsar la planta se utilizó un arreglo superalfa (Darlington) que consiste de dos transistores, conectados en configuración emisor seguidor, es decir sin resistores en el colector. De este modo, el potencial en el emisor de TR₂ sigue al potencial de entrada muy cercanamente, pero produce una alta ganancia de corriente.

Esta configuración es típica como elemento de salida, sobre todo para plantas que requieren de voltaje unipolar de alimentación. {Chesmond, 100,104}.

Como se ve en la fig. 22, el emisor de TR₁ (BC108) está conectado a la base de TR₂ (2N3055), o sea que la corriente de emisor de TR₁ se convierte en la corriente de base de TR₂, por esto, la ganancia de corriente del arreglo es muy alta, en nuestro caso, ambos transistores tienen una ganancia de alrededor de 200, por lo que la ganancia total de corriente es de 40,000. La diferencia consiste en que TR₁ solo puede manejar 100 mA mientras que TR₂ puede conducir hasta 15A, por lo cual está montado en un disipador de calor. {Sladdin, 145,154}

El diodo D1 (1N4004) se incluye para prevenir que altos potenciales inversos (fuerza contraelectromotriz) provenientes de los devanados del motor dañen al transistor. {Chesmond, 128}.

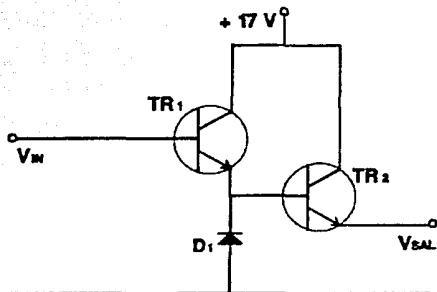


Fig. 22 Amplificador de potencia.

CONVERTIDOR DIGITAL — ANALÓGICO

Para convertir la salida digital de la computadora a un valor analógico hacia el actuador se usa un convertidor DAC0831, conectado en configuración de conversión continua, el cual recibe su entrada digital del puerto paralelo Centronics de la computadora, su salida es en forma de corriente analógica.

Para hacer esto, se conectan a tierra los pines CS (chip select), XFER, WR₁ y WR₂ del convertidor, y a Vcc la pata ILE (input latch enable). Por otra parte, la señal Strobe del puerto Centronics, se puentea de regreso como Acknowledge (ACK), para habilitar permanentemente la transmisión de datos. Los ocho bits del puerto paralelo se conectan directamente a las entradas D0 a D7 del convertidor.

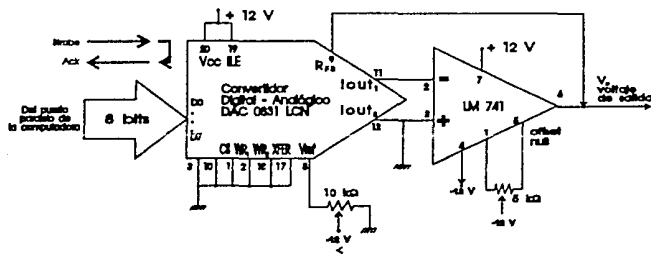


Fig. 23 Convertidor Digital Analógico.

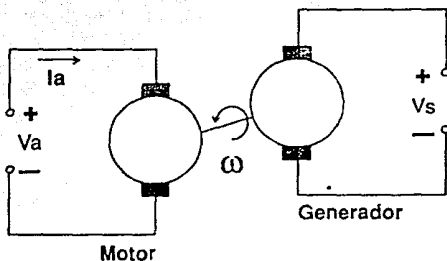


Fig. 24 Motor de corriente directa.

Dicha corriente entra a un amplificador operacional LM741, de propósito general, que convierte de corriente a voltaje de excitación para el amplificador de potencia. Este LM741 tiene control de offset null con un potenciómetro de precisión de 5 kohms. Este tipo de etapa tiene muchas ventajas, como alta ganancia, alta impedancia de entrada, baja impedancia de salida, etc. {Chesmond, 102}.

El convertidor está alimentado por +12v y -12v, regulados en forma local (LM7812 y LM7912), con una referencia negativa de potencial, ajustable con un potenciómetro de precisión de 10 kohms para lograr una alimentación al motor (en el punto Vi) de +9.00 volts a plena escala. Ver fig. 23.

MOTOR DE CORRIENTE DIRECTA

Como elemento esencial de la planta se usó un pequeño motor de corriente directa, con excitación independiente (magneto permanente). Este tipo de motores se utiliza en aplicaciones industriales cuando se requiere de una alta linealidad, donde el resto del equipo funciona también con corriente directa, y cuando se desea tener un alto par de arranque. Ver fig. 24.

Faraday fué el precursor en el diseño de este tipo de máquinas rotatorias, y en la actualidad son elementos típicos en equipos de cinta magnética, graficadores, copiadoras, manejadores de disco flexible, etc.

Los motores de CD se prestan con facilidad al control de velocidad, dadas sus características altamente lineales en un rango muy extenso de voltaje de armadura. {Electrocraft, 2-13}.

En nuestro caso, el motor de CD usado es de 12 volts de alimentación nominal, con una flecha de 9.5 cms de longitud, montada en un rotor (armadura) de 7 devanados, conectados a un conmutador de 7 placas, con dos escobillas.

El campo está formado por un imán permanente de dos polos. Este es el típico motor usado en servosistemas, con una relación flecha/diámetro > 1 . Dado que el campo es permanente, solo es posible controlar este motor por su armadura. En este caso, la potencia que se suministra a la

armadura debe ser relativamente alta, si bien se obtiene un par constante independiente de la velocidad (solo es función de la corriente de armadura). {Chesmond, 127,130}.

La función de transferencia del motor, controlado por armadura, puede representarse en su forma mas simple por la ecuación: {Neff, 315}.

$$\frac{\Omega(s)}{V_i(s)} = \frac{K_1}{\tau_1 s + 1} \quad (4)$$

Donde $\Omega(s)$ es la velocidad angular del motor y $V_i(s)$ es el voltaje de entrada. Aquí ambos se presentan en el dominio complejo s .

TACOGENERADOR DE CD

Como transductor de velocidad angular del motor se utilizó un pequeño tacogenerador marca Microswitch, acoplado a la flecha del motor, el cual es también una máquina electromagnética rotatoria, excitada en forma independiente con un imán permanente.

La función del tacogenerador es convertir la velocidad angular del motor en una señal eléctrica, la cual es más fácil de manejar, amplificar, etc. {Holman, 30}. Casi siempre el tacogenerador se coloca en el lazo de realimentación.

La teoría básica de los generadores es esencialmente la misma que para los motores, de hecho su función es justamente la inversa. Como inconvenientes, tiene la baja confiabilidad y el rizo producidos por el conmutador y escobillas. {Chesmond, 44}

Este tipo de generadores produce un voltaje proporcional a una velocidad angular

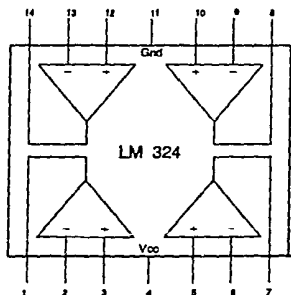


Fig. 25 Circuito LM324

(normalmente en el rango de 3 a 7 volts/1000 rpm) y es frecuentemente utilizado en las aplicaciones de servosistemas de lazo cerrado, cuando se quiere usar la realimentación de velocidad, también llamada realimentación tacométrica. En nuestro caso el tacogenerador produce 4.5 volts a plena escala (es decir, cuando se aplican 9 V en el devanado de armadura del motor).

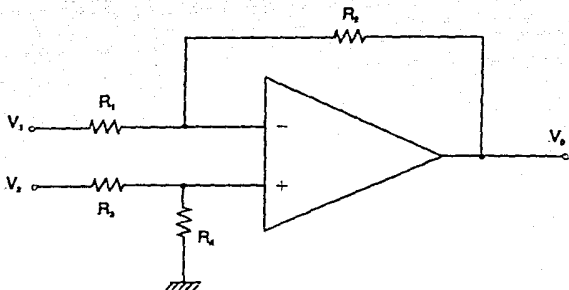


Fig. 26 Amplificador de instrumentación.

En este caso, la función de transferencia del tacogenerador es: {Neff, 315}.

$$H(s) = \frac{V(s)}{\Omega(s)} = K \quad (5)$$

Es decir que la relación de transformación de velocidad angular a voltaje de salida es constante.

AMPLIFICADORES DE INSTRUMENTACION

Para capturar la información de velocidad del motor, producida por el tacogenerador analógico acoplado a éste, se usan dispositivos de medición formados por amplificadores operacionales integrados en el circuito modelo LM324 (fig. 25), que contiene cuatro amplificadores operacionales de propósito general.

Este tipo de amplificador operacional, conectado como amplificador de instrumentación (fig.26), presenta una alta impedancia de entrada y una buena razón de rechazo en modo común, dependiendo ambas de los valores y tolerancias de las resistencias. Esta configuración es la más común para interconectar transductores a elementos de control o realimentación. {Millman, 574}.

En nuestro caso, los amplificadores se conectan en configuración no inversora, para que el potencial de salida no cambie de signo, ya que de allí se alimentará al convertidor analógico digital de la computadora, que solo admite voltajes positivos. El cable que se usó para la conexión a la computadora se muestra en la fig. 27.

La ecuación general para calcular la ganancia de este tipo de amplificador de instrumentación es {Millman, 574}:

$$V_0 = \frac{(R_1+R_2)R_4}{(R_3+R_4)R_1}V_2 - \frac{R_2}{R_1}V_1 \quad (6)$$

En nuestro caso particular, usaremos $R_1 = R_3$ y $R_2 = R_4$, de modo que la ecuación (6) (idealmente) se reduciría a:

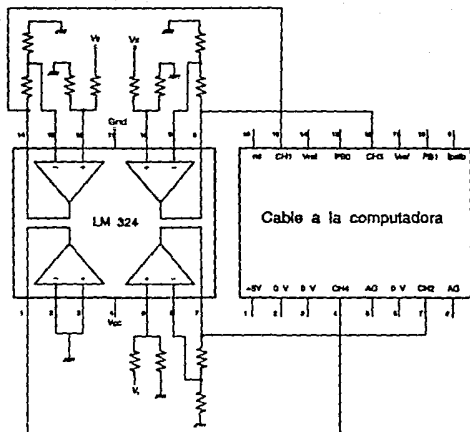


Fig. 27 Cable de los amplificadores a la computadora.

$$V_0 = \frac{R_2}{R_1}(V_2 - V_1) \quad (7)$$

por lo que este tipo de amplificador también se llama amplificador diferencial o de diferencias.

CALCULO DE DESVIACIONES

Si llamamos e_i al valor de la desviación real de cada resistencia, en lugar de las resistencias originales se substituyen en la ec. (6) los valores:

$$R_1 (1 + e_1)$$

$$R_2 (1 + e_2)$$

$$R_3 (1 + e_3)$$

$$R_4 (1 + e_4)$$

Donde $R_1 = R_3$ y $R_2 = R_4$, aunque no necesariamente $e_1 = e_3$ ni $e_2 = e_4$. Por lo tanto la ec. (6) queda:

$$V_{0e} = \frac{R_1(1+e_1)+R_2(1+e_2)}{R_1(1+e_3)+R_2(1+e_4)} \frac{R_2(1+e_4)}{R_1(1+e_1)} V_2 - \frac{R_2(1+e_2)}{R_1(1+e_1)} V_1 \quad (8)$$

Con resistencias cuya tolerancia sea de 5% (estándar comercial), así como los voltajes de entrada, se podrán calcular las posibles desviaciones máximas de tolerancia a la salida (para la

entrada al CAD de la computadora). Una primera aproximación de 10% es razonable, como se verá mas adelante.

Se conectarán los amplificadores en su configuración no inversora, por ser necesario que el potencial de salida con que se alimenta al convertidor A/D no cambie de signo.

Por este motivo, y considerando que en todos los casos se tendrá el voltaje $V_1 = 0$ (conectado a tierra), nuestra ecuación se simplifica, y la variación máxima se calcularía con la fórmula anterior, de la siguiente manera:

$$V_{0e} = \frac{R_1(1+e_1)+R_2(1+e_2)}{R_1(1+e_3)+R_2(1+e_4)} \frac{R_2(1+e_4)}{R_1(1+e_1)} V_2 \quad (9)$$

es decir, llamando a la ganancia real $A = V_{0e}/V_2$ y a la ganancia ideal $A_0 = R_2/R_1$ (ambas siempre positivas), queda:

$$\frac{A}{A_0} = \frac{R_1(1+e_1)(1+e_4) + R_2(1+e_2)(1+e_4)}{R_1(1+e_3)(1+e_1) + R_2(1+e_4)(1+e_1)} \quad (10)$$

dado que cualquier $e \ll 1$, $e^2 = 0$ para cualquier par de e 's, o sea que, utilizando:

$$A_0 = R_2/R_1$$

podemos reorganizar como:

$$\frac{A}{A_0} = \frac{(1+e_1+e_4) + A_0(1+e_2+e_4)}{A_0(1+e_1+e_3) + (1+e_1+e_3)} \quad (11)$$

analizando esta relación, y considerando el peor de los casos ($e = \pm 5\%$), veremos que la máxima desviación positiva de A/A_0 se da cuando el numerador sea máximo y el denominador sea mínimo, por ello, observando primero la predominancia de e_4 en el numerador, vemos que debe ser $e_4 = +5\%$ (lo cual fija su valor también en el denominador) luego observando la predominancia de e_1 en el denominador vemos que debe ser $e_1 = -5\%$ (lo cual fija su valor también en el numerador); finalmente sólo queda asignar en el numerador $e_2 = +5\%$ y en el denominador $e_3 = -5\%$.

En resumen, la desviación A/A_0 será máxima cuando:

$$e_1 = e_3 = -5\%$$

$$e_2 = e_4 = +5\%$$

sustituyendo estos valores vemos que $(1+e_1+e_4) = 1$, $(1+e_2+e_4) = (1+2e)$ y $(1+e_1+e_3) = (1-2e)$, donde $e=0.05$

por razones justamente inversas, se deduce que la máxima desviación negativa de A/A_0 será cuando:

$$e_1 = e_3 = +5\%$$

$$e_2 = e_4 = -5\%$$

ahora tenemos que $(1+e_1+e_4) = 1$, $(1+e_2+e_4) = (1-2e)$ y $(1+e_1+e_3) = (1+2e)$, es decir que se obtienen las siguientes relaciones hiperbólicas:

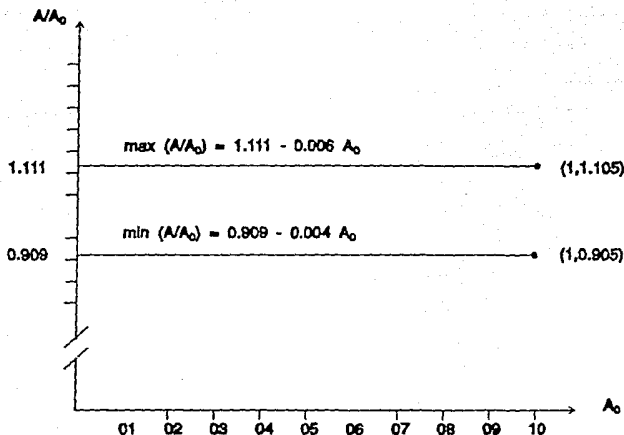


Fig. 28 Variación de ganancia A/A_0 en el amplificador.

$$\max \left(\frac{A}{A_0} \right) = \frac{1 + A_0(1+2e)}{A_0 + (1-2e)} \quad (12)$$

$$\min \left(\frac{A}{A_0} \right) = \frac{1 + A_0(1-2e)}{A_0 + (1+2e)} \quad (13)$$

es de notar que las desviaciones sólo dependen de la ganancia ideal A_0 (que por definición es positiva) y de las tolerancias de las resistencias e_i , mas no de los valores particulares de dichas resistencias R_i .

Con la relación anterior podemos construir la siguiente Tabla:

Ganancia	máx (A/A ₀)	mín (A/A ₀)
A ₀ → 0	1/(1-2ε)	1/(1+2ε)
	= 1.111 (+11.111 %)	= 0.909 (-9.091 %)
A ₀ = 1	(2+2ε)/(2-2ε)	(2-2ε)/(2+2ε)
	= 2.111.9 (+10.526 %)	= 1.9/2.1 (-9.524 %)
A ₀ → ∞	(1+2ε)	(1-2ε)
	= 1.1 (+10 %)	= 0.9 (-10 %)
Tabla 1. Variación de la ganancia A/A ₀ .		

También podemos apreciar que para valores de A₀ entre 0 y 1 la variación es prácticamente lineal, es decir que se puede aproximar por una recta para un rango tan pequeño de variación de ganancia; esto también se aprecia en la gráfica. Ver fig. 28.

Vemos que máx(A/A₀) pasa por los puntos (0,1.111) y (1,1.105) mientras que mín(A/A₀) pasa por (0,0.909) y (1,0.905); las rectas que podemos aproximar a estos datos son:

$$\max(A/A_0) = 1.111 - 0.006 A_0 \quad (14)$$

$$\min(A/A_0) = 0.909 - 0.004 A_0 \quad (15)$$

de aquí podemos ver que la diferencia entre ambas, en el peor de los casos (A₀=0) es de 0.202

También se obtendrán analítica y experimentalmente los valores exactos de los voltajes de salida, y con ello las desviaciones reales (ver programa Calcula).

Finalmente, la relación que proporciona el voltaje de salida del amplificador, en el peor de los casos de tolerancia, está dada por:

$$V_{\text{out}} = A_0 V_i \frac{1+A_0(1+2\varepsilon)}{A_0+(1-2\varepsilon)} \quad (16)$$

La forma mas sencilla de compensar dichas pequeñas diferencias que se presenten en los valores de salida, debidas a las tolerancias de los elementos, es mediante una calibración por software integrada al sistema de programas de computadora.

Esto se logrará alimentando el amplificador de instrumentación con un valor conocido y una precisión dada, y realizando una rutina de muestreo de los valores medidos por el CAD, los cuales se compararán contra el valor que se registra en un medidor externo ("exacto"). (Ver Procedimiento PROCcalibra en programa Menú).

Este valor externo es tomado por el operador e introducido mediante el teclado a la computadora, que calcula con ambos datos un factor de calibración para cada una de las 4 entradas del CAD, almacenándolas en un archivo en disco para su uso futuro.

VOLTAJES MAXIMOS

El siguiente procedimiento de escalamiento se hace para optimizar el rango de excursiones de voltaje a las entradas del convertidor analógico digital. (Chesmond, 318).

Dado que el Convertidor Analógico Digital (CAD) de la MicroBBC será quien reciba los valores entregados por los amplificadores de instrumentación, la salida de éstos deberá ser menor o igual que el potencial máximo de entrada (de referencia) del CAD, que es nominalmente de 1.8 volts.

Este valor será el que se use en todos los cálculos de ganancias. Hacer referencia a la figura 28.

A máxima velocidad, (peor de los casos), se tienen las siguientes condiciones (comprobadas experimentalmente):

$$V_i = 9.0 \text{ volts (voltaje de entrada)}$$

este será también el voltaje máximo que esperemos para V_x (ver figura anterior).

De los experimentos iniciales se encontró que en estas condiciones aparece en el tacómetro analógico un voltaje:

$$V_s = 4.10 \text{ volts (voltaje de salida)}$$

lo cual se usará para los siguientes cálculos.

CALCULO DE LAS GANANCIAS

Dado que todas las ganancias que manejarán nuestros amplificadores de instrumentación están entre 0 y 1, se sigue que las desviaciones teóricas máximas que podemos esperar en el peor de los casos son inferiores a +11.111% y -9.524%, de acuerdo a la Tabla calculada antes.

En la fig. 29 se muestra cómo se conectarán los amplificadores de instrumentación al motor y al tacogenerador, y con base en los valores máximos que se calculan a continuación, se encontrarán las ganancias que deberá tener cada amplificador.

Si observamos la fig. 29, se ve que el valor que asignemos a R_0 debe fijarse en un valor bajo para que su efecto de carga en el circuito del motor sea pequeño, pero a la vez que proporcione una caída de potencial suficiente para calcular la corriente que circula por ella. Por ello se ha seleccionado conectar allí una resistencia de 3.3 ohms.

Cuando el motor se encuentra girando a máxima velocidad se encontró que la caída de voltaje en R_0 es de:

$$V_r = V_i - V_s = 0.67 \text{ volts}$$

por lo que la potencia que disipa es:

$$P_x = V_r^2 / R_0 = (0.67)^2 / (3.3) = 0.136 \text{ watts}$$

se conectó, por lo tanto, una de 5 watts que está disponible comercialmente.

Ganancia A_{v1}

$$A_{v1} = V_{\text{motor}} / V_s = 1.8 / 4.10 = 0.439 = R_4 / R_3$$

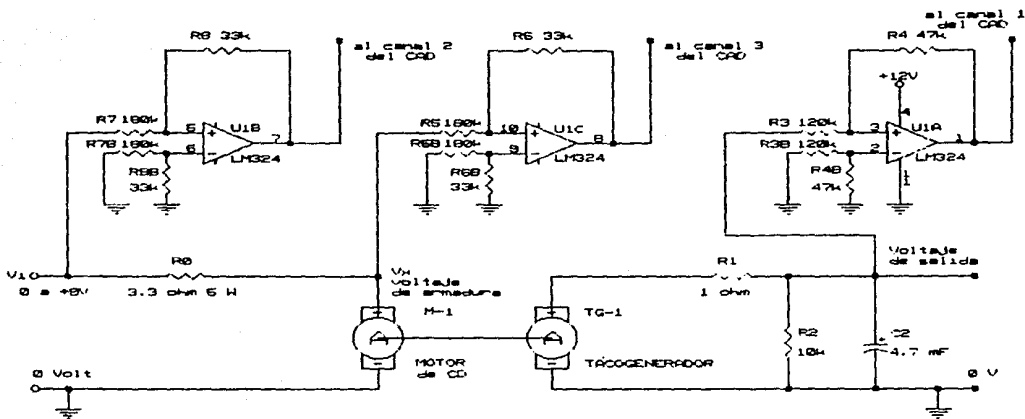


Fig 29. Conexión de los amplificadores a la planta.

Un par de resistencias que se aproxima a esta ganancia son:

$$R_1 = 47 \text{ k}\Omega, R_2 = 120 \text{ k}\Omega$$

con lo que:

$$A_{v1} = 0.3920 \text{ } (-10.78\%)$$

El voltaje ideal sería entonces $V_0 = 1.61$ volts, y con la fórmula de desviación (9) se obtienen la salida máxima de voltaje, que se presentaría cuando $e_1 = e_3 = -5\%$ y $e_2 = e_4 = +5\%$, y la salida mínima posible, cuando $e_1 = e_3 = +5\%$ y $e_2 = e_4 = -5\%$. Dado $V_2 = V_s$, Estos voltajes son: $1.78 \text{ v } (+10.87\%)$ y $1.46 \text{ v } (-9.33\%)$. Experimentalmente se midió que $V_0 = 1.60$ volts.

Ganancia A_{v2}

$$A_{v2} = V_{\text{max} \text{ salida}} / V_{\text{max}} = 1.8 / 9.0 = 0.200 = R_7 / R_8$$

Un par de resistencias que se aproxima a esta ganancia son:

$$R_7 = 33 \text{ k}\Omega, R_8 = 180 \text{ k}\Omega$$

con lo que:

$$A_{v2} = 0.183 \text{ } (-8.33\%)$$

El voltaje ideal sería entonces $V_0 = 1.65$ volts, y con la fórmula (9) se obtienen las salidas máxima y mínima de voltaje, igual que arriba, con $V_2 = V_{\text{max}}$. Estos voltajes son: $1.83 \text{ V } (+10.9\%)$ y $1.50 \text{ V } (-9.2\%)$. Experimentalmente se midió que $V_0 = 1.64$ volts.

Ganancia A_{v3}

Si bien V_x no es exactamente igual que V_i , la diferencia es pequeña, por lo que A_{v3} se hace igual que A_{v2} por simplicidad, es decir:

$$A_{v3} = A_{v2} = 1.8 / 8.33 = 0.216 = R_9 / R_{10}$$

$$R_9 = 33 \text{ k}\Omega, R_{10} = 180 \text{ k}\Omega$$

con lo que

$$A_{v3} = 0.183 \text{ } (-8.33\%)$$

El voltaje ideal aquí sería $V_0 = 1.53$ volts, y de este modo, los voltajes máximo y mínimo son: $1.69 \text{ V } (+10.92\%)$ y $1.39 \text{ V } (-9.22\%)$.

En la computadora se puede efectuar la resta de $V_x - V_i$ para dividir entre R_0 y obtener la corriente de entrada si se desea (integrado en el programa de captura).

RESUMEN

En este capítulo se ha citado la historia, desarrollo y descripción de la MicroBBC, así como sus accesorios, como son el convertidor analógico digital, el puerto paralelo, la fuente de poder para la planta, el motor de CD (planta), el tacogenerador de CD, el tacómetro digital, el convertidor digital analógico, etc.

Se diseñaron las interfases entre la computadora y la planta: el amplificador de potencia (actuador), los amplificadores de instrumentación, así como sus parámetros de interés, como son principalmente ganancias y voltajes.

§

3

OBTENCION EXPERIMENTAL Y ANALISIS DEL MODELO

FUNCION DE TRANSFERENCIA

La función de transferencia de una planta cualquiera puede hallarse básicamente de tres maneras: la primera de ellas sería con base en el conocimiento de los parámetros de cada uno de los bloques, así como las relaciones matemáticas que rigen las interacciones entre ellos; los parámetros más importantes serían las constantes de tiempo de cada bloque, así como sus ganancias respectivas.

Sin embargo, este primer método es más bien impráctico, ya que implica la medición experimental o el cálculo de muchos parámetros para cada bloque, que al final de cuentas, al combinarse, implicarían una serie de errores acumulativos, de los muchos tipos que contempla la metrología: truncamiento, aproximación, errores humanos, etc., es decir que la función de transferencia encontrada de esta manera sería una aproximación muy pobre a la planta real.

El segundo método de obtención de una función de transferencia, es la experimentación directa sobre la totalidad de la planta, considerada como una caja negra, a la cual se le aplica una excitación senoidal, de amplitud constante y conocida, haciendo variar la frecuencia de dicha excitación en el rango de operación de la planta, y registrándose la respuesta de la misma, para hacer luego un análisis de respuesta a la frecuencia. {Franklin et al., 74}.

El tercer método, que es el que mejor se adecua a nuestro caso, es también la experimentación directa, pero utilizando una excitación en escalón unitario, registrándose en el tiempo su respuesta (transitoria) al mismo. Este método es fácilmente aplicable dada la facilidad con que puede obtenerse dicha respuesta transitoria. {Franklin et al., 69,70}.

En el caso de este último método experimental, finalmente se aplican métodos estadísticos (ajuste de curvas) para obtener la ecuación que correctamente describe el funcionamiento de la planta como respuesta al escalón. De allí es fácil pasar a su función de transferencia.

Una nota precautoria es que debe tenerse una muy buena relación de señal a ruido, ya que la localización de polos por este método es muy sensible a la calidad de los datos originales. (Franklin et al., 74).

De cualquier modo es necesario un conocimiento teórico de la forma algebraica que tiene cada bloque en su respectiva función de transferencia, o al menos una idea general sobre el tipo de respuesta que puede esperarse del sistema en su conjunto.

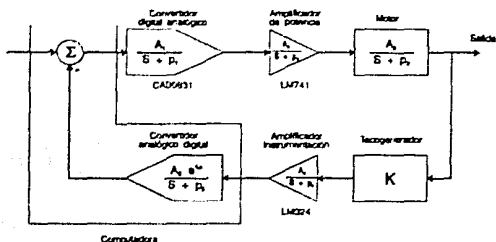


Fig. 30 Polos y ceros de la planta real.

FUNCION TEORICA

Como punto de partida, sabemos que el tacómetro puede aproximarse por una constante, el motor de CD se puede aproximar por un polo simple. En sentido estricto, el subsistema motor-tacogenerador-carga tiene una función de transferencia con tres ceros y seis polos, si se consideran efectos térmicos, resonancia torsional y otras características. (Electrocraft, 2.3.80).

De la misma manera, el amplificador de instrumentación y el amplificador de potencia se pueden aproximar por un polo cada uno.

El convertidor analógico digital se puede aproximar como un sostenedor de orden cero (muestreador), que produce un retraso en el tiempo y un polo, mientras que el convertidor digital analógico, a causa de su amplificador operacional de salida, podría aproximarse por un polo. Ver fig. 30.

Así las cosas, podríamos esperar una planta cuya función de transferencia teórica es de orden superior y tendría al menos unos cinco polos (haciendo algunas simplificaciones). Ver fig. 31.

FUNCION PRACTICA

Para propósitos prácticos, veremos que al menos dos de dichos polos son prácticamente imperceptibles en los datos experimentales, y se puede hacer una aproximación muy buena con únicamente tres polos, o bien como un retraso en el tiempo y un polo simple.

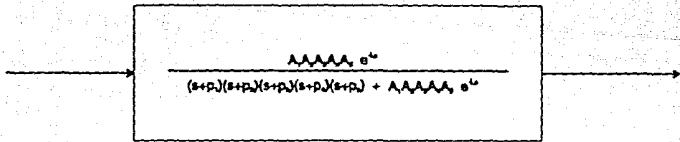


Fig. 31 Simplificación de polos y ceros de la planta.

La obtención de la función de transferencia en forma experimental a través de la respuesta transitoria de la planta se hace con el programa de captura de datos, y su análisis y transformación con los programas para traza de gráficas, transformación de datos y cálculo de parámetros.

En el caso de la aproximación por tres polos, se aplica el método logarítmico. Para dos polos puede usarse éste o bien el método diferencial de Lipka, descritos más adelante, mientras que para la aproximación por retraso y polo se utilizan las reglas de Ziegler y Nichols para curvas de reacción.

Si se deseara, se podrían comparar y combinar los tres métodos anteriores para obtener distintas aproximaciones a la función de transferencia.

METODO LOGARITMICO

Este método permite calcular, a partir de datos experimentales (ver fig. 32), los coeficientes y exponentes de una función con términos exponenciales que se supone del tipo:

$$w(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} + A_3 e^{-\frac{t}{T_3}} \quad (17)$$

considerando que los coeficientes A_0, A_1, A_2, A_3 , y las constantes de tiempo T_1, T_2 y T_3 son números reales, que además se asignan tal que $T_1 > T_2 > T_3$ (idealmente, las diferencias deben ser grandes (D'azzo, Houpis, 662)).

Este caso es el que se presenta en la respuesta transitoria de un sistema de tercer orden como el que queremos aproximar, si bien puede extenderse a sistemas de orden mayor. En un sistema así:

$$W(t) = W_0 - w(t) \quad (18)$$

sería la respuesta total del sistema.

Si se graficara $(t, \ln w(t))$ podríamos observar que la parte final de esta traza (en sentido estricto para $t > T_1$) se aproxima a una recta. Esto se debe a que si $T_1 > T_2 > T_3$, entonces $e^{-\frac{t}{T_1}}$ y

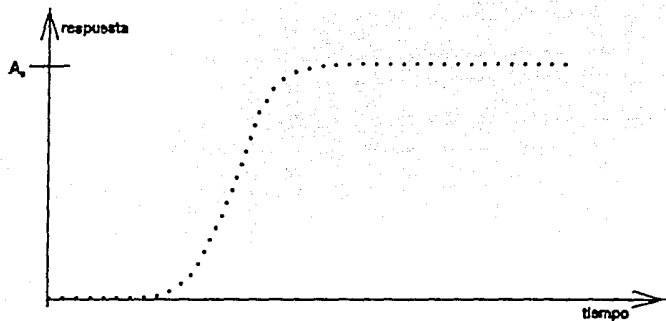


Fig. 32 Respuesta experimental de un sistema.

$e^{-\frac{t}{T_2}}$ decaen mas rápidamente y tienden a cero en esa región. En la práctica podríamos tomar aproximadamente el último tercio de los datos. Ver fig. 33. {D'azzo y Houpis (FCS), 662}.

Por ese motivo, es posible ajustar una recta (p.ej. por el método de mínimos cuadrados) de ecuación $y = mx + b$ a ese subconjunto de los datos, en la cual podemos identificar

$$y = LN w(t), x = t$$

es decir:

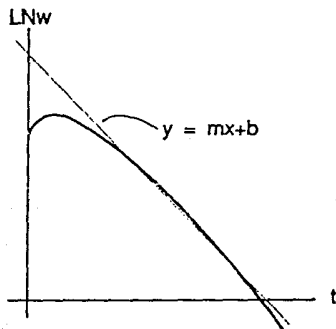


Fig. 33 Ajuste de recta sobre LN(w).

$$LN w(t) = mt + b$$

y dado que el término allí representado es:

$$w(t) = A_1 e^{-\frac{t}{T_1}}$$

$$LN w(t) = LN(A_1) - t/T_1$$

tenemos que:

$$A_1 = e^b, T_1 = -\frac{1}{m}$$

con lo cual podemos acceder analíticamente al término $A_1 e^{-\frac{t}{T_1}}$, para modificar nuestros datos originales de manera que:

$$v(t) = w(t) - A_1 e^{-\frac{t}{T_1}} = A_2 e^{-\frac{t}{T_2}} + A_3 e^{-\frac{t}{T_3}} \quad (19)$$

de modo que si ahora graficamos $LN v(t)$ versus t , podremos ajustar otra recta, $y = px + q$ (ahora para $T_2 < t < T_3$), es decir el último tercio de este nuevo trazo, de donde tendremos:

$$A_2 = e^q, T_2 = -\frac{1}{p}$$

Luego se repetiría el mismo procedimiento para A_3 y T_3 , es decir, hacer:

$$x(t) = w(t) - A_2 e^{-\frac{t}{T_2}} - A_1 e^{-\frac{t}{T_1}} = A_3 e^{-\frac{t}{T_3}} \quad (20)$$

para luego graficar $x(t)$, ajustar una recta y hallar A_3 y T_3 , con lo cual podremos completar nuestra ecuación original (17):

$$w(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} + A_3 e^{-\frac{t}{T_3}}$$

METODO DE LIPKA

Este método, sugerido por J. Lipka {Lipka, 194} se usa para hallar una aproximación (ajuste) de un conjunto de datos obtenidos experimentalmente (ver fig. 32), por medio de una ecuación con dos términos exponenciales (dos polos) y un término constante, basándose en el hecho de que el trazo de una ecuación exponencial en papel semilogarítmico se aproxima a una recta. Este método es analíticamente muy exacto, como se verá en su desarrollo teórico, pero hay que hacer notar que es muy sensible a la calidad de los datos originales. Para que este método funcione con corrección, los polos deben estar separados al menos por un orden de magnitud uno del otro.

Primeramente se hará el desarrollo teórico y luego se indicará como se aplica en la práctica.

Este método se incorporó en el programa de Gráficas del sistema, ya que es claramente adecuado para ajustar un conjunto de datos obtenidos a partir de una planta cuyo comportamiento transitorio ante una entrada escalón suponemos se apega, al menos en forma aproximada, a la siguiente ecuación:

$$W(t) = W_0 - w(t)$$

donde W_0 es el valor final (de estado estable) y:

$$w(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} \quad (21)$$

Suponiendo que nuestros datos se encuentran igualmente espaciados para los valores de la variable t , con incrementos cuyo valor sea h , hacemos las transformaciones:

$$t = t + h$$

obtenemos:

$$w'(t) = A_1 e^{-\frac{(t+h)}{T_1}} + A_2 e^{-\frac{(t+h)}{T_2}} \quad (22)$$

luego con:

$$t = t + 2h$$

queda:

$$\begin{aligned} w''(t) &= A_1 e^{-\frac{(t+2h)}{T_1}} + A_2 e^{-\frac{(t+2h)}{T_2}} \\ &= A_1 e^{-\frac{t}{T_1}} e^{-\frac{2h}{T_1}} + A_2 e^{-\frac{t}{T_2}} e^{-\frac{2h}{T_2}} \end{aligned} \quad (23)$$

en la ec. (21) multiplicando por la constante:

$$e^{-h(\frac{1}{T_1} + \frac{1}{T_2})}$$

queda:

$$w e^{-h(\frac{1}{T_1} + \frac{1}{T_2})} = A_1 e^{-\frac{t}{T_1}} e^{-h(\frac{1}{T_1} + \frac{1}{T_2})} + A_2 e^{-\frac{t}{T_2}} e^{-h(\frac{1}{T_1} + \frac{1}{T_2})}$$

y multiplicando la ec. (22) por otra constante:

$$-(e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}})$$

queda:

$$\begin{aligned} & -w'(e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}}) = \\ & = -A_1 e^{-\frac{t}{T_1}} e^{-\frac{h}{T_1}} (e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}}) - A_2 e^{-\frac{t}{T_2}} e^{-\frac{h}{T_2}} (e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}}) \\ & = -A_1 e^{-\frac{t}{T_1}} e^{-\frac{h}{T_1}} e^{-\frac{h}{T_1}} - A_1 e^{-\frac{t}{T_1}} e^{-\frac{h}{T_1}} e^{-\frac{h}{T_2}} + \\ & \quad -A_2 e^{-\frac{t}{T_2}} e^{-\frac{h}{T_2}} e^{-\frac{h}{T_1}} - A_2 e^{-\frac{t}{T_2}} e^{-\frac{h}{T_2}} e^{-\frac{h}{T_2}} \end{aligned}$$

que simplificada queda:

$$\begin{aligned} & = -A_1 e^{-\frac{t}{T_1}} e^{-\frac{2h}{T_1}} - A_1 e^{-\frac{t}{T_1}} e^{-h(\frac{1}{T_1} + \frac{1}{T_2})} + \\ & \quad -A_2 e^{-\frac{t}{T_2}} e^{-h(\frac{1}{T_1} + \frac{1}{T_2})} - A_2 e^{-\frac{t}{T_2}} e^{-\frac{2h}{T_2}} \end{aligned}$$

y al reagrupar:

$$= -e^{-k(\frac{1}{T_1} + \frac{1}{T_2})} \left(A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} \right) - \left(A_1 e^{-\frac{t}{T_1}} e^{-\frac{2h}{T_1}} + A_2 e^{-\frac{t}{T_2}} e^{-\frac{2h}{T_2}} \right)$$

al referirnos a (21) y (23) podemos poner la expresión como:

$$-w' \left(e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}} \right) = -we^{-k(\frac{1}{T_1} + \frac{1}{T_2})} - w''$$

o bien:

$$w'' = w' \left(e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}} \right) - we^{-k(\frac{1}{T_1} + \frac{1}{T_2})}$$

y al dividir entre w :

$$\frac{w''}{w} = \frac{w'}{w} \left(e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}} \right) - e^{-k(\frac{1}{T_1} + \frac{1}{T_2})} \quad (24)$$

La cual es una ecuación en diferencias, de segundo orden, si recordamos que $h = \Delta t$, $w' = w(t+h)$ y $w'' = w(t+2h)$. Al graficar esta relación, se aproximará a una recta de la forma:

$$y = px + q$$

donde:

$$y = w'' / w$$

son las ordenadas de la recta, mientras que:

$$x = w' / w$$

son las abscisas de la misma;

$$p = e^{-\frac{h}{T_1}} + e^{-\frac{h}{T_2}} \quad (25)$$

es la pendiente de la recta y:

$$q = -e^{-k(\frac{1}{T_1} + \frac{1}{T_2})} \quad (26)$$

es el intercepto al origen. Despejando de (26) queda:

$$\begin{aligned} -h \left(\frac{1}{T_1} + \frac{1}{T_2} \right) &= LN(-q) \\ -\frac{1}{T_1} &= \left(\frac{1}{h} \right) LN(-q) + \frac{1}{T_2} \end{aligned}$$

sustituyendo en (25) y reordenando:

$$\begin{aligned} p &= e^{k \frac{1}{h} LN(-q) - \frac{1}{T_1}} + e^{-\frac{h}{T_2}} \\ &= e^{LN(-q) - \frac{h}{T_1}} + e^{-\frac{h}{T_2}} \\ &= e^{LN(-q)} e^{\frac{h}{T_1}} + e^{-\frac{h}{T_2}} \end{aligned}$$

$$= -qe^{\frac{h}{T_1}} + e^{-\frac{h}{T_1}}$$

al multiplicarlo por $e^{-\frac{h}{T_2}}$ queda:

$$pe^{-\frac{h}{T_1}} = -q + e^{-\frac{2h}{T_1}}$$

es decir:

$$\left(e^{-\frac{h}{T_1}}\right)^2 - p\left(e^{-\frac{h}{T_1}}\right) - q = 0 \quad (27)$$

la cual es una ecuación cuadrática en $e^{-\frac{h}{T_1}}$, por lo tanto:

$$e^{-\frac{h}{T_1}} = \frac{p \pm \sqrt{p^2 - 4(-q)}}{2}$$

de allí que los polos son:

$$\boxed{-\frac{1}{T_1}, -\frac{1}{T_2} = \frac{1}{h} \text{LN} \left(\frac{p \pm \sqrt{p^2 + 4q}}{2} \right)} \quad (28)$$

si hacemos del término:

$$p^2 + 4q = r^2$$

para que el logaritmo natural de esa expresión exista como número real, se necesita que:

$$p - \sqrt{p^2 + 4q} > 0$$

$$p^2 > p^2 + 4q$$

$$q < 0$$

es decir que el intercepto al origen sea negativo. De allí que, para que los polos sean números reales:

$$\text{si } (p^2 + 4q) > 0$$

$$-\frac{p^2}{4} < q < 0$$

para que $p \neq r > 0$

NOTA: Esto no implica que una planta cualquiera no pueda tener un par de polos complejos conjugados, simplemente significa que tal caso no concuerda con lo supuesto por nosotros en este momento, ya que nuestra planta no muestra, a través de los datos experimentales, ninguna oscilación, la cual debería estar presente si los polos tuvieran una parte imaginaria.

Ahora, al conocer $-1/T_1$ y $-1/T_2$, los asignaremos de tal modo que $T_1 > T_2$ o sea que $(-1/T_2 + 1/T_1) < 0$, y hacemos uso de la ec. (21), multiplicando por $e^{\frac{t}{T_1}}$

$$\boxed{w e^{\frac{t}{T_1}} = A_1 + A_2 e^{(-\frac{1}{T_1} + \frac{1}{T_2})t}} \quad (29)$$

la cual al graficarse, de nuevo es de la forma:

$$y = mx + n$$

directamente con:

$$y = w(t) e^{\frac{t}{T_1}}, \quad x = e^{(-\frac{1}{T_1} + \frac{1}{T_2})t}$$

$$m = A_2, \quad n = A_1$$

lo cual completa la ecuación que buscamos.

De esta manera, lo que debe hacerse para encontrar los coeficientes y los exponentes que son las incógnitas de nuestra ecuación (21), deben graficarse en un sistema coordenado cartesiano los pares de datos:

$$\left(\frac{w'}{w}, \frac{w''}{w} \right)$$

donde w es el valor experimental de la función, mientras que w' es el valor en $t+h$, y w'' es el valor en $t+2h$.

Una vez hecha la gráfica, se hace un ajuste (p.ej. por el método de mínimos cuadrados) que aproxime una recta a ese conjunto de datos, obteniéndose un valor para la pendiente p y para el intercepto al origen q .

Se calcula $p^2 + 4q$, y conociendo el valor del incremento constante h , se obtienen los valores de $-1/T_1$ y $-1/T_2$ con la igualdad (28).

Conociendo $1/T_1$ y $1/T_2$ se calcula y se grafica la ecuación (29), que también se aproxima a una recta, se hace un ajuste de la misma y se obtiene un nuevo par pendiente e intercepto, de donde se obtienen los coeficientes A_1 y A_2 de nuestra función original, con los signos adecuados.

METODO DE ZIEGLER Y NICHOLS

Este método fué diseñado en los 50's por Ziegler y Nichols, basándose en trabajos previos (1936) de Callender, Hartree y Porter {Franklin et al., 69}.

Ziegler y Nichols analizaron plantas industriales reales (normalmente de orden superior) y observaron que su respuesta a una entrada de escalón unitario es una curva de tipo sigmoide, que llamaron curva de reacción, (ver fig. 34) a la cual se le puede ajustar una función de transferencia básica con una ecuación de la forma:

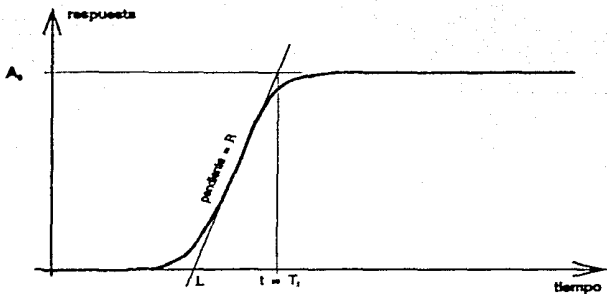


Fig. 34 Curva de reacción de Ziegler y Nichols.

$$G(s) = \frac{Re^{-Ls}}{T_1s+1} \quad (30)$$

es decir un retraso en el tiempo (también llamado retraso de transporte) de magnitud L , con un polo simple de valor $1/T_1$ y con una ganancia R/T_1 . La equivalente función de respuesta transitoria en el tiempo sería:

$$g(t) = \left(\frac{R}{T_1}\right) u(t-L) e^{-\frac{(t-L)}{T_1}} \quad (31)$$

donde $u(t-L)$ es un escalón unitario retrasado L unidades, es decir que la función es una exponencial simple retrasada L unidades en el tiempo. De aquí la respuesta total al escalón sería:

$$w(t) = W_o - g(t)$$

o sea:

$$w(t) = \left(\frac{R}{T_1}\right) u(t-L) \left(1 - e^{-\frac{(t-L)}{T_1}}\right) \quad (32)$$

Para calcular los parámetros R , L y T_1 se utiliza la gráfica de la respuesta de la planta, sobre la cual se traza una recta que pase por el punto de inflexión, es decir que es la recta de mayor pendiente que se pueda trazar sobre dicha respuesta (ver figura 34). {Alvarez y Alvarez, 2; Takahashi et al., 343, 369; Chesmond, 398}.

La pendiente de esta recta es llamada R , el punto donde la recta cruza el eje del tiempo es el retraso L , y la abscisa del punto donde cruza con el valor final (de estado estable) de la respuesta es la constante de tiempo T_1 .

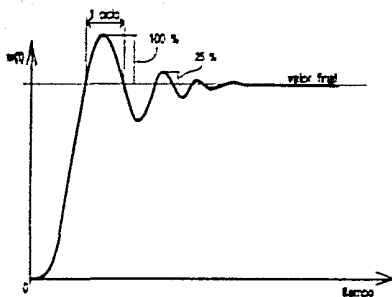


Fig. 35 Respuesta ideal de un sistema usando control PID.

Con estos parámetros R , L y T_1 también se podrán diseñar controladores PID en forma aproximada, como se verá más adelante, para que un sistema que utilice tal controlador tenga una respuesta óptima, de acuerdo a la definición de Ziegler y Nichols. Ver fig. 35. También es posible añadir más polos a este modelo si se desea, o bien combinar distintos métodos. (Franklin et al., 105).

OBTENCION DEL MODELO DE LA PLANTA

A continuación se describen los procedimientos seguidos para obtener el modelo de la planta (a través de su respuesta al escalón) iniciando con la conexión del equipo, la simulación y luego el cálculo de incógnitas. Ver fig. 36.

CONEXION DEL EQUIPO

Se conecta la fuente de poder (transformadores y puentes rectificadores) cuya salida deberá dar ± 17 volts aproximadamente (no regulada). Esta fuente tiene una salida de 1 Ampere y una de 2 Amperes. Ver figura 37.

A la salida de 1 Ampere se conecta el regulador de -12 volts (LM7912) que se utiliza como referencia en el circuito convertidor digital analógico, y como ajuste de salida nula en el amplificador operacional LM741.

A la salida de 2 Amperes se conecta el regulador de $+12$ volts (LM7812) que va a los circuitos. Esta salida deberá dar $+17$ volts aproximadamente cuando se encienda la fuente de poder. Aquí se conecta también la alimentación del amplificador de potencia.

Se conecta el cable que entra a la MicroBBC al puerto de cassette (DIN de 5 pines, usa solo 2 hilos), por un lado al borne negativo de la fuente (0 volts) y por el otro al borne negativo general de los circuitos y motor.

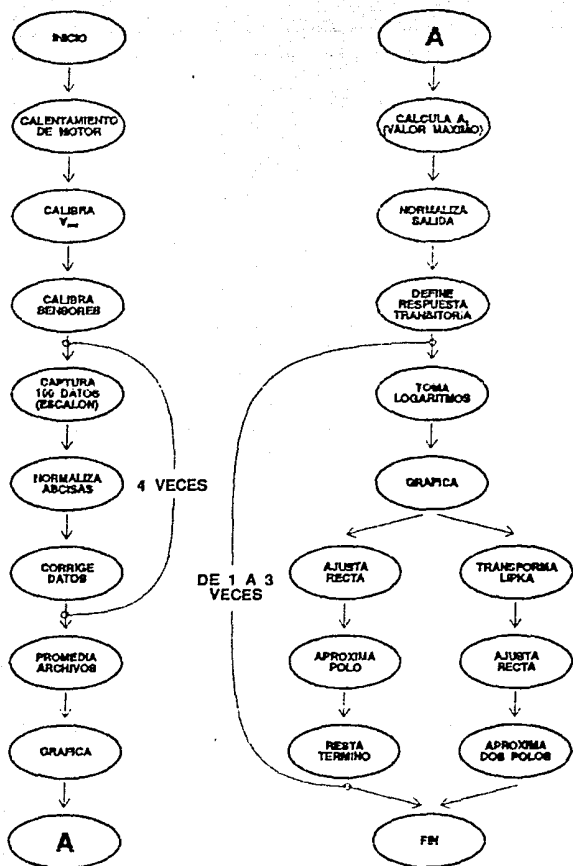


Fig. 36 Diagrama a bloques. Obtención de modelo de la planta

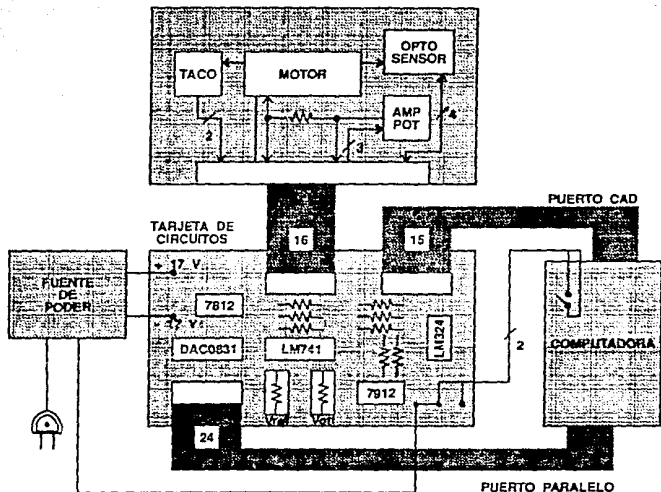


Fig. 37 Conexión de la planta.

Esto se debe a que se usará el relevador interno de la computadora como interruptor general de encendido, y como dispositivo de seguridad en caso de emergencia, ya que si se oprime la tecla ROMPE, o se apaga la computadora, el relevador se abre, cortando la corriente de los circuitos y del motor, evitando o reduciendo cualquier posible daño.

Se conecta el cable de las salidas (15 hilos) de los amplificadores operacionales de instrumentación al puerto CAD de la MicroBBC.

Se conecta el cable plano (24 hilos) que va del puerto paralelo de la MicroBBC (parte inferior del teclado) a la tarjeta donde está conectado el circuito convertidor digital analógico (DAC0831) y el amplificador operacional (LM741).

Se conecta el cable plano de 16 hilos que conecta a la planta (motor), junto con el tacómetro analógico, optosensor y amplificador de potencia a las otras tarjetas de circuitos.

Se conecta el tacómetro digital a la alimentación no regulada de +15 V (ya que tiene su propio regulador), a tierra y a la salida del sensor optoelectrónico que viene de la planta.

Se enciende el monitor y la computadora. Se coloca el disco de programas en la unidad de discos y se oprimen en el teclado simultáneamente las teclas CAMBIO y ROMPE. En este momento debe aparecer el menú del sistema.

Se enciende la fuente de poder, de modo que se carguen los capacitores de la fuente y se aliste para activar los reguladores y con ello las tarjetas de circuitos.

CALENTAMIENTO DEL MOTOR

Antes de proceder, es necesario calentar el motor para eliminar efectos térmicos indeseables, estabilizando al motor cerca de su punto de operación. Para hacer esto, se selecciona del menú la opción "Calentamiento de motor", la cual enciende y apaga el motor en forma consecutiva a intervalos crecientes y decrecientes, y luego con valores al azar. Este procedimiento tarda aproximadamente tres minutos.

VELOCIDAD MAXIMA

Al terminar la fase de calentamiento del motor, el programa de calentamiento genera el valor digital máximo de salida (plena escala de alimentación al motor) el cual sirve para ajustar el valor máximo de potencial de armadura (en lazo abierto) la primera vez que la planta se enciende. Esto es con el objeto de aprovechar el máximo rango de operación del motor de aquí en adelante, así como para tener siempre una misma referencia de voltaje.

Se utiliza el multímetro para medir el voltaje de alimentación que aparece en la armadura del motor. Este voltaje a plena escala se ajusta a 9.0 volts mediante el potenciómetro marcado V_{ref} , que es la referencia de potencial del convertidor digital analógico.

Para volver al menú principal se oprime ROMPE en el momento que se haya ajustado correctamente el potenciómetro.

CALIBRACION DE SENSORES

Ahora se selecciona la opción "Calibración de Sensores", apareciendo el mensaje "oprima una tecla para comenzar".

En este momento se obtendrán factores de equivalencia para los puertos de conversión analógica/digital, es decir, constantes que nos permitirán convertir cifras binarias adimensionales a valores equivalentes de voltaje o velocidad, ajustándose la computadora como instrumento de medición con respecto a un estándar aceptado (multímetro digital). {Holman, 27}.

En el caso del canal 1, las cifras obtenidas se equiparan a voltajes del tacogenerador analógico acoplado al motor, es decir equivalen a la velocidad angular del motor en rps, o bien a voltaje aplicado a la armadura del motor, en el caso del canal 2.

Se conecta el voltímetro digital a las terminales cuyo potencial será acondicionado por el amplificador de instrumentación y luego convertido de un valor analógico a uno digital por el CAD de la microcomputadora. Se debe empezar por el puerto de conversión 1, y repetirse para el 2. Los canales 3 y 4 no se utilizan.

En este momento el motor se enciende, luego se repite el mensaje "oprima una tecla", y al hacerlo, el CAD empieza a registrar valores, y nos muestra un promedio de las últimas 100

muestras obtenidas en el primer canal de conversión, así como el valor mínimo, el máximo, la diferencia absoluta y en porcentaje, lo cual nos da una idea de la precisión de la medición. Normalmente dicha diferencia de medición no superará el 2%.

Ahora se nos pide el valor observado por nosotros en el multímetro con lo cual se obtiene la razón de valores observado/medido (CAD) en ese puerto, lo cual nos da la constante de equivalencia.

Se cambia el multímetro a las terminales del puerto 2, repitiendo el procedimiento. Los canales 3 y 4 no se usan, por lo cual darán valores al azar y es irrelevante el valor que introduzcamos para la conversión.

Al terminar, la computadora apaga el motor y nos muestra el valor final de cada constante de conversión (4 en total), guardándolas en disco con el nombre de archivo D.sensor, ya que serán utilizadas posteriormente por los otros programas. Al terminar, automáticamente regresa al menú principal.

En este procedimiento se obtuvieron los siguientes valores:

$$\text{factor}(1) = 7.1886 \times 10^{-5}$$

$$\text{factor}(2) = 1.5623 \times 10^{-4}$$

$$\text{factor}(3) = 1.3996 \times 10^{-4}$$

$$\text{factor}(4) = 1.3755 \times 10^{-4}$$

SIMULACION DE ENTRADA ESCALON

Aquí se aplica una entrada escalón de voltaje de amplitud máxima (9 volts) a la armadura del motor, con lo cual se obtendrá y registrará su respuesta transitoria (velocidad angular vs tiempo). El procedimiento se repite cuatro veces, haciéndose después un promedio entre las medidas, para obtener finalmente un solo archivo de datos de simulación.

La variable a medir es la velocidad angular del motor, convertida primero por el tacogenerador a un valor de voltaje (4.5 volts a plena escala), luego escalada por el amplificador de instrumentación a un valor mas bajo (1.8 volts), y finalmente por el convertidor analógico digital a una cifra binaria (hex FFFF), que es lo que la computadora registrará.

Para llevar a cabo cada simulación se llevan a cabo los siguientes pasos:

Se selecciona la opción "Entrada de datos", se digita "100" para el número de datos, luego la opción de origen "Experimental".

Se nos ofrece ahora una selección de función excitadora (0: Impulso, 1: Escalón, 2: Rampa), a lo cual responderemos 1 (Escalón).

Se nos pide por ello el valor inicial del cual partirá el escalón, y el valor final al que llegará (ambos en porcentaje). Normalmente responderemos 0 para el valor inicial, y 100 para el máximo, que es el valor de alimentación a plena escala, que produce un valor hexadecimal &FF en la salida digital de 8 bits, equivalente a 9 volts de alimentación a la armadura.

En este momento la computadora está lista para capturar datos, para lo cual nos indica que debe oprimirse cualquier tecla para comenzar.

Al hacerlo, la computadora aplica la función escalón de voltaje al motor y de inmediato empieza a registrar datos, tanto del tiempo transcurrido (en centésimos de segundo) como de valores convertidos por el CAD, equivalentes a voltaje en el tacogenerador, y éstos a la velocidad angular del motor.

El tiempo total de captura para 100 datos es de poco menos de 2 segundos, en este lapso aparece el mensaje "capturando datos", luego de lo cual se apaga el motor y aparecen los mensajes "leyendo factores" y "convirtiendo datos". Aquí es donde las cifras adimensionales medidas por el CAD son convertidas a valores equivalentes de voltaje o velocidad.

Luego se nos pide un nombre de archivo para guardar este conjunto de datos (por ejemplo A.datos) y una descripción o texto (máx 40 caracteres) que los acompañe (p.ej. experim 15 dic 90 1300). De la misma manera se nos piden las unidades de tiempo (centisegundos) y las unidades de la variable medida $w(t)$, en este caso volts o rps.

Así los datos quedan guardados en disco, y al oprimir una tecla se procede a usar el programa de gráficas, donde se podrán observar los resultados.

Este procedimiento se repite cuatro veces para obtener un número igual de conjuntos de datos (p.ej. A.datos, B.datos, C.datos, D.datos).

GRAFICACION DE LA RESPUESTA TRANSITORIA

Dentro de la opción de Graficación etc. se da el nombre del archivo que recién se capturó (p.ej. A.Datos) y se oprime RETORNA. Con esto se abre el archivo y aparece el mensaje del conjunto de datos (p.ej. experim 15 dic 90 1300), la cantidad de datos almacenados y luego los mensajes "leyendo datos", "calculando máximos y mínimos" y finalmente "oprima cualquier tecla".

El programa está listo para manipular los datos leídos, para lo cual se oprime cualquier tecla, y al aparecer el menú de opciones se oprime "G" para que aparezca la gráfica de éstos, junto con el texto que los acompaña. Es posible seleccionar G para puntos aislados, H para línea punteada o J para trazo continuo.

Después de observar la gráfica se oprime la tecla "T" que nos permitirá ver la tabla de los valores leídos, se oprime P para verlos por la pantalla o I por impresora y entonces aparecerán: el texto que describe este archivo, el número de datos, los valores del origen de la gráfica y los valores máximos y mínimos de los valores del tiempo y de la función de salida.

Luego viene la tabla que muestra el número de cada dato, el valor del tiempo (en centisegundos), el valor de la función de salida, que es el voltaje del tacómetro (proporcional a la velocidad angular del motor) y el porcentaje de cada valor de dicha función con respecto al máximo registrado.

El despliegue se detiene cada 15 líneas para poder ser revisado, para avanzar otro tramo se debe oprimir la tecla "CAMBIO", hasta llegar al final de la tabla, donde aparece de nuevo "opción", como invitación a escoger una de las opciones disponibles. Se puede seleccionar "G" ("H" o "J") o "T" tantas veces como se desee.

NORMALIZACION DE ABCISAS

Lo primero que hay que hacer es normalizar las abscisas y eliminar los datos que no pertenecen a la respuesta transitoria al escalón, ya sea los que pertenecen al estado estable, o bien previos a la aplicación del escalón, si los hubiere.

Al ver tanto la tabla como la gráfica observemos que aparentemente hay datos faltantes que la computadora no pudo registrar, aunque sabemos que el muestreo se realiza exactamente cada cierto lapso constante, por la manera en que se ejecuta el ciclo de muestreo dentro del programa. De esta manera, en la gráfica aparecen como pequeños "huecos" entre cada conjunto de unos cinco puntos.

Sin embargo lo que sucede es que la computadora lleva un conteo de tiempo en centésimos de segundo aproximado al entero inmediato inferior, de modo que:

$$t = t_0$$

para:

$$t_0 \leq t < t_1$$

donde t_0 y t_1 son enteros en centisegundos mientras que t es un número real.

Por ello hemos de calcular las abscisas correctas de cada par ordenado (con fracciones del orden de décimos de milisegundo), para obtener una tabla de datos igualmente espaciados, lo cual resulta muy conveniente para poder aplicarle ciertos métodos estadísticos, además de ser analítica y visualmente más correcta.

Para hacer esto, se selecciona la opción "F" para volver al menú principal, donde se selecciona la opción 6: "Transformación de datos" ya que empezaremos a modificar gradualmente el archivo original de datos experimentales para obtener la respuesta transitoria, y más adelante, la función de transferencia.

Entrando al programa de Transformación de datos, se nos pregunta si usaremos 1 o 2 archivos, a lo que se responde "1".

Se nos pide el nombre del archivo de datos (p.ej. A.Datos). Aparecen el texto de los datos (p.ej. experim 15 dic 90 1300) y el mensaje "Leyendo datos".

Se nos pregunta si deseamos insertar datos, debido a que es posible agregar datos que hayan sido obtenidos por interpolación con la fórmula de Lagrange, o por otros medios. Por ahora respondemos "N".

A continuación debemos dar las relaciones de transformación que usaremos para modificar los datos originales, estas relaciones son cualquier combinación algebraica de los valores originales, incluidos: el número de dato $i\%$, el valor del tiempo $\{t(i\%,0)\}$, el valor de la función $\{w(i\%,0)\}$, los valores máximos $\{tmáx(0)\}$, $wmáx(0)\}$ y constantes reales.

El segundo subíndice {0} se refiere al archivo único de datos. Si hubiese dos archivos, se numerarían 0 y 1. Los nuevos datos se denominarán t' y w' .

Para obtener las abscisas equiespaciadas se transformarán los datos originales creando un nuevo archivo, en el cual la nueva abscisa será el número de dato $i\%$, mientras que la ordenada es el valor del tiempo en centisegundos, para poder obtener después la relación que los rige. De esta manera:

$$t' = i\%$$

$$w' = t(i\%, 0)$$

Para cubrir el máximo de valores obtenidos, se transformará a partir del dato 1, y hasta el 100.

Por lo tanto, al requerírnoslo la computadora anotaremos:

$$j = 1$$

$$k = 100$$

En seguida se nos pide un nuevo nombre de archivo para los datos modificados. Demos, por ejemplo, el nombre "A.absnor". También se nos pide un nuevo texto para estos datos, que puede ser "abscisas normalizadas" (podemos omitir las unidades, dejándolas en blanco).

De inmediato se transforman y se guardan estos datos modificados y al oprimir cualquier tecla, la computadora vuelve al programa de graficación, para que podamos ver el resultado de la transformación (ver pasos anteriores).

Al entrar al programa de graficación se le da el nombre del archivo recién creado (A.absnor), y al seleccionar G para ver la gráfica podemos ver que la relación de valor del tiempo al número de dato parece una recta (como se esperaba). Esta recta es de la forma:

$$y = mx + b$$

Para obtener la relación exacta, hacemos un ajuste de recta por Mínimos Cuadrados seleccionando la opción "M", a lo cual la computadora nos pide entre qué datos deseamos ajustar, y se le responde:

$$j = 1$$

$$k = 100$$

La computadora calcula la recta que mejor se ajuste al conjunto de datos, trazándola al momento sobre la gráfica original. Al pie de la gráfica nos muestra el valor calculado de la pendiente m y del intercepto al origen b . El valor r es el coeficiente de correlación, que mide la bondad del ajuste.

$$m = 1.197, b = 0.1085, r = 100 (\%)$$

m viene a ser el tiempo de muestreo en lazo abierto. Para poder ahora calcular en el archivo original el valor del tiempo exactamente en cada instante de muestreo, usaremos la siguiente ecuación:

$$t = m i\% + b$$

donde $i\%$ es el número de dato (del archivo original A.Datos) y m es la pendiente recién calculada.

Es interesante notar que por el Teorema del Muestreo, el valor máximo del polo que podríamos registrar aquí sería de aproximadamente 2.4 centisegundos, es decir el doble del tiempo de muestreo registrado antes ($= 1.2$ cs). Cualquier tiempo menor (es decir, mayor frecuencia), no podría reconstruirse fielmente.

En este momento también es conveniente observar que el incremento h (que se puede identificar también como Δt) entre dos abscisas consecutivas, será igual a la pendiente m calculada, este dato será útil mas adelante en el método de Lipka.

$$\Delta t = h = m = 1.197 \times 10^{-2} \text{ seg}$$

Ahora procederemos a transformar el archivo original de datos (A.Datos), usando el programa de transformación descrito antes, dando ahora las siguientes relaciones entre variables, sustituyendo los valores encontrados.

$$t' = m \cdot i\% + b, \quad w' = w$$

Este nuevo archivo podría llamarse A.Ndato, con la leyenda "datos normalizados".

Al graficar este nuevo conjunto de valores, vemos que ahora las abscisas están igualmente espaciadas, y ya no quedan huecos ni en la gráfica ni en la tabla.

Este procedimiento de normalización de abscisas se repite para cada conjunto de datos obtenido antes (4 en total).

Resumiendo, los valores normalizados en los cuatro archivos capturados son:

<u>Archivo</u>	<u>m (cs)</u>	<u>b (volts)</u>
A.absnor	1.197	10.85×10^{-2}
B.absnor	1.196	8.848×10^{-2}
C.absnor	1.195	8.727×10^{-2}
D.absnor	1.196	24.24×10^{-2}

PROMEDIO DE ARCHIVOS

Con los archivos A.Ndato y B.Ndato se hará un promedio E.Ndato, mientras que con C.Ndato y D.Ndato se hallará F.Ndato, para finalmente promediar E.Ndato con F.Ndato para dar I.Datos, que será el archivo de datos depurado que usaremos para los siguientes procedimientos.

El efecto de los promedios es suavizar los posibles errores aleatorios que se hayan introducido al hacer las simulaciones, de modo que su influencia se reduzca.

En sentido estricto, habría que realizar un número muy grande de muestreos y promedios para obtener una descripción más exacta del sistema. Las limitaciones de nuestra computadora nos impiden hacer esto, sin embargo con cuatro muestras se obtiene una aproximación razonable.

Para hacer los promedios se utiliza el programa de transformación de datos, declarando dos archivos de entrada en cada caso, dando los nombres correspondientes y aplicando a la totalidad de datos (100) las relaciones de transformación:

$$i'(i\%) = (i(i\%,0) + i(i\%,1))/2$$

$$w'(i\%) = (w(i\%,0) + w(i\%,1))/2$$

Este procedimiento se repite para producir los archivos antes mencionados, hasta llegar al 1.Datos.

En los datos de este nuevo archivo 1.Datos se encuentra contenida la respuesta transitoria en su totalidad, y parte de la respuesta en régimen permanente, es decir cuando el valor de la función se estabiliza en un valor prácticamente constante.

Examinando este archivo final podemos observar que las figuras de mérito recién calculadas quedan:

$$m = 1.197 \times 10^{-2} \text{ seg}$$

$$b = 7.741 \times 10^{-2} \text{ volts}$$

$$A_0 = 3.892 \text{ volts (valor máximo)}$$

ANALISIS DEL MODELO DE LA PLANTA

Lo que haremos ahora con el archivo depurado de datos de velocidad durante la respuesta del motor al escalón de 9 volts, será calcular los parámetros de la ecuación en el tiempo que describe dicha respuesta y con ello determinar las constantes de tiempo, la función de transferencia y otros datos de importancia.

Esto se hará mediante un análisis de regresión y otros métodos adecuados, todo con base en los planteamientos teóricos previos (se espera una aproximación de hasta tres polos simples).

Como se mencionó con anterioridad, hay tres maneras de aprovechar estos datos para encontrar la función de transferencia de la planta: el método logarítmico, el método de Lipka y el método de Ziegler y Nichols. Empezaremos con el primero, para hallar un aproximaciones de uno, dos y tres polos. Posteriormente se podrá hacer con ellos el análisis en lazo abierto de la planta.

RESPUESTA TRANSITORIA

Tras haber normalizado las abscisas de los datos originales, y hecho el promedio de las cuatro muestras, modificaremos el archivo 1.Datos dividiendo todos ellos entre 9, recordando que esa es la magnitud del escalón aplicado como excitación. Con esto quedarán normalizadas tanto abscisas como ordenadas, en la representación de la respuesta al escalón unitario.

Ahora es necesario insertar un punto, que sabemos correcto, si bien no aparece en nuestro archivo de datos. Este corresponde al tiempo $t=0$, en el cual $w=0$, es decir que el sistema está en reposo (condiciones iniciales nulas) hasta antes de aplicar la excitación. Por ello al indicárse nos si deseamos insertar un dato, responderemos que sí, después de $j=0$, ya que el dato irá al principio del archivo.

Tras ello daremos:

$$t(1,0)=0$$

$$w(1,0)=0$$

y no habrá más inserciones. Guardaremos el archivo como 1.Normal.

Por otra parte, solo conservaremos aquellos datos con valores mayores o iguales al 1% del valor máximo que como veremos, son suficientes para calcular la respuesta transitoria. Es decir que haremos:

$$t'(i\%) = t(i\%,0)$$

$$w'(i\%) = w(i\%,0)/9$$

$$j=1, k=46$$

recordando que el primer subíndice ($i\%$) es el número de dato y el segundo es el número de archivo.

Luego restaremos cada dato del valor final de estado estable que tiene la función (w_{max}), con lo cual solo quedarán los términos exponenciales (respuesta transitoria propiamente dicha) de la ecuación en el tiempo que buscamos, como se explicará más adelante.

Las transformaciones que haremos ahora serán:

$$t'(i\%) = t(i\%,0)$$

$$w'(i\%) = w_{max}(0) - w(i\%,0)$$

$$j=1, k=45$$

Este nuevo archivo se guarda con el nombre 1.Transi y con el texto "respuesta transitoria". Es útil recordar que:

$$w_{max}(0) = A_0 = 0.4324 \text{ volts}$$

$$t_{max} = 53.9 \times 10^{-2} \text{ seg}$$

$$\Delta t = 1.196 \times 10^{-2} \text{ seg}$$

METODO LOGARITMICO

Ahora se modifica de nuevo el conjunto de datos (1.transi) de manera similar a la descrita antes, pero haciendo ahora que:

$$t'(i\%) = t(i\%, 0)$$

$$w'(i\%) = \text{LN } w(i\%, 0)$$

y guardando el nuevo conjunto con el nombre 1.Logtra. Este archivo de logaritmos de los datos experimentales será el eje principal para el cálculo de coeficientes y polos mediante este método, como se verá.

APROXIMACION DE UN POLO

Como primera aproximación al análisis del sistema, consideremos que se rige por una relación entrada — salida de un solo polo.

Es decir que si nuestra ecuación buscada de respuesta del sistema al escalón unitario es:

$$w(t) = A_0 - A_1 e^{-\frac{t}{T_1}} \quad (33)$$

donde $A_0 = w_{\text{max}}(0) = 0.4324$ volts

es el valor final (de estado estable) de la función y T_1 es la constante de tiempo (el polo es $1/T_1 = \tau_1$), entonces:

$$A_0 - w(t) = A_1 e^{-\frac{t}{T_1}}$$

representa la parte transitoria de la función, aproximada con una sola constante de tiempo T_1 .

Al graficar los datos del archivo 1.Logtra observemos que el trazo se podría aproximar por una recta, la cual corresponde al factor exponencial que buscamos.

De esta manera, se puede ajustar ese conjunto de datos a una recta de la forma

$$z = mt + b$$

Donde $z = \text{LN}(w)$.

Dado que estamos graficando $\text{LN}(w)$ contra t , se tiene:

$$A_1 = e^b$$

$$T_1 = -1/m$$

donde T_1 es la constante de tiempo que corresponde a la función de transferencia. Estos cálculos están integrados en el programa de graficación, de modo que basta seleccionar, estando la gráfica en la pantalla, la opción "M" (mínimos cuadrados) para obtener el mejor ajuste de una recta entre los puntos

$$j=1 \text{ y } k=20$$

obteniéndose la recta:

$$z = 0.1057 t - 0.5809$$

la bondad del ajuste es $r=99.18\%$.

Inmediatamente después seleccionamos la opción "L" (método logarítmico), que incorpora dichos cálculos dentro del programa de graficación, de donde se obtiene:

$$A_1 = 0.5594 \text{ volts}$$

$$T_1 = 9.643 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 10.57 \text{ Hz}$$

Por condiciones iniciales, sabemos que $y(0)=0$, por lo cual es necesario que $A_0=w_{\max}(0)$, por lo tanto obtendremos un nuevo valor de T_1 , que corresponda al valor deseado de A_1 , oprimiendo la opción A en el programa de graficación, introduciendo el nuevo valor de $A_1 (=0.4324)$, con lo que quedan los nuevos datos:

$$A_1 = 0.4324 \text{ volts}$$

$$T_1 = 12.03 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 8.315 \text{ Hz}$$

Aquí se vé que la constante de tiempo es unas diez veces mayor que el tiempo de muestreo en lazo abierto ($1.196 \times 10^{-2} \text{ seg}$), por lo que podemos suponer que la reconstrucción de la función a partir de los datos experimentales es fidedigna.

Al tener los valores de A_1 y T_1 , podemos acceder analíticamente al término:

$$A_1 e^{-\frac{t}{T_1}}$$

Con el cual podemos completar nuestra ecuación original, que queda:

$$w(t) = 0.4324 (1 - e^{-\frac{t}{12.03}}) \quad (34)$$

La cual podemos recalcular analíticamente para después comparar con el original experimental.

Para hacer esto, se toma la opción F del programa de graficación (fin) y el número 6 (transformación), para luego seleccionar el archivo de entrada 1.Normal, y substituyendo en $w'(i\%)$ la ecuación recién calculada, es decir dando las transformaciones:

$$t'(i\%) = t(i\%, 0)$$

$$w'(t) = w_{\max}(0) * (1 - \text{EXP}(t(i\%, 0)/12.03))$$

$$j=1, k=46$$

Después de un momento, la computadora termina los cálculos, y le daremos al archivo el nombre 1.UnPol, la cual podemos graficar, seleccionando luego la opción O (otra gráfica) y dando el nombre de archivo 1.Datos, que es la original (experimental), donde podemos observar que la aproximación es razonablemente buena.

También es posible restar los valores originales a la función recién hallada, usando el programa de transformación con los archivos 1.Normal y 1.Unpol:

$$t'(i\%) = t(i\%,0)$$

$$w'(t) = (w(i\%,1) - w(i\%,0)) * 100 / w_{\max}(0)$$

$$j=1, k=46$$

Este archivo, que llamaremos 1.Dif1, al ser graficado muestra que los errores máximos son de +14.96% y -4.975%. Comprobando con ello que la aproximación es buena.

A partir de los resultados hallados, es fácil ver que la función de transferencia de la planta aproximada así, es:

$$\frac{W(s)}{I(s)} = \frac{A_0}{T_1} \frac{1}{s + \frac{1}{T_1}} \quad (35)$$

Donde $I(s) = 1/s$ es el escalón que se aplicó como excitación a la entrada.

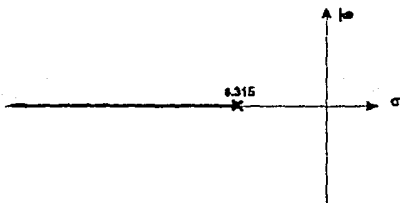


Fig. 38 Lugar de raíces, un solo polo.

y donde A_0 / T_1 es la ganancia de la función de transferencia. También es posible graficar el lugar de las raíces para esta aproximación, donde veremos que se trata de una planta estable, con un polo en $1/T_1 = 8.315$ Hz

Es decir que habría que sustituir los valores:

$$A_0 = 0.4324 \text{ volts}$$

$$T_1 = 12.03 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 8.315 \text{ Hz}$$

De modo que al sustituir, la función de transferencia queda:

$$\frac{W(s)}{1(s)} = \frac{3.594}{s+8.315} \quad (36)$$

APROXIMACION CON DOS POLOS

Se puede ver que la aproximación analítica con un solo polo podría mejorarse considerando un sistema de segundo orden, por lo cual ahora supondremos que la ecuación que describe la respuesta del sistema al escalón unitario es de la forma:

$$w(t) = A_0 - A_1 e^{-\frac{t}{T_1}} - A_2 e^{-\frac{t}{T_2}} \quad (37)$$

igual que antes, con A_0 como valor final pero ahora con dos polos $1/T_1$ y $1/T_2$. Con esto se tendría:

$$A_0 - w(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} \quad (38)$$

como parte transitoria de la función.

Para retomar el procedimiento anterior, ahora con dos polos, utilizaremos el mismo archivo logarítmico creado antes, 1. Logira, pero consideraremos que en él sólo la parte final del trazo (la última mitad) se aproxima a una recta, que corresponde al polo dominante del sistema.

Si suponemos que las constantes de tiempo son diferentes: $T_1 > T_2$ (D'azzo y Houppis, 60), entonces el término que buscamos será $A_1 e^{-\frac{t}{T_1}}$, es decir:

$$A_2 e^{-\frac{t}{T_2}} = 0$$

en ese intervalo.

De esta manera, se puede ajustar ese conjunto de datos a una recta de la forma

$$z = mt + b$$

Donde $z = LN(w)$.

Dado que estamos graficando $LN(w)$ versus t , se tiene:

$$A_1 = e^b$$

$$T_1 = -\frac{1}{m}$$

los cuales describen el polo dominante ($1/T_1$) de la función de transferencia. Como antes, basta oprimir "M" (mínimos cuadrados) para obtener el mejor ajuste de una recta entre los puntos:

$$j=25, k=43$$

de donde se halla la recta:

$$z = -0.1283 t - 1.605 \times 10^{-2}$$

la bondad del ajuste encontrado es $r=99.25\%$.

Inmediatamente después, seleccionamos la opción "L" (método logarítmico), que incorpora los mencionados cálculos dentro del programa de graficación, de donde se obtiene:

$$A_1 = 0.9841 \text{ volts}$$

$$T_1 = 7.792 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 12.834 \text{ Hz}$$

Al tener los valores de A_1 y T_1 , podemos obtener analíticamente el término:

$$A_1 e^{-\frac{t}{T_1}}$$

El cual podemos restar algebraicamente a nuestra ecuación original (archivo 1.Transi), mediante el programa de transformación de datos. Es decir que ahora haremos:

$$t'(i\%, 0) = t(i\%, 0)$$

$$w'(i\%, 0) = w(i\%, 0) - A_1 \cdot \exp(t(i\%, 0)/T_1)$$

substituyendo los valores recién encontrados de A_1 y T_1 queda:

$$w'(i\%, 0) = w(i\%, 0) - 0.9841 \cdot \exp(t(i\%, 0)/7.792)$$

$$j=1, k=46$$

Con esto nuestros datos experimentales, considerados ahora sólo en la porción $t < T_1$, corresponderán al término faltante $A_2 e^{-\frac{t}{T_2}}$.

$$A_0 - w(t) - A_1 e^{-\frac{t}{T_1}} = A_2 e^{-\frac{t}{T_2}}$$

Este nuevo conjunto de datos recibirá el nombre de archivo 2.tra2, el cual se convierte de manera similar a la antes citada, considerando sólo la porción no afectada por la transformación anterior. Aquí es importante notar que en el rango de interés del archivo 2.tra2 se tiene:

$$w(i) < 0$$

por lo que hay que cambiar el signo de las ordenadas en toda la transformación, cuyo significado es que A_2 tiene signo opuesto al que se supuso (o sea $A_2 < 0$), sin que esto afecte los cálculos que hay que hacer. Por lo tanto:

$$t'(i\%,0) = t(i\%,0)$$

$$w'(i\%,0) = \text{LN}(w(i\%,0))$$

$$j = 1$$

$$k = 20$$

Al convertir estos datos les llamaremos 1.Log2, y luego al graficarlos observemos que el trazo casi se puede aproximar por una recta que corresponde al factor exponencial que buscamos.

De esta manera, se ajusta ese conjunto de datos a una recta:

$$z = nt + c$$

donde:

$$z = \text{LN}(w).$$

Dado que estamos graficando $\text{LN}(w)$ versus t , se deduce que:

$$A_2 = e^c$$

$$T_2 = -1/n$$

los cálculos están integrados en el programa de graficación, por ello basta oprimir "M" para obtener el ajuste de la recta entre los puntos $j=1$ y $k=20$, obteniéndose la recta:

$$z = 0.1872 t - 0.8195$$

la bondad de este ajuste es de $r=99.32\%$.

La opción "L" (método logarítmico) ahora da:

$$A_2 = 0.4406 \text{ volt}$$

$$T_2 = 5.341 \times 10^{-2} \text{ seg}$$

$$\tau_2 = 18.723 \text{ Hz}$$

el proceso se podría repetir varias veces para refinarlo y llegar a los valores que sean más adecuados, considerando que para condiciones iniciales nulas debería cumplirse que:

$$A_0 - A_1 - A_2 = 0$$

por lo cual dados A_0 y A_1 , debe ser $A_2 = 0.5517$.

Esto se logra oprimiendo la tecla 'A', con lo cual se nos pide el valor deseado para A_2 (0.5517), y acto seguido se traza una nueva recta y se calculan los valores adecuados a nuestra elección.

$$A_2 = 0.5517 \text{ volts}$$

$$T_2 = 4.833 \times 10^{-2} \text{ seg}$$

$$\tau_2 = 20.691 \text{ Hz}$$

Lo cual completa la ecuación propuesta. En este caso, al afinar y sustituir valores queda:

$$w(t) = 0.4324 - 0.9841e^{-\frac{t}{7.792}} + 0.5517e^{-\frac{t}{4.833}} \quad (39)$$

La cual podemos recalcular analíticamente para después comparar con el original experimental (datos muestreados).

Para hacer esto, se toma la opción F del programa de graficación (fin) y el número 6 (transformación), para luego seleccionar el archivo de entrada 1.Normal, y substituyendo en $w'(i\%)$ la ecuación recién calculada, es decir dando las transformaciones:

$$t'(i\%) = t(i\%,0)$$

$$w'(t) = w_{\max}(0) - 0.9841 \cdot \text{EXP}(t(i\%,0)/7.792) + 0.5517 \cdot \text{EXP}(t(i\%,0)/4.833)$$

$$j=1, k=46$$

y le daremos al archivo el nombre 1.Dospol, el cual podemos graficar, seleccionando luego la opción O (otra gráfica) y dando el nombre de archivo 1.Ndato, que es el original, donde podemos observar que la correlación entre ambas ha mejorado mucho. Al hacer la resta de datos como se hizo en la aproximación de un polo, se halla que los errores máximos son 1.62%, +7.1%, con una notable mejoría.

Con los datos hallados y haciendo las manipulaciones algebraicas necesarias, se llega a la función de transferencia para la planta:

$$\frac{W(s)}{I(s)} = \frac{B_0(s+C_0)}{(s+\frac{1}{T_1})(s+\frac{1}{T_2})} \quad (40)$$

donde la ganancia B_0 es:

$$B_0 = A_0 \left(\frac{1}{T_1} + \frac{1}{T_2} \right) - \frac{A_1}{T_2} - \frac{A_2}{T_1} \quad (41)$$

que con los valores:

$$A_0 = 0.4324, A_1 = 0.9841, A_2 = -0.5517, T_1 = 7.792 \times 10^{-2}, T_2 = 4.833 \times 10^{-2}$$

resultó ser $B_0 = 1.214$

$$C_0 = -\frac{A_0}{B_0 T_1 T_2} \quad (42)$$

es el cero de la función en el plano complejo s , que dá $C_0 = 94.553$.

Substituyendo estos valores quedaría:

$$\frac{W(s)}{I(s)} = \frac{1.214(s+94.553)}{(s+12.834)(s+20.691)} \quad (43)$$

Con esta función será posible trazar el lugar de las raíces, hallando que es un sistema estable con polos en:

$$s = -12.834, -20.691$$

así como un cero en $s = -94.553$

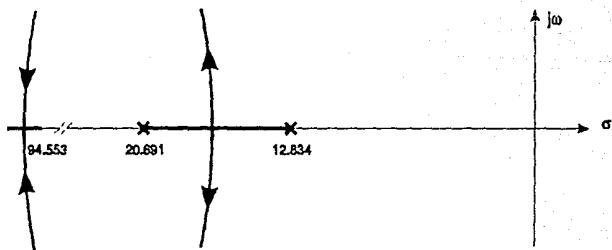


Fig. 39 Lugar de raíces. Dos polos.

El trazo de las ramas que se desprenden del eje real es curvo, debido a que estas son en realidad uno solo, y forman parte de una circunferencia, cuyo radio es muy grande para poder ser representado con claridad aquí, dado el valor del cero del lado izquierdo.

Nótese que podría darse el caso de que alguno de los valores hallados (particularmente los ceros) se encontrara en el semiplano derecho, inconsistencia que se debe a errores en las mediciones experimentales, cálculos y aproximaciones, y tal caso debe tenerse en mente al momento de diseñar un controlador con base en este modelo. {Franklin et al., 74}

De lo hecho hasta este momento, se desprende que la aproximación de un polo ajusta a la perfección la parte final de los datos experimentales; la de dos polos mejora mucho el modelo en los puntos intermedios, pero falta detalle en los primeros datos. Por lo anterior, lo mejor será hacer una aproximación de tres polos, para tratar de corregir esos errores en los primeros puntos, como se verá más adelante.

APROXIMACION CON TRES POLOS

La aproximación de dos polos aún se puede mejorar considerando un tercer polo, extendiendo la ecuación original a:

$$w(t) = A_0 - A_1 e^{-\frac{t}{T_1}} - A_2 e^{-\frac{t}{T_2}} - A_3 e^{-\frac{t}{T_3}} \quad (44)$$

como respuesta al escalón, con lo que:

$$A_0 - w(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} + A_3 e^{-\frac{t}{T_3}}$$

representa la parte transitoria de la función.

donde A_0 es el valor final, y T_1 , T_2 y T_3 son las constantes de tiempo, en orden decreciente de importancia.

El cálculo de todos los coeficientes y polos se logra subdividiendo la gráfica del archivo 1. Logra en tres segmentos, de los cuales el último se aproxime por una recta, y después repetir en dos ocasiones el procedimiento de resta algebraica del factor hallado, seguido de una nueva toma de logaritmos, para hallar los otros factores con este método.

Se parte otra vez del archivo 1. Logra, haciendo ahora que sólo el trazo final (unos diez datos) se aproxime por una recta, la cual corresponde al primer factor exponencial que buscamos, ya que considerando (como antes) $T_1 > T_2 > T_3$ [D'arzo y Houppis, 60], entonces el término que buscamos será:

$$A_1 e^{-\frac{t}{T_1}}$$

es decir que:

$$A_2 e^{-\frac{t}{T_1}} - A_3 e^{-\frac{t}{T_1}} = 0$$

en ese intervalo.

De esta manera, se puede ajustar ese conjunto de datos a una recta de la forma

$$z = mt + b$$

Donde $z = \text{LN}(w)$.

Dado que estamos graficando $\text{LN}(w)$ versus t , se deduce que:

$$A_1 = e^b$$

$$T_1 = -1/m$$

los cuales corresponden al polo dominante de la función de transferencia. Estos cálculos están integrados al programa, de modo que basta seleccionar la opción "M" (mínimos cuadrados) para obtener el ajuste de una recta entre los puntos $j=30$ y $k=43$, obteniéndose la recta:

$$z = -0.1412 t + 0.5566$$

con bondad de ajuste $r=99.15\%$

Luego seleccionamos la opción "L" (método logarítmico), que incorpora dichos cálculos dentro del programa de graficación, de donde se obtiene directamente:

$$A_1 = 1.745 \text{ volts}$$

$$T_1 = 7.081 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 14.122 \text{ Hz}$$

Al tener los valores de A_1 y T_1 , podemos acceder analíticamente al término:

$$A_1 e^{-\frac{t}{T_1}}$$

El cual podemos restar algebraicamente a nuestra ecuación original (archivo 1.Transi), mediante el programa de transformación de datos. Es decir que ahora haremos:

$$t'(i\%, 0) = t(i\%, 0)$$

$$w(i\%, 0) = w(i\%, 0) - A_1 \cdot \text{EXP}(t(i\%, 0)/T_1)$$

substituyendo los valores recién encontrados de A_1 y T_1 . Con esto nuestros datos experimentales, considerados ahora sólo en la porción $t > T_1$, corresponderán a los términos 2 y 3:

$$A_0 - w(t) - A_1 e^{-\frac{t}{T_1}} = A_2 e^{-\frac{t}{T_2}} + A_3 e^{-\frac{t}{T_3}} \quad (45)$$

Este nuevo conjunto de datos recibirá el nombre de archivo 3.tra2, el cual se convierte de manera similar a la anterior.

También aquí se halla que:

$$w(i\%) < 0$$

por lo que hay que cambiar el signo antes de sacar logaritmos, y el coeficiente A_2 tendrá signo negativo, contrariamente a lo supuesto inicialmente.

Las transformaciones son:

$$t'(i\%, 0) = t(i\%, 0) \text{ y}$$

$$w'(i\%, 0) = \text{LN}(w(i\%, 0))$$

$$j = 1, k = 29$$

Al convertir estos datos les llamaremos 1.Logbis, y luego al graficarlos observemos que el trazo final se puede aproximar ahora también por una recta, correspondiente al siguiente factor exponencial que buscamos.

De esta manera, se puede ajustar ese conjunto de datos a una recta de la forma:

$$z = nt + c$$

donde $z = \text{LN}(w)$.

Dado que estamos graficando $\text{LN}(w)$ versus t , se deduce que:

$$A_2 = e^c$$

$$T_2 = -1/n$$

Ya que estos cálculos están integrados en el programa de graficación, al igual que antes basta seleccionar, estando la gráfica en la pantalla, la opción "M" (mínimos cuadrados) para obtener el mejor ajuste de una recta entre los puntos $j=15$ y $k=29$, obteniéndose ahora la recta:

$$z = 0.1951 t + 0.7116$$

$$r=99.31\%$$

Seleccionamos la opción "L" (método logarítmico) se obtiene:

$$A_2 = -2.037 \text{ volts}$$

$$T_2 = 5.125 \times 10^{-2} \text{ seg}$$

$$\tau_2 = 19.512 \text{ Hz}$$

Exactamente el mismo procedimiento (restar el término hallado, sacar logaritmo, ajustar una recta) se aplica para hallar los valores faltantes.

Por ello se calcula a partir de 1.trabis:

$$w'(i\%) = w(i\%,0) + 2.037 \cdot \text{EXP}(t(i\%,0)/5.125)$$

$$j=1, k=14$$

quedando como archivo 3.tra, y al sacar logaritmos con:

queda el archivo 3.log, que al graficarse parece una recta entre los puntos:

$$j=1, k=11$$

y al oprimir L para aplicar el método logarítmico se obtiene:

$$A_2 = 0.7565 \text{ volts}$$

$$T_2 = 4.540 \times 10^{-2} \text{ seg}$$

$$\tau_2 = 22.03 \text{ Hz}$$

El proceso se podrá repetir varias veces para llegar a los valores que parezcan más adecuados, considerando que para condiciones iniciales nulas se cumple que:

$$A_0 - A_1 - A_2 - A_3 = 0 \quad (46)$$

en cuyo caso debe tenerse que

$$A_3 = A_0 + A_1 + A_2 = 0.7244$$

por lo que se afina este último valor del coeficiente oprimiendo 'A' y dando el valor 0.7244. Finalmente, se llegó a los valores:

$$A_3 = 0.7244 \text{ volts}$$

$$T_3 = 4.693 \times 10^{-2} \text{ seg}$$

$$\tau_3 = 21.308 \text{ Hz}$$

y con los otros valores ya hallados se puede formar la siguiente tabla:

i	A_i	T_i	τ_i
	volts	$\times 10^{-2} \text{ seg}$	Hz
0	0.4324	—	—
1	1.745	7.081	14.122
2	-2.037	5.125	19.512
3	0.7244	4.693	21.308

Tabla 2. Tres polos. Resumen de valores

Y así se tiene completa nuestra ecuación:

$$w(t) = A_0 - A_1 e^{-\frac{t}{\tau_1}} - A_2 e^{-\frac{t}{\tau_2}} - A_3 e^{-\frac{t}{\tau_3}}$$

que al sustituir valores queda:

$$w(t) = 0.4324 - 1.745 e^{-\frac{t}{7.081}} + 2.037 e^{-\frac{t}{5.125}} - 0.7244 e^{-\frac{t}{4.693}} \quad (47)$$

La cual podemos recalcular analíticamente para después comparar con el original experimental.

Para hacer esto se toma la opción F del programa de graficación (fin) y el número 6 (transformación), para luego seleccionar el archivo de entrada 1.Normal, y al substituir en $w'(i\%)$ la ecuación recién calculada, es decir dando las transformaciones:

$$t'(i\%) = t(i\%, 0)$$

$$w'(t) = w_{\max}(0) - 1.745 \cdot \text{EXP}(-t(i\%, 0)/7.081) + 2.037 \cdot \text{EXP}(-t(i\%, 0)/5.125) - 0.7244 \cdot \text{EXP}(-t(i\%, 0)/4.693)$$

después de un momento, la computadora termina los cálculos, y le daremos al archivo el nombre 1.Trepol, la cual podemos graficar, seleccionando luego la opción O (otra gráfica) y dando el nombre de archivo 1.Normal, que es la original, donde podremos observar la cercana correlación que hay entre ambas.

Al hacer la resta de archivo como antes, se halla que las desviaciones máximas son ahora: -4.864% , $+3.577\%$. Estos datos de error se guardan en un archivo con el nombre 3.error.

ANALISIS EN LAZO ABIERTO (3 POLOS)

Utilizando la ecuación encontrada de respuesta al escalón unitario,

$$w(t) = A_0 - A_1 e^{-\frac{t}{T_1}} - A_2 e^{-\frac{t}{T_2}} - A_3 e^{-\frac{t}{T_3}}$$

y tomando su transformada de Laplace, se tiene:

$$W(s) = \frac{A_0}{s} - \frac{A_1}{(s + \frac{1}{T_1})} - \frac{A_2}{(s + \frac{1}{T_2})} - \frac{A_3}{(s + \frac{1}{T_3})} \quad (48)$$

que, al desarrollar la suma algebraica, queda:

$$W(s) = \frac{a_0 s^3 + a_1 s^2 + a_2 s + a_3}{s(s + \frac{1}{T_1})(s + \frac{1}{T_2})(s + \frac{1}{T_3})} \quad (49)$$

donde:

$$a_0 = A_0 - A_1 - A_2 - A_3 = 0$$

a_0 debe ser cero por tenerse condiciones iniciales nulas.

$$a_1 = \left(\frac{1}{T_1} + \frac{1}{T_2} + \frac{1}{T_3}\right) A_0 - \left(\frac{1}{T_2} + \frac{1}{T_3}\right) A_1 - \left(\frac{1}{T_1} + \frac{1}{T_3}\right) A_2 - \left(\frac{1}{T_1} + \frac{1}{T_2}\right) A_3 \quad (50)$$

$$a_2 = \frac{1}{T_1 T_2} (A_0 - A_3) + \frac{1}{T_1 T_3} (A_0 - A_2) + \frac{1}{T_2 T_3} (A_0 - A_1) \quad (51)$$

$$a_3 = \frac{A_0}{T_1 T_2 T_3} \quad (52)$$

Por lo tanto, al factorizar el numerador, podemos poner la siguiente ecuación:

$$\frac{W(s)}{1(s)} = \frac{a_1(s^2 + \frac{a_2}{a_1}s + \frac{a_3}{a_1})}{(s + \frac{1}{T_1})(s + \frac{1}{T_2})(s + \frac{1}{T_3})} \quad (53)$$

o bien:

$$\frac{W(s)}{1(s)} = \frac{B_0(s+C_1)(s+C_2)}{(s + \frac{1}{T_1})(s + \frac{1}{T_2})(s + \frac{1}{T_3})} \quad (54)$$

donde:

$$B_0 = a_1$$

$$C_{1,2} = \frac{-a_2 \pm \sqrt{a_2^2 - 4a_3a_1}}{2a_1}$$

Los valores numéricos hallados son:

$$B_0 = a_1 = 332.826 \times 10^{-3}$$

$$a_2 = 116.893$$

$$a_3 = 2.539 \times 10^3$$

$$C_{1,2} = 23.255, 327.891$$

También se puede usar el programa de cálculos varios, donde se integran las igualdades anteriores, y sustituyendo los valores hallados tenemos que la función de transferencia de la planta es:

$$\frac{W(s)}{1(s)} = \frac{0.3328(s+23.255)(s+327.891)}{(s+14.122)(s+19.512)(s+21.308)} \quad (55)$$

el coeficiente correspondiente a la tercera potencia de s (que en sentido estricto debe ser cero) representa un error, cuya magnitud también se halla al sustituir valores, y es de:

$$a_0 = \text{error} = 2.0977 \times 10^{-7}$$

Recuérdese que al hacer esta aproximación de tres polos, también podría obtenerse un sistema de fase no mínima por tener algún cero en el semiplano derecho {D'azzo y Houpis (LCS), 272}, si bien el error introducido sería pequeño.

El lugar de raíces en el plano complejo s , de la aproximación de tres polos hallada, se presenta en la figura 40, donde los trazos verticales que se separan del eje real forman en realidad una sola rama, que forma una circunferencia, pero cuyo radio es demasiado grande para poder ser representado con claridad aquí, dado el valor relativo de los ceros del extremo izquierdo del eje real. (Ver fig. 40)

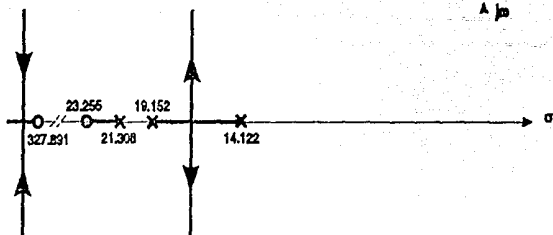


Fig. 40 Lugar de raíces. Tres polos.

METODO DE LIPKA

En los datos del archivo 1.transi se encuentra contenida la totalidad de la respuesta transitoria, la cual se aproximó anteriormente con modelos hallados con el método logarítmico, si bien también es posible aproximarla por un modelo de dos polos, mediante el método de Lipka.

Este método es más exacto cuando los dos polos dominantes se hallan separados al menos por un orden de magnitud, ya que de otra forma los cálculos pierden exactitud. Por otra parte, este método es muy sensible a la calidad original de los datos. Aunque en nuestro caso particular no se cumple esta condición, según se ve con el método logarítmico, de cualquier forma es ilustrativo ver el resultado arrojado por este método.

El valor final A_0 queda igual, siendo ahora necesario hallar los coeficientes A_1 y A_2 , y los polos $1/T_1$ y $1/T_2$ de la ecuación:

$$w(t) = A_0 - A_1 e^{-\frac{t}{T_1}} - A_2 e^{-\frac{t}{T_2}}$$

Las transformaciones que haremos ahora serán (ver desarrollo teórico):

$$l'(i\%) = w(i\%+1,0)/w(i\%,0)$$

$$w'(i\%) = w(i\%+2,0)/w(i\%,0)$$

recordando que el primer subíndice es el número de dato y el segundo es el número de archivo.

En esta transformación ajustaremos los subíndices para incluir solo la parte más representativa, ya que este método es muy sensible a la calidad de los datos, por ello daremos:

$$j = 1$$

$$k = 20$$

Este nuevo archivo se guarda con el nombre 1.Lipka y con el texto "metodo Lipka".

Al graficar estos datos veremos que se agrupan cerca de una recta, que se puede ajustar entre los límites:

$$j = 1$$

$$k = 9$$

este ajuste dá como resultado la relación lineal:

$$y = 1.576 x - 0.6179$$

$$r = 97.49 \%$$

De acuerdo al desarrollo teórico hecho antes, esta recta sirve para hallar el par de polos que describen la ecuación en el tiempo de respuesta transitoria, dada arriba, es decir que hallaremos $1/T_1$ y $1/T_2$, dados por:

$$-\frac{1}{T_1}, -\frac{1}{T_2} = \frac{1}{h} LN \left(\frac{p \pm \sqrt{p^2 + 4q}}{2} \right) \quad (57)$$

donde $h=1.198$ es la diferencia entre abscisas, y $p=1.576$, $q=-0.6179$ son los valores recién hallados con la recta ajustada, por lo que se encuentra que:

$$T_1 = 7.061 \times 10^{-2} \text{ seg}$$

$$T_2 = 3.833 \times 10^{-2} \text{ seg}$$

$$\left(\frac{1}{T_2} - \frac{1}{T_1} \right) = -11.927 \text{ Hz}$$

Ahora con estos dos polos se hacen las siguientes transformaciones, notando que:

$$\left(\frac{1}{T_2} - \frac{1}{T_1} \right) < 0$$

$$t'(i\%) = \text{EXP} \left((1/T_2 - 1/T_1) * t(i\%, 0) \right)$$

$$w'(i\%) = w(i\%, 0) * \text{EXP}(t/T_2)$$

al graficar este nuevo conjunto de datos, se obtendrá casi una recta, con abscisas decrecientes y ordenadas casi constantes, que producirán una relación casi lineal al ajustar por mínimos cuadrados entre los puntos:

$$j = 1$$

$$k = 9$$

produciendo la recta:

$$y = 0.673 x + 1.093$$

constante T_1 (-9.4%). Esto confirma la sensibilidad del método a la calidad de los datos originales, así como a la relativa cercanía que hay entre los dos polos.

El respectivo lugar de raíces de la aproximación de Lipka se muestra en la fig. 41.

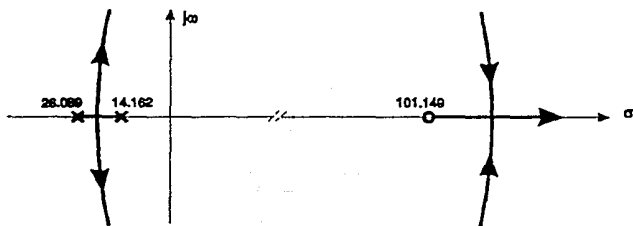


Fig. 41 Lugar de raíces. Dos polos por el método de Lipka.

METODO DE ZIEGLER Y NICHOLS

Con este método es posible describir la respuesta del sistema mediante una función de transferencia de la forma:

$$\frac{W(s)}{I(s)} = \frac{R e^{-Ls}}{T_1 s + 1} \quad (59)$$

es decir un retardo en el tiempo (también llamado retardo de transporte) de magnitud L , con un polo simple de valor $1/T_1$ y con una ganancia R/T_1 . La función equivalente en el tiempo sería:

$$g(t) = \left(\frac{R}{T_1}\right) u(t-L) \left(1 - e^{-\frac{(t-L)}{T_1}}\right) \quad (60)$$

donde $u(t-L)$ es un escalón unitario retrasado L unidades, es decir que la función es una exponencial simple retrasada L unidades en el tiempo.

Para calcular los parámetros R , L y T_1 se utiliza la gráfica de la respuesta de la planta (1.Ndato) sobre la cual se traza una recta que pase por el punto de inflexión, es decir que es la recta de mayor pendiente que se pueda trazar sobre dicha respuesta (ver figura). {Alvarez y Alvarez, 2; Takahashi et al., 343, 369; Chesmond, 398}.

Esto se logra en el programa de graficación, seleccionando el archivo 1.Ndato, y luego aplicando la opción "M" (mínimos cuadrados), dando los límites:

$$j = 3$$

$$k = 4$$

que corresponden al despegue inicial de la respuesta, donde la pendiente es máxima, lo cual da como resultado la recta de características:

$$m = 2.947 \times 10^{-2}$$

$$b = -5.692 \times 10^{-2}$$

A partir de lo cual se selecciona la opción Z (Ziegler), que proporciona directamente los valores:

$$R = 2.947$$

$$A_0 = 0.3755 \text{ volts}$$

$$T_1 = 12.74 \times 10^{-2} \text{ seg}$$

$$L = 1.932 \times 10^{-2} \text{ seg}$$

Sin embargo, al notar que debe tenerse que:

$$R/T_1 = A_0 = 0.4324 \text{ volts}$$

vemos que podemos combinar el método de Ziegler, para calcular una primera aproximación, y combinarlo con el método logarítmico (es decir el valor de la constante de tiempo $T_1=9.296 \times 10^{-2}$ seg) para hallar una aproximación mas refinada a la ecuación.

Esto se hace directamente en el programa de graficación: tras seleccionar la opción de Ziegler, se oprime A para afinar el ajuste y se le dá el valor de la constante de tiempo deseada ($T_1=9.643 \times 10^{-2}$) y el punto experimental por el que se desea que pase la recta modificada ($i\%=3$).

Al hacer esto, se llega a los siguientes valores:

$$R = 4.170 \times 10^{-2} \text{ volt/seg}$$

$$T_1 = 9.643 \times 10^{-2} \text{ seg}$$

$$\tau_1 = 10.370 \text{ Hz}$$

$$L = 2.116 \times 10^{-2} \text{ seg}$$

Entonces, la función de transferencia viene a ser:

$$W(s) = \frac{4.170 e^{-2.116s}}{9.643s+1}$$

(61)

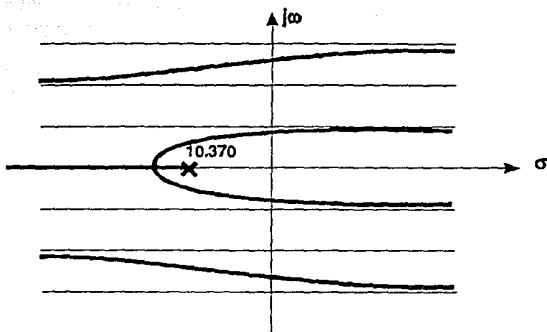


Fig. 42 Lugar de raíces. Un polo y retraso.

Con esta ecuación es posible trazar el respectivo lugar de raíces en el plano s como se ve en la fig. 42 (que tiene infinitas ramas).

Con dicha función de transferencia también de inmediato se puede pasar a la descripción del sistema en el tiempo como respuesta al escalón con la ecuación:

$$w(t) = 0.4324 u(t-2.116) \left(1 - e^{-\frac{(t-2.116)}{9.643}} \right) \quad (62)$$

la cual puede graficarse y compararse contra los datos originales, viéndose que la aproximación es bastante buena pero no tanto como la de tres polos por el método logarítmico.

RESUMEN DE FUNCIONES

Como hemos visto, hay varias posibles aproximaciones a la función de transferencia de nuestra planta. Para propósitos de diseño de los controladores digitales, usaremos la función hallada para la aproximación de un polo, por su evidente simplicidad.

La de Ziegler y Nichols (polo y retraso) podría usarse porque con ella ya se tienen tablas para el cálculo directo de los coeficientes del controlador digital con algoritmo PID, pero la de tres polos, a pesar de ser la más apegada a los datos experimentales, produciría un sistema de ecuaciones muy complejo.

De cualquier modo, se resumen aquí las funciones de transferencia halladas, que son :

$$\frac{W_1(s)}{1(s)} = \frac{\frac{A_0}{T_1}}{\left(s + \frac{1}{T_1}\right)} \quad (35)$$

$$\frac{W_2(s)}{1(s)} = \frac{B_0(s+C_0)}{\left(s + \frac{1}{T_1}\right)\left(s + \frac{1}{T_2}\right)} \quad (40)$$

$$\frac{W_3(s)}{1(s)} = \frac{B_0(s+C_1)(s+C_2)}{\left(s + \frac{1}{T_1}\right)\left(s + \frac{1}{T_2}\right)\left(s + \frac{1}{T_3}\right)} \quad (54)$$

$$\frac{W_4(s)}{1(s)} = \frac{\frac{R}{T_1} e^{-Ls}}{\left(s + \frac{1}{T_1}\right)} \quad (59)$$

Que al substituir los valores numéricos hallados quedan como sigue:

$$\frac{W_1(s)}{1(s)} = \frac{3.594}{(s+8.315)} \quad (36)$$

$$\frac{W_2(s)}{1(s)} = \frac{1.214(s+94.553)}{(s+12.834)(s+20.691)} \quad (63)$$

$$\frac{W_3(s)}{1(s)} = \frac{0.333(s+23.255)(s+327.891)}{(s+14.122)(s+19.512)(s+21.308)} \quad (64)$$

$$\frac{W_4(s)}{1(s)} = \frac{5.019 \times 10^{-3} e^{-2.116s}}{(s+10.370)} \quad (65)$$

Para cada caso se presenta la respectiva gráfica del lugar de raíces (figuras 38 a 42) en el plano complejo s , para observar las diferencias que existen entre ellos.

En el caso de las aproximaciones anteriores de dos y tres polos, es mejor representar las respectivas funciones de transferencia separándolas en factores mas simples, para facilitar de este modo el siguiente paso a la correspondiente transformada z . Es decir, convertiremos las funciones a:

$$\begin{aligned} \frac{W_2(s)}{1(s)} &= \frac{\alpha_{21}}{s+\tau_1} + \frac{\alpha_{22}}{s+\tau_2} \\ \frac{W_3(s)}{1(s)} &= \frac{\alpha_{31}}{s+\tau_1} + \frac{\alpha_{32}}{s+\tau_2} + \frac{\alpha_{33}}{s+\tau_3} \end{aligned} \quad (66)$$

donde las alfas se hallan por el método de expansión en fracciones parciales, quedando, en el caso de dos polos:

$$\alpha_{21} = \frac{B_0(C_0 - \tau_1)}{(-\tau_1 + \tau_2)} \quad (67)$$

$$\alpha_{22} = \frac{B_0(C_0 - \tau_2)}{(-\tau_2 + \tau_1)} \quad (68)$$

y en el caso de la función de transferencia de tres polos:

$$\alpha_{31} = \frac{B_0(C_1 - \tau_1)(C_2 - \tau_1)}{(-\tau_1 + \tau_2)(-\tau_1 + \tau_3)} \quad (69)$$

$$\alpha_{32} = \frac{B_0(C_1 - \tau_2)(C_2 - \tau_2)}{(-\tau_2 + \tau_1)(-\tau_2 + \tau_3)} \quad (70)$$

$$\alpha_{33} = \frac{B_0(C_1 - \tau_3)(C_2 - \tau_3)}{(-\tau_3 + \tau_1)(-\tau_3 + \tau_2)} \quad (71)$$

que al sustituir los valores numéricos conocidos, quedan, para dos polos:

$$\alpha_{21} = 12.627$$

$$\alpha_{22} = -11.413$$

por lo que al sustituir queda:

$$\frac{W_2(s)}{I(s)} = \frac{12.627}{(s+12.834)} + \frac{-11.413}{(s+20.691)} \quad (72)$$

y para tres polos:

$$\alpha_{31} = 24.637$$

$$\alpha_{32} = -39.706$$

$$\alpha_{33} = 15.402$$

y ahora al sustituir queda:

$$\frac{W_3(s)}{I(s)} = \frac{24.637}{(s+14.122)} + \frac{-39.706}{(s+19.512)} + \frac{15.402}{(s+21.308)} \quad (73)$$

RESUMEN

En resumen se halla que la mejor aproximación analítica a la planta real se halla con el modelo de tres polos aplicando el método logarítmico, por ser el que arroja errores mas pequeños a lo largo de la totalidad de la respuesta. La ecuación que describe tal aproximación es la (64). En este caso se halla una planta estable, cuyo polo principal está cerca de 8.3 Hz.

Por otra parte, la aproximación de un polo (ecuación (36)) es un punto de partida excelente para iniciar el diseño del controlador, ya que de otro modo se obtendría un conjunto de ecuaciones muy complejo para resolver. En este caso, el polo único está cerca de 12 Hz.

Por este motivo, se usarán dichas aproximaciones para ilustrar el cálculo de los controladores digitales P, PI y PID.

Una aproximación más exacta a la función de transferencia requeriría muchas más pruebas y trabajo algebraico y estadístico, sin que reporte grandes beneficios, ya que de cualquier forma el diseño de los controladores digitales será en gran parte empírico, y puede cambiarse de un momento a otro para modificar la respuesta del sistema en lazo cerrado.

§

4 CONTROLADORES DIGITALES P, PI Y PID

TIEMPO DE MUESTREO

Para determinar el tiempo de muestreo en lazo cerrado, se aplicará el programa controlador a la planta, dando inicialmente valores arbitrarios a las constantes del algoritmo controlador, exclusivamente con el propósito de hallar el tiempo que necesita el computador para muestrear, hacer los cálculos del algoritmo y luego aplicar el valor digital de salida, es decir para efectuar el ciclo completo de control digital.

Para hacer esto, se selecciona del menú principal el programa Control (núm. 8), donde aparece la opción de seleccionar lazo abierto o lazo cerrado. Se elige cerrado oprimiendo la tecla C, luego se nos pide el valor de referencia que se desea tener a la salida de la planta (en este caso, elegiremos 3 volts), y al pedírsenos los valores para las constantes del algoritmo, K_P , K_I y K_D , daremos los valores 10, 10, 10.

Ahora es posible dar el número de ciclos que la computadora aplicará el algoritmo de control a la planta (2-100). Se seleccionan 100. Al oprimir la tecla de espacio la computadora efectúa la serie de ciclos y mide el tiempo total que requirió para hacerlo, tras lo cual divide dicho tiempo entre el número de ciclos (100), y despliega el valor que se fijó como referencia, el valor final de salida, el valor final del controlador, y el promedio de tiempo que necesitó por cada ciclo, es decir, el tiempo de muestreo.

Con los valores citados antes, se halla que el tiempo de muestreo promedio para el algoritmo completo (PID), es de 3.55 centisegundos. Este es el valor que se usará de aquí en adelante para el cálculo de los controladores. Es útil recordar que el tiempo de muestreo en lazo abierto fué de 1.198 cs.

Nótese también que no es posible reducir significativamente el valor de este tiempo de muestreo, a menos que se sacrificara la facilidad de usar un lenguaje estructurado de alto nivel, como lo es BASIC, y en su lugar usar lenguaje ensamblador del microprocesador 6502, en cuyo caso habría que compilar primero dichas rutinas para el cálculo del algoritmo.

De cualquier manera, el tiempo de muestreo en lazo cerrado jamás podrá ser menos que el tiempo de muestreo en lazo abierto (1.2 cs), que está limitado ya por el hardware de la computadora, es decir, por la velocidad con que el CAD interno de la MicroBBC es capaz de convertir valores analógicos a digitales, inclusive, al agregar los cálculos, aunque estén en ensamblador, se requerirá un tiempo finito para su determinación, por lo que el tiempo de muestreo en lazo cerrado, en tal caso, sería todavía de unos 2.5 cs o tal vez más.

Es importante hacer notar que frecuencias de muestreo por debajo de cinco veces la frecuencia natural del sistema, podrían hacer que la planta se volviese inestable. Se recomiendan tiempos de muestreo de al menos diez o veinte veces la frecuencia natural de la planta, para poder usar controladores continuos con total confianza. {Franklin et al., 555}.

TRANSFORMADA Z

Podemos representar la función de transferencia en forma digital (en la variable compleja z), para las aproximaciones de uno, dos y tres polos.

Sin embargo, no es posible dar una representación válida de la función $W(s)$ (un polo con un retraso), hallada antes con el método de Ziegler y Nichols, debido a que el tiempo de muestreo en lazo cerrado que podemos utilizar es muy grande, comparado con la constante de tiempo T_I del sistema. Existiría dicha función en z si pudiésemos seleccionar el tiempo de muestreo T del tal manera que:

$$L = NT$$

donde N es un número entero. {Takahashi et al., 485 y ss.}. Es decir, el retardo de transporte debe ser un múltiplo entero del tiempo de muestreo. En cuyo caso, dado:

$$W(s) = \frac{R e^{-Ls}}{T_1 s + 1}$$

se tendría:

$$W(z) = z^{-N} \left(\frac{R (1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} \right)$$

Como en nuestro caso particular tenemos:

$$L = 2.116 \times 10^{-2} \text{ seg}, T = 3.55 \times 10^{-2} \text{ seg}; L/T = 0.596$$

no es un número entero, por lo cual no podemos acceder a la definición anterior.

Para representar los tres casos hallados con el método logarítmico (ecs. 36, 72 y 73), utilizaremos la siguiente transformación: {Franklin et al., 545, 556}

$$W(z) = (1 - z^{-1}) Z \left(\frac{W(s)}{s} \right) \quad (74)$$

De esta manera, haciendo las correspondientes transformaciones algebraicas, y considerando que el tiempo de muestreo en lazo cerrado es:

$$T = 3.55 \times 10^{-2} \text{ seg}$$

$$(T = 1.198 \times 10^{-2} \text{ seg en lazo abierto})$$

las funciones de transferencia en forma digital quedan:

$$W_1(z) = \frac{A_0 (1 - e^{-\frac{T}{T_1}})}{T_1 (z - e^{-\frac{T}{T_1}})} \quad (75)$$

$$W_2(z) = \frac{\alpha_{21}(1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} + \frac{\alpha_{22}(1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} \quad (76)$$

$$W_3(z) = \frac{\alpha_{31}(1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} + \frac{\alpha_{32}(1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} + \frac{\alpha_{33}(1 - e^{-\frac{T}{T_1}})}{z - e^{-\frac{T}{T_1}}} \quad (77)$$

en estos casos, hagamos las simplificaciones de notación:

$$\beta_i = \alpha_i (1 - e^{-\frac{T}{T_i}})$$

$$\gamma_i = e^{-\frac{T}{T_i}}$$

de modo que se llega a:

$$W_1(z) = \frac{\beta_1}{z - \gamma_1} \quad (78)$$

$$W_2(z) = \frac{\beta_{21}}{z - \gamma_{21}} + \frac{\beta_{22}}{z - \gamma_{22}} \quad (79)$$

$$W_3(z) = \frac{\beta_{31}}{z - \gamma_{31}} + \frac{\beta_{32}}{z - \gamma_{32}} + \frac{\beta_{33}}{z - \gamma_{33}} \quad (80)$$

al sustituir valores quedan:

$$W_1(z) = \frac{9.185 \times 10^{-3}}{z - 0.7444} \quad (81)$$

$$W_2(z) = \frac{4.621}{z - 0.634} + \frac{-5.938}{z - 0.480} \quad (82)$$

$$W_3(z) = \frac{9.714}{z - 0.606} + \frac{-19.844}{z - 0.500} + \frac{8.173}{z - 0.469} \quad (83)$$

Ahora estas funciones se vuelven a factorizar para hallar los correspondientes ceros en el plano complejo z . Haciendo las manipulaciones algebraicas correspondientes las ecuaciones quedan:

$$W_2(z) = \frac{\delta_{20}(z + \delta_{21})}{(z - \gamma_{21})(z - \gamma_{22})} \quad (84)$$

$$W_3(z) = \frac{\delta_{30}(z^2 + \delta_{31}z + \delta_{32})}{(z - \gamma_{31})(z - \gamma_{32})(z - \gamma_{33})} \quad (85)$$

donde γ_i son los polos de las respectivas funciones, en el plano z .

Para la función de dos polos las deltas son:

$$\delta_{20} = \beta_{21} + \beta_{22} \quad (86)$$

$$\delta_{21} = \frac{\beta_{21}\gamma_{22} + \beta_{22}\gamma_{21}}{\delta_{20}} \quad (87)$$

que al sustituir valores quedan:

$$\delta_{20} = -1.317$$

$$\delta_{21} = -1.174$$

Para la función de tres polos, los valores de las deltas son:

$$\delta_{30} = \beta_{31} + \beta_{32} + \beta_{33} \quad (88)$$

$$\delta_{31} = -\delta_{30} (\beta_{31}(\gamma_{32} + \gamma_{33}) + \beta_{32}(\gamma_{31} + \gamma_{33}) + \beta_{33}(\gamma_{31} + \gamma_{32})) \quad (89)$$

$$\delta_{32} = \delta_{30} (\beta_{31}\gamma_2\gamma_3 + \beta_{32}\gamma_1\gamma_3 + \beta_{33}\gamma_1\gamma_2) \quad (90)$$

cuyos valores, al sustituir datos quedan:

$$\delta_{30} = -1.957$$

$$\delta_{31} = -1.472$$

$$\delta_{32} = +0.453$$

Dicha ecuación de tres polos se podría factorizar aún mas, ahora así:

$$W_3(z) = \frac{\delta_{30}(z + \lambda_{31})(z + \lambda_{32})}{(z - \gamma_1)(z - \gamma_2)(z - \gamma_3)} \quad (91)$$

donde las λ_{3i} son los ceros de la función (que podrían ser reales o complejos conjugados). Es fácil llegar a que:

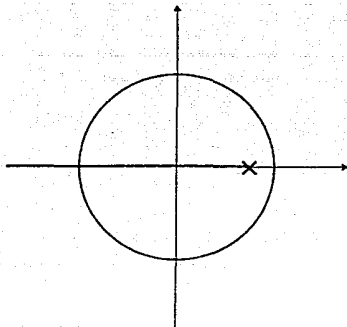


Fig. 43. Lugar de raíces discreto. Un solo polo.

$$\lambda_{-31,32} = \frac{-\delta_{31} \pm \sqrt{\delta_{31}^2 - 4\delta_{32}}}{2}$$

(92)

Al sustituir los valores numéricos queda el siguiente resumen:

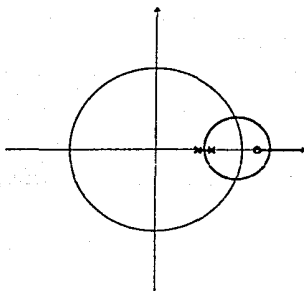


Fig. 44. Lugar de raíces discreto. Dos polos.

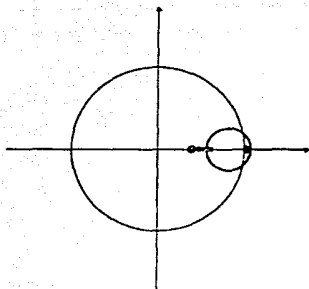


Fig. 45 Lugar de raíces discreto. Tres polos.

$$W_1(z) = \frac{9.185 \times 10^{-3}}{z - 0.744} \quad (93)$$

$$W_2(z) = \frac{-1.317(z - 1.174)}{(z - 0.634)(z - 0.480)} \quad (94)$$

$$W_3(z) = \frac{-1.957(z - 1.034)(z - 0.438)}{(z - 0.606)(z - 0.500)(z - 0.469)} \quad (95)$$

Los respectivos gráficos del lugar de raíces discretos se muestran en las figuras 43 a 45.

DESCRIPCION DE LOS ALGORITMOS

Los algoritmos de control utilizados en nuestro sistema, al igual que los usados en una gran variedad de sistemas de propósito general, se basan en el tipo PID (Proporcional, Integral, Derivativo), el cual ha sido el principal algoritmo para el control de procesos desde hace más de medio siglo, cuando se diseñaron para procesos neumáticos. {Auslander et al., 1,3}.

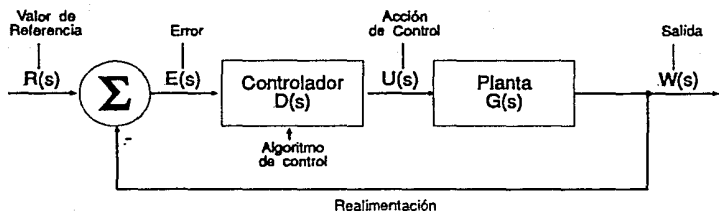


Fig. 46 Planta continua con controlador D(s).

Estos algoritmos son a menudo satisfactorios en la mayoría de las plantas y se pueden aplicar y modificar con facilidad en la computadora digital, en un ambiente de bajo poder de cómputo. {Alvarez y Alvarez, 7.1; Auslander et al., 1,3; Auslander et al. (ISC), 276}. Ver la figura 46.

Acción Proporcional

El control más simple es el de tipo proporcional (también llamado acción P) el cual es una ganancia estática, ignorando las condiciones dinámicas en el mecanismo del controlador (si bien tiene un efecto en ellas), aplicada a la diferencia entre la variable de referencia y el valor medido a la salida de la planta, es decir que la acción proporcional solo depende de la magnitud del error en ese instante. {Takahashi et al., 190; Richards, 176}.

La función de transferencia del controlador en general se define como:

$$D(s) = \frac{U(s)}{E(s)} \quad (96)$$

donde, en el caso de la acción proporcional:

$$U(s) = k_p E(s) \quad (97)$$

y además (ver fig. 46):

$$E(s) = R(s) - W(s) \quad (98)$$

En estas ecuaciones, $U(s)$ es la acción de control propiamente dicha, que se aplica a la entrada de la planta, $E(s)$ es el error que resulta de restar $W(s)$ de $R(s)$, es decir el valor de la salida de la planta restado del valor de referencia fijado por el operador. Este error se aplica como entrada al controlador.

Es importante hacer notar que el control proporcional puro es inaplicable en sistemas digitales, ya que produce resultados adversos (inestabilidad), esto se debe a la naturaleza del control

proporcional en sí, emporado por los efectos del muestreo. Incluso en plantas continuas puede producir oscilaciones sostenidas. {Franklin et al., 560}.

En sistemas neumáticos, hay esencialmente dos tipos de elementos que ejercen acción proporcional: los de fuerza-distancia y los de fuerza-equilibrio, y se usan ya sea en la trayectoria directa o en el lazo de realimentación. En el caso de sistemas electrónicos se utilizan amplificadores de tensión para ejercer la acción proporcional. {Ogata, 175, 186; Takahashi et al., 190}. En nuestro caso, la computadora se encarga de ejercer ésta y otras acciones de control.

El efecto del control P es mas evidente en la parte transitoria de la respuesta, ya que incrementa el sobretiro, si bien produce el error de estado estable (también llamado *offset*). {Auslander et al., 201; Chesmond, 186; Takahashi et al., 190}.

Dicho de otro modo, la ganancia proporcional pura produce sistemas tipo cero, que son aquellos que presentan error estático, que es una diferencia constante entre la señal de referencia y la variable de salida, o bien una incapacidad de seguir una entrada escalón. {Richards, 176}.

La ganancia proporcional, por sí sola, incrementa la velocidad de la respuesta y disminuye el amortiguamiento, por otra parte, valores grandes de ganancia proporcional, sobre todo en sistemas de orden superior, pueden causar inestabilidad a pesar de que disminuyen el error de estado permanente, y para la mayoría de los casos reales existe un límite superior práctico para la magnitud de esta ganancia, para poder tener una planta estable y a la vez moderadamente amortiguada. {Franklin et al., 96; Richards, 177}.

Acción Integral

Otro tipo de controlador que se puede utilizar es el tipo integral (también llamado acción I), que se basa en la integral de los errores pasados. Casi siempre se combina con el control proporcional, o bien con el proporcional y el derivativo a la vez. De no ser así, se le llama también control "flotante" {Chesmond, 185}. Su definición, utilizando la variable compleja s es:

$$U(s) = \frac{k_i}{s} E(s) \quad (99)$$

Este controlador integral elimina el error de estado estable y reduce el sobretiro. Su efecto es similar a una red de atraso, cuando se combina con la acción proporcional. {Takahashi et al., 386; Richards, 261}

La acción integral hace que se eleve el orden de la ecuación característica, por lo que produce sistemas de tipo 1, es decir que se elimina el *offset* o error de estado estacionario. Este tipo de acción mejora la respuesta dinámica del sistema, si bien obliga a reducir el valor de la constante proporcional para conservar la estabilidad. Por otra parte, la frecuencia natural del sistema decrece un poco y aumenta el tiempo de asentamiento. {Franklin et al, 100; Richards, 178, 184, 261}.

Acción Derivativa

El control derivativo (acción D) es una ganancia dinámica que responde a la tasa de cambio del error de/dt . Siempre se usa junto con la acción P y su efecto es colaborar a la estabilidad del sistema, al añadir un cero de lazo abierto a la izquierda del lugar de las raíces. Su efecto es similar a una red de adelanto, cuando se combina con dicho control proporcional. También mejora la

respuesta transitoria en general, ya que reduce las oscilaciones y reduce el tiempo de asentamiento. {Franklin et al., 100; Takahashi et al., 336,386; Richards, 264}.

Sin embargo, este tipo de acción debe usarse con cuidado cuando la señal de error tiene mucho ruido, ya que éste tendería a ser amplificado. {Chesmond, 184}. La relación que define la acción derivativa es:

$$U(s) = k_d s E(s) \quad (100)$$

Acción Proporcional, Integral y Derivativa (PID)

Los tres tipos se pueden combinar para dar una ley general de control del tipo PID, pudiéndose implantar como elementos electrónicos o neumáticos, o bien como un programa de computadora digital (en el caso discreto). {Chesmond, 179; Takahashi et al., 196}.

$$U(s) = k_p E(s) + \frac{k_i}{s} E(s) + k_d s E(s) \quad (101)$$

desde luego, otra forma de expresar esta ley de control en el dominio del tiempo sería: {Auslander et al. (ICS), 277}

$$u(t) = k_c \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d}{dt} e(t) \right] \quad (102)$$

Donde k_c corresponde a la ganancia proporcional, k_c / T_i a la ganancia integral y $k_c T_d$ a la ganancia derivativa.

En el caso digital, el controlador se convierte en una serie de instrucciones de un programa corriendo dentro del ordenador, representando una ecuación en diferencias, para calcular el valor de salida. {Chesmond, 318}. Esta ecuación en diferencias se define en detalle mas adelante.

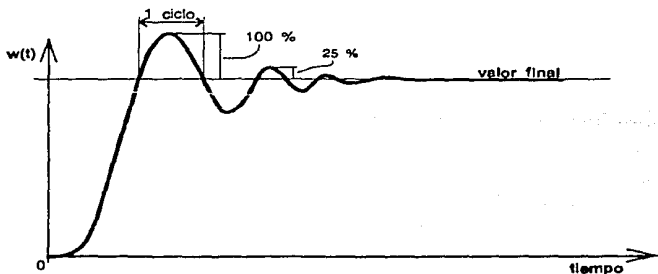


Fig. 47 Respuesta óptima de un sistema con controlador PID.

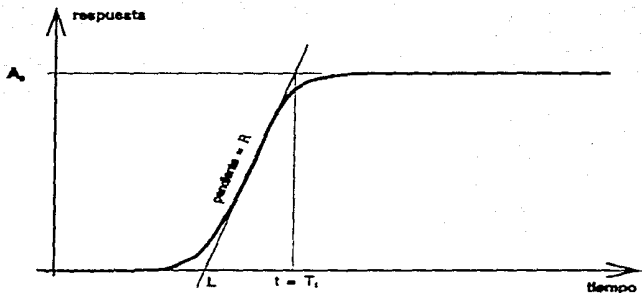


Fig. 48 Curva de reacción de Ziegler y Nichols.

Este tipo de controladores (P, PI, PID) fueron retomados y generalizados por Ziegler y Nichols en los 1950's, tomando como base la optimización (minimización) de la integral del valor absoluto del error ($r(t) \rightarrow y(t)$); este diseño dá un modo dominante de oscilación con una respuesta de un cuarto de amortiguamiento (del 100% al 25% en un ciclo), que normalmente es un buen compromiso entre respuesta rápida y estabilidad adecuada. Ver figura 47. {Takahashi et al., 344,345}

DISEÑO DEL CONTROLADOR CONTINUO

De acuerdo a la curva de reacción que Ziegler y Nichols definieron (fig. 48), con los parámetros de la curva de reacción R , L y T_I , detallados en el capítulo anterior, es posible diseñar una primera aproximación a un bloque controlador del tipo P, PI o PID, {Alvarez y Alvarez, 7.1; Takahashi et al., 343} cuya forma y parámetros (para el caso continuo) se definen a continuación.

La ecuación del controlador PID es:

$$\frac{U(s)}{E(s)} = D(s) = k_c \left(1 + \frac{1}{T_I s} + T_d s \right) \quad (103)$$

Las constantes de la ecuación se hallan con la siguiente tabla (la num. 3), si se usa dicha aproximación de Ziegler y Nichols al modelo (continuo) de la planta:

	k_c	T_i	T_d
P	$\frac{1.0}{RL}$		
PI	$\frac{0.9}{RL}$	$3.3L$	
PID	$\frac{1.2}{RL}$	$2.0L$	$0.5L$

Tabla 3. Método de Ziegler - Nichols (continuo)

aquí se ve que al aparecer el término integral, que reduce un poco la estabilidad, es necesario disminuir el término proporcional para conservar ésta, mientras que al agregarse el término derivativo, con efectos estabilizadores, de nuevo es posible elevar el término proporcional y reducir el valor de T_i , (aumentar la constante integral).

Ya que la determinación de estos valores es meramente empírica (experimental), normalmente se toman solo como aproximaciones que deben ajustarse después. De hecho, la absoluta optimización de los valores del controlador por este método no está garantizada. Con un método propuesto un método —también experimental— más detallado (y más complejo) para calcular estas ganancias. {Richards, 282}.

DISEÑO DEL CONTROLADOR DISCRETO

Para el caso de un controlador discreto, sin embargo, es necesario redefinir el controlador en función del tiempo de muestreo, y utilizar un algoritmo de velocidad (como se define adelante), el cual es generalmente preferido por ser más simple de calcular, más seguro y más fácil de aplicar con la computadora. {Auslander et al. (ICS), 277; Takahashi et al., 476}

De esta manera, nos queda una ecuación de controlador en forma discreta (ecuación en diferencias), con algoritmo de velocidad, de la siguiente manera, que llamaremos ecuación (104): {Alvarez y Alvarez, 7.5; Auslander et al. 201; Auslander et al. (ICS), 277; Richards, 438; Takahashi et al., 477}

$$u(k) - u(k-1) = K_P \{y(k-1) - y(k)\} + K_I \{r(k) - y(k)\} + K_D \{2y(k-1) - y(k-2) - y(k)\}$$

donde $r(k)$ es la variable de referencia (setpoint), $y(k)$ es la salida de la planta y $u(k)$ es la salida del controlador (variable manipulada). Aquí se ve que el algoritmo calcula el incremento de $u(k)$, más que $u(k)$ misma, por ello se llama algoritmo de velocidad. También es importante notar la presencia de los términos $2y(k-1)$, $-y(k-2)$, que son los valores de la salida de la planta en instantes muestrales anteriores. Hacer también referencia a la fig. 46.

Es digno de recalcar que se supone que el valor $r(k)$ se considera constante, lo cual —sin ser del todo correcto— simplifica grandemente la ecuación del controlador, además de que esto hace que no se presente un “brinco” derivativo en el caso de un cambio repentino de valor de referencia; por otra parte, también debe apreciarse que por la simplificación anterior, el término $r(k)$ solo aparece en el término integral, por lo cual dicho término siempre debe estar presente en el algoritmo, es decir que, en el caso discreto, siempre deberá implantarse la acción I, PI, o PID. Si se desea eliminar el término integral del algoritmo, entonces la simplificación de $r(k)$ no se puede usar. Esta restricción no se aplica en el caso continuo. {Auslander et al. (ICS), 277; Franklin et al., 560; Takahashi et al., 477}.

Utilizando la definición dada antes (104) al controlador PID, los valores discretos de K_P , K_I y K_D , quedan definidos en función de los parámetros continuos (k_c , T_i , T_d) hallados antes por métodos tradicionales, (por ejemplo, lugar de las raíces). {Auslander et al. (ICS), 277}.

$$K_P = k_c \left(1 - \frac{1}{2} \right) \quad K_I = k_c \frac{T_i}{T} \quad K_D = k_c \frac{T_d}{T} \quad (105)$$

en ellos el valor T es el tiempo de muestreo.

De otra manera, utilizando la aproximación de Ziegler y Nichols al modelo discreto de la planta, los coeficientes K_P , K_I , K_D no solo dependen de los parámetros originales k_c , T_i y T_d , además del tiempo de muestreo T , sino también del parámetro L (retardo) como se vé en la tabla siguiente.

Es importante notar, sin embargo, que esta aproximación no será útil en nuestro caso, por no tenerse un tiempo de muestreo (T) al menos un orden de magnitud inferior a la menor constante de tiempo de la planta. {Alvarez y Alvarez, 7.6; Richards, 438}

Algoritmo	K_P	K_I	K_D
P	$\frac{1}{R(L+T)}$		
PI	$\frac{0.9}{R\left(L+\frac{T}{2}\right)} - \frac{1}{2}K_I$	$\frac{0.27T}{R\left(L+\frac{T}{2}\right)^2}$	
PID	$\frac{1.2}{R\left(L+\frac{T}{2}\right)} - \frac{1}{2}K_I$	$\frac{0.6T}{R\left(L+\frac{T}{2}\right)^2}$	$\frac{0.6}{RT}$
notas	(1,3)	(1,3)	(2,3)
<p>NOTAS</p> <p>(1) excepto si L/T tiende a 0 en cuyo caso se decrecen K_P y K_D</p> <p>(2) excepto si L/T es casi entero en cuyo caso $K_D = 0.5/RT$</p> <p>(3) en cualquier caso la frecuencia de muestreo $1/T$ debe ser grande comparada con la mayor frecuencia de la planta.</p>			
<p><i>Tabla 4. Método de Ziegler - Nichols, controlador discreto.</i></p>			

DISCRETIZACION

Hay otros enfoques al diseño de los controladores digitales. Unos se basan en la determinación inicial del controlador analógico $D(s)$ para la planta continua, para luego discretizar el resultado obtenido, a través de transformaciones de los polinomios en s a polinomios en z , de modo que el controlador se exprese como $D(z)$. La relación $D(z)$ hallada así no necesariamente es única, depende de la elección que se haga del método de transformación. {Franklin et al., 548}.

Algunas de las transformaciones posibles y mas comunes se listan a continuación. (tabla num. 5) {Chesmond, 317}.

	s	s^2	s^3
Método	<i>intercambiar la potencia de s por:</i>		
Primera diferencia	$\frac{1-z^{-1}}{T}$	$\left(\frac{1-z^{-1}}{T}\right)^2$	$\left(\frac{1-z^{-1}}{T}\right)^3$
Transformada Z	$\frac{1-z^{-1}}{T}$	$\frac{(1-z^{-1})^2}{T^2 z^{-1}}$	$\frac{2(1-z^{-1})^3}{T^3 z^{-1}(1+z^{-1})}$
Tustin	$\frac{2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}{T}$	$\frac{4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2}{T^2}$	$\frac{8\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^3}{T^3}$
Boxer Thaler	$\frac{2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}{T}$	$\frac{12(1-z^{-1})^2}{T^2(1+10z^{-1}+z^{-2})}$	$\frac{2(1-z^{-1})^3}{T^3 z^{-1}(1+z^{-1})}$
Madwed Truxal	$\frac{2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}{T}$	$\frac{6(1-z^{-1})^2}{T^2(1+4z^{-1}+z^{-2})}$	$\frac{24(1-z^{-1})^3}{T^3(1+11z^{-1}+11z^{-2}+z^{-3})}$

Tabla 5. Discretización de potencias.

Por otra parte, la frecuencia de muestreo debe ser muy superior a la frecuencia mayor del sistema, para que la aproximación sea válida.

Franklin cita principalmente la regla de Tustin, que es básicamente una fórmula de integración numérica, y un método, también aproximado, de mapeo de polos y ceros del plano s al plano z , mediante la aplicación directa de la definición $z=e^{sT}$ para luego igualar las ganancias de baja frecuencia. Este método de mapeo tiene también una variante, que consiste en añadir ceros en $z=1$ al numerador de $D(z)$, para lograr un promedio de valores de entrada anteriores con actuales, al igual que en la regla de Tustin. Este método de mapeo es el que requiere menos algebra de todos, por lo cual normalmente es el preferido. {Franklin et al., 548-554}.

Cada implementación dará resultados distintos, que podrían resultar mas apropiados en algunos casos que en otros, ya que debido a los procesos de muestreo y cuantización, no hay una $D(z)$ físicamente realizable que exhiba una respuesta a la frecuencia idéntica a $D(s)$ en todas las frecuencias, por lo tanto $D(z)$ debe escogerse de acuerdo al rango de frecuencias de interés. {Chesmond, 317}.

DISEÑO EN EL PLANO Z

Como se mencionó arriba, una alternativa para el cálculo de controladores digitales, es diseñar controladores analógicos y luego representar éstos en la forma digital que mejor les corresponda. Dicha opción solo es recomendable cuando la frecuencia de muestreo es muy superior a la máxima frecuencia del sistema, (al menos 10 o 20 veces mayor) de otro modo, sólo debe considerarse como una primera aproximación a la solución final en cuanto a la síntesis de un algoritmo digital. {Franklin et al., 556}.

Otra alternativa de diseño mas adecuada, y mas general, particularmente cuando la frecuencia de muestreo no es muy superior a la mayor frecuencia del sistema, es considerar el sistema en su totalidad como una entidad discreta, en cuyo caso lo que se convierte de la forma continua a la discreta es la función de transferencia de la planta $G(s)$, expresándose como $G(z)$, normalmente mediante la transformación: {Franklin et al., 556}

$$G(z) = (1 - z^{-1}) Z\left(\frac{G(s)}{s}\right) \quad (106)$$

Una vez hecha la transformación, es posible sintetizar directamente los parámetros del controlador digital, con métodos y técnicas similares a las usadas en el plano complejo s , (lugar de las raíces, por ejemplo) pero adaptadas al plano z , para luego pasar a una ecuación en diferencias que es fácilmente implantable en la computadora digital.

De cualquier forma, el diseño de sistemas de control requiere inevitablemente de un proceso de tanteo y modificación sucesiva, hasta que se alcancen los objetivos prefijados, ya que tal vez no haya un sistema único que satisfaga nuestras necesidades. Es más, puede haber muchos (incluso infinitos, o ninguno) sistemas que cumplan los requerimientos dados. {Ogata, 519}. Esto significa que hay que seguir un proceso de ensayo y error para la selección de valores que rindan un funcionamiento adecuado de la planta. {Richards, 184}.

CALCULO DE LAS CONSTANTES PID

Para el cálculo de las constantes del controlador digital, utilizaremos la función de transferencia continua de un solo polo, hallada en el capítulo anterior. Los ajustes de dichos valores se pueden hacer en la práctica.

Si se descara utilizar la función de transferencia continua de tres polos, se obtendría un sistema de ecuaciones muy complejo y difícil de resolver, ya que el controlador completo PID representa dos polos de lazo cerrado, mientras que la función de transferencia tendría de por sí tres polos más, por lo que habría que resolver un polinomio de quinto grado, al tener que asignar cinco raíces (polos de lazo cerrado), de los cuales una es una raíz real, y las otras cuatro pueden ser reales o bien pares complejos conjugados. Un diseño en estas circunstancias es altamente impráctico. {Takahashi et al., 329}.

Para convertir la función de transferencia original de la forma continua a la respectiva forma discreta, se utiliza la transformación mencionada antes (106), de la manera descrita a continuación.

NOTA: En el capítulo anterior se llamaba $W(s)/I(s)$ a las funciones de transferencia, por ser la razón de velocidad angular del motor al escalón de voltaje de entrada. Llamaremos $G(s)$ a la definición de esta relación, que es, propiamente, la función de transferencia de la planta.

Dado:

$$G(s) = \frac{A_0}{T_1} \frac{1}{s + \frac{1}{T_1}} \quad (107)$$

se aplica la transformada Z, {Franklin et al., 545} y se obtiene:

$$G(z) = \frac{A_0}{T_1} \frac{1 - e^{-\frac{T}{T_1}}}{z - e^{-\frac{T}{T_1}}} \quad (108)$$

haciendo las equivalencias:

$$\beta = \frac{A_0}{T_1} (1 - e^{-\frac{T}{T_1}})$$

$$\gamma = e^{-\frac{T}{T_1}}$$

en las que tendremos $\beta > 0$, $\gamma > 0$, ya que $0 < T < T_1$, con lo que $0 < e^{-\frac{T}{T_1}} < 1$, lo cual es una condición necesaria para tener estabilidad (polo dentro del círculo unitario en el plano z).

De esta manera podremos poner la función de transferencia discreta $G(z)$ de la siguiente manera:

$$G(z) = \frac{\beta}{z - \gamma} \quad (109)$$

Una vez hecho esto, se sustituyen los valores conocidos de A_0 y T_1 , hallados antes, y el valor del tiempo de muestreo T , que se halló por experimentación, para tener los valores de β y γ .

$$A_0 = 0.4324 \text{ volts}, T_1 = 12.03 \times 10^{-2} \text{ seg}, T = 3.55 \times 10^{-2} \text{ seg}$$

con lo que se encuentra:

$$\beta = 9.185 \times 10^{-3} \text{ volts/seg}$$

$$\gamma = 744.460 \times 10^{-3} \text{ (adimensional)}$$

es decir que la función $G(z)$ tiene un polo de lazo abierto en $z=0.744$. Como el numerador de la función de transferencia no depende de z , se sigue que se tiene un cero en el infinito, quedando un diagrama de lugar de las raíces en el plano complejo z como se ve en la figura (43).

Por otra parte, se define el controlador digital $D(z)$, en su forma mas general (proporcional, integral y derivativo), así:

$$D(z) = K_P + K_I \frac{z}{z-1} + K_D \frac{z-1}{z} \quad (110)$$

el cual ya se había definido antes en función del tiempo y de la variable compleja s .

Para obtener los valores de K_P , K_I y K_D , seleccionaremos una por una las acciones de control, y se hará una síntesis del controlador en lazo cerrado en cada caso.

ACCION PROPORCIONAL

Aquí el controlador digital (110) se simplificará, dejándolo como una acción proporcional pura, quedando:

$$D(z) = K_P \quad (111)$$

En lazo cerrado, se tendría que la función de transferencia es (ver fig. 46 al inicio del capítulo):

$$\frac{W(z)}{R(z)} = \frac{D(z)G(z)}{1+D(z)G(z)} \quad (112)$$

donde la ecuación característica, cuyas raíces son los polos de lazo cerrado, es:

$$1 + D(z)G(z) = 0 \quad (113)$$

al substituir las ecuaciones para $G(z)$ y $D(z)$ queda:

$$1 + K_P \frac{\beta}{z-\gamma} = 0$$

es decir:

$$z - \gamma + \beta K_P = 0 \quad (114)$$

y para que haya estabilidad es necesario que el polo en el plano z esté dentro del círculo unitario, por lo que:

$$\left| -\gamma + \beta K_P \right| < 1$$

por lo que las opciones disponibles son:

$$\begin{aligned} -\gamma + \beta K_P &< 1 \\ -(-\gamma + \beta K_P) &< 1 \end{aligned}$$

y de allí que, despejando K_P en ambos casos, los valores límites que se hallan son:

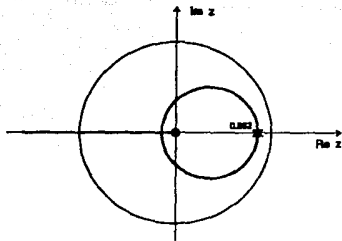


Fig. 50 Acción Integral. Dos raíces reales iguales.

ACCION INTEGRAL

Ahora la ecuación del controlador se reduce a la acción integral pura, es decir:

$$D(z) = K_I \frac{z}{z-1} \quad (116)$$

por lo tanto, la ecuación característica $1 + D(z)G(z) = 0$ será:

$$1 + K_I \frac{z}{z-1} \frac{\beta}{z-\gamma} = 0$$

que al efectuar las manipulaciones algebraicas correspondientes queda:

$$z^2 + z(\beta K_I - (\gamma + 1)) + \gamma = 0 \quad (117)$$

cuyas raíces pueden ser ambas reales (iguales o distintas), $z = a, b$, en cuyo caso la ecuación cuadrática queda de la forma:

$$(z+a)(z+b) = z^2 + (a+b)z + ab \quad (118)$$

al igualar coeficientes y despejar K_I y el polo b (en función del a), tendríamos que:

$$b = \frac{\gamma}{a} \quad (119)$$

$$K_I = \frac{a+b+\gamma+1}{\beta} \quad (120)$$

Para que haya estabilidad, es evidente que los límites para el polo a están dados por $\gamma \leq a \leq 1$, con lo que $1 \geq b \geq \gamma$, para que ambos polos estén dentro del círculo unitario. De este modo, el valor máximo (en cualquiera de los dos extremos) para K_I es:

$$K_I = \frac{2(\gamma+1)}{\beta} \quad (121)$$

que al sustituir valores de γ y β da:

$$K_I = 379.851$$

ambos polos serían reales e iguales para los valores:

$$a = b = \sqrt{\gamma} = 862.821 \times 10^{-3}$$

con lo que se tendría:

$$K_I = \frac{2\sqrt{\gamma} + \gamma + 1}{\beta} = 377.802$$

el cual sería el valor mínimo de K_I , donde el par de polos se convierten en un polo doble. Ver fig. 50.

La otra posibilidad es que las raíces de la ecuación (117) sean un par complejo conjugado, del tipo $z = -a \pm jb$, que daría una ecuación cuadrática de la forma:

$$z^2 + 2az + (a^2 + b^2) = 0 \quad (122)$$

igualando ambas ecuaciones y luego despejando se tiene:

$$K_I = \frac{2a + \gamma + 1}{\beta} \quad (123)$$

$$b = \sqrt{\gamma - a^2} \quad (124)$$

de allí que como $\gamma \neq 0$, a y b no pueden ser ambos cero. Si $a=0$, $b=\sqrt{\gamma}$ y viceversa.

Es posible, por lo tanto, fijar el valor de a desde $-\sqrt{\gamma}$ hasta $+\sqrt{\gamma}$ (± 0.862), con lo que los límites de b van de cero a $\sqrt{\gamma}$ y de regreso a cero, y los límites para K_I son:

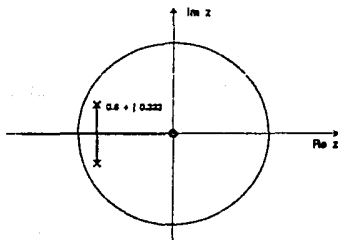


Fig. 51 Acción integral. Dos raíces complejas conjugadas.

$$\frac{-2\sqrt{\gamma}+\gamma+1}{\beta} < K_I < \frac{2\sqrt{\gamma}+\gamma+1}{\beta} \quad (125)$$

al sustituir valores, los límites son:

$$2.0498 < K_I < 377.802$$

Es decir, que si fijáramos la parte real del valor de los polos en -0.8 , tendríamos que:

$$b = 0.3231$$

$$K_I = 15.728$$

En cualquier caso, el numerador de la función de transferencia es $\beta K_I z$, es decir que existe un cero de lazo cerrado en $z=0$. Dado que hay dos polos (sean reales o complejos), debe también haber dos ceros, por lo tanto el otro cero está en el infinito, como se ve en el diagrama de lugar de las raíces de la fig. 51.

ACCION PROPORCIONAL INTEGRAL

Al aplicar la acción PI, el controlador digital queda en la forma:

$$D(z) = K_P + K_I \frac{z}{z-1} \quad (126)$$

con lo que la ecuación característica $1 + D(z)G(z) = 0$ queda:

$$1 + \left(K_P + K_I \frac{z}{z-1} \right) \frac{\beta}{z-\gamma} = 0$$

la cual al efectuar las transformaciones algebraicas queda en la forma:

$$z^2 + z(\beta(K_P+K_I) - (\gamma+1)) + (\gamma - \beta K_P) = 0 \quad (127)$$

Al igual que antes, tenemos dos opciones: un par de raíces reales (iguales o distintas), y un par de raíces complejas conjugadas.

En el primer caso usaríamos de nuevo:

$$(z+a)(z+b) = z^2 + (a+b)z + ab \quad (128)$$

Sin embargo, ahora al igualar términos y despejar K_P y K_I , tendremos:

$$K_P = \frac{\gamma - ab}{\beta} \quad (129)$$

$$K_I = \frac{a+b+ab+1}{\beta} \quad (130)$$

Ambas constantes quedan en función de los polos a, b que deseemos asignar. Considerando que los valores máximos que pueden tomar a y b son ± 1 , es posible mostrar que los valores que K_P y K_I toman, en tales casos son:

para $(a, b) = (-1, -1)$:

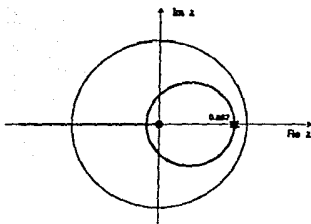


Fig. 52 Acción Proporcional Integral. Dos polos reales.

$$K_P = \frac{\gamma - 1}{\beta} = -27.821$$

$$K_I = 0$$

para valores del par (a, b) de $(1, -1)$ y $(-1, 1)$

$$K_P = \frac{\gamma + 1}{\beta} = 189.925$$

$$K_I = 0$$

y para el par $(a, b) = (1, 1)$:

$$K_P = \frac{\gamma - 1}{\beta} = -27.821$$

$$K_I = \frac{4}{\beta} = 435.494$$

de allí que los límites de K_P y K_I son:

$$-27.821 \leq K_P \leq 189.925$$

$$0 \leq K_I \leq 435.494$$

Si pusiéramos (arbitrariamente) ambas raíces iguales, $a = b = \sqrt{\gamma}$, quedarían (ver fig. 52)

$$K_P = 0$$

$$K_I = \frac{2\sqrt{\gamma} + \gamma + 1}{\beta} = 377.787$$

y si hiciéramos $a = b = 0$, quedaría:

$$K_P = \frac{\gamma}{\beta} = 81.052$$

$$K_I = \frac{1}{\beta} = 108.873$$

Si optamos por la solución de dos raíces complejas conjugadas, volveríamos a utilizar:

$$z^2 + 2az + (a^2 + b^2) = 0 \quad (131)$$

en cuyo caso, al igualar coeficientes tendríamos:

$$K_P = \frac{\gamma - (a^2 + b^2)}{\beta}$$

$$K_I = \frac{a^2 + 2a + b^2 + 1}{\beta}$$

ahí calcularemos los máximos de K_P y K_I , considerando que al ser raíces complejas conjugadas, en el borde del círculo unitario, tendremos que:

$$a^2 + b^2 = 1$$

con esto vemos que, en cualquier caso:

$$K_P = \frac{\gamma - 1}{\beta} = -27.821$$

es el mínimo, y el máximo (para $a=b=0$) es:

$$K_P = \frac{\gamma}{\beta} = 81.045$$

con lo que también podemos poner:

$$K_I = \frac{2}{\beta}(a+1)$$

y se hallan para K_I los mismos máximos y mínimos de antes, es decir:

$$0 \leq K_I \leq 435.494$$

El numerador de la función de transferencia es $D(z)G(z)$, que ahora queda:

$$z(\beta K_P + K_I) - \beta K_P$$

es decir que hay un cero finito de lazo cerrado en:

$$z = \frac{\beta K_P}{\beta K_P + K_I} \quad (132)$$

Por ejemplo, calculemos los valores de las constantes para una asignación de polos complejos en $z = -0.6 \pm j 0.2$

Los valores hallados son:

$$K_P = 37.503$$

$$K_I = 21.775$$

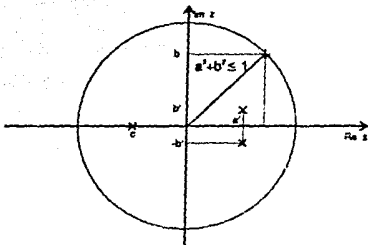


Fig. 54 Posibles polos de lazo cerrado.

al sustituir estos valores numéricos, se halla que el cero finito es: (ver fig. 53).

$$z = 15.573 \times 10^{-3}$$

Se podría generalizar la ubicación de las raíces del controlador (polos y ceros), para que haya estabilidad, como se vé en la fig. 54.

ACCION PROPORCIONAL INTEGRAL DERIVATIVA

Ahora el controlador digital tiene la forma PID completa, es decir:

$$D(z) = K_p + K_I \frac{z}{z-1} + K_D \frac{z-1}{z} \quad (133)$$

el denominador de la función de transferencia, es decir la ecuación característica, es:

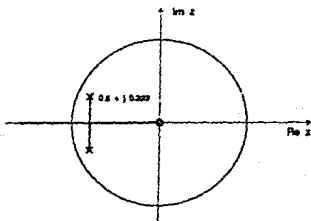


Fig. 53 Acción Proporcional Integral. Dos polos complejos.

$$1 + \left(K_P + K_I \frac{z}{z-1} + K_D \frac{z-1}{z} \right) \left(\frac{\beta}{z-\gamma} \right) = 0$$

al efectuar las transformaciones correspondientes la ecuación característica queda:

$$z^3 + z^2 (-(\gamma+1) + \beta(K_P+K_I+K_D)) + z(\gamma-\beta(K_P+2K_D)) + \beta K_D = 0 \quad (134)$$

al ser una ecuación cúbica, ahora se debe tener una raíz real y un par complejo conjugado. Si este fuera al caso, llamaremos a estas raíces:

$$z_{1,2} = -a \pm jb, \quad z_3 = -c$$

entonces la ecuación característica quedaría:

$$z^3 + z^2(2a+c) + z(a^2+b^2+2ac) + c(a^2+b^2) = 0 \quad (135)$$

igualando ambas partes se tiene que:

$$\begin{aligned} -(\gamma+1) + \beta(K_P+K_I+K_D) &= 2a+c \\ \gamma-\beta(K_P+2K_D) &= a^2+b^2+2ac \\ \beta K_D &= c(a^2+b^2) \end{aligned}$$

al despejar las constantes del controlador PID digital queda:

$$K_P = \frac{\gamma-(2c+1)(a^2+b^2)-2ac}{\beta} \quad (136)$$

$$K_I = \frac{(c+1)(a^2+b^2+2a+1)}{\beta} \quad (137)$$

$$K_D = \frac{c(a^2+b^2)}{\beta} \quad (138)$$

las limitaciones son, para tener estabilidad, que los polos propuestos estén dentro del círculo unitario, es decir que:

$$|c| < 1, \quad a^2+b^2 < 1 \quad (139)$$

al sustituir el valor $a^2+b^2 = 1$, de modo que, por inspección de las igualdades anteriores, se tienen los valores máximos de dos de las constantes del algoritmo para $c=1$, $a=1$, y sus mínimos para $c=-1$, $a=-1$, de la siguiente manera:

$$0 \leq K_I \leq \frac{8}{\beta} = 870.985 \quad (140)$$

$$-\frac{1}{\beta} = -108.873 \leq K_D \leq +\frac{1}{\beta} = 108.873 \quad (141)$$

y para K_P se tienen los máximos para $c=-1$, $a=1$, y los mínimos para $c=1$, $a=-1$, así:

$$\frac{\gamma-1}{\beta} = -27.871 \leq K_P \leq \frac{\gamma+3}{\beta} = +407.671 \quad (142)$$

Por ejemplo, fijando valores para los polos de lazo cerrado en:

$$z = -0.8 \pm j 0.2, 0.9$$

se halla que las constantes del controlador son:

$$K_P = 30.528$$

$$K_I = 16.549$$

$$K_D = 66.630$$

Al despejar los ceros del numerador $G(z)D(z)$, se llega a:

$$C_{1,2} = \frac{K_P + 2K_D \pm \sqrt{K_P^2 - 4K_D K_I}}{2(K_P + K_I + K_D)}$$

que al sustituir, describe los ceros complejos conjugados: (ver fig. 55)

$$C_{1,2} = 0.720 + j 0.259$$

Es importante recordar que no es particularmente ventajoso tratar de transplantar soluciones específicas del plano s al plano z , normalmente basta seleccionar las raíces dentro del círculo unitario en el plano z , lo cual es una tarea mucho más fácil. {Franklin et al., 561}.

Por otra parte, como se mencionó al principio, si para diseñar las constantes PID se deseara considerar la función de transferencia de lazo abierto con tres polos hallada con el método logarítmico, y aquí expresada ya en el dominio de z :

$$G(z) = \frac{a}{z-\alpha} + \frac{b}{z-\beta} + \frac{c}{z-\gamma} \quad (143)$$

y con el controlador digital completo PID:

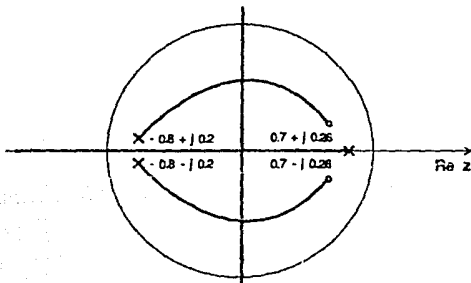


Fig. 55 Tres polos, acción PID.

$$D(z) = K_P + K_I \frac{z}{z-1} + K_D \frac{z-1}{z}$$

se tendría que trabajar con una ecuación característica de lazo cerrado (que se obtiene de $1 + D(z)G(z) = 0$) que resulta en un polinomio de quinto grado, que es:

$$z^5 + b_0 z^4 + b_1 z^3 + b_2 z^2 + b_3 z + b_4 = 0 \quad (144)$$

y cuyos coeficientes son:

$$b_0 = (K_P + K_I + K_D) - (\alpha + \beta + \gamma + 1) \quad (145)$$

$$b_1 = \left\{ \beta\gamma + (\alpha + 1)(\beta + \gamma) + \alpha - (K_P + K_I + K_D) [a(\beta + \gamma) + b(\alpha + \gamma) + c(\alpha + \beta)] \right. \\ \left. - (K_P + 2K_D)(a + b + c) \right\} \quad (146)$$

$$b_2 = \left((K_P + K_I + K_D)(a\beta\gamma + b\alpha\gamma + c\alpha\beta) + (K_P + 2K_D)(a(\beta + \gamma) + b(\alpha + \gamma) + c(\alpha + \beta)) \right. \\ \left. + K_D(a + b + c) - ((\alpha + 1)\beta\gamma + \alpha(\beta + \gamma)) \right) \quad (147)$$

$$b_3 = \alpha\beta\gamma - (K_P + 2K_D)(a\beta\gamma + b\alpha\gamma + c\alpha\beta) - K_D(a(\beta + \gamma) + b(\alpha + \gamma) + c(\alpha + \beta)) \quad (148)$$

$$b_4 = K_D(a\beta\gamma + b\alpha\gamma + c\alpha\beta) \quad (149)$$

el cual es evidentemente mucho más difícil de solucionar, ya que se tienen cinco ecuaciones simultáneas, con tres variables que son K_P , K_I y K_D .

APLICACION DEL CONTROLADOR

Los cálculos anteriores tomaron como base la aproximación a la planta con un modelo de un solo polo, si bien vimos que un modelo más exacto sería el de tres polos. Sin embargo, dicho modelo de tres polos haría ya muy complejo el cálculo de los coeficientes del controlador digital, K_P , K_I y K_D . Por ello, tales cifras halladas con la aproximación de un polo se utilizarán como punto de partida para un refinamiento manual ya durante la implantación del controlador.

Para aplicar el controlador digital recién calculado, basta correr el programa "Control", llamándolo desde el menú principal, en donde se nos pregunta si deseamos hacer control de lazo abierto o de lazo cerrado (oprimiendo A ó C).

El control de lazo abierto sirve el propósito de mostrar el funcionamiento de la salida digital, así como mostrar la variación de velocidad del motor y las respectivas medidas obtenidas a través de la computadora, por ejemplo, el valor mínimo de voltaje que debe aplicarse al motor para vencer la banda muerta (aproximadamente 2 volts), y la velocidad máxima de salida al aplicar los 9 volts a la entrada (aproximadamente 3.9 volts). También se utiliza el control de lazo abierto para calibrar por vez primera los sensores en la tarjeta del convertidor D/A, junto con los amplificadores de instrumentación.

El control de lazo cerrado ejecuta hasta 100 ciclos de medición y cálculo, como se describe en el capítulo 1, en la sección correspondiente al programa "Control", realizando muestreos sucesivos y generando los valores de salida al convertidor digital analógico, de acuerdo al algoritmo de velocidad del controlador, dado por la ecuación (104):

$$u(k) - u(k-1) = K_P \{y(k-1) - y(k)\} + K_I \{r(k) - y(k)\} + K_D \{2y(k-1) - y(k-2) - y(k)\}$$

donde K_P , K_I y K_D son los coeficientes proporcional, integral y derivativo que se van a aplicar a través de la computadora

Al entrar al programa de control de lazo cerrado, se nos pide el valor deseado (referencia) de voltaje a la salida del tacogenerador (equivalente a la velocidad del motor), el cual se fija en 3.00 volts, tras de lo cual se nos piden los mencionados coeficientes K_P , K_I y K_D , que han sido calculados antes.

Como se mencionó, estos valores también pueden ser ajustados en pruebas sucesivas de ensayo y error, hasta que se logre una respuesta adecuada.

Tras dar los valores K_P , K_I y K_D , se nos pide el número de ciclos de muestreo y cálculo que deseamos efectuar (2—100), ya que el programa es propiamente una simulación del control digital que operaría en forma ininterrumpida. Aquí se dá el valor de 100.

Esto simplemente es con el objeto de registrar los datos de respuesta al escalón obtenidos con la aplicación del controlador digital, para luego graficarlos y observar sus características, así como los cambios que se producen al variar los coeficientes K_P , K_I y K_D .

Es importante recordar que en el caso del control digital directo, es indispensable ejercer el control de tipo integral ($K_I > 0$), ya que dada la forma del algoritmo de velocidad del controlador la variable de referencia $r(k)$ que se considera constante aparece solamente en ese término, para evitar un brinco derivativo cuando la referencia cambia. {Takahashi et al., 477}.

Por lo anterior, no debemos omitir el término integral bajo ninguna circunstancia. De este modo, en esta aplicación digital solo ejerceremos acciones I, PI, ID o PID, mientras que en un controlador analógico tradicional la restricción no es aplicada.

Luego la computadora muestra el valor final de referencia, el valor final de salida de la planta,

el valor final de salida del controlador digital y un promedio del tiempo de muestreo en lazo cerrado para este ejercicio en particular. Luego pide el nombre de archivo bajo el cual se guardarán los datos de esta simulación, así como las unidades requeridas.

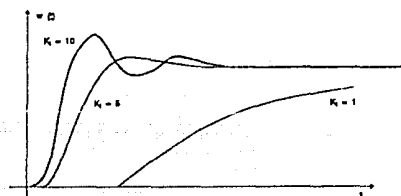


Fig. 56 Acción Integral. $K_I=1, 5, 10$.

ACCION INTEGRAL (I)

Al ejercer la acción integral pura sobre la planta (control flotante), es decir fijando $K_P = 0$ se observa que para valores bajos de ganancia ($1 < K_I < 10$) la planta responde lentamente, en forma sobreamortiguada y con un retardo de transporte mayor (ver fig. 56), mientras que para valores altos de ganancia ($K_I > 30$), se presentan oscilaciones sostenidas en la

respuesta (ver fig. 57). Con valores aún mayores de ganancia integral ($K_I=80$), la respuesta incluso se deforma. Esto se debe, como se mencionó antes, a que el control de tipo integral tiende a introducir inestabilidades, si bien aplicado correctamente anula el error de estado estacionario.

El valor ideal de ganancia está alrededor de $K_I = 15$, como se puede observar en la figura 57. Este valor es ideal en el sentido que produce una respuesta transitoria con un amortiguamiento moderado y tiempo de asentamiento corto.

En todos los casos en que se aplicó el controlador integral puro se observó un tiempo de muestreo promedio de 3.33 centisegundos (33 ms). En las figuras 56 y 57 se aprecia la forma en que varía la respuesta de la planta al control integral puro en la forma discreta, es decir con $K_P=0$ y $K_D=0$, haciendo variar K_I desde 1 hasta 80.

ACCION PROPORCIONAL INTEGRAL (PI)

Cuando se aplica la acción proporcional e integral a la vez, utilizando el valor óptimo de ganancia integral ($K_I=15$) y haciendo variar el valor de K_P desde 1 hasta 200, se observa que para valores bajos de ganancia proporcional ($1 < K_P < 10$) el cambio en la respuesta es relativamente poco (con respecto al anterior), si bien se reduce un poco el sobretiro y disminuye el período de oscilación.

Al aumentar el valor de K_P la planta en lazo cerrado empieza a mostrar oscilaciones incluso desde antes de acercarse al valor de estado estacionario, el sobretiro desaparece y aumenta el amortiguamiento.

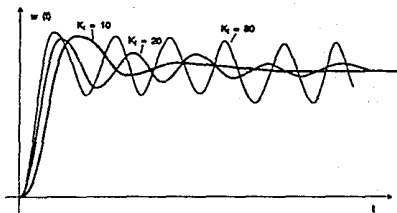


Fig. 57 Acción integral. $K_I = 10,20,30$.

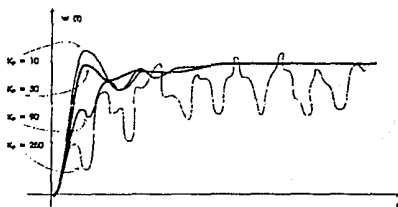


Fig. 58 Acción proporcional Integral.

Al llegar a valores grandes de K_P ($K_P > 200$), el aspecto de la gráfica nos hace ver que la planta tiene ya un comportamiento oscilatorio, incluso al grado de deformarse la forma de onda senoidal.

El valor ideal de K_P parece estar en $K_P=30$, en donde se presenta un sobretiro pequeño y un buen tiempo de asentamiento.

Con la aplicación del controlador en la forma PI, el tiempo de muestreo se ha elevado a 34.4 ms, ya que la computadora tiene ahora que efectuar más cálculos por tener dos términos en el algoritmo. Véase la fig. 58.

ACCION INTEGRAL DERIVATIVA (ID)

Ahora se aplican las acciones integral y derivativa (dejando $K_P=0$), y utilizando el valor ideal de $K_I=15$ y haciendo variar el valor de K_D desde 1 hasta 300.

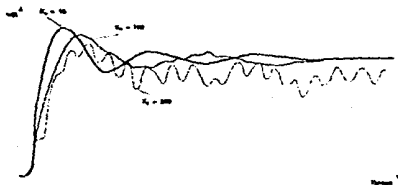


Fig. 59 Acción integral y derivativa.

Para valores bajos de K_D ($K_D < 30$), se evidencia muy poco cambio en la respuesta del sistema, siendo la frecuencia de oscilación lo que se afecta primordialmente, y disminuyendo un poco el sobretiro. (ver fig. 59).

Para valores medianos de K_D ($40 < K_D < 200$), se empieza a apreciar la amplificación de ruido (como se mencionó en la parte teórica), por lo que el valor de estado estable no se mantiene fijo. Para $K_D=300$, la respuesta del sistema incluso se deforma y pierde su aspecto senoidal, viéndose claramente como ruido amplificado.

Por lo anterior, la ganancia derivativa K_D debe tener un valor ideal de aproximadamente 60, donde aún no se amplifica ruido y sí disminuye la frecuencia de oscilación.

En los casos en que se aplicó este controlador, se observó un tiempo de muestreo en promedio de 34.7 ms (un poco mayor que en la acción PI).

En la figura 59 se aprecia la variación de la forma de respuesta al variar K_D , para poder comparar el efecto de distintos valores.

ACCION PROPORCIONAL, INTEGRAL Y DERIVATIVA (PID)

Esta sería la conjunción de las tres acciones básicas de control, la cual se logra dando los valores hallados a las tres constantes K_P , K_I y K_D . Como se mencionó con anterioridad, sería posible aún modificar estos valores para afinar la forma del control que se está aplicando a la planta.

Utilizando los valores ideales hallados antes: $K_P=30$, $K_I=15$ y $K_D=60$, se observa en la figura 60 que la parte transitoria tiene un pequeñísimo sobretiro, y el tiempo de asentamiento es entonces casi inmediato. (Ver fig. 60).

Es de esta forma como se ve la efectividad del control PID aplicado adecuadamente. También aquí se observó un aumento en el tiempo de muestreo promedio, llegando a un valor de 35.5 ms. Es útil recordar que el tiempo de muestreo en lazo abierto (cuando se capturaron los datos experimentales originales) era de 19.2 ms, de modo que el pasar al control en lazo cerrado, el tiempo de muestreo se elevó aproximadamente en un 90%, debido a que la computadora tiene no solo que muestrear, sino efectuar cálculos para obtener el valor de salida.

En sentido estricto, se aspira a que el tiempo de muestreo y cálculo fuesen despreciables con respecto a la menor constante de tiempo de la planta, de modo que fuera posible variar a voluntad el tiempo de muestreo, es decir poder seleccionarlo a nuestro arbitrio.

Desafortunadamente esto no es posible con la MicroBBC, dadas sus limitaciones en cuanto a velocidad de conversión y a velocidad de microprocesador (2 MHz), por lo cual es la planta la que debe seleccionarse de acuerdo a la computadora y no al contrario.

De cualquier forma, la MicroBBC podría utilizarse con mayor eficiencia en cualquier planta de una sola entrada y una sola salida cuya menor constante de tiempo sea cercana a 40 ms, o si se deseara muestrear un proceso con dos variables, cada una de ellas debería tener constantes de tiempo superiores a 80 ms, para que la computadora pudiese darse abasto para reconstruir la función de la planta a partir de datos experimentales muestreados (Teorema del Muestreo). Una computadora mas veloz podrá a su vez controlar plantas también mas rápidas.

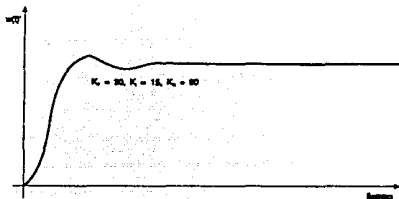


Fig. 60 Acción proporcional, integral y derivativa.

ANALISIS EN LAZO CERRADO

Conociendo los valores hallados para las constantes K_p , K_i y K_D , es posible sustituirlos en las ecuaciones de función de transferencia halladas antes, para luego hallar las raíces correspondientes en el plano complejo z , pudiendo así trazar los lugares de las raíces y hacer un análisis de estabilidad.

De hecho, esto ya se llevó a cabo en la sección de diseño de los controladores, con valores muy cercanos a los que se hallaron aquí por experimentación directa, aplicando la computadora como controlador sobre la planta.

RESUMEN

En este capítulo se presentaron los equivalentes discretos a las funciones de transferencia continuas de la planta, halladas en el capítulo anterior, así como sus valores numéricos y sus trazas de lugar de las raíces.

Se analizaron y diseñaron los controladores digitales aplicables a través de la computadora, observándose de paso la eficiencia del algoritmo PID que fué descrito aquí mismo, desglosándolo en cada una de sus acciones de control, y luego calculando las constantes que se aplican en cada caso. También se hicieron diagramas de lugar de las raíces.

Se presentaron posibles transformaciones de controladores analógicos tradicionales a sus equivalentes discretos, para los casos en que el tiempo de muestreo es pequeño comparado con la menor constante de tiempo de la planta. Por ejemplo, se citó la aproximación de Ziegler y Nichols, la regla de Tustin y otras transformaciones.

Se aplicó el controlador digital a la planta y se registró el comportamiento de la misma, tanto en forma de texto como en gráficas alusivas.

CONCLUSIONES

EVALUACION DEL SISTEMA

La interfase hombre - máquina del sistema de programas (interacción humana con la computadora que hace las veces de controlador) está basada esencialmente en el manejo de algunos menús. Si bien este acercamiento provee algunas facilidades, no es del todo amigable dadas las limitaciones de la máquina, tanto en memoria RAM como en disco.

Este trabajo de tesis provee una primera aproximación a un sistema de adquisición y análisis de datos para el diseño de sistemas de control discreto, utilizando una microcomputadora de bajo costo; es muy fácil de usar y puede simplificar la tarea de analizar y diseñar sistemas pequeños.

El sistema desarrollado es capaz de capturar y analizar los datos del comportamiento de una planta cuya constante de tiempo principal (una sola variable) sea mayor a unos 25 milisegundos (es decir una frecuencia menor de unos 40 Hz) por ejemplo sistemas electromecánicos, térmicos o hidráulicos, y es posible determinar un modelo analítico de hasta tres polos de lazo abierto para la planta, o bien simplemente con uno o dos polos si deseamos simplificar el trabajo. Estos modelos analíticos presentan poco error comparados con los datos experimentales (en algunos casos menores al 5%).

Por otra parte, el sistema es capaz de controlar una variable de una planta, con un tiempo de muestreo en lazo cerrado de unos 35.5 milisegundos (aproximadamente 30 Hz), y podría ampliarse el concepto a una planta con hasta cuatro variables, elevándose el tiempo de proceso necesariamente a unos 142 ms (alrededor de 7 Hz).

El uso de gráficas y colores en pantalla facilita la comprensión de los datos cuyo sistema representan, así como la evaluación de los cambios hechos en el desarrollo de un controlador digital, ya que permite ejercitar la planta con dicho controlador casi al momento en que fue diseñado.

Las interfaces diseñadas son sencillas y se pueden modificar con facilidad, de modo que es posible adaptar el sistema a otras plantas y microcomputadoras con relativamente pocos cambios.

No se usa el lenguaje ensamblador del microprocesador 6502A debido a que la mayor limitante del sistema no es en realidad la velocidad del microprocesador, sino la velocidad de conversión del CAD mPD7002 (aproximadamente 12 ms en lazo abierto), y que forma parte de la computadora. Sin embargo, sería posible incluir porciones compiladas en ensamblador para reducir el tiempo de cálculos del algoritmo PID, con lo que podría darse cabida a plantas más rápidas.

ERRORES Y LIMITACIONES

Como principales errores y limitaciones del sistema en su conjunto, se deben considerar los siguientes:

- a) Errores de cuantización, dado que la longitud de palabra de la computadora es 8 bits. Las máquinas más modernas usan palabras de 16 ó 32 bits, con lo cual se reduce mucho dicho error.
- b) Tamaño de la memoria RAM, ya que de los 32 kB de RAM, quedan solo unos 20 kB para datos y programa. En la actualidad memorias RAM en el orden de 1,000 ó 2,000 kilobytes son muy comunes en computadoras personales, y aun mucho más en Estaciones de Trabajo (p. ej. HP Apollo 9000), que tienen ya el poder que antes se consideraba exclusivo de minicomputadoras y mainframes.
- c) Tamaño del almacenamiento en disco. El disco de la MicroBBC solo puede almacenar por cara hasta 31 archivos, o bien 200 kB, lo que sucede primero, (total en ambas caras de 62 archivos ó 400 kB) mientras que en máquinas más avanzadas, la capacidad en disco (duro) es del orden de decenas de megabytes, y el número de archivos es prácticamente ilimitado, utilizando una estructura de árbol lógico.
- d) Velocidad de reloj. El microprocesador 6502A de la BBC corre a solo 2 MHz, mientras que las computadoras personales más recientes corren desde 10 hasta 33 MHz, y máquinas muy especializadas pueden correr aún a 50 MHz o más.
- e) Convertidor analógico digital con 4 canales y tasa cruda de muestreo de solo 100 muestras por segundo, en un solo canal, que al añadir cálculos de algoritmo en lazo cerrado caen hasta alrededor de 25. Los modernos sistemas de cómputo pueden incluir con facilidad 16 canales de conversión, con tasa cruda de muestreo (sin cálculos) de 1,000,000 de muestras por segundo.
- f) Capacidad de despliegue visual. La MicroBBC otorga un despliegue con resolución de 320 x 200 píxeles, en cuatro colores, y con velocidad de trazado relativamente baja. Los modernos sistemas personales de cómputo ofrecen ya resoluciones de 1024 x 768 píxeles, (alrededor del cuádruple) utilizando hasta 256 colores y una alta velocidad de trazado.

Cabe aclarar, sin embargo, que los modernos sistemas de cómputo tienen un costo al menos un orden de magnitud superior al de la MicroBBC, es decir entre 5,000 y 15,000 dólares (o más), sin considerar costos de periféricos, interfases, software, etc. Este tipo de inversiones solo son accesibles a grandes universidades, centros de investigación o empresas especializadas.

ERRORES DE ALGORITMO

Se encontró que el método de ajuste propuesto por Lipka, para un sistema con dos polos, no es adecuado para esta planta en particular, ya que una de las premisas que asume dicho método es que

la diferencia relativa entre tales polos sea de al menos un orden de magnitud (de 1 a 10, por ejemplo), además de ser muy sensible a la calidad de los datos, para que los cálculos resulten efectivos. Como en nuestro caso la diferencia entre los polos no llega a ser siquiera el doble, se hallaron resultados equívocos: ganancia negativa en la función de transferencia (lo que hablaría de realimentación positiva) y un cero en el semiplano derecho (lo que indicaría una posible inestabilidad).

También se halló que el método de diseño de algoritmos PID propuesto por Ziegler y Nichols no fué de gran utilidad, en este caso debido a que la frecuencia de muestreo en lazo cerrado (fija) que era posible utilizar, es muy cercana a la frecuencia principal de la planta (apenas algo más del doble, cuando debería ser alrededor de veinte veces superior), lo cual hace que las aproximaciones supuestas por ellos no sean válidas. Por este mismo motivo no es posible representar en el plano s la función de transferencia en s hallada con este método.

DESARROLLOS POTENCIALES

A pesar de sus limitaciones, este sistema sería capaz de evolucionar hacia algo más completo y complejo, ya que se podrían diseñar y añadir rutinas, por ejemplo para trazar lugares de raíces (continuos o discretos), trazas de Nyquist, trazas de Bode, análisis de estabilidad, solución de polinomios, etc. Esto podría ser de gran utilidad a nivel académico e industrial.

Por otra parte, por ser un sistema escrito en BASIC, es fácilmente comprensible y asequible para un gran número de personas, y podría ser adaptado con relativa facilidad a otras máquinas (por ejemplo, Commodore 64C, Atari 64, Apple II, etc.), de capacidades similares a la MicroBBC, en donde se le podrían hacer modificaciones o mejoras.

También es factible acoplar otras plantas, particularmente aquellas cuyo tiempo de respuesta sea relativamente lento ($T \geq 25ms$, $\tau \leq 40Hz$, p.ej. un sistema térmico), de modo que la computadora no tenga problema en muestrear y controlar una variable. Para ello solo sería necesario modificar las interfaces (amplificadores de instrumentación y actuador de potencia).

Otra opción en dichas circunstancias, (considerando siempre las limitaciones) sería ampliar el concepto de control a un sistema multiplexado de hasta cuatro variables, ya que la máquina cuenta con cuatro canales de conversión.

También se podrían escribir programas sencillos de comunicación con una computadora de jerarquía superior (p.ej. una PC) a través del puerto serial RS423 con que cuenta la microcomputadora BBC, es decir formando una configuración de control supervisor.

Finalmente, no se descarta también la posibilidad de adaptarlo a computadoras personales compatibles con IBM, sea en Pascal o en algún otro lenguaje, tomando este sistema como punto de partida para implantarlo en una configuración de cómputo más poderosa, sea esta una PC de 8 ó 16 bits, o incluso una Estación de Trabajo o minicomputadora de 32 bits.

**Esta página fué dejada
intencionalmente en blanco.**

APENDICES

En esta sección se presenta el material que no fue descrito en detalle anteriormente.

SISTEMA OPERATIVO Y HARDWARE

El Sistema Operativo de la MicroBBC está contenido en un ROM de 32 kB. Esto hace posible que no tenga que cargarse cada vez que se enciende la máquina, por otra parte, consume así un mínimo de RAM.

La MicroBBC provee al usuario con comandos para manejar las salidas a pantalla, entrada del teclado y del convertidor analógico digital, además del resto del hardware, todo a través de rutinas del sistema operativo.

Para asegurar compatibilidad con el todo el rango de computadoras de Acorn siempre es deseable, cuando sea posible, el utilizar dichas rutinas del Sistema Operativo para manejar el hardware, y no acceder directamente al mismo con código máquina. {Holmes y Dickens, 150}

En la MicroBBC el hardware está mapeado en la memoria principal, lo cual significa que cualquier registro de dispositivos de hardware aparece en el espacio de direccionamiento de la memoria principal.

Dicha memoria principal está subdividida en páginas de 256 bytes cada una, numeradas desde página 00 hasta la página hex FF, donde hex significa hexadecimal (base 16).

La página de memoria hex FE (256 bytes) que empieza en la dirección de memoria hex FE00 y llega hasta FEFF está reservada especialmente para el hardware dentro de la MicroBBC. Esta página tiene asignado el nombre de "Sheila". {Holmes y Dickens, 152}

Parte del hardware contenido en la MicroBBC es un subsistema integral de conversión analógica digital. Este sistema provee de 4 canales de entrada, cada uno de los cuales puede medir un voltaje entre cero y 1.8 volts (nominal). Este voltaje es convertido a un formato digital que el microprocesador puede manejar. {Holmes y Dickens, 365}

Este circuito convertidor analógico digital es del tipo integrador, circuito μ PD7002, {Holmes y Dickens, 369} que al aplicársele un voltaje de entrada entre 0 y 1.8 volts genera un valor digital de

salida, de 8 ó 12 bits, proporcional al voltaje aplicado. Este valor digital es presentado al microprocesador en una palabra de 16 bits. Este subsistema se controla mediante un registro de lectura - escritura que está en la dirección de Sheila hex FEC0. Su asignación cuando se escribe en él es la siguiente (por bits):

- b0-1 Selecciona el canal de conversión (0 a 3) e inicia la conversión.
- b2 bandera que se pone en 0.
- b3 0 = conversión de 8 bits (duración 4 ms)
 1 = conversión de 12 bits (duración 10 ms)
- b4-7 no se usan.

cuando se lee de él, la asignación del registro es la que sigue:

- b0-1 regresa el canal seleccionado (0 a 3)
- b2 no se usa
- b3 regresa 0 si conversión de 8 bits y 1 si de 12 bits.
- b4-5 Los dos bits más significativos de la conversión.
- b6 0 = convertidor ocupado
 1 = convertidor no ocupado
- b7 0 = conversión completa
 1 = conversión incompleta

Hay otros dos registros de datos que contienen el valor convertido, el byte alto se lee de Sheila hex FEC1 y los cuatro bits menos significativos están en los bits 4-7 de la dirección de Sheila FEC2. Estos cuatro bits serán muy inexactos en el modo de conversión de 8 bits, dependiendo de las cualidades particulares de cada chip 7002. Los valores devueltos por la instrucción ADVAL varían por lo tanto entre 0 y 65535 (hex FFFF).

En las direcciones FE60 a FE6F se encuentran el puerto paralelo (centronics) y el puerto del usuario, que son controlados por un chip 6522 (Versatile Interface Adaptor), el puerto A para el puerto paralelo y el B para el usuario. Las líneas del puerto A están reforzadas para dar una mayor capacidad de corriente, si bien esto las obliga a ser exclusivamente de salida.

FUNCIONES ESPECIALES

Una función especial de uso común es el escalón unitario. Esta función fué introducida por Oliver Heaviside a finales del siglo XIX. El escalón unitario $u(t)$ se define como:

$$u(t) = \begin{cases} 0, & \text{para } t \leq 0 \\ 1, & \text{para } t > 0 \end{cases}$$

Dicha notación es conveniente para representar el cierre de un interruptor en $t=0$. Su transformada de Laplace es $1/s$.

Otra función de uso muy común es la función rampa. Esta función se puede definir como la integral de $u(t)$, es decir que es una función creciente de t , para $t > 0$, y cero para $t \leq 0$, su gráfica es una recta de pendiente unitaria.

Ahora por derivación de la función $u(t)$, se podría definir la función impulso unitario, o delta de Dirac, que tiene valor cero para todos los valores de t excepto en $t=0$, donde vale infinito. Otra forma de definir esta función, es mediante sus propiedades de integración, es decir que:

$$\int i(t) dt = \begin{cases} 0, & \text{para } t \neq 0 \\ 1, & \text{para } t = 0 \end{cases}$$

Esta función impulso también es muy importante en el estudio de otros sistemas {Valenburg, 242}

PARTICULARIDADES DEL LENGUAJE BASIC-BBC

La MicroBBC cuenta con el lenguaje de programación nativo BASIC. El lenguaje BASIC para la MicroBBC fué desarrollado entre 1981/1982 por Acorn Computers Ltd. de Cambridge, Inglaterra. {Coll, 8}

El lenguaje BASIC está disponible en un ROM de 32 kB al momento de encender la MicroBBC, o bien se puede acceder a él simplemente dando la instrucción *BASIC. El hecho de estar en ROM hace que este BASIC sea muy rápido.

Una característica sobresaliente de este lenguaje BASIC es la de ser un lenguaje estructurado de alto nivel, ya que permite la codificación de funciones y procedimientos por nombre alfanumérico, proporcionando recursividad absoluta, siendo posible definir variables locales o bien globales, además de pasar parámetros entre procedimientos y/o del programa principal, haciendo obsoleto el uso de instrucciones del tipo GOTO y GOSUB, tan comunes en otros dialectos de BASIC.

Otra característica atractiva del lenguaje BASIC es la incorporación de un ensamblador mnemónico de instrucciones máquina para el microprocesador 6502A, accesible directamente desde BASIC. Esto provee de un ambiente muy poderoso para este ensamblador y permite que el código máquina sea fácilmente incorporado en programas escritos en BASIC. {Holmes y Dickens, 6}

A continuación se provee una lista de algunas de las instrucciones de BASIC cuyo significado podría diferir un poco con respecto al de otros dialectos de BASIC, así como aquellas instrucciones

que son exclusivas de esta máquina. Se omiten las instrucciones comunes en otros dialectos, así como aquellas que resulten fáciles de comprender en el contexto de los programas.

A pesar de todo, la mayoría de los programas que se elaboraron, son fácilmente transportables a otras máquinas que cuenten con un hardware similar al de la MicroBEC, verbigracia, un puerto para palancas de juego (joysticks) analógico y un puerto de salida paralelo de 8 bits. Tales máquinas serían la Apple II, Atari 64 o 128 y Commodore 16, 64 y 128, o algunas otras menos conocidas.

@% Controla el despliegue de datos alfanuméricos. Su formato es el de cuatro bytes hexadecimales, por ejemplo: `@%=&B1B2B3B4`. El primer byte es llamado B4. Si B4=01 indica que las cadenas deben ser formateadas de acuerdo a la variable `@%`; si B4=00 las cadenas no se formatean. B3 selecciona el formato básico: 00 formato general, 01 formato exponencial, 02 formato fijo. B2 controla el número total de caracteres impresos. B1 selecciona el ancho de los campos de impresión. Lo normal es `@%=&90A`

ADVAL Obtiene el valor de conversión analógico - digital del canal dado en el argumento (1-4). Este valor digital vá de 0 a 65520 (resolución de 8 ó 12 bits, escalado a 16) para voltajes de entrada de 0 a 1.8 volts. `ADVAL(0)` retorna el número de canal que completó su conversión más recientemente, y el estado de los interruptores de cada canal. Cada canal tarda 4 ms para 8 bits o 10 ms para 12 bits en terminar su conversión. Argumentos negativos se usan para determinar el estado de los varios registros internos de la computadora, i.e. el teclado, RS423, impresora, sonido, etc.

ASC Comando que retorna el código ASCII de un carácter alfanumérico.

AUTO Comando usado para obtener de forma automática los números de línea al digitar un programa en BASIC.

BGET Trae hacia la memoria de la computadora un byte de un archivo en disco, el cual debe haber sido ya abierto para lectura.

BPUT Escribe un byte desde la memoria de la computadora hacia un archivo en disco, el cual debe haber sido abierto previamente.

CALL Comando que hace que la computadora ejecute una porción de código máquina (ensamblado), que ya debe estar presente en la memoria. Es posible pasar parámetros a dicho segmento.

CHAIN Instrucción para cargar y correr automáticamente un programa en BASIC.

CLEAR Borra la memoria de trabajo, hace cero todas las variables o arreglos, enteras, reales y textuales, excepto `A%` a `Z%` y `@%`.

CLG Borra la pantalla de gráficos, dejándola en el color de fondo actual. El cursor gráfico se mueve a (0,0).

CLOSE# Cierra el archivo cuyo número de canal se dá en el argumento. `CLOSE#0` cierra todos los archivos.

CLS Borra el área de texto de la pantalla, dejando solo el color lógico de fondo. El cursor de texto se mueve a su origen (esquina superior izquierda).

- COLOR** Selecciona el color del fondo de la pantalla o bien en que la computadora imprimirá el texto. Esto depende también del modo de pantalla. Por ejemplo, en el modo 1 hay 4 colores (0-3): Negro, rojo, amarillo y blanco.
- COUNT** Cuenta los caracteres impresos usando PRINT, ya sea a la pantalla o al puerto RS423, hasta encontrar un cambio de línea.
- DEFFN** Inicia la definición de una función, la cual siempre retorna un valor.
Ver DEFPROC.
- DEFPROC** Informa a la computadora del inicio de una definición de procedimiento, el cual puede ser llamado por su nombre en cualquier parte del programa. Esta definición debe estar después de la instrucción END. Un procedimiento puede realizar varias funciones y no necesariamente retorna un valor. Es posible transferir parámetros a un procedimiento o función.
- DIM** Define la dimensión de un arreglo de variables, reservándoles memoria RAM. También se usa para reservar memoria para código en lenguaje ensamblador 6502.
- DIV** Operador división para números enteros. Entrega a su vez un número entero.
- DRAW** Esta instrucción funciona en los modos de pantalla 0,1,2,4 y 5. Dibuja una línea que inicia en la posición actual del cursor de gráficas y termina en las coordenadas que se dan con el comando DRAW. La pantalla se extiende de (0,0) a (1279,799).
- ENDPROC** Indica el fin de una definición de procedimiento. Ver DEFPROC, DEFFN, PROC, LOCAL.
- ENVELOPE** Se usa en conjunción con SOUND para controlar el volumen y el tono de un sonido mientras se está tocando.
- EOF#** Función que detecta el final de un archivo. Tiene un valor booleano.
- ERL** Número de línea donde se encontró un error.
- ERR** Número (clave) de error encontrado.
- EVAL** Evalúa una función algebraica a partir de una cadena alfanumérica, dando valores a las variables que formen parte de la cadena.
- EXP** Función matemática exponencial, que eleva el número e (2.718...) a la potencia especificada.
- EXT#** Determina la longitud de un archivo en disco, una vez que ha sido abierto.
- FN** Llamada a una función cuyo nombre sigue al prefijo FN. Esta función devuelve un valor numérico o alfabético. Ver DEFPROC, DEFFN, PROC, LOCAL.
- GCOL** Color de gráficas. El primer número indica el modo de graficación: absoluto, AND, OR, EOR, invertido, el segundo número indica el color lógico a usar, el cual es mayor de 128 si se refiere al color de fondo de la gráfica.
- GET** Función que espera hasta que se oprima una tecla, regresando el código ASCII de dicha tecla.
- GETS** Función que espera hasta que se oprima una tecla, regresando el carácter de dicha tecla.

HIMEM Dirección de la ubicación de memoria más alta disponible para programa y variables. Normalmente está justo antes de la memoria de pantalla. Cambia con el modo de pantalla seleccionado.

INKEY Espera un tiempo especificado o hasta que se oprima una tecla, en cuyo caso regresa el valor ASCII de dicha tecla, si no devuelve -1. El parámetro que lleva es el tiempo de espera en centésimos de segundo, o si es negativo el código de la tecla que se espera sea oprimida.

INKEY\$ Espera un tiempo especificado a que se oprima una tecla, regresando el carácter que sea oprimido, o bien una cadena vacía si el tiempo expira y no se oprimió ninguna tecla.

INSTR Busca en una cadena la existencia de una sub-cadena. Si ésta existe regresa 1 de otro modo regresa 0.

INT Convierte un número con parte decimal a uno entero, aproximando al entero inferior.

LOAD Lee de disco o red a memoria RAM un programa BASIC, sin empezar a ejecutarlo.

LOCAL Informa a la MicroBBC que las variables nombradas son de uso local, para que el valor de las mismas externas al procedimiento o función no se vea afectado. Ver DEFFN, DEFPROC.

LOMEM Proporciona la dirección más baja de memoria donde BASIC comienza a almacenar las variables del programa. Normalmente igual a TOP, que es la dirección del final de programa BASIC.

MOD Operación binaria que proporciona el residuo con signo de una división de números enteros.

MODE Cambia el modo de despliegue, hace variar la resolución, número de colores y tamaño de letra. El cambio de modo borra la pantalla. Hay 8 modos. En los programas desarrollados siempre se usó el modo 1.

Modo	resolución	colores texto	
0	640x200	2	80x25
1	320x200	4	40x25
2	160x200	16	20x25
3	n/a	2	80x22
4	320x200	2	40x25
5	160x200	4	20x25
6	n/a	2	40x22
7	n/a	16	40x20

MOVE Mueve el cursor gráfico a una posición dada, en coordenadas cartesianas, considerando que la pantalla mide 1280 por 800 puntos.

OLD Recupera el programa BASIC que estaba en memoria justo antes de oprimir

ROMPE.

ON Transferencia de control en función de un valor o evento, que apunta a una lista de líneas.

OPENIN Abre un archivo solo para lectura. Devuelve el número de canal asignado al archivo.

OPENOUT Abre un archivo solo para escritura. Devuelve el número de canal asignado al archivo.

OPENUP Abre un archivo para actualización (lectura y/o escritura). Devuelve el número de canal asignado al archivo.

OPT Opción de salida al momento de ensamblar código máquina del procesador 6502A. Seguida de un parámetro, dá las siguientes alternativas: 0 = suprime mensajes de error y listado. 1 = Suprime mensajes de error, reporta listado. 2 = Suprime listado, reporta mensajes de error. 3 = Reporta mensajes de error y listado. Esta seudo operación solo puede presentarse dentro de bloques de lenguaje ensamblador.

OSCLI Intérprete de línea del Sistema Operativo. Transfiere una cadena de comandos al Sistema Operativo para su ejecución. Esto es comandos que normalmente empiezan con un asterisco, y que no son ejecutados por BASIC sino por el Sistema Operativo.

PAGE Seudovariante que proporciona la dirección de inicio del programa en BASIC. Normalmente igual a la dirección más baja disponible en RAM, alineado por páginas. Una página son &100 bytes (hex), 256 bytes (decimal).

PI Valor de pi, 3.14159265, frecuentemente necesario.

PLOT Instrucción de usos múltiples para trazar puntos, rectas y triángulos en BASIC. El primer parámetro indica el tipo de trazo, y los dos siguientes son las coordenadas hacia donde se dirige el trazo. En particular, PLOT21 traza una línea punteada desde la posición actual del cursor gráfico hasta la coordenada especificada, PLOT69 traza un solo punto en la coordenada especificada. PLOT4 equivale a MOVE y PLOT5 equivale a DRAW.

POS Dá la posición horizontal actual del cursor de texto.

PRINT# Imprime (almacena) datos hacia un archivo en disco. Los números enteros se almacenan como un código &40, seguido del número en complemento a 2, en 4 bytes. Los números reales se almacenan como &FF seguidos de 5 bytes: 4 de la mantisa (31 bits de magnitud y 1 de signo) y 1 byte de exponente (complemento a dos en exceso de 128). Las cadenas se almacenan como &00, seguido de 1 byte contador de caracteres y luego la cadena en orden inverso.

PROC Llamada a un procedimiento (subrutina) cuyo nombre sigue a la palabra PROC.

PTR# Fija un apuntador para seleccionar el siguiente elemento que al que se accederá en un archivo.

REPEAT Hace que la MicroBBC ejecute un conjunto de instrucciones hasta que se cumpla una condición. Parte de la estructura REPEAT...UNTIL. Es posible anidar hasta 20 ciclos.

RETURN Se usa al final de una subrutina (a la cual se llegó con GOSUB) para regresar al punto del programa principal de donde fué llamada.

RND Genera un número al azar. RND(1) lo genera entre 0 y 1, RND(N) con N1 lo genera entre 1 y N. RND(-N) recalibra el generador de números.

SPC Imprime varios espacios en la pantalla. Solo se puede usar con PRINT o INPUT. EL número que le sigue indica la cantidad de espacios.

SAVE Almacena el programa BASIC de memoria a un archivo en disco, con un nombre de hasta siete caracteres.

SGN Determina si un número es positivo, negativo o cero. La función regresa: -1 para números negativos, 0 para cero y +1 para números positivos.

- SOUND** Instrucción generadora de sonidos, siendo capaz de hacerlo en tres canales de tono y uno de ruido al mismo tiempo. Los parámetros que le siguen indican canal, intensidad, tono y duración.
- SQR** Obtiene la raíz cuadrada de un número real positivo.
- TIME** Seudovariable que da o asigna el cronómetro interno, de cuatro bytes, en centésimos de segundo.
- TOP** Valor de la primera ubicación libre de memoria después del programa BASIC.
- TRACE** Hace que la MicroBBC imprima el número de cada línea conforme se está ejecutando. Util para depurar programas.
- USR** Función que permite llamar (correr) segmentos de programa escritos en código máquina, que retornan un valor al programa principal (mientras que CALL no lo hace), a través de los registros A%, C%, X% y Y%.
- VAL** Valor numérico de una cadena numerica. Si hay caracteres alfabéticos el valor retornado es cero.
- VDU** Función que accede a la unidad de despliegue visual, con varias funciones. Se listan brevemente las funciones disponibles, de acuerdo al código que les sigue. Si el código es de 32 a 255 se genera en pantalla un carácter a partir del código ASCII.
- VDU0** No hace nada.
- VDU1** Envía el siguiente carácter a la impresora.
- VDU2** Activa el puerto de la impresora.
- VDU3** Desactiva el puerto de la impresora.
- VDU4** Escribe texto en el cursor de texto.
- VDU5** Escribe texto en el cursor de gráficas.
- VDU6** Activa las unidades VDU.
- VDU7** Hace un sonido corto.
- VDU8** Retrocede el cursor un espacio.
- VDU9** Avanza el cursor un espacio.
- VDU10** Baja el cursor una línea.
- VDU11** Sube el cursor una línea.
- VDU12** Borra el área de texto.
- VDU13** Mueve el cursor al principio de la línea.
- VDU14** Habilita el modo de paginación.
- VDU15** Deshabilita el modo de paginación.
- VDU16** Borra el área de gráficas.
- VDU17** Define el color del texto.
- VDU18** Define el color de las gráficas.
- VDU19** Redefine el color lógico.
- VDU20** Restaura los colores lógicos estándar.

- VDU21** Deshabilita las unidades VDU.
- VDU22** Selecciona un nuevo modo de pantalla.
- VDU23** Reprograma la definición de un carácter en RAM.
- VDU24** Define la ventana de gráficas.
- VDU25** Grafica con los 3 códigos siguientes.
- VDU26** Restaura las ventanas estándar.
- VDU27** No hace nada. (Escape)
- VDU28** Define la ventana de texto.
- VDU29** Define el origen de las gráficas.
- VDU30** Lleva el cursor de texto arriba a la izquierda.
- VDU31** Mueve el cursor de texto a X,Y.
- VDU127** Retrocede y borra.

VPOS Dá la posición vertical actual del cursor de texto. A continuación se describen algunos de los comandos *FX del Sistema Operativo, que permiten el acceso a varias de las características especiales de la MicroBBC.

- FX4** Activa o desactiva (*FX4,1) las teclas de edición del cursor.
- FX5,1** Selecciona el puerto paralelo como puerto de salida de impresión.
- FX5,2** Selecciona el puerto serial RS423 como salida de impresión.
- FX6** Indica que no se filtren caracteres al enviarse caracteres a impresión, para que las líneas avancen correctamente.
- FX8** Fija la velocidad de transmisión en baudios a través del puerto RS423. Normalmente *FX8,4 lo fija a 1200 baud.
- FX16** Selecciona la cantidad de canales habilitados para conversión analógica-digital (1-4).
- FX17** Fuerza el inicio de conversión analógica - digital en el canal indicado.
- FX18** Reinicializa las teclas de función definibles por el usuario.
- FX25** Restaura las definiciones (suaves) de caracteres en memoria RAM.
- FX128** Lee el canal de CAD, los gatillos de disparo y la última conversión.
- FX134** Lee la posición del cursor de texto.
- FX135** Lee el carácter que se encuentra en la posición del cursor de texto.
- FX137** Enciende o apaga el motor de la grabadora de cassettes.
- FX210** Enciende o apaga el sonido.
- FX230** Activa o desactiva la tecla de Escape.

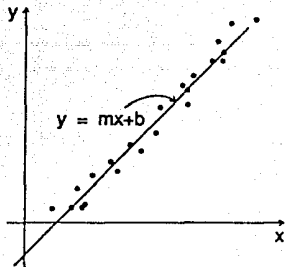


Fig. 61 Dispersión de datos y ajuste de recta.

METODO DE MINIMOS CUADRADOS

Es necesario en varias ocasiones representar el comportamiento de un conjunto de datos muestrales por una ecuación. La gráfica del conjunto de puntos se llama diagrama de dispersión, (ver fig. 61) y el proceso de encontrar la ecuación de la curva que mejor los describe recibe el nombre de ajuste de curvas o regresión. Cuando nuestro conjunto de datos (x_i, y_i) se aproxima a una recta, se habla de regresión lineal o simple. {Lipka, 161; Torres, 156; Holman, 86}.

Si llamamos x a la variable independiente y suponemos que nuestra función $y=f(x)$ se aproxima a una recta, entonces tendremos que:

$$y = a_0 + a_1x$$

es la ecuación de la recta que mejor describe nuestro conjunto de datos, y tendrá como característica el que se aproxime a todos los puntos de manera tal que se minimice el cuadrado de los errores que existan de cada punto a dicha recta (de allí el nombre de mínimos cuadrados), esto es, que si llamamos "e" a la diferencia entre un punto calculado y un punto muestral, entonces:

$$\sum e^2 = \text{mínimo}$$

Es decir que la medida de la bondad del ajuste de dicha recta será la suma de los cuadrados de los errores individuales de cada punto a la recta, lo cual implica que el error será una variable aleatoria con media de distribución cero {Miller, 319}.

Para hallar esta recta, se determinan las constantes a_0 y a_1 resolviendo el conjunto de ecuaciones lineales siguientes, llamadas ecuaciones normales para la recta de mínimos cuadrados.

$$\sum y = a_0N + a_1\sum x$$

$$\sum xy = a_0\sum x + a_1\sum x^2$$

donde N es el número de datos muestrales, Σx la suma de los valores de x , Σy la suma de valores de y , Σxy la suma de los productos xy y Σx^2 la suma de los cuadrados de x . De este modo queda:

$$a_0 = \frac{\Sigma y(\Sigma x^2) - \Sigma x(\Sigma xy)}{N(\Sigma x^2) - (\Sigma x)^2}$$

$$a_1 = \frac{N(\Sigma xy) - (\Sigma x)(\Sigma y)}{N(\Sigma x^2) - (\Sigma x)^2}$$

También se define una medida de la bondad del ajuste llamada correlación (en este caso lineal) que varía entre 0 y 1, siendo mejor el ajuste entre más próxima esté de 1, significando que los puntos están más próximos a la recta, y que está dada por:

$$r = \frac{\Sigma xy}{\sqrt{(\Sigma x^2)(\Sigma y^2)}}$$

Este método de mínimos cuadrados se incorporó al programa de gráficas (programa "Traza", procedimiento PROCmincua), para poder ajustar rectas a los varios conjuntos de puntos muestrales, siendo posible seleccionar entre qué puntos se realiza el ajuste, calculando automáticamente N , Σx , Σx^2 , Σy , Σxy y con estos datos el intercepto al origen a_0 (llamado b), la pendiente a_1 (llamada m) y la correlación r .

MÉTODO DE INTERPOLACION DE LAGRANGE

Este método de interpolación, ideado por el destacado matemático francés Louis de Lagrange en el siglo XVIII es también llamado ajuste polinomial, es fácilmente aplicable en computadoras digitales y se utiliza para obtener valores de una función experimental (muestreada), cuyas abscisas pueden estar o no igualmente espaciadas, para un valor de la variable independiente que no se haya encontrado experimentalmente. {Lipka, 263; Torres, 145,41}

Esto se logra haciendo pasar por los $n+1$ puntos muestrales un polinomio de grado n de la forma:

$$\begin{aligned} y(x) = & A_0(x-x_1)(x-x_2)(x-x_3) \dots (x-x_n) + \\ & + A_1(x-x_0)(x-x_2)(x-x_3) \dots (x-x_n) + \\ & + \dots + \\ & + A_n(x-x_0)(x-x_1)(x-x_2) \dots (x-x_{n-1}) + \end{aligned}$$

donde los coeficientes $A_0, A_1, A_2, \dots, A_n$, llamados coeficientes de Lagrange, se determinan de modo tal que el polinomio pase por los puntos muestrales. Nótese que el término que contiene A_i carece del factor $(x-x_i)$.

Para calcular dichos coeficientes se sustituye cada par de puntos muestrales en la ecuación anterior, obteniéndose un sistema de ecuaciones lineales dado por:

$$y_0 = A_0(x_0-x_1)(x_0-x_2)(x_0-x_3) \dots (x_0-x_n)$$

$$y_1 = A_1(x_1-x_0)(x_1-x_2)(x_1-x_3) \dots (x_1-x_n)$$

$$y_n = A_n(x_n - x_0)(x_n - x_1)(x_n - x_2) \dots (x_n - x_{n-1})$$

de modo que los coeficientes, al sustituir, son:

$$A_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3) \dots (x_0 - x_n)}$$

$$A_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_n)}$$

...

$$A_n = \frac{y_n}{(x_n - x_0)(x_n - x_1)(x_n - x_2) \dots (x_n - x_{n-1})}$$

lo cual substituyendo en la ecuación original da:

$$y(x) = \frac{(x-x_1)(x-x_2)(x-x_3) \dots (x-x_n)}{(x_0-x_1)(x_0-x_2)(x_0-x_3) \dots (x_0-x_n)} y_0 + \\ + \frac{(x-x_0)(x-x_2)(x-x_3) \dots (x-x_n)}{(x_1-x_0)(x_1-x_2)(x_1-x_3) \dots (x_1-x_n)} y_1 + \dots + \\ + \frac{(x-x_0)(x-x_1)(x-x_2) \dots (x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2) \dots (x_n-x_{n-1})} y_n +$$

Este polinomio, llamado forma de Lagrange, es de grado n y pasa por los puntos muestrales, pudiéndose utilizar para calcular $y(x)$ para una x distinta a las que se tienen. Si el grado del polinomio es 2, se tiene una simple interpolación lineal. No es conveniente utilizar un grado muy alto, ya que el polinomio oscila demasiado a pesar de pasar por los puntos tabulares. Un grado de 4 es bueno.

Este método se utilizó en el programa de gráficas y tablas (procedimiento PROCLagrange), para observar valores interpolados, siendo posible seleccionar en cada ocasión el grado del polinomio, escogiendo los puntos entre los cuales se ejecutaría la interpolación.

TRANSFORMADA DE LAPLACE

Un sistema es descrito matemáticamente por una ecuación íntegro-diferencial, que caracteriza las relaciones entrada-salida del sistema.

La manipulación de estas ecuaciones, que a veces contienen términos exponenciales, trascendentes o no sinusoidales, se simplifica cuando son transformadas de la representación íntegro-diferencial a otra forma más fácil de manejar aplicando la transformación de Laplace.

Al terminar de manipular las nuevas ecuaciones es posible regresar a la forma original íntegro-diferencial aplicando la transformación inversa de Laplace.

Dada la función del tiempo $f(t)$, tal que $f(t) = 0$ para $t \leq 0$, entonces $F(s)$ es su transformada de Laplace, donde $s = \sigma + j\omega$ es una variable compleja.

La transformación de Laplace se define como:

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt$$

mientras que la transformación inversa se define como:

$$f(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st} ds$$

en la práctica, se utilizan tablas de transformadas para las funciones más comunes. La siguiente es una muy breve, donde a y b son constantes:

<i>Tabla de transformadas de Laplace.</i>	
$F(s)$	$f(t)$
a_s	a
$\frac{1}{s^n}$	$\frac{t^{n-1}}{n-1}$
$\frac{1}{(s-a)}$	e^{at}
$\frac{1}{(s-a)^n}$	$\frac{t^{n-1}e^{at}}{(n-1)!}$
$\frac{1}{(s-a)(s-b)}$	$\frac{1}{a-b}(e^{at} - e^{bt})$
$\frac{s}{(s-a)(s-b)}$	$\frac{1}{a-b}(ae^{at} - be^{bt})$

Tabla 6. Transformadas de Laplace

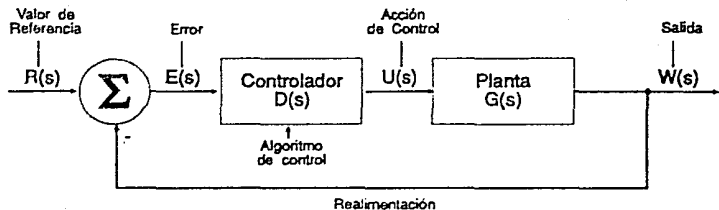


Fig. 62 Sistema de control.

FUNCION DE TRANSFERENCIA

La relación de la señal de salida a la señal de entrada de un sistema se llama función de transferencia. Estas señales son normalmente expresadas como transformadas de Laplace (variable compleja s), con la suposición de que todas las condiciones iniciales son cero. También se puede expresar como una razón de polinomios en z , para el caso discreto, o como funciones matriciales complejas, para sistemas multivariables. Este concepto de función de transferencia es la base de todos los análisis de sistemas de control, y se considera en la gran mayoría de los textos sobre esta materia. Ver fig. 62.

METODO DEL LUGAR DE LAS RAICES

Hay varios métodos analíticos para determinar la estabilidad de un sistema, todos dependen del hecho de que si $\sigma > 0$ el sistema será inestable. Describiremos brevemente un método gráfico, llamado lugar de las raíces, para determinar la estabilidad de un servosistema.

Considérese un sistema de lazo cerrado, cuya función de transferencia es:

$$F(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

La expresión entera es llamada función de transferencia de lazo cerrado; $G(s)H(s)$ es llamada la función de transferencia de lazo abierto.

El objetivo es determinar las raíces (polos) del denominador de $F(s)$ de la función de transferencia de lazo cerrado, puesto que ellos caracterizan la respuesta del sistema. Por lo tanto, la ecuación a resolver es:

$$1 + G(s)H(s) = 0$$

la cual es llamada la ecuación característica. Esta ecuación puede ser resuelta numéricamente para determinar los polos. Sin embargo, deseamos encontrar la dependencia de los polos de algún parámetro tal como la ganancia del sistema. Esto es necesario ya que el parámetro puede no ser conocido exactamente, o porque podemos desear cambiar el parámetro y observar como afecta esto a la localización de los polos.

Un modo de lograr esto es utilizando el método del lugar de las raíces, el cual básicamente involucra los polos de lazo cerrado en relación a los polos y ceros de lazo abierto (raíces del numerador), y un parámetro k , el cual es normalmente la ganancia que nos interesa observar. El método es descrito con las siguientes seis reglas.

Regla 1. Para construir el lugar de las raíces, se obtiene la ecuación característica y la reorganizamos en el formato:

$$1+k \frac{(s-z_1)(s-z_2) \dots (s-z_n)}{(s-p_1)(s-p_2) \dots (s-p_m)} = 0$$

donde k es el parámetro de interés (usualmente la ganancia) el cual se supone positivo. El segundo término en el miembro izquierdo de la ecuación es la función de transferencia de lazo abierto. Entonces localizamos los polos de lazo abierto y los ceros en el plano complejo s .

Regla 2. Encontrar los puntos inicial y final del lugar de las raíces, puesto que en todos los sistemas reales el número de polos de lazo abierto es más grande o igual al número de ceros ($n \geq m$). El lugar de las raíces comienza para $k=0$ en los polos de lazo abierto y termina en un cero de lazo abierto o en el infinito. Hay n ramas que terminarán en un cero y habrá $(n-m)$ ramas que terminarán en el infinito a lo largo de las asíntotas.

Regla 3. Determinar el lugar de las raíces en el eje real. Un punto sobre el eje real yace sobre un lugar de las raíces si el número total de polos y ceros de lazo abierto sobre el eje real a la derecha del punto es impar.

Regla 4. Determinar las asíntotas de los lugares de las raíces para las $(n-m)$ ramas que terminan en el infinito a lo largo de dichas asíntotas. El ángulo de las asíntotas es igual a:

$$\theta = \frac{2\pi(2N+1)}{(n-m)}, \text{ para } N = [0, 1, 2, \dots, (n-m-1)]$$

todas las asíntotas intersectan al eje real en el punto $\bar{\sigma}$, dado por:

$$\bar{\sigma} = \frac{(p_1+p_2+\dots+p_m)-(z_1+z_2+\dots+z_n)}{(n-m)}$$

Regla 5. Encontrar los puntos de escape y de entrada. Si el lugar de las raíces se encuentra entre dos polos adyacentes sobre el eje real, hay por lo menos un punto de escape. Dichos puntos son aquellos donde el trazo del lugar de las raíces se separa del eje real. Similarmente, si el lugar de las raíces se localiza entre dos ceros en el eje real, hay como mínimo un punto de entrada, donde el trazo del lugar de las raíces vuelve a unirse al eje real. Si el lugar de las raíces está entre un cero y un polo sobre el eje real, no existen puntos de escape ni de entrada.

Si la ecuación característica está dada por:

$$1+k \frac{B(s)}{A(s)}$$

entonces la localización de los puntos de entrada está dada por las soluciones de:

$$A'(s)B(s) - A(s)B'(s) = 0$$

donde la prima indica derivación con respecto a s . El valor de k en aquellos puntos es igual al producto de las distancias a todos los polos divididos por las distancias a todos los ceros.

Regla 6. Encontrar los puntos donde los lugares de las raíces intersectan al eje imaginario.

Estos puntos pueden ser hallados sustituyendo $s = j\omega$ en la ecuación característica. Igualando las partes real e imaginaria a cero permite la solución para k y ω . El valor de k en ese punto es importante ya que determina el valor del parámetro que causará que el sistema se convierta en inestable.

GLOSARIO

Amortiguamiento. Término que describe el decaimiento de la amplitud de una señal oscilatoria.

Amortiguamiento crítico. La respuesta de un sistema críticamente amortiguado a una perturbación escalón es el retorno a su estado de equilibrio sin llegar más allá de dicho estado (sin oscilaciones), en el mínimo tiempo posible.

Ancho de banda. El rango de frecuencias en el cual la magnitud de la ganancia del sistema expresada en dB está dentro de la banda de los 3 dB.

Banda muerta. Rango de señales de entrada para el cual no hay respuesta del sistema.

Cero. Término de la gráfica el lugar de los polos para las frecuencias a las cuales la ganancia del sistema se hace cero.

Constante del motor. El cociente del par del motor a la potencia de entrada al mismo.

Constante de regulación de velocidad. Pendiente de la característica velocidad/par del motor.

Constante de tiempo eléctrica. La constante de tiempo eléctrica de un servomotor de CD es el cociente de la inductancia de la armadura a la resistencia de la misma.

Constante de tiempo mecánica. El tiempo para que un servomotor de CD sin carga alcance el 63.2% de su velocidad final, después de aplicarle un escalón de voltaje.

Corriente a plena carga. Corriente de armadura del motor operado a su pleno par de carga y velocidad.

Decibel (dB). Medida logarítmica de la ganancia de un sistema. $A \text{ (db)} = 20 \log A$.

Diagrama de bloque. Representación esquemática simplificada de un sistema, con cada componente representado por un bloque colocado en el orden de flujo de la señal a través del sistema.

Ecuación característica. La ecuación característica de un servosistema es $1+GH = 0$, donde G es la función de transferencia de la trayectoria de avance de la señal y H es la función de transferencia de la trayectoria de realimentación.

Frecuencia de corte. Las frecuencias a las cuales las asíntotas de la curva de ganancia en la gráfica de Bode se intersectan.

Frecuencia de paso. Frecuencia a la cual la magnitud del producto de la ganancia de la trayectoria de avance y la ganancia de la trayectoria de realimentación es la unidad.

Función de Transferencia. Razón de las transformadas de Laplace de la señal de salida a la señal de entrada de un sistema.

Ganancia. Cociente de la señal de salida de un sistema a la señal de entrada al mismo.

Ganancia de lazo. El producto de las ganancias de las trayectorias de avance y de realimentación.

Gráfica de Bode. Gráfica de la magnitud de la ganancia del sistema, en dB, y la fase de la ganancia del mismo, en grados, contra la frecuencia de la señal de entrada senoidal en escala logarítmica.

Linealidad. Para un sistema de control de velocidad es la máxima desviación entre la velocidad real y la velocidad fijada, expresada como un porcentaje de la velocidad fijada.

Margen de fase. El ángulo de fase de una ganancia de lazo, menos 180 grados, en la frecuencia de paso.

Margen de ganancia. Magnitud de la ganancia del sistema, a la frecuencia en la cual el ángulo de fase del producto de las ganancias de la trayectoria de avance y realimentación es de -180 grados.

Orden del sistema. Grado de la ecuación característica del sistema.

Polo. Término de una gráfica del lugar de las raíces, para las frecuencias a la cual la ganancia del sistema tiende a infinito. Raíces de la ecuación característica.

Razón de amortiguamiento. Medida del amortiguamiento de un sistema, expresado como el cociente entre el amortiguamiento actual y el amortiguamiento crítico.

Reacción de armadura. La producción de un campo magnético cambiado por 90 grados eléctricos con respecto a la dirección del campo magnético del estator, producido por la corriente de armadura.

Red de adelanto. Una red la cual disminuye el retraso entre la señal de entrada del sistema y la señal de salida.

Red de retardo. Una red la cual incrementa el retraso entre la señal de entrada del sistema y la señal de salida.

Regulación de velocidad. Para un sistema de control de velocidad, es la variación de la velocidad real contra la velocidad fijada, expresada en porcentaje.

Rizo del par. Variación cíclica del par generado. Tiene una frecuencia dada por el producto de la velocidad angular del motor y el número de segmentos del conmutador.

Tiempo de caída. El tiempo para que la amplitud de la respuesta de un sistema decaiga al 37% de un valor de estado estable después de la supresión de la señal que forzaba dicho estado.

Tipo del sistema. Número de polos de la ganancia de lazo localizados en el origen del plano complejo.

Velocidad a plena carga. Velocidad de un motor operado a un voltaje dado, y un par a plena carga.

BIBLIOGRAFÍA

Durante la elaboración de esta Tesis, se consultó libremente la literatura sobre el tema, de modo que es natural que la presentación elegida para ciertos puntos se haya visto influida, en ocasiones muy directamente, por tales consultas, hecho que no necesariamente se reconoce debidamente en el cuerpo del texto. Para subsanar parcialmente esta falla, se presenta la lista completa de textos consultados.

La referencia a éstos dentro del texto se hizo citando el autor principal, seguido por el número de las páginas consultadas.

CIRCUITOS Y SISTEMAS

Gabell, Robert y Roberts, Richard.
Señales y Sistemas Lineales.
LIMUSA 1975.

Gerez, V. y Murray M.
Teoría de Sistemas y Circuitos

Representaciones y Servicios de Ingeniería
S.A. 1974.

Reid J., Gary.
Linear Systems Fundamental
McGraw-Hill International Book Company
1983.

COMPUTACION

Coll, John y Acorn Computers Ltd
BBC Microcomputer User Guide.
British Broadcasting Corporation 1983.

Holmes, Mark and Dickens, Adrian.
The New Advanced User Guide for the BBC
Micro
Adder Publishing Co. 1987.

Murray, Ian.
Designing Software Projects for the BBC
Micro
Century Communications 1985.
IAM A.Caso 001.64 MUR

Peatman, John.
Microcomputer Based Design.
McGraw-Hill International Book Company
1977.

Siewiorek, D.P., Bell C.G. and Newell, A.
Computer Structures.

ELECTRICIDAD

Gourishankar, V.
Conversión de Energía Electromecánica.
Representaciones y Servicios de Ingeniería
S.A. 1975.

Kip, Arthur.
Fundamentos de Electricidad y Magnetismo.
McGraw-Hill International Book Company
1972.

Meisel, Jerome.
Principios de Conversión de Energía
Electromecánica.

ELECTRONICA

Fairchild Semiconductor 1976.
Linear Integrated Circuits

Millman, J. and Halkias, C.
Integrated Electronics.
McGraw-Hill International Book Company
1972.

Millman, J.
Microelectronics.
Digital and Analog Circuits and Systems.
McGraw-Hill International Book Company
1979.

National Semiconductor Corp. 1978.
Linear Applications Handbook.

Principles and Examples.
McGraw-Hill International Book Company
1982.

Stone, Harold.
Microcomputer Interfacing.
Addison Wesley 1982.

McGraw-Hill International Book Company
1969.

Thaler, George y Wilcox, Milton.
Máquinas Eléctricas.
LIMUSA 1974.

Valkenburg, M.E.
Análisis de Redes.
LIMUSA 1983.

Wildi, T. y De Vito, M.J.
Experimentos con Equipo Eléctrico.
LIMUSA 1980.

National Semiconductor Corp. 1981.
Logic Databook.

Schilling and Belove.
Electronic Circuits, discrete and integrated.
McGraw-Hill International Book Company
1968.

Sladdin, M.
Elementary Electronics.
Hodder and Stoughton 1983.
IAM 537.5 SLA

INGENIERIA DE CONTROL

Alvarez Gallegos, Jaime y Alvarez Gallegos, Joaquín.
Control Digital. Informe Técnico.
CINVESTAV, IPN 1986.

Auslander, Takahashi y Rabins.
Introducción a Sistemas y Control.
McGraw-Hill, 1974.
UAM Azcapotzalco, QA402/A9.58/34308

Cadzw, J.
Discrete Time Systems
Prentice Hall 1973.
UAM Azc. QA402/C3.2

Chesmond, C.J.
Control System Technology.
Edward Arnold (Publishers)
Great Britain 1982.

Davis, A. Sidney
Retroalimentación y Sistemas de Control.
Fondo Educativo Interamericano 1977.

D'azzo J.J. and Houps C.H.
Feedback Control System analysis
and synthesis.
McGraw-Hill International Book Company
1966.

D'azzo J.J. and Houps C.H.
Linear Control System analysis and design.
McGraw-Hill International Book Company
1981.

Distefano, J., Stubberud, A. y Williams.
Retroalimentación y Sistemas de Control.
Schaum, McGraw-Hill International Book
Company 1972.

Dephande, B. Pradeep y Ash, H. Raymond
Elements of Computer Process
with advanced Control Applications.
Instrument Society of America, USA 1982.

Dorf C. Richard.
Modern Control Systems.
Addison-Wesley 1980.

Electrocraft Corporation
D.C. Motors, speed controls and servosystems.
Pergamon Press 1977.
UAM Azcapotzalco TK2611/D3/67245
ISBN 0 08 021715 X

Franklin G.F., Powell J.D.
Digital Control of Dynamic Systems.
Addison Wesley Publishing Company 1980.
UAM Azc. TJ216/F7.3/109365

**Franklin G.F., Powell J.D.,
Emami-Naeini A.**
Feedback Control of Dynamic Systems.
Addison Wesley Publishing Company 1986.
UAM Azc. TJ216/F7.4/109364

Houps, C. and Lamont, B.
Digital Control Systems.
McGraw-Hill International Book Company
1985.

Kuo, B.C.
Digital Control Systems.
Holt, Rinehart and Winston 1980.

Mayr, Otto
The origins of Feedback Control.
MIT Press, 1970.
UAM Azc.

Neff, Herbert.
Continuous and Discrete Linear Systems

Harper and Row 1984.
UAM Azc. QA402 N4.2

Ogata, Katsuhiko.
Ingeniería de Control Moderna.
Prentice Hall 1980.

Rao, Guthikonda.
Complex Digital Control Systems.
Van Nostrand Reinhold Co. 1979.

Richards, R.J.
An introduction to Dynamics and control.

MATEMATICAS

Derrick, William y Grossman, Stanley
Ecuaciones diferenciales con aplicaciones.
Fondo Educativo Interamericano 1984.

Edwards and Penney.
Ecuaciones Diferenciales Elementales.
Prentice Hall Hispanoamericana 1986.

Miller, I. y Freund E.J.
Probabilidad y Estadística para Ingenieros.
Prentice Hall 1987.

METODOS NUMERICOS

Lipka, Joseph.
Computaciones Gráficas y Mecánicas.
CECSA 1975.

METROLOGIA

Holman, J.P.
Métodos Experimentales para Ingenieros.

OTROS

Rose, J.
La revolución Cibernética.
Fondo de Cultura Económica 1978.

Longman, 1979.
UAM Azc. TJ213/R5.3

Schwarzenbach, J. and Gill, K.F.
System Modelling and Control.
John Wiley & Sons 1978.

Takahashi Y., Rabins M., Auslander D.
Control and Dynamic Systems
Addison Wesley Publishing Company 1970.
IIE TJ213/T28 UAM TJ213

Spiegel, Murray
Teoría y problemas de Estadística
chaum, McGraw-Hill International Book
Company 1970.

Wylie, C. Ray
Advanced Engineering Mathematics.
McGraw-Hill Int'l Book Company 1975.

Torres, J. y Czitrom V.
Métodos para la solución de problemas.
con computadora digital.
Representaciones y Servicios
de Ingeniería 1980.

McGraw-Hill International Book Company
1981.

Xerox Corporation.
Ventura Publisher Professional Extension
Reference Guide. 1989.