



CUVyTT-BUAP



Control de Procesos por Computadora





OBJETIVO GENERAL DEL CURSO

El material del curso proporcionará la Teoría de Control que está relacionada con el análisis y diseño de sistemas controlados por computadora, haciendo énfasis sobre los conceptos e ideas básicas.

OBJETIVOS ESPECIFICOS

Conocer las herramientas de representación y análisis de sistemas muestreados, así como la técnica para verificar la estabilidad. Además de establecer las características indispensables para tener esquemas de control en los que se involucre el uso de sistemas computacionales, como interfaces y automatización.



CONTENIDO Y ESQUEMAS DEL CURSO

Unidad I INTRODUCCIÓN AL CONTROL POR COMPUTADORA

- 1.1 Introducción.
- 1.2. Tecnología de las computadoras.
- 1.3. Teoría de control por computadora.
- 1.4. Como se desarrolla la teoría.

Unidad II MUESTREO DE SEÑALES CONTINUAS.

- 2.1. Introducción.
- 2.2. Descripción de los mecanismos de muestreo.
- 2.3. El teorema de muestreo.
- 2.4. Proceso de muestreo como un modulador de amplitud de pulso.
- 2.5. Retenedores.
- 2.6. Reconstrucción.
- 2.6. Enmascaramiento o frecuencia. "Aliasing".
- 2.7. Aspectos prácticos de la elección del periodo de muestreo.



Unidad III MODELOS MATEMATICOS ORIENTADOS A LA COMPUTADORA: SISTEMAS DISCRETOS.

3.1. Introducción.

3.2. Muestreo de un sistema continuo en el espacio de estados.

3.3. Transformación de los modelos del espacio de estados.

3.4. Modelos de entrada-salida.

3.5. La transformada Z.

3.5.1 Definiciones

3.5.2 Propiedades

3.5.3 Inversión de la transformada z

3.5.4 Teorema de convolución

3.5.5 Funciones de transferencia de sistemas muestreados

3.6. Polos y ceros.

3.7. Selección de la frecuencia de muestreo.



Unidad IV MODELOS ORIENTADOS AL PROCESO.

4.1. Introducción.

4.2. Sistema controlado por computadora.

4.3. El modelo de modulación.

4.4. Respuesta en frecuencia.

4.5. Convertidores D/A Y A/D

4.6 Muestreadores retenedores



Unidad V ANALISIS DE SISTEMAS DISCRETOS.

5.1. Introducción.

5.2. Estabilidad.

5.3 Criterios Algebraicos

5.3.1 Criterio de Schun-Cohn

5.3.2 Criterio de Jury

5.3.3 Criterio de Routh Hurwitz modificado

5.3.4 Criterio de Jury alterno

5.4 Controlabilidad, alcanzabilidad u observabilidad

5.5 Técnicas de diseño y realización de sistemas de control.

5.6 Ecuación de estado discreta

5.7 Solución de la ecuación de estado

5.8 Métodos para obtener la matriz de transición de estados discreta



Unidad VI CONVERSIÓN DEL DISEÑO ANALÓGICO

6.1 Introducción.

6.2 Métodos alternativos para aproximaciones.

6.3 Controladores Digitales PID.

6.4 Métodos de ajuste.

6.4 Implementaciones prácticas.

Unidad VII Tarjeta de Adquisición y Control.

7.1 El bus de la PC.

7.2 Especificaciones de la tarjeta.

7.3 Selección de la Dirección Base.

7.4 Asignamiento de las terminales de los conectores.

7.5 Estructura de los registros y formato.

7.6 Descripción general del diagrama de bloques.

7.7 Descripción de cada bloque.



Unidad VII Programación de la tarjeta de Adquisición y Control.

8.1 Funciones de programación en C para DOS.

8.2 Funciones de programación en C para Windows.

8.3 Programación bajo Objetos.

8.4 Sistema GENIE.

8.5 Editor de Estrategias.

8.6 Bloques del Editor de Estrategias.

8.7 Creando Estrategias prácticas.

8.8 Creando Presentaciones Gráficas.

8.9 Runtime Prácticas.



CUVyTT-BUAP



CRITERIOS DE EVALUACION

EXAMENES	60	%
SIMULACION EN COMPUTADORA Y CIRCUITOS	10	%
PROYECTO	20	%
TAREAS Y PARTICIPACION	10	%

TEXTOS Y REFERENCIAS REQUERIDAS

Computer- Controlled Systems. Theory and Design.
Karl Astrom y Bjorn Wittenmark Third Edition.

Modern Control Design with Matlab and Simulink
Ashish Tewari Ed. Wiley.

Señales y Sistemas.
Haykin y Van Veen Ed. Limusa Wiley.



CUVyTT-BUAP



Sistemas de Control para Ingeniería
Norman S. Nise Tercera Edición Ed. CECSA.

Digital control systems
B.C.Kuo
Holt, Rinehart and Winston

Genie Reference Manual. Data Acquisition and Control Software.
ADVANTECH.

Digital Control Engineering.
M. Copal.

Digital Control Systems.
Rof Isermann.

Digital Control Systems. Theory, Hardware, Software.
Constantine H. Houppis. Gary B. Lamont.

Control por Computadora. Descripción interna y diseño optimo.
Aníbal Ollero Baturone. Ed Alfaomega Marcombo

Digital Control of Dynamic Systems
Gene F. Franklin J. David Powel.

Tecnicas Discretas en Ingeniería de Sistemas. Tomo I y II
Francisco J. Jauffred M. Alberto Moreno Bonett

Control digital de sistemas de control lineal
Y. Sevely
Dunod Université

Industrial Control Electronics. Applications and Design.
J. Michael Jacob.
Prentice Hall.



CUVyTT-BUAP



Sistemas de Control en Tiempo Discreto
Katsuhiko Ogata. Ed. Prentice Hall.

Modern digital control systems
R.G. Jacouot
Marcel Decker, Inc

Sistemas y Circuitos. Digitales y Analógicos.
A. Papoulis. M. Bertran.

Dinámica de Sistemas y Control. Eronini-Umez-Eronini. Editorial Thomson-
Learnig.

Feedback Control Systems Fourth Edition Charles L. Phillips, Royce D. Harbo
Editorial Prentice Hall.

Sistemas de Control Moderno. Análisis y Diseño
Grantham y Vincent. Ed. LIMUSA

Automatic Control Systems.
Benjamin C. Kuo and Farid Golnaraghi. Ed. Wiley Eighth Edition

Introducción a los Sistemas de Control con Computador.
Juan Maria Pérez Oria y Santiago Arnaltes Gómez.



Ciencia centraba toda su atención en describir e interpretar los fenómenos naturales

La ciencia constituye un cuerpo de conocimientos crecientes, formado por la sucesión de las ideas, reflexiones y más todavía de las experiencias y las acciones de una gran corriente de pensadores y trabajadores

Fuerza motriz del desarrollo de la sociedad.



La ciencia no se dedica exclusivamente al pensamiento, sino al pensamiento llevado continuamente a la práctica y renovado en la práctica.

Por esto, no se puede conocer a la ciencia separada de la tecnología.

La fusión de la ciencia y la técnica es un proceso cada vez más acelerado que ha llevado a la producción automatizada.



Los procesos productivos tienden a utilizar el control automático para mejorar la productividad, la calidad, abaratar costos y evitar riesgos.

Un especialista en control tenía que dominar la tecnología de las computadoras analógicas para los cálculos.

En los sistemas de control se utilizaban las tecnologías mecánica, neumática y electrónica de tipo analógico.



Las computadoras digitales son una herramienta para el análisis y el diseño de sistemas de control.

El especialista en control dispone de herramientas como Matlab, SIMULINK, Control Classic, Simnon, etc.



El propósito de estos apuntes es presentar los fundamentos de la teoría de control que está relacionada con el análisis y diseño de sistemas, haciendo énfasis sobre los conceptos e ideas básicas.

La computadora digital con un software para los cálculos y simulaciones de forma que se le puedan dejar los detalles más tediosos.



Sistemas dinámicos pueden estudiarse en cuatro etapas:

- 1.- Descripción física del sistema (modelado físico).
- 2.- Descripción matemática del sistema (construcción del modelo).
- 3.- Análisis de la descripción matemática (solución del modelo).
- 4.- Síntesis de una manifestación preferida del sistema (diseño del sistema).



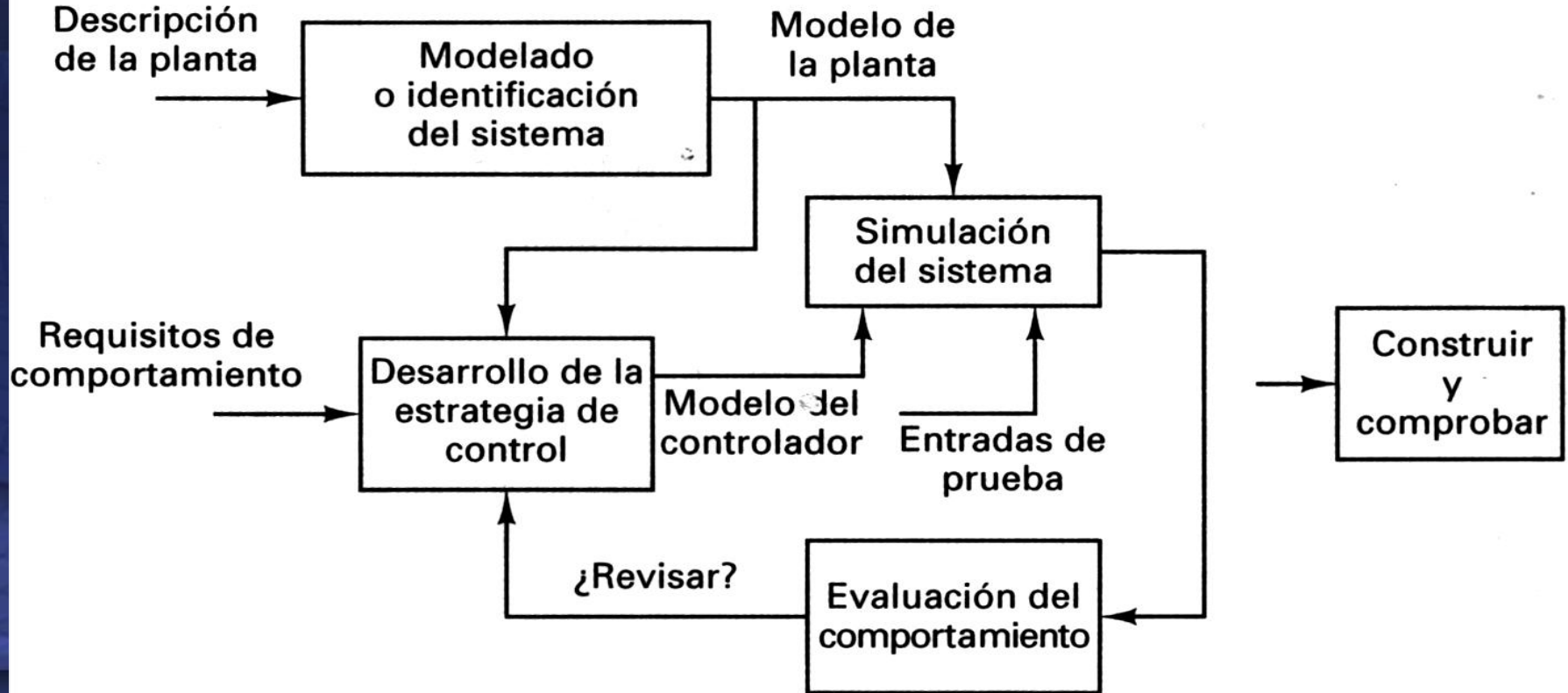
Fundamentos

La teoría de control convencional o clásica es aplicable solamente a sistemas lineales e invariantes en el tiempo, de una sola entrada y de una sola salida.

La teoría de control moderna es aplicable con múltiples entradas y múltiples salidas, lineales o no lineales, variante o invariantes con el tiempo.

La teoría de control moderno es esencialmente un método en el dominio en el tiempo y se basa en el concepto de estado.

Un procedimiento de Diseño





Antecedentes Históricos

El utilizar computadoras digitales como componentes de sistemas de control y medición surgió alrededor de 1950. Las primeras aplicaciones ensayadas fueron el control de misiles y de aviones.

Las computadoras eran demasiado grandes, consumían demasiada potencia y no eran suficientemente fiables. Por esta razón se desarrollaron computadoras de propósito específico -analizadores diferenciales digitales (DDA, de Digital Differential Analyzers)- para las primeras aplicaciones aeroespaciales.

Los desarrollos más importantes en el control y monitoreo por computadora se produjeron en las industrias de proceso.

Períodos

- *a) Período de los pioneros (1955).*
- *b) Período de control digital directo (1962).*
- *c) Período de la minicomputadora (1967).*
- *d) Período de la microcomputadora (1972).*
- *e) Período de Algoritmos (1988).*
- *f) Control Distribuido.*



CUV y TT-BUAP



Después de las conversaciones preliminares se decidió probar en una unidad de polimerización de la refinería de Port Arthur, Texas. Un grupo de ingenieros de Texaco y TRW hicieron un estudio de factibilidad completo, que requirió alrededor de 30 hombres/año. Se diseñó un sistema de control por computadora para la unidad de polimerización basado en una computadora RW-300. El sistema de control se puso en marcha el 12 de marzo de 1959. El sistema controlaba 26 flujos, 72 temperaturas, 3 presiones y 3 composiciones. Las funciones esenciales eran minimizar la presión del reactor, determinar la distribución óptima entre la alimentación de los cinco reactores, controlar la entrada de agua caliente a partir de la actividad catalítica y determinar la recirculación óptima.



EVOLUCION DE LOS SISTEMAS DE PROCESAMIENTO DE LA INFORMACION

- PRIMERA GENERACION (BULBOS)
- SEGUNDA GENERACION (TRANSISTORES)
- TERCERA GENERACION (CIRCUITOS INTEGRADOS)
- CUARTA GENERACION (MICROPROCESADORES)
- QUINTA GENERACION (PROGRAMACION)



1.1.1 El periodo de los pioneros

El trabajo hecho por TRW y Texaco sentó un precedente importante y despertó un interés en las industrias de procesos, entre los fabricantes de computadoras y en las organizaciones de investigación. Los fabricantes de computadoras iniciaron muchos estudios de factibilidad ya que estaban interesados en aprender la nueva tecnología y en conocer como debería ser un control por computadora adecuado. Los sistemas con computadora que se utilizaban en este periodo eran lentos, caros y poco confiables. Las tareas más importantes de la computadora eran encontrar las condiciones de operación óptimas, realizar la organización y la planificación de la producción y elaborar



CUVyTT-BUAP



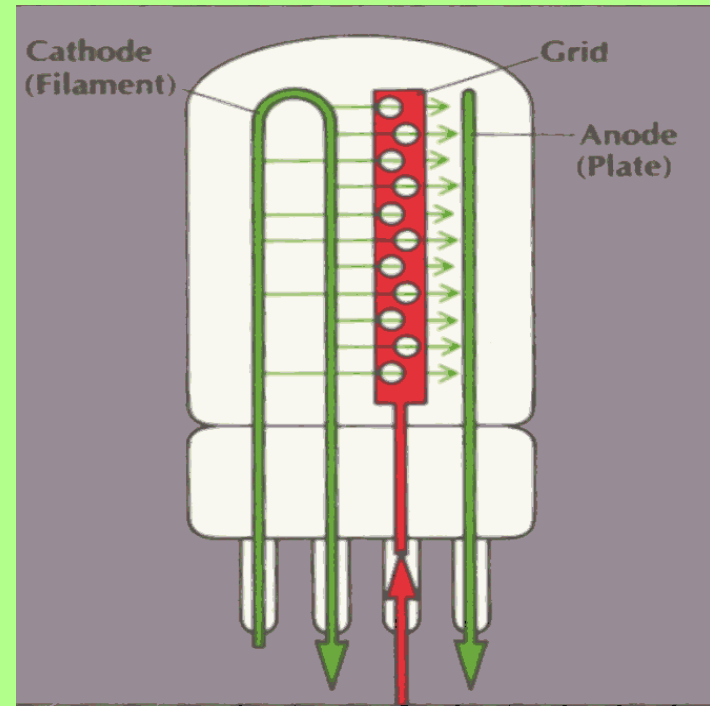
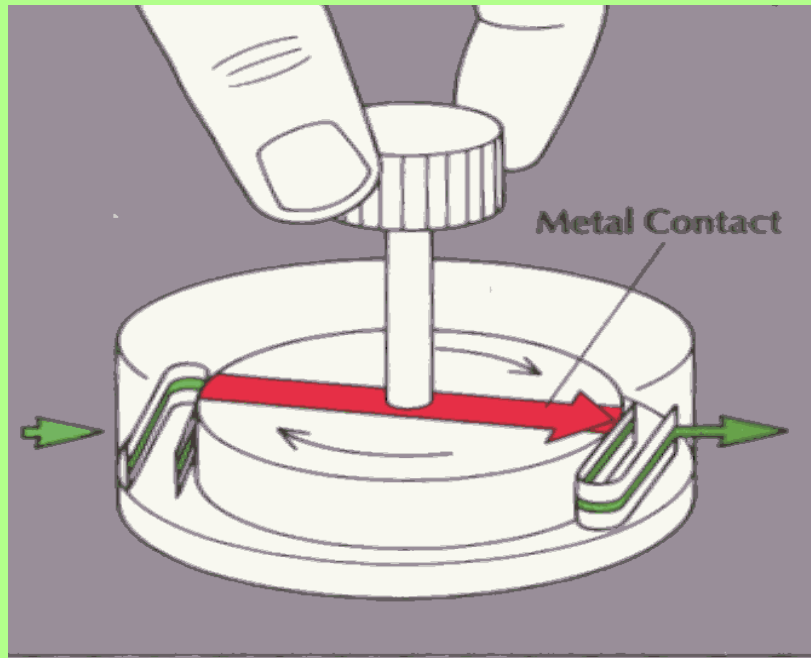
A menudo los avances se veían entorpecidos por una falta de conocimientos de proceso. También fue quedando claro que no era suficiente con ver los problemas desde el punto de vista de la optimización estática, sino que era necesario elaborar modelos dinámicos. Una buena parte de los estudios de factibilidad estaba dedicada a encontrar los modelos matemáticos adecuados. Durante los estudios de factibilidad se fue aprendiendo que el control de procesos impone a las computadoras unas exigencias especiales. La necesidad de una rápida respuesta a las peticiones del proceso llevó a desarrollar la *característica de las interrupciones*, que consiste en un dispositivo físico especial que permite a un acontecimiento exterior interrumpir a la computadora en el trabajo que está haciendo para que pueda responder a tareas del proceso que son más urgentes. Muchos de los sensores que se necesitaban no se encontraban disponibles. En marzo de 1961 se habían instalado ya 37 sistemas. Un año más tarde el número había crecido hasta 159. Las aplicaciones comprendían el control de acerías, industrias químicas y plantas de generación de energía eléctrica. Los estudios de factibilidad continuaron a lo largo de los años 60 y los 70.



1a. GENERACION (40's)

- **FABRICACION CON BULBOS**
- **DISPOSITIVOS VOLUMINOSOS**
- **ALTO CONSUMO DE POTENCIA**
- **PROGRAMACION ALAMBRICA (DURA)**
- **MUY BAJA CONFIABILIDAD**
- **MUY BAJA CAPACIDAD DE PROCESAMIENTO**

1a. GENERACION (40's)

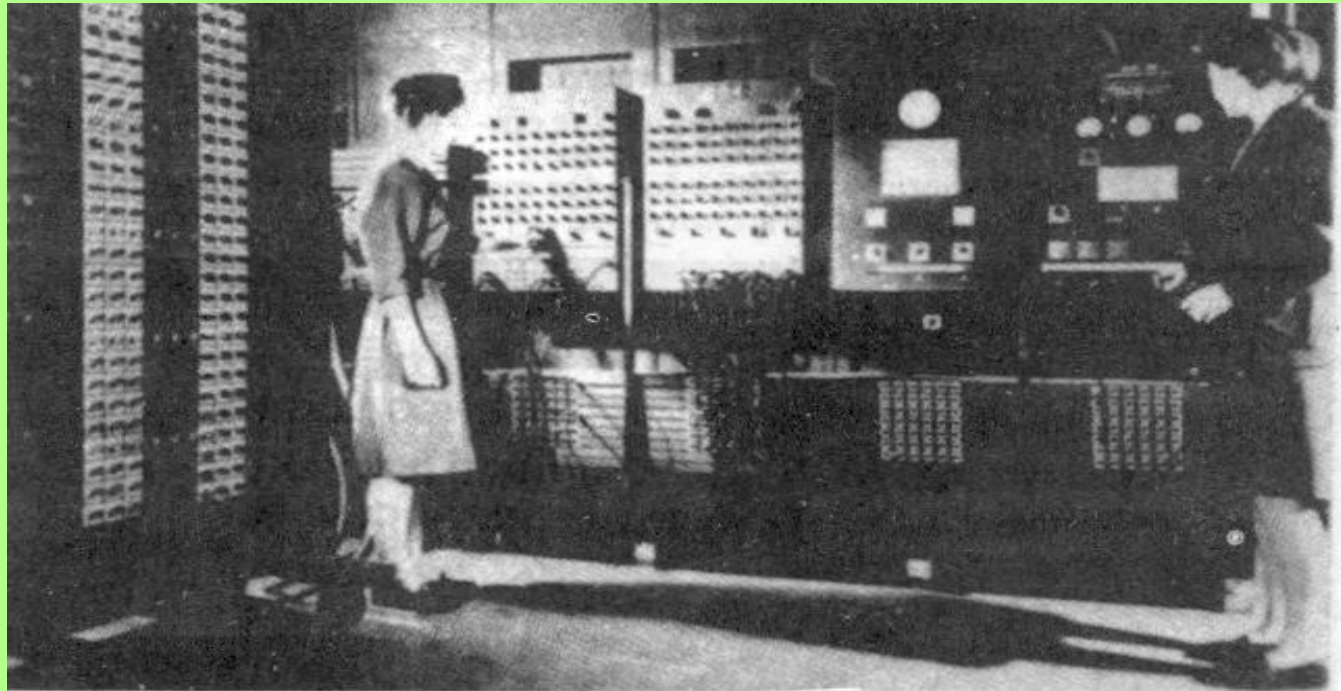


Comparación entre Transistor y Bulbo



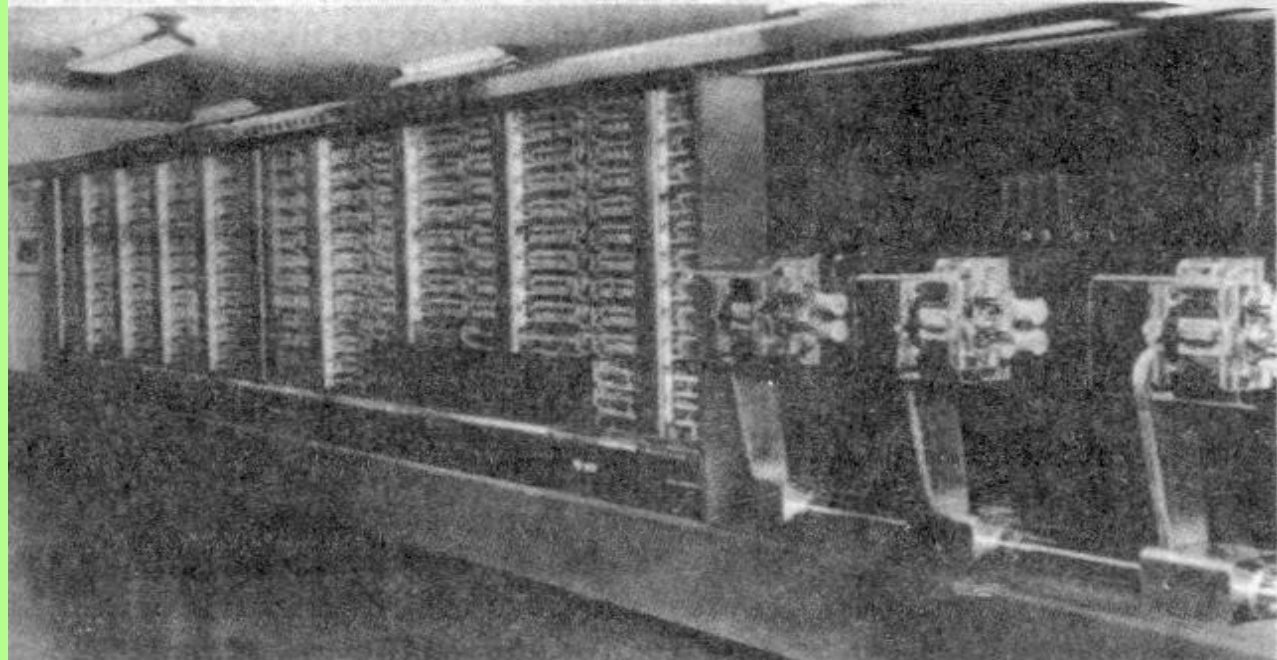


1a. GENERACION (40's)





1a. GENERACION (40's)





1.1.2 El periodo del control digital directo

Las primitivas instalaciones de computadoras de control operaban en modo supervisor, bien como guía de operador, bien como control de consigna. En ambos casos hacía falta el equipo de control analógico ordinario. Imperial Chemical Industries (ICI) en Inglaterra se apartó drásticamente de este enfoque en 1962. Se sustituyó la totalidad de la instrumentación analógica de control de procesos por una computadora, una Argus de Ferranti. La computadora medía 224 variables y controlaba 129 válvulas directamente. Era el comienzo de una nueva era en el control de procesos: simplemente la tecnología analógica se sustituía directamente por la tecnología digital; la función del sistema era la misma. Se acuñó el nombre *Control Digital Directo* (DDC, del inglés Direct Digital Control) para destacar que la computadora controlaba el proceso directamente. En 1962 una computadora típica de control de procesos podía sumar dos números en $100\mu\text{s}$ y multiplicarlos en 1 ms. El MTBF (Mean Time Between Failures, en español Tiempo Medio Entre Fallos) era de alrededor de 1, 000 horas.



El costo era el argumento más fuerte para cambiar la tecnología. El costo de la tecnología analógica aumentaba linealmente con el número de lazos de control; el costo inicial de una computadora digital era grande, pero el costo de añadir un lazo adicional era pequeño. El sistema digital era entonces más barato para los sistemas grandes. Además la comunicación con el operario podía realizarse desde un panel de comunicaciones. El panel empleado en el sistema de ICI era muy simple: una pantalla digital y unos pocos pulsadores. La flexibilidad era otra de las ventajas de los sistemas DDC: un sistema analógico se modificaba recableándolos y uno digital solo necesitaba ser programado nuevamente. Otras de las ventajas de los Sistemas DDC eran: facilidad de



2a. GENERACION (50's)

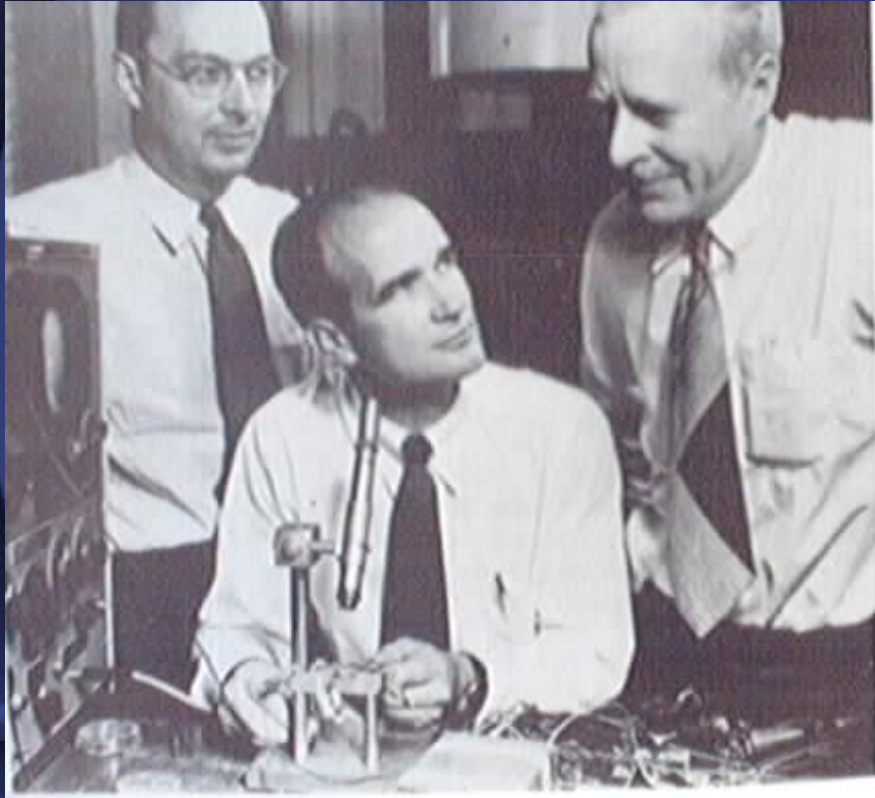
- FABRICACION CON TRANSISTORES
- ALAMBRADO COMPLEJO
- MENOR CONSUMO DE POTENCIA
- PROGRAMACION POR INSTRUCCIONES (BLANDA)
- MAYOR CONFIABILIDAD
- MAYOR CAPACIDAD DE PROCESAMIENTO
- MENOR COSTO

CUVyTT-BUAP

Primer Transistor



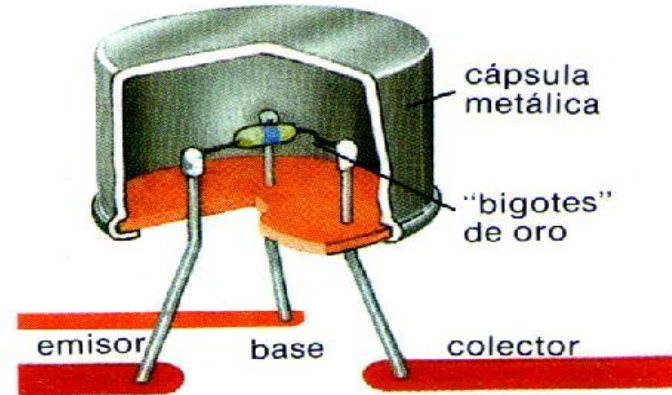
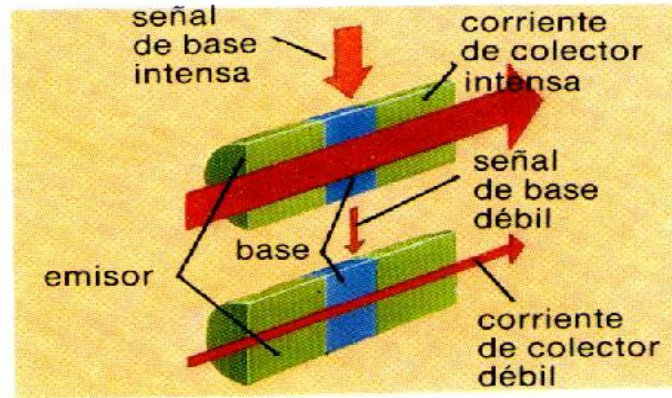
Inventores del transistor



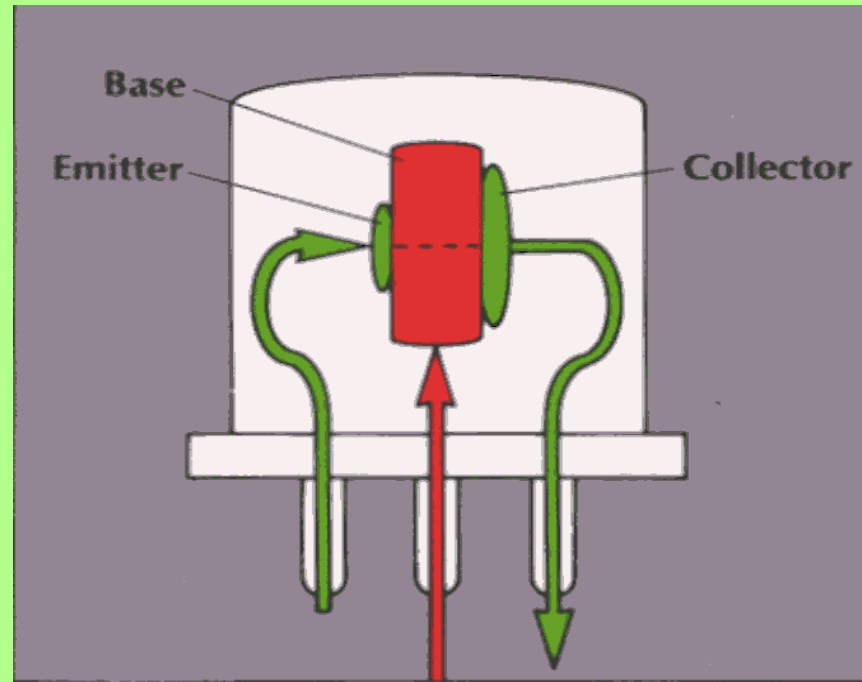
Bardeen,
Shockley
y Brattain

Lab. Bell
1947

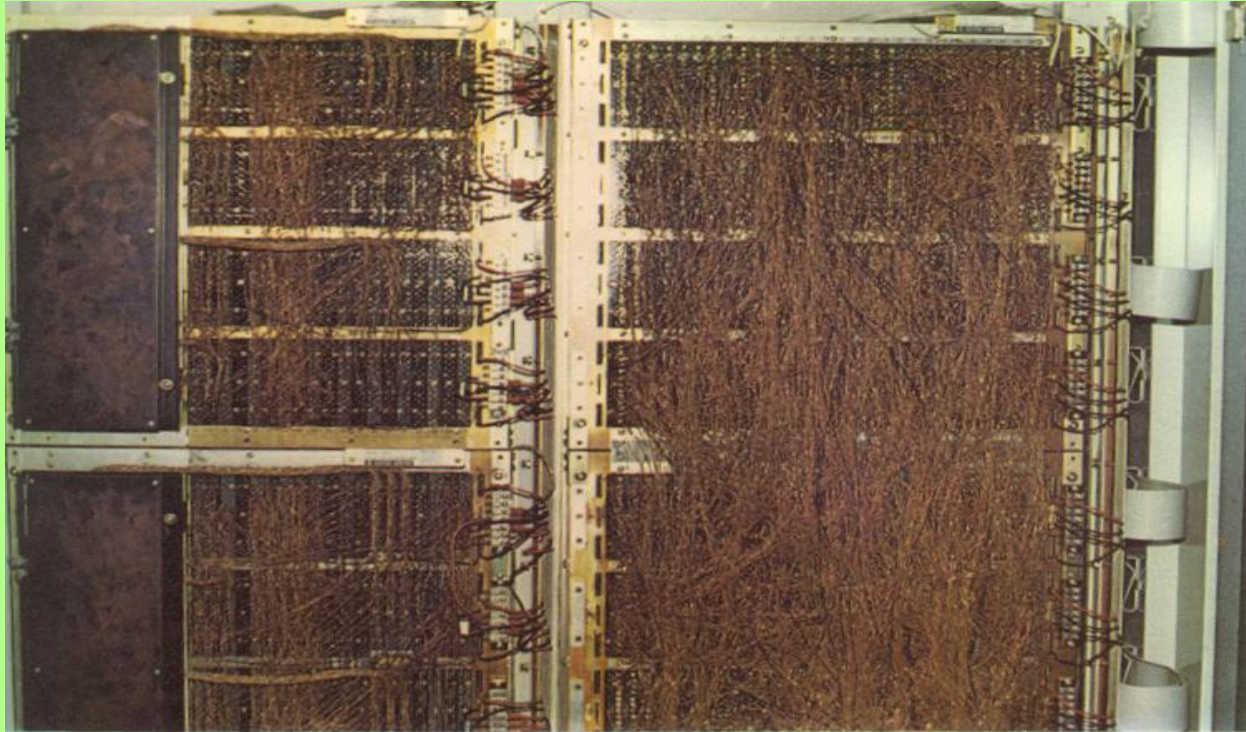
Transistor Actual



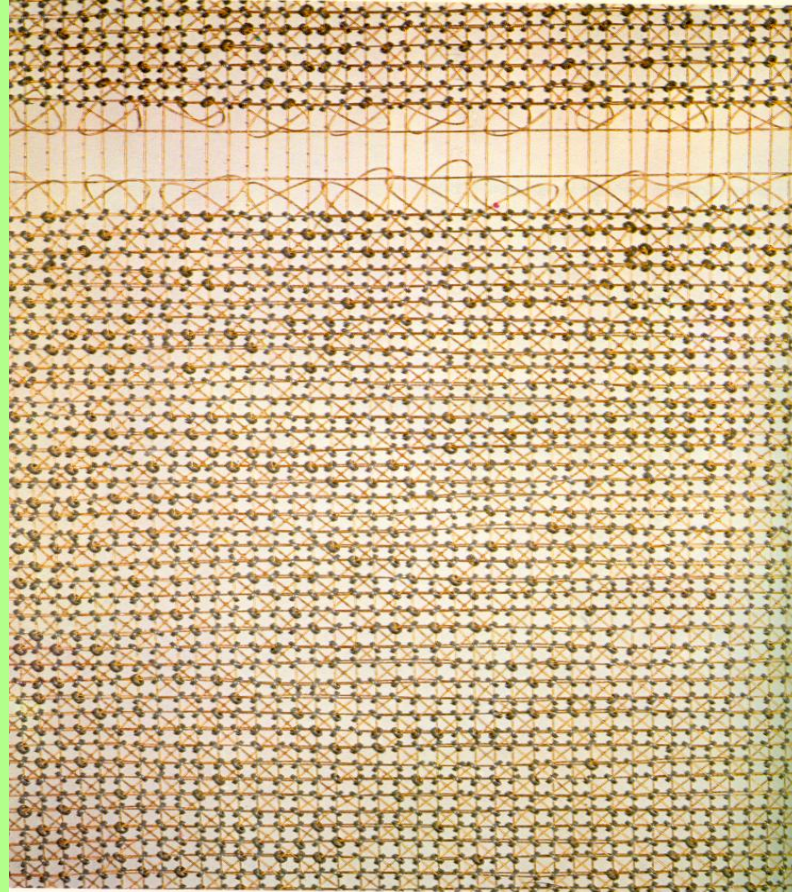
2a. GENERACION (50's)



2a. GENERACION (50's)



2a. GENERACION (50's)

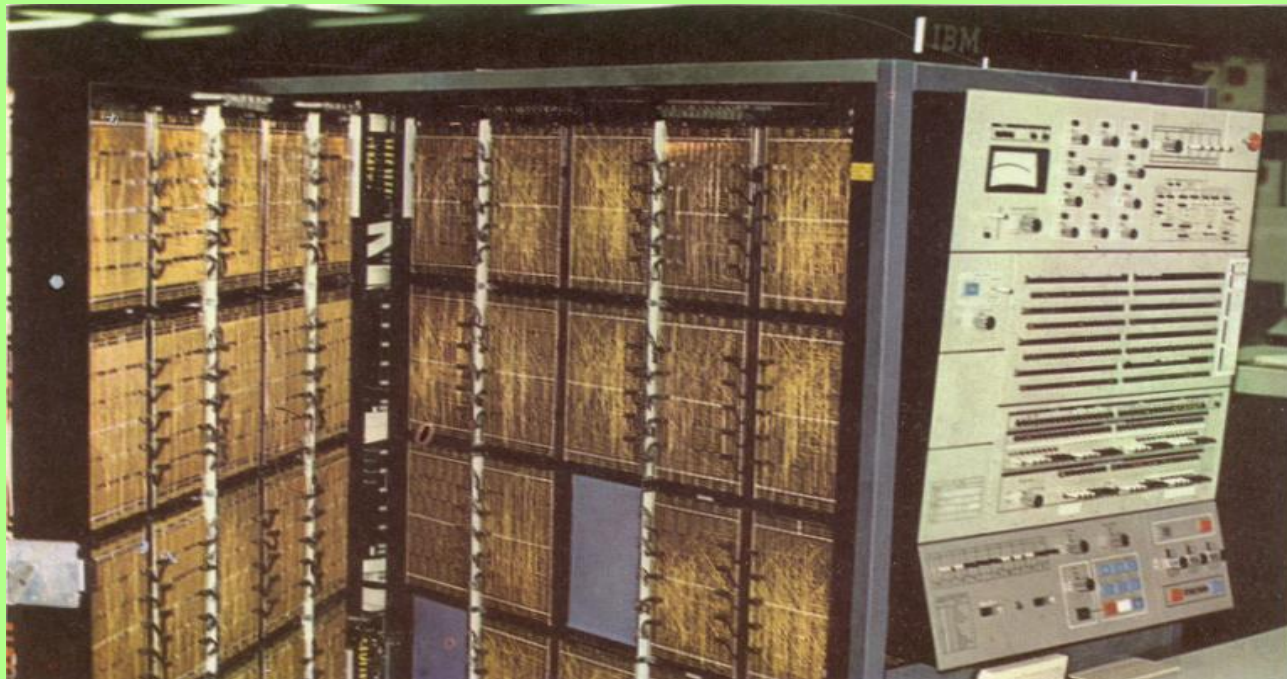


2a. GENERACION (50's)



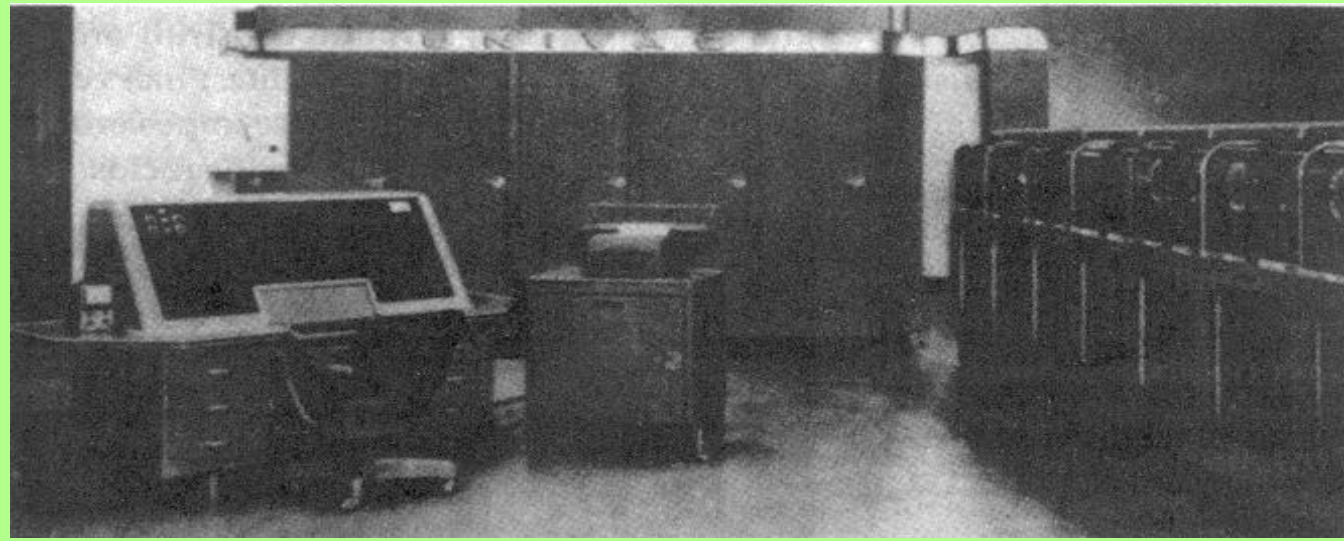


2a. GENERACION (50's)





2a. GENERACION (50's)





1.1.3 El periodo de las minicomputadoras

Durante los años 60 hubo un desarrollo sustancial en la tecnología de las computadoras digitales. Los requisitos de las computadoras de control de procesos se cumplían perfectamente gracias al progreso en la tecnología de los circuitos integrados. Las computadoras digitales resultaban más pequeñas, más rápidas, más fiables y más baratas. Se elaboró el término *minicomputadora* para las nuevas computadoras que surgieron.

El desarrollo de las minicomputadoras combinado con el creciente conocimiento alcanzado sobre el control de procesos con computadora durante los periodos inicial y de DDC produjo un rápido incremento en las aplicaciones de control por computadora. Una computadora de proceso típico de ese periodo tenía una longitud de palabra de 16 bits. La memoria principal era de 8K a 124K palabras. Se solía emplear un disco como memoria secundaria. Una computadora típica de ese periodo fue el CDC 1700, con un tiempo de suma de $2 \mu\text{s}$ y un tiempo de multiplicación de $7 \mu\text{s}$. El MTBF de la unidad central era de alrededor de 20, 000 horas.



3a. GENERACION (60's)

- **FABRICACION CON CIRCUITOS INTEGRADOS**
- **ALAMBRADO REDUCIDO**
- **MUY BAJO CONSUMO DE POTENCIA**
- **GRAN DESARROLLO DE LA PROGRAMACION**
- **FACILIDAD DE MANEJO**
- **AUMENTO EN SU CAPACIDAD Y VELOCIDAD**
- **DESARROLLO DE REDES**
- **AUN PARA APLICACIONES ESPECIFICAS**

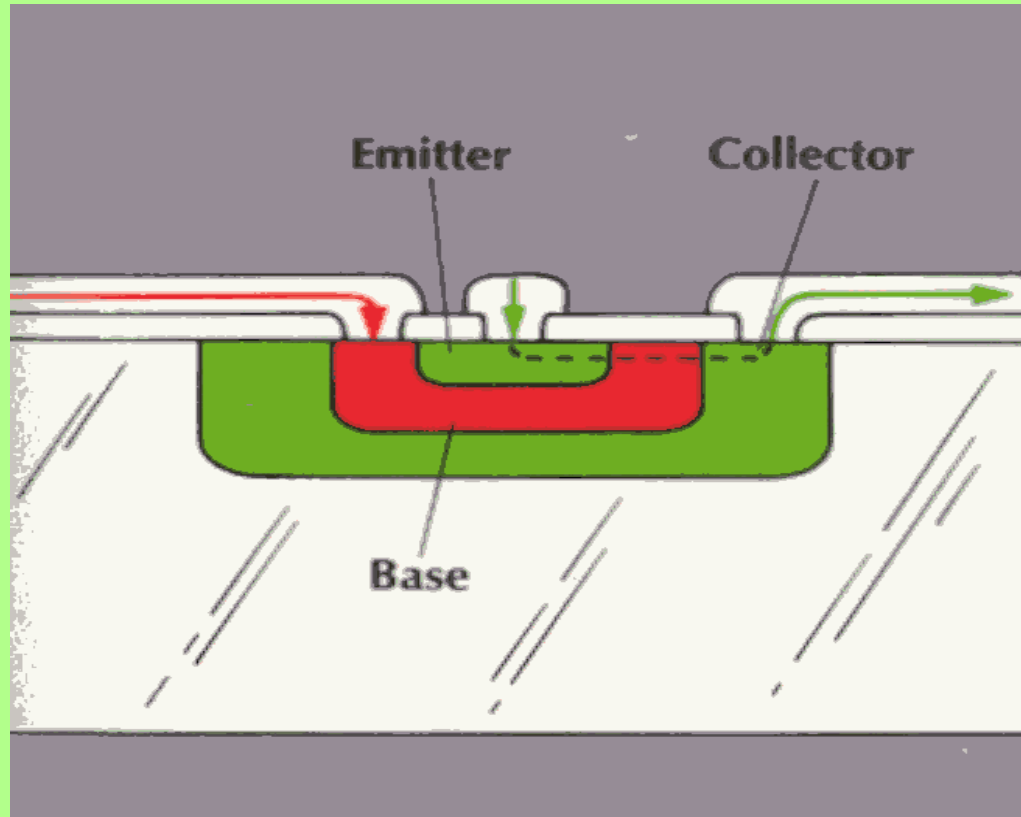
Primer Circuito integrado



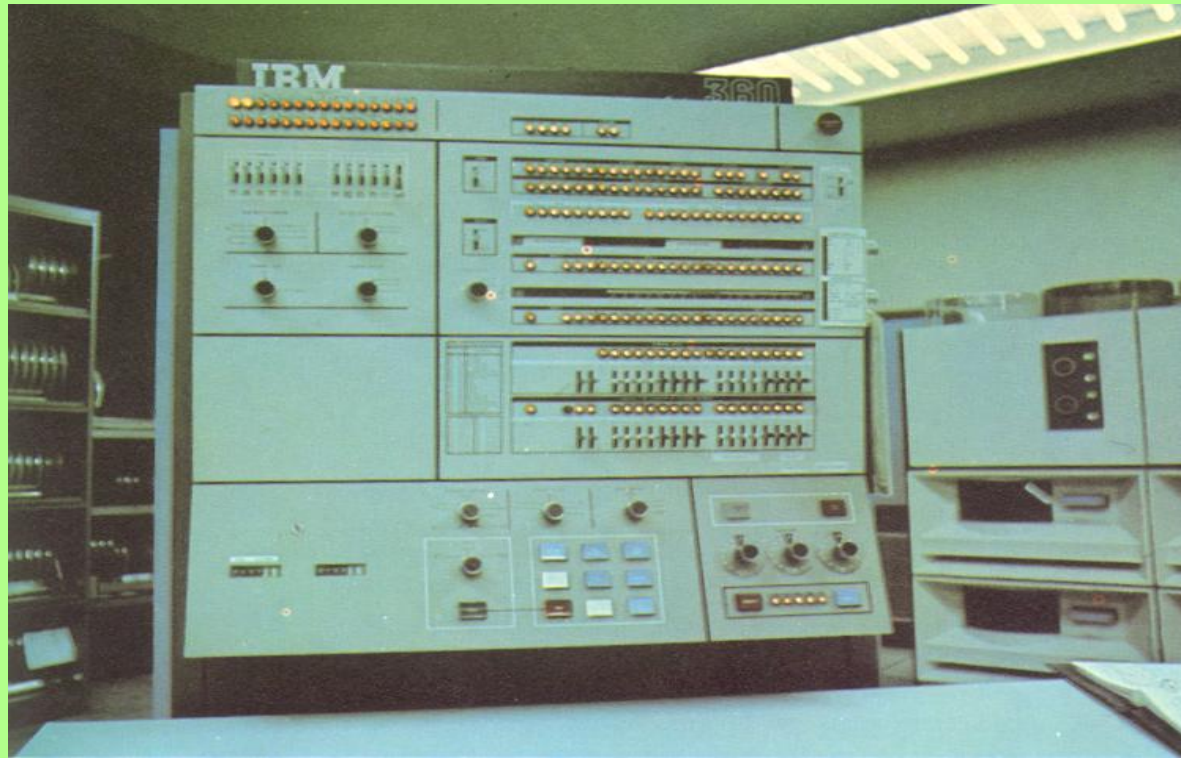
Comparación de Ctos. Integrados con otros objetos



3a. GENERACION (60's)



3a. GENERACION (60's)





1.1.4 El periodo de las microcomputadoras

La minicomputadora era todavía un sistema grande. Incluso aunque las prestaciones siguieron aumentando y los precios disminuyendo, el precio de una minicomputadora en 1975 era todavía de unos 10, 000 dólares. Esto significa que un pequeño sistema rara vez costaba menos de 100, 000 dólares. El control por computadora estaba todavía fuera del alcance de un gran número de problemas de control. Pero con la llegada de la microcomputadora en 1972, el precio de una computadora en una tarjeta con las prestaciones de una minicomputadora de 1975 descendió a 500 dólares en 1980. Otra consecuencia fue que la potencia de cálculo digital en 1980, comprada en cantidad, llegó a valer solamente 50 dólares. Esto significa en la práctica que el control por computadora podía considerarse en ese momento como una alternativa que había que tener en cuenta, sin importar lo reducido del tamaño de la aplicación.



4a. GENERACION (70's)

- FABRICACION CON MICROPROCESADORES
- SENCILLEZ EN SU CONSTRUCCION
- AUN MAS BAJO CONSUMO DE POTENCIA
- DESARROLLO DE LA COMPATIBILIDAD EN LA PROGRAMACION
- GRAN DESARROLLO DE PERIFERICOS
- DESARROLLO DE PAQUETES DE PROGRAMACION
- ALTA CONFIABILIDAD
- MUY BAJO COSTO
- DE APLICACIÓN GENERALIZADA.

4a. GENERACION (70's)

INTEL
4004



INTEL
8080



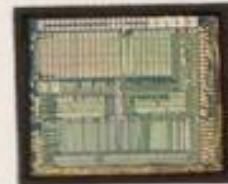
MOS TECHNOLOGY
6502



MOTOROLA
68000

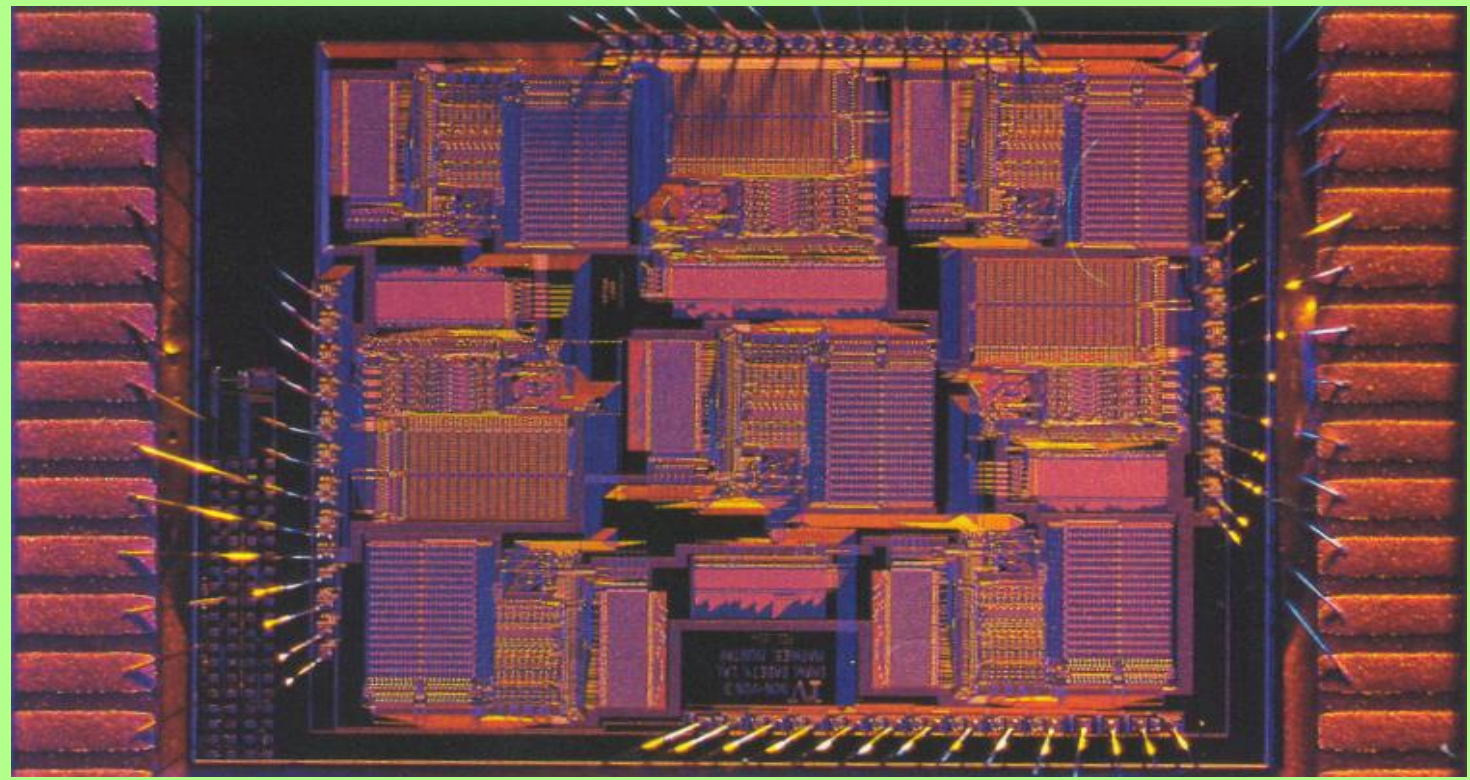


HEWLETT-PACKARD
SUPERCHIP





4a. GENERACION (70's)

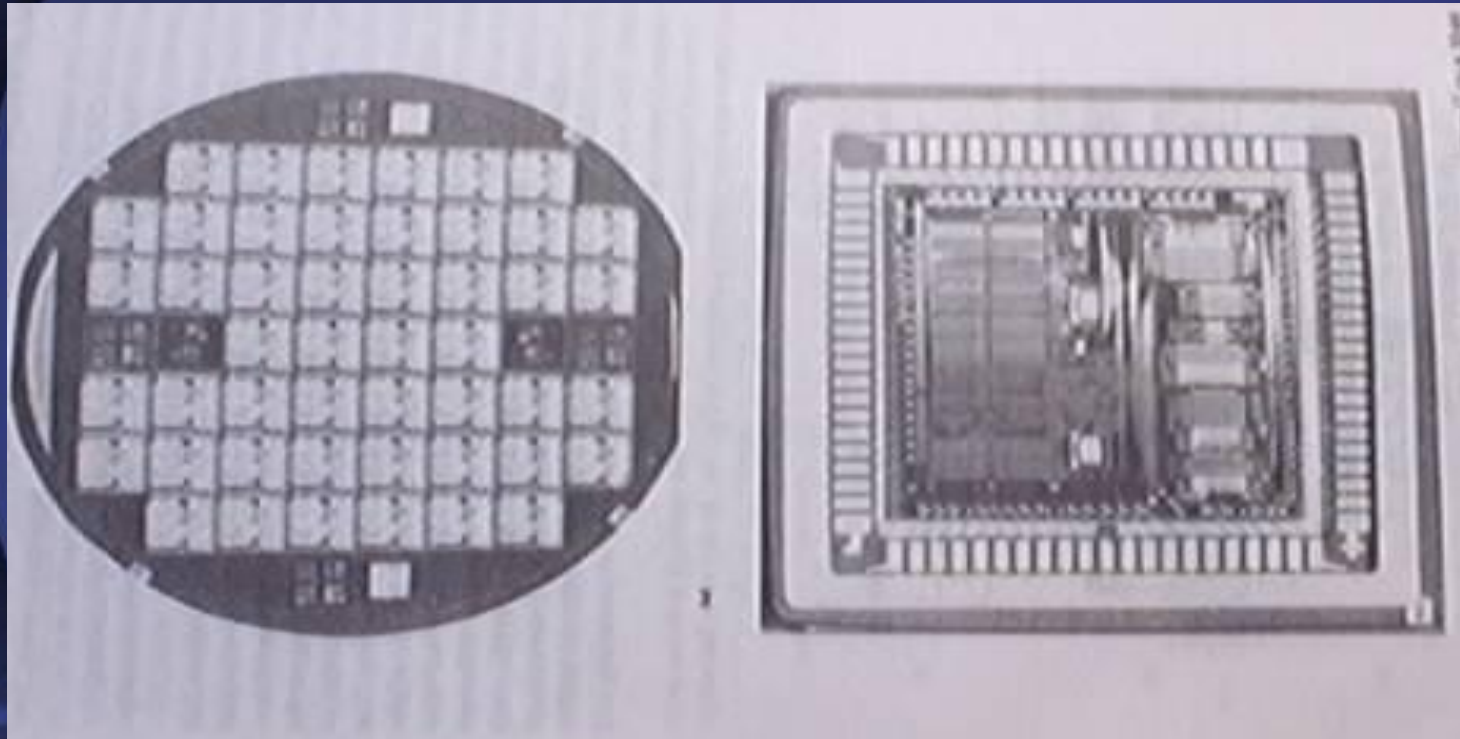




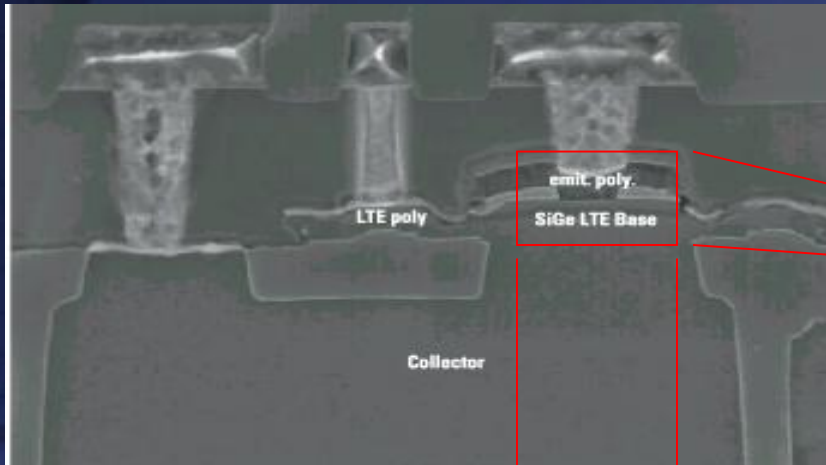
4a. GENERACION (70's)



Microchip Actual

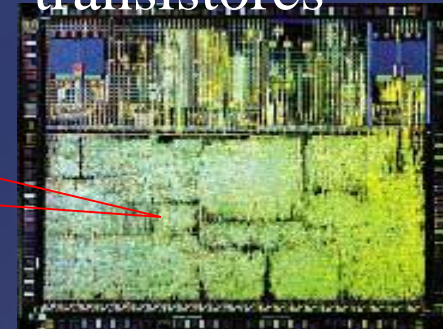


El transistor como celda unitaria



$0.13 \mu\text{m} = 1.3 \text{ cm} / 100,000$

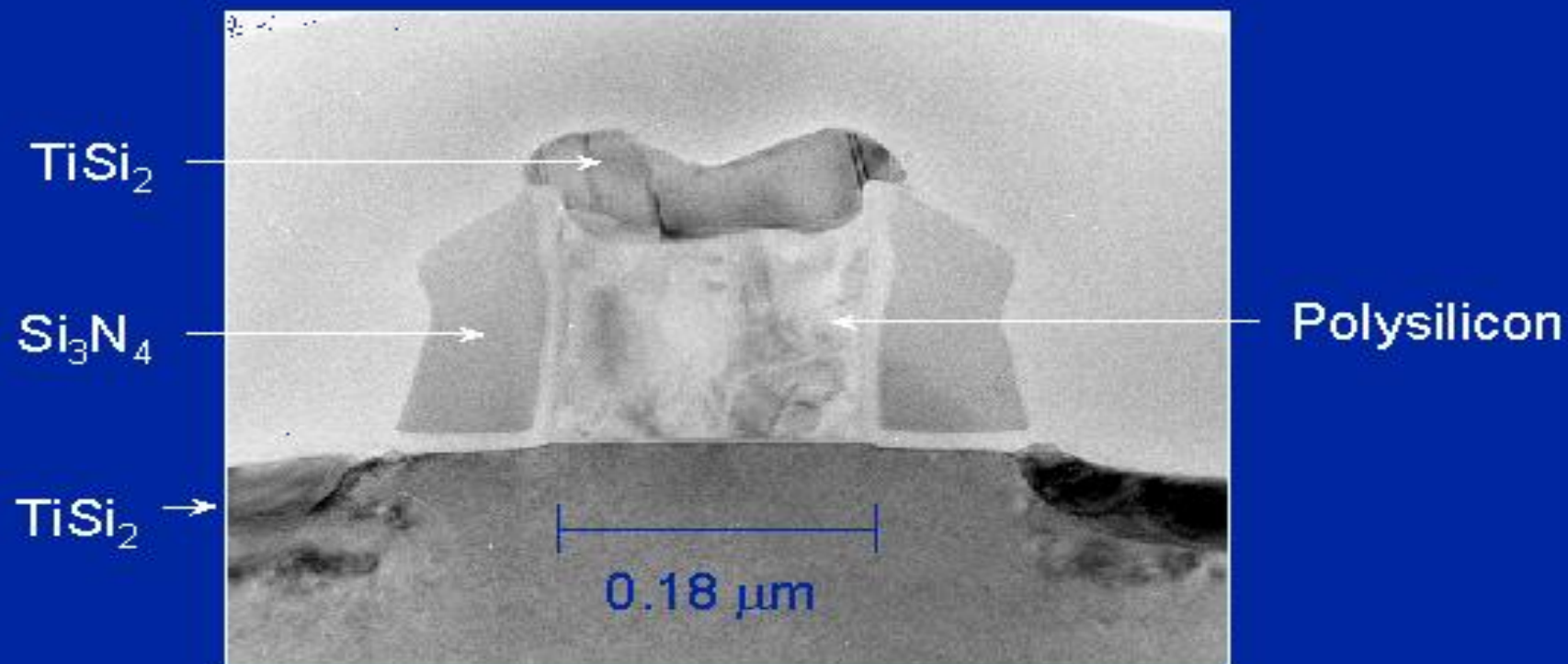
Microprocessor Con 7.5 millones de transistores



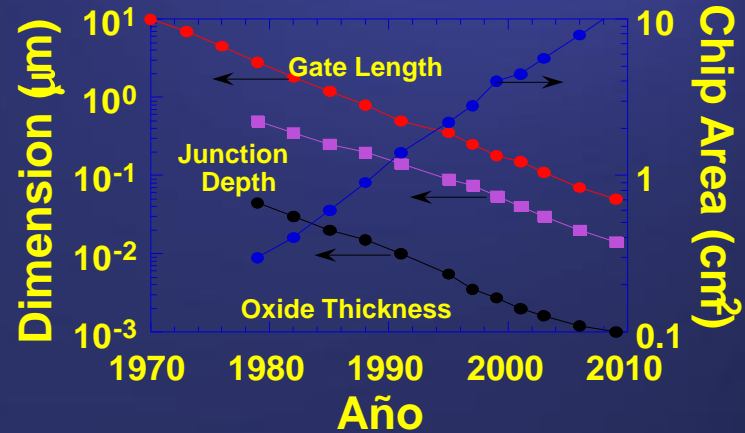
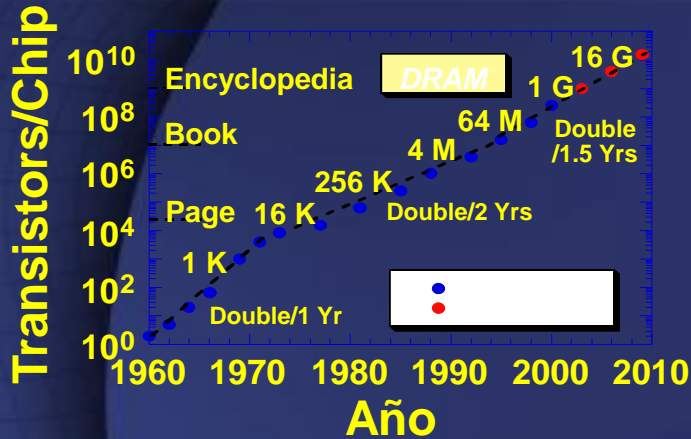
1.22
cm

1.35 cm

0.18 μ Transistor Cross-Section



almacenamiento de la información en ICs



- Ahora una auto en 1970 comparando con un circuito integrado

	1970	1993
velocidad (mph)	100	2,500,000
Potencia (mpg)	20	1,740,000
Costo (1992 \$)	20,000	40
Masa (kg)	1,500	0.06
Kilometros (Lt)	20	80,000

Intel Pentium 4 42,000,000 Transistors





Procesador Intel® Core™ i7-980X Extreme

Edition

3,33 GHz de velocidad del núcleo

Hasta 3,6 GHz con la tecnología Intel® Turbo Boost

6 núcleos y 12 subprocesos con la tecnología Intel® Hyper-Threading

12 MB de caché Intel® inteligente

3 canales de memoria DDR3 de 1066 MHz

Tecnología de procesamiento de fabricación de 32 nm

731 millones de transistores (1.170 millones en el Core i7 980x, con 6 núcleos y 12 MiB de memoria caché).

Sofisticada administración de energía, puede colocar un núcleo no utilizado en modo sin energía.

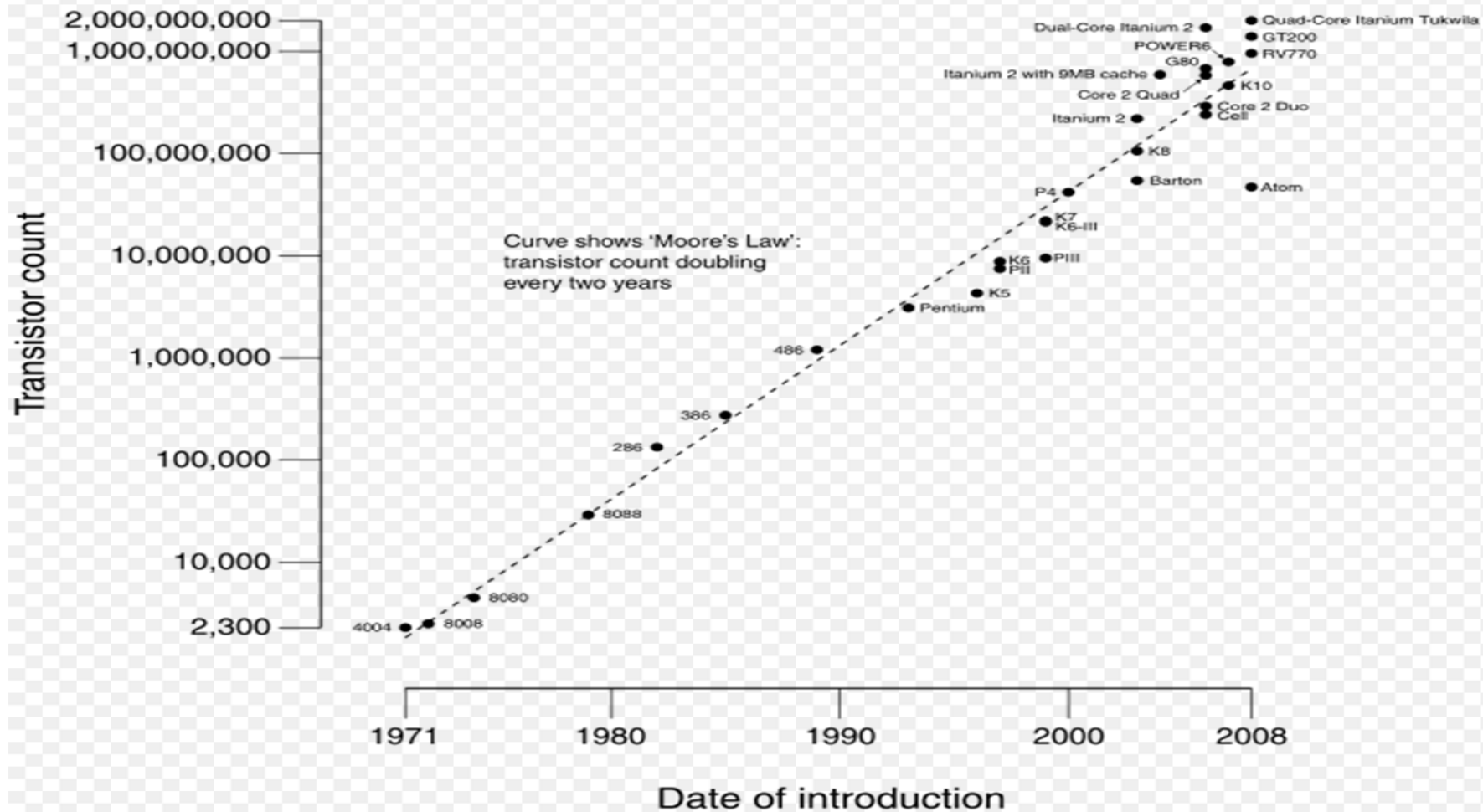
Capacidad de overclocking muy elevada

(se puede acelerar sin problemas hasta los 4-4,1 GHz).



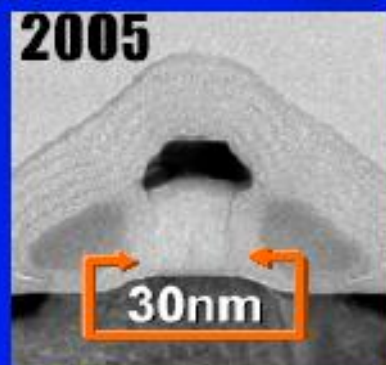
Fecha	Nombre	Número de transistores	Finura grabado (micrones)	Frecuencia Reloj	Ancho de datos	MIPS
1971	4004	2 300		108 kHz	4 bits / 4 bits de bus	
1974	8080	6 000	6	2 MHz	8 bits / 8 bits de autobús	0,64
1979	8088	29 000	3	5 MHz	16 bits / 8 bits de autobús	0,33
1982	80286	134 000	1,5	6 a 16 MHz (20 MHz de AMD)	16-bit bus bits/16	1
1985	80386	275 000	1,5	16 a 40 MHz	De 32-bit bus bits/32	5
1989	80486	1 200 000	1	16 a 100 MHz	De 32-bit bus bits/32	20
1993	Pentium	3 100 000	0,8 a 0,28	60 a 233 MHz	De 32-bit bus bits/64	100
1997	Pentium II	7 500 000	0,35 a 0,25	233-450 MHz	De 32-bit bus bits/64	300
1999	Pentium III	9 500 000	0,25 a 0,13	Desde 450 hasta 1 400 MHz	De 32-bit bus bits/64	510
2000	Pentium 4	42 000 000	0,18 a 0,085	1,3 a 3,8 GHz	De 32-bit bus bits/64	1 700
2004	Pentium 4D "Prescott"	125 000 000	0,09 a 0,085	2,66 a 3,6 GHz	De 32-bit bus bits/64	9 000
2006	Core 2 Duo™	291 000 000	0,085	2.4 GHz (E6600)	De 64-bit bus bits/64	22 000
2007	Core 2 Quad™	2 291 000 * 000	0,085	3 GHz (Q6850)	De 64-bit bus bits/64	2 * 22 000 (?)
2008	Core 2™ Duo (Penryn)	410 000 000	0,045	3.33 GHz (E8600)	De 64-bit bus bits/64	~ 24 200
2008	Core™ 2 Quad (Penryn)	2 410 000 * 000	0,045	3.2GHz (QX9770)	De 64-bit bus bits/64	~ 2 * 24 200
2008	Intel Core i7 (Nehalem)	731 000 000	0,045 (2008) 0,032 (2009)	2,66 GHz (Core i7 920) 3.33 GHz (Core i7 ext. Ed. 975)	De 64-bit bus bits/64	?
2009	Intel Core i5/i7 (Lynnfield)	774 000 000	0,045 (2009)	2,66 GHz (i5 750) 2,93 GHz (Core i7 870)	De 64-bit bus bits/64	76383
2010	Intel Core i7 (Gulftown)	1 170 000 000	0,032	3.33 GHz (Core i7 980X)	De 64-bit bus bits/64	147600

CPU Transistor Counts 1971-2008 & Moore's Law

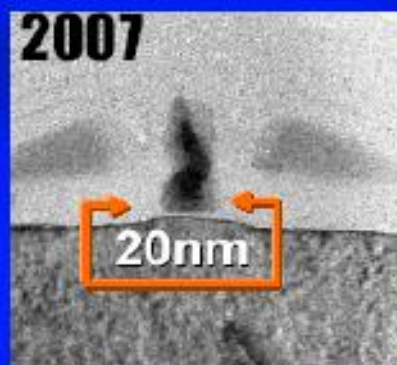




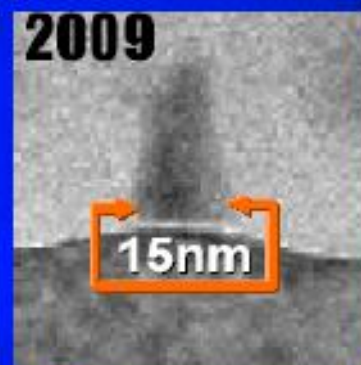
0.13µm process



65nm process

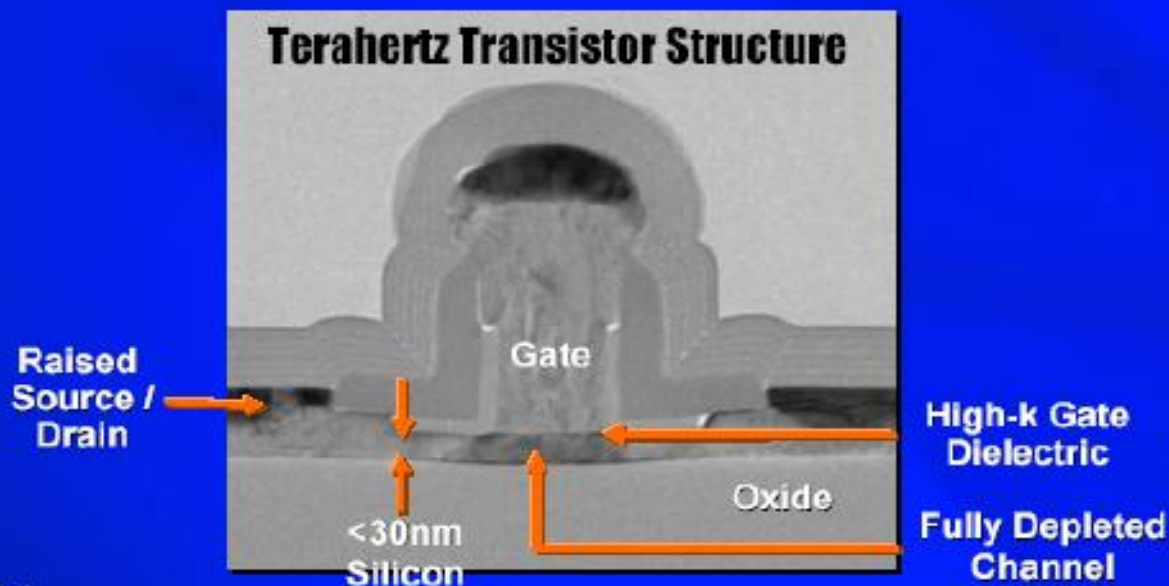


45nm process



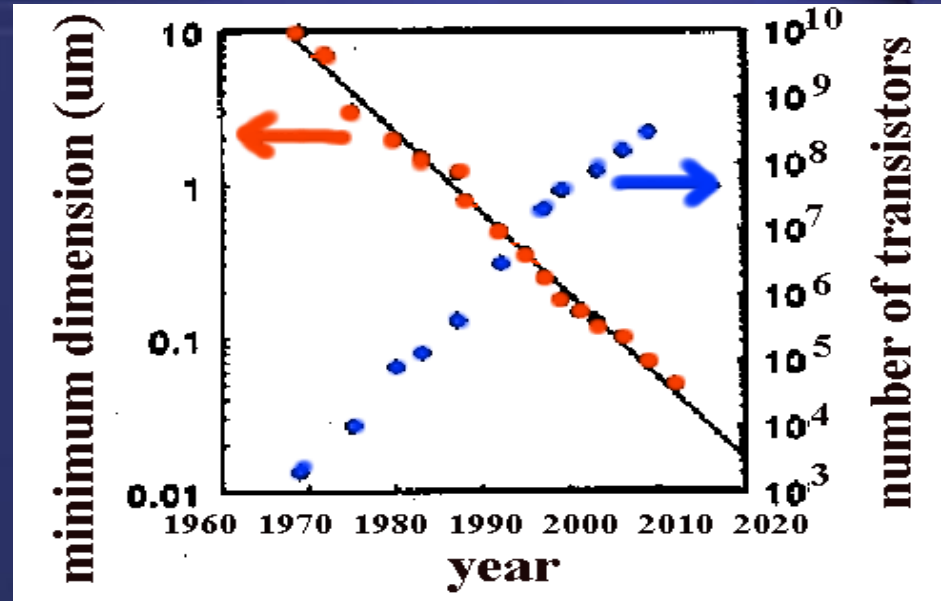
32nm process

Terahertz Transistor Structure



Moore's law

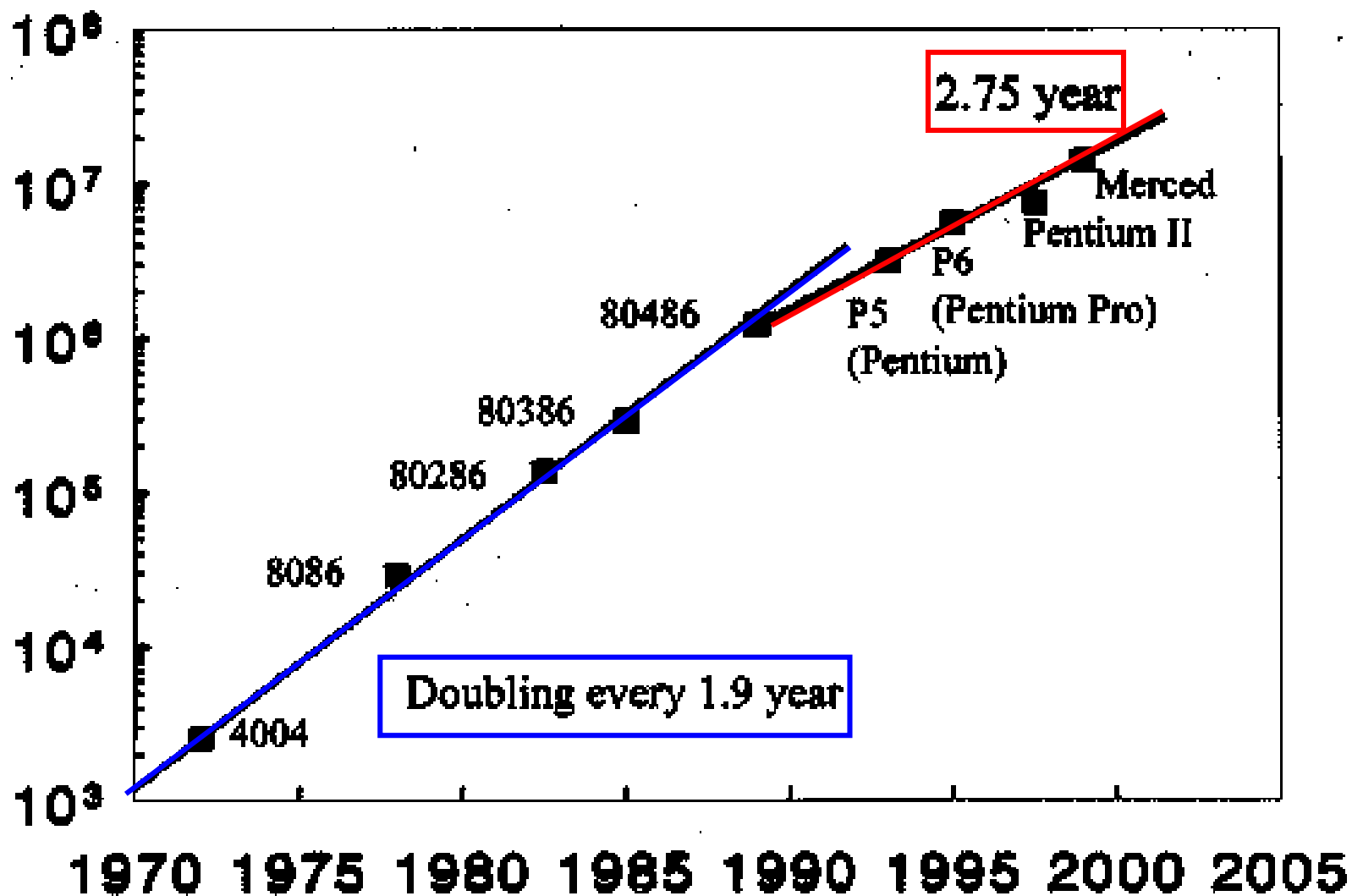
- Doubling of the transistor density on a manufactures die every 1.5 years (by scaling of dimensions with a factor 0.7 every three years)
- At the same time the switching speed increases with a factor 1.4 every three years
- At the same time the cost per function decreases with 35% per year



• Exponential growth for over 30 years!

• Gordon Moore, Electronics, pg. 114 (1965)

Number of Transistors





CUVyTT-BUAP



**Microprocesador
encapsulado**



oblea

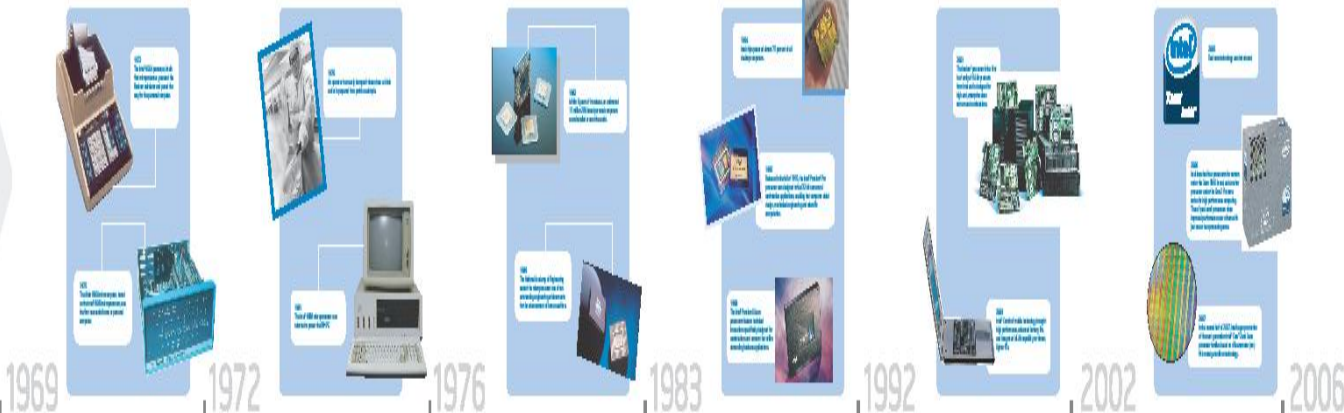
THE EVOLUTION OF A REVOLUTION

EXPLORE THE INTEL TECHNOLOGY INNOVATIONS THAT HAVE CHANGED THE WORLD.



The Revolution Begins

Throughout history, new and improved technologies have transformed the human experience. In the 20th century, the pace of change sped up radically as we entered the computing age. For nearly 40 years Intel innovations have continuously created new possibilities in the lives of people around the world.



The Revolution Continues

Intel continues to deliver on the promise of Moore's Law with the introduction of powerful multi-core technologies, transforming the way we live, work, and play once again.

Moore's Law
 Intel's 18th Annual Gordon Moore Award for the number of transistors on the world's most advanced microprocessors has increased by a factor of approximately 100 percent for a year, performance doubling every 18 months.

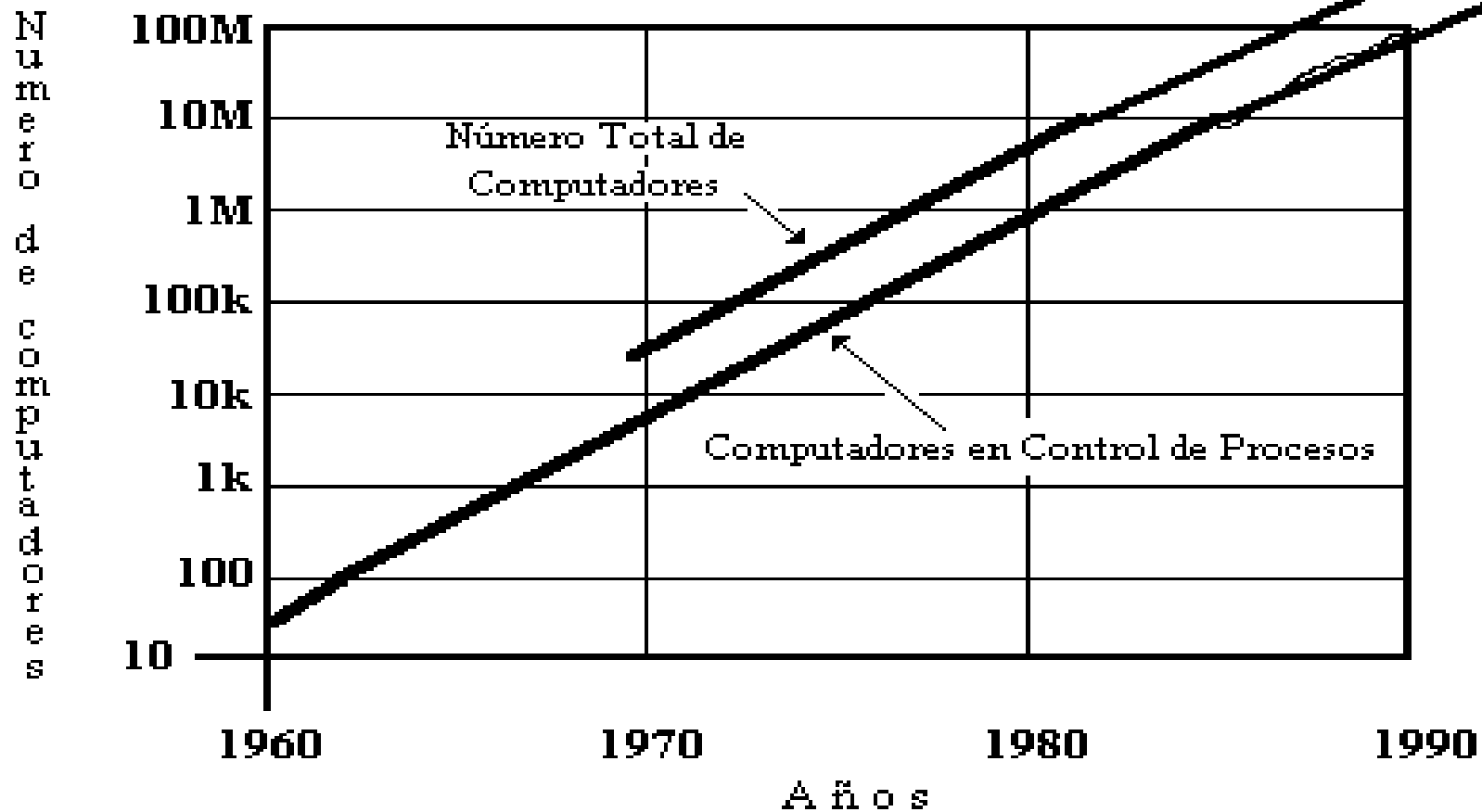
Intel 100K process transistor 100K 100K Hz 2,300 transistors 10µm	Intel 8008 process transistor 6K 500-800 KHz 3,500 transistors 10µm	Intel 8080 process transistor 6K 2 MHz 4,500 transistors 6µm	Intel 8085 process transistor 6K 5 MHz 29,000 transistors 3µm	Intel 8088 process transistor 6K 5 MHz 29,000 transistors 3µm	Intel 8086 process transistor 29K 6 MHz 134,000 transistors 1.5µm	Intel 286 process transistor 275K 16 MHz 275,000 transistors 1.5µm	Intel 386 process transistor 2.75M 25 MHz 1,200,000 transistors 1µm	Intel Pentium process transistor 3.1M 66 MHz 3,100,000 transistors 0.8µm	Intel Pentium 2 process transistor 5.5M 200 MHz 5,500,000 transistors 0.6µm	Intel Pentium 3 process transistor 7.5M 300 MHz 7,500,000 transistors 0.25µm	Intel Pentium 4 process transistor 9.5M 500 MHz 9,500,000 transistors 0.18µm	Intel Pentium 5 process transistor 15.5M 1.5 GHz 42,000,000 transistors 0.18µm	Intel Pentium D process transistor 291M 1.7 GHz 55,000,000 transistors 90nm	Intel Pentium E process transistor 291M 1 GHz 220,000,000 transistors 0.13µm
Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100	Transistors per die increased by a factor of 100

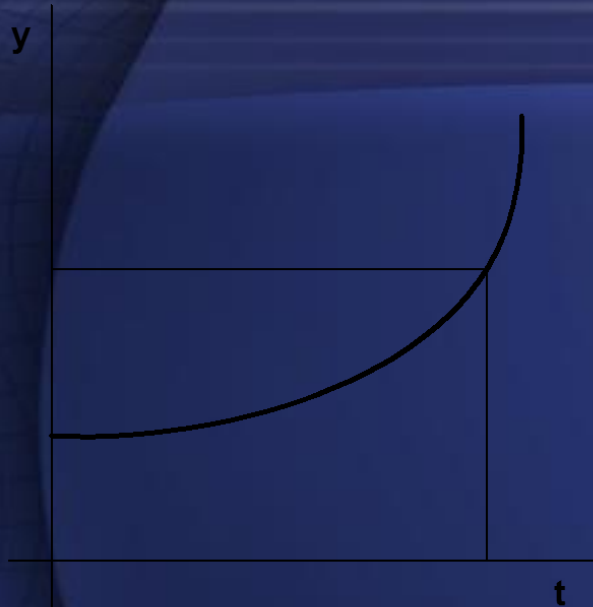


5a. GENERACION (80's)

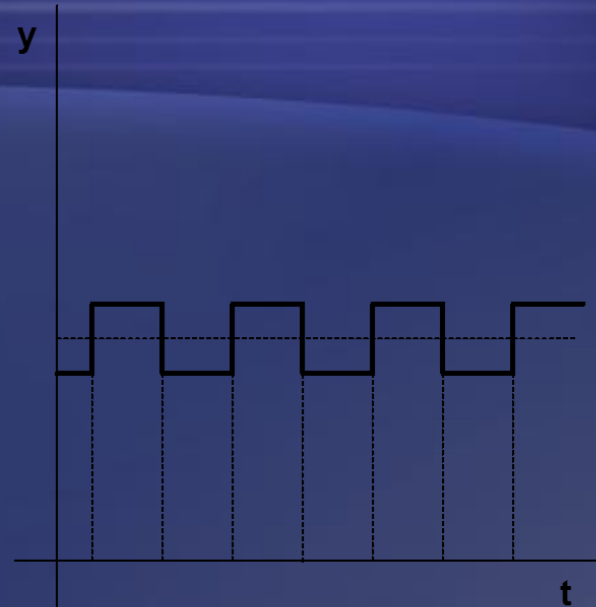
- AVANCES EN LA PROGRAMACION PRINCIPALMENTE
- MAYOR AUMENTO EN LA CAPACIDAD Y VELOCIDAD
- AUMENTO SIGNIFICATIVO EN LA CAPACIDAD DE LAS MEMORIAS
- DESARROLLO DE LA INTELIGENCIA ARTIFICIAL
- DESARROLLO DE LOS SISTEMAS ABIERTOS

Computadoras en Control de Procesos





Variable Analógica



Variable Digital

TIPOS DE VARIABLES DE ACUERDO A SU COMPORTAMIENTO EN EL TIEMPO



LAS SEÑALES SE CLASIFICAN DE ACUERDO A LA INFORMACIÓN ÚTIL QUE TRANSPORTAN



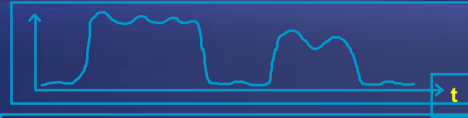


SEÑALES ANALÓGICAS



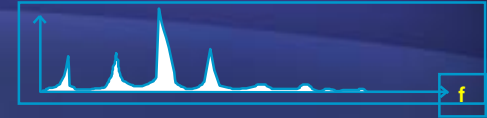
DC

- Temperatura
- Presión
- Flujo
- Esfuerzo



Dominio del tiempo

- ECG
- Presión de sangre
- Transientes
- Cromatografía



- Vibración
- Voz
- Sonar

Consideraciones para seleccionar tarjetas

Exactitud

Resolución

Resolución

Tasa de muestreo

Tasa de muestreo

Exactitud

Exactitud

Disparos

Disparo

Filtros

SEÑALES DIGITALES



Señal On-Off

Entrada:
Cierre o apertura de un interruptor

Salida:
Abre o cierra una válvula



Tren de pulsos

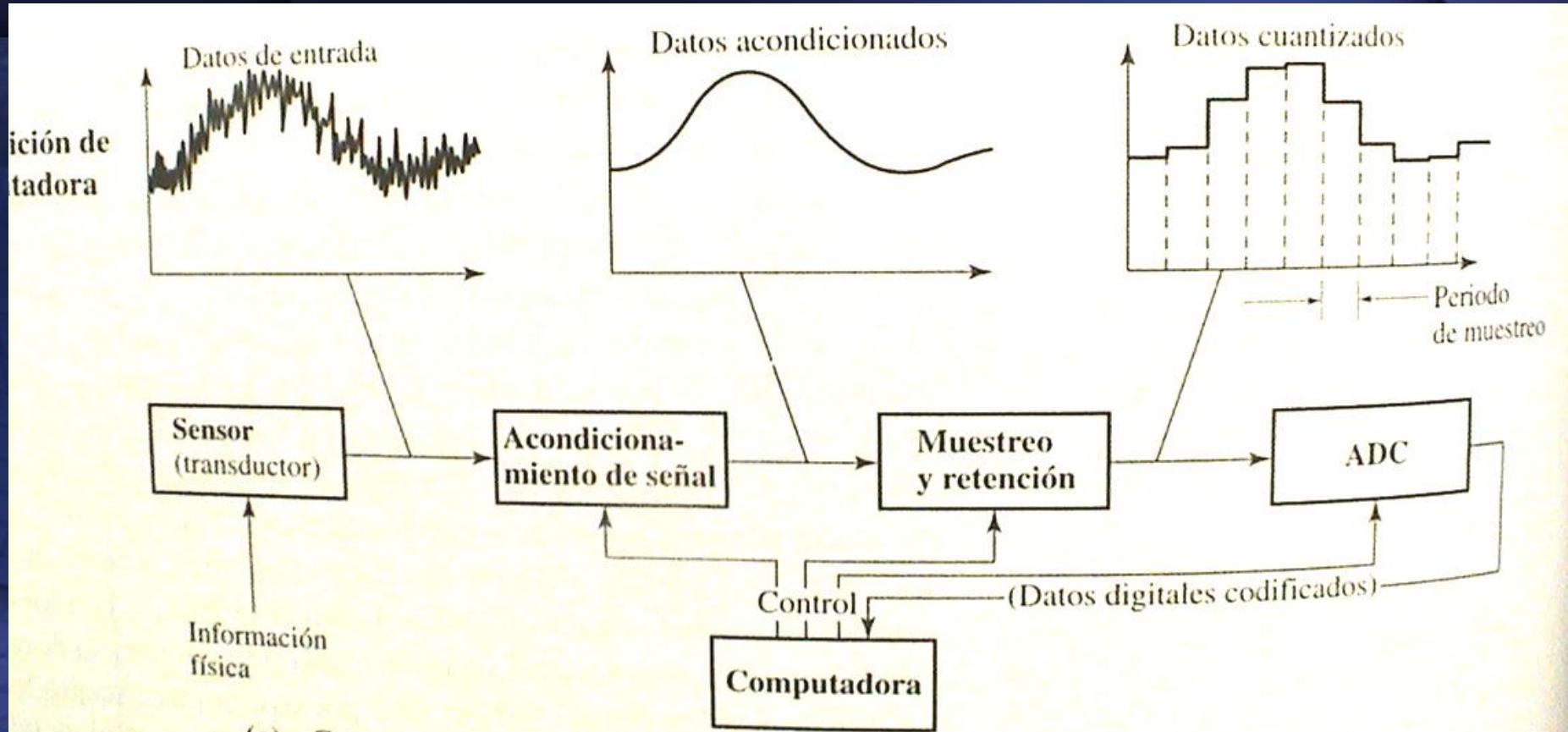
Entrada:
Lectura de un encoder

Salida:
Mueve un motor a pasos

Consideraciones para seleccionar tarjetas

Voltaje y corriente de salida
Número de canales

Resolución
Frecuencia de reloj



(a) Componentes funcionales de adquisición de datos analógicos por computadora



SEÑALES DISCRETAS O DIGITALES

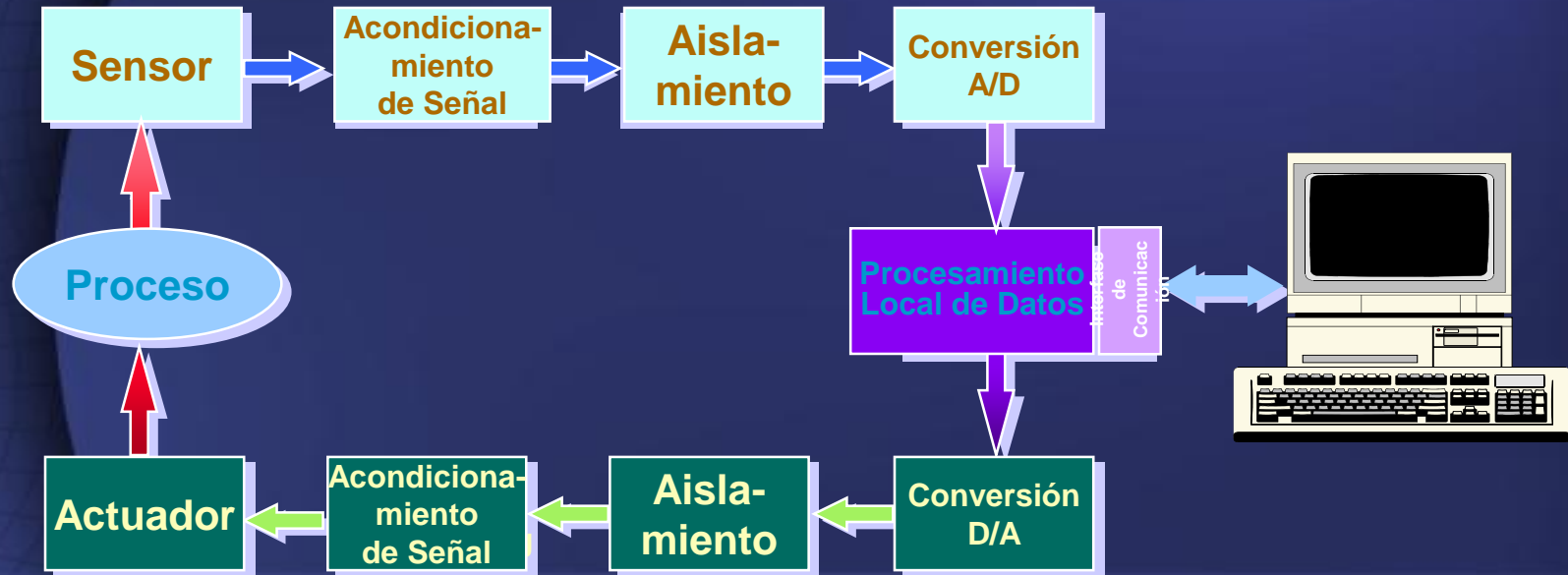
- DEFINEN CUALQUIER MAGNITUD O EVENTO
- PUEDEN SER REPRESENTADAS EN DIFERENTES SISTEMAS NUMERICOS
- SE PUEDEN PROCESAR, COMBINAR Y SECUENCIAR CON MAYOR FACILIDAD
- SE PUEDEN MODULAR CON MAS FACILIDAD Y CONFIABILIDAD
- SE PUEDEN MULTIPLEXAR CONFIABLEMENTE
- EL EFECTO DE SU DEGRADACION ES MENOR QUE CON LAS DIGITALES
- REQUIEREN MENOS ESPACIO DE MEMORIA PARA SU ALMACENAMIENTO

ÁREAS

- *a) Conocimiento del proceso.*
- *b) Tecnología de la medida.*
- *c) Tecnología de las computadoras.*
- *d) Teoría de control.*

CUV y TT-BUAP

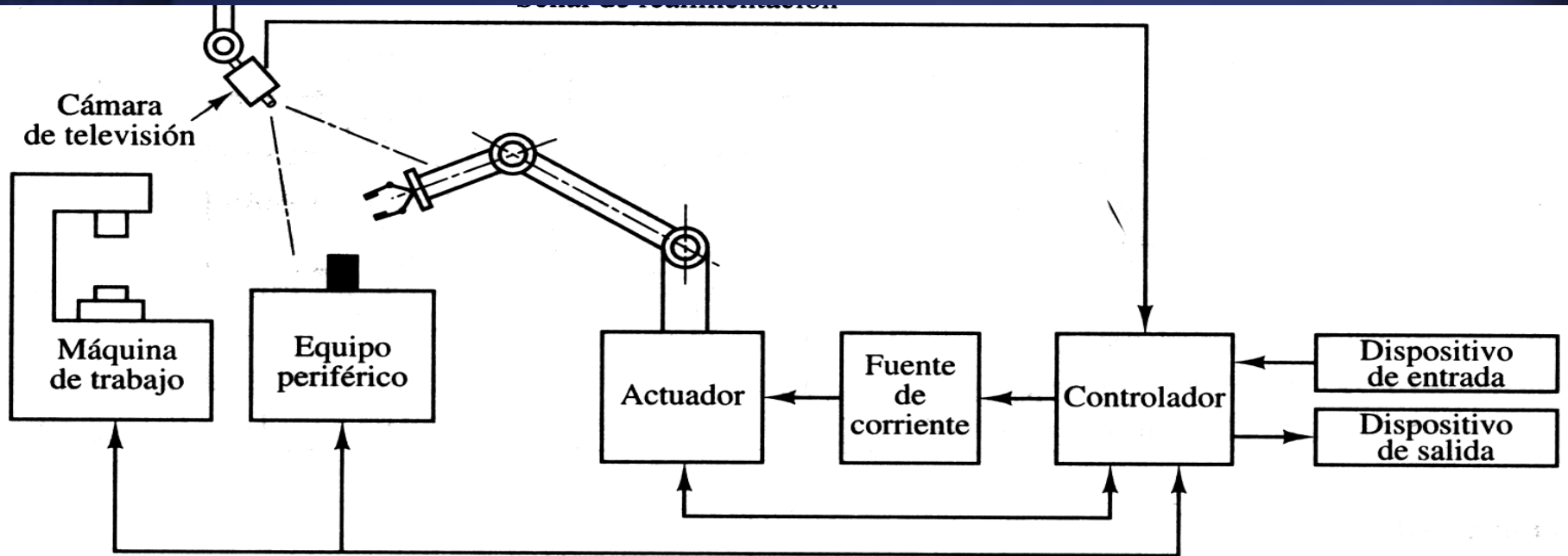
Lazo de Adquisición de Datos y Control





Fundamentos

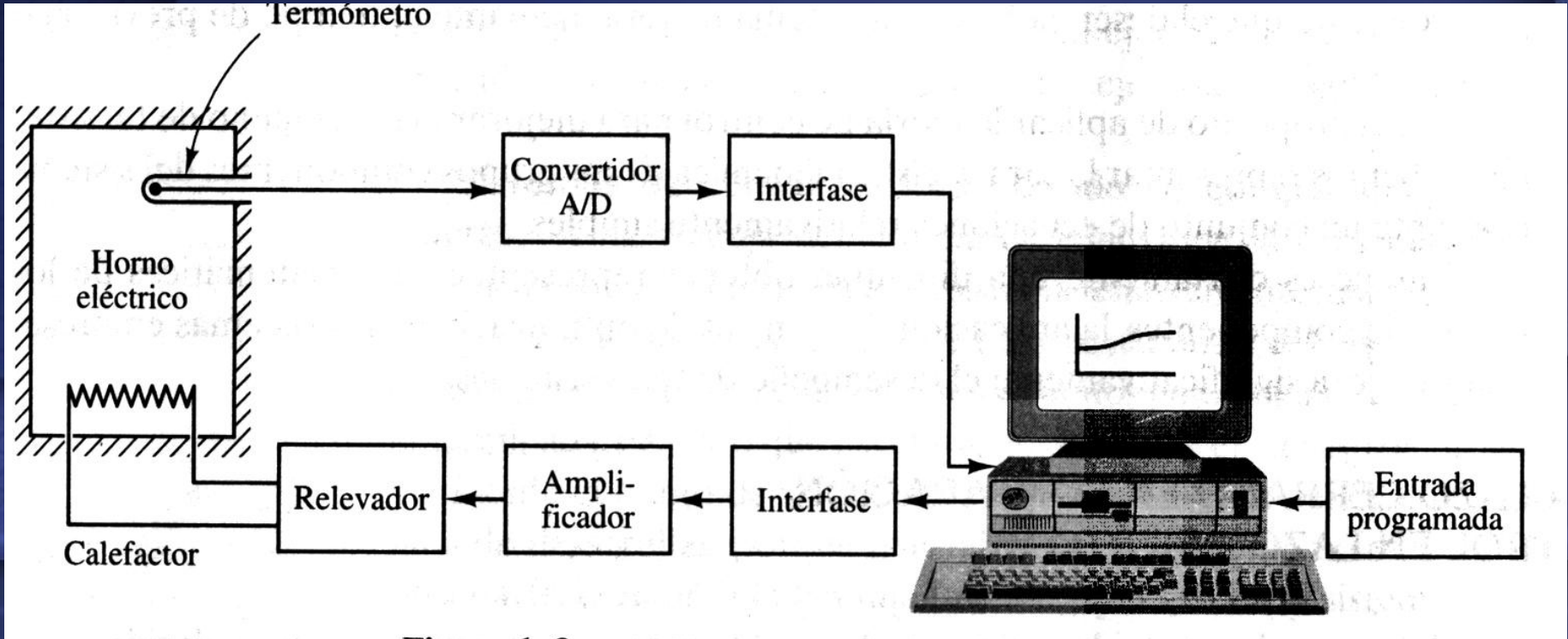
Sistema de Control de un Robot

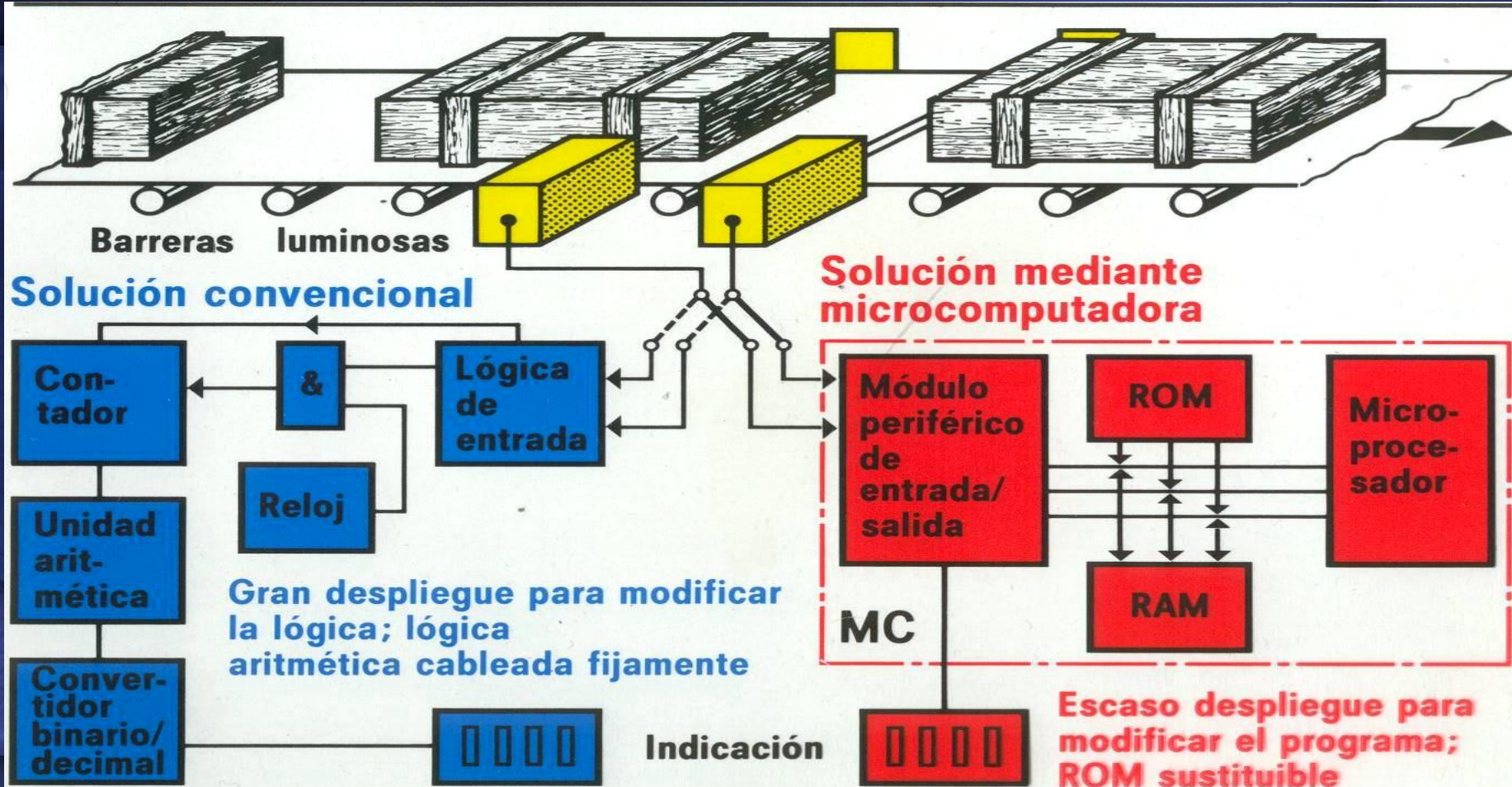




Fundamentos

Sistema de Control de temperatura





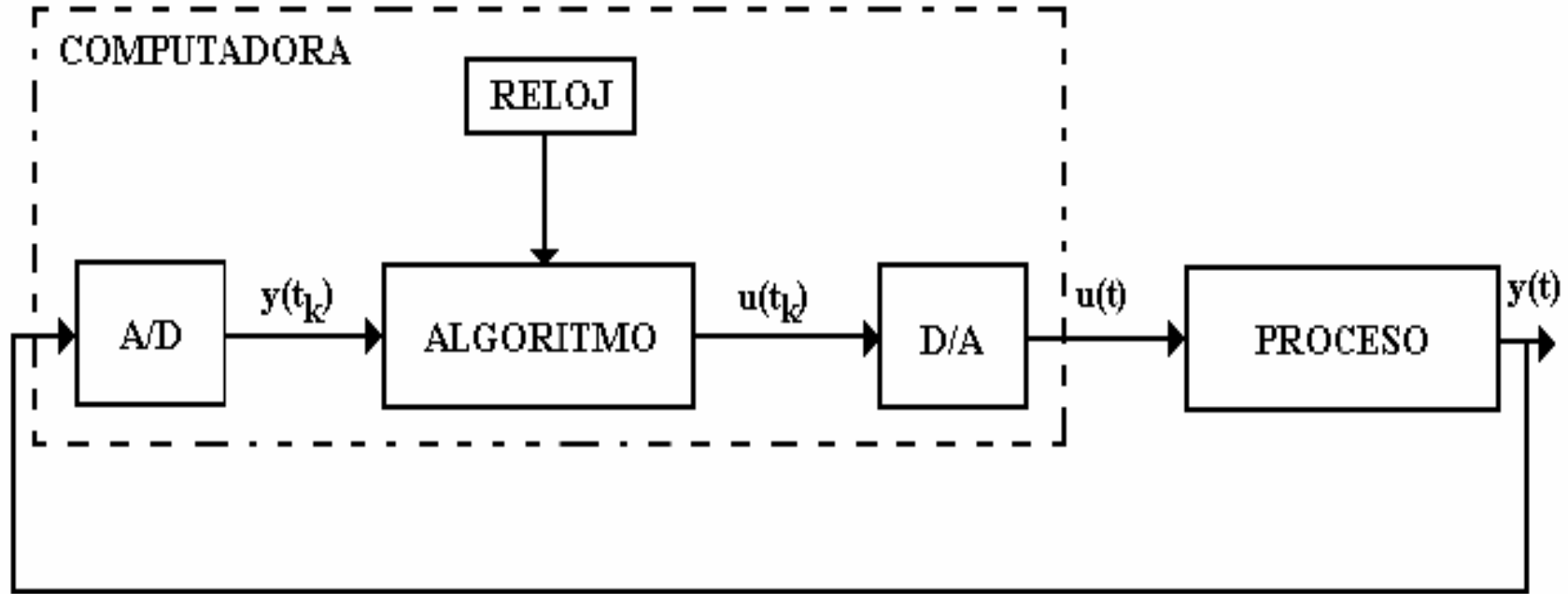


Figura 1.3 Diagrama esquemático de un sistema controlado por computadora.

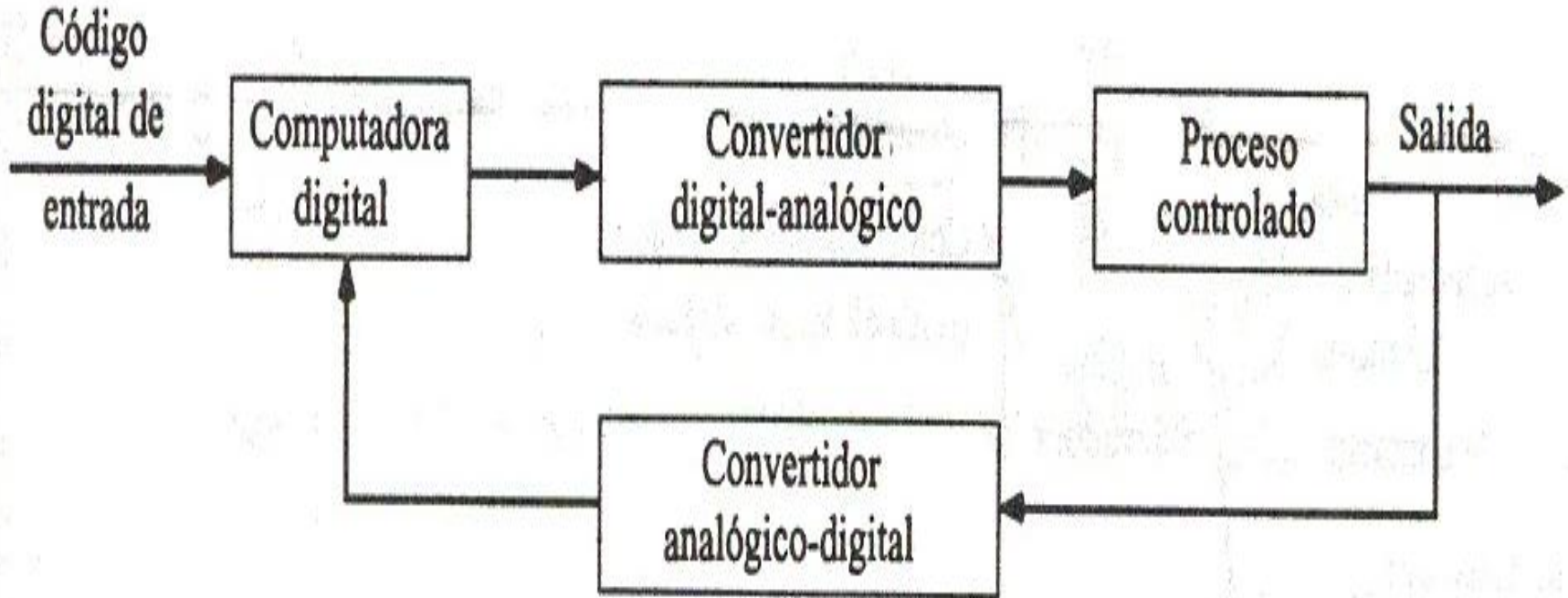
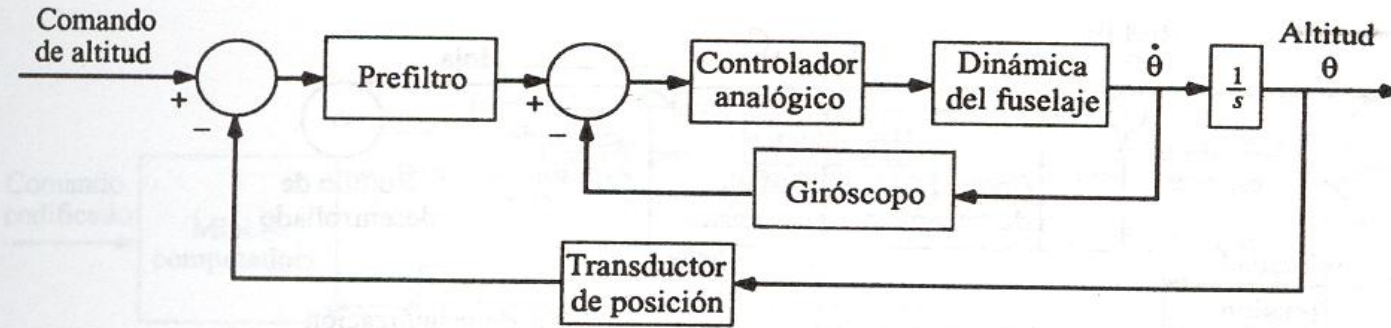
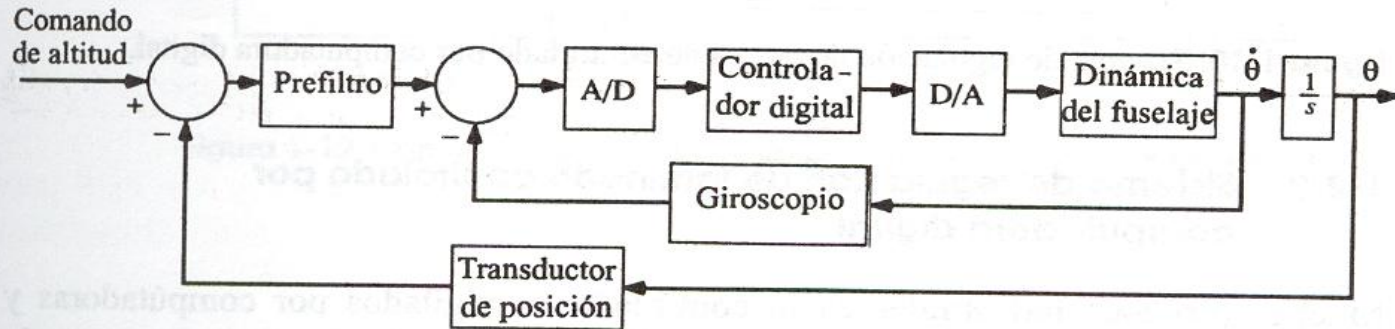


Figura 1-3. Sistema de control digital común.



(a)



(b)

Figura 1-8. a) Sistema de control simplificado de un solo eje para un piloto automático con datos analógicos. b) Sistema de control simplificado de un solo eje para un piloto automático con datos digitales.

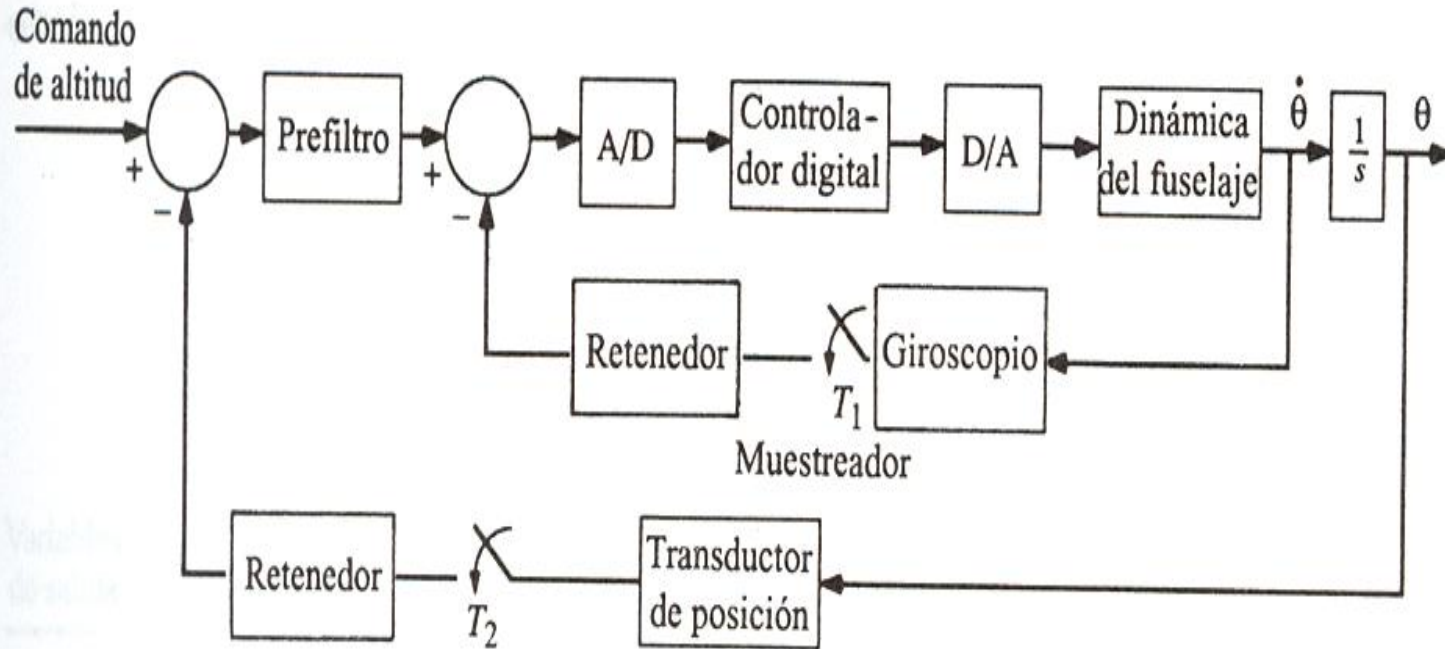


Figura 1-9. Sistema de control digital simplificado de un solo eje para un piloto automático con muestreo múltiple.

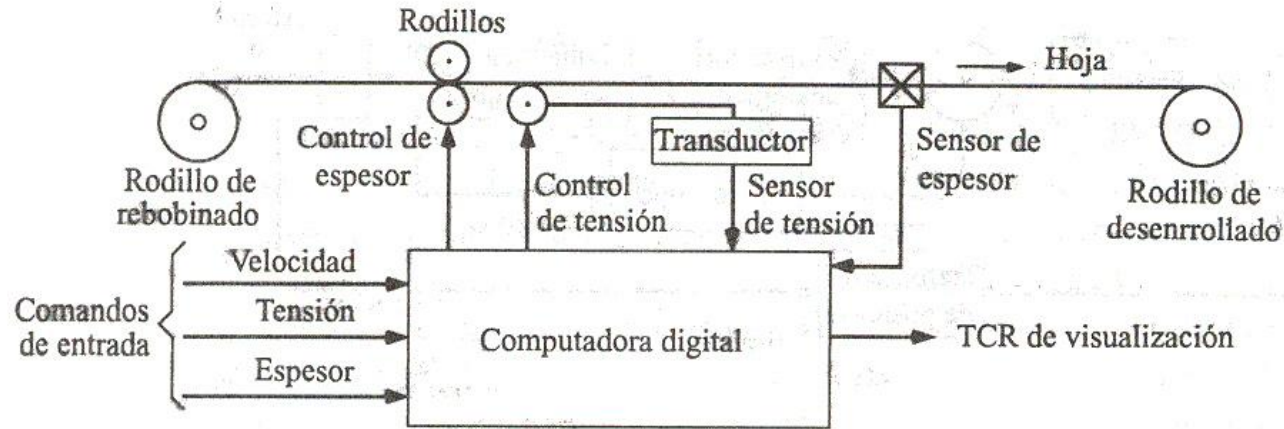


Figura 1-10. Sistema de regulación de laminado controlado por computadora digital.

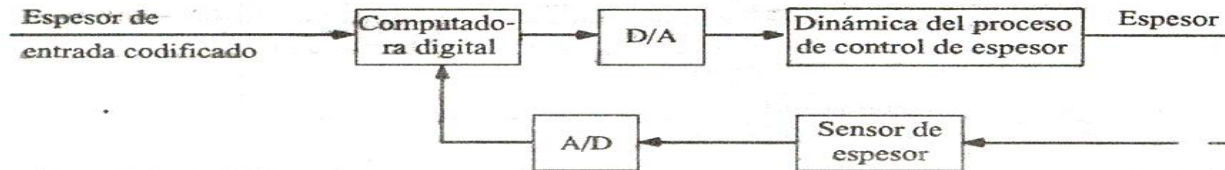


Figura 1-11. Control de espesor del sistema de regulación de laminado.

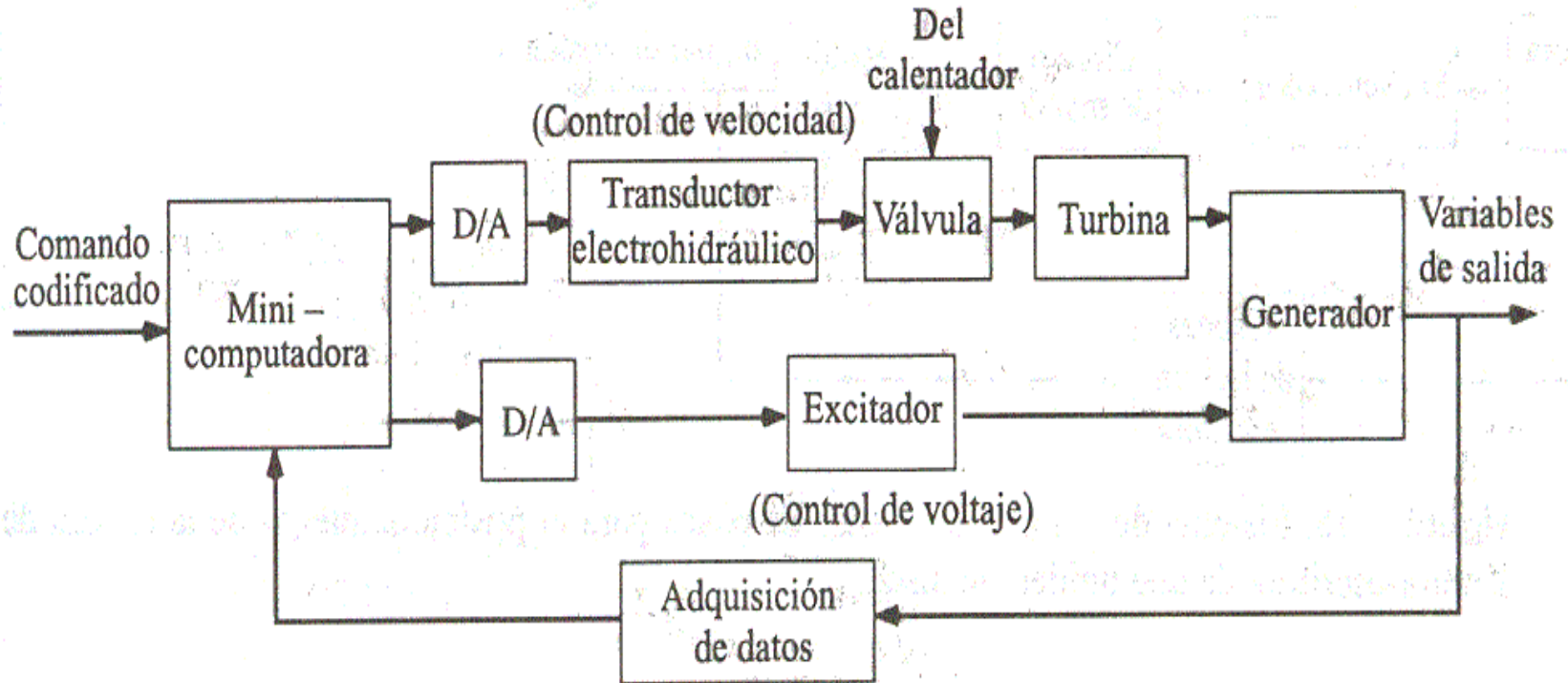


Figura 1-12. Control por computadora de una unidad de turbina y generador.

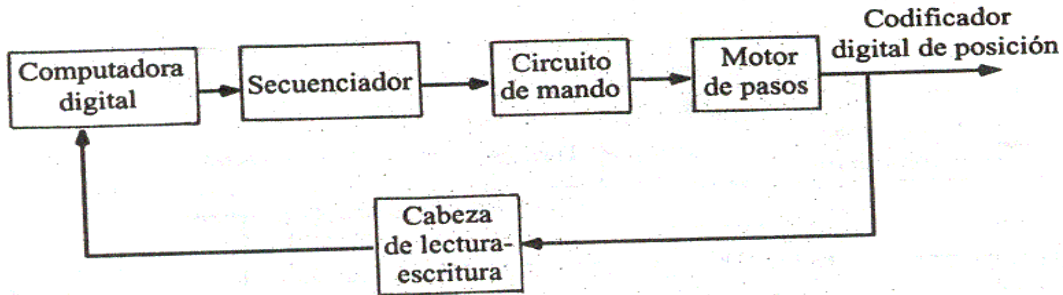


Figura 1-15. Sistema de control del motor de pasos para el posicionamiento de la cabeza de lectura-escritura de una unidad de disco.

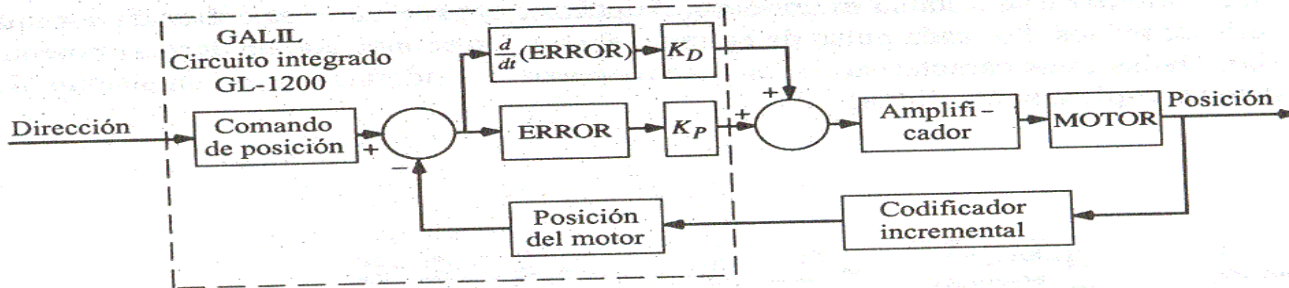


Figura 1-16. Sistema de control de posición con el controlador de un solo circuito integrado GL1200.

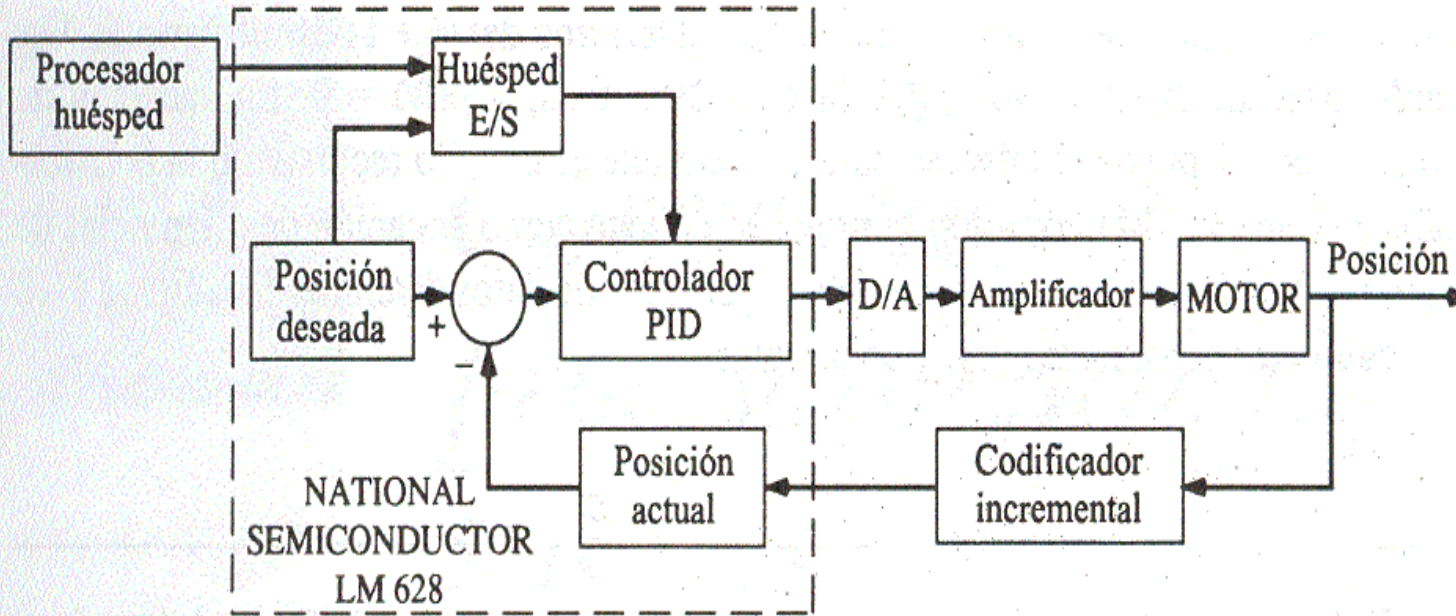


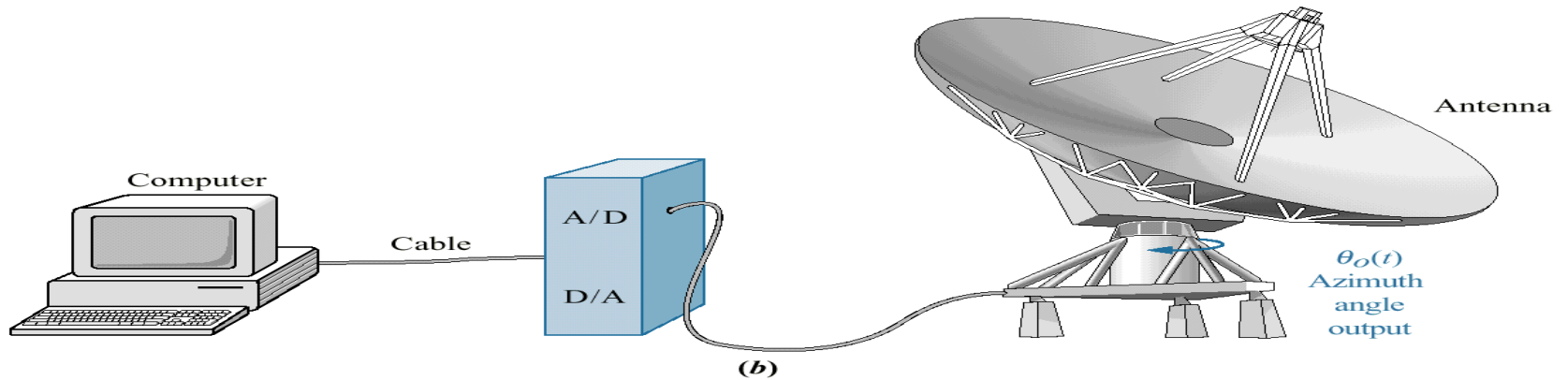
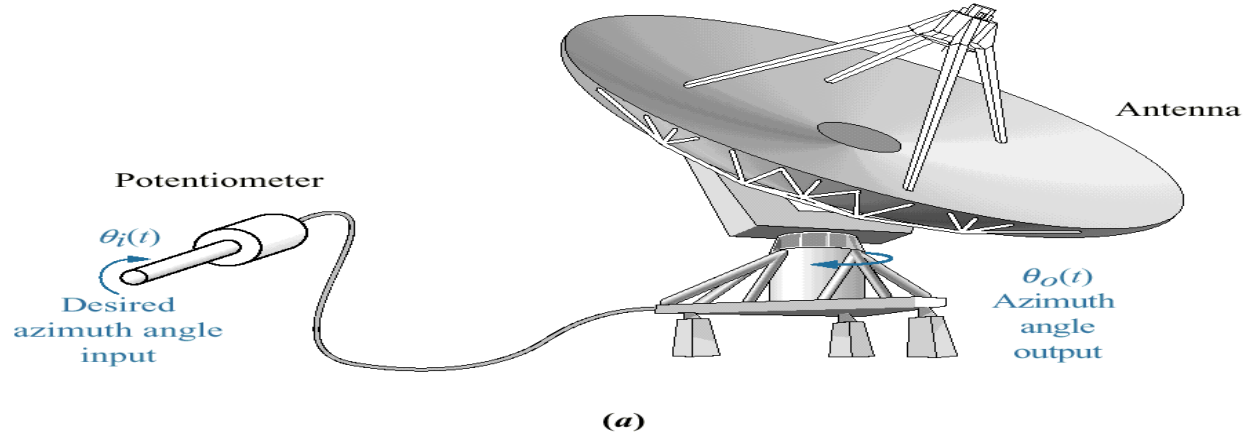
Figura 1-17. Sistema de control de posición con el microprocesador de un solo circuito integrado LM628.

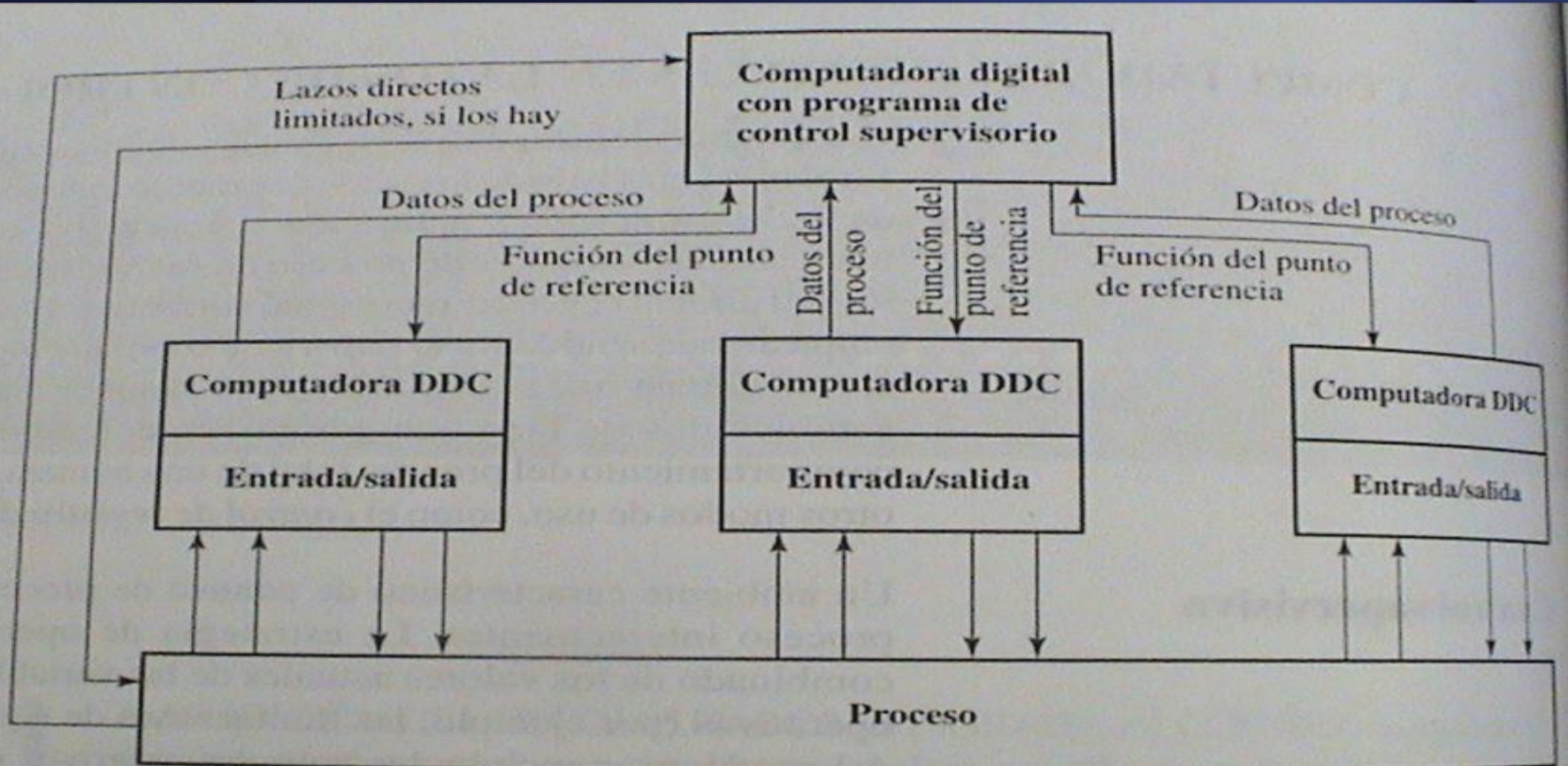
Figure 13.1

Conversion of antenna azimuth position control system from:

a. analog control to

b. digital control

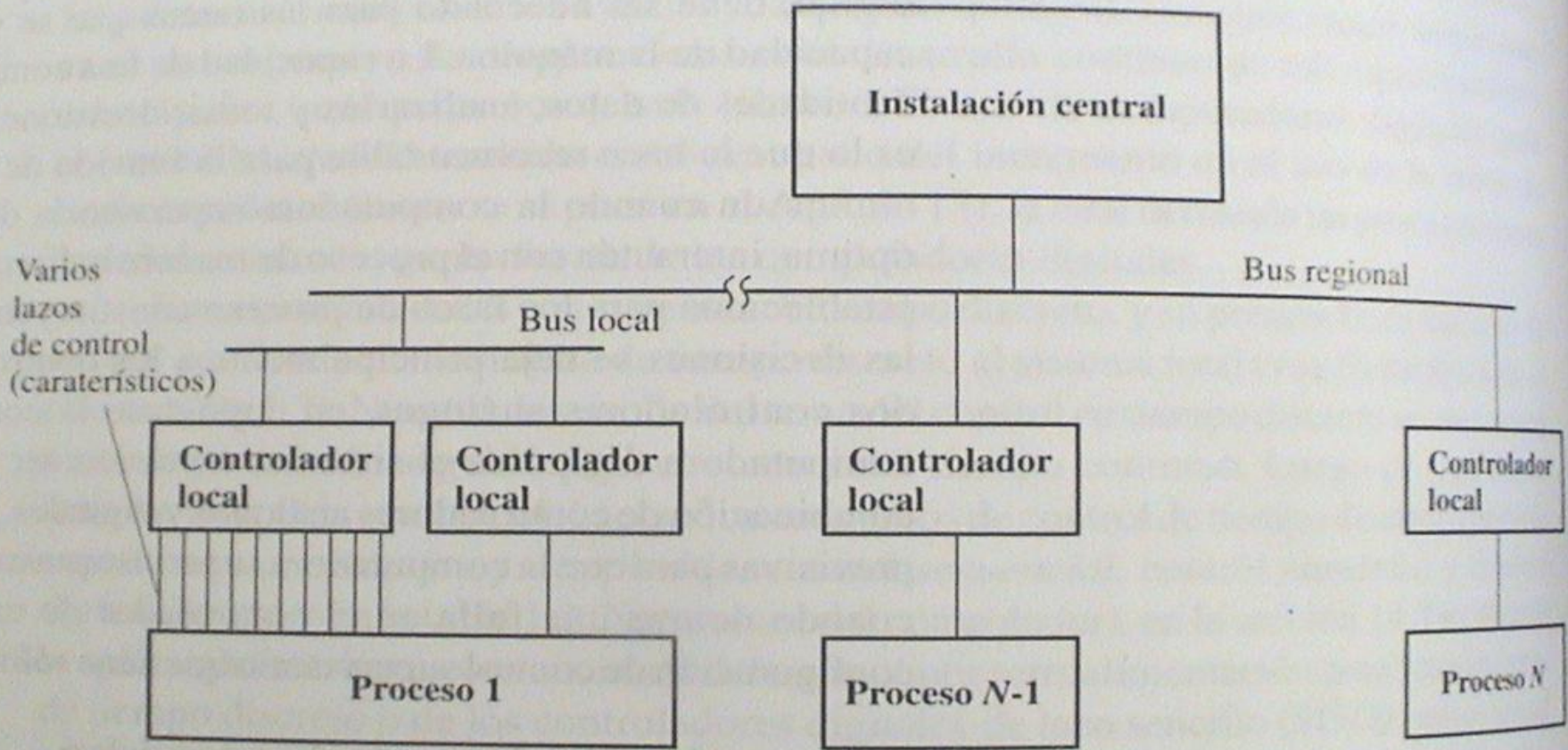




(a) Sistema de control supervisorio



CUVyTT-BUAP



(b) Un sistema de control distribuido



Características del Control Digital

Como características básicas del control digital se pueden mencionar las siguientes:

- No existe límite en la complejidad del algoritmo. Cosa que sí sucedía anteriormente con los sistemas analógicos.
- Facilidad de ajuste y cambio. Por el mismo motivo anterior un cambio en un control analógico implica, en el mejor de los casos, un cambio de componentes si no un cambio del controlador completo.
- Exactitud y estabilidad en el cálculo debido a que no existen derivadas u otras fuentes de error.



- Uso de la computadora con otros fines (alarmas, archivo de datos, administración, etc.)
- Costo vs. número de lazos. No siempre se justifica un control digital ya que existe un costo mínimo que lo hace inaplicable para un número reducido de variables.
- Tendencia al control distribuido o jerárquico. Se ha pasado de la idea de usar un único controlador o computador para toda una planta a la de distribuir los dispositivos inteligentes por variable o grupos de estas e ir formando estructuras jerárquicas. En cuanto a la arquitectura de un lazo de control es



RAZONES DEL CONTROL

- **SEGURIDAD**
- **ESTABILIDAD**
- **OPTIMIZACION**
- **PROTECCION AMBIENTAL**

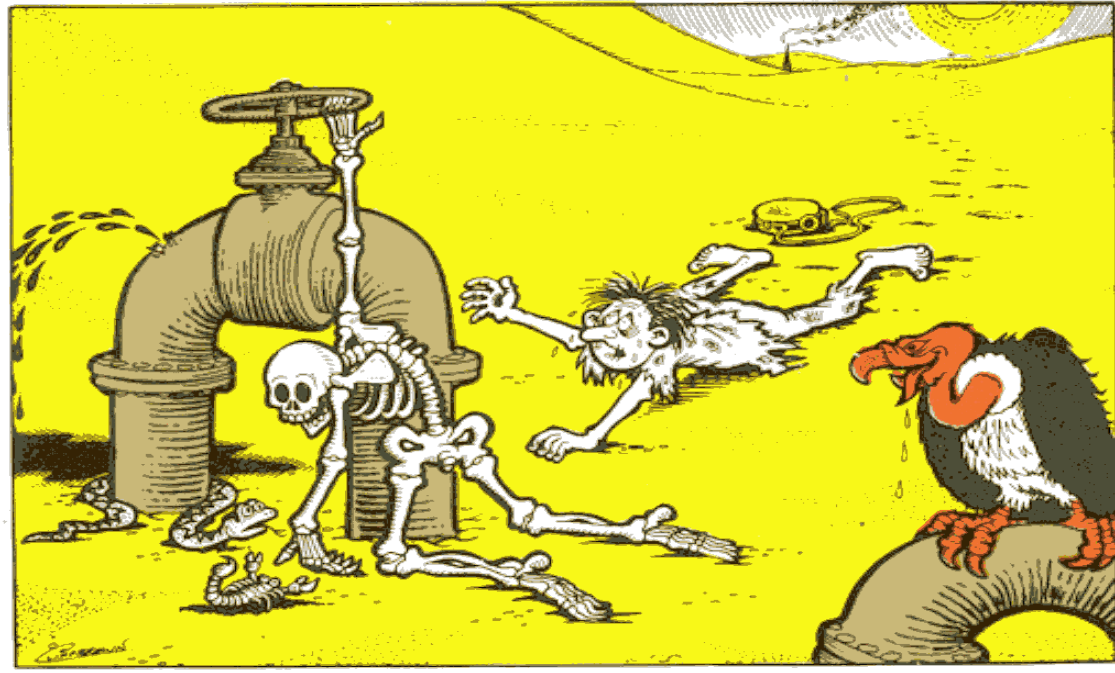


SEGURIDAD

**PRESERVAR BAJO CUALQUIER CONDICION
LA INTEGRIDAD DEL PERSONAL Y EQUIPO
INVOLUCRADO EN LA OPERACIÓN DE LOS
PROCESOS**



SEGURIDAD

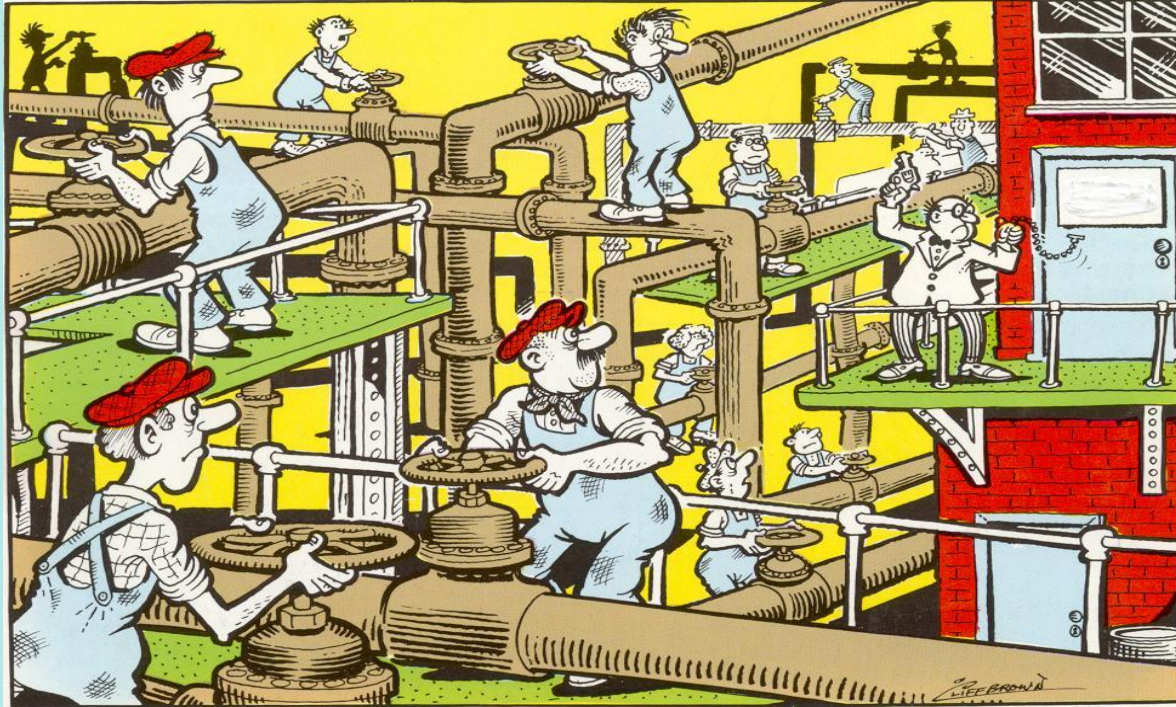


SEGURIDAD



ESTABILIDAD

**ASEGURAR LAS CONDICIONES DE
OPERACIÓN DE LOS PROCESOS PARA
MANTENER CONTINUAMENTE LA
CALIDAD DE LOS PRODUCTOS DENTRO
DE LOS LIMITE ESPECIFICADOS**

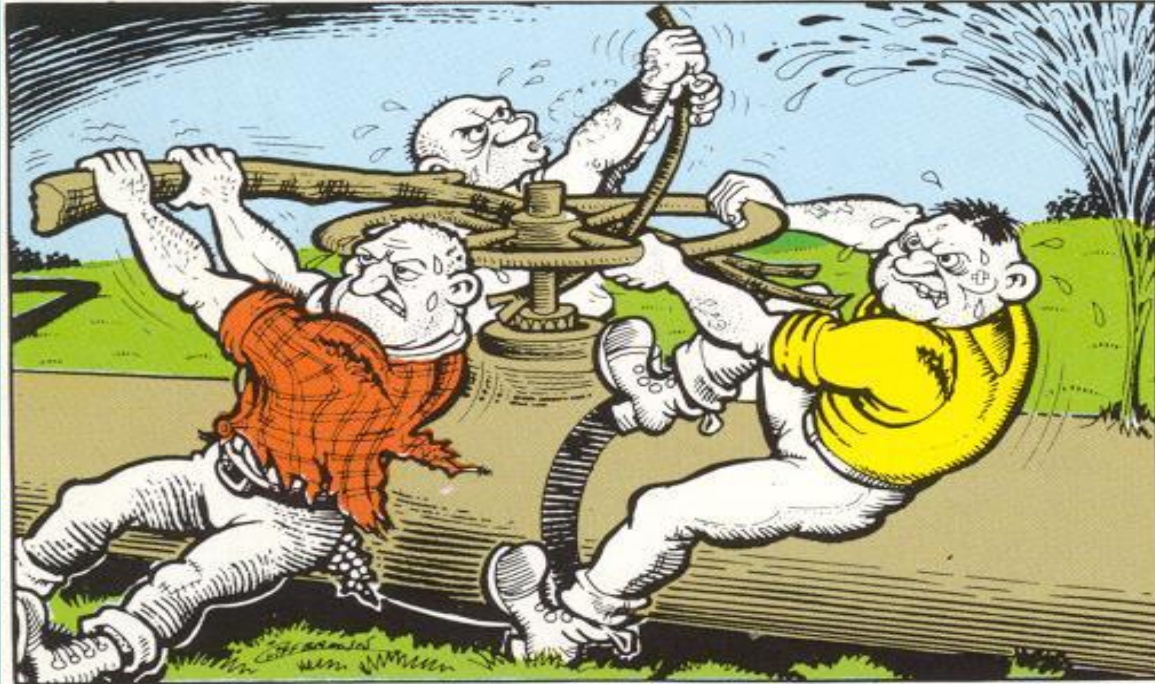


ESTABILIDAD



OPTIMIZACION

**ASEGURAR EL MAXIMO BENEFICIO
ECONOMICO EN LA OPERACIÓN DE
LOS PROCESOS INDUSTRIALES**

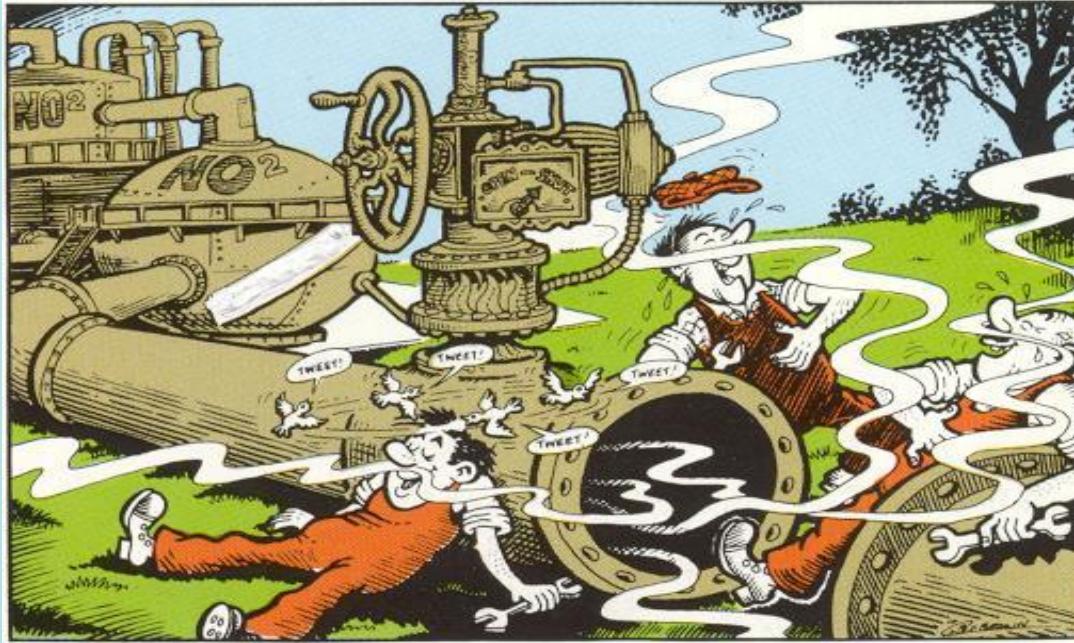


OPTIMIZACION



PROTECCION AMBIENTAL

**REDUCIR A SU MINIMA EXPRESION EL
IMPACTO ECOLOGICO DE LOS EFLUENTES
DEL PROCESO, PARA CUMPLIR CON TODAS
LAS NORMATIVIDADES APLICABLES**



PROTECCION AMBIENTAL



SISTEMAS DE INSTRUMENTACION Y CONTROL

CONSTITUYEN ARREGLOS DE DISPOSITIVOS MUTUAMENTE INTERRELACIONADOS, QUE CONFORMAN PATRONES ESPECÍFICOS DE COMPORTAMIENTO, TENDIENTES A PERMITIR LA OPERACIÓN AUTOMÁTICA DE LOS PROCESOS INDUSTRIALES, TENIENDO COMO PREMISAS LA SEGURIDAD, ESTABILIDAD, CONTROLABILIDAD, PRODUCTIVIDAD Y EFICIENCIA DE ESTOS PROCESOS.



SISTEMAS DE CONTROL

**SISTEMAS DE ADQUISICIÓN DE DATOS.
SISTEMAS DE CONTROL SUPERVISORIO.
SISTEMAS DE CONTROL DIGITAL DIRECTO.
SISTEMAS DIGITALES DE CONTROL DISTRIBUIDO.**

SISTEMAS BASADOS EN MICROPROCESADORES

**COMPUTADORAS PERSONALES.
MINICOMPUTADORAS.
INTERFASES HOMBRE-MÁQUINA.
SISTEMAS MÍNIMOS.
BUS INDUSTRIAL.
DISPOSITIVOS INTELIGENTES DE CAMPO.**



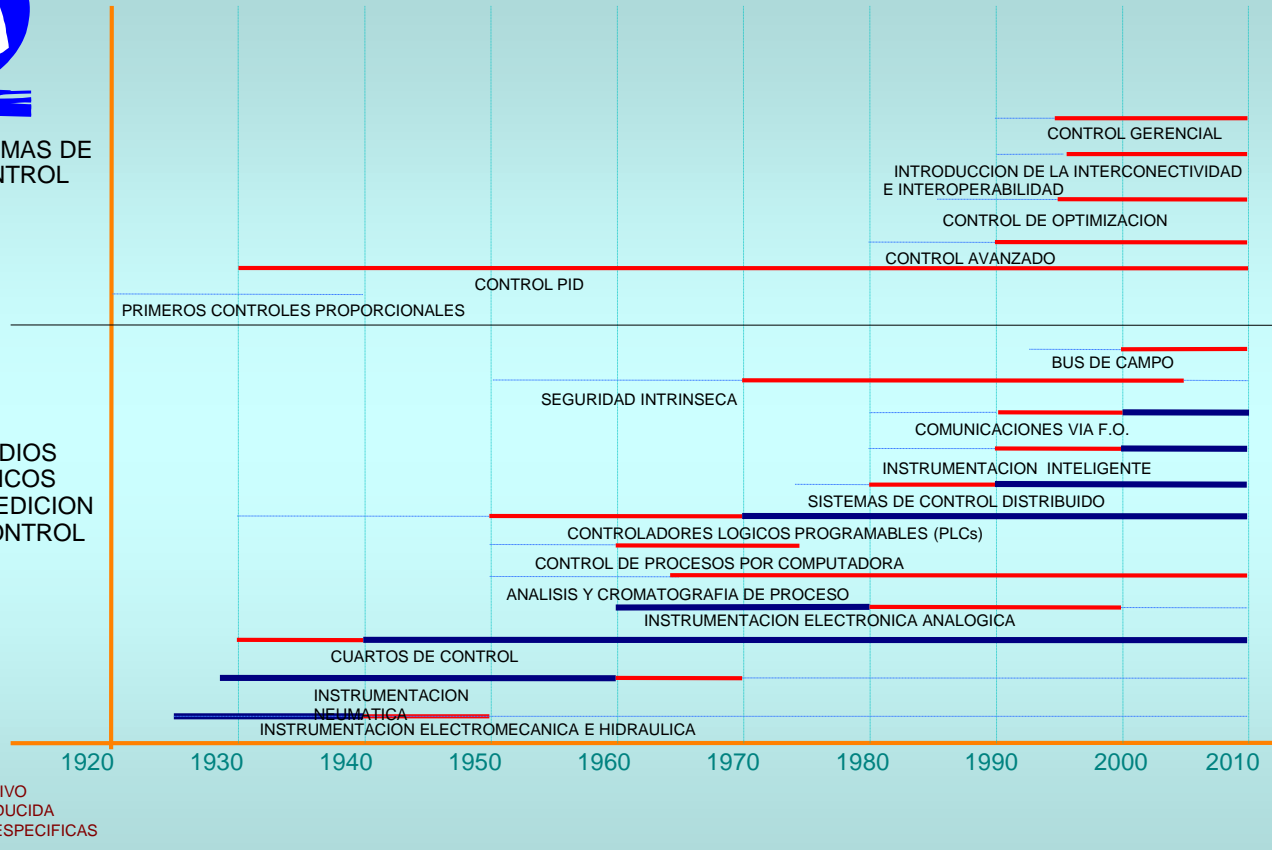
PERIODO	ANTES DE 1920	1930 A 1940	1940 A 1950	1950 A 1960	1960 A 1970	1970 A 1980	1980 A 1990	1990 EN ADEL.
INSTRUMENTACION	MEDICIONES LOCALES ELEMENTOS FINALES DE CONTROL MANUALES	PRIMEROS SERVOMECANISMOS ACTUADORES NEUMATICOS PRIMEROS ANALIZADORES	PRIMEROS INSTRUMENTOS ELECTRONICOS PRIMERAS CELDAS DE PRESION DIFERENCIAL	SE INTRODUCE LA CROMATOGRAFIA DE GASES SE DESARROLLAN NUEVOS PRINCIPIOS DE MEDICION	SE DESARROLLAN NUEVOS CONTROLADORES ELECTRONICOS MAS CAPACES SE DESARROLLAN NUEVOS TIPOS DE VALVULAS DE CONTROL	SE INTRODUCEN LOS MICRO-PROCESADORES EN LA INSTRUMENTACION	SE DESARROLLAN INSTRUMENTOS CON MEJOR EXACTITUD Y CONFIABILIDAD, CON PRECIOS REDUCIDOS	SE DESARROLLAN INSTRUMENTOS "INTELIGENTES" CON FUNCION MULTIPLE
SISTEMAS DE CONTROL	CONTROLES MECANICOS CONTROL DE DOS POSICIONES	PRIMEROS CONTROLES NEUMATICOS PID PRIMEROS CONTROLES LOGICOS PROGRAMABLES	SE DESARROLLA LA TEORIA MODERNA DEL CONTROL AUTOMATICO PRIMERAS TECNICAS DE ENTONAMIENTO	SE DEFINEN LAS BASES DEL CONTROL SUPERVISORIO Y DEL CONTROL DIGITAL DIRECTO	SE DEFINEN LAS BASES DEL CONTROL DIGITAL DISTRIBUIDO SE DESARROLLAN LOS PRIMEROS PLC'S DIGITALES	SE INCREMENTA LA CAPACIDAD DE LOS SISTEMAS DE CONTROL DISTRIBUIDO Y DE LOS PLC'S	SE DESARROLLAN LAS PRIMERAS APLICACIONES DE CONTROL AVANZADO	SE INTRODUCEN LOS CONCEPTOS DE INTEROPERABILIDAD E INTERCONECTIVIDAD
TELEMETRIA	INSTRUMENTACION LOCAL	TRANSMISION NEUMATICA	TRANSMISION ELECTRICA	TRANSMISION 4-20 mA	SE DESARROLLAN LOS PRIMEROS SISTEMAS DE TELEMETRIA	SE DESARROLLAN LOS PRIMEROS SISTEMAS DE FIBRA OPTICA	SE NORMALIZAN LOS PRIMEROS PROTOCOLOS DIGITALES	SE INTRODUCEN LOS CANALES DE CAMPO

Evolución de los Sistemas de Instrumentación y Control Industrial

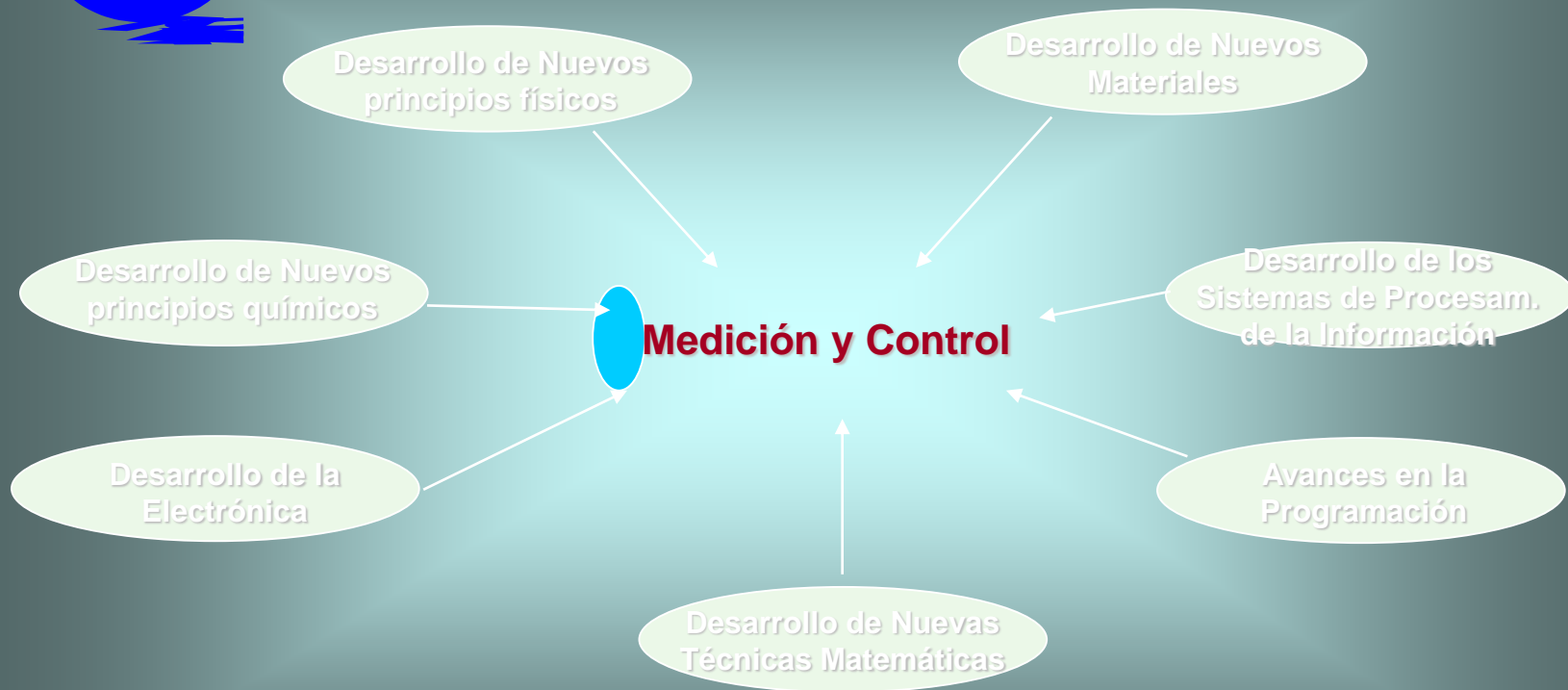


SISTEMAS DE CONTROL

MEDIOS FISICOS DE MEDICION Y CONTROL



Evolución de los Sistemas de Instrumentación y Control Industrial



PRINCIPALES PROMOTORES DE LOS AVANCES EN LA INSTRUMENTACION Y EL CONTROL INDUSTRIAL

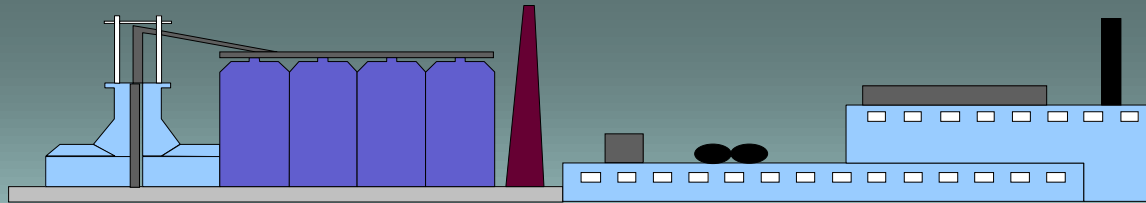


AUMENTO EN
LOS INSUMOS Y EN LA
MANO DE OBRA

TLC

NUEVAS
REGULACIONES
AMBIENTALES

MAYORES
REQUERIMIENTOS EN
PRODUCTIVIDAD, CALIDAD
Y EFICIENCIA

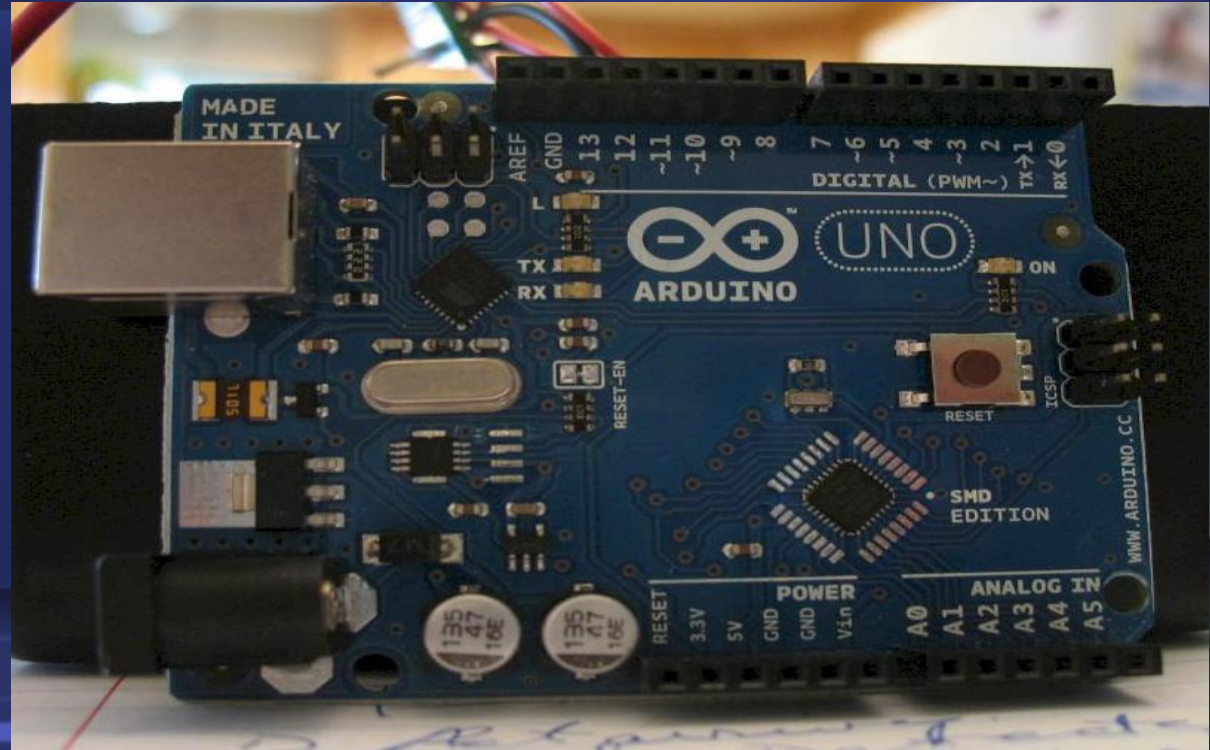


Presiones sobre la Industria

Arduino



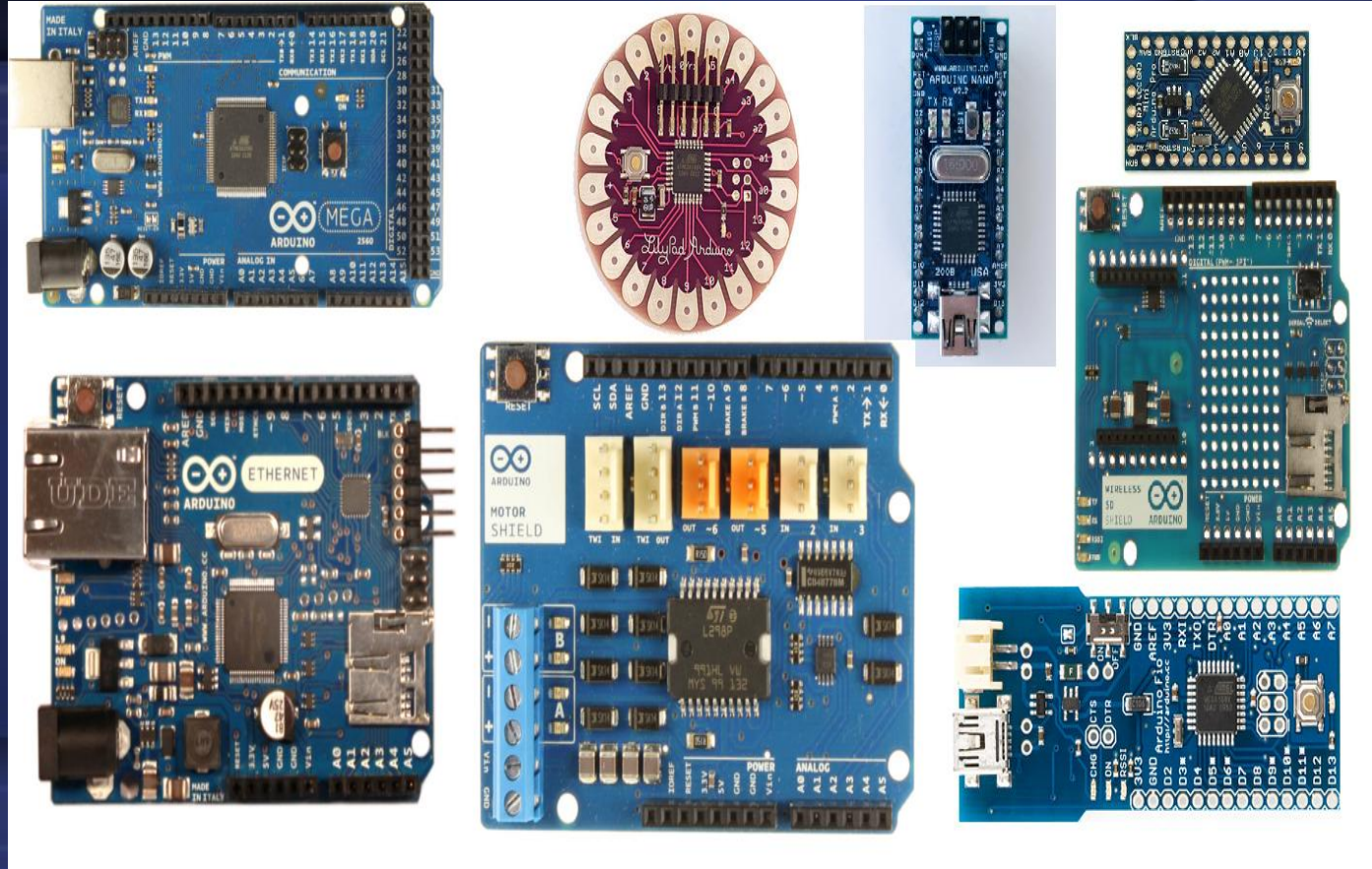
FCEE
Facultad de Ciencias
de la Electrónica



Dr. Jaime Cid Monjaraz



Arduino

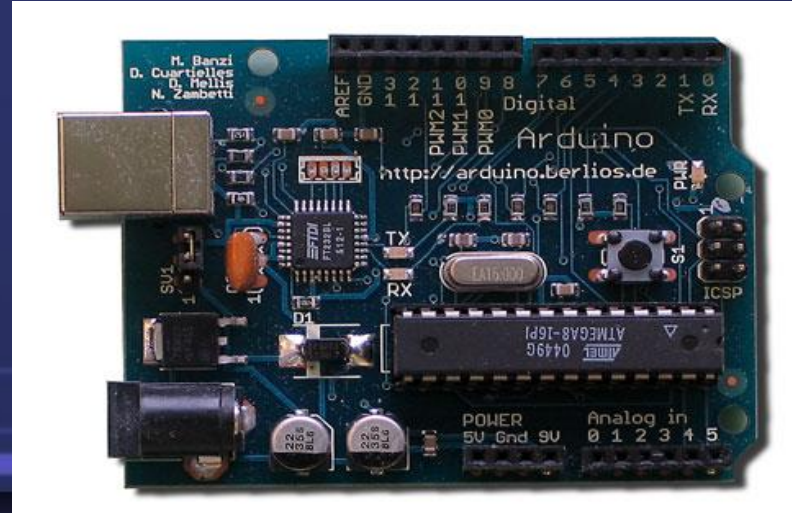
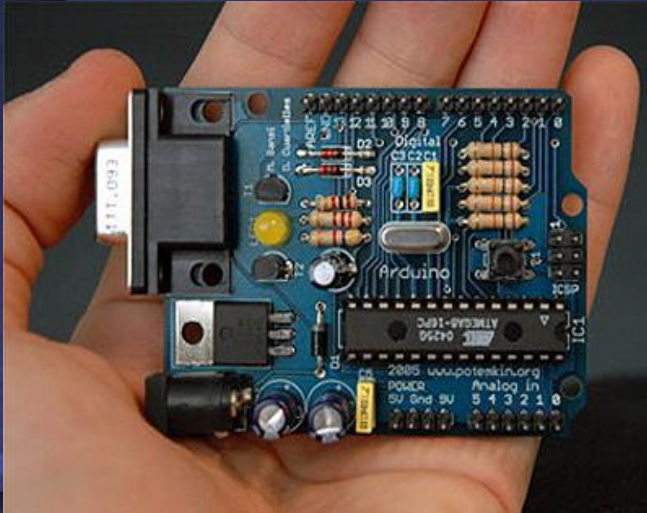


Dr. Jaime Cid Monjaraz



Arduino es una plataforma

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.



- Barato: Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- Entorno de programación simple y claro: El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores AT-MEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.

Otros modelos oficiales:

SmartProjects (Italia)



Nano



Duemilanove

Sparkfun (EEUU)



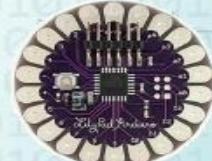
Diecimila



Pro



Pro Mini



Lilypad



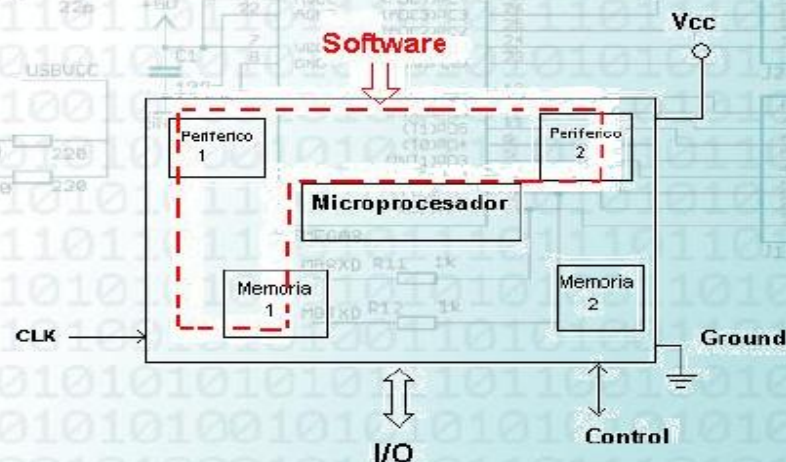
Características

Microcontrolador	ATmega328
Voltaje de operación	5V
Voltaje de entrada (recomendada)	7-12V
Voltaje de entrada (límites)	6-20V
Pines Digitales E/S	14 (de los cuales 6 proveen salidas PWM)
Pines de entradas Analógicas	6
Corriente DC por pin E/S	40 mA
Corriente DC para pin 3.3V	50 mA
Memoria Flash	32 KB de los cuales 0.5 KB son usados para bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Frecuencia del reloj	16 MHz



¿Qué es un microcontrolador (μC)?

- Circuito integrado con las 3 unidades funcionales de una computadora:
 - CPU (Unidad central de procesamiento)
 - Memoria
 - Periféricos de I/O



Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 μ A
 - Power-save Mode: 0.75 μ A (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P



Figure 2-1. Block Diagram

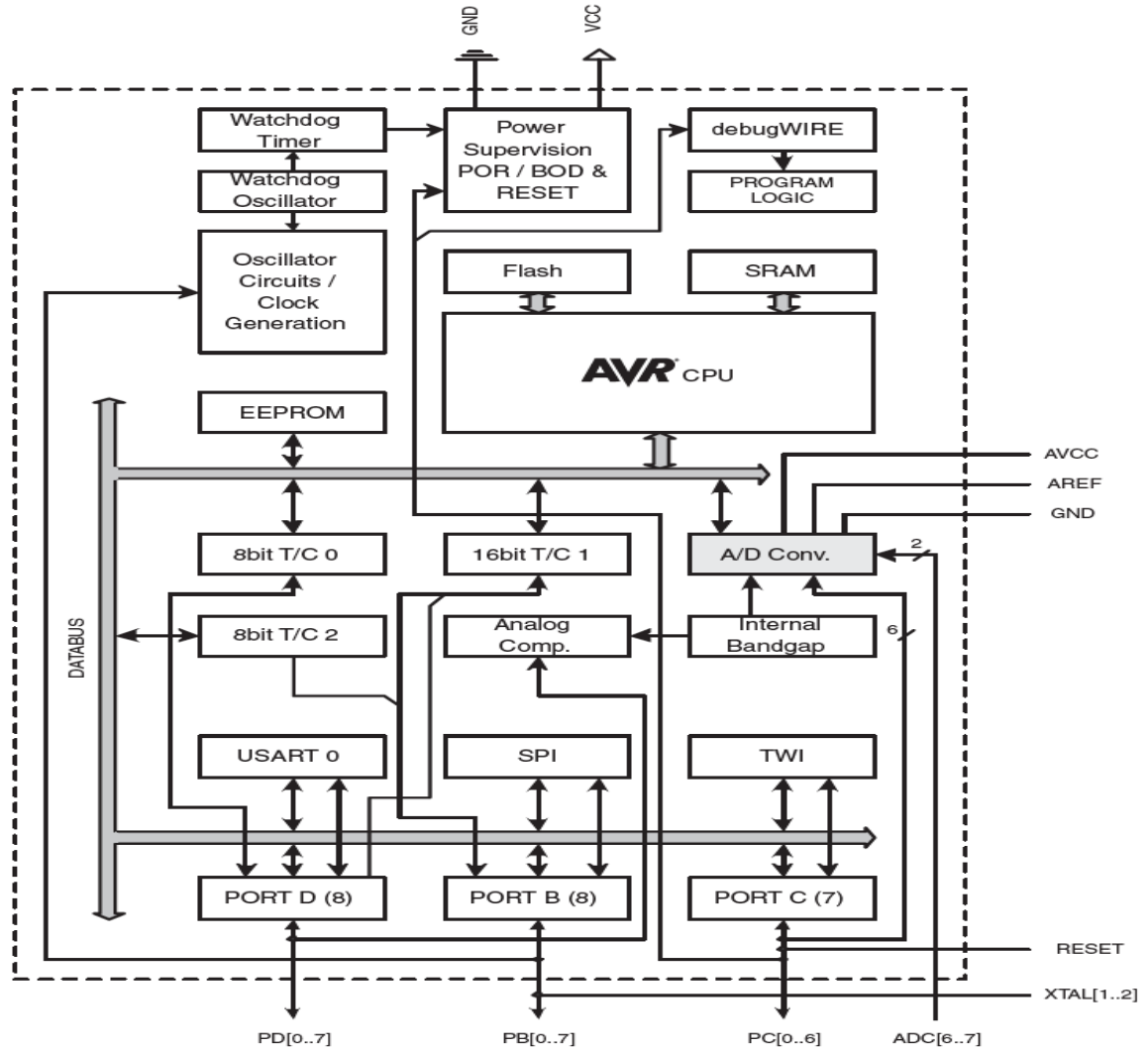




Figure 6-1. Block Diagram of the AVR Architecture

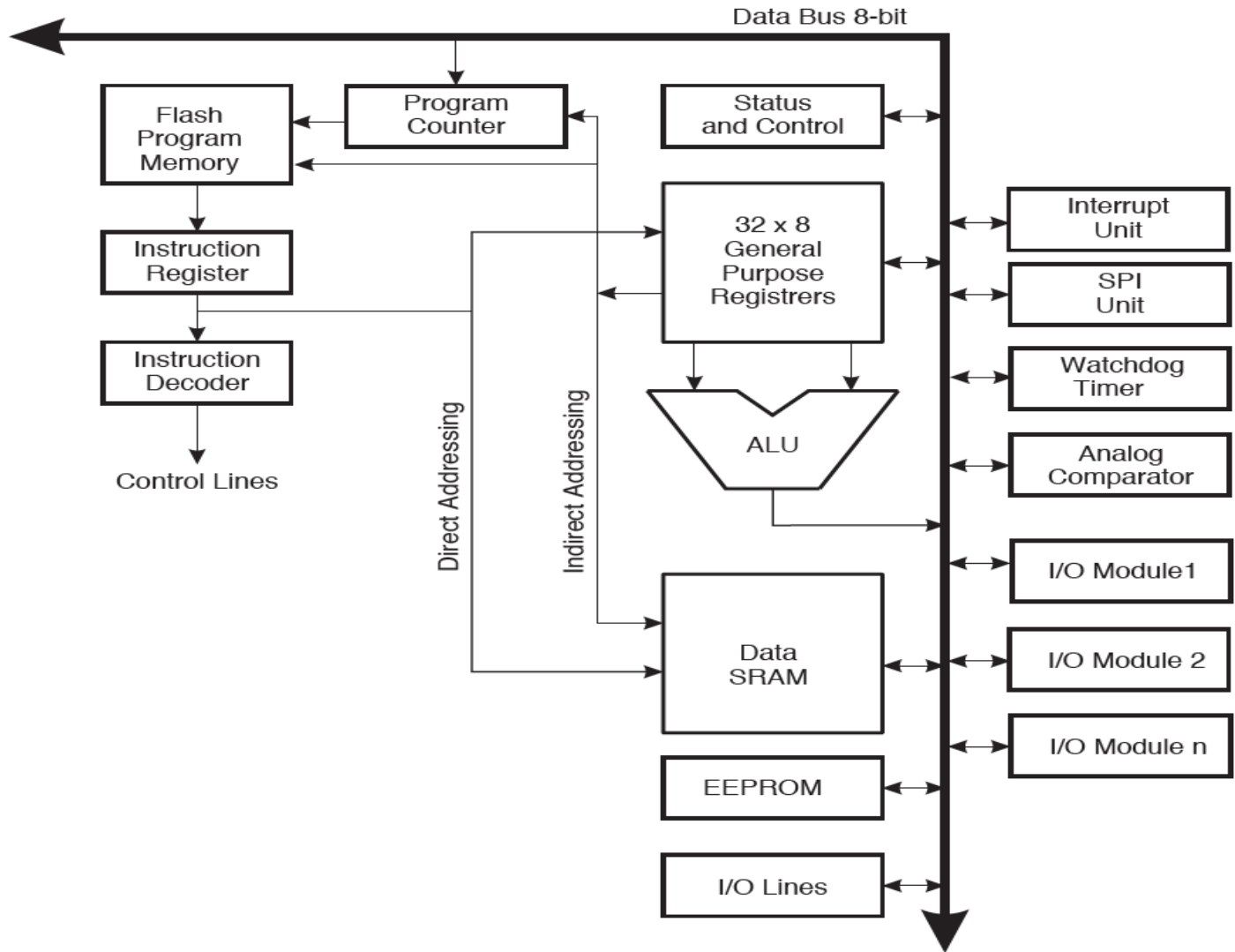
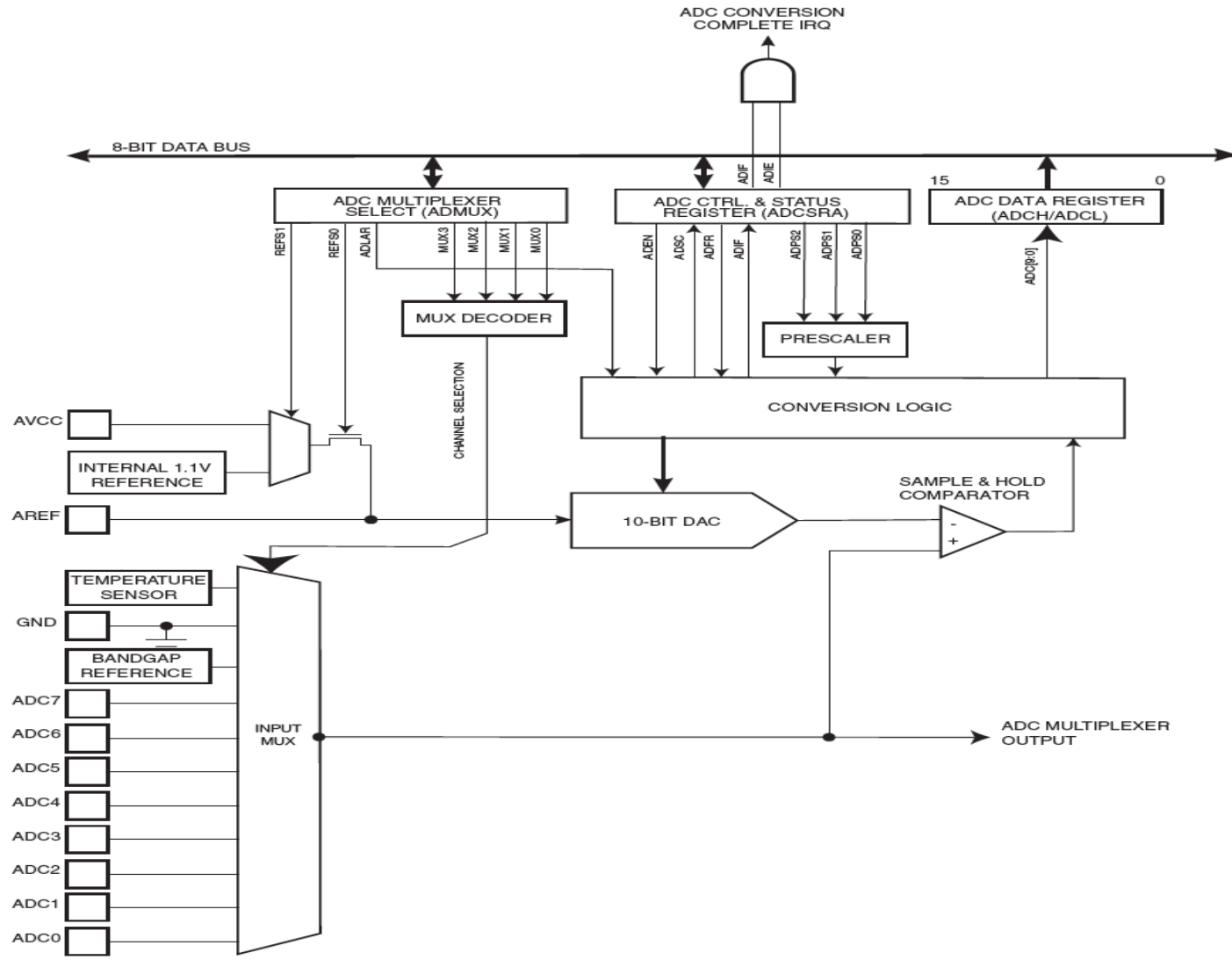


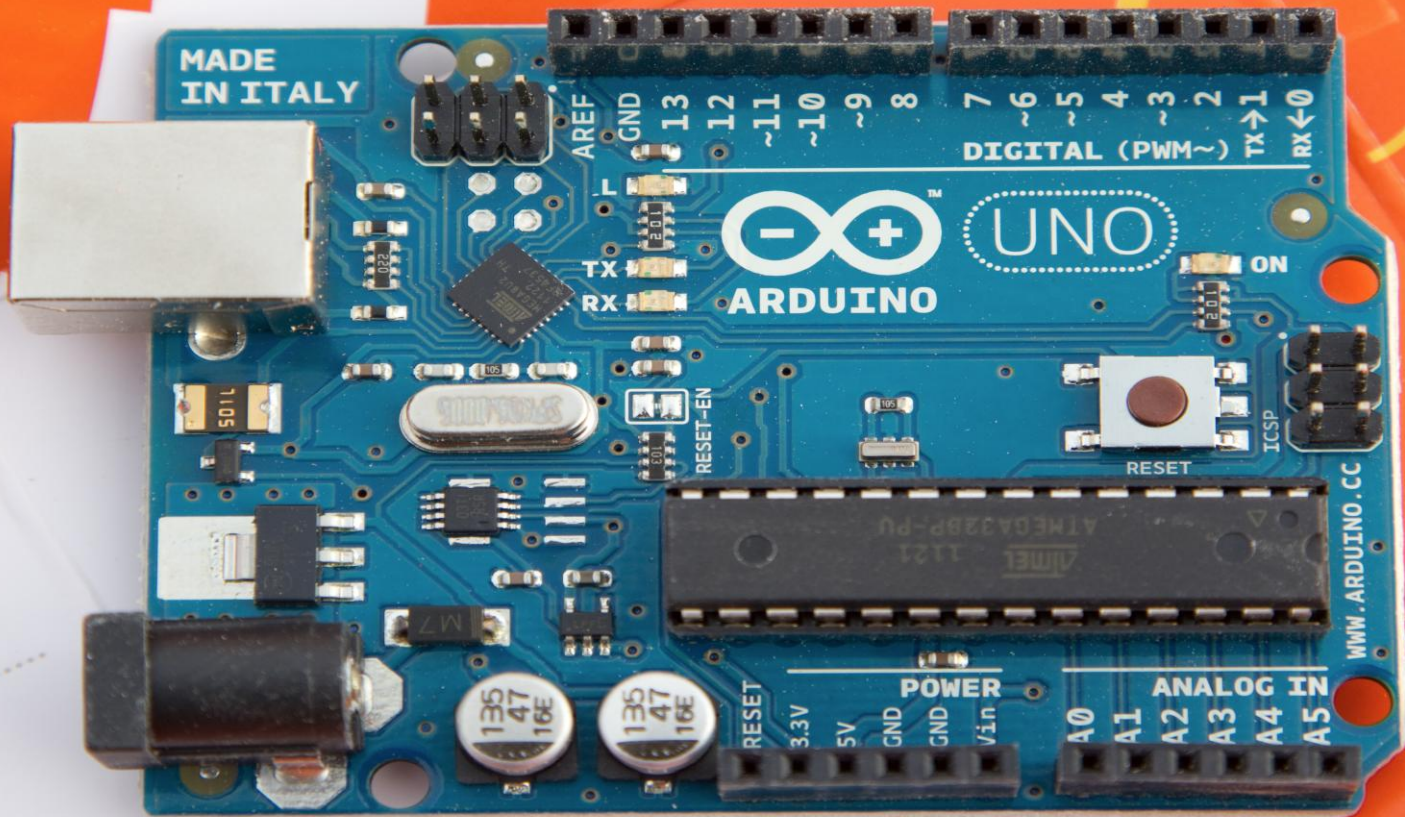


Figure 23-1. Analog to Digital Converter Block Schematic Operation,



Proyecto Arduino

- Plataforma de hardware y software libre (CC).
- Formado por:
 - Microcontrolador Atmel (AVR)
 - Circuito que facilita el uso:
 - USB
 - Pines externos.
 - Cristal cuarzo 16MHz
- <http://www.arduino.cc/es/>



MADE
IN ITALY

ARDUINO

UNO

WWW.ARDUINO.CC

AREF GND 13 12 11 10 9 8 7 6 5 4 3 2 1
DIGITAL (PWM ~) TX → RX ←

RESET 3.3V 5V GND GND Vin
POWER

A0 A1 A2 A3 A4 A5
ANALOG IN

ATMEGA328P-PU

ATMEGA328P-PU 1121

5011

136 47 16E

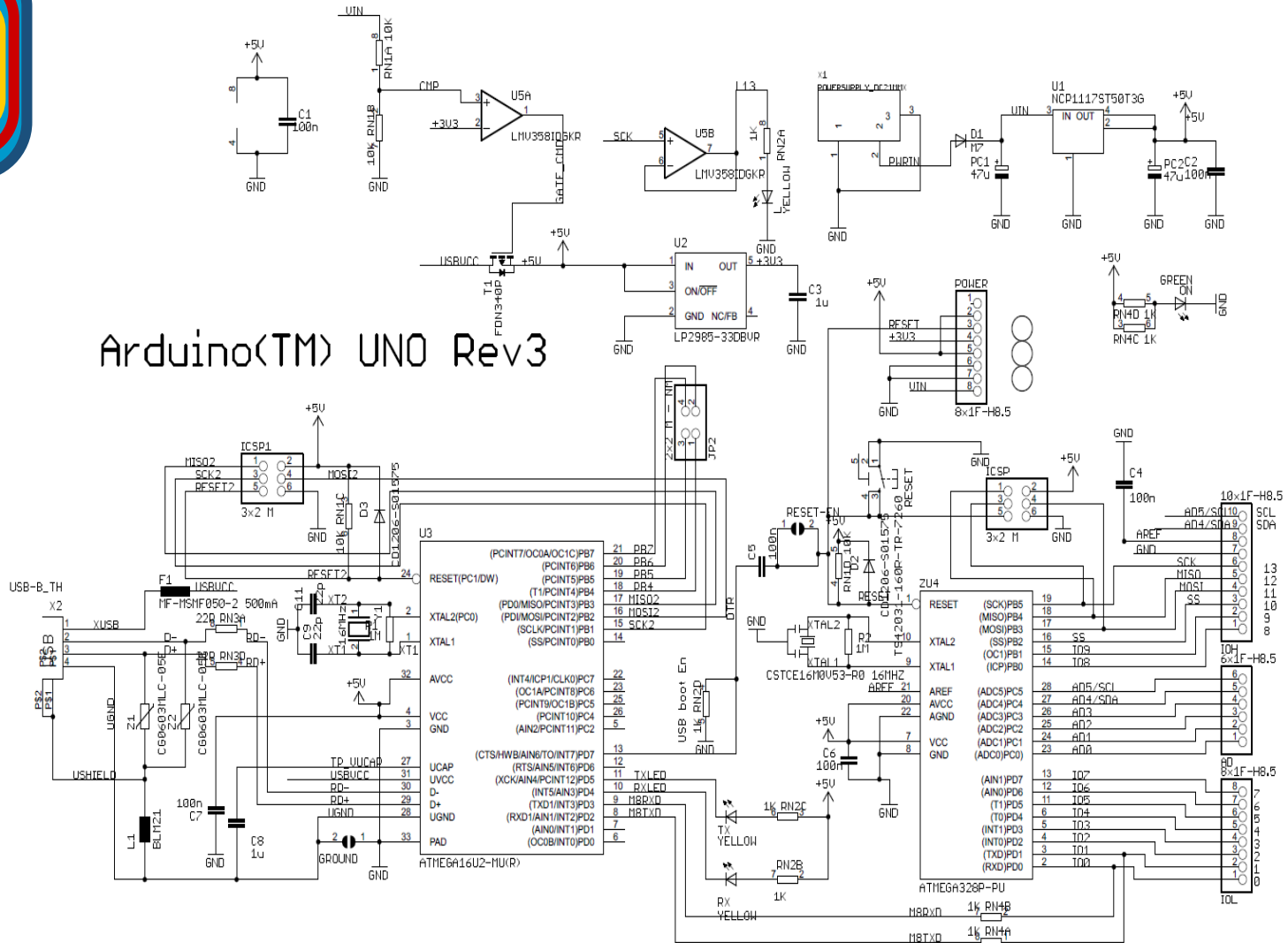
136 47 16E

ON

RESET

ICSP

Arduino(TM) UNO Rev3



Arduino UNO (ONE)

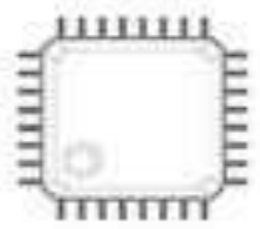
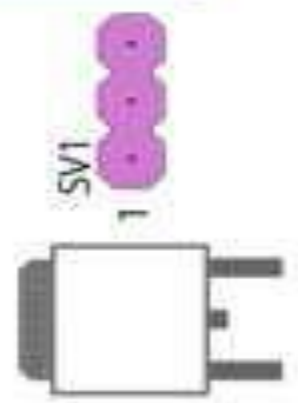




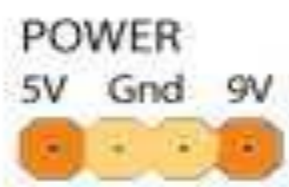
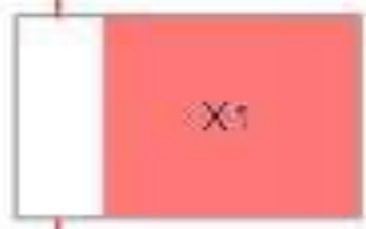
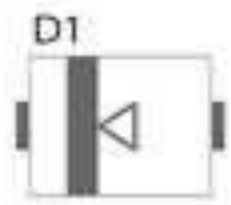
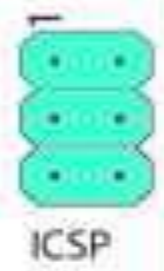
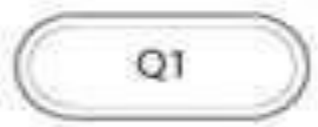
PWM2
PWM1
PWM0

Arduino

PWR
E



<http://arduino.berlios.de>



Software de Arduino

- Escrito en Java y basado en Processing, avr-gcc y otros programas de código abierto.
- Librerías de clases:
- IDE:
 - **Multiplataforma**
 - Se programa en **C/C++**
- Descarga: <http://arduino.cc/es/Main/Software>

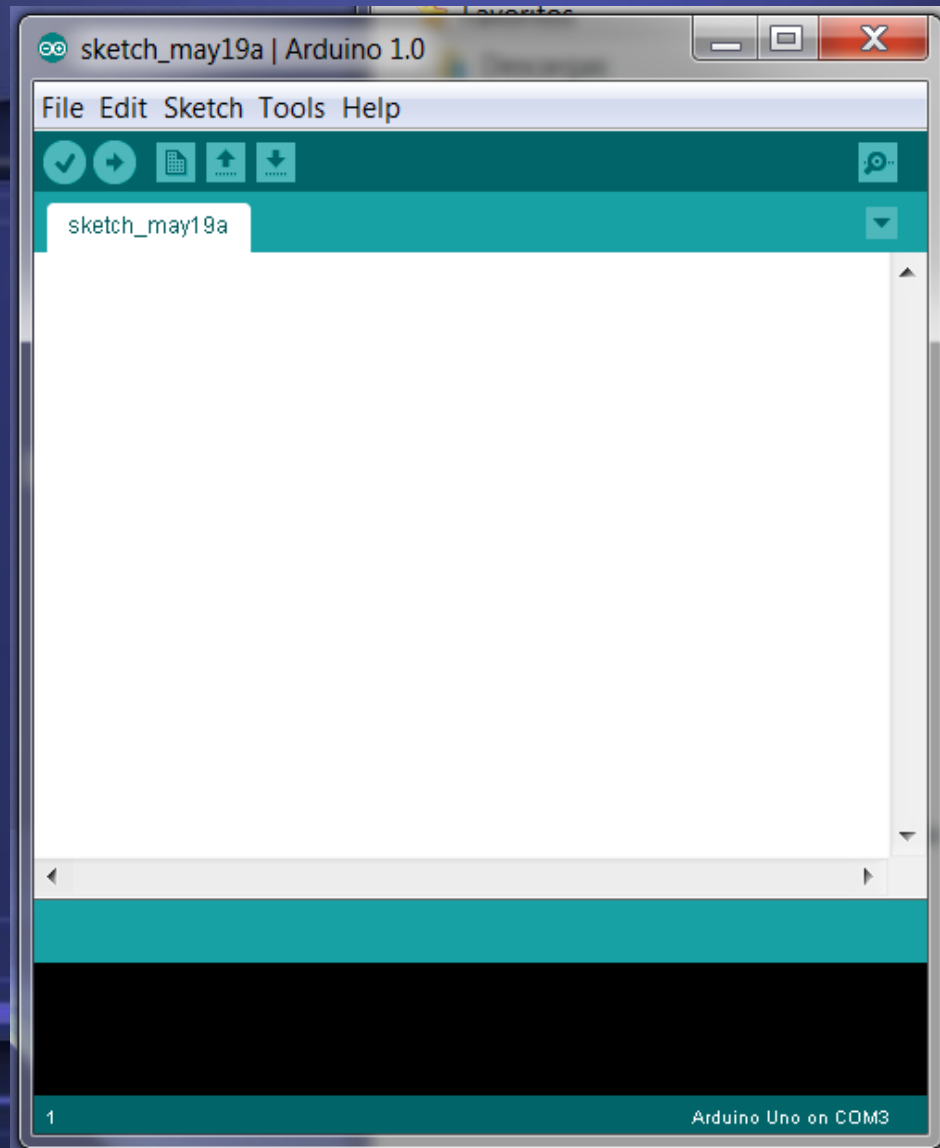
Antes de comenzar

- Descarga el programa de www.arduino.cc
- Descomprime la carpeta
- Conecta el Arduino
- Instala los drivers FTDI
- Anota el puerto COM del Arduino

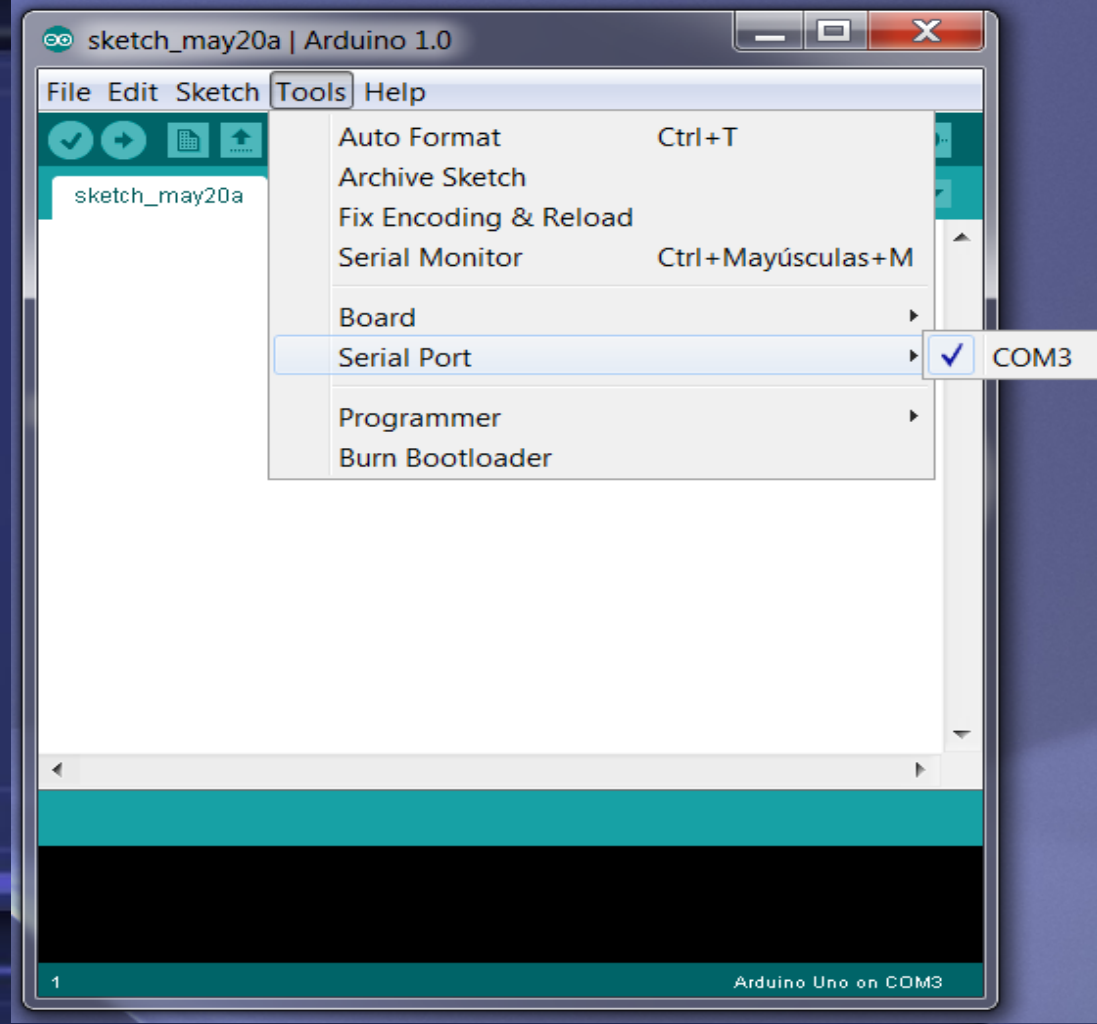
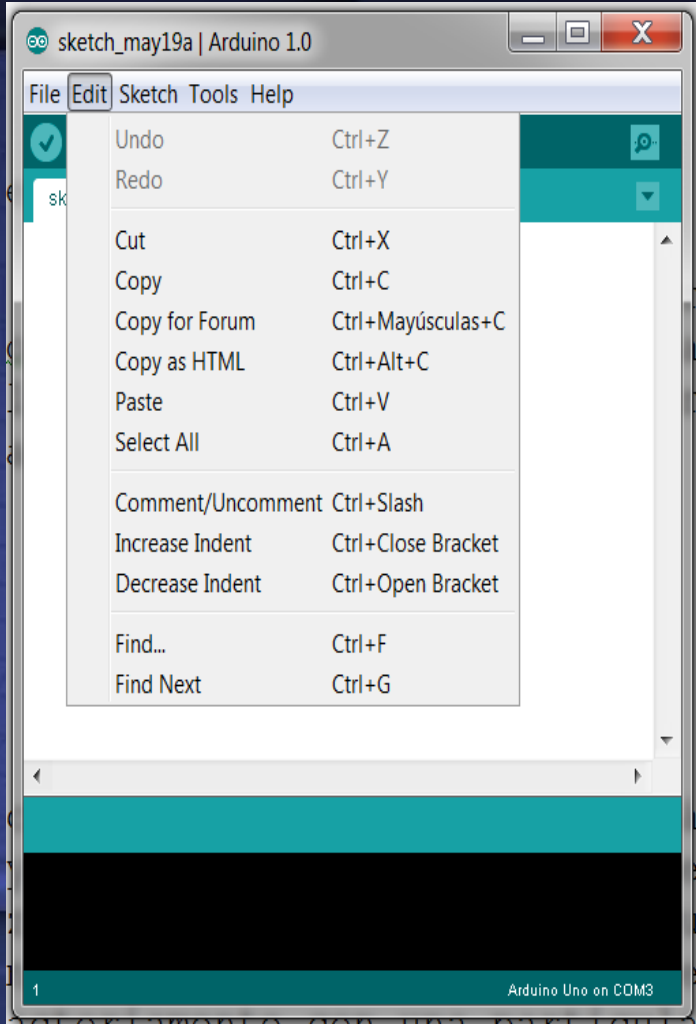
SOFTWARE ENTORNO DE DESARROLLO

Para programar la placa es necesario descargarse de la página web de Arduino (<http://www.arduino.cc/en/Main/Software>) el entorno de desarrollo (IDE).

En caso disponer de una placa USB es necesario instalar los drivers.



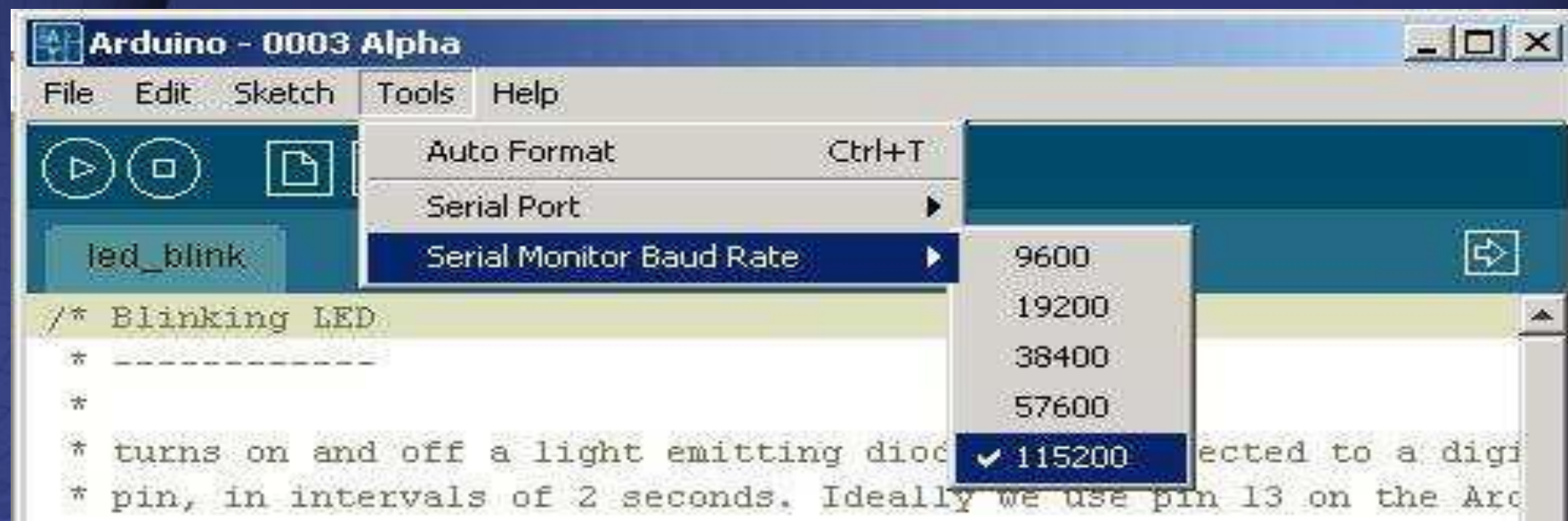
Configuración del puerto serie



En Windows, si desconocemos el puerto al que está conectado nuestra placa podemos descubrirlo a través del “Administrador de dispositivos”



También debemos configurar la velocidad a la que la placa y el PC se comunican. Esto lo hacemos desde el menú “Serial Monitor Baud Rate”. El valor por defecto es de 115200 baudios



File Edit Sketch **Tools** Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Serial Monitor Ctrl+Mayúsculas+M

Board

Serial Port

Programmer

Burn Bootloader

- Arduino Uno
- Arduino Duemilanove w/ ATmega328
- Arduino Diecimila or Duemilanove w/ ATmega168
- Arduino Nano w/ ATmega328
- Arduino Nano w/ ATmega168
- Arduino Mega 2560 or Mega ADK
- Arduino Mega (ATmega1280)
- Arduino Mini w/ ATmega328
- Arduino Mini w/ ATmega168
- Arduino Ethernet
- Arduino Fio
- Arduino BT w/ ATmega328
- Arduino BT w/ ATmega168
- LilyPad Arduino w/ ATmega328
- LilyPad Arduino w/ ATmega168
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
- Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
- Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
- Arduino NG or older w/ ATmega168

New	Ctrl+N
Open...	Ctrl+O
Sketchbook	
Examples	
Close	Ctrl+W
Save	Ctrl+S
Save As...	Ctrl+Mayúsculas+S
Upload	Ctrl+U
Upload Using Programmer	Ctrl+Mayúsculas+U
Page Setup	Ctrl+Mayúsculas+P
Print	Ctrl+P
Preferences	Ctrl+Comma
Quit	Ctrl+Q

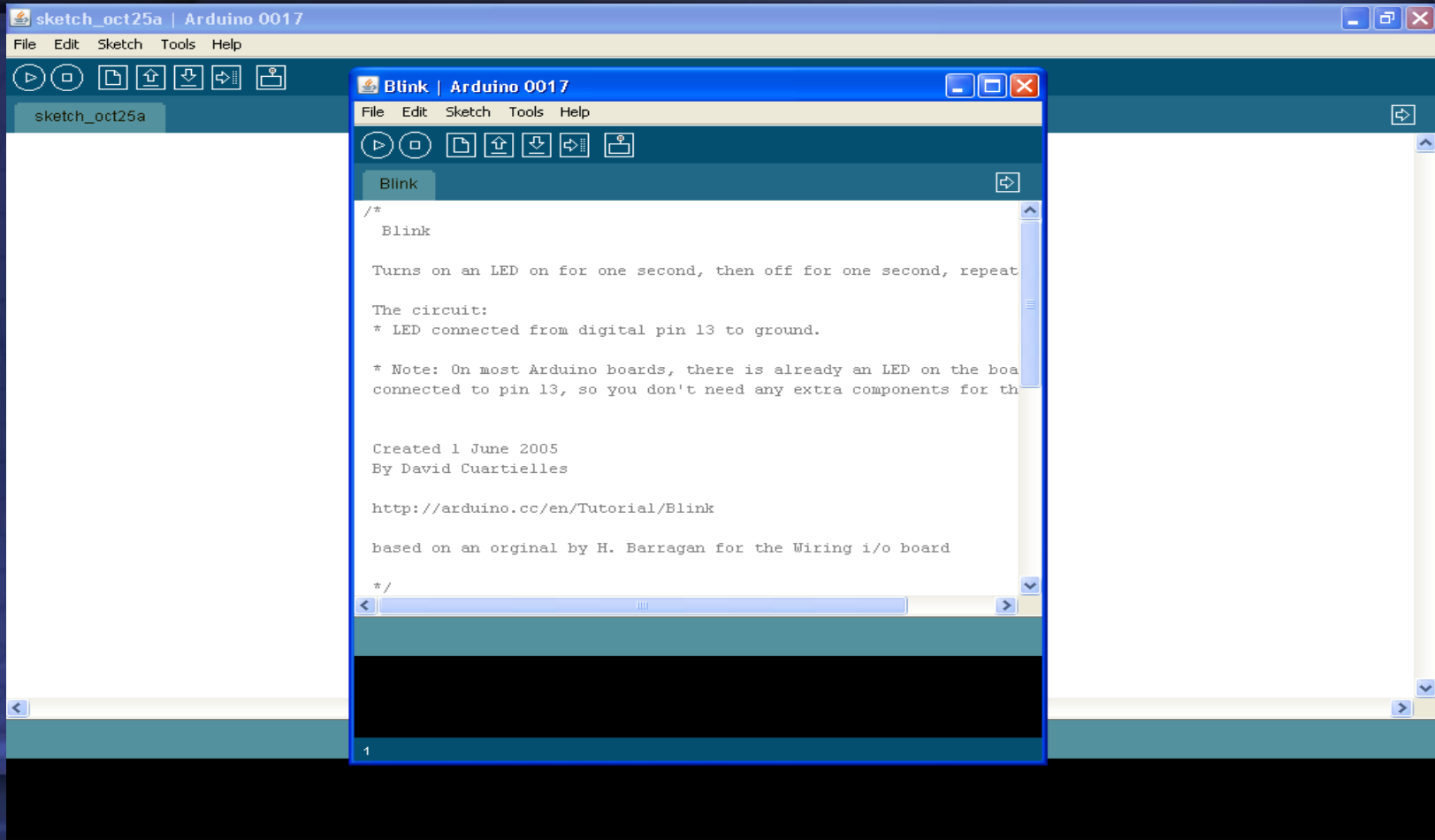
```
delay(1000); // wait for a second
```

```
}
```

```
<
```

1

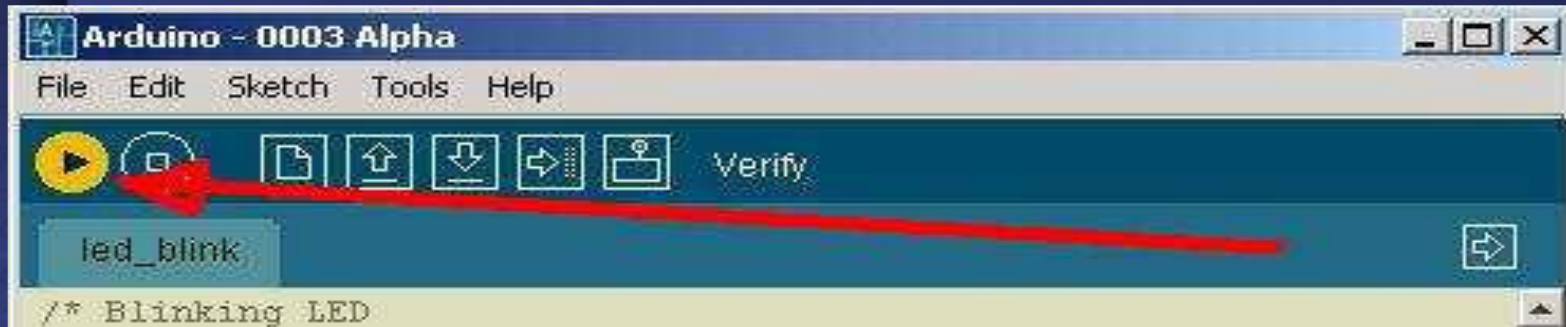
- 1.Basics
 - 2.Digital
 - 3.Analog
 - 4.Communication
 - 5.Control
 - 6.Sensors
 - 7.Display
 - 8.Strings
 - ArduinoISP
 - EEPROM
 - Ethernet
 - Firmata
 - LiquidCrystal
 - SD
 - Servo
 - SoftwareSerial
 - SPI
 - Stepper
 - Wire
- AnalogReadSerial
 - BareMinimum
 - Blink
 - DigitalReadSerial
 - Fade



SUBIENDO EL PROGRAMA DE EJEMPLO A LA PLACA ARDUINO

El ejemplo “led_blink” lo único que hace es parpadear un LED que esté colocado en el pin número 13 de la placa. Vamos a ver qué hay que hacer para subir el programa a la placa Arduino.

Primero comprobamos que el código fuente es el correcto. Para ello pulsamos el botón de verificación de código que tiene forma de triángulo inclinado 90 grados.



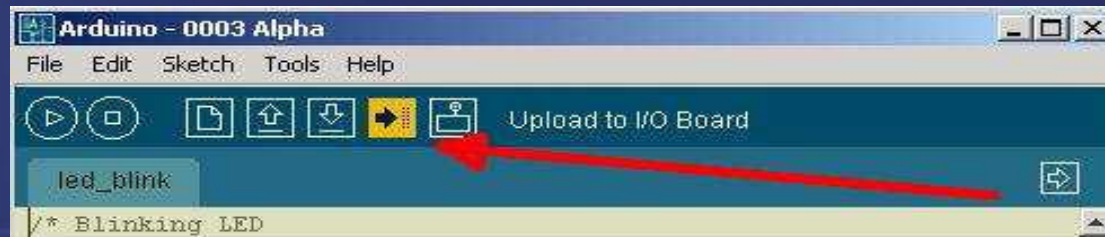
Si todo va bien deberá aparecer un mensaje en la parte inferior de la interfaz indicando “Done compiling”.

```
digitalWrite(ledPin, LOW); // sets the LED off
delay(1000); // waits for a second
}
```

Done compiling.

1

Una vez que el código ha sido verificado procederemos a cargarlo en la placa. Para ello tenemos que pulsar el botón de reset de la placa e inmediatamente después pulsar el botón que comienza la carga.

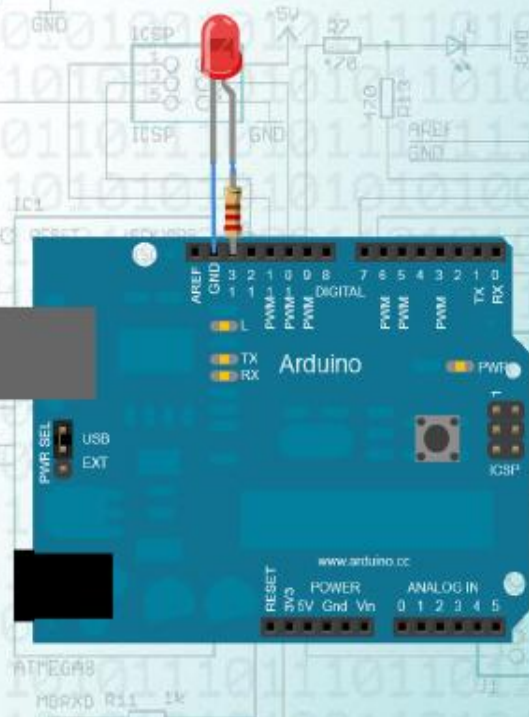


"Hola Mundo"

```
void setup() {  
  pinMode(13, OUTPUT); //Pin digital 13 como output.  
}  
  
void loop() {  
  digitalWrite(13, HIGH); //Enciende el LED  
  delay(1000); //Espera 1000 milisegundos  
  digitalWrite(13, LOW); //Apaga el LED  
  delay(1000); //Espera 1000 milisegundos  
}
```


“Hola Mundo”

- Parpadeo de un LED
- Conectar el polo positivo del LED (el más largo) al pin 13 y el otro al GND.



Código del primer programa

```
int ledPin = 13
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  digitalWrite(ledPin, HIGH)
  delay(1000)
  digitalWrite(ledPin, LOW)
  delay(1000)
}
```


Analog In - Entradas análogas

Pins análogos son entradas análogas. Reciben tensiones entre 5V y 0 voltios. Los pines análogos, al contrario de los digitales, no necesitan ser declarados como modo INPUT o OUTPUT .

Conversión análogo digital o ADC (analog to digital converter)

convertir tensiones de 0 a 5 voltios en números enteros que van del 0 al 1023. En otras palabras, representa la información en números de 10 bits.

Son AI: resistencias variables : potenciómetro - fotocélula - FSR

Comandos básicos

analogRead(pin), Lee o captura el valor de entrada del especificado pin análogo. Placa Arduino realiza una conversión análoga a digital de 10 bits. Esto quiere decir que mapeará los valores de voltaje de entrada, entre 0 y 5 voltios, a valores enteros comprendidos entre 0 y 1023.

Analog In - Comunicación Serial

Esto nos permite comunicarnos con un computador e intercambiar datos o simplemente monitorear que pasa con el sketch que esta corriendo en Arduino.

Podemos visualizar los datos procedentes de la tarjeta usando **Monitorización del Puerto Serie** (último botón a la derecha).

A veces nos interesa poder mandar datos de los sensores hacia el computador o incluso poder mandar comandos desde el PC a Arduino. Por ejemplo, si queremos visualizar, la lectura de un potenciómetro. Si la comunicación en serie esta activada, no se podrán usar los pines 0 y 1 como entrada/salida digital.

Es recomendable dejar tiempos de espera entre los envíos de datos para ambos sentidos (uso por ejemplo de un `delay(10)`) ,ya que se puede saturar o colapsar el puerto.

Analog In - Comunicación Serial

- Comandos básicos:

Serial.begin

configura el puerto serie a una velocidad determinada. Se expresa en bits por segundo. Va en el **setup()**.

Serial.print()

Descompone un número obtenido de un sensor, por ejemplo, en símbolos ASCII y los lanza uno a uno por el puerto serie en modo de caracteres ASCII. Por ejemplo, el número 100 se representará con la secuencia de números ASCII: 49, 48, 48.

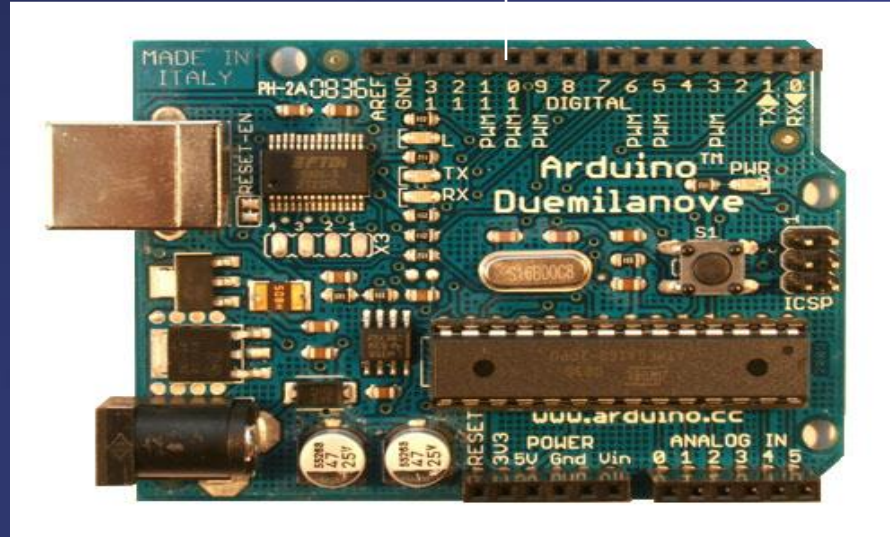
Serial.println(): lanza el valor 13, que quiere decir retorno de carro y el valor 10 que quiere decir fin o salto de línea por el puerto serie.

Serial.print(dato,BYTE): lanza el valor dato por el puerto serie, en modo Byte o Binario.

Analog Out - PWM

* PWM: 3, 5, 6, 9, 10 y 11.

PWM outputs de 8-bits utilizando función `analogWrite()`.



Para que me sirve?

Manejo de intensidad: luz, T, sonido, etc.

Analog Out - PWM

PWM (Pulse-width modulation) o serial de modulación por ancho de pulso.

Ancho de pulso, representa al ancho (en tiempo) del pulso con una modulación o cantidad de pulsos (estado on/off) por segundo.

El periodo es medido en segundos y la frecuencia en hertz

La señal PWM se utiliza como técnica para controlar circuitos analógicos, comúnmente usadas para el control de motores DC (si decrementas la frecuencia, la inercia del motor es mas pequeña y el motor se mueve mas lentamente), ajustar la intensidad de brillo de un LED, etc.

En Arduino la señal de salida PWM (pines 3, 5, 6, 9, 10 y 11) es una señal de frecuencia constante (30769 Hz) y que solo nos permite cambiar el "duty cycle" o el tiempo que el pulso esta activo (on) o inactivo (off), utilizando la función `analogWrite()`.

Analog Out - PWM

Con el siguiente código y con solo realizar modificaciones en los intervalos de tiempo que el pin seleccionado tenga valor HIGH o LOW, a través de la función `digitalWrite ()`, generamos la señal PWM.

```
// señal PWM //
int digPin = 10; // pin digital 10
void setup() {
  pinMode(digPin, OUTPUT); // pin en modo salida
}
void loop() {

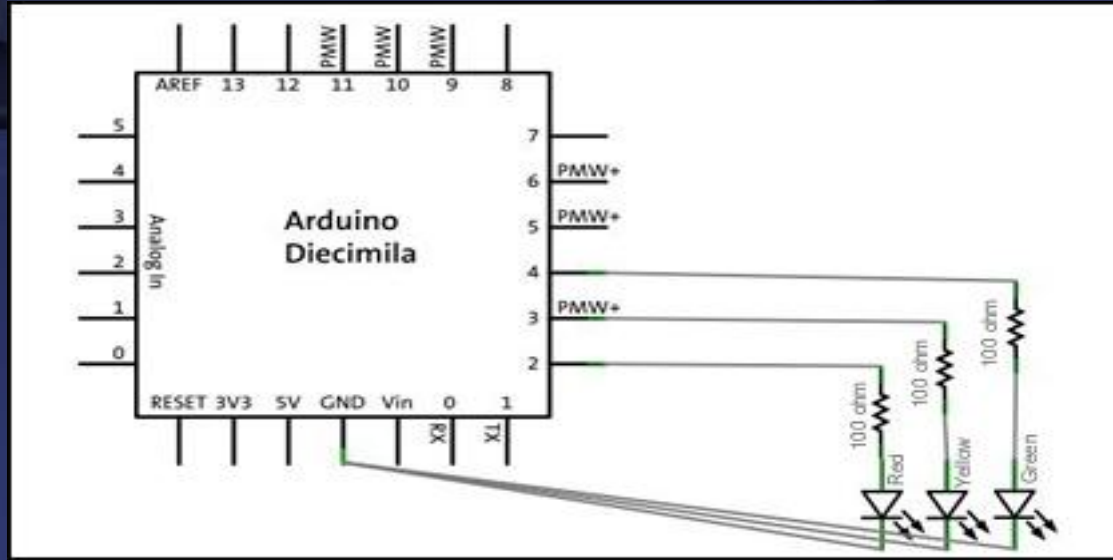
  digitalWrite(digPin, HIGH); // asigna el valor HIGH al pin
  delay(500); // espera medio segundo
  digitalWrite(digPin, LOW); // asigna el valor LOW al pin
  delay(500); // espera medio segundo

}
```

Analog Out - PWM

El programa pone el pin a HIGH una vez por segundo, la frecuencia que se genera en dicho pin es de 1 pulso por segundo o 1 Hertz de pulso de frecuencia (periodo de 1 segundo) . Cambiando la temporización del programa, podremos cambiar la frecuencia de la señal. Por ejemplo, si cambiamos las dos líneas con delay(500) a delay(250), multiplicaremos la frecuencia por dos, de forma que estamos enviando el doble de la cantidad de pulsos por segundo que antes.,o sea que el LED ya no blinkea, sino que esta brillando al 50% de su brillo normal.

Ahora cambia los números del LED en 1/4 del tiempo en que esta off. Has correr el sketch y veras que el brillo es de 25%. Esta técnica se llama **pulse width modulation (PWM)**, la idea es que el LED blinkea tan rápido que no te das cuenta, pero cambias su rango de brillo entre el tiempo on y el tiempo off. También se aplica a motores y es controlada con la instrucción `analogWrite()`



P1. Manejo de una salida digital. Intermitente (*BLINK).

P2. Secuencia Básica de 3 LEDs. (SEMAFORO)

<http://www.youtube.com/watch?v=I3U9I7s5IMM>

O

(*semaforo 2)

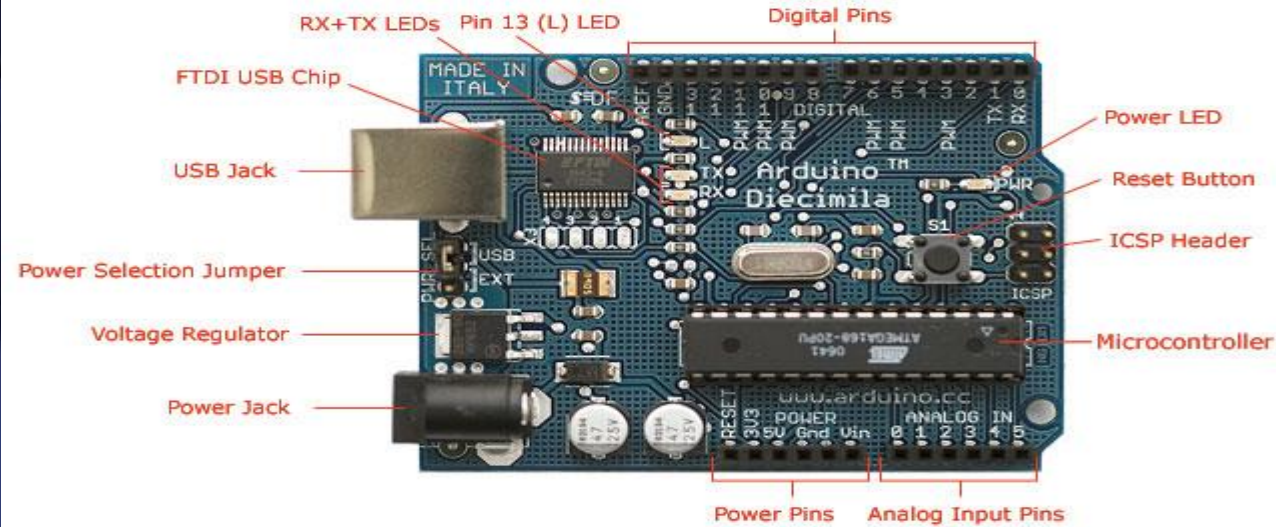
http://www.youtube.com/watch?v=wzqun5cktXk&feature=player_embedded

Salidas/Entradas Digitales (DI - DO)

Pines digitales son salidas o entradas que reciben niveles altos (5V) o bajos (0V) de tensión y que son interpretados como un 1 o un 0 respectivamente.

Para controlar estas salida /entradas, se utilizan los siguientes comandos:

- `pinMode(pin, mode)` Configura el pin como entrada o salida pin corresponde al número del pin y mode puede ser INPUT o OUTPUT
- `digitalWrite(pin, value)` Escribe un 0 o un 1 (0 o 5V) en el pin especificado
- `int digitalRead(pin)` Lee el valor de un pin digital.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Nota: Los pines digitales 0 y 1 pueden ser utilizados como salidas o entradas siempre que se estén utilizando para la comunicación serial

Salidas/Entradas Digitales (DI - DO)

Pins digitales (pines análogos no necesitan ser declarados como modo INPUT o OUTPUT) son salidas o entradas digitales, esto es niveles altos (5V) o bajos (0V) de tensión que pasan por cada uno de los pines excepto los **pines 0 (TX) y 1 (RX)** que se emplean para la comunicación en serie o comunicación de Arduino con otros dispositivos.

Para controlar estas salida /entradas, se utilizan ciertos comandos:

pinMode

digitalWrite

delay

digitalWrite(LED, LOW)

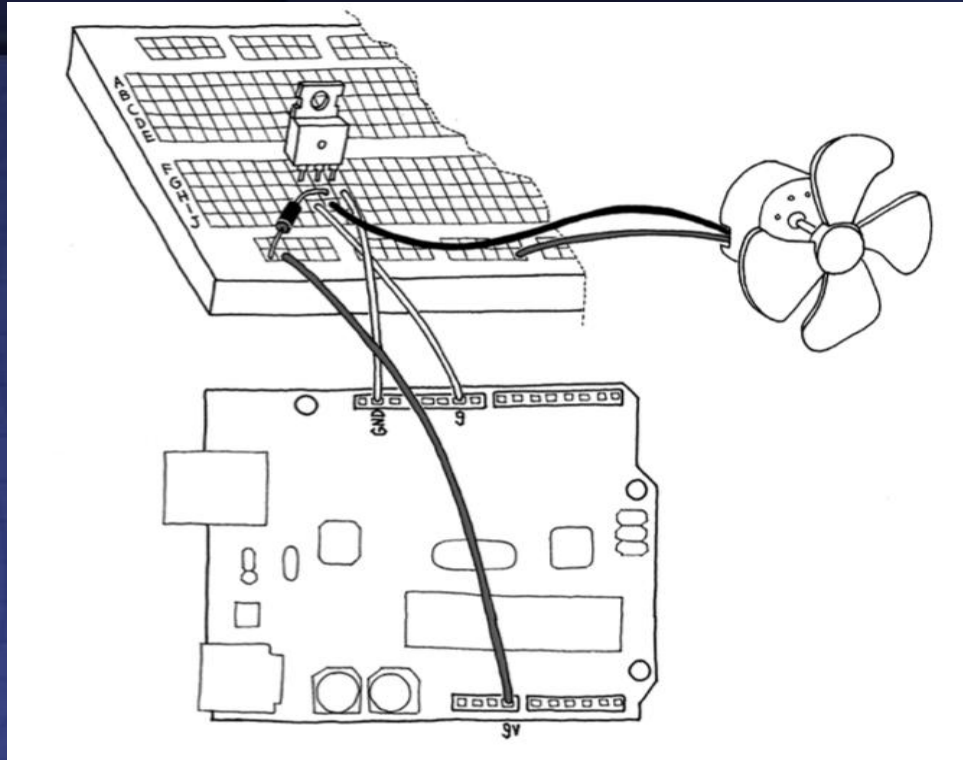
digitalWrite(LED, HIGH)

Ejemplo en código

```
int ledPin = 13; // LED conectado al pin 13
int inPin = 7;   // pulsador conectado al pin 7
int val = 0;    // Variable para almacenar el valor leído

void setup()
{
  pinMode(ledPin, OUTPUT); // configura el pin 13 como salida
  pinMode(inPin, INPUT);  // configure el pin 7 como entrada
}

void loop()
{
  val = digitalRead(inPin); // Lee el valor del pin 7
  digitalWrite(ledPin, val); // Enciende el LED si el pulsador está
                             // presionado
}
```



Arduino conectado a un MOSFET (IRF250)

```
Int MotorPin = 9;
```

```
void setup()
```

```
{
```

```
  // configura el pin 9  
  como salida
```

```
  pinMode(MotorPin,  
  OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
  // Enciende el Motor
```

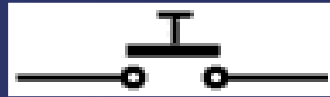
```
  digitalWrite(ledPin, val);
```

```
}
```

Manejo de una entrada digital, lectura de un pulsador

Descripción del ejercicio: Conectaremos un pulsador a la entrada digital 2 para luego leer si este se encuentra presionado utilizando el comando `digitalRead (inPin)`

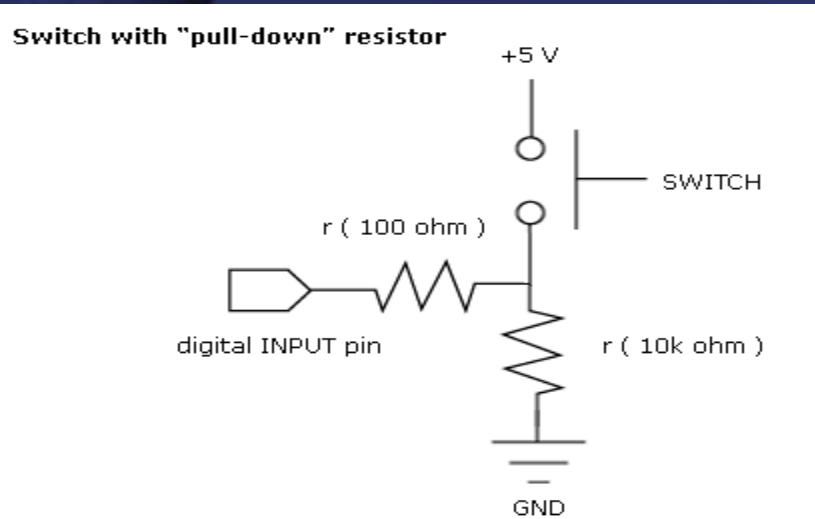
El pulsador es un componente que conecta dos puntos de un circuito cuando es presionado.



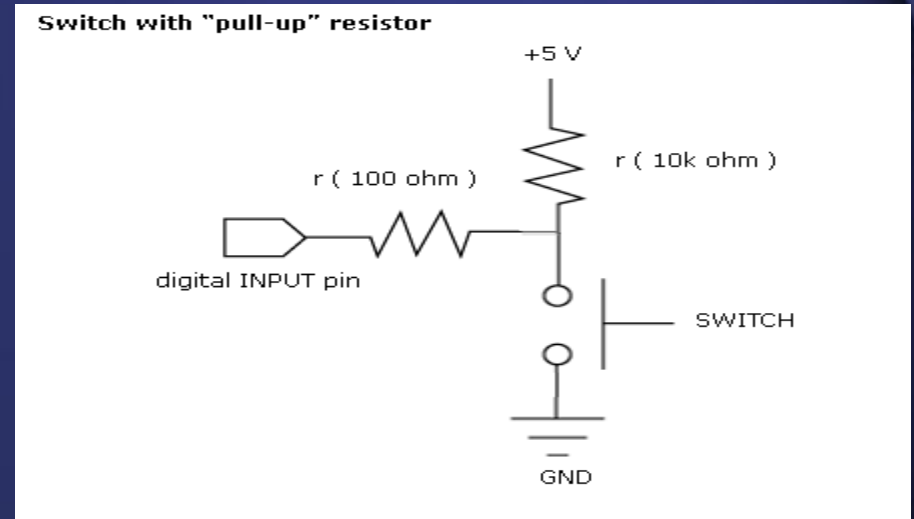
Para generar una señal de tensión con el pulsador, se necesita un divisor de tensión.

Manejo de una entrada digital, lectura de un pulsador

Ejemplo de tipos de conexionado:



En alto cuando se presiona el botón



En bajo cuando se presiona el botón

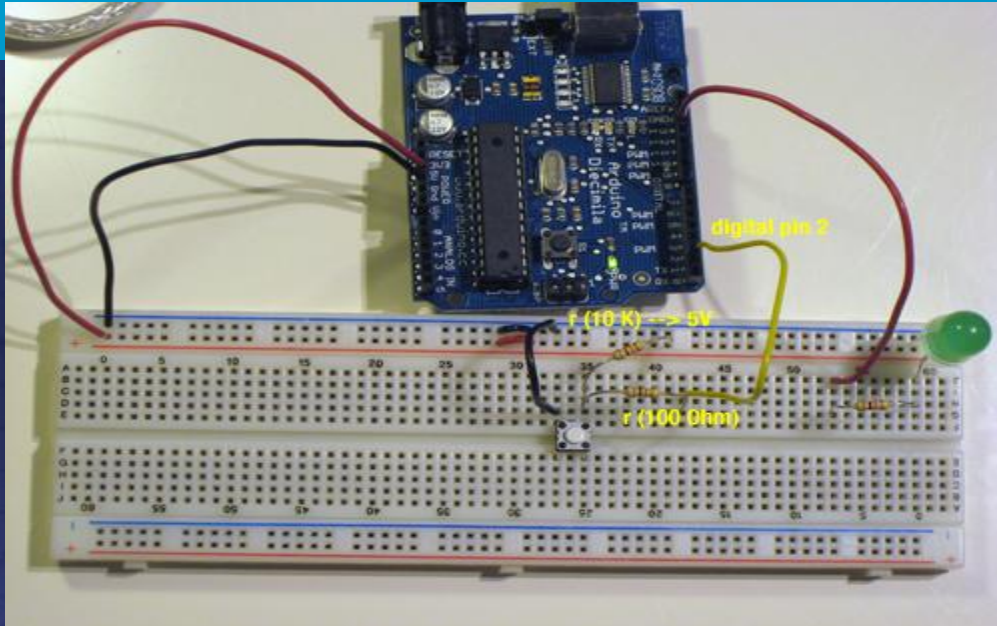
Manejo de una entrada digital, lectura de un pulsador

Elementos necesarios:

- Una resistencia de 10K Ohms.
- Una resistencia de 100 Ohms.
- Un pulsador.
- Un diodo LED (Utilizaremos el que está en la placa)
- Cables para realizar las conexiones.

Utilizaremos el esquema de conexión pull-down, junto con un pulsador, para conectar a un pin de entrada digital, y de esta forma, poder saber cuando el pulsador es presionado. Si el pulsador está presionado, el valor del pin 2 será de 0 voltios (LOW) en caso contrario será de + 5 voltios (HIGH).

Manejo de una entrada digital, lectura de un pulsador



Recordemos que a diferencia de esta imagen utilizaremos el led incorporado en la placa.

Código para Lectura del pulsador

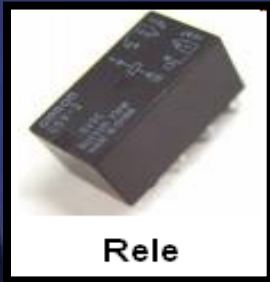
```
int ledPin = 13; // Escoge el pin para el led
int inPin = 2; // Escoge el pin para el pulsador
int val = 0; // variable para leer el estado del pin

void setup() {
  pinMode(ledPin, OUTPUT); // declara el LED como salida
  pinMode(inPin, INPUT); // declara el pulsador como entrada
}

void loop() {
  val = digitalRead(inPin); // Lee el valor del pulsador
  if (val == HIGH) // verifica si la entrada es HIGH (Pulsador sin presionar)
  {
    digitalWrite(ledPin, LOW); // apaga el LED
  }
  else
  {
    digitalWrite(ledPin, HIGH); // Enciende el LED
  }
}
```

Control de un relé

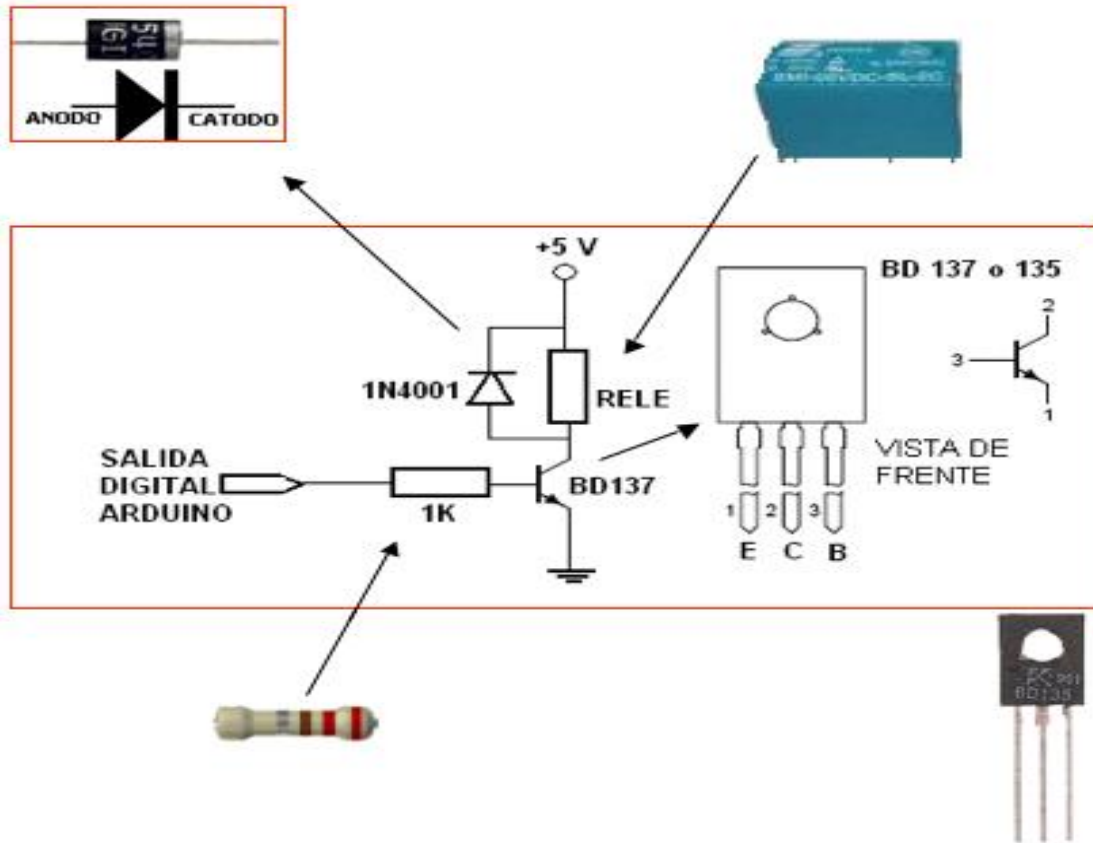
Este sencillo ejemplo enseña como encender una ampollita de 220V de corriente alterna (AC) mediante un circuito de 5V de corriente continua (DC) controlado por Arduino. Se puede utilizar con cualquier otro circuito de 220V con un máximo de 10A (con el relé del ejemplo).



Relé

Componentes: el Relé es un dispositivo, que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de un electroimán, se acciona un juego de uno varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Práctica: Control de un relé



o transistor NPN

codigo Control de un relé

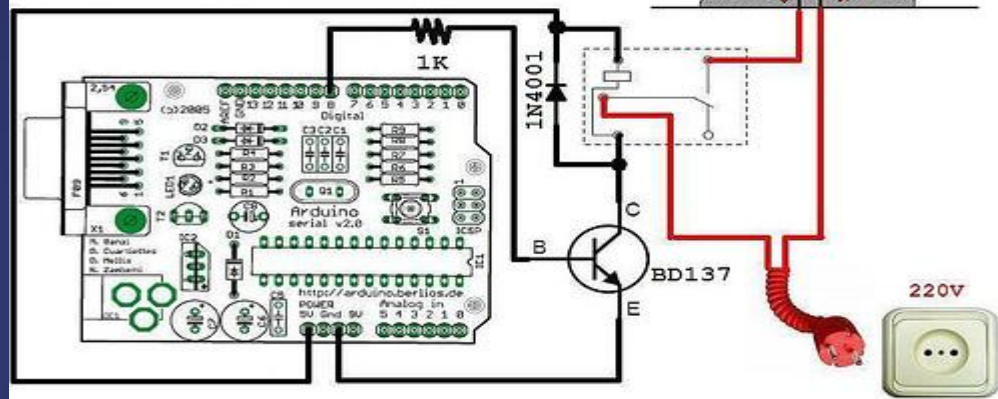
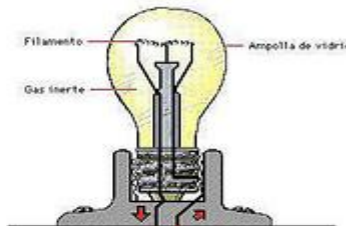
```
/* Enciende y apaga una ampolleta de 220V, cada 2 segundos,  
mediante un relé conectado al PIN 8 de Arduino */  
  
int relayPin = 8;           // PIN al que va conectado el relé  
  
void setup()  
{  
  pinMode(relayPin, OUTPUT); //Configura el Pin8 como salida  
}  
  
void loop()  
{  
  digitalWrite(relayPin, HIGH); // ENCENDIDO  
  delay(2000);  
  digitalWrite(relayPin, LOW);  // APAGADO  
  delay(2000);  
}
```

Control de un relé



Relé

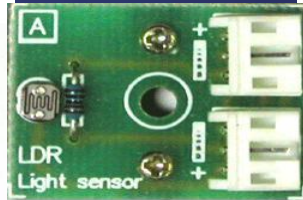
5V DC
220V AC



Elementos analógicos de entrada

Los sensores analógicos a diferencia de los digitales entregan una salida continua, ya sea de voltaje o corriente. Es decir, no solo 0 o 5V , si no que también valores intermedios.

Cada sensor posee su propia escala por lo que probablemente tendrás que ocupar matemáticas para poder calcular el valor correcto de la medición.



Funciones para el manejo de entradas análogas

`int analogRead(pin)` Lee el valor de un pin analógico.
Entrega un valor entre 0 y 1023 que representa 0 o 5V es decir 4.9mV por unidad.

```
int ledPin = 9;           // LED conectado al pin 9
int analogPin = 3;       // potenciómetro conectado al pin 3
int val = 0;             // variable para almacenar el valor
                          leído
void setup()
{
  pinMode(ledPin, OUTPUT); // Configura el pin como
  salida
}
void loop()
{
  val = analogRead(analogPin); // Lee el pin de
  entrada (valor entre 0 a 1023)
}
```

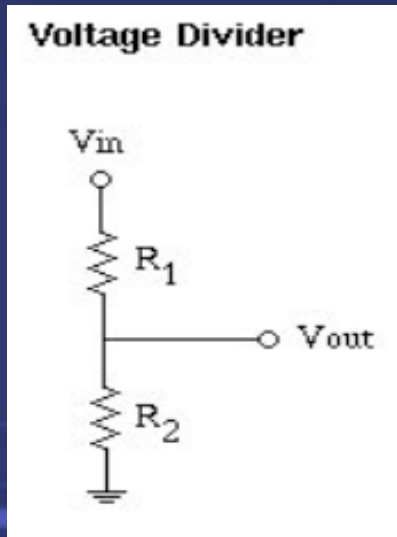
Arduino y sus entradas analógicas

Arduino tiene un conversor análogo digital interno de 6 canales accesibles desde los pines marcados en la placa como ANALOG IN 0-5. Estos pueden leer un voltaje que se encuentre entre los 0 y los 5 voltios. La medición tiene una resolución de 10 bits (1024 valores) lo que nos daría una resolución de $(5/1024) = 0.0048 = 4.8$ mv.

En este caso (ADC interno) para poder hacer la medición buscaremos la forma de implementar un divisor de voltaje con el sensor que estemos trabajando, el valor de la resistencia de división deberá estar en el mismo orden de magnitud que pueda dar la máxima lectura del sensor.

Divisores de tensión para acoplar sensores

En electrónica, un divisor de voltaje (o tensión o potencial) es un circuito lineal simple que produce un voltaje como output (V_{out}) que es una fracción de su voltaje como input (V_{in}). División de voltaje se refiere a la partición del voltaje.

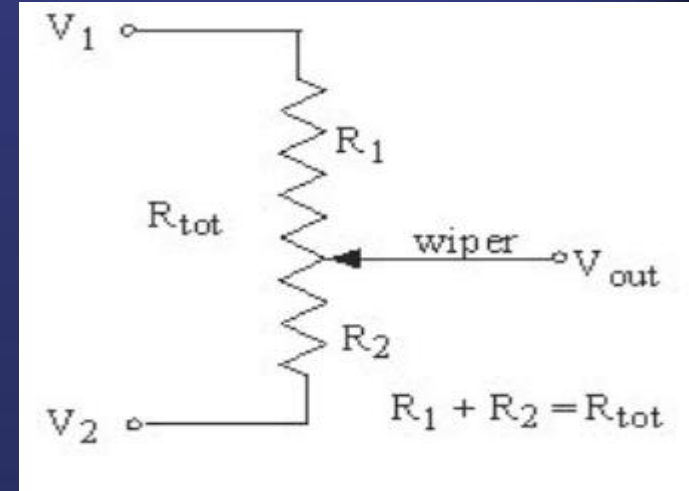


$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

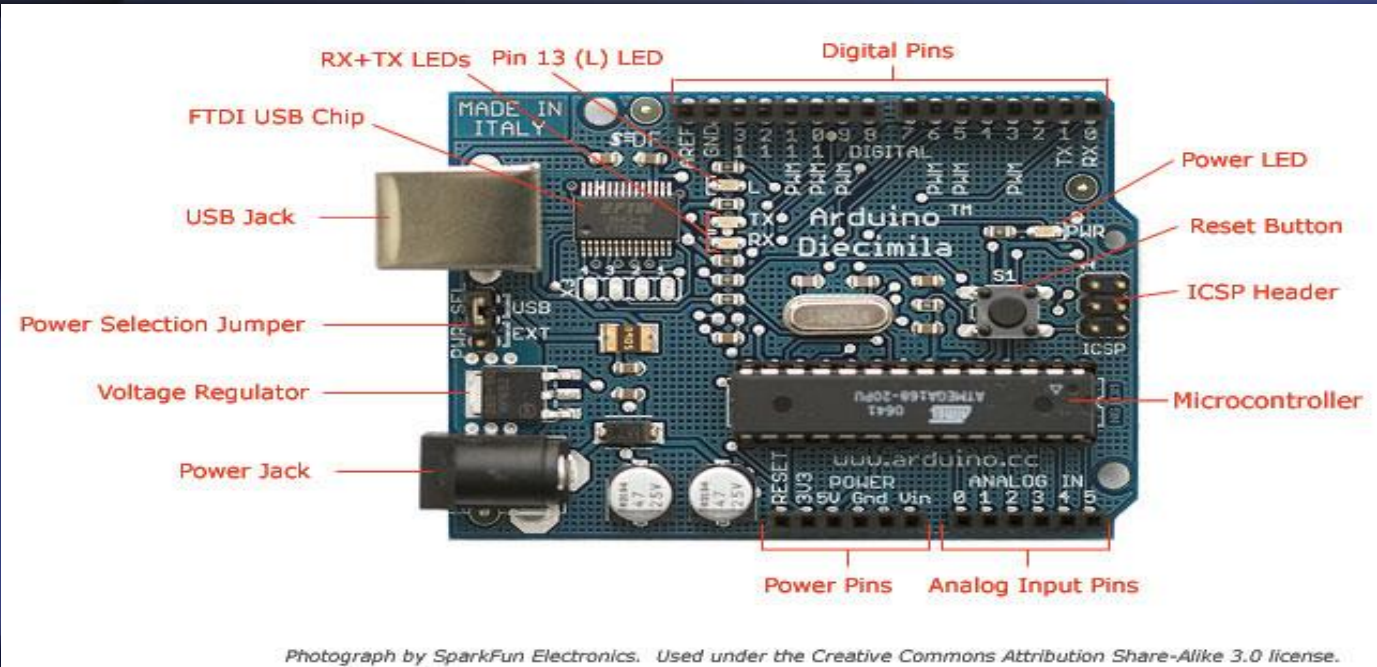
Divisores de tensión para acoplar sensores

Un ejemplo simple de divisor de voltaje, consiste en dos resistencias en serie o un potenciómetro. Es comúnmente utilizado para crear un voltaje de referencia y también puede ser usado como un atenuador de a baja frecuencia.

Para poder leer estos cambios en la resistencia, los pondremos en un circuito (divisor de voltaje) y pasaremos una corriente a través de ellos de tal forma que podemos medir el cambio de voltaje resultante sobre el sensor. ya hecha la medición tendremos que convertir ese voltaje a un valor digital, para este trabajo se utiliza el componente electrónico llamado conversor analógico digital (ADC).



Arduino y sus entradas analógicas



Nota: Las entradas analógicas al igual que las digitales soportan valores de 0 a 5V.

Potenciómetro para simular una entrada analógica

- El potenciómetro es un dispositivo electromecánico que consta de una resistencia de valor fijo sobre la que se desplaza un contacto deslizante, el cursor, que la divide eléctricamente.

$$V_{out} = \left(\frac{R_1}{R_1 + R_2} \right) * V_{in} \text{ (Aplicando la ley de Ohm)}$$

- Un potenciómetro es especificado por su resistencia total, R , entre los terminales externos 1 y 3; El movimiento del cursor origina un cambio en la resistencia medida entre el terminal central, 2, y uno cualquiera de los extremos.

Este cambio de resistencia puede utilizarse para medir desplazamientos lineales o angulares de una pieza acoplada al cursor.

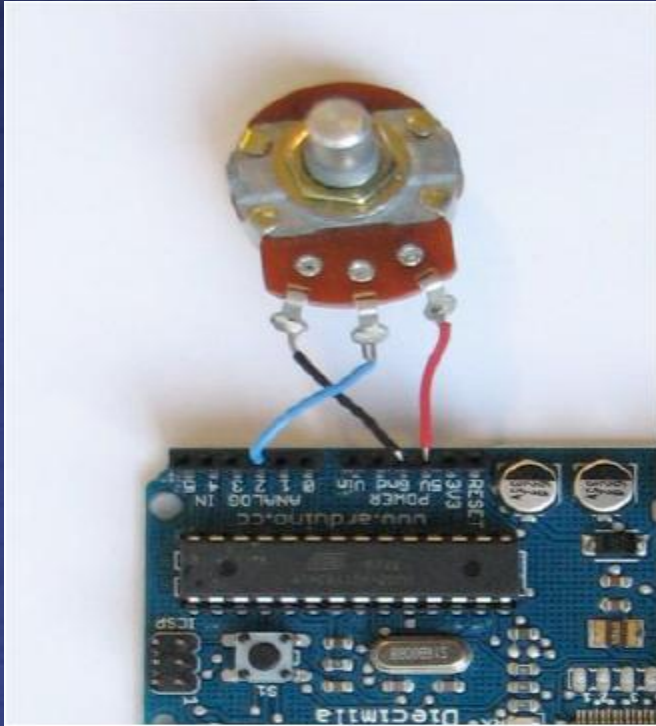
Se conectan en paralelo al circuito y se comporta como un divisor de tensión.

Potenciómetro como una entrada analógica

```
int potPin = 2; // Pin A/I pot
int ledPin = 13; // Pin D/O LED
int val = 0; // variable guarda valor del sensor
```

```
void setup()
{
  pinMode(ledPin, OUTPUT); // ledPin output
}
```

```
void loop()
{
  val = analogRead(potPin); // lee valor del sensor
  digitalWrite(ledPin, HIGH); // enciende LED
  delay(val); //detiene el programa por un tiempo
  // determinado por la variable
  digitalWrite(ledPin, LOW); // apaga el LED
  delay(val); //detiene el programa por un tiempo
  // determinado por la variable
}
```



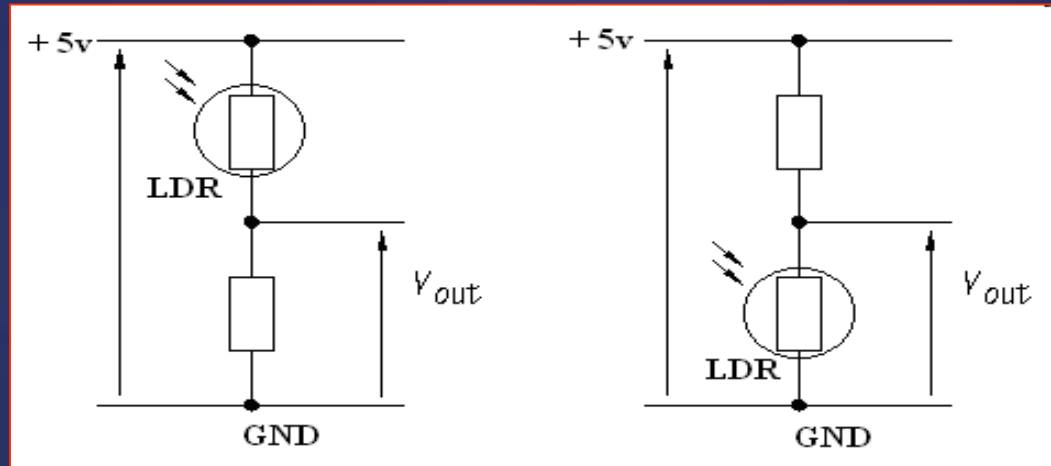
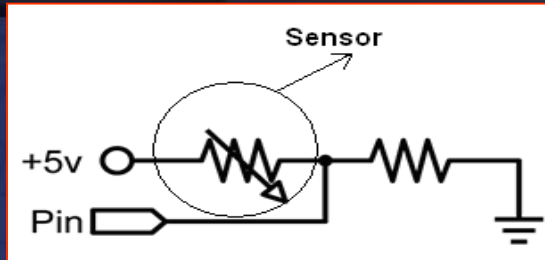
Midiendo Luz con Arduino

- Una fotocélula (o LDR -Light-Dependent Resistor) es una resistencia que varía su intensidad según la luz que recibe. Cuanta más luz menos resistencia ejerce a la corriente



- Una LDR considera un sensor de luminosidad, se pueden usar para apagar las luces es de día o encenderlas cuando es de noche. Estas variables son de tipo analógico, como medir el sonido, la inclinación, la presión o el desplazamiento.

Práctica: Midiendo Luz con Arduino



Listado de componentes:

- * 1 LDR
- * 1 Resistencia de $1k\Omega$
- * Un par de cables

Nota: El LDR del Kit ya viene con la resistencia por lo que no necesitamos agregar nada adicional

Midiendo Luz con Arduino

* En set de codigos AI_lee LDR

```
int LightPin = 3; // selecciona el pin de entrada para el sensor de luz

int ledPin = 13; // selecciona el pin para el LED

int val = 0; // variable para almacenar el valor capturado desde el sensor

void setup() {

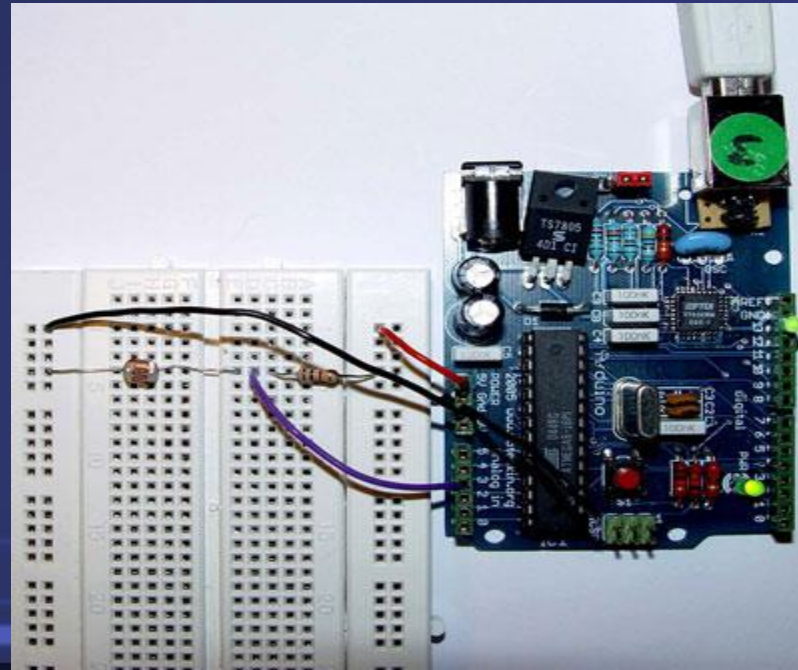
    pinMode(ledPin, OUTPUT); // declara el ledPin en modo salida
    serial.begin(9600);
}

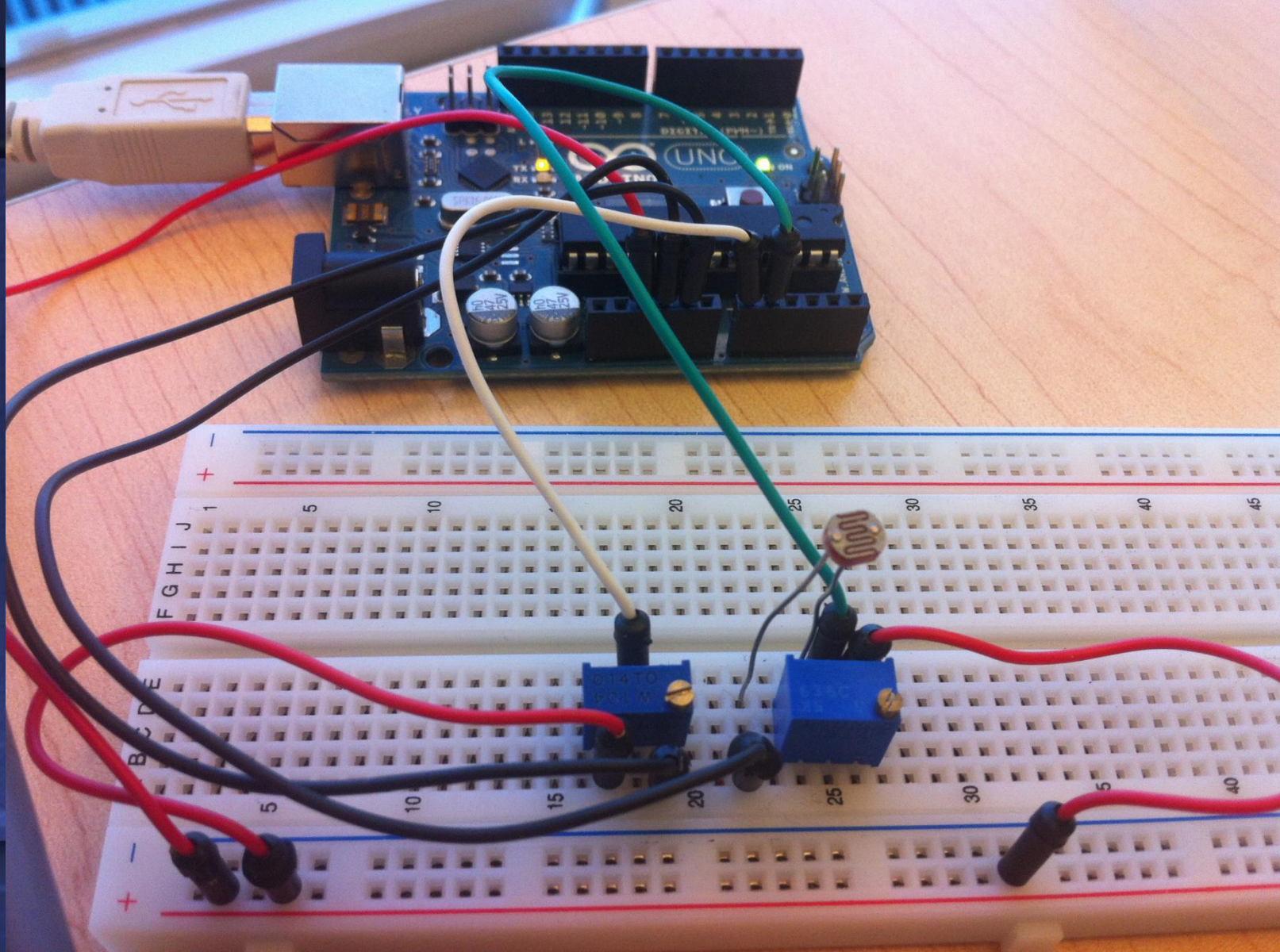
void loop() {

    val = analogRead(LightPin); // read the value from the sensor
    digitalWrite(ledPin, HIGH); // enciende el LED
    delay(val); // detiene el programa por un tiempo
    digitalWrite(ledPin, LOW); // apaga el LED
    delay(val); // detiene el programa por un tiempo
    serial.println ("Sensor de luz: \d", val);
}
```

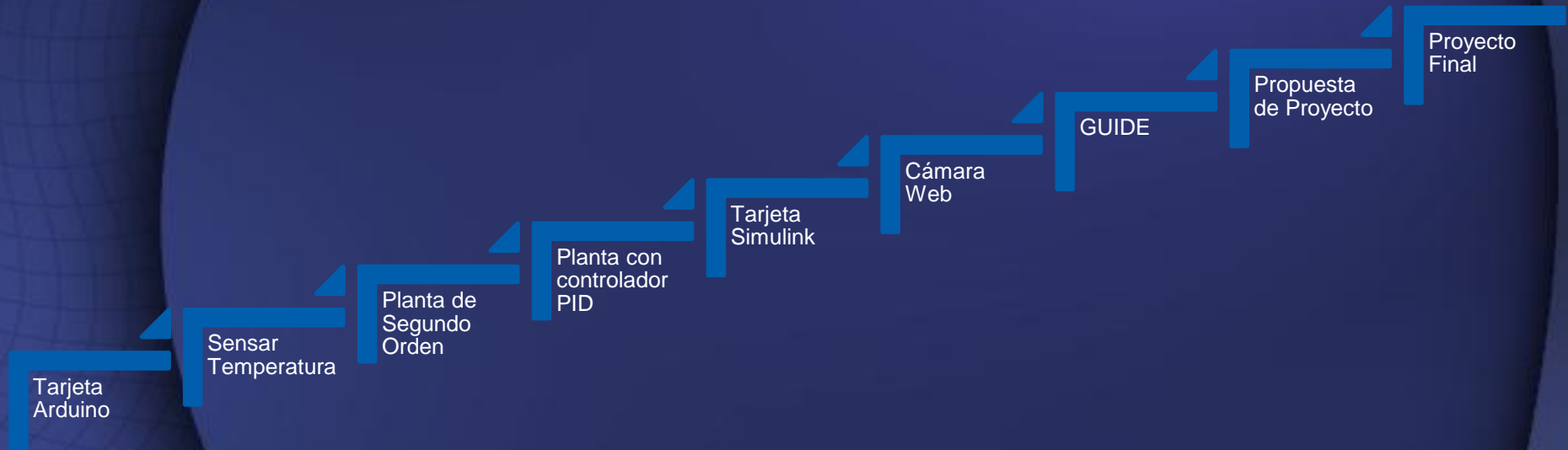
Midiendo Luz con Arduino

Recuerda encender el monitor serial para leer los valores del LDR





Prácticas





Unidad II MUESTREO DE SEÑALES CONTINUAS

OBJETIVO:

Comprender los mecanismos del muestreo e introducir algunos de los conceptos fundamentales y notaciones para los sistemas muestreados. Ilustrar al problema del enmascaramiento de frecuencias.

2.1 Introducción

Se entiende por muestreo <<la acción o proceso de tomar una pequeña parte o cantidad de algo para su examen o análisis>>. En el contexto de control y de las comunicaciones, muestreo significa que *una señal continua en el tiempo se reemplaza por una secuencia de números*, los cuales representan valores de la señal en ciertos instantes.

El muestreo es una prioridad fundamental de los sistemas controlados con computadora debido a la naturaleza discreta de las computadoras digitales. Las variables analógicas del proceso se muestrean en combinación con la conversión analógica y posteriormente se digitalizan para su procesamiento. De este modo, la señal continua que representa la variación en el tiempo de las variables del proceso se convierte en una secuencia de números, los cuales se procesan en la computadora digital. El procesamiento da una nueva secuencia de números, que se convierten en una señal continua y se aplican al proceso. El proceso de convertir una secuencia de números en una señal continua se denomina reconstrucción de la señal.

Debido a que este es el muestreo es una propiedad básica de los sistemas controlados por computadora, es necesario tener un buen conocimiento del proceso de muestreo.



2.2. Descripción de los mecanismos de muestreo

Para la tarea del análisis resulta útil tener una descripción del muestreo. Muestrear una señal continua simplemente significa que se reemplaza la señal por sus valores en conjunto de puntos discretos. Se considera el conjunto de los números enteros $Z = \{\dots -1, 0, 1, \dots\}$ y sea $\{tk: k \in Z\}$ un subconjunto de los números reales denominado instantes de muestreo. Entonces, la versión muestreada de la señal f es la secuencia $\{f(tk): k \in Z\}$. El muestreo es una

operación lineal. A menudo, los instantes de muestreo se distribuyen en intervalos regulares en el tiempo, es decir,

$$tk = k \cdot h$$

En este caso se denomina muestreo periódico y h es el periodo de muestreo o intervalo de muestreo. La frecuencia correspondiente $f_s = 1/h$ (Hz) se denomina frecuencia de muestreo.

También pueden utilizarse otros esquemas más complejos. Por ejemplo, pueden utilizarse diferentes periodos de muestreo para diferentes lazos de control. Esta técnica se denomina muestreo multiperíodo y puede considerarse como la superposición de varios muestreos periódicos.

El caso del muestreo periódico es bien conocido. La mayor parte de la teoría se centra en este caso; sin embargo, el muestreo multiperíodo esta incrementando su importancia debido al uso creciente de los sistemas multiprocesador. Con los modernos programas para procesos concurrentes es posibles diseñar sistemas como si estuviesen formados por diferentes procesadores funcionando asincrónicamente. También se encuentran ventajas técnicas en el uso de diferentes velocidades de muestreo para diferentes variables.

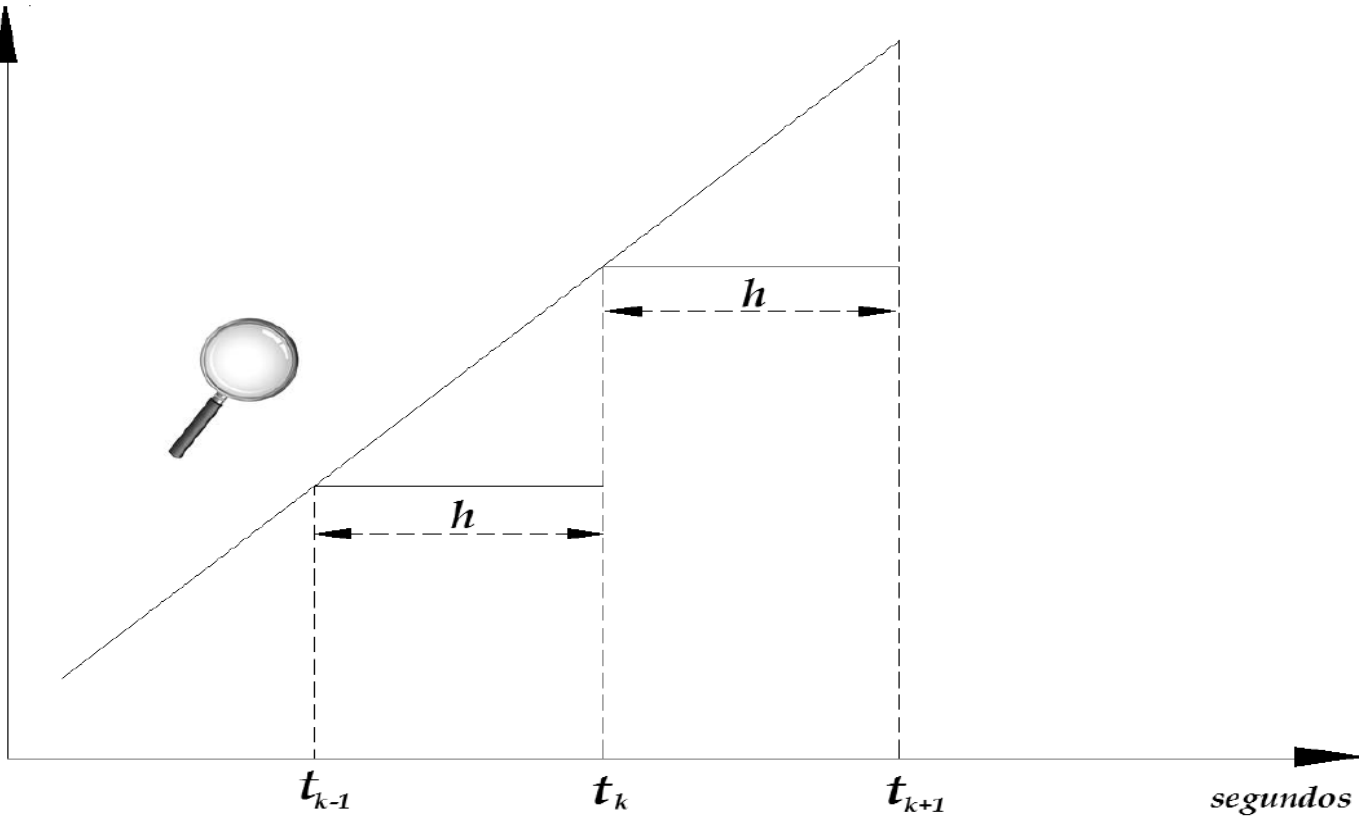


Figura 9.6 Muestreo periódico.

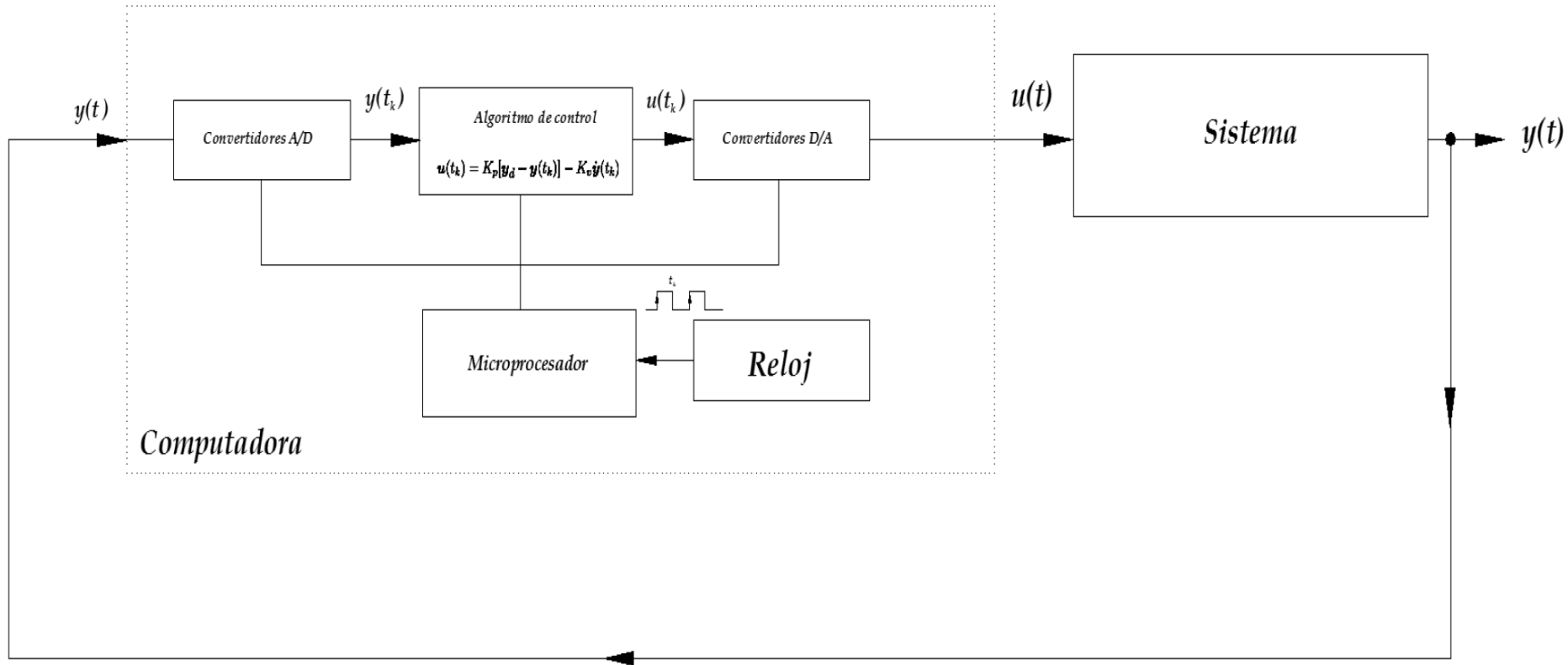


Figura 9.1 Sistema controlado por computadora.

2.3 El teorema de muestreo

Se pierde muy poca información muestreando una señal continua si los instantes de muestreo están muy próximos; sin embargo, gran parte de la información referente a la señal puede perderse si los puntos de muestreo están alejados. Esta propiedad se ilustra en la figura 2.1, en el cual muestra una señal senoidal. La figura muestra que una señal senoidal no puede distinguirse de una señal nula si la frecuencia de la señal continua esta definida de forma única por su versión muestreada. El siguiente teorema de las condiciones da las condiciones para el caso del muestreo periódico.

Figura 2.1 Pérdida de información debida a un muestreo lento. La curva senoidal se muestrea a una velocidad de dos muestras por periodo.

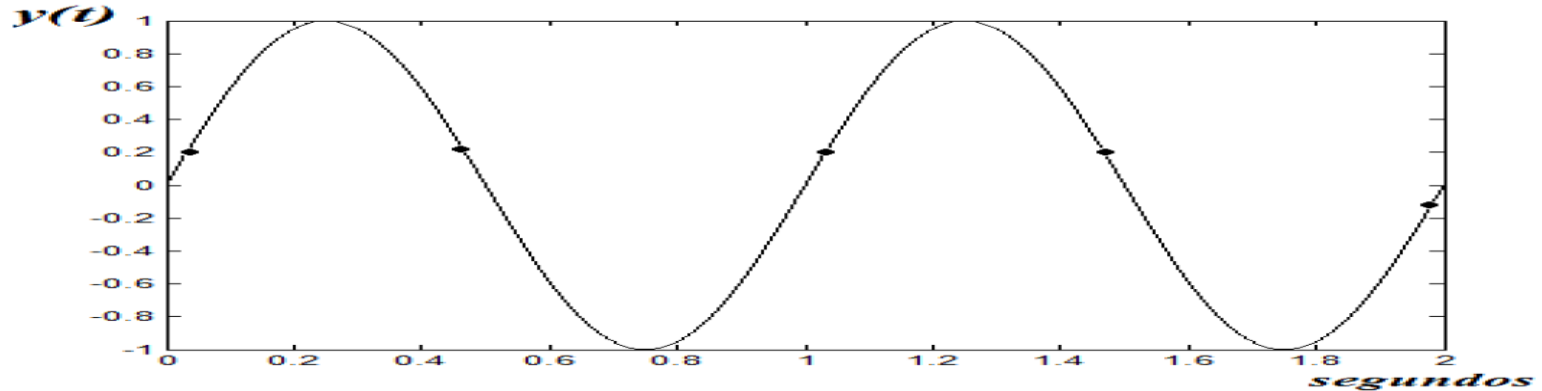
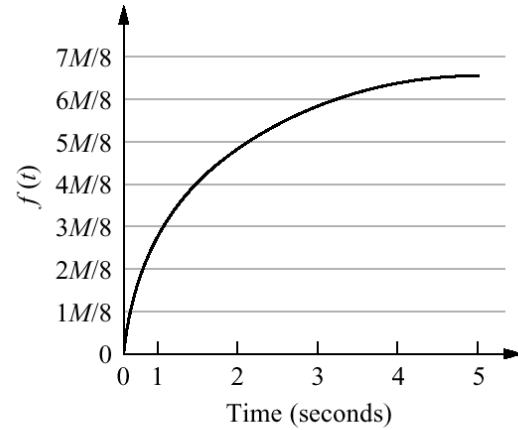


Figura 9.4 Señal continua y 5 puntos discretos.

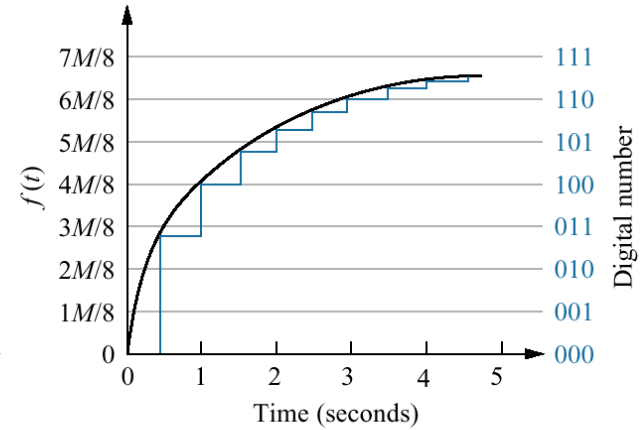
Figure 13.4

Steps in analog-to-digital conversion:

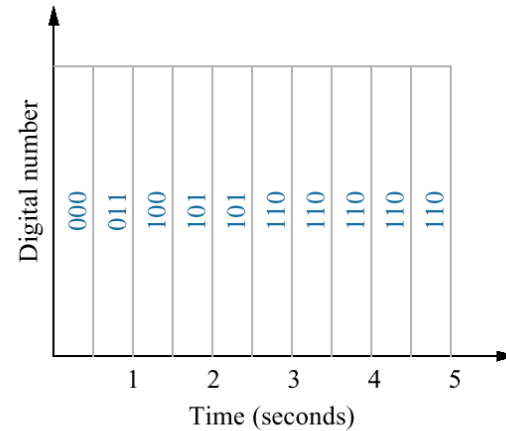
- a. analog signal;
- b. analog signal after sample-and-hold;
- c. conversion of samples to digital numbers



(a)



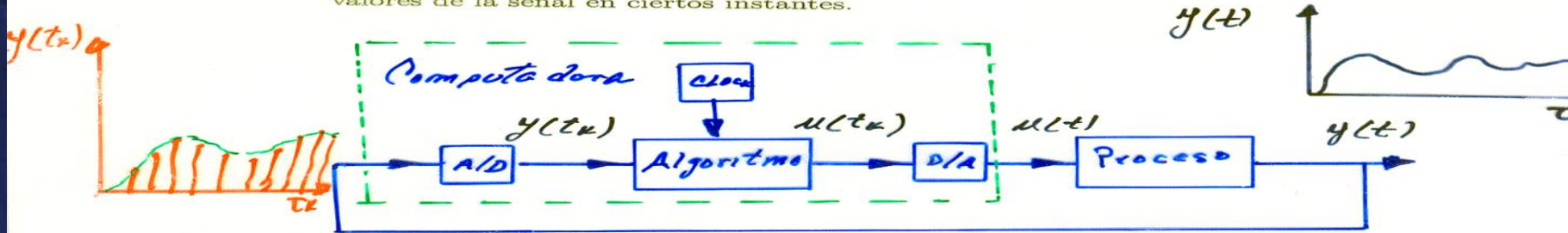
(b)



(c)

Muestreo de Señales Continuas

- Una definición del término **muestreo** (*sampling*) significa el proceso de tomar una pequeña parte o cantidad de algo para su prueba y análisis.
- Para propósitos de control el término **muestreo** significa que la señal en tiempo continuo es remplazada por una secuencia de números, los cuales representan los valores de la señal en ciertos instantes.



- El proceso de convertir una secuencia de números dentro de una señal variante en el tiempo continuo es llamada **reconstrucción de la señal**.

$$Z = \{\dots, -1, 0, 1, \dots\}$$

$$\{t_k = kh : k \in Z, h \in \mathbb{R}\}.$$

donde h [seg] es el período de muestreo y $f_s = 1/h$ [Hz] es llamada la frecuencia de muestreo.

- El muestreo es una *operación lineal*.

Sea $f(t)$ una señal continua, la versión muestreada de la señal f está dada por la siguiente secuencia:

$$\{f(t_k) : k \in Z\}$$



2.4 Reconstrucción

La inversa de la operación de muestreo, es decir, la conversión de una secuencia de números $\{ f(tk): k \text{ pertenece a } \mathbb{Z} \}$ en función continua $f(t)$, se denomina reconstrucción. En sistemas controlados por computadora es necesario convertir las acciones de control, calculada por la computadora como una secuencia de números, en una señal continua que puede aplicarse al proceso. El filtrado digital es del mismo modo necesario convertir la repetición de la señal filtrada, dada como una secuencia de números, en una función continua. En esta sección se estudian diferentes tipos de reconstrucciones.

RECONSTRUCCIÓN DE SHANNON

Para el caso de señales periódicas de banda limitada se deduce del teorema del muestreo que la reconstrucción viene dada por la ecuación (2.1). Esta reconstrucción se denomina *reconstrucción de Shannon*. La ecuación (2.1) define la operación inversa del muestreo, que puede considerarse como un operador lineal. Sin embargo no es un operador casual, puesto que los valores de f en los instantes t están expresados en función de los valores pasados $\{f(kh), k = t/h\}$, y de los valores futuros $\{f(kh), k = t/h\}$. Esto implica que la reconstrucción de Shannon no es útil en control con computadora, pero puede usarse en algunos problemas de comunicaciones donde se aceptan

determinados retardos. Otros inconvenientes de la reconstrucción de Shannon reside en su complejidad y en que sólo puede aplicarse en muestreo periódico. Resulta, por tanto, útil disponer de otros tipos de reconstrucciones.



Teorema 2.1 - Teorema de muestreo de Shannon. Una señal continua con una transformada de Fourier nula fuera del intervalo $(-\infty, \infty)$ está determinada de forma única por sus valores en puntos equidistantes si la frecuencia de muestreo es superior a $2\omega_0$. La señal continua puede calcularse a partir de la señal muestreada mediante la fórmula de interpolación.

$$f(t) = \sum_{k=-\infty}^{\infty} f(kh) \frac{\text{Sen } \omega_s (t - kh)/2}{\omega_s (t - kh)/2} \quad (2.1)$$

donde ω_s es la frecuencia angular de muestreo en radianes por segundo (rad/s).

Demostración. consideremos la señal f y sea F su transformada Fourier.

$$F(\omega) = \int_{-\infty}^{\infty} e^{i\omega t} f(t) dt \quad 2.2$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} F(\omega) d\omega \quad 2.3$$

y definamos

$$F_s(\omega) = 1/h \sum_{k=-\infty}^{\infty} F(\omega + k\omega_s) \quad 2.4$$

La demostración se basa en la observación de que la muestra $f(kh)$ pueden considerarse como los coeficientes de la serie de Fourier de la función periódica $F_s(\omega)$. El desarrollo de Fourier de F_s es

$$F_s(\omega) = \sum_{k=-\infty}^{\infty} C_k e^{-ik\omega h} \quad 2.5$$

donde los coeficientes están definidos por

$$C_k = \frac{1}{\omega_s} \int_0^{\omega_s} e^{ik\omega h} F_s(\omega) d\omega$$

Utilizando la definición de los coeficientes de Fourier, es inmediato demostrar que

$$C_k = f(kh) \quad 2.6$$



$$C_k = f(kh)$$

2.6

De este modo se obtiene que la señal muestreada $\{f(kh), k = \dots, -1, 0, 1, \dots\}$, define de modo único la función $F_s(w)$. La función F , por consideraciones del teorema, es nula fuera del intervalo $(-\omega_0, \omega_0)$. si $w_s > 2w_0$ de (2.4) se obtiene

$$F(w) = \begin{cases} hF_s(w) & |w| \leq \frac{w_s}{2} = \omega_N \\ 0 & |w| > \frac{w_s}{2} \end{cases}$$

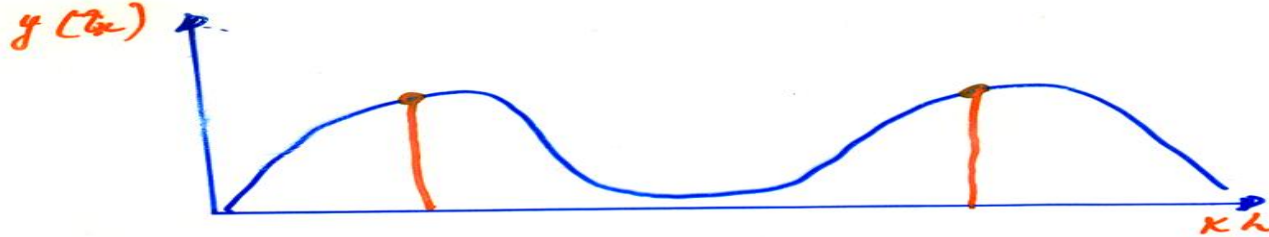
frecuencia de Nyquist.

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} F(\omega) d\omega = \frac{h}{2\pi} \int_{-w_s/2}^{w_s/2} e^{i\omega t} F_s(\omega) d\omega = \frac{h}{2\pi} \int_{-w_s/2}^{w_s/2} e^{i\omega t} \sum_{k=-\infty}^{\infty} e^{ik\omega h} f(kh) d\omega$$

$$f(t) = \sum_{k=-\infty}^{\infty} f(kh) \frac{h}{2\pi} \int_{-w_s/2}^{w_s/2} e^{i\omega t - i\omega kh} d\omega = \sum_{k=-\infty}^{\infty} f(kh) \frac{h}{2\pi i (t - kh)} e^{i\omega t - i\omega kh} \Big|_{-w_s/2}^{w_s/2}$$

$$= \sum_{k=-\infty}^{\infty} f(kh) \frac{\text{Sen } w_s(t - kh)/2}{\pi(t - kh)/h}$$

y como $w_s h = 2\pi$ Q.E.D



Teorema de muestreo de Shannon. Una señal en tiempo continuo con transformada de Fourier cero fuera del intervalo $(-\omega_o, \omega_o)$ está dada en forma única por sus valores en puntos equidistantes si la frecuencia es mayor a $2\omega_o$.

Además, la señal en tiempo continuo puede ser calculada a partir de la señal muestreada mediante la siguiente regla de interpolación:

$$f(t) = \sum_{k=-\infty}^{\infty} f(kh) \frac{\sin \omega_s(t - kh)/2}{\omega_s(t - kh)/2} \quad (1)$$

donde ω_s [rad/seg] es la frecuencia angular de muestreo.

- $\omega_N = \omega_s/2$ es llamada la frecuencia de Nyquist.
- La ecuación (1) es llamada reconstrucción de Shannon y define la operación inversa del proceso de muestreo.

Desventajas:

- ♣ Es un operador no causal, debido a que el valor de f en el instante de tiempo t es expresado en términos de sus valores pasados $\{f(kh) : k \leq t/h\}$ así como de sus valores futuros $\{f(kh) : k > t/h\} \Rightarrow$ que la reconstrucción de Shannon no es útil para control en computadora.
- ♣ Es complicada.
- ♣ El teorema de Shannon es válido para muestreos periódicos de señales de banda limitada.

2.5 Enmascaramiento o solapamiento de frecuencias.

Si una señal continua que tiene una transformada de Fourier F se muestrea periódicamente se obtiene de las señales muestreadas $f(kh)$, $k = \dots -1, 0, 1, \dots$ puedan interpretarse como los coeficientes de Fourier de la función F_s .

De este modo, la función F_s puede interpretarse como la transformada de Fourier de la señal muestreada. La función es periódica de un período igual a la frecuencia de muestreo ω_s . Si la señal continua no tiene componentes de frecuencia superiores a la frecuencia de Nyquist, la transformada de Fourier es simplemente una repetición periódica de la transformada de Fourier de la señal continua vease la figura 2.4:



figura 2.4 Relación entre la transformada de Fourier para señal continua y muestreo para diferentes frecuencias de muestreo

Se obtiene de la ecuación 2.4 que el valor de la transformada de Fourier de la señal muestreada a la frecuencia ω es la suma de los valores de la transformada de Fourier de la señal continua a las frecuencias $\omega + n\omega_s$. Después de muestrear, ya no es posible separar la contribución de cada una de las frecuencias. La frecuencia ω puede. De este modo, considerarse como el



enmascaramiento de $\omega + n\omega_s$. Se acostumbra a considerar únicamente las frecuencias positivas. La frecuencia ω es entonces la máscara de

$$\omega_s - \omega, \omega_s + \omega, 2\omega_s - \omega, 2\omega_s + \omega \dots \text{ donde } 0 \leq \omega \leq \omega_N$$

Después de muestrear, una frecuencia no puede distinguirse de las que enmascaradas. El enmascaramiento fundamental para una frecuencia

$\omega_i > \omega_N$ viene expresado por

$$\omega = |(\omega_i + \omega_N) \bmod(\omega_s) - \omega_N|$$

Prefiltrado

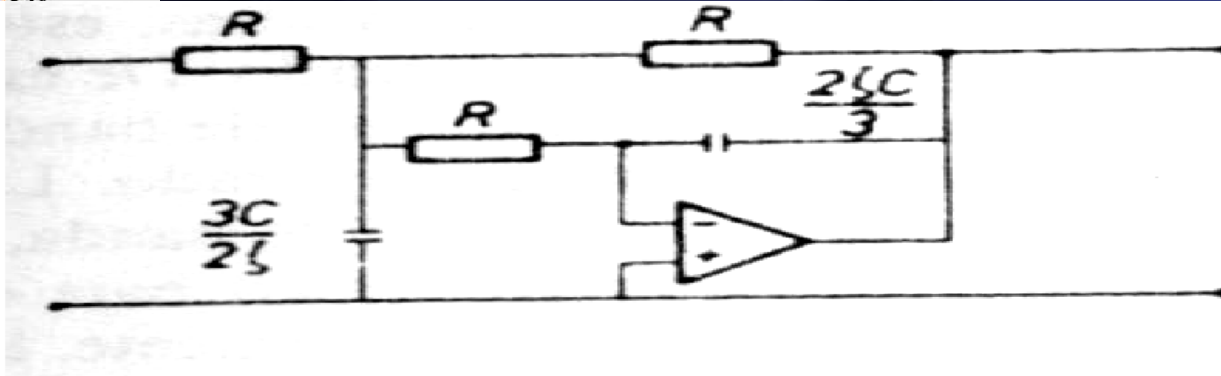


Figura 2.5 Realización mediante un amplificador operacional de un filtro de segundo orden. La función de transferencia del filtro es dada por la ecuación de segundo orden $G(s)$ con $w = 1/RC$.

Order	Butterworth		ITAE		Bessel	
	ω/ω_0	ζ	ω/ω_0	ζ	ω/ω_0	ζ
2	1	0.71	1.00	0.71	1.00	0.87
4	1	0.38	1.33	0.32	3.39	0.62
	1	0.92	0.75	0.83	3.02	0.96
6	1	0.26	1.30	0.32	5.14	0.49
	1	0.71	0.98	0.60	5.57	0.82
	1	0.97	0.79	0.93	4.34	0.98

Tabla 2.1 Amortiguamiento ζ y frecuencia natural w para filtros Butterworth, ITAE (Integral en el tiempo del error absoluto), y de Bessel. Los filtros de orden superior se obtienen poniendo en cascada filtros de la forma de segundo orden. Aquí, w_0 es la frecuencia natural deseada del filtro.



CUV_yTT-BUAP

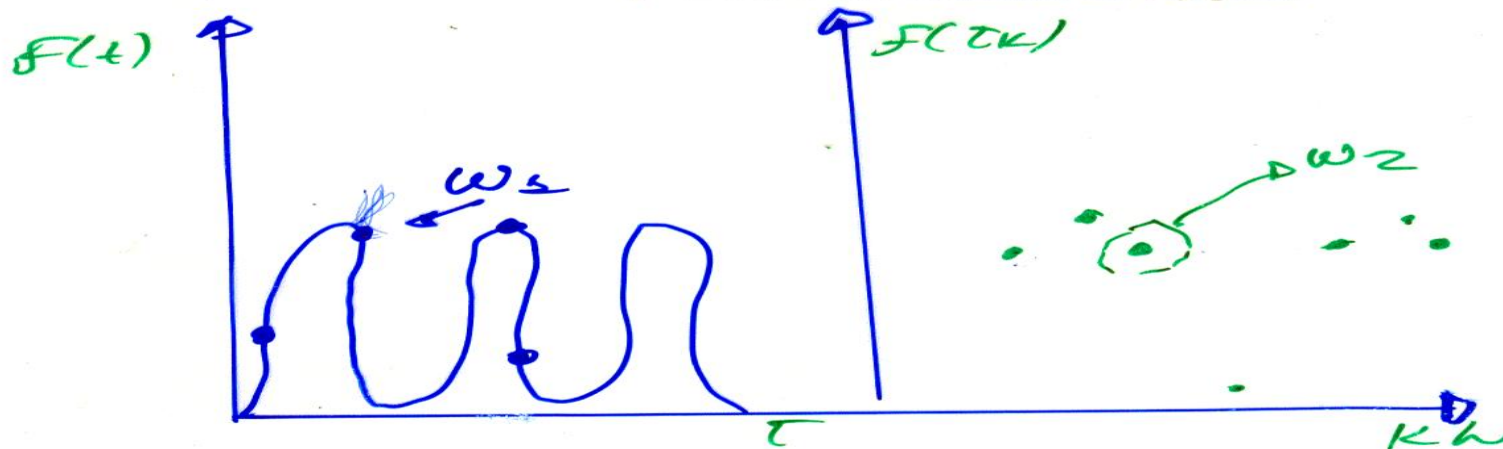


Order	Butterworth		ITAE		Bessel	
	ω/ω_0	ζ	ω/ω_0	ζ	ω/ω_0	ζ
2	1	0.71	1.00	0.71	1.00	0.87
4	1	0.38	1.33	0.32	3.39	0.62
	1	0.92	0.75	0.83	3.02	0.96
6	1	0.26	1.30	0.32	5.14	0.49
	1	0.71	0.98	0.60	5.57	0.82
	1	0.97	0.79	0.93	4.34	0.98

Aliasing

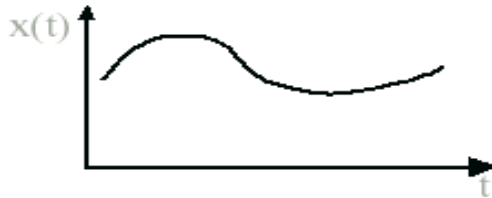
Es un peculiar efecto que aparece cuando la señal muestreada contiene frecuencias que son mayores que la mitad del periodo de muestreo. Después de muestrear no es posible separar esas frecuencias.

- El muestreo es una operación lineal, sin embargo es no causal, esto explica porque nuevas frecuencias son creadas en el proceso del muestreo.
- Las señales reales no tienen transformada de Fourier fuera de su banda de frecuencia, de ahí que las componentes de frecuencias altas sean transformadas a frecuencias bajas en el proceso de muestreo.
- La *frecuencia de aliasing* es una componente de frecuencia baja que aparece en la señal reconstruida a partir de los datos adquiridos (muestreados) por usar un periodo de muestreo insuficiente.
- Es recomendable filtrar la señal continua para obtener una señal filtrada (analógica) sin las frecuencias superiores a la frecuencia de Nyquist.

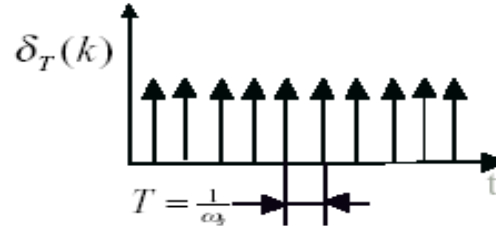


Muestreo de señales de tiempo continuo

Señal continua



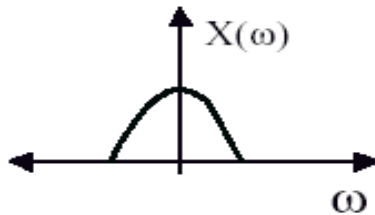
Tren de impulsos



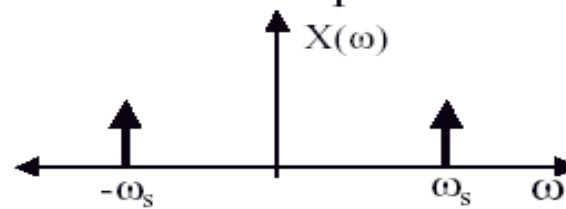
Señal muestreada



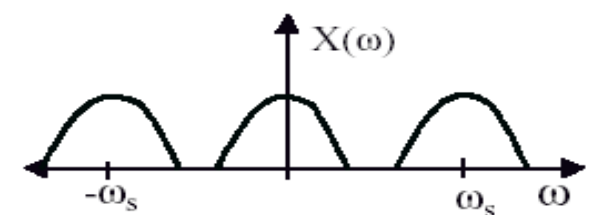
Espectro de Fourier de $f(t)$



Espectro de Fourier del tren de impulsos



Espectro de Fourier de la señal muestreada





Aliasing (apodamiento).

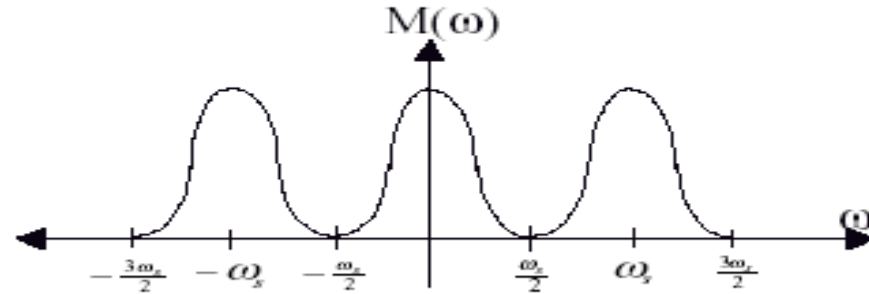
Es un fenómeno relacionado con el muestreo de una señal continua.

Una consecuencia importante del apodamiento es la incapacidad de distinguir entre dos señales cuyas frecuencias son múltiplos enteros de la frecuencia de muestreo.

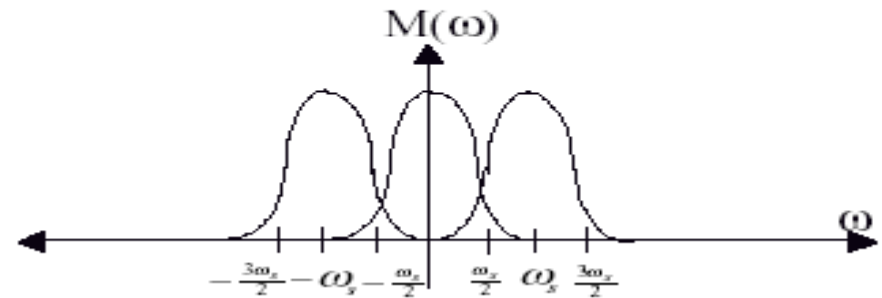
Teorema de muestreo:

“La frecuencia de muestreo ω_s debe ser al menos del doble del valor de la frecuencia mas alta presente en la señal de interés”.

Muestreo sin *aliasing*



Muestreo con *aliasing*

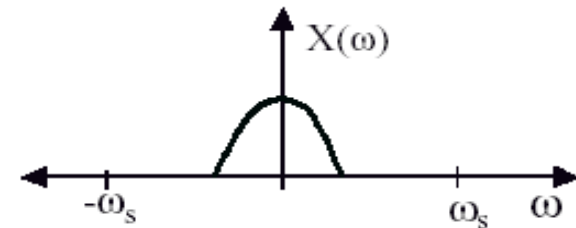
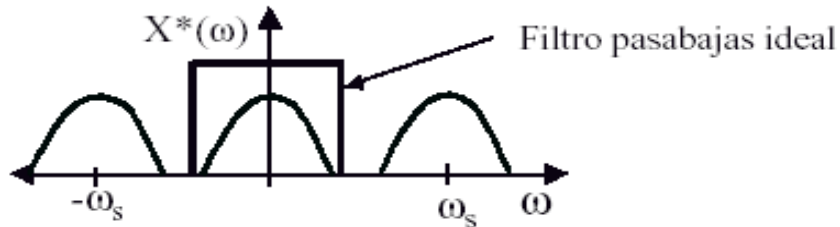


El proceso de recuperación de datos representa el hacer pasar la señal de datos muestreados a través de un filtro pasabajas para obtener la señal original



$$FPBI = \begin{cases} 1 & -\frac{\omega_s}{2} < \omega < \frac{\omega_s}{2} \\ 0 & \text{en otro lado} \end{cases}$$

Espectro de Fourier de la señal muestreada



MANTENEDOR DE ORDEN CERO (ZOH)

Una reconstrucción casual simple está definida por

$$f(t) = f(t_k), \quad t_k \leq t < t_{k+1} \quad (2.2)$$

Esto significa que la señal reconstruida es constante a intervalos, continua por la derecha e igual a la señal muestreada en los instantes de muestreo. De este modo, el valor reconstruido se mantiene constante hasta el siguiente periodo de muestreo.

Debido a su simplicidad, el mantenedor de orden cero (ZOH de inglés Zero-Order Hold) es muy común en sistemas controlados con computadora. Los convertidores D/A comerciales suelen diseñarse de tal modo que cada valor de una conversión se mantienen constantes hasta que se realiza una nueva conversión.

El mantenedor de orden cero tiene la ventaja de que puede utilizarse para muestreo aperiódico. Debe observarse, sin embargo, que la reconstrucción definida por (2.2) proporciona una operación exacta de la operación inversa al muestreo solamente para señales que son continuas por la derecha y constantes sobre cada intervalo de muestreo. Para otro tipo de señal, la reconstrucción de (2.2) produce errores (véase figura 2.2). Para el muestreo periódico de una señal con una derivada privada suave, puede obtenerse la siguiente estimación del error. El máximo valor de error viene dado por

$$e_{\text{ZOH}} = \max_k \left| F(t_{k+1}) - F(t_k) \right| \leq h \max \left| f'(t) \right|$$

donde f' es la derivada de f .

Figura 2.2 Muestreo y reconstrucción con un mantenedor de orden cero de una señal continua.

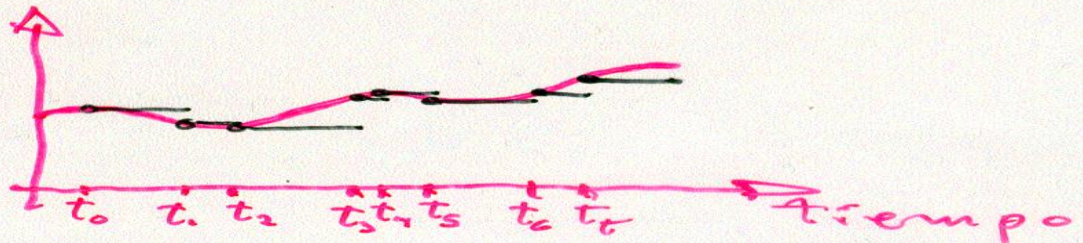
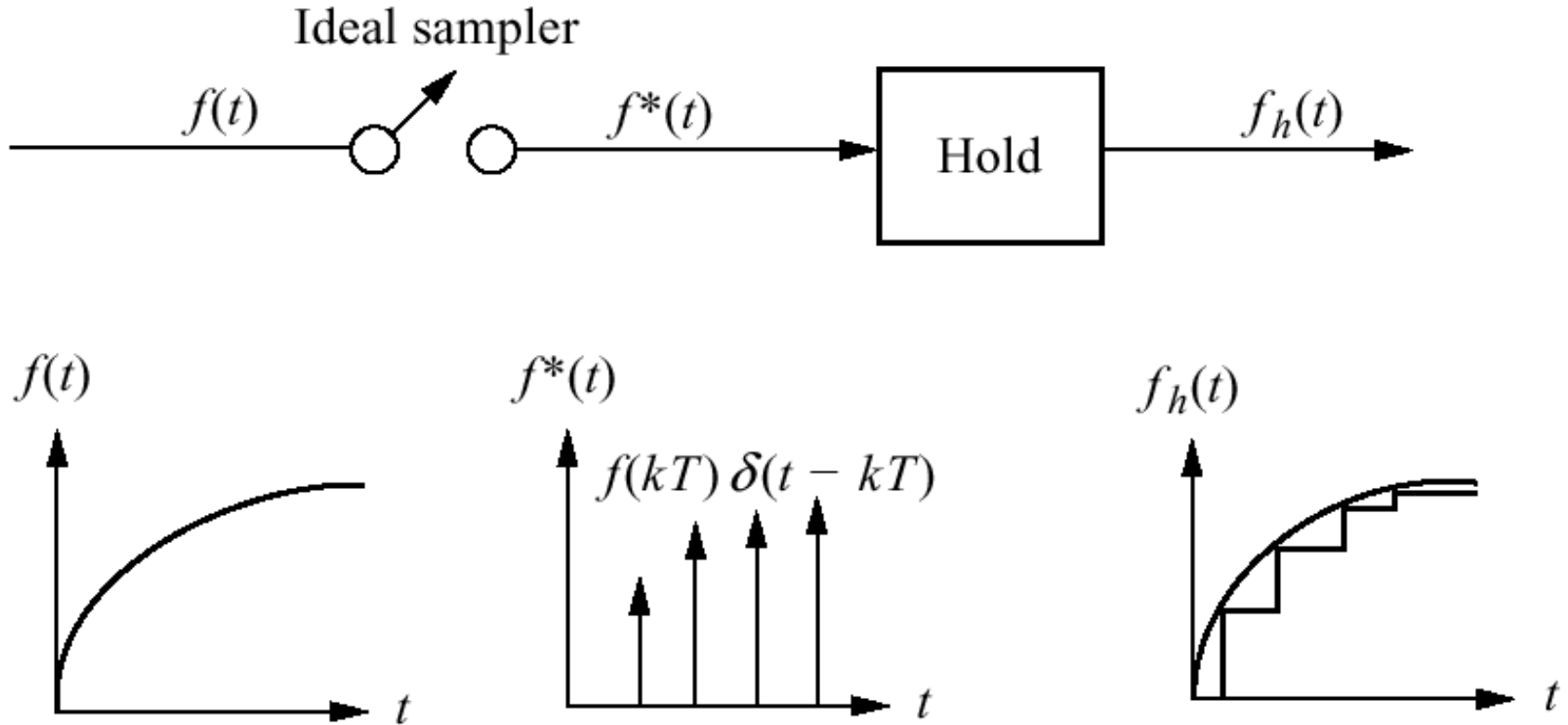


Figure 13.7

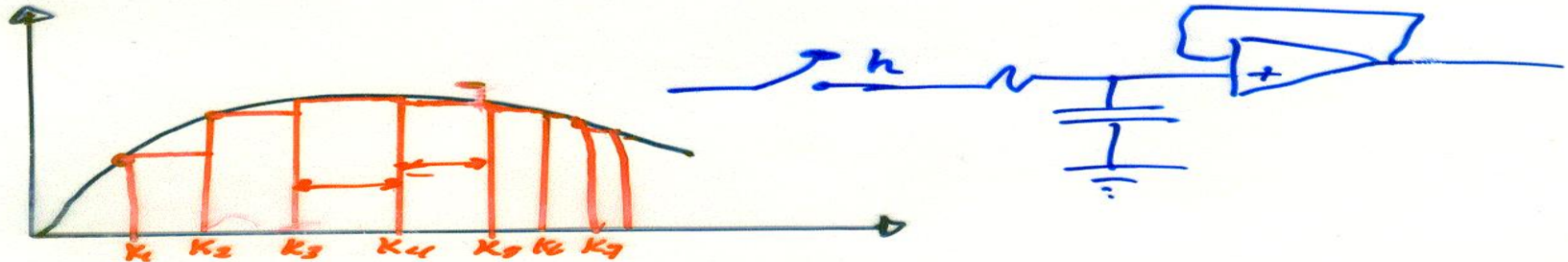
Ideal sampling and the zero-order hold



Retenedor de Orden Cero

Una sencilla reconstrucción causal está dada por

$$f(t) = f(t_k), \quad t_k \leq t < t_{k+1}$$



- El retenedor de orden cero, puede ser pensado como una extrapolación usando polinomios de grado cero.
- La reconstrucción de la señal es constante a pedazos, continua en la parte derecha e igual a la señal muestreada en los instantes de muestreo.
- El valor de la reconstrucción se mantiene constante hasta el próximo instante de muestreo.
- El retenedor de orden cero puede ser usado para muestreo no periódico.
- Para un muestreo periódico de una señal $f(t)$ con una primera derivada suave $\dot{f}(t)$, el siguiente estimado del error e_{ZOH} es obtenido, cuyo valor máximo está dado como:

$$e_{ZOH} = \max_k |f(t_{k+1} - f(t_k))| \leq h \max_t \dot{f}(t).$$

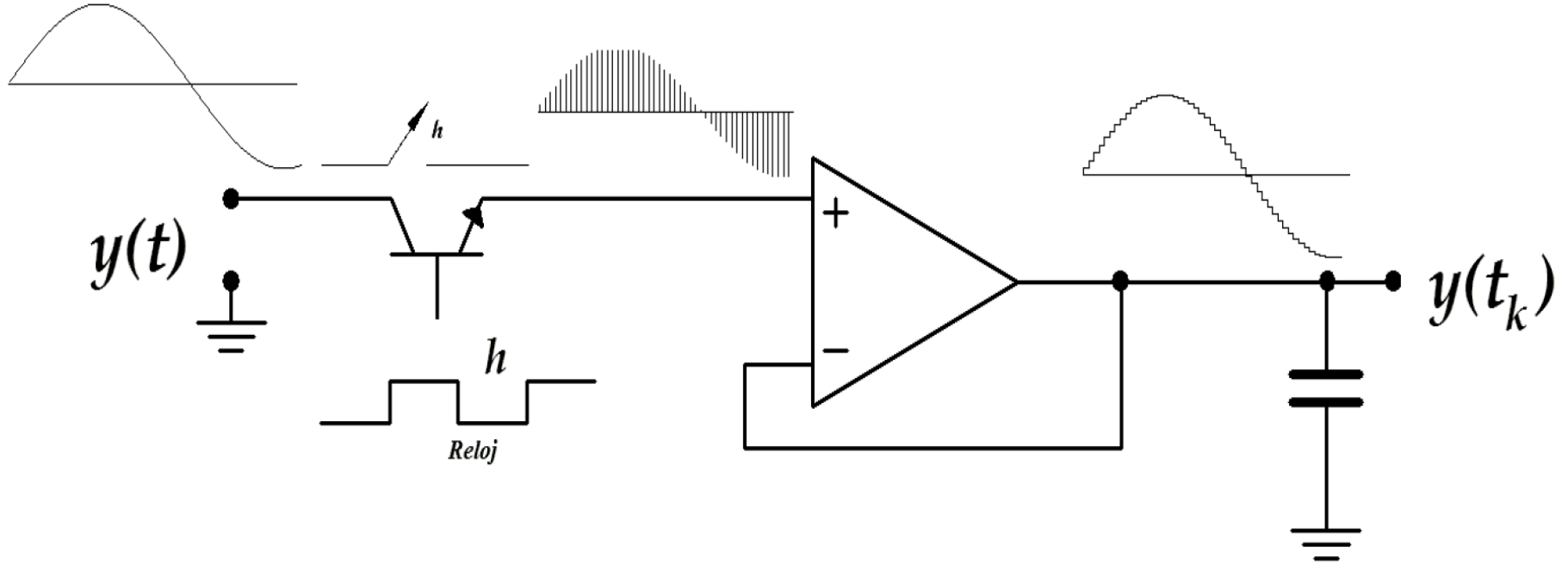


Figura 9.2 Circuito electrónico retenedor y muestreador de orden cero.

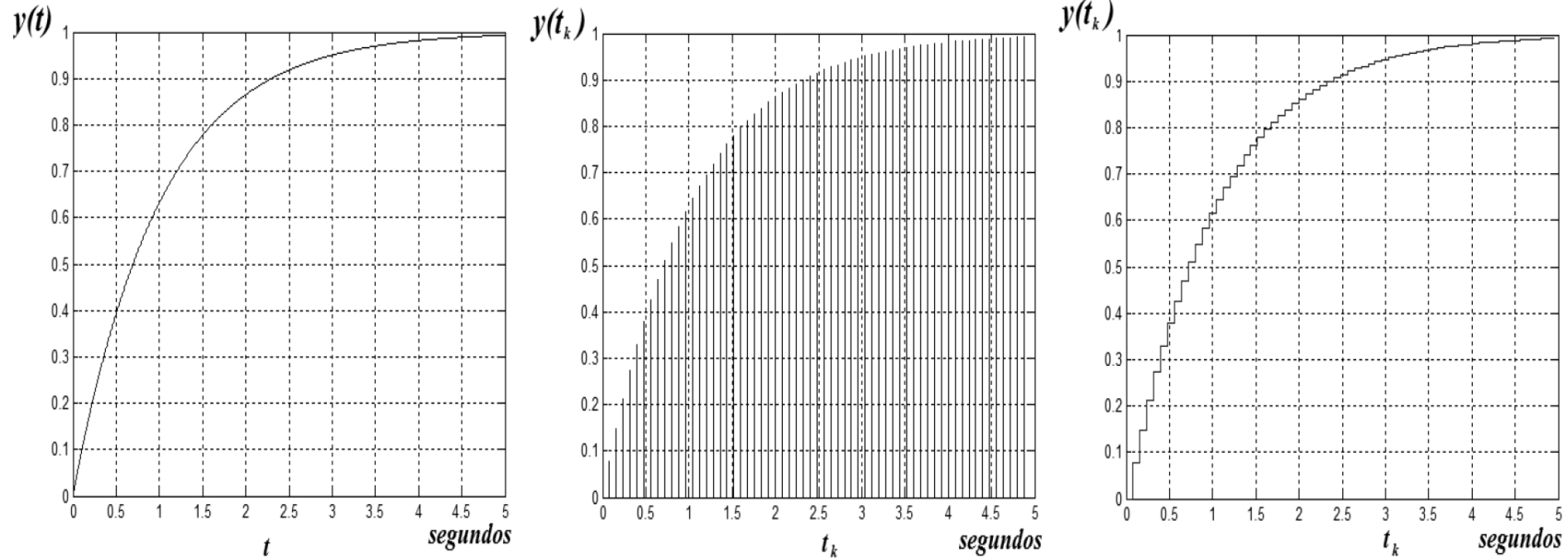


Figura 9.3 Proceso de discretización de una señal continua $y(t)$.

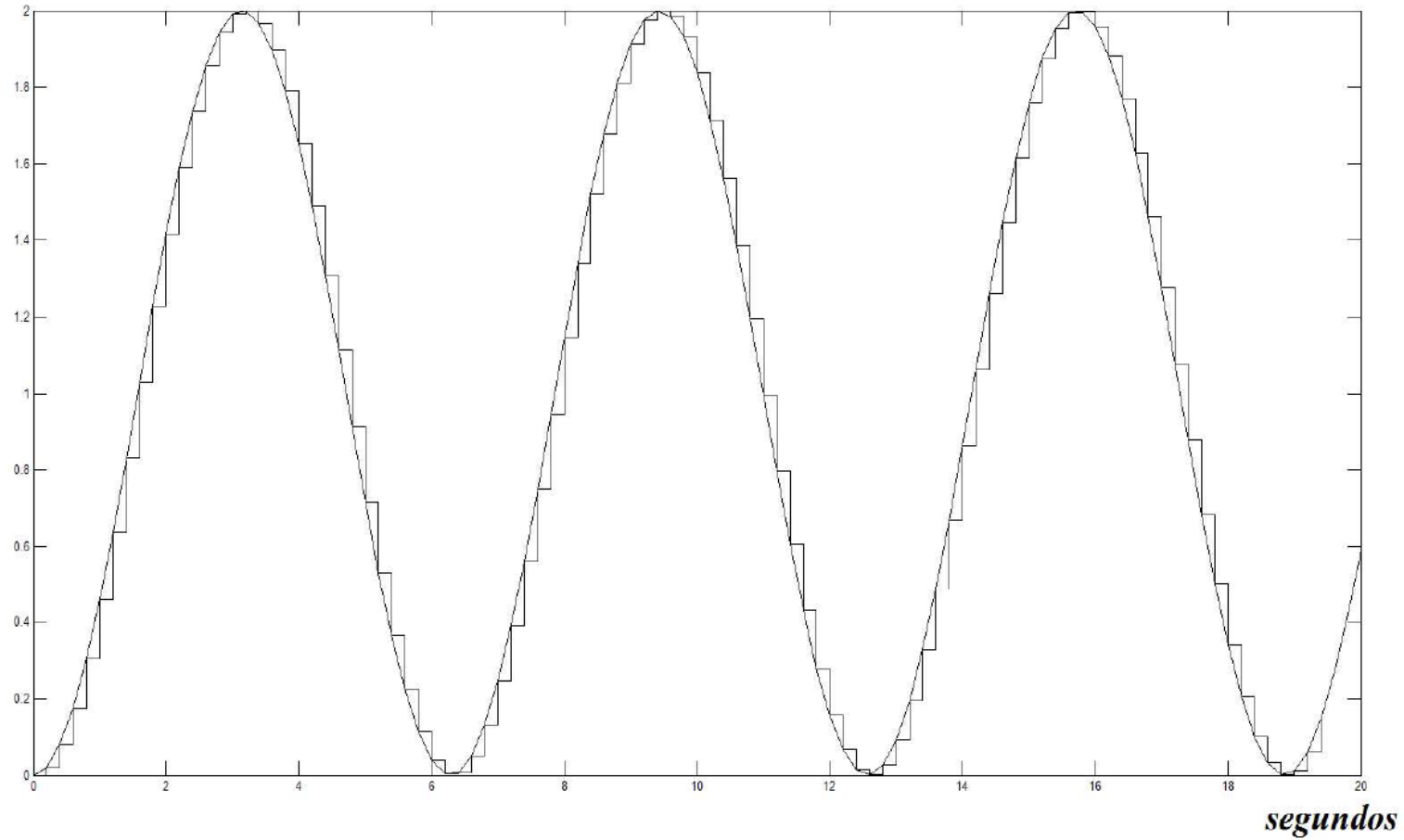


Figura 9.5 Señal discreta con retenedor de orden cero.

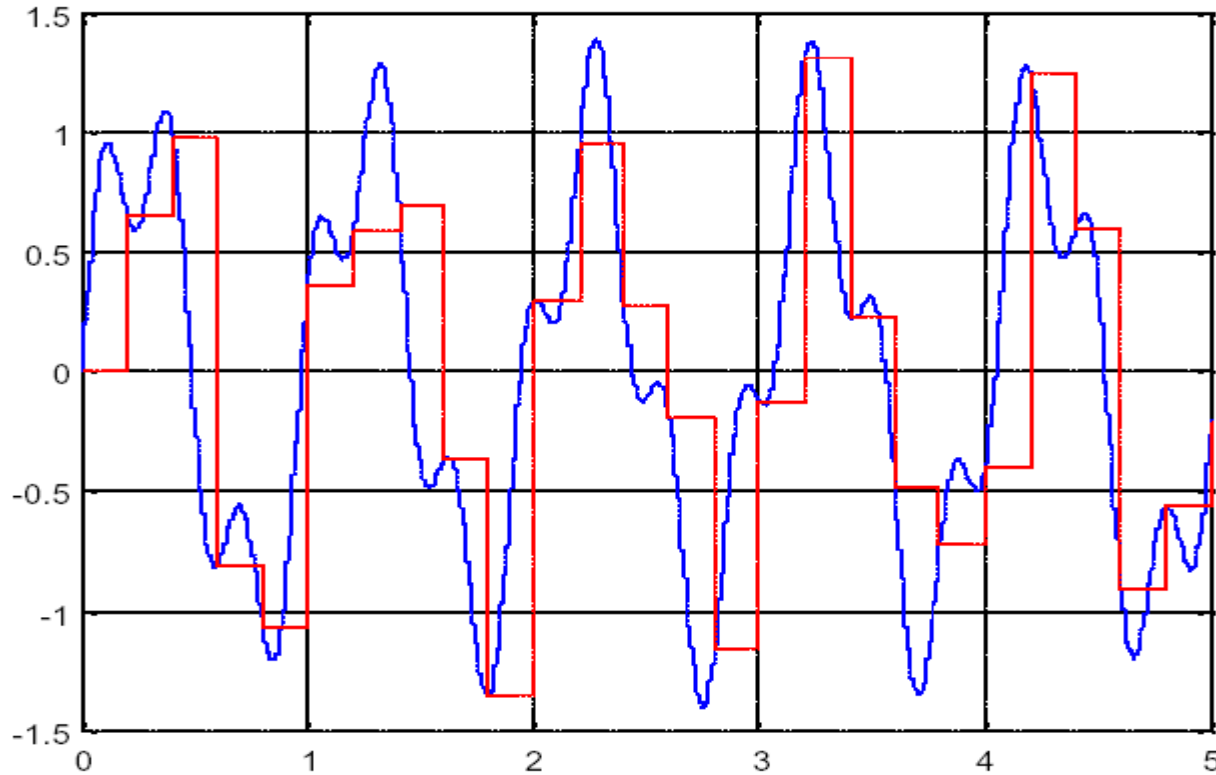


Ilustración 1-14 Reconstrucción con Bloqueador de Orden Cero

MANTENEDORES DE ORDEN SUPERIOR

El mantenedor de orden cero puede considerarse como una extrapolación que utiliza un polinomio de grado cero. Para funciones suaves, pueden obtenerse errores de reconstrucción más pequeños extrapolando con polinomios de orden superior. Una extrapolación polinómica casual de primer orden da

$$f(t) = f(t_k) - \frac{t - t_k}{t_k - t_{k-1}} [f(t_k) - f(t_{k-1})], \quad t_k \leq t < t_{k+1}$$

La reconstrucción se obtiene dibujando una línea recta entre las muestras más cercanas. El mantenedor de primer orden se ilustra en la figura 2.3.

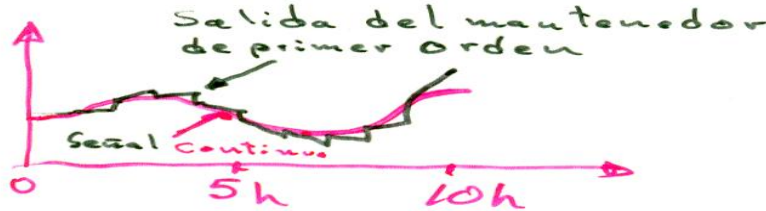


Figura 2.3 Muestreo y reconstrucción con un mantenedor de primer orden de una señal continua.

El mayor error que se obtiene cuando se utiliza un mantenedor de primer orden viene dado por

$$e_{FOH} = \max_k \max_t \left| f(t) - f(t_k) - \frac{t - t_k}{t_k - t_{k-1}} [f(t_k) - f(t_{k-1})] \right|$$

Para muestreo periódico de señales con derivada segunda suave, el error se puede estimar por



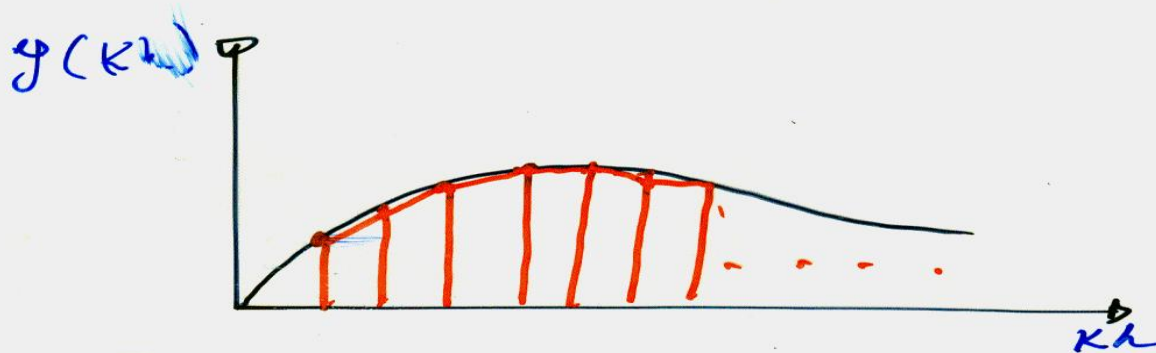
$$e_{\text{FOH}} \leq h^2 \max_t |f''(t)|$$

Para señales con derivadas suaves de mayor orden superior pueden realizarse una extrapolación polifónica de orden superior. Estos tipos de reconstrucciones no son muy comunes en la práctica debido a su facultad de realización.

Retenedores de Orden Superior

Para funciones suaves, es posible obtener errores de reconstrucción pequeños usando polinomios de extrapolación de orden superior. Por ejemplo para un polinomio de primer orden causal

$$f(t) = f(t_k) + \frac{t - t_k}{t_k - t_{k-1}} [f(t_k) - f(t_{k-1})], \quad t_k \leq t < t_{k+1}$$



El error más grande está dado como:

$$e_{FOH} = \max_k \max_t \left\{ f(t) - f(t_k) - \frac{t - t_k}{t_k - t_{k-1}} [f(t_k) - f(t_{k-1})] \right\}$$

- Para muestreo periódico de señales con segunda derivada suave el error de estimación está acotado de la siguiente manera:

$$e_{FOH} \leq h^2 \max_t |f''(t)|$$

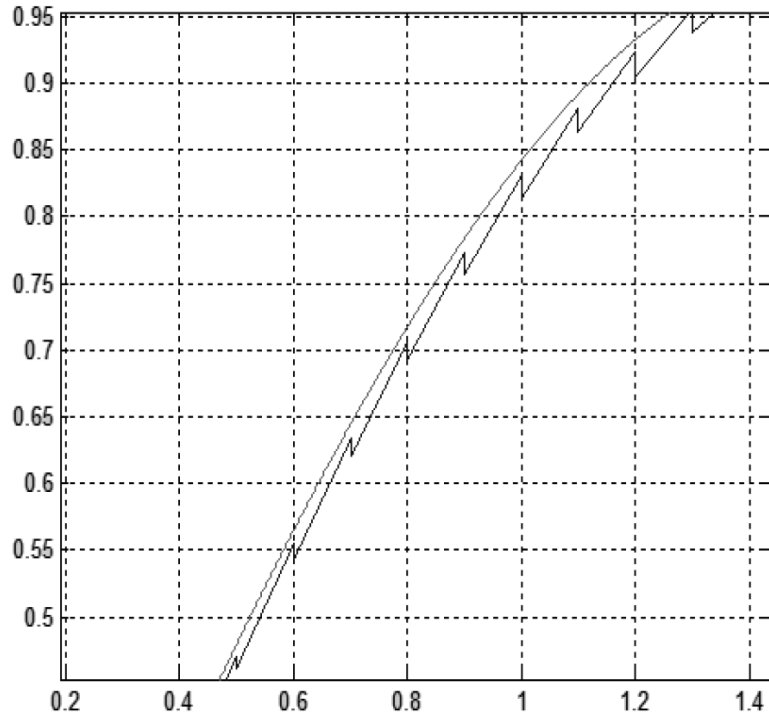


Figura 9.7 Señal muestreada con un retenedor de primer orden.

Tiempo discreto t_k

La señal continua $y(t)$ está definida para $\forall t$, siendo el tiempo cualquier número real: $t \in \mathbb{R}$ $t = 0, 0.1, 0.45, 0.66, \dots, 5.6789, 6.122, \dots$. Por otro lado, el tiempo discreto t_k no puede tomar cualquier valor, únicamente múltiplos del periodo de muestreo h , de ahí su nombre tiempo discreto: $t_k = kh = \{h, 2h, 3h, 4h, \dots, nh\}$, por ejemplo para un conjunto de n muestras, considerando $h = 1$ mseg., se tiene: $t_k = \{\underbrace{0.001}_{t_1}, \underbrace{0.002}_{t_2}, \underbrace{0.003}_{t_3}, \underbrace{0.004}_{t_4}, \dots, \underbrace{5}_{t_{n-2}}, \underbrace{5.001}_{t_{n-1}}, \underbrace{5.002}_{t_n}\}$; valores como $t_k = 0.023$ o $t_k = 1.11$ no puede tomar el tiempo discreto t_k para ningún $k \in \mathbb{N}$, puesto que no representan múltiplos del periodo de muestreo.

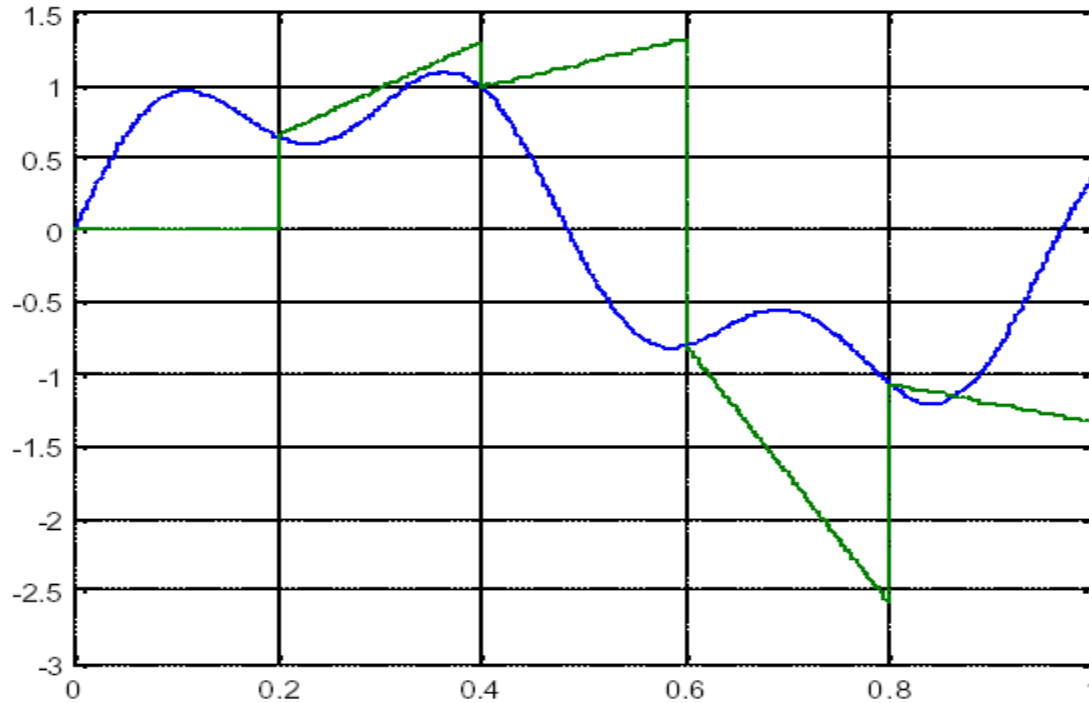


Ilustración 1-15 Bloqueador de Orden Uno

Circuito Muestreador y mantenedor (S / H).

Un convertidor A / D también puede describirse como un muestreador y mantenedor. El circuito mantenedor conserva constante el voltaje analógico durante la conversión a una representación digital. En primer lugar, desarrollaremos un modelo detallado para el circuito muestreador y mantenedor.

Un modelo del muestreador y mantenedor

La figura 4.3 muestra el esquema de un circuito analógico muestreador y mantenedor. Se supone que el circuito continúa con un amplificador de muy alta impedancia de entrada. El circuito funciona del siguiente modo: cuando se cierra el interruptor de muestreo, el condensador se carga al voltaje de entrada a través de la resistencia R. Cuando el interruptor de muestreo se abre, el condensador mantiene su voltaje hasta el siguiente periodo de muestreo.

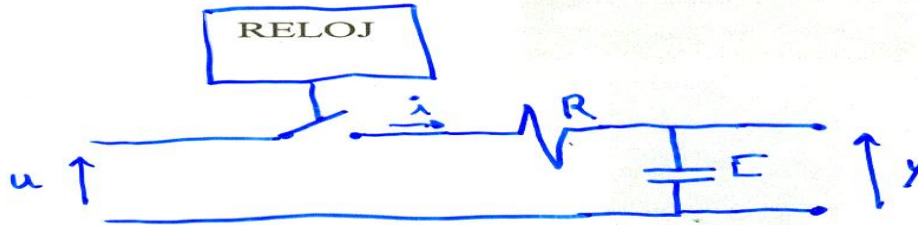


figura 4.3 Diagrama esquemático de un circuito muestreador y mantenedor.

Para describir el sistema se introduce una función m , la cual describe la apertura y cierre del interruptor de muestreo. Esta función está definida por

$$m(t) = \begin{cases} 1 & \text{si el interruptor está abierto} \\ 0 & \text{si el interruptor está cerrado} \end{cases}$$

Entonces, la corriente viene dada por

$$i = \frac{u - y}{R} m$$



CUVyTT-BUAP



De este modo, la corriente se modula por la función m , la cual se denomina función de modulación. Si la impedancia de entrada del circuito que sigue al muestreador y mantenedor es grande, el voltaje del condensador está definido por

$$C \frac{dy(t)}{dt} = i(t) = \frac{u(t) - y(t)}{R} m(t) \quad (4.1)$$

La ecuación diferencial (4.1) es un sistema lineal que varía con el tiempo. La variación en el tiempo está causada por la modulación. Si el periodo de muestreo es h y el interruptor permanece cerrado en τ segundos en cada muestreo, la función m tiene la forma mostrada de la figura 4.4 Como m es una función periódica, el sistema resulta ser un sistema periódico.

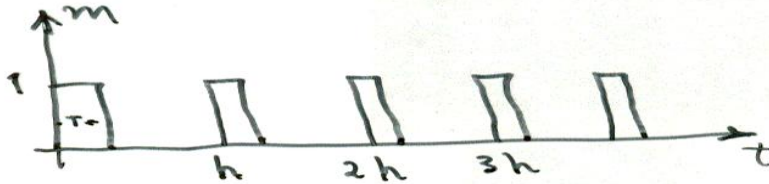


Figura 4.4. Grafico de la función de modulación m con periodo h y anchura de pulso τ .

Una vez que se ha obtenido el modelo matemático del circuito, puede estudiarse su respuesta a una señal de entrada u . De la ecuación (4.1) se obtiene que el voltaje del condensador es constante cuando el interruptor está abierto. Cuando el interruptor se cierra, el voltaje alcanza la señal de entrada u como un sistema dinámico de primer orden con una constante de tiempo RC . La constante de tiempo RC del circuito debe ser considerablemente menor que la anchura del pulso; de otro modo, el condensador no dispone de tiempo para cargarse al voltaje de entrada cuando el circuito se cierra.

La figura 4.5 muestra los resultados de una simulación del circuito muestreador y mantenedor. Con los parámetros elegidos, la anchura del pulso τ es tal que la señal de entrada cambia significativamente cuando se cierra el interruptor.

Mustreadores prácticos

En la práctica, un muestreador no se realiza como se representa en la figura 4.3. Una realización común se muestra en la figura 4.7. Para mantener la carga en el condensador se utiliza un amplificador operacional. El circuito de la figura 4.7 también se puede describir por la ecuación

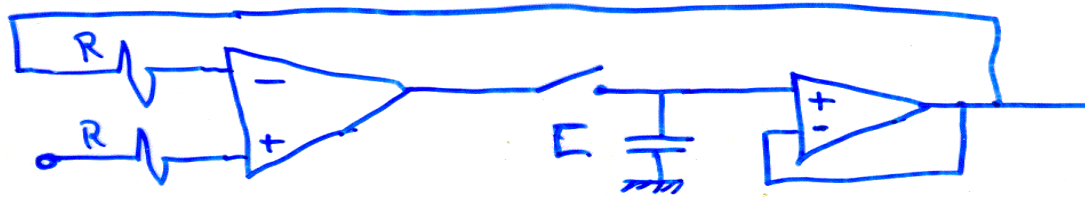


Figura 4.7 Una realización práctica de un circuito muestreador y mantenedor.

Para evitar problemas de ruidos electricos y derivaciones a masa es importante aislar galvánicamente a la computadora de las señales del proceso. Esto puede llevarse a cabo con el circuito de la figura 4.8 el cual se denomina circuito de condensador

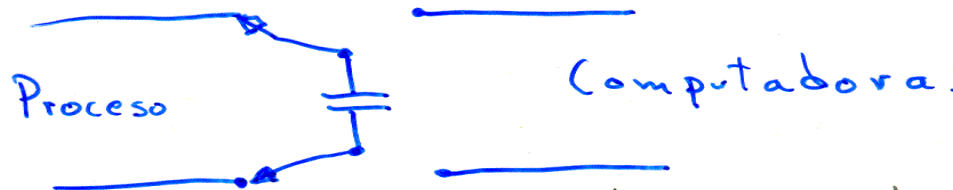


Figura 4.8 Circuito muestreador y mantenedor basado en la realización del condensador flotante. El circuito combina el aislamiento eléctrico con la acción



de muestrear y mantener. El condensador se carga al voltaje de entrada cuando éste se conecta a la línea de entrada. Cuando el condensador se conecta al convertidor D/A mantiene su voltaje. El aislamiento eléctrico se obtiene porque el condensador está conectado al proceso o al convertidor D/A de la computadora de control. En la práctica es muy común cargar el condensador a través de un amplificador operacional.

Una idealización matemática

El esquema de modulación de pulso es fácil de simular pero difícil de analizar. Por tanto, introduciremos una idealización matemática más fácil de utilizar. Parece razonable diseñar el circuito muestreador y mantenedor para que la anchura del pulso t sea menor que el periodo de muestreo. También parece razonable elegir la constante de tiempo RC más corta que la anchura del pulso. Entonces, la corriente a través del condensador estará formada por pulsos cortos. Tanto la altura como la integral en el tiempo de un pulso es proporcional a la diferencia $u - y$ entre el voltaje de entrada u y el voltaje del condensador y en los instantes de muestreo

En la idealización, los pulsos de corriente se reemplazan por impulsos. Por simplicidad, la integral de los impulsos se elige proporcional a la señal de entrada u en los instantes de muestreo. Entonces, el condensador se reemplaza por un integrador. Ya que los pulsos se eligieron proporcionales a u y no a $u - y$, es necesario anular la integral cada vez que llega un nuevo pulso. Por consiguiente, la corriente se representa por

$$u^* = um \quad (4.2)$$

donde

$$m(t) = \sum_{k=-\infty}^{\infty} \delta(t - kh) \quad (4.3)$$

y es la función delta δ (comparese con (4.1)). La señal u^* se denomina representación muestreada de la señal continua u . Resulta útil recordar que u^* está relacionada con la corriente a través del condensador del circuito muestreador y mantenedor de la figura 4.3

La señal u^* puede considerarse como una modulación de u por una señal portadora de la forma de un tren de impulsos. El modelo, por lo tanto, se denomina modelo de modulación por tren de impulsos. La señal u^* se introduce para representar una señal muestreada de modo que pueda procesarse con un filtro lineal



$$i = \frac{u - y}{R} m \quad [1.2]$$

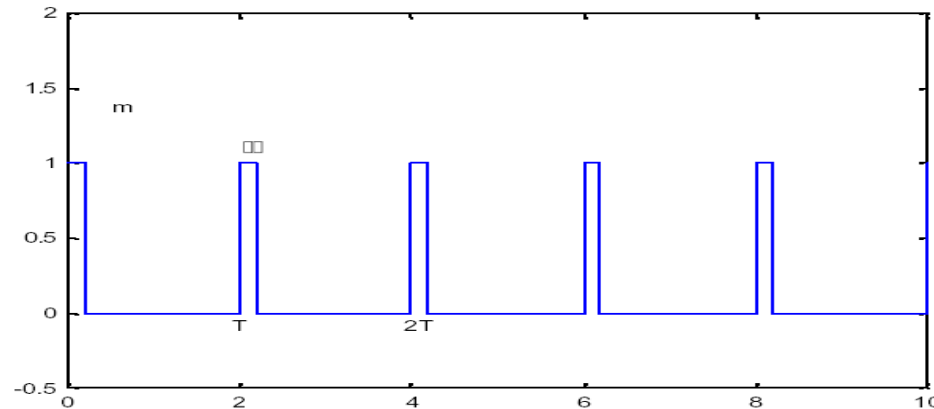
O sea es modulada por m lo que es llamada *función de modulación*. Si la impedancia del circuito que sigue al de muestreo y retención es la alta tensión en el condensador es

Si la impedancia del circuito siguiente es grande, la tensión en el condensador es

$$C \frac{dy}{dt} = i(t) = \frac{u(t) - y(t)}{R} m(t) \quad [1.3]$$

La ecuación diferencial [1.3] es un sistema variante en el tiempo. Esta variación es causada por la modulación. Si el período de muestreo T es constante y si el contacto está cerrado τ segundos en cada muestreo, la función m tiene la forma de la figura siguiente. Como m es periódica, el sistema también lo será

Es un sistema lineal variante en el tiempo



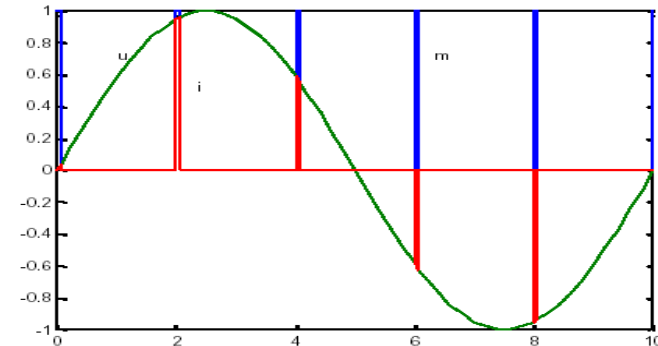
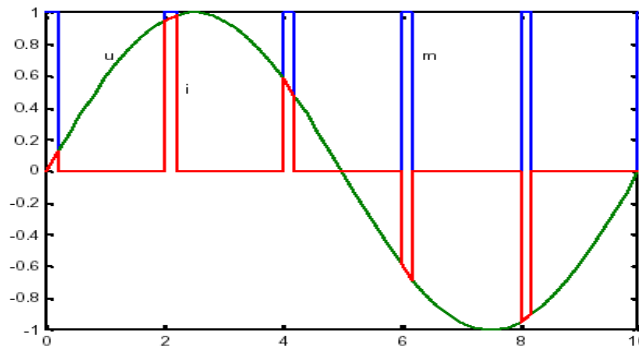


Con la constante de tiempo $RC=0,01s$, el ancho del pulso $t = 0,2s$ es lo suficientemente grande como para apreciar variaciones de u cuando el contacto está cerrado.

En la segunda experiencia se eligió $t = 0,05s$ y se observa que es una buena elección ya que el circuito llega rápidamente a su valor e inmediatamente permanece constante hasta el siguiente período.

- Cuando $m = 0$, la tensión en C es constante
- Cuando $m = 1$, es un sistema de primer orden con constante de tiempo RC . La salida tiende a la entrada.
- RC debe ser $\ll \tau$

1.3.3. Simulación





CUV_yTT-BUAP



El esquema de modulación de pulso es fácil de simular pero difícil de analizar. Introduciremos una idealización matemática para hacerlo más sencillo. Es razonable diseñar un muestreador con t mucho más corto que el período de muestreo y su constante RC menor que t . La corriente por el condensador será un corto pulso. La altura y la integral en el tiempo de un pulso son proporcionales a la diferencia entre u e y en el instante de muestreo.

- $RC \ll \tau$
- $\tau \ll T$
- i tiende a un impulso de amplitud proporcional a la diferencia entre u e y .
- Se reemplaza el pulso por un impulso
- Es casi un integrador
- Hay que inicializar a cero la integral cuando llega el nuevo pulso



El circuito mantenedor

El circuito mantenedor puede representarse como un integrador que se anula automáticamente después de un periodo de muestreo. Tal sistema tiene la función de transferencia

$$G(s) = \frac{1}{s} (1 - e^{-sh}) \qquad G(s) = \frac{1}{s} (1 - e^{-sT})$$

La respuesta a un impulso de la función de transferencia $1/s$ es un escalón unidad y la respuesta a un impulso de $(1/s)\exp(-sh)$ es un escalón unidad retardado h unidades de tiempo. La diferencia de ambas respuestas da la respuesta a un pulso de altura unidad y duración h .

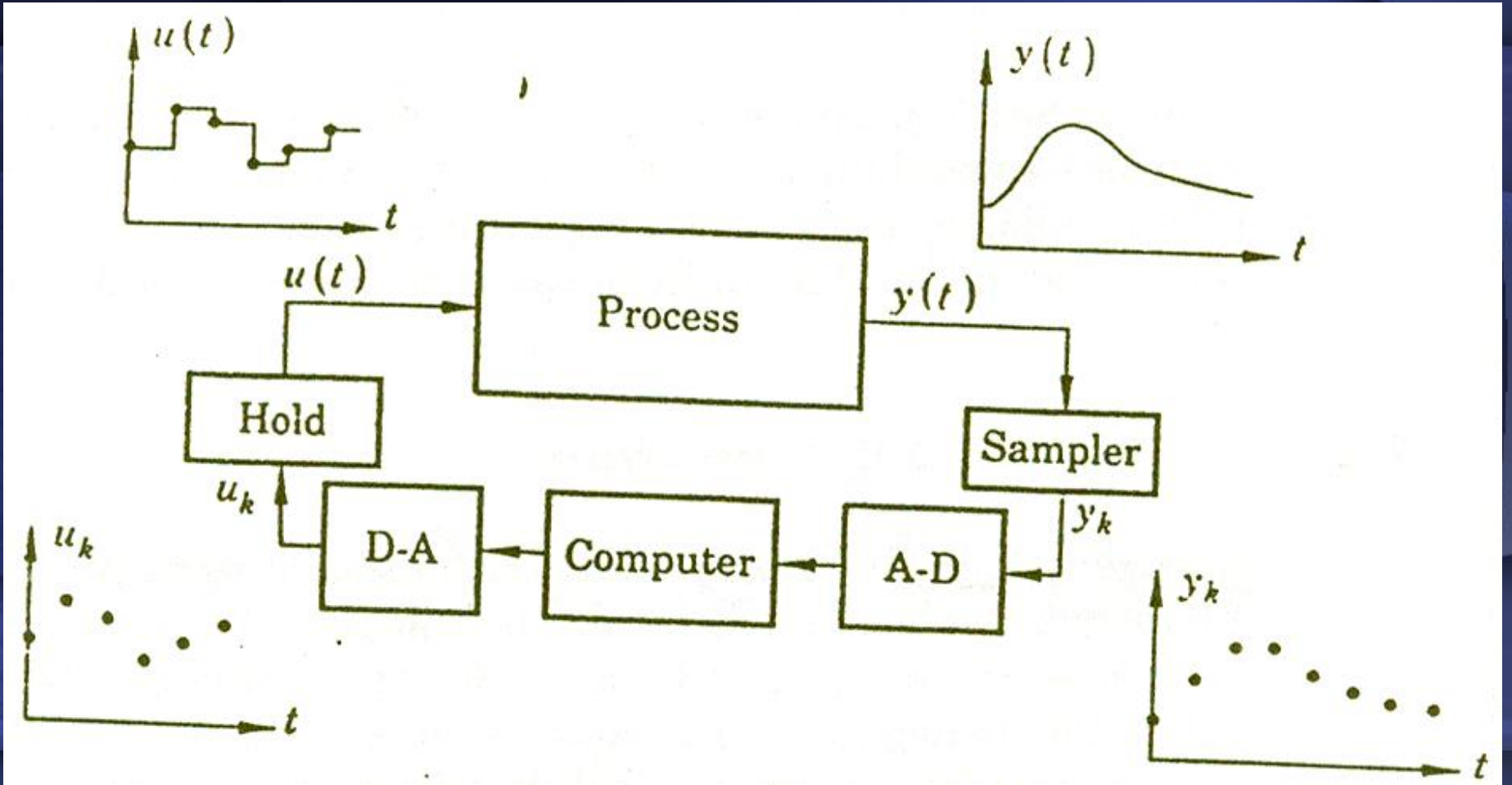
Obsérvese que la ganancia en estado estacionario del circuito mantenedor es $G(0) = h$. El muestreo ideal, tiene una ganancia $1/h$. Por tanto, la combinación de un muestreador y un mantenedor deben tener una ganancia unidad en estado estacionario. De este modo, para muestreos muy rápidos, el circuito muestreador y mantenedor actúa como un sistema continuo con función de transferencia unidad.

En este caso, el modelo idealizado para el circuito muestreador y mantenedor se obtiene combinando un muestreador con modulación de impulso definido (4.2) y (4.3) con un circuito mantenedor definido por (4.4). Una representación en diagramas de bloques para este sistema se muestra en la figura 4.9. Ya que el modulador de impulsos es un sistema periódico, se deduce que el muestreador y mantenedor también es un sistema periódico.





CUV_yTT-BUAP





Modelos matemáticos orientados a la computadora: Sistemas Discretos

Objetivo:

Proporcionar descripciones matemáticas de los sistemas controlados por computadora desde el punto de vista de la computadora. Desarrollar los conceptos y métodos necesarios para trabajar con los sistemas discretos.



3.1 Introducción

En esta unidad se introducen los modelos matemáticos para los sistemas controlados por computadora. Se estudia el sistema tal como lo ve la computadora. La computadora recibe medidas desde el proceso en instantes discretos y transmite las nuevas señales de control en instantes discretos. El objetivo, entonces, es describir el cambio en las señales de un muestreo a otro y no tener en cuenta el comportamiento entre muestreos. Debe destacarse que los modelos matemáticos orientados a la computadora dan el comportamiento solamente en los puntos de muestreo; el proceso físico sigue siendo un sistema continuo. Sin embargo, enfocar



Un punto que hay que tratar con algún cuidado es la dependencia del tiempo de un sistema muestreado. En la presente unidad se alude el problema de la dependencia del tiempo estudiando las señales en unos instantes de tiempo que están sincronizados con el reloj de la computadora. Esto es lo que a veces se llama *modelo estroboscópico*, ya que solo se obtienen observaciones intermitentes de las señales. Esto produce modelos descritos por ecuaciones en diferencias en las formas del espacio de estados y de entrada-salida. En las secciones siguientes trata el problema de encontrar la representación discreta de un modelo continuo dado en el espacio de estados utilizando un mantenedor de orden cero. También se



dato. Las secciones estudian la transformación de los modelos en el espacio de estados y la conexión entre los modelos en el espacio de estados y la conexión entre los modelos en el espacio de estados y de entrada-salida. Para escribir los modelos de entrada salida se emplean operadores de desplazamiento. El calculo con el operador de desplazamiento es equivalente al empleo de operadores diferenciales en los sistemas continuos. El equivalente discreto de la transformada de Laplace es la transformada z , que se expone.

3.2 Muestreo de un sistema continuo en el espacio de estados

Un problema fundamental es como describir un sistema continuo que esta conectado a una computadora por convertidores A/D y D/A. Consideremos el sistema de la figura 3.1. Las señales en la computadora son la secuencias $\{u(t_k)\}$ y $\{y(t_k)\}$.

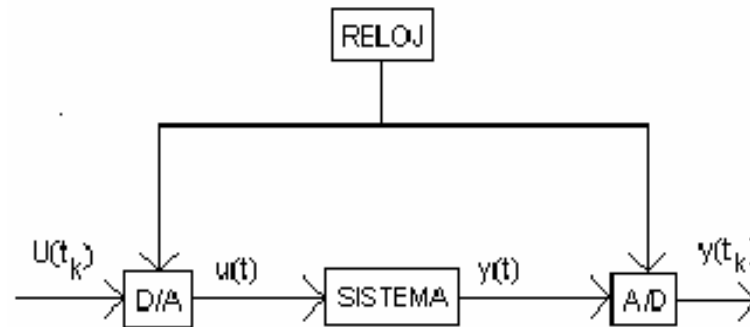


Figura 3.1 Diagrama de bloques de un sistema continuo conectados a unos convertidores A/D y D/A.

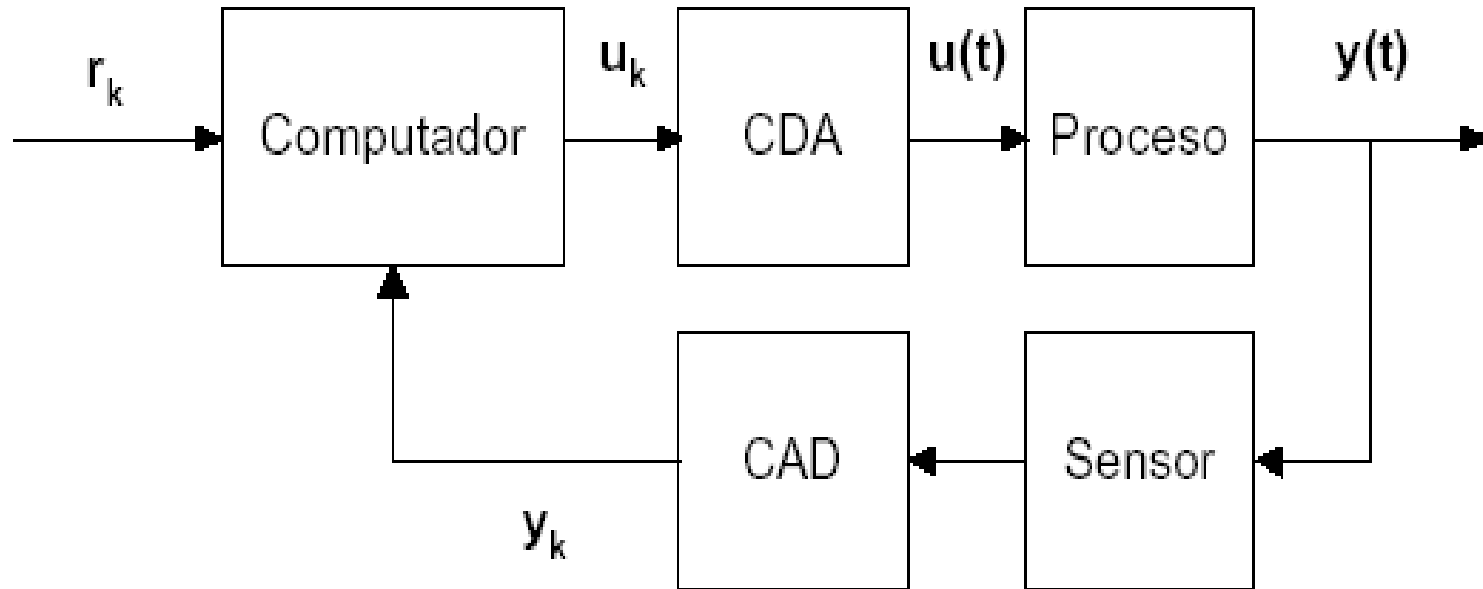
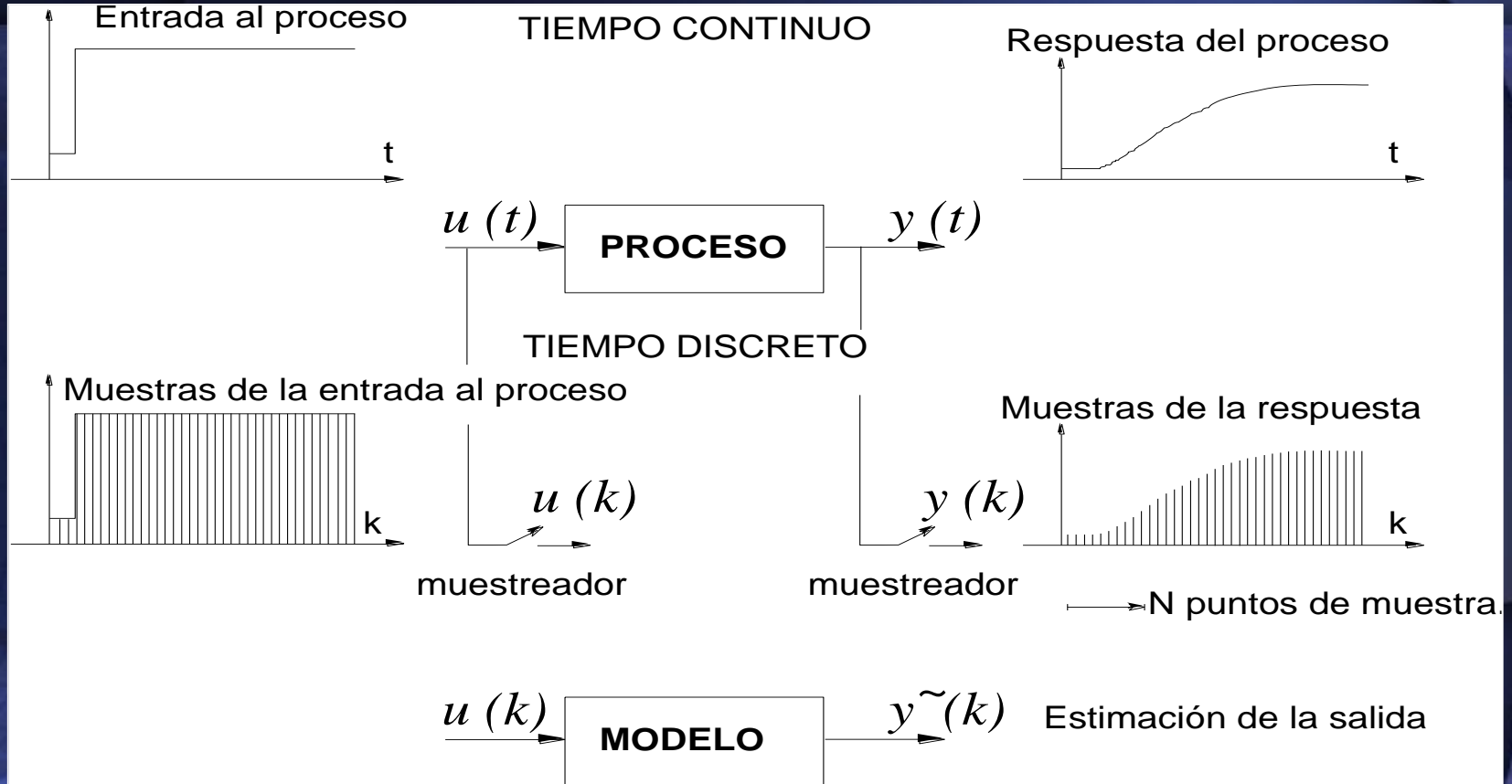


Ilustración 1-1 Lazo típico de Control Digital



CUVyTT-BUAP



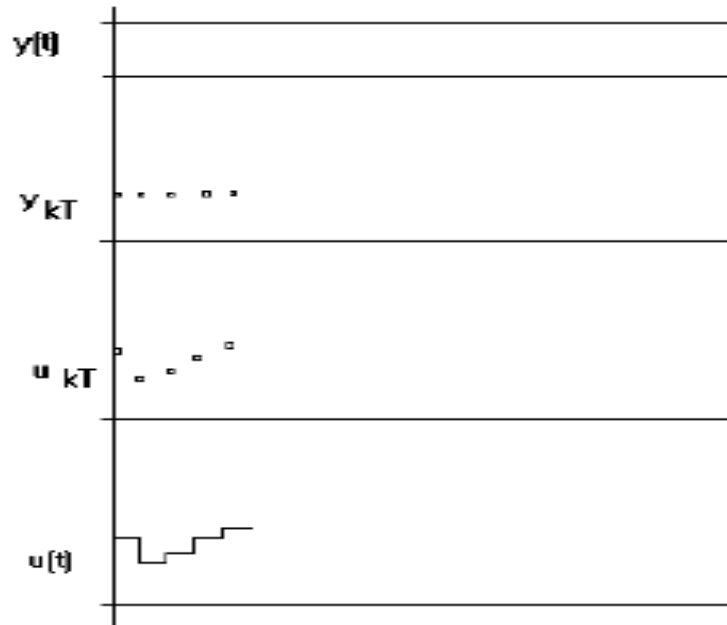


Ilustración 1-4 Muestreo de una señal continua



1.2. Secuencias

En síntesis lo que ve el computador no es más que una *secuencia* de números. Es necesario entonces recordar algunas de las operaciones básicas entre secuencias.

La forma de escribir una secuencia es la siguiente:

$$\{u_k\} = \{\dots u_{-3}, u_{-2}, u_{-1}, u_0, u_1, \dots\} \quad [1.1]$$

Se verán a continuación algunas operaciones, por ejemplo una suma de secuencias está dada por la siguiente ecuación

$$\{u_k\} = \{x_k + v_k\} \quad [1.2]$$

y el producto por una constante,

$$\{y_k\} = \{m x_k\} \quad [1.3]$$

Dos secuencias muy usuales en control son el *impulso* y el *escalón* que tienen la forma siguiente:

$$\{\delta_k\} = \{1, 0, 0, \dots\} \quad [1.4]$$

$$\{1_k\} = \{1, 1, 1, \dots\}$$

Su representación gráfica está dada en la Ilustración 1-5.

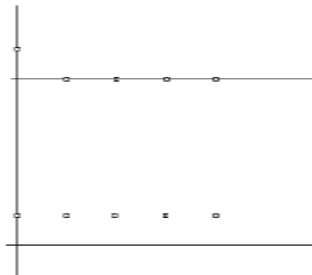


Ilustración 1-5 Secuencias Impulso y Escalón



Ilustración 1-6 Sistema Discreto

Un ejemplo de sistema discreto es un *Sumador* de modo que la secuencia de salida sea la suma de los valores de entrada hasta ese instante:

$$\{y_k\} = \left\{ \sum_{i=1}^k u_i \right\} \quad [1.5]$$

Del mismo modo un *Promediador* puede escribirse como,

$$\{y_k\} = \left\{ \frac{1}{3} (u_{k-1} + u_k + u_{k+1}) \right\} \quad [1.6]$$



El problema clave consiste en encontrar la relación entre estas dos secuencias. Encontrar el equivalente discreto de un sistema continuo es lo que se llama *muestreo de un sistema continuo*. El modelo obtenido se llama también modelo estroboscópico porque da la relación entre las variables del sistema sólo en los instantes de muestreo. Para obtener las descripciones deseadas es necesario describir los convertidores y el sistema.



EL modelo de espacio de estados en tiempo continuo tiene la siguiente estructura:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad (9.1)$$

$$y(t) = \mathbf{c}^T \mathbf{x}(t) \quad (9.2)$$

donde $\mathbf{x} \in \mathbb{R}^n$ es la variable de estado y representa la solución de la ecuación diferencial de primer orden, $\dot{\mathbf{x}} \in \mathbb{R}^n$ es la variación temporal de la variable de estado, los parámetros del sistema son: $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^n$, la señal de entrada es $u(t)$ y la respuesta o salida del sistema está expresada por $y(t)$.



Muestreo de un sistema con mantenedor de orden cero

Una situación corriente en control con computadora es que el convertidor D/A está construido de tal forma que mantiene constante la señal analógica hasta que se ordena una nueva conversión. Por tanto, es natural tomar como instante de muestreo, t_k , os momentos en que cambia el control. Como la señal de control es discontinua, es preciso especificar su comportamiento durante las discontinuidades. El convenio adoptado es que la señal es continua por la derecha. La señal de control se representa entonces por la señal muestreada $\{u(t_k); k=..., -1, 0, 1, \dots\}$.



La solución analítica de la ecuación (9.1) está dada de la siguiente forma (ver sección 8.3):

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{A(t-\sigma)}Bu(\sigma)d\sigma \quad (9.3)$$

el proceso para obtener un sistema discreto del modelo de espacio de estados consiste en muestrear la ecuación (9.3) tomando la siguiente forma:

$$\mathbf{x}(t_k) = e^{A(t_k-t_{k-1})}\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k-\sigma)}Bu(\sigma)d\sigma$$
$$\mathbf{x}(t_k) = e^{A\Delta t_k}\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A\Delta\sigma}Bu(\sigma)d\sigma \quad (9.4)$$

$$(9.5)$$

donde se ha considerado que el tiempo inicial t_0 representa el tiempo de muestreo anterior t_{k-1} y t el tiempo de muestreo actual t_k y $\Delta t_k = t_k - t_{k-1}$, $\Delta\sigma = t_k - \sigma$.



Suponga que el muestreo se realiza en forma periódica, por lo que el intervalo entre dos muestras consecutivas es h , es decir $t_k - t_{k-1} = h$, entonces el intervalo de integración de t_{k-1} a t_k equivale a integrar sobre el intervalo $[0, h]$, esto conlleva a: $\Delta t_k = t_k - t_{k-1} = h$, $\Delta \sigma = t_k - \sigma = h$, la ecuación (9.4) obtiene la siguiente expresión:

$$\mathbf{x}(t_k) = e^{Ah} \mathbf{x}(t_{k-1}) + \int_0^h e^{A\nu} B u(\nu) d\nu \quad (9.6)$$

se ha introducido una nueva variable al interior de la integral (variable muda ν), debido que $\Delta \sigma = t_k - \sigma = h$, entonces la introducción de h en la matriz exponencial admite una nueva representación por medio de ν .



Considere que en el muestreo se emplea un retenedor de orden cero, la representación de escalones en cada periodo de muestreo hace que la señal $u(t_{k-1})$ sea constante en cada intervalo de integración $[0, h]$, por lo que se puede obtener una mejor simplificación matemática:

$$\mathbf{x}(t_k) = e^{Ah} \mathbf{x}(t_{k-1}) + \int_0^h e^{A\nu} B d\nu u(t_{k-1}) \quad (9.7)$$

$$\mathbf{x}(t_k) = \Phi \mathbf{x}(t_{k-1}) + \Gamma u(t_{k-1}) \quad (9.8)$$

$$\Phi = e^{Ah} \quad (9.9)$$

$$\Gamma = \int_0^h e^{A\nu} B d\nu \quad (9.10)$$

al retirar de la integral la señal de entrada u queda en función del tiempo de muestreo anterior $u(t_{k-1})$ debido a que el límite inferior de la integral inicia en cero y representa t_{k-1} .



Para un muestreo período h ,

$$t_k = k \cdot h$$

el modelo dado en (3.3) se simplifica al sistema invariante

$$\begin{aligned} X(kh + h) &= \Phi X(kh) + \Gamma u(kh) \\ y(kh) &= Cx(kh) + Du(kh) \end{aligned} \quad (3.6)$$

donde

$$\begin{aligned} \Phi &= e^{A h} \\ \Gamma &= \int_0^h e^{A s} ds B \end{aligned} \quad (3.7)$$



La conversión de un modelo es espacio de estados en tiempo continuo a tiempo discreto, considerando un retenedor de orden cero se realiza bajo el siguiente procedimiento:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B u(t) \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) \end{aligned} \quad \Rightarrow \quad \begin{aligned} \mathbf{x}(t_k) &= \Phi \mathbf{x}(t_{k-1}) + B u(t_{k-1}) \\ y(t_k) &= \mathbf{c}^T \mathbf{x}(t_k) \\ \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{A\sigma} B d\sigma \end{aligned}$$

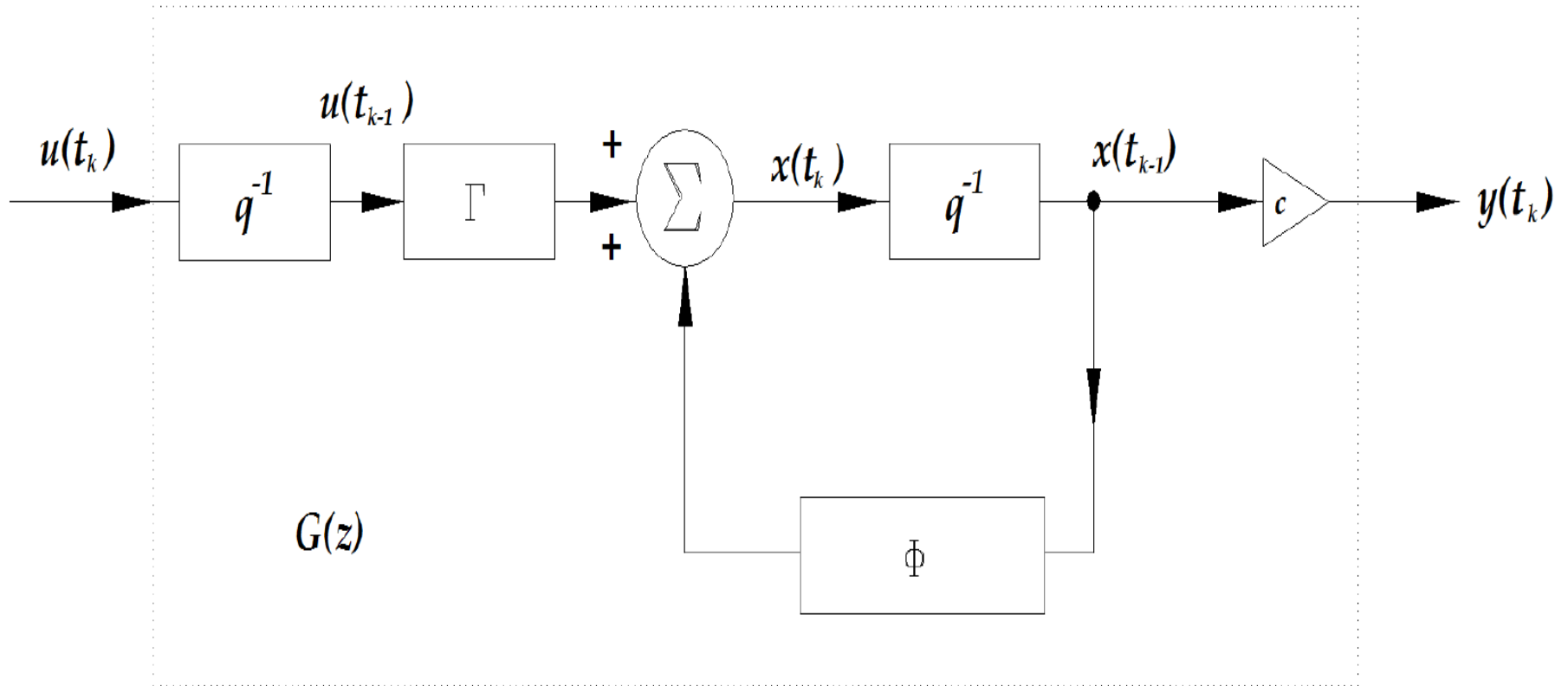




Figura 9.9 Diagrama a bloques del modelo en espacio de estados discreto.

En general dependiendo del exponente n en $q^{\pm n}$ se aplican corrimientos a la izquierda o derecha bajo las siguientes reglas sobre el operador escalar q :

 $q^{-1}x(t_k) = x(t_{k-1}), q^{-2}x(t_k) = x(t_{k-2}), \dots q^{-n}x(t_k) = x(t_{k-n})$

 $q^1x(t_k) = x(t_{k+1}), q^2x(t_k) = x(t_{k+2}), \dots q^nx(t_k) = x(t_{k+n})$

El operador de corrimiento q permite encontrar la relación para una función de transferencia racional expresada directamente con corrimientos a la derecha o izquierda. Por ejemplo:

$$\mathbf{x}(t_k) = \Phi \mathbf{x}(t_{k-1}) + \Gamma u(t_{k-1}) \quad \Rightarrow \quad \mathbf{x}(t_k) = \Phi q^{-1} \mathbf{x}(t_k) + \Gamma q^{-1} u(t_k) \Rightarrow [\Phi - q^{-1}I] \mathbf{x}(t_k) = q^{-1} \Gamma u(t_k)$$

$$\mathbf{x}(t_k) = [I - q^{-1}\Phi]^{-1} q^{-1} \Gamma u(t_k)$$

$$y(t_k) = \mathbf{c}^T \mathbf{x}(t_k) = \mathbf{c}^T [I - q^{-1}\Phi]^{-1} q^{-1} \Gamma u(t_k)$$

La función de transferencia que relaciona $y(t_k)$ con $u(t_k)$ se le denomina operador de transferencia pulso:

$$G(q) = \mathbf{c}^T \underbrace{[I - q^{-1}\Phi]^{-1} q^{-1}}_{\text{Corrimiento a la izquierda}} \Gamma = \mathbf{c}^T \underbrace{[qI - \Phi]^{-1}}_{\text{Corrimiento a la derecha}} \Gamma \quad (9.12)$$



¿Cómo calcular Φ y Γ ?

Los cálculos requeridos para muestrear un sistema continuo son la evaluación de una matriz exponencial y la integración de una matriz exponencial. Esto se puede hacer de muchas formas, por ejemplo, utilizando las siguientes técnicas:

Desarrollo en serie de la matriz exponencial

$$e^{At} = (sI - A)^{-1}$$

La transformada de Laplace- La transformada de Laplace de $\exp(At)$ es $(sI-A)^{-1}$.

El teorema de Cayley Hamilton.

Transformación a la forma de Jordan



CUVγTT-BUAP



Los cálculos a mano son factible para sistemas de bajo orden, $n \leq 2$, y para orden de sistema superior con estructuras especiales. Una forma de simplificar los cálculos es calcular

$$\Psi = \int_0^h e^{As} ds = Ih + \frac{Ah^2}{2!} + \frac{A^2h^3}{3!} + \dots + \frac{A^i h^{i+1}}{(i-1)!} + \dots \quad (3.8)$$

$$\Psi = \int_0^T e^{A\tau} d\tau = IT + \frac{AT^2}{2!} + \dots + \frac{A^i T^{i+1}}{(i+1)!}$$

Las matrices Φ , γ , Γ vienen dadas por

$$\Phi = I + A\Psi$$

$$\Gamma = \Psi B$$

(3.9)

La evaluación con computadora puede hacerse mediante diferentes algoritmos numéricos



- Teorema de Cayley-Hamilton

$$\det(\lambda I - A) = \Delta(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0$$

una matriz satisface su polinomio característico

$$A^n + a_{n-1}A^{n-1} + \dots + a_1A + a_0 = 0$$

o lo que es lo mismo

$$A^n = -a_{n-1}A^{n-1} - \dots - a_1A - a_0$$

$$A^n = f(I, A, A^2, \dots, A^{n-1})$$

pero si se multiplica por A también quedará un función

$$\begin{aligned} A^{n+1} &= -a_{n-1}A^n - \dots - a_1A^2 - a_0A \\ &= -a_{n-1}(-a_{n-1}A^{n-1} - \dots - a_1A - a_0) - \dots - a_1A^2 - a_0A \\ &= f(I, A, A^2, \dots, A^{n-1}) \end{aligned}$$



es decir toda potencia de la matriz original se puede expresar como una combinación lineal de $(I, A, A^2, \dots, A^{n-1})$ y por lo tanto cualquier polinomio $f(A)$.

También A satisface polinomios de grado menor a $n-1$. Esto se cumple cuando el polinomio característico tiene raíces múltiples. En este caso se define el *polinomio mínimo* que es de menor grado que el característico y depende del tamaño de los bloques de Jordan.

Todo polinomio puede ser expresado como una división de polinomios, de la forma

$$f(\lambda) = q(\lambda)\Delta(\lambda) + h(\lambda) \quad [1.16]$$

pero el primer término es nulo, con lo que queda, calculado en A , lo siguiente

$$f(A) = h(A) \quad [1.17]$$

este polinomio es de grado $n-1$. La ecuación [1.16] se cumple para todos los autovalores de A

En el caso del cálculo de e^{AT} se puede considerar como un polinomio de grado infinito y calcularlo en base al polinomio característico de A .



Ejemplo 1. Teorema Cayley-Hamilton

Se desea calcular e^{AT} con la matriz

$$A = \begin{bmatrix} 0 & 1 \\ -12 & 7 \end{bmatrix}$$

cuyo polinomio característico es

$$\Delta(\lambda) = |\lambda I - A| = \begin{vmatrix} \lambda & -1 \\ 12 & \lambda - 7 \end{vmatrix} = \lambda^2 - 7\lambda + 12 = (\lambda - 4)(\lambda - 3)$$

el polinomio a calcular es

$$f(\lambda) = e^{\lambda T}$$

el polinomio h tendrá orden 1 y será de la forma:

$$h(\lambda) = \alpha_1 \lambda + \alpha_0$$

y se cumple que

$$f(\lambda) = e^{\lambda T} = h(\lambda) = \alpha_1 \lambda + \alpha_0$$

que calculado para los autovalores de A resulta

$$\begin{cases} e^{4T} = \alpha_1 4 + \alpha_0 \\ e^{3T} = \alpha_1 3 + \alpha_0 \end{cases}$$

de donde se puede deducir el valor de las constantes

$$\begin{cases} \alpha_1 = e^{4T} - e^{3T} \\ \alpha_0 = -3e^{4T} + 4e^{3T} \end{cases}$$

reemplazando

$$e^{AT} = \alpha_0 I + \alpha_1 A = (-3e^{4T} + 4e^{3T}) I + (e^{4T} - e^{3T}) A$$

$$e^{AT} = \begin{bmatrix} -3e^{4T} + 4e^{3T} & e^{4T} - e^{3T} \\ -12(e^{4T} - e^{3T}) & 4e^{4T} - 3e^{3T} \end{bmatrix}$$



Ejemplo 3.1- Integrador doble

El integrador doble se describe por el sistema de ecuaciones

$$\begin{aligned}\dot{X} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= [1 \quad 0]x\end{aligned}\quad (3.10)$$

por tanto,

$$\begin{aligned}\Phi &= e^{Ah} = I + Ah + \frac{A^2 h^2}{2} + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & h \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \\ \Gamma &= \int_0^h \begin{bmatrix} s \\ 1 \end{bmatrix} ds = \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix}\end{aligned}$$

El modelo discreto de (3.10) es

$$\begin{aligned}X(kh + h) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(kh) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(kh) \\ y(kh) &= [1 \quad 0]x(kh)\end{aligned}\quad (3.11)$$



Ejemplo 3.2-Motor

Un modelo sencillo y normalizado de un motor eléctrico de corriente continua es el dado por

$$\dot{X} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} X + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$
$$y = [0 \quad 1]x$$

El método de la transformada de Laplace da

$$(sI - A)^{-1} = \begin{bmatrix} s+1 & 0 \\ -1 & s \end{bmatrix}^{-1} = \frac{1}{s(s+1)} \begin{bmatrix} s & 0 \\ 1 & s+1 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{(s+1)} & 0 \\ \frac{1}{s(s+1)} & \frac{1}{s} \end{bmatrix}$$

Por lo tanto

$$\Phi = e^{Ah} = \begin{bmatrix} e^{-h} & 0 \\ 1 - e^{-h} & 1 \end{bmatrix}$$

y

$$\Gamma = \int_0^h \begin{bmatrix} e^{-v} \\ 1 - e^{-v} \end{bmatrix} dv = \begin{bmatrix} 1 - e^{-h} \\ h - 1 + e^{-h} \end{bmatrix}$$



Solución de la ecuación del sistema

Los sistemas discretos invariantes se pueden describir mediante la ecuación en diferencias

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) \\y(k) &= C x(k)\end{aligned}\tag{3.12}$$

Por simplicidad se considera el periodo de muestreo como la unidad de tiempo $h=1$.

Supongamos que tanto $x(k_0)$ como las señales de entrada $u(k_0), u(k_0-1), \dots$, están dadas.



CUVγTT-BUAP



¿Cómo evolucionará entonces el estado? Es posible resolver (3.12) por simple iteración.

$$x(k_0+1) = \Phi x(k_0) + \Gamma u(k_0)$$

$$x(k_0+2) = \Phi x(k_0+1) + \Gamma u(k_0+1)$$

$$= \Phi^2 x(k_0) + \Phi \Gamma u(k_0) + \Gamma u(k_0+1)$$

·
·
·

$$x(k) = \Phi^{k-k_0} x(k_0) + \Phi^{k-k_0-1} \Gamma u(k_0) + \dots + \Gamma u(k_0+1)$$

$$= \Phi^{k-k_0} x(k_0) + \sum_{j=k_0}^{k-1} \Phi^{k-j-1} \Gamma u(j) \quad (3.13)$$



El inverso del muestreo



El muestreo define una aplicación de los sistemas continuos, como (3.1), sobre los sistemas discretos, como (3.6). Un ejemplo sencillo nos muestra que la aplicación no siempre puede invertirse.

Ejemplo 3.3

No existe ninguna ecuación diferencial de primer orden que, al muestrearla, nos dé la ecuación en diferencias

$$x(kh+h) = -0.5 x(kh) + u(kh)$$

porque la ecuación

$$e^{\alpha h} = -0.5$$

no tiene solución real, ya que la exponencial siempre es positiva.

El modelo de (3.6) es entonces más general que (3.1). Sin embargo, si la matriz ϕ no tiene valores propios en el eje real negativo, existe el correspondiente sistema continuo, que puede obtenerse resolviendo (3.7) y (3.9) para obtener A y B . De (3.7) se deduce que

$$A = \ln(\phi) / h = \ln(\phi^{1/h})$$

donde $\ln(\phi)$ es la matriz función logarítmica. Nótese que algunas veces $\ln(\phi)$ no es única. La ecuación (3.9) da

$$B = \Psi^{-1} \Gamma$$

Si el sistema no contiene ningún integrador, entonces $\Phi - I$ es no singular, y de (3.9) se deduce que

$$B = \Psi^{-1} \Gamma = B = (A\Psi)^{-1} A \Gamma = (A - I)^{-1} A \Gamma \quad (3.14)$$

donde Ψ viene dada por (3.8).

El muestreo inverso de un oscilador armónico da otra visión del problema del enmascaramiento presentado.



Ejemplo 3.4 -El oscilador armónico

El sistema discreto

$$x(kh + h) = \begin{bmatrix} \cos \alpha h & \text{sen } \alpha h \\ -\text{sen } \alpha h & \cos \alpha h \end{bmatrix} x(kh) + \begin{bmatrix} 1 - \cos \alpha h \\ \text{sen } \alpha h \end{bmatrix} u(kh)$$

puede obtenerse muestreando un sistema continuo con

$$A = \begin{bmatrix} 0 & \beta \\ -\beta & 0 \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 0 \\ \beta \end{bmatrix}$$

donde

$$\beta = \alpha + (2\pi/h) n \quad n = 0, 1, \dots$$

En este caso el problema inverso tiene muchas soluciones. Esto es lo que ocurre generalmente si la matriz Φ tiene valores propios complejos.

Obsérvese que siempre existe un único β en el intervalo $-\omega_N \leq \beta \leq \omega_N$ donde $\omega_N = \pi / h$ es la frecuencia de Nyquist asociada con el periodo de muestreo h .



3.3.3. Transformación de los modelos del espacio de estados

Se ha descrito un sistema discreto para el modelo de la ecuación 3.42. Lo mismo que en los sistemas continuos, se puede introducir nuevas coordenadas en el espacio de estados.

Supongamos que T es una matriz no singular y se define un nuevo vector de estado $z(k) = Tx(k)$. Entonces

$$\begin{aligned}z(k+1) &= Tx(k+1) = T\Phi x(k) + T\Gamma u(k) = T\Phi T^{-1}z(k) + T\Gamma u(k) \\ &= \Phi z(k) + \Gamma u(k)\end{aligned}\tag{3.51}$$

y

$$y(k) = Cx(k) + Du(k) = CT^{-1}z(k) + Du(k) = C^*z(k) + Du(k)\tag{3.52}$$

Las matrices Φ, Γ y C , dependen entonces del sistema de coordenadas escogido para representar el estado.



La ecuación característica

$$\det|\lambda I - \Phi| = 0 \quad (3.53)$$

es invariante cuando se introducen nuevos estados mediante una matriz de transformación T no singular. Demostración.

$$\begin{aligned} \det|\lambda I - \Phi^*| &= \det|\lambda ITT^{-1} - T\Phi T^{-1}| = \det|T| \det|\lambda I - \Phi| \det|T^{-1}| \\ &= \det|\lambda I - \Phi| \end{aligned} \quad (3.54)$$

Las coordenadas pueden escogerse de forma que se obtengan formas sencillas de las ecuaciones del sistema.



Forma diagonal Supongamos que Φ tiene valores propios distintos. Entonces existe una matriz T tal que

$$T\Phi T^{-1} = \begin{bmatrix} \lambda_1 & \dots & 0 \\ & \ddots & \\ 0 & \dots & \lambda_n \end{bmatrix} \quad (3.55)$$

donde λ_i son los valores propios de Φ . En este caso se obtiene un conjunto de ecuaciones en diferencias de primer orden desacopladas

$$\begin{aligned} z_1(k+1) &= \lambda_1 z_1(k) + \beta_1 u(k) \\ &\vdots \\ z_n(k+1) &= \lambda_n z_n(k) + \beta_n u(k) \\ y(k) &= \gamma_1 z_1(k) + \dots + \gamma_n z_n(k) \end{aligned} \quad (3.56)$$

Ahora la solución del sistema de ecuaciones es simple. Cada modo tendrá la solución

$$z_1(k) = \lambda_1^k z_1(0) + \sum_{j=0}^{k-1} \lambda_j^{k-j-1} \beta_j u(j) \quad (3.57)$$

Si Φ tiene valores propios múltiples (eigenvalores), entonces no es posible diagonalizar Φ . Cualquier matriz puede, sin embargo, transformarse a una forma de Jordan. En esta forma la matriz transformada, Φ , tiene los valores propios en la diagonal y algunos unos en la línea situada sobre la diagonal.



Forma observable Supongamos que Φ tiene la ecuación característica

$$\det|\lambda I - \Phi| = \lambda^n + a_1\lambda^{n-1} + \dots + a_n = 0 \quad (3.58)$$

y que

$$\omega_0 = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix} \quad (3.59)$$

es no singular. Entonces existe una matriz de transformación tal que el sistema transformado es de la forma

$$z(k+1) = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & & \vdots \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix} z(k) + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u(k) \quad (3.60)$$

$$y(k) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} z(k)$$

la que se llama forma canónica observable. Esta forma tiene la ventaja de que a partir de ella es fácil encontrar el modelo entrada-salida y determinar un observador adecuado.



Forma controlable Supongamos que la ecuación característica de Φ es la ecuación 3.58 y que

$$\omega_c = [\Gamma \quad \Phi\Gamma \quad \dots \quad \Phi^{n-2}\Gamma \quad \Phi^{n-1}\Gamma] \quad (3.61)$$

es no singular. Entonces existe una transformación tal que el sistema transformado es

$$z(k+1) = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} z(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k) \quad (3.62)$$
$$y(k) = [b_1 \quad \dots \quad b_n] z(k)$$

que recibe el nombre de forma canónica controlable. La ventaja de esta forma es que mediante ella también es fácil de calcular el modelo entrada-salida y la ley de control de realimentación. Las formas observables y controlables también reciben el nombre de formas asociadas.



1.9. Modelos de Entrada Salida

Un sistema dinámico se puede describir bien por su modelo interno o por su modelo externo. Los modelos internos, como el modelo en variables de estado, describen toda relación entre las variables del sistema. Los modelos externos dan solo la relación entre la entrada y la salida del sistema.

En esta sección se verá primero que la relación entrada - salida para un sistema lineal puede expresarse por una función de respuesta impulsional. También se mostrará que el operador desplazamiento puede usarse para obtener directamente la relación entrada salida, la cual lleva a caracterizar el comportamiento entrada salida en términos de los operadores de la respuesta impulsional, en forma similar a lo que ocurre en los sistemas continuos.



1.11. Operador Desplazamiento

El uso del operador diferencial $p = d/dt$ es adecuado para el trabajo con ecuaciones lineales diferenciales a coeficientes constantes. Un operador análogo se definirá para sistemas expresados en ecuaciones en diferencias lineales a coeficientes constantes. En la definición de este operador los sistemas son vistos como operadores que relacionan señales de entrada con señales de salida. Es necesario especificar el rango de validez del operador es decir definir la clase de señal de entrada y qué tipo de actuación tiene el operador sobre las señales. Para los cálculos con este operador todas las señales son consideradas como secuencias doblemente infinitas, debe ir desde $-\infty$ a $+\infty$. Por conveniencia el período de muestreo es elegido como la unidad de tiempo $T = 1$.

- Operador Adelanto

El operador *desplazamiento hacia adelante* se llama q . Tiene la propiedad

$$qf_k = f_{k+1} \quad [1.81]$$

Norma

$$\|f\| = \sup_k |f_k| \quad [1.82]$$



$$\|f\|^2 = \sum_{k=-\infty}^{\infty} f_k^2 \quad [1.83]$$

de lo que se deduce que el operador desplazamiento tiene norma unitaria. Esto significa que las operaciones con el operador desplazamiento son más simples que el cálculo diferencial ya que el operador diferencial es no acotado.

- Operador Retardo

La inversa del operador desplazamiento hacia adelante es llamado operador desplazamiento hacia atrás y su notación es q^{-1} . Resulta

$$q^{-1} f_k = f_{k-1} \quad [1.84]$$

Notemos la importancia de definir el rango del operador para secuencias doblemente infinitas dado que de otra manera el operador desplazamiento hacia atrás no podría existir. En problemas relacionados con la ecuación característica del sistema, como estabilidad y orden del sistema es mejor utilizar el operador desplazamiento hacia adelante. En cambio en problemas relacionados con la causalidad es conveniente usar el operador desplazamiento hacia atrás.



El cálculo con estos operadores da una compacta descripción de los sistemas y hace muy fácil relacionar variables ya que el manejo de ecuaciones en diferencias se reduce a un problema puramente algebraico. En muchos textos se usa z como el operador desplazamiento al igual que la variable compleja de la transformada en Z . Esta diferenciación es la misma que se hace entre la variable compleja s de la transformada de

Laplace y el operador diferencial $p = \frac{d}{dt}$.



Este operador desplazamiento es usado para simplificar el manejo de las ecuaciones en diferencias de un alto orden. Sea la ecuación

$$y_{k+n_a} + a_1 y_{k+n_a-1} + \dots + a_{n_a} y_k = b_0 u_{k+n_b} + \dots + b_{n_b} u_k \quad [1.85]$$
$$n_a > n_b$$

$$\left(q^{n_a} + a_1 q^{n_a-1} + \dots + a_{n_a} \right) y_k = \left(b_0 q^{n_b} + b_1 q^{n_b-1} + \dots + b_{n_b} \right) u_k \quad [1.86]$$

$$A(q) = q^{n_a} + a_1 q^{n_a-1} + \dots + a_{n_a} \quad [1.87]$$

$$B(q) = b_0 q^{n_b} + b_1 q^{n_b-1} + \dots + b_{n_b}$$

$$A(q) y_k = B(q) u_k \quad [1.88]$$

expresado en función del operador retardo

$$y_k + a_1 y_{k-1} + \dots + a_{n_a} y_{k-n_a} = b_0 u_{k-d} + \dots + b_{n_b} u_{k-d-n_b} \quad [1.89]$$



con $d = n_a - n_b$ exceso de polos

polinomio recíproco

$$A^*(q) = 1 + a_1 q^{n_a-1} + \dots + a_{n_a} q^{n_a} = q^{n_a} A(q^{-1}) \quad [1.90]$$

[1.89] se puede escribir

$$A^*(q^{-1})y_k = B^*(q^{-1})u_{k-d} \quad [1.91]$$

Cuidado: A^{**} no necesariamente es A , por ejemplo

$$A(q) = q$$

$$A^*(q) = qq^{-1} = 1 \quad [1.92]$$

$$A^{**}(q) = 1$$



1.12. Función de Transferencia

El cálculo con el operador desplazamiento permite expresar la relación entrada salida como una función racional del operador desplazamiento tanto hacia adelante como hacia atrás. A esta función se la denomina función de transferencia y es fácilmente obtenible de la descripción del sistema eliminando las variables internas usando operaciones algebraicas.

Considérese, por ejemplo, un modelo en variables de estado. Para obtener la relación entrada salida se debe eliminar el vector de estado:

$$x_{k+1} = qx_k = \Phi x_k + \Gamma u_k \quad [1.97]$$

$$(qI - \Phi) x_k = \Gamma u_k \quad [1.98]$$

$$y_k = Cx_k + Du_k = \{C(qI - \Phi)^{-1} \Gamma + D\} u_k \quad [1.99]$$

Función de Transferencia

$$H(q) = C(qI - \Phi)^{-1} \Gamma + D \quad [1.100]$$

expresada en operador retardo

$$H^*(q^{-1}) = C(I - q^{-1}\Phi)^{-1} q^{-1}\Gamma + D = H(q) \quad [1.101]$$

Para sistemas UEUS (SISO)

$$H(q) = C(qI - \Phi)^{-1} \Gamma + D = \frac{B(q)}{A(q)} \quad [1.102]$$

Si el sistema es de orden n , y A y B no tiene factores comunes, A es de grado n

A es el *polinomio característico* del sistema lo que implica que el sistema se puede escribir:

$$y_k + a_1 y_{k-1} + \dots + a_{n_a} y_{k-n_a} = b_0 u_{k-d} + \dots + b_{n_b} u_{k-d-n_b} \quad [1.103]$$

generalmente $b_0 = 0$

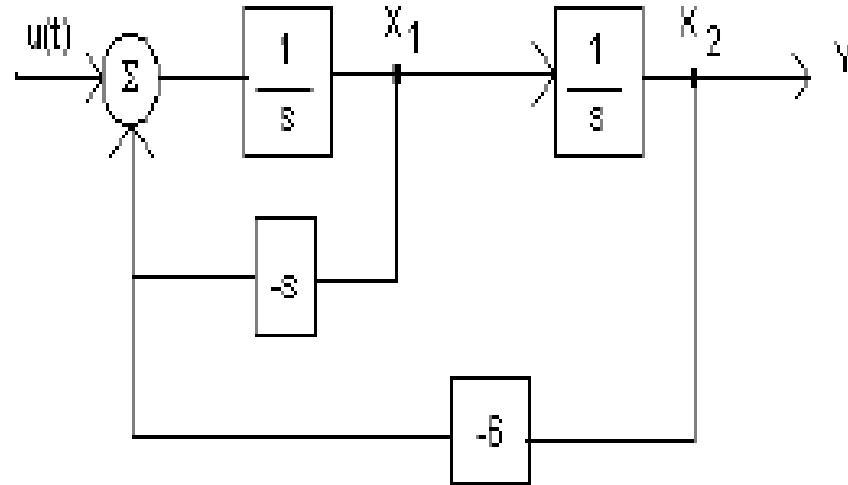


Figura 3.11: Diagrama a bloques del sistema continuo.



Considérese el sistema con la descripción de variables de estado :

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= C^T x(t) + Du(t) \\ \dot{x}_2 &= x_1\end{aligned}\tag{3.73}$$

donde:

$$A = \begin{bmatrix} -5 & -6 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\tag{3.74}$$

$$C^T = [0 \quad 1] \quad D = 0$$



Para calcular la función de transferencia, se forma:

$$sI - A = \begin{bmatrix} s + 5 & 6 \\ -1 & s \end{bmatrix} \quad (3.75)$$

y se calcula :

$$(sI - A)^{-1} = \frac{Adj(sI - A)}{|sI - A|} = \frac{[Cof(sI - A)]^T}{|sI - A|} \quad (3.76)$$

$$= \frac{\begin{bmatrix} s & -1 \\ 6 & s + 5 \end{bmatrix}}{s(s+5)+6} = \frac{\begin{bmatrix} s & -1 \\ 6 & s + 5 \end{bmatrix}}{s^2+5s+6} \quad (3.77)$$



se tiene la función de transferencia de la forma :

$$G_p(s) = \frac{Y(s)}{U(s)} = C^T (sI - A)^{-1} B = [0 \quad 1] \frac{\begin{bmatrix} s & -1 \\ 6 & s+5 \end{bmatrix}}{s^2 + 5s + 6} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.78)$$



También se puede determinar los ceros del sistema a partir de las matrices descriptivas de las variables de estado (A , B , C y D). Los resultados anteriores se pueden combinar para expresar la función de transferencia total de una manera distinta, sin necesidad de realizar el producto de matrices, ni encontrar la matriz inversa, la forma compacta se presenta a continuación:

$$G_p(s) = \frac{\det \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}}{\det[sI - A]} \quad (3.79)$$



Tomando el sistema del ejemplo anterior donde se puede encontrar la ecuación de la planta:

$$A = \begin{bmatrix} -5 & -6 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.80)$$

$$C^T = [0 \quad 1] \quad D = 0$$

se encuentra la función de transferencia de la planta:

$$G_p(s) = \frac{\det \begin{bmatrix} s+5 & 6 & -1 \\ -1 & s & 0 \\ 0 & 1 & 0 \end{bmatrix}}{\det \left[\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -5 & -6 \\ 1 & 0 \end{bmatrix} \right]} = \frac{1}{\det \begin{bmatrix} s+5 & 6 \\ -1 & s \end{bmatrix}} \quad (3.81)$$

$$G_p(s) = \frac{1}{s^2 + 5s + 6} \quad (3.82)$$



CUVyTT-BUAP



transferencia total de una manera compacta .

$$G_p(s) = \frac{\det \left[\begin{array}{c|c} sI - A & -b \\ \hline c & d \end{array} \right]}{\det [sI - A]}$$

Función de transferencia compacta

Ejemplo 3.6

Tomando el sistema del ejemplo 3.5 donde

$$A = \begin{bmatrix} -5 & -6 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad D = 0$$

encontrar la ecuación de la planta

$$G_p(s) = \frac{\det \left[\begin{array}{c|c} s+5 & -6 & -1 \\ -1 & s & 0 \\ \hline 0 & 1 & 0 \end{array} \right]}{\det \left[\begin{array}{c|c} s & 0 \\ 0 & s \end{array} \right] - \begin{bmatrix} -5 & -6 \\ 1 & 0 \end{bmatrix}} = \frac{1}{\det \begin{bmatrix} s+5 & -6 \\ -1 & s \end{bmatrix}}$$

$$G_p(s) = \frac{1}{s^2 + 5s + 6}$$



Ejemplo 10. Función de Transferencia del Doble Integrador

$$\begin{aligned} H(q) &= [1 \quad 0] \begin{bmatrix} q^{-1} & -1 \\ 0 & q^{-1} \end{bmatrix}^{-1} \begin{bmatrix} 0,5 \\ 1 \end{bmatrix} = \frac{0,5(q+1)}{(q-1)^2} \\ &= \frac{0,5(q^{-1} + q^{-2})}{1 - 2q^{-1} + q^{-2}} \end{aligned}$$

Ejemplo 12. Sistema en Forma Canónica Observable

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} -a_1 & 1 \\ -a_2 & 0 \end{bmatrix} x_k + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k \\ y_k &= [1 \quad 0] x_k \end{aligned}$$

$$H(q) = [1 \quad 0] \begin{bmatrix} q + a_1 & 1 \\ a_2 & q \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \frac{b_1 q + b_2}{q^2 + a_1 q + a_2}$$



- Algunas Funciones de Transferencia

Equivalencia entre la función de transferencia continua y el sistema muestreado con bloqueador de orden cero

$$G(s) \qquad H(q) = \frac{b_1 q + b_2}{q^2 + a_1 q + a_2}$$

$$\frac{1}{s} \qquad \frac{T}{q-1}$$

$$\frac{1}{s^2} \qquad \frac{T^2 (q+1)}{2(q-1)^2}$$

$$e^{-sT} \qquad q^{-1}$$

$$\frac{a}{s+a} \qquad \frac{1 - e^{-aT}}{q - e^{-aT}}$$

$$\frac{a}{s(s+a)} \qquad \frac{\frac{1}{a}(aT - 1 + e^{-aT})q + \frac{1}{a}(1 - e^{-aT} - aTe^{-aT})}{q^2 - (1 + e^{-aT})q + e^{-aT}}$$

$$\frac{a^2}{(s+a)^2} \qquad \frac{(1 - e^{-aT}(1 + aT))q + e^{-aT}(e^{-aT} + aT - 1)}{q^2 - 2e^{-aT}q + e^{-2aT}}$$

$$\frac{ab}{(s+a)(s+b)} \qquad \frac{\left(\frac{b(1 - e^{-aT}) - a(1 - e^{-bT})}{b-a} \right) q + \frac{a(1 - e^{-bT})e^{-aT} - b(1 - e^{-aT})e^{-bT}}{b-a}}{q^2 - (e^{-aT} + e^{-bT})q + e^{-(a+b)T}}$$

$$\frac{s+c}{(s+a)(s+b)} \left(\frac{e^{-bT} - e^{-aT} + \frac{c(1-e^{-bT})}{b} - \frac{c(1-e^{-aT})}{a}}{b-a} \right) q + \frac{c}{ab} e^{-(a+b)T} + \frac{b-c}{b(a-b)} e^{-aT} + \frac{c-a}{a(a-b)} e^{-bT}$$

$$q^2 - (e^{-aT} + e^{-bT})q + e^{-(a+b)T}$$

$$b_1 = 1 - e^{-\zeta\omega_0 T} \left(\cos(\omega_0\sqrt{1-\zeta^2}T) + \frac{\zeta\omega_0}{\omega_0\sqrt{1-\zeta^2}} \text{sen}(\omega_0\sqrt{1-\zeta^2}T) \right)$$

$$\frac{\omega_0^2}{s^2 + 2\zeta\omega_0 + \omega_0^2} \quad b_2 = e^{-2\zeta\omega_0 T} + e^{-\zeta\omega_0 T} \left(\frac{\zeta\omega_0}{\omega_0\sqrt{1-\zeta^2}} \text{sen}(\omega_0\sqrt{1-\zeta^2}T) - \cos(\omega_0\sqrt{1-\zeta^2}T) \right)$$

$$a_1 = -2e^{-\zeta\omega_0 T} \cos(\omega_0\sqrt{1-\zeta^2}T)$$

$$a_2 = e^{-2\zeta\omega_0 T}$$

$$b_1 = \frac{1}{\omega_0\sqrt{1-\zeta^2}} e^{-\zeta\omega_0 T} \text{sen}(\omega_0\sqrt{1-\zeta^2}T)$$

$$\frac{s}{s^2 + 2\zeta\omega_0 + \omega_0^2} \quad b_2 = -b_1$$

$$a_1 = -2e^{-\zeta\omega_0 T} \cos(\omega_0\sqrt{1-\zeta^2}T)$$

$$a_2 = e^{-2\zeta\omega_0 T}$$



3.5. Transformada Z

La transformada discreta análoga a la de Laplace es la transformada Z una herramienta muy apropiada para estudiar las ecuaciones en diferencias lineales con o sin condiciones iniciales. La transformada Z se aplica a una serie de secuencias temporales semifinitas sobre el plano complejo. Nótese la diferencia en rango para la transformada Z y para el cálculo con operador q . La variable Z es una variable compleja y debe distinguirse del operador q .

Transformada Z considérese la señal discreta $[f(kh) : k = 0, 1, \dots]$. La transformada Z de $f(kh)$ se define como:

$$F(z) = \sum_{k=0}^{\infty} f(kh)z^{-k} \quad (3.83)$$



donde z es una variable compleja. La transformada Z de f se representa por Z , o $F(z)$. La transformada inversa viene dada por:

$$f(kh) = \frac{1}{2\pi i} \oint F(z) z^{k-1} dz \quad (3.84)$$

donde el contorno de integración encierra todas las singularidades de $F(z)$.

Sea $y(kh) = kh$ para $k \geq 0$ entonces:

$$Y(z) = 0 + hz^{-1} + 2hz^{-2} + \dots = h(z^{-1} + 2z^{-2} + \dots) = \frac{hz}{(z-1)^2} \quad (3.85)$$

En la tabla se dan algunas propiedades de la transformada z . Se nota que las fórmulas para el desplazamiento hacia adelante y hacia atrás no son las mismas. Esto es consecuencia de la hipótesis de que las secuencias temporales son semifinitas. Algunas de las propiedades de la Transformada



1.- Definición:

$$F(z) = \sum_{k=0}^{\infty} f(kh)z^{-k} \quad (3.86)$$

2.- Inversión:

$$f(kh) = \frac{1}{2\pi i} \oint F(z)z^{k-1}dz \quad (3.87)$$

3.- Linealidad:

$$Z(\alpha f + \beta g) = \alpha Z(f) + \beta Z(g) \quad (3.88)$$

4.- Corrimiento en el tiempo:

$$Zq^{-n} f = z^{-n} F \quad (3.89)$$

$$Z(q^n f) = z^n (F - F_1)$$

donde:

$$F_1(z) = \sum_{j=0}^{n-1} f(jh)z^{-j} \quad (3.90)$$

5.- Teorema del valor inicial:

$$f(0) = \lim_{z \rightarrow \infty} F(z) \quad (3.91)$$

6.- Teorema del valor Final: Si $(1 - z^{-1})F(z)$ no tiene bastantes polos dentro del círculo unitario, entonces:

$$\lim_{k \rightarrow \infty} f(kh) = \lim_{z \rightarrow 1} (1 - z^{-1})F(z) \quad (3.92)$$

7.- Convolución:

$$Z(f \otimes g) = Z\left(\sum_{n=0}^k f(n)g(k-n)\right) = (Zf) \otimes (Zg) \quad (3.93)$$



LA TRANSFORMADA Z

Definición:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

Desde un sentido estrictamente matemático, la transformada z es la expansión en serie de Laurent de la función $X(z)$ alrededor del origen donde los coeficientes de la serie de tiempo $x(n)$ son los coeficientes de la expansión de la serie..

Su utilidad desde el punto de vista de la ingeniería, radica en el hecho, que un sistema continuo puede ser modelado usando la transformada z, y ser convertido en un programa para fines de simulación o control.



Series geométricas

Serie geométrica finita:

$$\sum_{k=0}^N \alpha^k = \frac{1 - \alpha^{N+1}}{1 - \alpha} \quad \text{con} \quad |\alpha| < 1$$

Serie geométrica infinita:

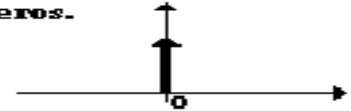
$$\sum_{k=0}^{\infty} \alpha^k = \frac{1}{1 - \alpha} \quad \text{con} \quad |\alpha| < 1$$

Ejemplos:

1.- Impulso unitario.

$$x(k) = \begin{cases} 1 & ; k=0. \\ 0 & ; k \neq 0. \end{cases}$$

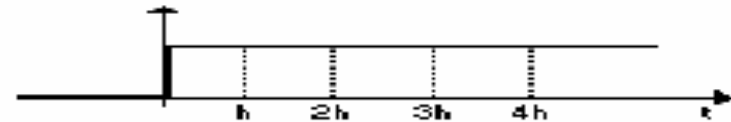
k -- pertenece a los enteros.



$$Z[x(k)] = \sum_{k=0}^{\infty} x(k) z^{-k} = 1$$

2.- Escalón unitario.

$$U(k) = \begin{cases} 1 & ; k < 0 \\ 0 & ; k \geq 0 \end{cases}$$



$$Z[U(k)] = \sum_{k=0}^{\infty} U(kh) z^{-k} = 1 + z^{-1} + z^{-2} + \dots + z^{-k}$$

$$Z[U(+)] = \frac{z}{z-1} = \frac{\frac{z}{z}}{\frac{1}{z}(z-1)} = \frac{1}{(1-\frac{1}{z})}$$

$$= \frac{1}{(1-z^{-1})}$$



Ejemplos:

a).- Impulso unitario.

$$\delta(k) = \begin{cases} 1, & k = 0. \\ 0, & k \neq 0. \end{cases} \quad (3.94)$$

su transformada Z es:

$$Z[\delta(k)] = \sum_{k=0}^{\infty} \delta(k)Z^{-k} = 1 \quad (3.95)$$

b).- Escalón unitario.

$$u(k) = \begin{cases} 0, & k < 0. \\ 1, & k \geq 0. \end{cases} \quad (3.96)$$

$$\begin{aligned} Z[u(kh)] &= \sum_{k=0}^{\infty} u(kh)z^{-k} = 1 + z^{-1} + z^{-2} + \dots + z^{-k} \\ &= \frac{z}{z-1} = \frac{\frac{z}{z}}{\frac{1}{z}(z-1)} = \frac{1}{(1-\frac{1}{z})} \\ &= \frac{1}{(1-z^{-1})} \end{aligned} \quad (3.97)$$

3.- Rampa

$$r(t) = \begin{cases} Kt & t > 0 \\ 0 & t < 0 \end{cases}$$

$$Z[r(t)] = \sum_{k=0}^{\infty} khZ^k = hZ^1 + 2hZ^2 + 3hZ^3 + \dots = h(Z^1 + Z^2 + \dots)$$

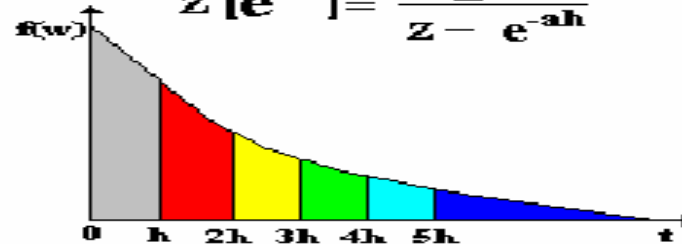
$$Z[r(t)] = \frac{hZ}{(Z-1)^2}$$

4.-

$$f(t) = e^{-at}$$

$$Z[e^{-at}] = \sum_{k=0}^{\infty} e^{-akh} z^{-k} = 1 + e^{-ah} z^{-1} + e^{-2ah} z^{-2} + \dots$$

$$Z[e^{-at}] = \frac{z}{z - e^{-ah}}$$



5.-

$$Z[\text{Sen } wt] = \frac{z \text{ sen } wh}{z^2 - 2z \cos wh + 1}$$



c).- Rampa

$$r(t) = \begin{cases} kt, & t > 0. \\ 0, & t < 0. \end{cases} \quad (3.98)$$

$$Z[r(t)] = \sum_{k=0}^{\infty} khz^{-k} = hz^{-1} + 2hz^{-2} + 3hz^{-3} + \dots = h(z^{-1} + z^{-2} + \dots) \quad (3.99)$$

$$Z[r(t)] = \frac{hz}{(z-1)^2} \quad (3.100)$$

d).- Exponencial

$$f(t) = e^{-at} \quad (3.101)$$

$$Z[e^{-at}] = \sum_{k=0}^{\infty} e^{-akh} z^{-k} = 1 + e^{-ah} z^{-1} + e^{-2ah} z^{-2} + \dots \quad (3.102)$$

$$Z[e^{-at}] = \frac{z}{z - e^{-ah}} \quad (3.103)$$

e).- Senoidal

$$Z[\text{sen}(wt)] = \frac{z \text{sen}(wh)}{z^2 - 2z \cos(wh) + 1} \quad (3.104)$$



Propiedades de la transformada z

Linealidad

Serie de tiempo $x(n) = aw(n) + by(n)$

Aplicando la definición de la transformada

$$X(z) = \sum_{n=-\infty}^{\infty} [aw(n) + by(n)]z^{-n}$$

Re ordenando la serie de tiempo

$$X(z) = a \sum_{n=-\infty}^{\infty} [w(n)]z^{-n} + b \sum_{n=-\infty}^{\infty} [y(n)]z^{-n}$$

La transformada z queda:

$$X(z) = aW(z) + bY(z)$$

Corrimiento

Serie de tiempo $y(n) = x(n - m)$

Aplicando la definición de la transformada

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n - m)z^{-n}$$

Re ordenando la serie de tiempo

$$Y(z) = \sum_{k=-\infty}^{\infty} x(k)z^{-(k+m)} \quad k = n - m$$

$$Y(z) = z^{-m} \sum_{k=-\infty}^{\infty} x(k)z^{-k}$$

La transformada z queda:

$$Y(z) = z^{-m} X(z)$$



Propiedades de la transformada z

Propiedad	Serie de tiempo	Transformada z	Región de Convergencia
	$x(n)$ $y(n)$	$X(z)$ $Y(z)$	$r_{cx} < z < r_{ax}$ $r_{cy} < z < r_{ay}$
Linealidad	$ax(n)+by(n)$	$aX(z)+bY(z)$	At least $\max(r_{cx}, r_{cy})$ $< z < \min(r_{ax}, r_{ay})$
Corrimiento en el tiempo	$x(n-m)$	$z^{-m}X(z)$	$r_{cx} < z < r_{ax}$
Convolución	$\sum_{k=-\infty}^{\infty} x(k)y(n-k)$	$X(Z)Y(z)$	Al menos $\max(r_{cx}, r_{cy})$ $< z < \min(r_{ax}, r_{ay})$
Multiplicación exponencial	$a^n x(n)$	$X(a^{-1}z)$	$ a r_{cx} < z < a r_{ax}$
Multiplicación en el tiempo	$nx(n)$	$-z \frac{dX(z)}{dz}$	$r_{cx} < z < r_{ax}$



Propiedades de la transformada z

Producto	$x(n)y(n)$	$\frac{1}{2\pi j} \oint_c X(w)Y\left(\frac{z}{w}\right)w^{-1}dw$	$r_{cx}r_{cy} < z < r_{ax}r_{ay}$
Correlación	$\sum_{k=-\infty}^{\infty} x(k)y(n+k)$	$X(z^{-1})Y(z)$	Al menos $\max(r_{cx}, r_{cy}) < z < \min(r_{ax}, r_{ay})$
Traspuesta en el tiempo	$x(-n)$	$X(z^{-1})$	$r_a^{-1} < z < r_c^{-1}$



Ejemplo 1 de transformada z

Determinar la transformada z de la siguiente serie de tiempo

$$x(n) = 4\delta(n+1) - 2\delta(n) + \pi\delta(n-5)$$

Usando la propiedad de corrimiento en el tiempo:

$$X(z) = 4z - 2z^0 + \pi z^{-5}$$



La transformada z puede emplearse para resolver ecuaciones en diferencias, por ejemplo:

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) \\y(k) &= Cx(k)\end{aligned}\tag{3.105}$$

Si se toma la transformada z de ambos lados:

$$\sum_{k=0}^{\infty} x(k+1)z^{-k} = Z\left[\sum_{k=0}^{\infty} z^{-k} x(k) - x(0)\right]\tag{3.106}$$

$$= \sum_{k=0}^{\infty} \Phi z^{-k} x(k) + \sum_{k=0}^{\infty} \Gamma z^{-k} u(k)\tag{3.107}$$

Por lo tanto:

$$Z[X(z) - x(0)] = \Phi X(z) + \Gamma U(z) \Rightarrow X(z) = (zI - \Phi)^{-1}[zx(0) + \Gamma U(z)]\tag{3.108}$$

y

$$Y(z) = C(zI - \Phi)^{-1}zx(0) + C(zI - \Phi)^{-1}\Gamma U(z)\tag{3.109}$$



Se puede introducir la función de transferencia discreta:

$$H(z) = C(zI - \Phi)^{-1}\Gamma = \frac{\det \begin{bmatrix} zI - \Phi & -\Gamma \\ C & D \end{bmatrix}}{\det[zI - \Phi]} = \frac{Y(z)}{U(z)} \quad (3.110)$$

La secuencia temporal $y(k)$ puede ya obtenerse mediante la transformación inversa.



CUVYTT-BUAP



f	$L(f)$	$Z(f)$
1 (escalón)	$\frac{1}{s}$	$\frac{z}{z-1}$
kT	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{2}(kT)^2$	$\frac{1}{s^3}$	$\frac{T^2z(z+1)}{2(z-1)^3}$
$e^{-\frac{kT}{\tau}}$	$\frac{\tau}{1+s\tau}$	$\frac{z}{z-e^{-\frac{T}{\tau}}}$
$1-e^{-\frac{kT}{\tau}}$	$\frac{1}{s(1+s\tau)}$	$\frac{z\left(1-e^{-\frac{T}{\tau}}\right)}{(z-1)\left(z-e^{-\frac{T}{\tau}}\right)}$
$\text{sen}(\omega kT)$	$\frac{\omega}{s^2 + \omega^2}$	$\frac{z \text{sen}(\omega T)}{(z^2 - 2z \cos(\omega T) + 1)}$

Table 13.1
 Partial table of z- and s-transforms

	f(t)	F(s)	F(z)	f(kT)
1.	$u(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$	$u(kT)$
2.	t	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$	kT
3.	t^n	$\frac{n!}{s^{n+1}}$	$\lim_{a \rightarrow 0} (-1)^n \frac{d^n}{da^n} \left[\frac{z}{z - e^{-aT}} \right]$	$(kT)^n$
4.	e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z - e^{-aT}}$	e^{-akT}
5.	$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$	$(-1)^n \frac{d^n}{da^n} \left[\frac{z}{z - e^{-aT}} \right]$	$(kT)^n e^{-akT}$
6.	$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$	$\frac{z \sin \omega T}{z^2 - 2z \cos \omega T + 1}$	$\sin \omega kT$
7.	$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$	$\frac{z(z - \cos \omega T)}{z^2 - 2z \cos \omega T + 1}$	$\cos \omega kT$
8.	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2 + \omega^2}$	$\frac{ze^{-aT} \sin \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$	$e^{-akT} \sin \omega kT$
9.	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2 + \omega^2}$	$\frac{z^2 - ze^{-aT} \cos \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$	$e^{-akT} \cos \omega kT$

Table 13.2
z-transform theorems

Theorem	Name
1. $z\{af(t)\} = aF(z)$	Linearity theorem
2. $z\{f_1(t) + f_2(t)\} = F_1(z) + F_2(z)$	Linearity theorem
3. $z\{e^{-at}f(t)\} = F(e^{aT}z)$	Complex differentiation
4. $z\{f(t - nT)\} = z^{-n}F(z)$	Real translation
5. $z\{tf(t)\} = -Tz \frac{dF(z)}{dz}$	Complex differentiation
6. $f(0) = \lim_{z \rightarrow \infty} F(z)$	Initial value theorem
7. $f(\infty) = \lim_{z \rightarrow 1} (1 - z^{-1})F(z)$	Final value theorem

Note: kT may be substituted for t in the table.

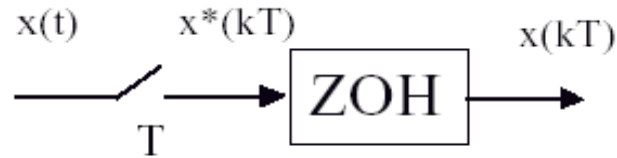


Retenedores de orden cero (ZOH)

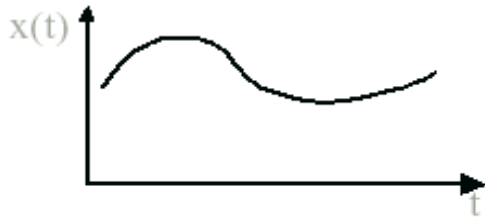
En los sistemas reales, para la correcta conversión digital analógica, así como para el procesamiento de la información, es necesario contar con sistemas que mantengan el valor de la señal de datos muestreados en valores constantes entre instantes de muestreo

A los sistemas que realizan esta función se les denomina retenedores, de los cuales el más común en sistemas de control es el de orden cero

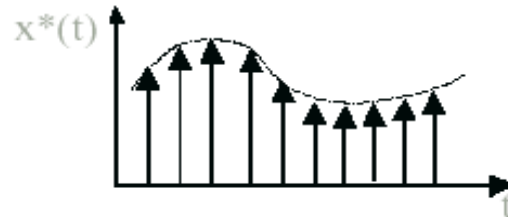
Equivalencia de un retenedor de orden cero



Señal continua



Señal muestreada



Señal retenida





La función de transferencia de un retenedor de orden cero es un elemento de retardo de transporte definido por:

$$ZOH(s) = \frac{1 - e^{-sT}}{s}$$

Para su uso en sistemas de control de tiempo continuo de utiliza la aproximación de *Padé*

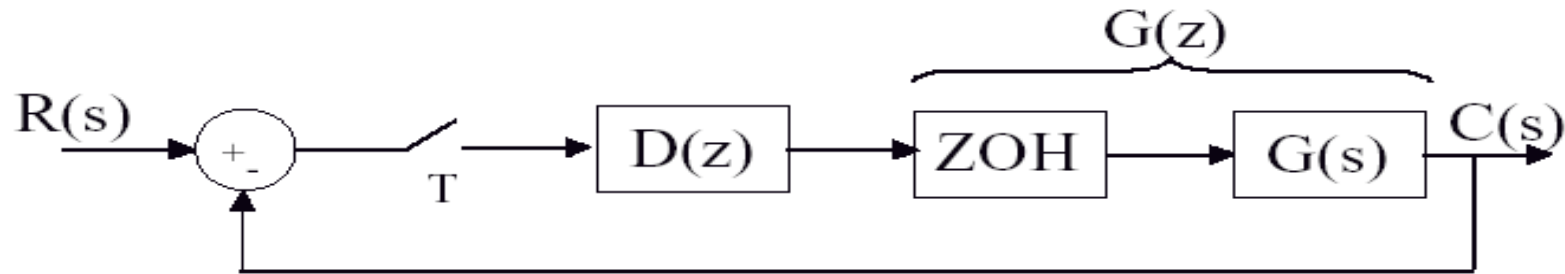
$$ZOH(s) \cong \frac{kT}{1 + sT/2}$$

Y para su uso en sistemas de control de tiempo discreto hay que recordar que $z = e^{sT}$ y se tiene

$$ZOH(z) = (1 - z^{-1}) \cdot Z \left\{ \frac{1}{s} \right\}$$

Efecto de muestreadores y retenedores en un lazo de control

El sistema de control:



Tiene la función de transferencia:

$$C(z) = \frac{G(z)}{1 + D(z)G(z)}$$



Del diagrama de bloques podemos sacar dos conclusiones fundamentales:

- La función de transferencia del controlador $D(z)$ puede o una función diseñada directamente en el dominio del tiempo discreto (Control digital directo CDD) o discretizada de algún control analógico previamente diseñado (control digital basado en control analógico) por cualquier método de transformación conforme,,
- La función de la planta siempre es una función continua y su discretización solo se lleva a cabo para fines de análisis y siempre debe incluir una función de retención asociada.

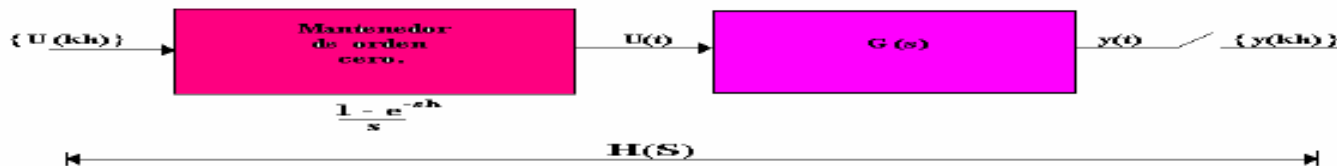
Cálculo de la función de transferencia discreta.

Es posible determinar la función de transferencia discreta directamente a partir de la función de transferencia continua. Sea el sistema descrito por la función de transferencia $G(s)$ precedido por un mantenedor de orden cero (véase la fig. 3.3).

La función de transferencia discreta queda determinada de forma única por la respuesta a una señal dada. Consideremos, por ejemplo, una entrada escalón unitaria. La secuencia $[U(kh)]$ es entonces una secuencia de unos, y la señal $U(t)$ es también un escalón unitario. Representemos por $Y(s)$ la transformada de Laplace de $y(t)$ es decir :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Y(s)}{1/s} \Rightarrow Y(s) = \frac{G(s)}{s}$$

Figura 3.3 MUESTREO DE UN SISTEMA CONTINUO.





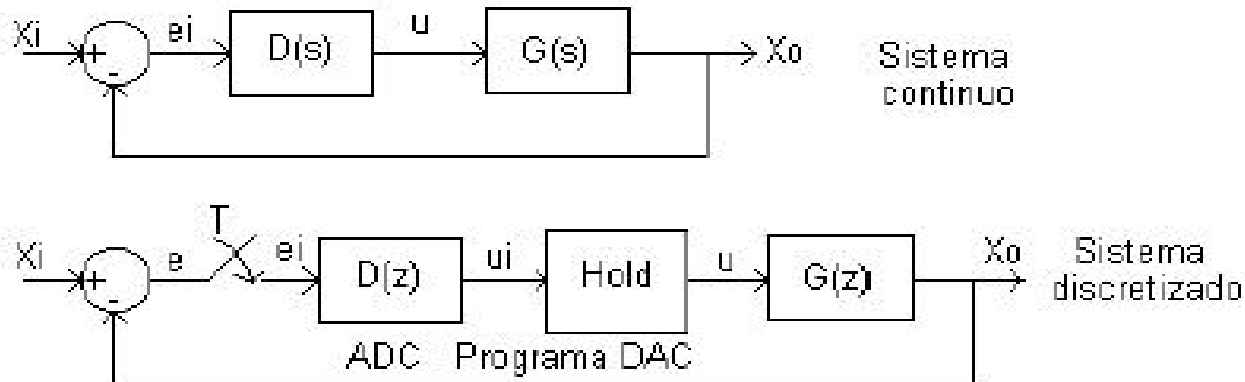
Salida del sistema:

$$\begin{aligned} Z(H) &= Z[\{ (1 - e^{-sh}) / s \} G(s)] \\ &= Z[\{ (1 - e^{-sh}) \cdot (G(s) / s) \}] \\ &= (1 - Z^{-1}) \cdot Z[(G_p(s) / s)] \end{aligned}$$

La salida muestreada $\{y(kh)\}$ tiene como transformada $z Y' = Zy = Z \delta^{-1} Y$ y la división de Y' por la función de transferencia discreta de la entrada que es $z / (z-1)$, nos da:

$$H(z) = (1 - z^{-1}) Y'(z) \quad (3.21)$$

Procedimiento de discretización de distemas analógicos





La función de transferencia discreta se obtiene entonces de la forma siguiente:

- 1.- Determinar la función temporal que corresponde a $G(s)/s$.
- 2.- Determinar la transformada z correspondiente (por uso de una tabla).
- 3.- Multiplicar por $1 - z^{-1}$ para encontrar la función de transferencia discreta con un mantenedor de orden cero.



$$\mathbf{x}(t) = e^{A(t-t_0)} \mathbf{x}(t_0) + \int_{t_0}^t e^{A(t-\tau)} B u(\tau) d\tau. \quad (2)$$

La relación entre las variables del sistema en los instantes de muestreo se determina estableciendo el tiempo de muestreo t_k , de aquí que la solución (2) se vea como

$$\mathbf{x}(t_k) = e^{A(t_k-t_{k-1})} \mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k-t_{k-1})} B u(t_{k-1}) dt_{k-1} \quad (3)$$

y el sistema discreto completo sería

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + \Gamma u_{k-1} \quad (4)$$

$$y_k = C \mathbf{x}_k$$

donde

$$\Phi = e^{Ah} \quad (5)$$

$$\Gamma = \int_0^h e^{Ah} B dh$$



Problema 3.2b . *Obtener el sistema discreto correspondiente al siguiente sistema continuo, usando un circuito retenedor de orden cero.*

$$\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 2y = \frac{du}{dt} + 3u \quad (7)$$

Este sistema presenta la siguiente función de transferencia

$$G(s) = \frac{s + 3}{s^2 + 3s + 2} = \frac{x_1}{u} \cdot \frac{y}{x_1}$$

y el siguiente modelo en espacio de estado

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + u \\ y &= 3x_1 + x_2. \end{aligned}$$



A partir de este modelo se tiene que

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [3 \quad 1].$$

Utilizando uno de los métodos más comunes para calcular la matriz e^{Ah} basado en

$$e^{Ah} = \sum_{k=0}^{n-1} \alpha_k A^k,$$

este sistema de orden $n = 2$ se vería así

$$e^{Ah} = \begin{bmatrix} \alpha_0 & 0 \\ 0 & \alpha_0 \end{bmatrix} + \begin{bmatrix} 0 & \alpha_1 \\ -2\alpha_1 & -3\alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 & \alpha_1 \\ -2\alpha_1 & \alpha_0 - 3\alpha_1 \end{bmatrix}.$$



El sistema de ecuaciones puede expresarse de la siguiente forma para facilitar el cálculo de las constantes α_0 y α_1

$$e^{\lambda_1 h} = \alpha_0 + \alpha_1 \lambda_1 \quad (8)$$

$$e^{\lambda_2 h} = \alpha_0 + \alpha_1 \lambda_2, \quad (9)$$

los valores propios λ_1 y λ_2 se calculan mediante el determinante $[\lambda I - A]$ así que

$$\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} = \begin{bmatrix} \lambda & -1 \\ 2 & \lambda + 3 \end{bmatrix} = \lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2) = 0$$

por lo tanto

$$\lambda_1 = -1$$

$$\lambda_2 = -2,$$

y ahora las ecuaciones (8) y (9) quedarían como

$$e^{-h} = \alpha_0 - \alpha_1 \quad (10)$$

$$e^{-2h} = \alpha_0 - 2\alpha_1. \quad (11)$$



El siguiente paso es despejar α_0 de (10)

$$\alpha_0 = \alpha_1 + e^{-h}, \quad (12)$$

enseguida se restan las ecuaciones (10) y (11) obteniendo

$$\alpha_1 = e^{-h} - e^{-2h} = e^{-h}(1 - e^{-h}).$$

ahora sólo sustituimos el valor de α_1 en (12) y tenemos que

$$\alpha_0 = e^{-h}(2 - e^{-h}).$$

La matriz e^{Ah} sería

$$\Phi = e^{Ah} = \begin{bmatrix} e^{-h}(2 - e^{-h}) & e^{-h}(1 - e^{-h}) \\ -2e^{-h}(1 - e^{-h}) & e^{-h}(2e^{-h} - 1) \end{bmatrix}$$



El producto $e^{Ah}B$ que se observa en (5) quedaría como

$$e^{Ah}B = \begin{bmatrix} e^{-h}(1 - e^{-h}) \\ e^{-h}(2e^{-h} - 1) \end{bmatrix},$$

El último paso es obtener la integral Γ y el resultado fue el siguiente:

$$\Gamma = \begin{bmatrix} e^{-h}\left(\frac{e^{-h}}{2} - 1\right) + \frac{1}{2} \\ e^{-h}(1 - e^{-h}) \end{bmatrix},$$

por lo tanto el sistema discreto completo es

$$\begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} = \begin{bmatrix} e^{-h}(2 - e^{-h}) & e^{-h}(1 - e^{-h}) \\ -2e^{-h}(1 - e^{-h}) & e^{-h}(2e^{-h} - 1) \end{bmatrix} \begin{bmatrix} x_{1k-1} \\ x_{2k-1} \end{bmatrix} + \begin{bmatrix} e^{-h}\left(\frac{e^{-h}}{2} - 1\right) + \frac{1}{2} \\ e^{-h}(1 - e^{-h}) \end{bmatrix} u_{k-1}$$

$$y_k = [3 \quad 1] \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix}$$



La función de transferencia discreta se obtiene entonces de la forma siguiente:

- 1.- Determinar la función temporal que corresponde a $G(s)/s$.
- 2.- Determinar la transformada z correspondiente (por uso de una tabla).
- 3.- Multiplicar por $1 - z^{-1}$ para encontrar la función de transferencia discreta con un mantenedor de orden cero.

En el ejemplo que se muestra se encontrará la función de transferencia discreta de una ecuación de segundo orden, se realizará por medio de fracciones parciales buscando la transformada z , la función de transferencia original es:

$$G_p(s) = \frac{1}{s^2 + 5s + 4} \quad (3.142)$$

dividiendo entre s y obteniendo su representación en terminos de fracciones parciales queda así:

$$\frac{G_p(s)}{s} = \frac{1}{s^3 + 5s^2 + 4s} = \frac{1}{12(s + 4)} - \frac{1}{3(s + 1)} + \frac{1}{4s} \quad (3.143)$$



$$\begin{aligned} Z\left(\frac{1}{5s^3+5s^2+4s}\right) &= Z\left(\frac{1}{12(s+4)}\right) - Z\left(\frac{1}{3(s+1)}\right) + Z\left(\frac{1}{4s}\right) \\ &= \frac{1}{12}\left(\frac{e^{-4h}z}{z-e^{-4h}}\right) - \frac{1}{3}\left(\frac{ze^{-h}}{z-e^{-h}}\right) + \frac{1}{4}\left(\frac{z}{z-1}\right) \\ &= \frac{1}{12}\left(\frac{ze^{4h}}{ze^{4h}-1}\right) - \frac{1}{3}\left(\frac{ze^h}{ze^h-1}\right) + \frac{1}{4}\left(\frac{z}{z-1}\right) \\ &= \frac{z^2(3ze^{5h}+(1-4z)e^{4h}+(z-4)e^h+3)}{12(ze^{4h}-1)(ze^h-1)(z-1)} \end{aligned} \tag{3.144}$$

Multiplicando por el retenedor de orden cero

$$H_p(z) = (1-z^{-1})Z\left(\frac{G_p(s)}{s}\right) = \frac{(e^{4h}-4e^h+3)z^{-1}+3e^{5h}-4e^{4h}+e^h}{12z^{-2}-(12e^{4h}+12e^h)Z^{-1}+12e^{5h}} = \frac{Y(z)}{U(z)} \tag{3.145}$$

despejando $Y(z)$ queda

$$Y(z)[12z^{-2}-(12e^{4h}+12e^h)z^{-1}+12e^{5h}] = U(z)[(e^{4h}-4e^h+3)z^{-1}+3e^{5h}-4e^{4h}+e^h] \tag{3.146}$$



Tiene la forma general de

$$H_p(z) = \frac{b_2 z^{-2} + b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + a_0} \quad (3.147)$$

donde:

$$b_0 = 3e^{5h} - 4e^{4h} + e^h$$

$$b_1 = e^{4h} - 4e^h + 3$$

$$b_2 = 0$$

(3.148)

$$a_2 = 12$$

$$a_1 = -12e^{4h} - 12e^h$$

$$a_0 = 12e^{5h}$$

la ecuación de diferencias de salida es:

$$y(k) = \frac{b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) - a_1 y(k-1) - a_2 y(k-2)}{a_0} \quad (3.149)$$



Ahora se trabajará con la misma ecuación de segundo orden 3.142, nada mas que con su representación en variables de estado:

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0] \quad D = 0 \quad (3.150)$$

$$\det[\lambda I - Ah] = \begin{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & h \\ -4h & -5h \end{bmatrix} \end{bmatrix} = \lambda^2 + 5\lambda h + 4h^2 \Rightarrow \lambda_0 = -4h$$
$$\lambda_1 = -h \quad (3.151)$$

sustituyendo en

$$e^{\lambda_0} = \alpha_0 + \alpha_1 \lambda_0$$

$$e^{\lambda_1} = \alpha_0 + \alpha_1 \lambda_1$$

$$e^{-4h} = \alpha_0 + \alpha_1(-4h)$$

$$e^{-h} = \alpha_0 + \alpha_1(-h)$$

(3.152)

se resuelve para α

$$\Rightarrow \alpha_0 = \frac{\begin{bmatrix} e^{-4h} & -4h \\ e^{-h} & -h \end{bmatrix}}{\begin{bmatrix} 1 & -4h \\ 1 & -h \end{bmatrix}} = \frac{4he^{-h} - he^{-4h}}{-h+4h} = \frac{4e^{-h} - e^{-4h}}{3} \quad (3.153)$$

$$\alpha_1 = \frac{\begin{bmatrix} 1 & e^{-4h} \\ 1 & e^{-h} \end{bmatrix}}{3h} = \frac{e^{-h} - e^{-4h}}{3h}$$

$$\begin{aligned} e^{Ah} = \alpha_0 I + \alpha_1 Ah &= \begin{bmatrix} \frac{4e^{-h} - e^{-4h}}{3} & 0 \\ 0 & \frac{4e^{-h} - e^{-4h}}{3} \end{bmatrix} + \begin{bmatrix} 0 & \frac{e^{-h} - e^{-4h}}{3} \\ -\frac{4}{3}(e^{-h} - e^{-4h}) & -\frac{5}{3}(e^{-h} - e^{-4h}) \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{3}(4e^{-h} - e^{-4h}) & \frac{1}{3}(e^{-h} - e^{-4h}) \\ \frac{4}{3}(e^{-4h} - e^{-h}) & \frac{1}{3}(4e^{-4h} - e^{-h}) \end{bmatrix} = \Phi(h) \end{aligned} \quad (3.154)$$



CUVYTT-BUAP



$$\Gamma = \int_0^h e^{Ah} B dh = \int_0^h \begin{bmatrix} \frac{1}{3}(e^{-h} - e^{-4h}) \\ \frac{1}{3}(4e^{-4h} - e^{-h}) \end{bmatrix} dh = \begin{bmatrix} \frac{1}{12}(e^{-4h} - 4e^{-h} + 3) \\ \frac{1}{3}(e^{-h} - e^{-4h}) \end{bmatrix} \quad (3.155)$$

para este caso se tienen 2 estados:

$$\begin{aligned} x_1(k+1) &= \Phi_{11}x_1(k) + \Phi_{12}x_2(k) + \Gamma_{11}u(k) \\ x_2(k+1) &= \Phi_{21}x_1(k) + \Phi_{22}x_2(k) + \Gamma_{21}u(k) \\ y(k) &= [1 \quad 0]x(k) \end{aligned} \quad (3.156)$$

donde

$$\begin{aligned} \Phi_{11} &= \frac{1}{3}(4e^{-h} - e^{-4h}) \\ \Phi_{12} &= \frac{1}{3}(e^{-h} - e^{-4h}) \\ \Gamma_{11} &= \frac{1}{12}(e^{-4h} - 4e^{-h} + 3) \\ \Phi_{21} &= \frac{4}{3}(e^{-4h} - e^{-h}) \\ \Phi_{22} &= \frac{1}{3}(4e^{-4h} - e^{-h}) \\ \Gamma_{21} &= \frac{1}{3}(e^{-h} - e^{-4h}) \end{aligned} \quad (3.157)$$



CUVYTT-BUAP



En el siguiente ejemplo se va a deducir un sistema discreto a partir del sistema continuo de un integrador triple, cuando se emplea un mantenedor de orden cero.

$$\frac{d^3 y(t)}{dt^3} = u(t)$$

$$s^3 Y(s) = U(s) \Rightarrow \frac{Y(s)}{U(s)} = \frac{1}{s^3}$$

$$\Rightarrow A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0 \quad 0] \quad d = 0$$

(3.135)

$$\det [\lambda I - Ah] = \det \left[\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & h & 0 \\ 0 & 0 & h \\ 0 & 0 & 0 \end{bmatrix} \right] = \det \begin{bmatrix} \lambda & -h & 0 \\ 0 & \lambda & -h \\ 0 & 0 & \lambda \end{bmatrix}$$

$$= \lambda(\lambda^2) = \lambda^3 = 0 \Rightarrow \lambda_1 = \lambda_2 = \lambda_3 = 0$$



$$\begin{aligned}e^{At} &= \alpha_0 I + \alpha_1 Ah + \alpha_2 A^2 h^2 \\e^{\lambda_1} &= \alpha_0 I + \alpha_1 \lambda_1 + \alpha_2 \lambda_1^2 \quad \Rightarrow 1 = \alpha_0 \\e^{\lambda_2} &= \alpha_0 I + \alpha_1 \lambda_2 + \alpha_2 \lambda_2^2 \\e^{\lambda_3} &= \alpha_0 I + \alpha_1 \lambda_3 + \alpha_2 \lambda_3^2\end{aligned}\tag{3.136}$$

derivando para encontrar los otros valores de α se tiene:

$$\begin{aligned}\frac{de^{\lambda_2}}{d\lambda_2} &= e^{\lambda_2} = \alpha_1 + \alpha_2 2\lambda_2 \quad \Rightarrow \quad 1 = \alpha_1 \\ \frac{de^{\lambda_3}}{d\lambda_3} &= e^{\lambda_3} = 2\alpha_2 \quad \Rightarrow \quad \frac{1}{2} = \alpha_2\end{aligned}\tag{3.137}$$

de la siguiente ecuación

$$e^{Ah} = \alpha_0 I + \alpha_1 Ah + \alpha_2 A^2 h^2 = \Phi(h)\tag{3.138}$$



se obtiene

$$\Phi(h) = 1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 1 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} h + \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} h^2$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & h & 0 \\ 0 & 0 & h \\ 0 & 0 & 0 \end{bmatrix} + \frac{h^2}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$e^{Ah} = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} = \Phi(h)$$

(3.139)



se encuentra $\Gamma(h)$ a partir de la siguiente ecuación:

$$\begin{aligned}\Gamma(h) &= \int_0^h \Phi(h) B dh = \int_0^h \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} dh \\ &= \int_0^h \begin{bmatrix} \frac{h^2}{2} \\ h \\ 1 \end{bmatrix} dh = \begin{bmatrix} \frac{h^3}{6} \\ \frac{h^2}{2} \\ h \end{bmatrix} = \Gamma(h)\end{aligned}\tag{3.140}$$

por lo tanto el sistema discreto es:

$$x(k+1) = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^3}{6} \\ \frac{h^2}{2} \\ h \end{bmatrix} u(k)\tag{3.141}$$

$$y(k) = [1 \quad 0 \quad 0] x(k)$$



La función de transferencia discreta se obtiene entonces de la forma siguiente:

- 1.- Determinar la función temporal que corresponde a $G(s)/s$.
- 2.- Determinar la transformada z correspondiente (por uso de una tabla).
- 3.- Multiplicar por $1 - z^{-1}$ para encontrar la función de transferencia discreta con un mantenedor de orden cero.



Un modelo sencillo y normalizado de un motor eléctrico de corriente continúa en su representación en variables de estado está dado por:

$$\begin{aligned} A &= \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} & B &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C &= [0 \quad 1] & D &= 0 \end{aligned} \quad (3.116)$$

Por lo cual se puede encontrar la función de transferencia de este modo:

$$G_P(s) = \frac{\begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}}{[sI - A]} = \frac{\begin{bmatrix} s + 1 & 0 & -1 \\ -1 & s & 0 \\ 0 & 1 & 0 \end{bmatrix}}{\begin{bmatrix} s + 1 & 0 \\ -1 & s \end{bmatrix}} = \frac{-1(-1)}{s(s + 1)} = \frac{1}{s(s + 1)} \quad (3.117)$$



Encontrar ahora:

a).-

$$\frac{G_p(s)}{s} = \frac{1}{s^2(s+1)} \quad (3.118)$$

b).- Si se obtiene la transformada Z de $\frac{G_p(s)}{s} = ?$.

Primero se presentará la expresión $\frac{1}{s^2(s+1)}$ en fracciones parciales, es decir:

$$\frac{1}{s^2(s+1)} = \frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1} \quad (3.119)$$

si se encuentra la transformada Z :

$$Z\left(\frac{1}{s^2(s+1)}\right) = Z\left(\frac{1}{s^2}\right) - Z\left(\frac{1}{s}\right) + Z\left(\frac{1}{s+1}\right) \quad (3.120)$$



$$\begin{aligned} &= \frac{hz}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{z-e^{-h}} \\ &= \frac{ze^h}{ze^h-1} - \frac{z(z-h-1)}{(z-1)^2} \quad (3.121) \\ &= \frac{z[(z(h-1)+1)e^h+z-h-1]}{(ze^h-1)(z-1)^2} \end{aligned}$$

c).- Se multiplica por $(1-z^{-1})$ para encontrar la función de transferencia discreta con un mantenedor de orden cero.

$$\begin{aligned} H(z) &= (1-z^{-1})Z\left(\frac{G_p(s)}{s}\right) \\ &= (1-z^{-1})\left(\frac{z[(z(h-1)+1)e^h+z-h-1]}{(ze^h-1)(z-1)^2}\right) \quad (3.122) \\ &= \frac{(1-z^{-1})z[(z(h-1)+1)e^h+z-h-1]}{(z-1)^2(ze^h-1)} \\ &= \frac{[z(h-1)+1]e^h+z-h-1}{(z-1)(ze^h-1)} \end{aligned}$$



Otra forma de encontrar la función de transferencia discreta es con la ecuación 3.113 vista anteriormente:

$$H(z) = C(zI - \Phi)^{-1}\Gamma = \frac{\det \begin{bmatrix} zI - \Phi & -\Gamma \\ C & D \end{bmatrix}}{\det[zI - \Phi]} = \frac{Y(z)}{U(z)} \quad (3.123)$$

Para ello es necesario encontrar primero Φ y Γ esto se realiza en este ejemplo utilizando el teorema de Cayley-Hamilton

$$\det[\lambda I - Ah] = \det \left[\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} -h & 0 \\ h & 0 \end{bmatrix} \right] = \det \begin{bmatrix} \lambda + h & 0 \\ -h & \lambda \end{bmatrix} = (\lambda + h)\lambda = 0$$

$$\Rightarrow \lambda_1 = 0$$

$$\lambda_2 = -h$$

$$e^0 = \alpha_0 + \alpha_1 \lambda_1 \Rightarrow \alpha_0 = 1$$

$$e^{-h} = \alpha_0 + \alpha_1 \lambda_2 = 1 + \alpha_1(-h) \Rightarrow \alpha_1 = \frac{1 - e^{-h}}{h}$$

$$e^{Ah} = \alpha_0 I + \alpha_1 Ah = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1 - e^{-h}}{h} \begin{bmatrix} -h & 0 \\ h & 0 \end{bmatrix} = \begin{bmatrix} e^{-h} & 0 \\ 1 - e^{-h} & 1 \end{bmatrix} = \Phi \quad (3.124)$$



y se encuentra Γ a partir de:

$$\begin{aligned} \Gamma &= \int_0^h e^{Ah} B dh = \int_0^h \begin{bmatrix} e^{-h} & 0 \\ 1 - e^{-h} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} dh \\ &= \int_0^h \begin{bmatrix} e^{-h} \\ 1 - e^{-h} \end{bmatrix} dh = \begin{bmatrix} 1 - e^{-h} \\ e^{-h} + h - 1 \end{bmatrix} \end{aligned} \quad (3.125)$$

ahora la función de transferencia discreta se obtiene de la siguiente forma:

$$H(z) = [0 \quad 1] \begin{bmatrix} z - e^{-h} & 0 \\ e^{-h} - 1 & z - 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 - e^{-h} \\ e^{-h} + h - 1 \end{bmatrix} = \frac{\begin{bmatrix} zI - \Phi & -\Gamma \\ C & D \end{bmatrix}}{[zI - \Phi]} = \frac{y(z)}{u(z)} \quad (3.126)$$

$$H(z) = \frac{\det \begin{bmatrix} z - e^{-h} & 0 & e^{-h} - 1 \\ e^{-h} - 1 & z - 1 & 1 - h - e^{-h} \\ 0 & 1 & 0 \end{bmatrix}}{\det \begin{bmatrix} z - e^{-h} & 0 \\ e^{-h} - 1 & z - 1 \end{bmatrix}}$$

$$H(z) = \frac{(z - e^{-h})(-1 + h + e^{-h}) + (e^{-h} - 1)(e^{-h} - 1)}{(z - e^{-h})(z - 1)} = \frac{ze^{-h} - he^{-h} - e^{-h} + zh - z + 1}{-ze^{-h} + e^{-h} + z^2 - z}$$

$$H(z) = \frac{[z(h-1)+1]e^h + z - h - 1}{(ze^h - 1)(z - 1)} \quad (3.127)$$

que es el resultado igual al que se encontró en la ecuación 3.122.



Poniendo la función de transferencia discreta en otros términos queda como:

$$H(z) = \frac{z(e^{-h} + h - 1) + 1 + e^{-h} - he^{-h}}{z^2 - (1 + e^{-h})z + e^{-h}} \quad (3.128)$$

A partir de la función de transferencia $H(z)$ se puede encontrar la ecuación de salida discreta en términos de:

$$y(kh) = f(y(k-1), y(k-2), u(k-1), u(k-2)) \quad (3.129)$$

si dividimos la función de transferencia obtenida anteriormente entre z^2 se tiene lo siguiente:

$$H(z) = \frac{z^{-1}(e^{-h} + h - 1) + z^{-2}(1 + e^{-h} - he^{-h})}{1 - (1 + e^{-h})z^{-1} + e^{-h}z^{-2}} = \frac{Y(z)}{U(z)} \quad (3.130)$$



Despejando $Y(z)$ en términos de $U(z)$ de la función de transferencia anterior queda así

$$Y(z) - (1 + e^{-h})z^{-1}Y(z) + e^{-h}z^{-2}Y(z) = (e^{-h} + h - 1)z^{-1}U(z) + (1 - e^{-h} - he^{-h})z^{-2}U(z) \quad (3.131)$$

si aplicamos la siguiente transformación

$$\begin{aligned} zY(z) & \text{---} \text{---} \text{---} \text{---} \text{---} y(k+1) \\ z^{-1}Y(z) & \text{---} \text{---} \text{---} \text{---} \text{---} y(k-1) \end{aligned} \quad (3.132)$$

nos queda en terminos de la ecuación de diferencias de salida $y(kh)$

$$y(kh) = (1 + e^{-h})y(k-1) - e^{-h}y(k-2) + (h + e^{-h} - 1)u(k-1) + (1 - (1+h)e^{-h})u(k-2) \quad (3.133)$$

por simplicidad si se toma $h = 1$ la ecuación de $y(kh)$ se reduce a:

$$y(k) = 1,3678y(k - 1) - 0,3678y(k - 2) + 0,3678u(k - 1) + 0,2642u(k) \quad (3.134)$$

y se puede establecer un algoritmo para calcular $y(k)$ si se toma como entrada un escalón unitario como entrada:

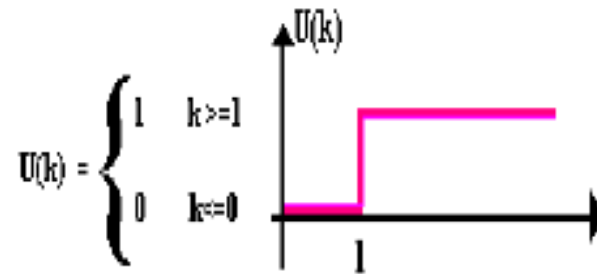
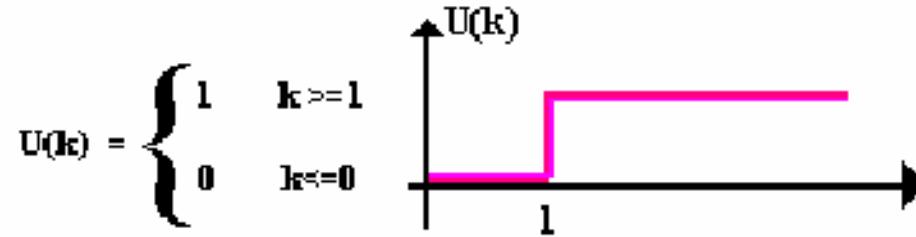


Figura 3.13: Gráfica de entrada escalón unitario.

g).- Establecer un algoritmo para calcular $y(k)$ si se toma como entrada :



k	u(k)	y(k)
0	0	0
1	1	0
2	1	0.36
3	1	...
4	...	

Grafica de $y(k)$



Ejercicios de tarea previos al examen parcial: Del Libro de Karl J. Astrom Computer- Controlled Systems

2.1

2.2

2.3

2.5

2.8

2.9

2.11

2.12

9.8.4 Obtener la representación en espacio de estados discreto de los siguientes sistemas:

a)

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -8 & -10 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = [1 \ 0] x$$

b)

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -3 & -7 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$
$$y = [1 \ 0] x$$

9.8.5 Obtener la función de transferencia pulso $G(q)$ y $G(z)$ de los siguientes sistemas:

a)

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & -5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = [1 \ 1] x$$

b)

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = [1 \ 0] x$$

9.8.7 Realice los pasos analíticos necesarios para obtener la función de transferencia $G(z)$ de:

a) $G(s) = \frac{s+1}{s^2+4s+5}$.

b) $G(s) = \frac{1}{s^2+0.34s+0.02}$.



3.1 Respuesta al impulso y al escalón de sistemas LTI

El estudio de la respuesta al impulso (Delta de Dirac) se debe a que la función de transferencia de un sistema continuo en el tiempo es la transformada de Laplace de su respuesta al impulso con condiciones iniciales nulas. Es muy común estudiar el comportamiento de la dinámica de los sistemas usando la respuesta al escalón, es decir $U(s) = 1/s$, por lo que

$$Y(s) = G(s) \frac{1}{s}$$

Es muy útil definir una serie de parámetros que describen algunas propiedades relevantes de la dinámica de los sistemas. Para estas definiciones, consideraremos funciones de transferencia estables teniendo como respuesta al escalón la Figura 8:

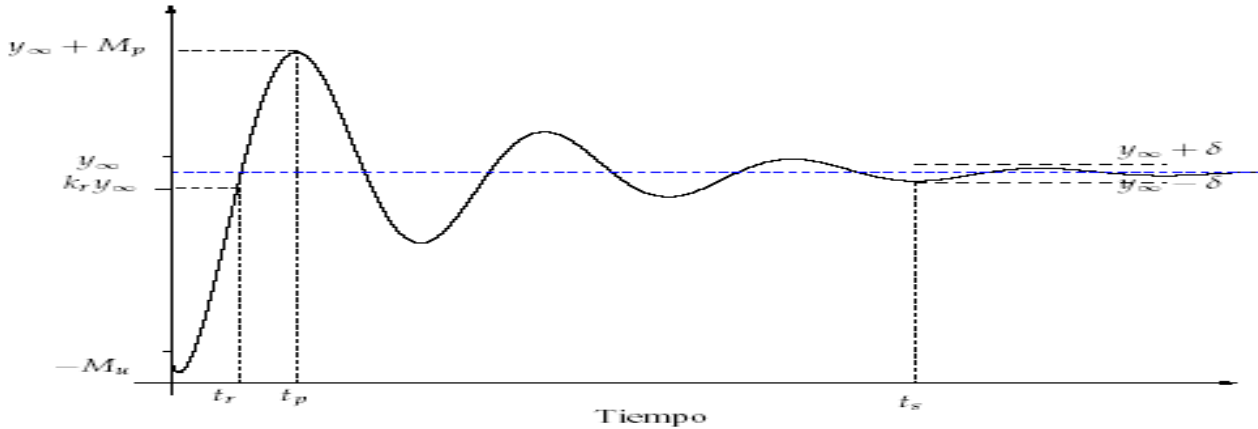


Figura 8: Indicadores de la respuesta escalón

- **valor en el régimen estacionario, y_∞ :** el valor final de la respuesta al escalón (esto no tiene sentido si el sistema tiene polos en el SPD¹).
- **Tiempo de crecimiento, t_r :** el tiempo que transcurre hasta el instante en el cual la respuesta al escalón alcanza, la primera vez, el valor $k_r y_\infty$. La constante k_r varía según el autor, comúnmente se toma tanto 0.9 o 1.
- **Sobre error, M_p :** el máximo valor por el que la respuesta al escalón excede su valor final. Generalmente se expresa como un porcentaje de y_∞ .
- **subvalor, M_w :** el máximo (valor absoluto) por el que la respuesta al escalón pasa por debajo del cero.
- **Tiempo de establecimiento, t_s :** el tiempo transcurrido hasta que la respuesta al escalón ingresa (sin dejarlo en tiempo subsiguiente) a una banda $\pm \delta$, alrededor del valor final. Esta banda δ , generalmente se define como un porcentaje de y_∞ , 2% a 5%

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

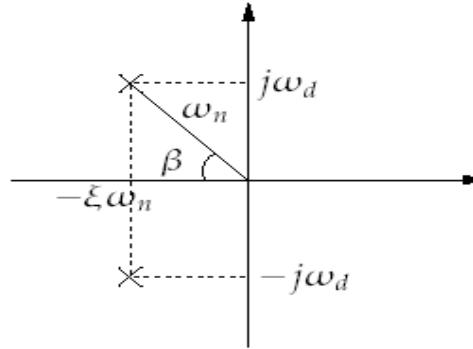


Figura 9: Función transferencia y ubicación de un par de polos complejos conjugados

Para el caso de un par de polos complejos conjugados, Figura 9, donde ξ ($0 < \xi < 1$) es conocido como el *factor de amortiguamiento* y ω_n , como la *frecuencia natural*. Definimos también la *frecuencia natural de amortiguamiento*, ω_d como

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

Los polos complejos conjugados de este sistema, s_1 y s_2 , pueden expresarse la la forma

$$s_{1,2} = -\xi\omega_n \pm j\omega_d = \omega_n e^{\pm j(\pi - \beta)}$$

Las características de la respuesta al escalón unitario de este sistema nos lleva a calcular los indicadores descriptos en la Figura 8:



Tiempo de crecimiento

Para este caso tomaremos $k_r = 1$, y obtenemos

$$t_r = \frac{\pi - \beta}{\omega_d} \approx \frac{1.8}{\omega_n}$$

Sobre valor

El máximo valor de la salida, M_p ocurre en el tiempo $t_p = \frac{\pi}{\omega_d} = \frac{T_d}{2}$, entonces

$$M_p = e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}}$$

Tiempo de establecimiento

Para un error en el régimen estacionario de 1% tenemos que

$$e^{-\xi\omega_n t_s} = 0.01, \text{ por lo que } t_s = \frac{4.6}{\xi\omega_n}$$



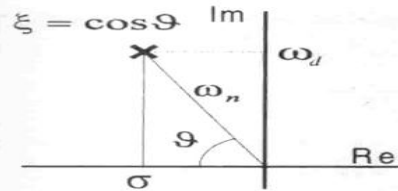
CUV y TT-BUAP CONTROL I



RESPUESTA TEMPORAL DE UN SISTEMA DE SEGUNDO ORDEN AL ESCALÓN UNITARIO

$$G(S) = \frac{K_c \omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2}$$

Polos en $-\sigma \pm \omega_d j$ con $\sigma = \xi \omega_n$ $\omega_d = \omega_n (1 - \xi^2)^{1/2}$



- $\xi < 0$ Sistema inestable.
- $0 < \xi < 1$ Sistema estable subamortiguado (polos complejos).
- $1 \leq \xi$ Sistema estable sobreamortiguado (polos reales).

Figura 4.0-2

CASO DE SISTEMA SUBAMORTIGUADO ($0 < \xi < 1$)

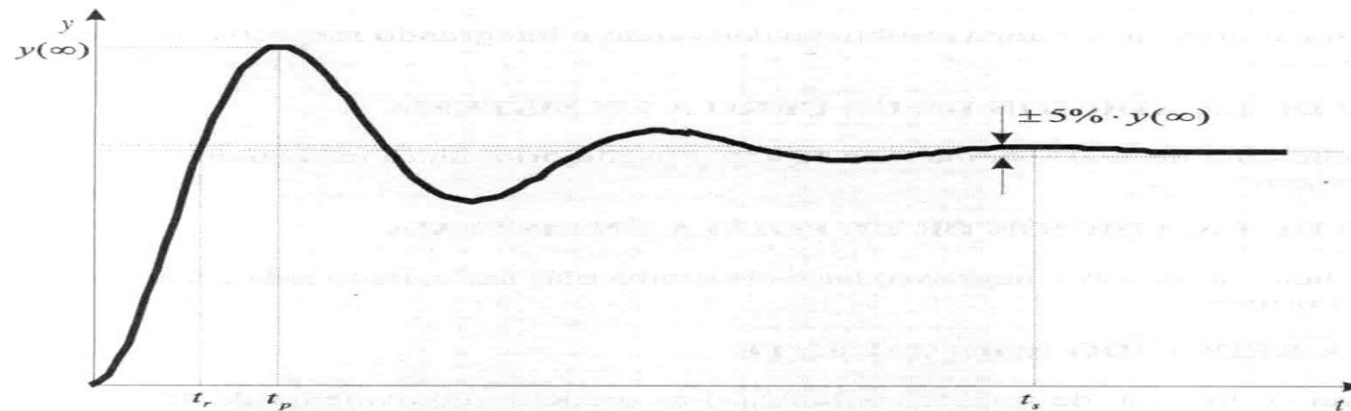


Figura 4.0-3

Figure 4.17

Pole plot for an underdamped second-order system

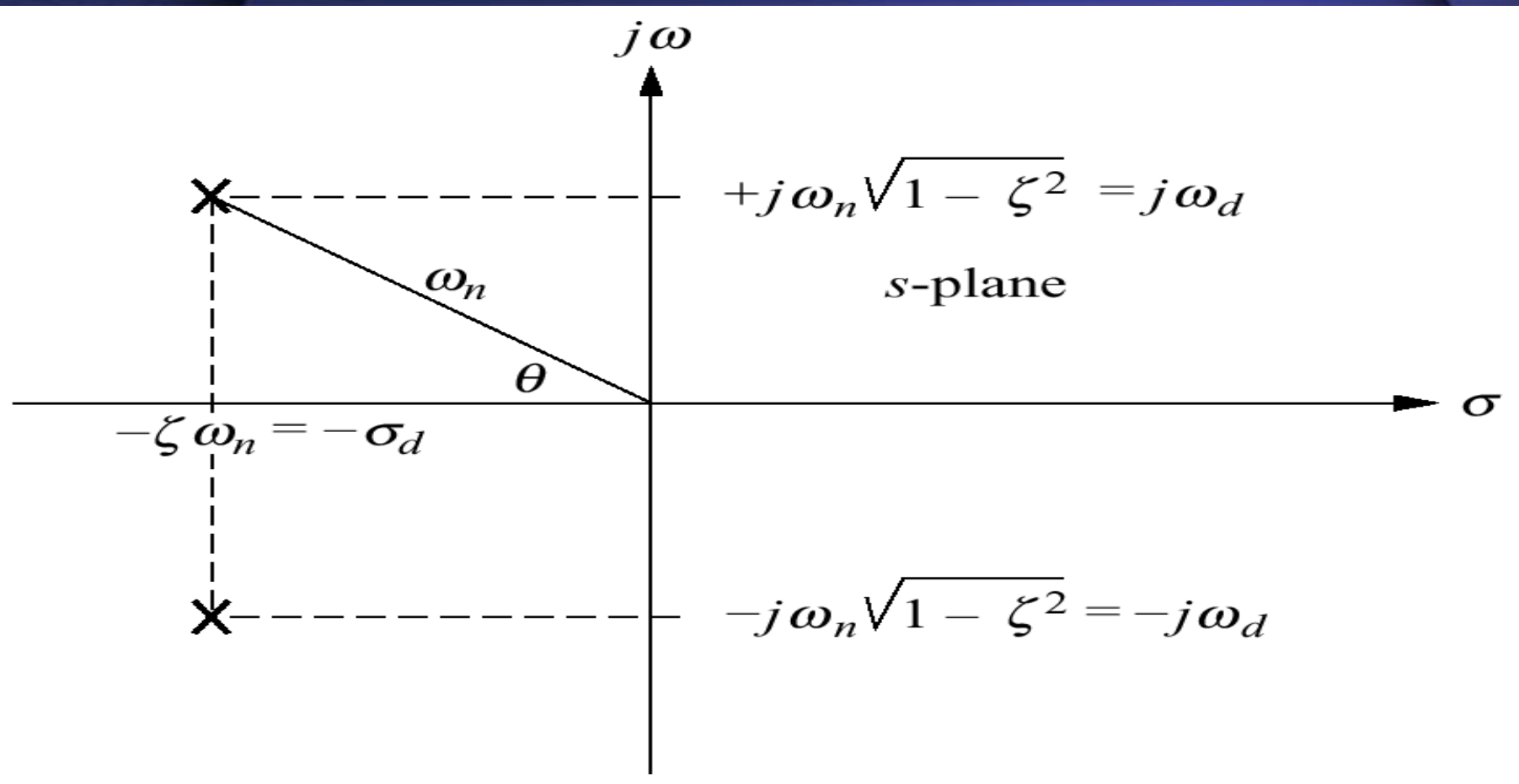


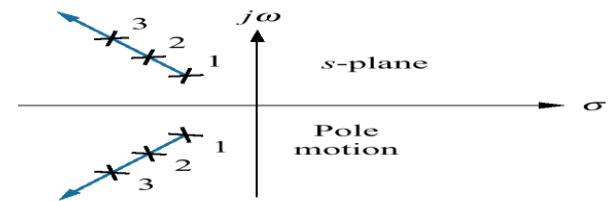
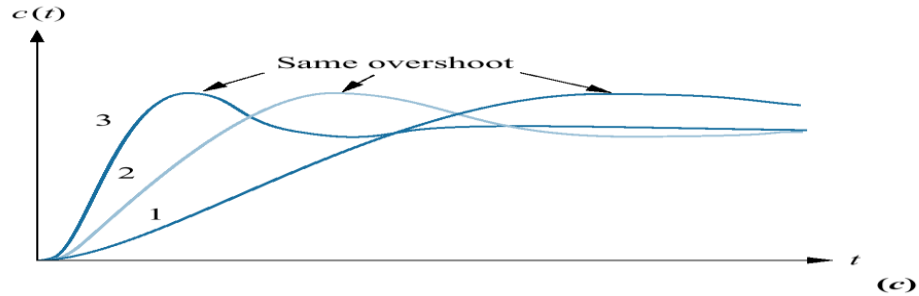
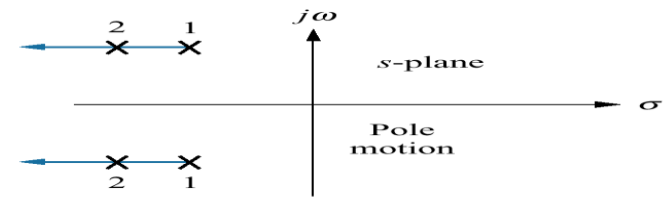
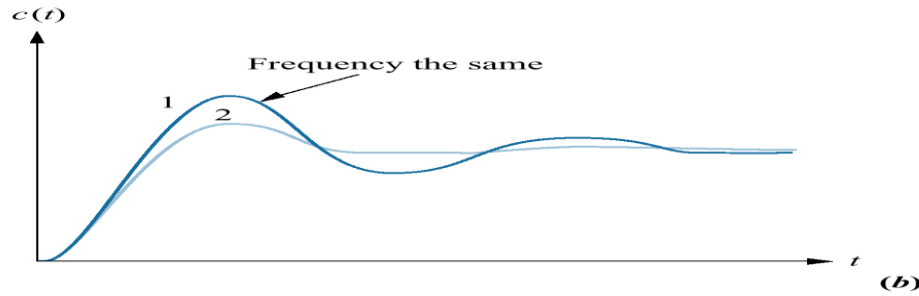
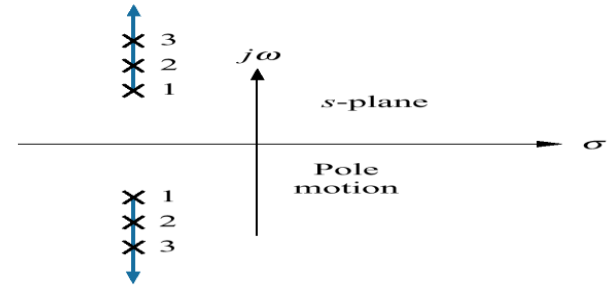
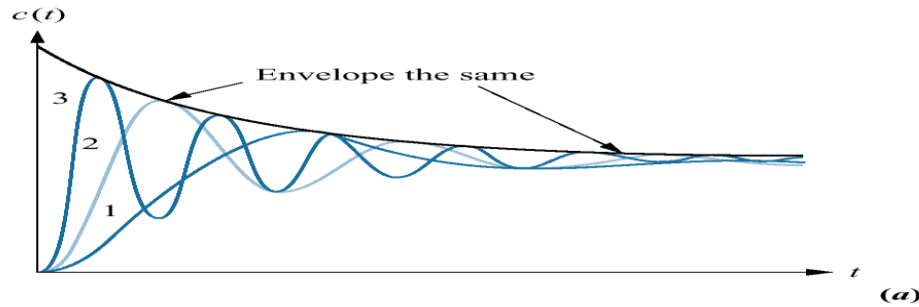
Figure 4.19

Step responses of second-order underdamped systems as poles move:

a. with constant real part;

b. with constant imaginary part;

c. with constant damping ratio





CUV y TT - BUAP CONTROLES



$$y(t) = K_c \left(1 - \frac{1}{\sqrt{1 - \xi^2}} e^{-\sigma t} \sin(\omega_d t + \theta) \right)$$

Ganancia estática: valor final $K_c = y(\infty)$

Porcentaje de sobreoscilación: $M_p = e^{\frac{-\pi}{\tan \theta}} 100$

Valor de pico: $y_p = y_\infty \left(1 + \frac{M_p}{100} \right)$

Tiempo de pico: $t_p = \frac{\pi}{\omega_d}$

Tiempo de subida: $t_r = \frac{\pi - \theta}{\omega_d}$

Tiempo de establecimiento: $t_s = \frac{\pi}{\sigma}$

Pendiente en el origen: nula.

La respuesta al impulso y rampa se obtiene derivando e integrando respectivamente la respuesta al escalón.

EFECTO DE LA ADICIÓN DE UN CERO A UN SISTEMA

Un cero adicional hace al sistema más rápido y oscilatorio, tanto más cuanto más cerca esté el cero del origen.

EFECTO DE LA ADICIÓN DE UN POLO A UN SISTEMA

Un polo adicional (de valor negativo) hace al sistema más lento, tanto más cuanto más cerca esté el polo del origen.

SISTEMA REDUCIDO EQUIVALENTE

Un sistema de función de transferencia $M_f(s)$ es reducido equivalente de $M(s)$ si teniendo el primero menor número de polos y/o ceros que el segundo las respuestas temporales de ambos son similares.

Figure 4.7

Second-order systems, pole plots, and step responses

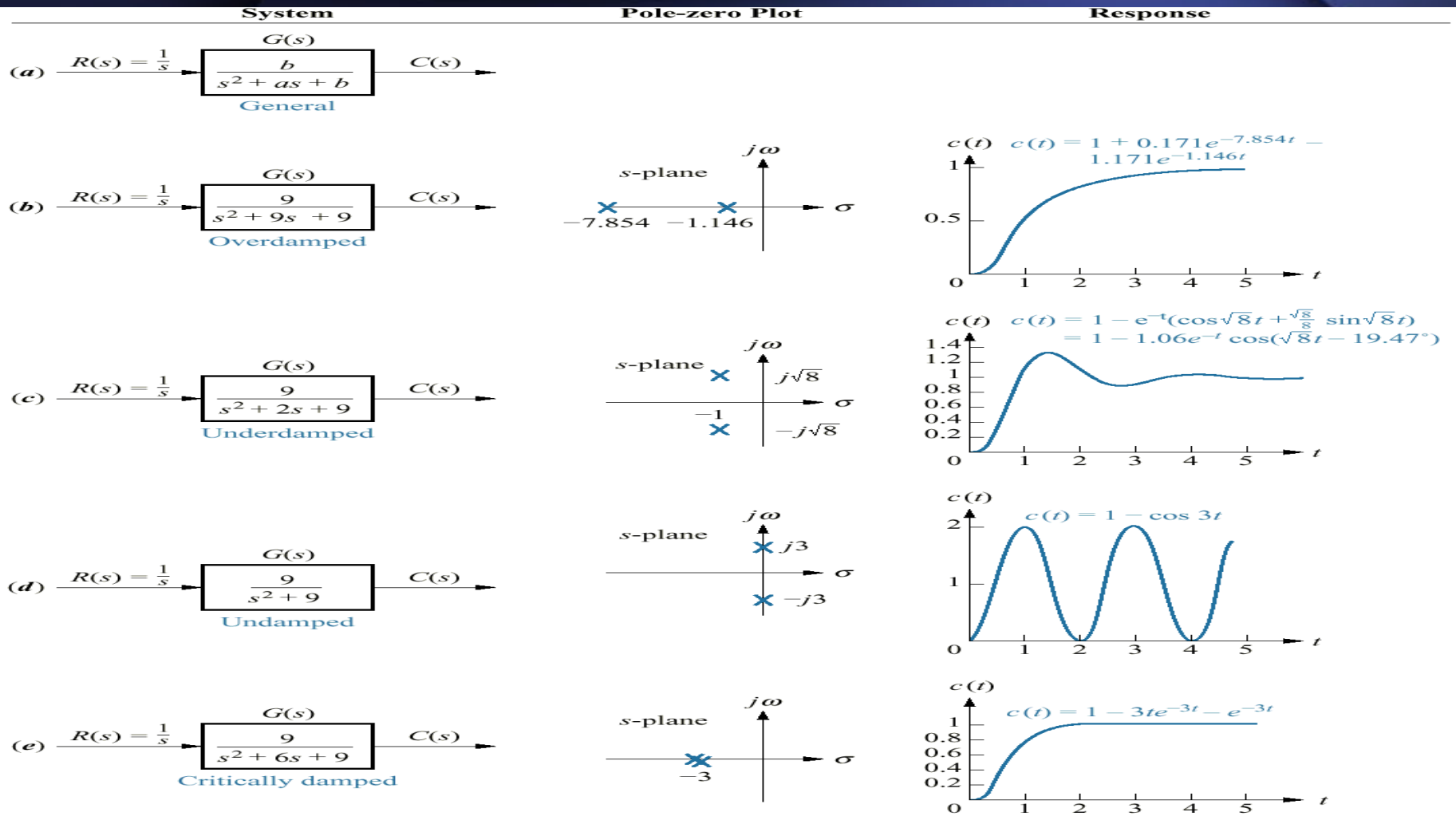
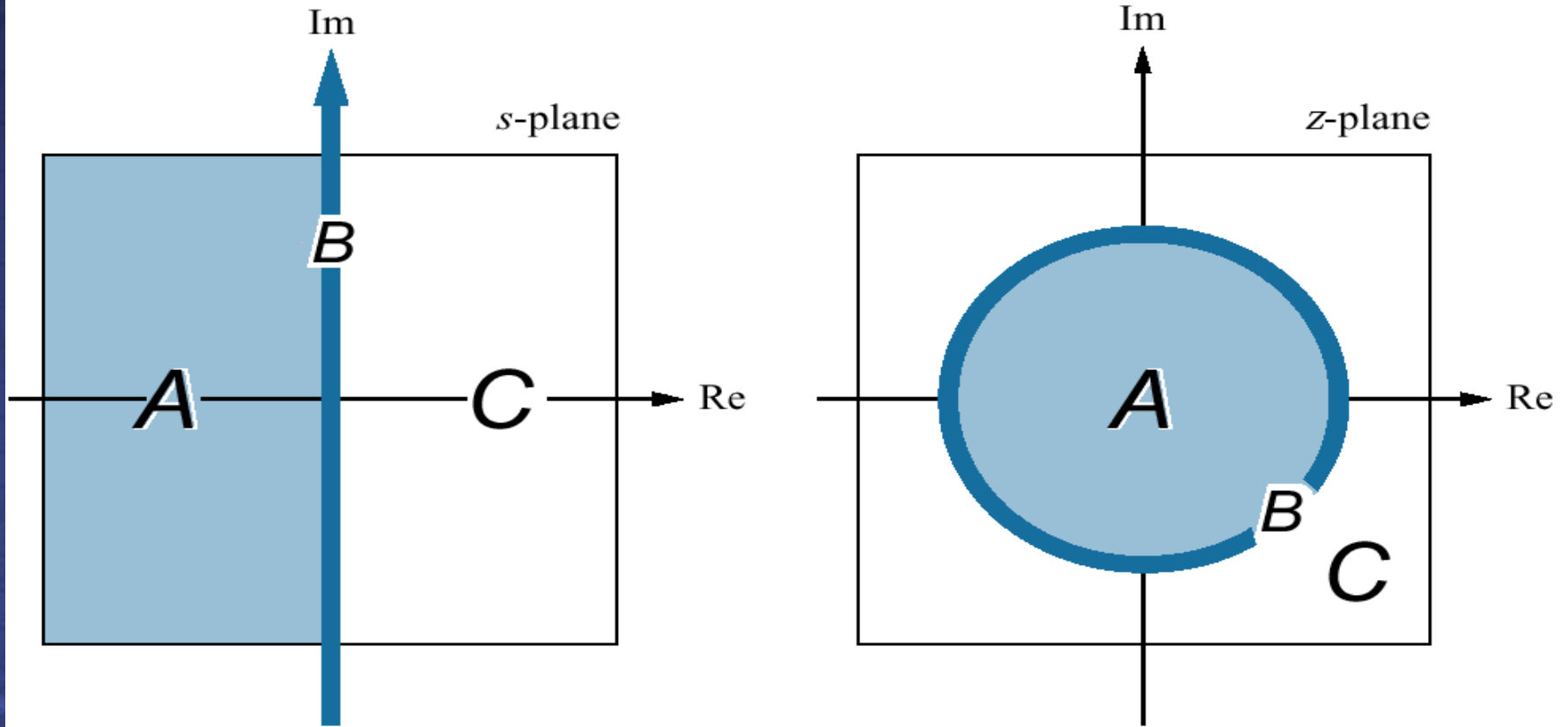


Figure 13.13

Mapping regions of the s -plane onto the z -plane

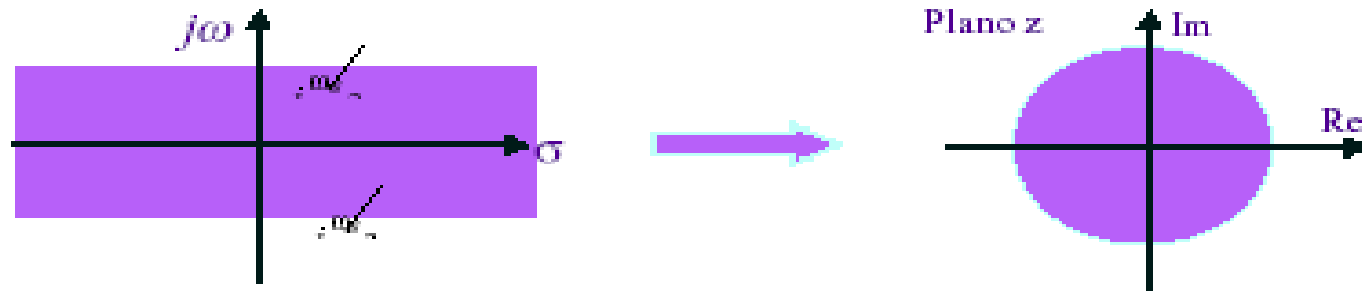


Transformación invariante al impulso

El algoritmo de discretización es:

$$z = e^{sT}$$

Esta transformación parte el plano s en franjas que pueden generar fácilmente problemas de aliasing





Para efectuar la transformación se expande la función $H(s)$ en fracciones parciales de la forma:

$$H(s) = \frac{A_1}{s+a_1} + \frac{A_2}{s+a_2} + \dots + \frac{A_n}{s+a_n}$$

Y se sustituye cada término s por otros de la forma:

$$H(z) = \frac{A_1 z}{z + e^{-a_1 T}} + \frac{A_2 z}{z + e^{-a_2 T}} + \dots + \frac{A_n z}{z + e^{-a_n T}}$$



Características

- $D(z)$ preserva la respuesta al impulso de $D(s)$.
- Si $D(s)$ es estable $D(z)$ posiblemente también lo sea.
- No preserva la respuesta en frecuencia.
- Las frecuencias transformadas en $D(z)$ que son múltiplos de ω_s pueden ocasionar aliasing.
- Si $D(s)$ es una función complicada se requiere de expandir la función en fracciones parciales.

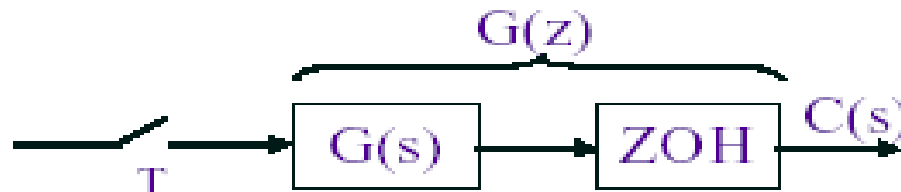
Transformación invariante al impulso con amarre artificial

El algoritmo de discretización es:

$$D(z) = \xi \left\{ \frac{1 - e^{-sT}}{s} D(s) \right\} = (1 - z^{-1}) \underbrace{\xi \left\{ \frac{D(s)}{s} \right\}}$$

Transformación invariante
al impulso

En diagramas a bloques:





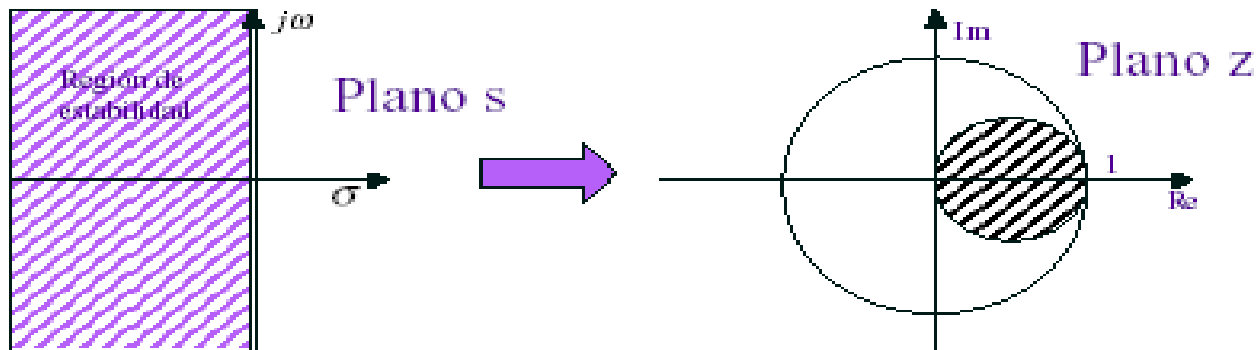
Características

- No preserva las respuestas al impulso y en frecuencia.
- Si $D(s)$ es estable $D(z)$ también lo es.
- Requiere $D(s)$ sea expandida en fracciones parciales.

Transformación por diferencias finitas hacia atrás (backward)

Algoritmo de transformación

$$s = \frac{1}{T} \cdot \frac{1}{1 - z^{-1}}$$





Como se puede apreciar, esta transformada comprime la región estable del plano s en una zona reducida del plano z , lo cual ocasiona que la zona de altas frecuencias del plano s no sean mapeadas a la zona de altas frecuencias del plano z

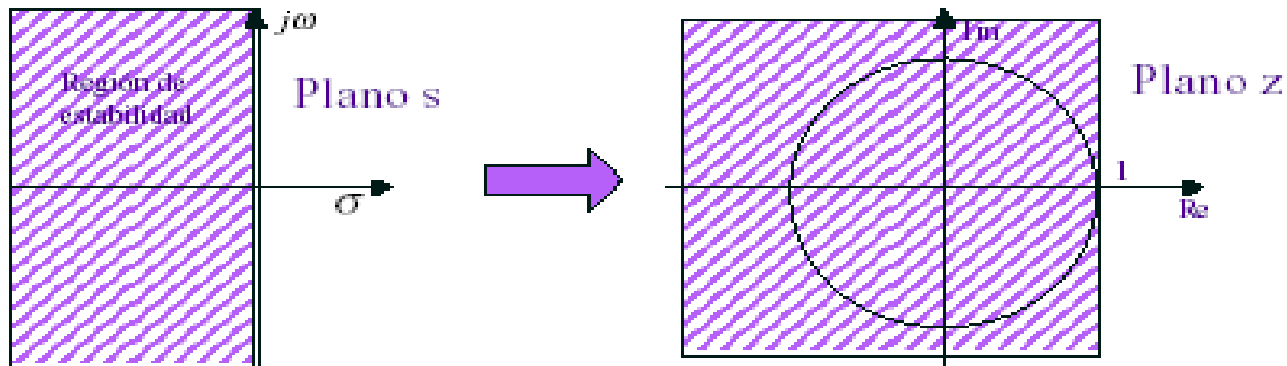
Características

- No requiere factorización de la función de transferencia.
- Si $D(s)$ es estable $D(z)$ también lo es.
- No preserva ni la respuesta al impulso ni la respuesta en frecuencia.

Transformación por diferencias finitas hacia adelante (forward)

Algoritmo de transformación

$$s = \frac{1}{T} \cdot \frac{1 - z^{-1}}{z^{-1}}$$





Como se puede apreciar, esta transformada traslada el origen del plano s a $z=-1$ lo que ocasiona que una función estable en el plano s puede comportarse como inestable en el plano z

Características

- No requiere factorización de la función de transferencia.
- Si $D(s)$ es estable $D(z)$ no necesariamente lo es.
- No preserva ni la respuesta al impulso ni la respuesta en frecuencia.



TRANSFORMACION BILINEAL - METODO DE TUSTIN

Un método de digitalización es el tratar el problema como si fuera una integración numérica. Supóngase que:

$$\frac{U(s)}{Y(s)} = G(s) = \frac{1}{S}$$

esto es, integración pura. Por tanto,

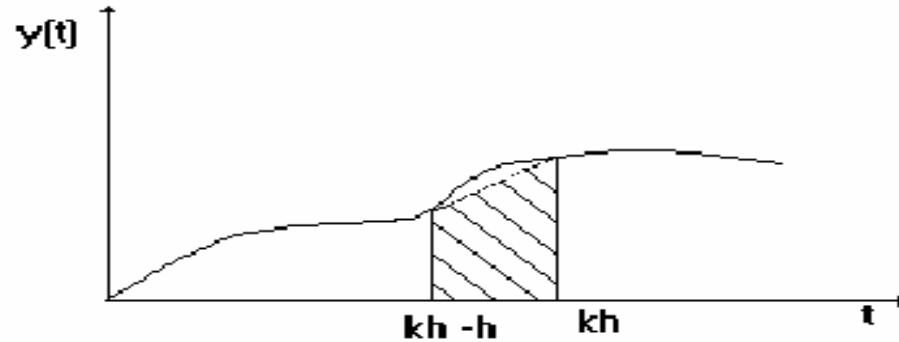
$$\begin{aligned} U(kh) &= \int_0^{kh-h} y(t) dt + \int_{kh-h}^{kh} y(t) dt \\ &= U(kh - h) + \text{area bajo } y (+) \text{ sobre el ultimo } h \end{aligned}$$

donde h es el periodo de muestreo. Para abreviar $u(kh)$ generalmente se escribe $u(k)$ y la tarea en cada paso es emplear integración trapezoidal, esto es, aproximar $y(t)$ por medio de una recta entre dos muestras ver (figura). Por tanto, la ecuación anterior se convierte en:

$$U(kh) = U(kh - h) + \frac{h}{2} [y(kh - h) + y(kh)]$$



CUV_yTT-BUAP



o, tomando la transformación z:

$$U(kh) - U(kh - k) = \frac{h}{2} [y(kh) + y(kh - h)]$$

$$U(z) - U(z)z^{-1} = \frac{h}{2} [y(z) + y(z)z^{-1}]$$

$$U(z)(1 - z^{-1}) = \frac{h}{2} y(z)(1 + z^{-1})$$

Para:

$$G(s) = \frac{a}{s + a}$$

$$\Rightarrow U(z) = \frac{h}{2} \frac{(1 + z^{-1})}{1 - z^{-1}} Y(z)$$

$$\therefore \frac{U(z)}{Y(z)} = \frac{1}{\frac{2}{h} \frac{(1 - z^{-1})}{(1 + z^{-1})}}$$



la aplicación de la misma aproximación de integración da:

$$G(z) = \frac{a}{\frac{2(1-z^{-1})}{h(1+z^{-1})} + a}$$

de hecho la situación

$$S = \frac{2(1-z^{-1})}{h(1+z^{-1})}$$

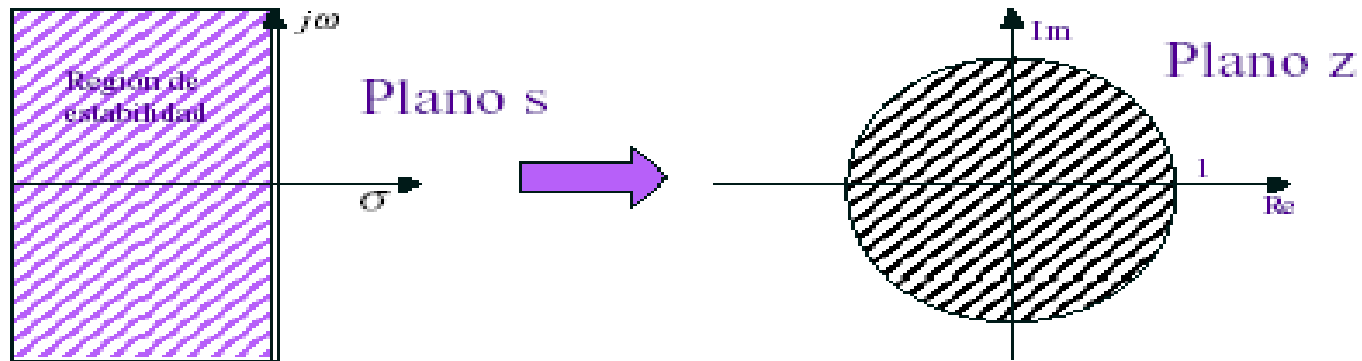
en cualquier $G(s)$ conduce a una $G(z)$ basada en la fórmula de integración trapezoidal.

Esto denomina aproximación de tustin o bilineal.

Transformación bilineal (Tustin)

Algoritmo de transformación

$$s = \frac{2}{T} \cdot \frac{1 + z^{-1}}{1 - z^{-1}}$$





Como se puede apreciar, esta transformada comprime la región estable del plano s dentro del círculo unitario del plano z .

Características

- No requiere factorización de la función de transferencia.
- Si $D(s)$ es estable $D(z)$ también lo es.
- Preserva tanto la respuesta al impulso como la respuesta en frecuencia..
- Crea una distorsión en la zona de altas frecuencias



PROPIEDADES DE LA TRANSFORMADA BILINEAL

- 1.- Es fácil de aplicar
- 2.- Transformar todo el semiplano izquierdo del plano dentro del círculo unidad en el plano Z.
- 3.- Si $G(s)$ es estable también lo es $G(z)$.

Ejemplo :

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{sustituyendo } s = \frac{2}{h} \frac{1-z^{-1}}{1+z^{-1}}$$



CUV_yTT-BUAP



$$G(s) = \frac{K\omega_n^2}{\left[\frac{2}{h} \frac{(1-z^{-1})}{1+z^{-1}} \right]^2 + 2\xi\omega_n \left(\frac{2}{h} \frac{(1-z^{-1})}{1+z^{-1}} \right) + \omega_n^2}$$

factorizando:

$$G(s) = \frac{K\omega_n^2(1+z^{-1})^2}{\left(\frac{2}{h}\right)^2 (1-z^{-1})^2 + 2\xi\omega_n \left(\frac{2}{h}\right) (1-z^{-1})(1+z^{-1}) + \omega_n^2(1+z^{-1})}$$

$$G(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{a_0 + a_1z^{-1} + a_2z^{-2}}$$

$$b_0 = h^2 K \omega^2$$

$$b_1 = 2h^2 K \omega^2$$

$$b_2 = h^2 K \omega^2$$

$$a_0 = 4\xi h \omega + h^2 \omega^2 + 4$$

$$a_1 = 2h^2 \omega^2 - 8$$

$$a_2 = h^2 \omega^2 + 4 - 4\xi h \omega$$



3.7 Selección de frecuencia de muestreo.

Un periodo de muestreo demasiado grande hace imposible la reconstrucción de la señal continua. Un periodo de muestreo demasiado corto aumentara la carga de la computadora. El problema de la selección analiza la relación entre la frecuencia de muestreo y los polos del sistema continuo.

Es útil caracterizar el periodo de muestreo por una variable sin dimensiones y esto tiene una fácil interpretación física. En los sistemas oscilatorios es natural normalizar respecto al periodo de oscilación; en los sistemas no oscilatorios, el tiempo de subida es un factor de normalización natural.

Introduzcamos N_r , como el número de periodos de muestreo por tiempo de subida, Un periodo de muestreo por tiempo de subida,

$$N_r = \frac{T_r}{h}$$

Para sistemas de primer orden, el tiempo de subida es igual a la constante de tiempo. Por tanto es razonable escoger N_r entre 2 y 4. Para un sistema de segundo orden con un amortiguamiento ζ y una frecuencia natural w_o el tiempo de subida viene dado por:

$$T_r = w_o^{-1} e^{\phi} \tan \phi$$

donde $\zeta = \cos \phi$. Para un amortiguamiento alrededor de $\zeta = 0.7$ esto da

$$w_o h = 0.5 - 1 \quad \text{entre } 0.5 \text{ y } 1$$

donde w_o esta medido en radianes por segundo. Las figuras 3.6 y 3.8 ilustran la elección del periodo de muestreo para diferentes señales. Es razonable escoger el periodo de muestreo de forma que

$$N_r = \frac{T_r}{h} = 2 - 4 \quad \text{entre } 2 \text{ y } 4$$

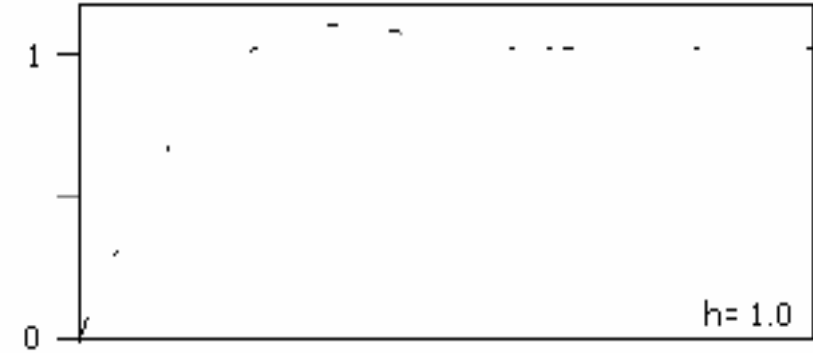
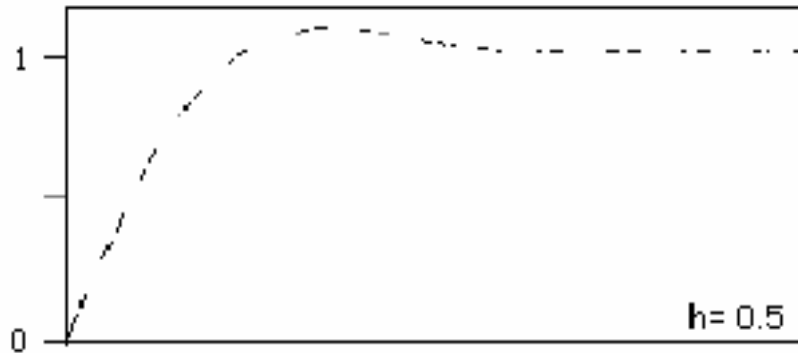
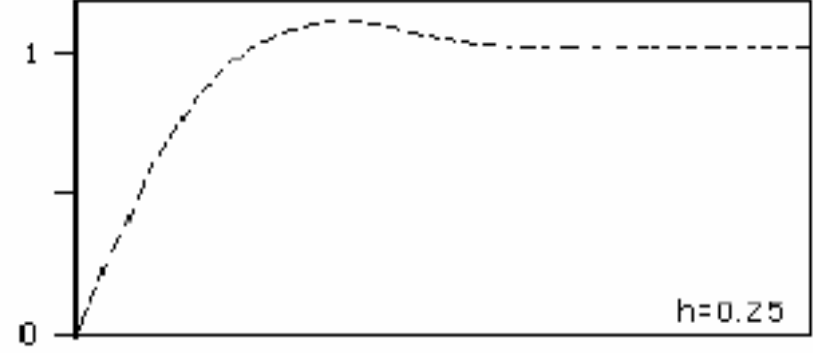
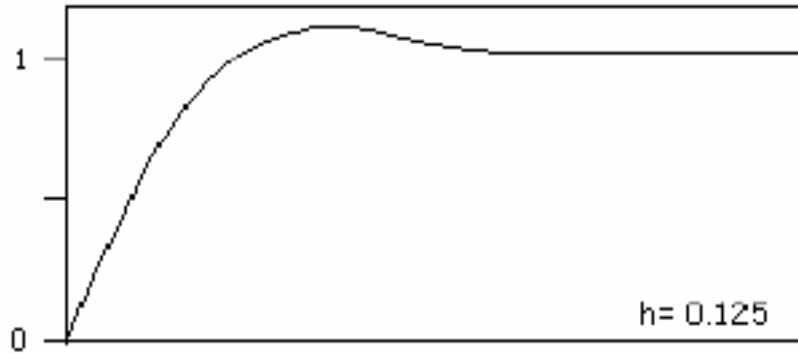


fig. 3.6



CUV_yTT-BUAP



comportamiento requerido

dinámica propia del sistema

perturbaciones

actuadores

sensores

cómo fue modelado

Período de muestreo muy grande

imposibilita la reconstrucción

Mucho tiempo en lazo abierto

Período de Muestreo muy corto

Incrementa la carga del computador

Introduce errores numéricos

Si el sistema tiene retardo

$$T \approx \frac{1}{4} - \frac{1}{8} t_d$$

Dependiendo del tipo de proceso

Caudal 1seg

Presión 5 seg

Nivel 10 seg

Temperatura 20 seg

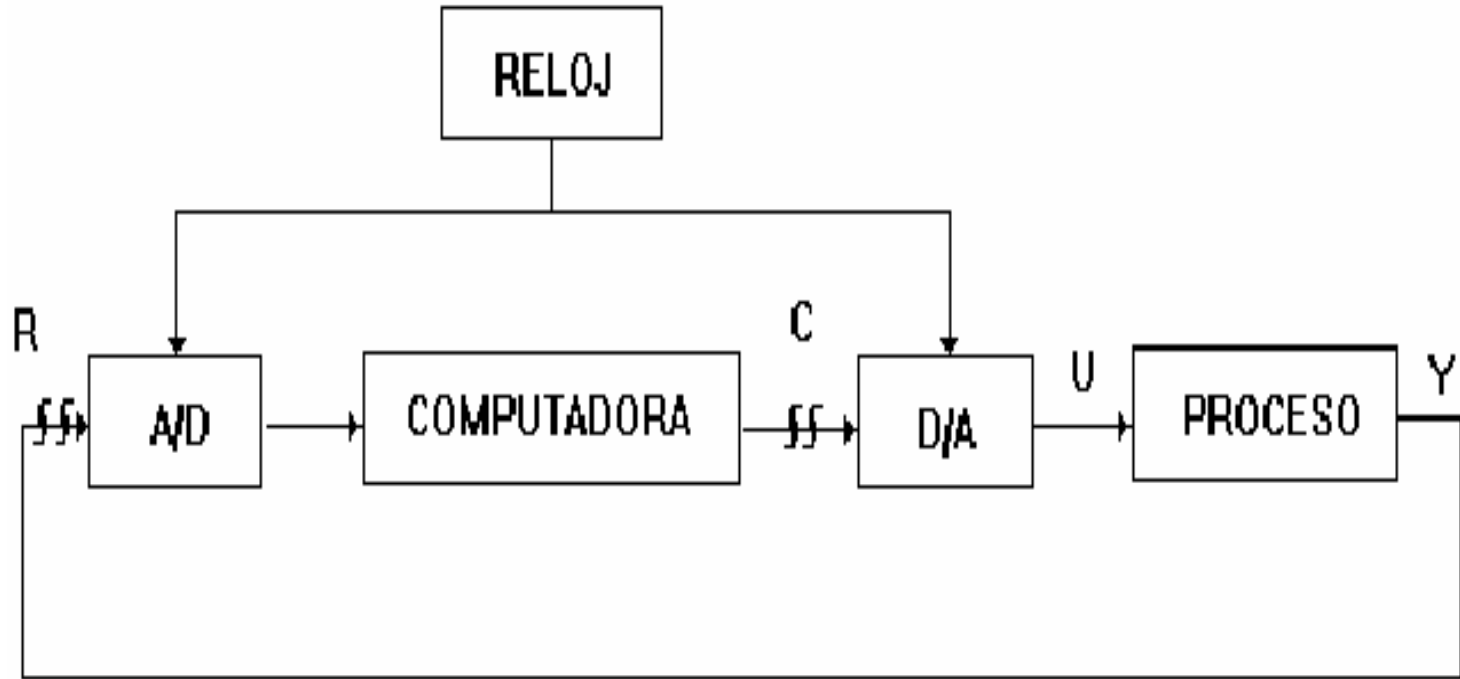


Figura 4.1 Diagrama esquemático de un sistema controlado con computadora.



secuencia de operaciones en un sistema controlado por computadora. El algoritmo de la computadora es el siguiente:

- 1.-Esperara hasta que ocurra un pulso de reloj.
- 2.-Realizar una conversión analógica a digital.
- 3.-Calcular las variables de control.
- 4.- Realizar la conversión digital a analógica.
- 5.-Actualizar el estado del regulador.
- 6.-Volver de nuevo al punto 1.

Ya que las operaciones en una computadora tardan algún tiempo en realizarse, existe un retardo entre los pasos 2 y 4. La relación entre las diferentes señales en el sistema se muestran en la figura 4.2 Cuando la ley de control se realiza con un computadora es muy importante estructurar el programa para minimizar el tiempo de calculo que requiere el paso 3

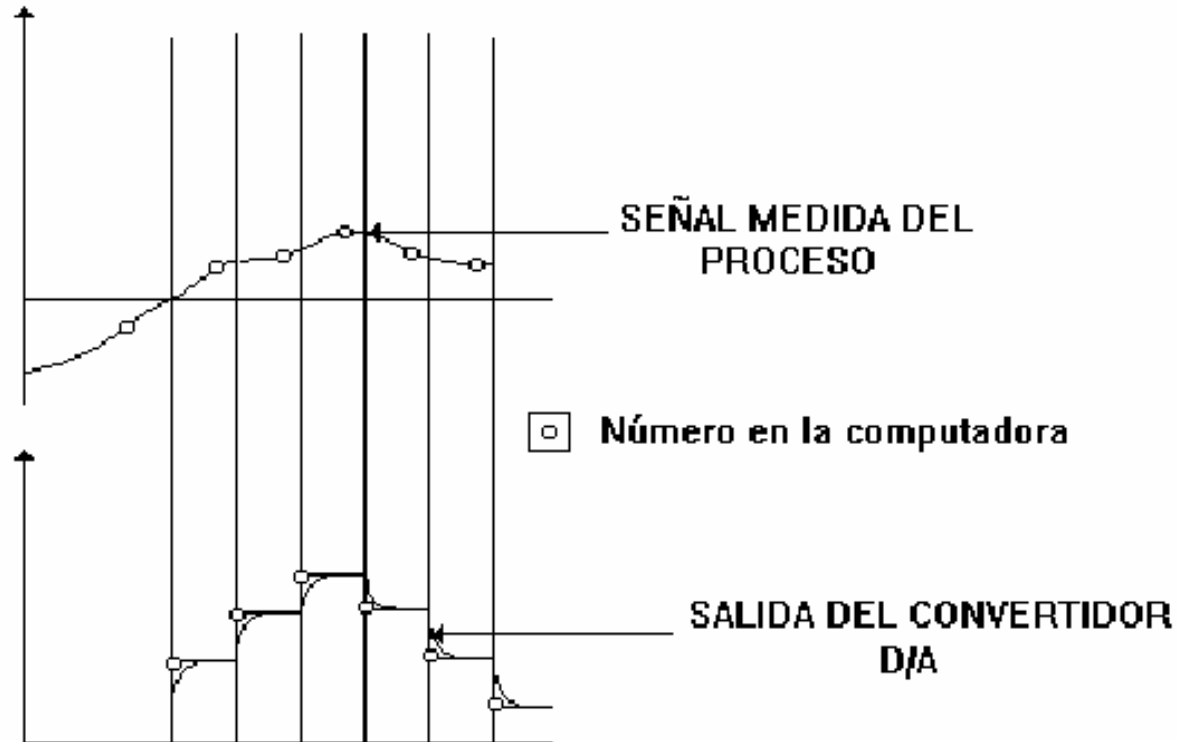


Figura 4.2 relación entre la señal medida, la señal de control y su representación en la computadora



Implementación de un control digital.

Una vez que se ha llegado a la función de transferencia de un controlador digital que satisfaga los requisitos de operación de un sistema, es necesario transformar su función de transferencia en un algoritmo numérico.

Dada la función de transferencia del controlador
$$\frac{M(z)}{E(z)} = k \cdot \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}}$$

multiplicando en cruz
$$M(z) \cdot (a_0 + a_1 z^{-1}) = E(z) \cdot k \cdot (b_0 + b_1 z^{-1})$$

aplicando la transformada z inversa y aplicando la propiedad del corrimiento

$$m(n) = k \cdot \left(\frac{b_0}{a_0} \cdot e(n) + \frac{b_1}{a_0} \cdot e(n-1) \right) - \frac{a_1}{a_0} m(n-1)$$



Pseudo código de un control digital.

Inicio

inicializar variables y constantes.

a_1, a_2, b_1

$e(n)=0$

$e(n-1)=0$

$u(n)=0$

$u(n-1)=0$

Capturar el valor de la variable

$c(nT)=ADC$

calcular el error:

$e(n)=r(nT)-c(nT)$

evaluar la acción de control:

$m(nT)=a_1 * e(n) + a_2 * e(n-1) + b_1 * u(n-1)$

Emitir la acción de control:

$DAC(m(n) + m(n-1))$

Actualizar variables:

$e(n-1)=e(n)$

$m(n-1)=m(n)$

GOTO captura valor de la variable

Ejemplo 1

Transformar la función del controlador continuo a su equivalente discreto.

$$H(s) = \frac{a}{s+a}$$

Usando la transformación invariante al impulso con ZOH:

$$ZOH(s) \leftrightarrow (1-z^{-1}) \zeta \left\{ \frac{H(s)}{s} \right\}$$

Aplicando las reglas 1,2,3 y 5:

$$H(z) = k(1-z^{-1}) \zeta \left\{ \frac{a}{s(s+a)} \right\} = (1-z^{-1}) \zeta \left\{ \frac{1}{s} - \frac{1}{s+a} \right\}$$

$$H(z) = k(1-z^{-1}) \left[\frac{z^{-1}(1-e^{-aT})}{(1-z^{-1})(1-e^{-aT}z^{-1})} \right]$$

$$H(z) = k \frac{1-e^{-aT}}{z-e^{-aT}}$$

Agregando un cero en $z=-1$

$$H(z) = k \frac{(z+1)(1-e^{-aT})}{z-e^{-aT}}$$

Ajustando la ganancia:

$$D(s) \Big|_{s=0} = D(z) \Big|_{z=1}$$

$$\frac{a}{s+a} \Big|_{s=0} = k \frac{(z+1)(1-e^{-aT})}{z-e^{-aT}} \Big|_{z=1} = 1$$

$$2k = 1$$

$$k = \frac{1}{2}$$



ANÁLISIS DE SISTEMAS DISCRETOS.

Objetivo: Introducir los conceptos de estabilidad, alcanzabilidad y observabilidad en sistemas discretos.

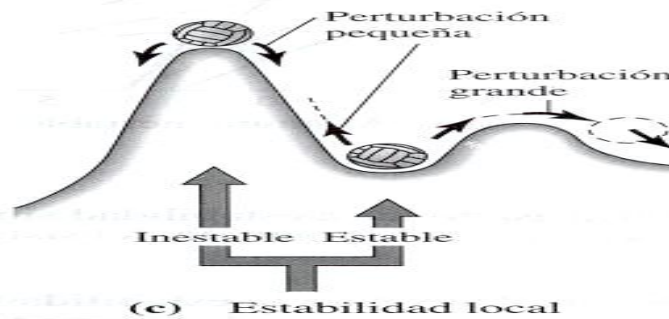
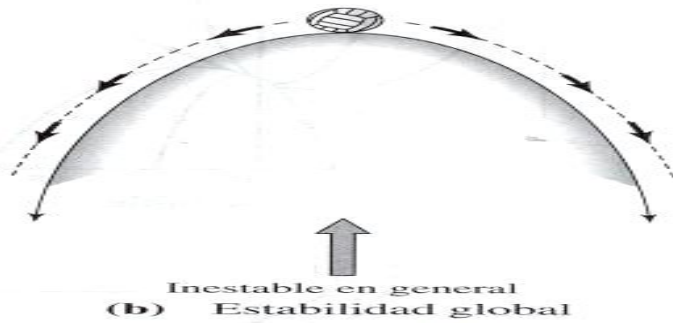
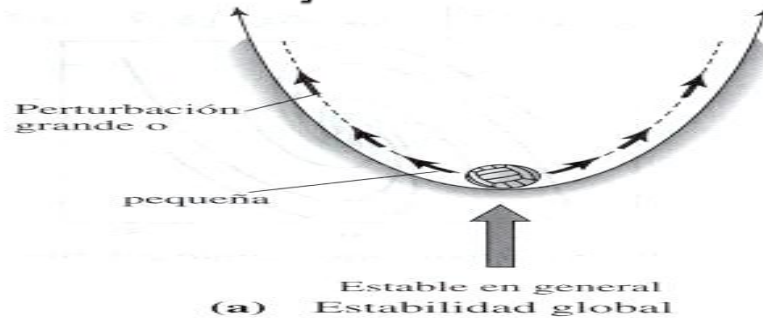


5.1. - Estabilidad

La estabilidad es una especificación básica que deben satisfacer los sistemas de control. Desde el punto de vista de la descripción externa se dice que un sistema lineal invariante en el tiempo es estable, si ante una entrada acotada se produce una salida acotada para todas las posibles condiciones iniciales («Bounded in put-bounded output» o «BIBO stability» en terminología inglesa). Existen diversos



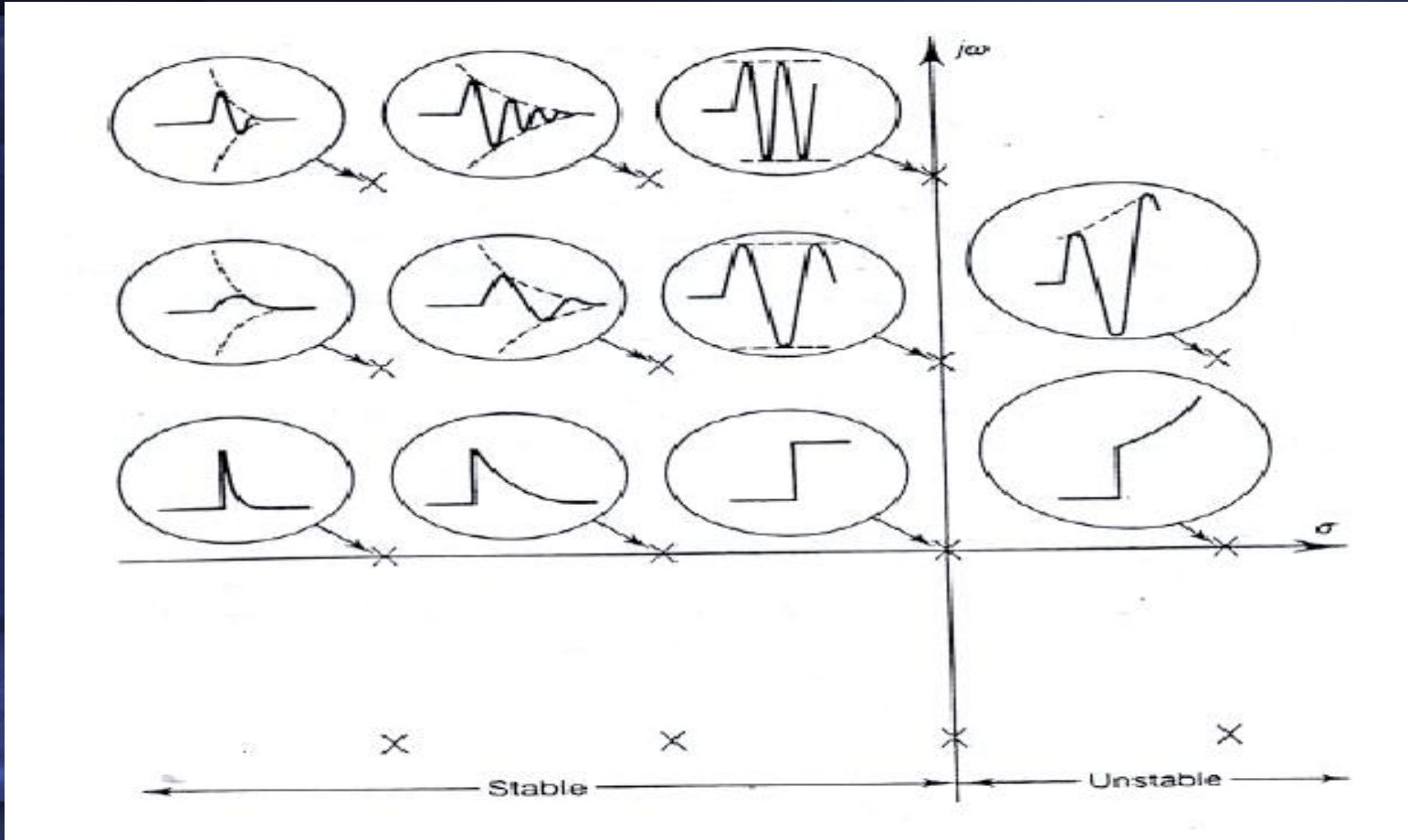
en el tiempo. De esta forma pueden aplicarse los conocidos criterios de Routh-Hurwitz, sobre la ecuación característica, y Nyquist en el dominio de la frecuencia; así como utilizar las graficas de Bode, con la respuesta en frecuencia, y del lugar de las raíces en el plano s . Nótese que el límite de estabilidad en el plano s es el eje imaginario, y que este eje se transforma, mediante $z = e^{sT}$ en una circunferencia, con centro en el origen y radio unidad, que es el límite de estabilidad en el plano z . Por consiguiente, el criterio de Routh-Hurwitz no puede ser aplicado directamente, sustituyéndose por los criterios de Schur-Cohn y de Jury.





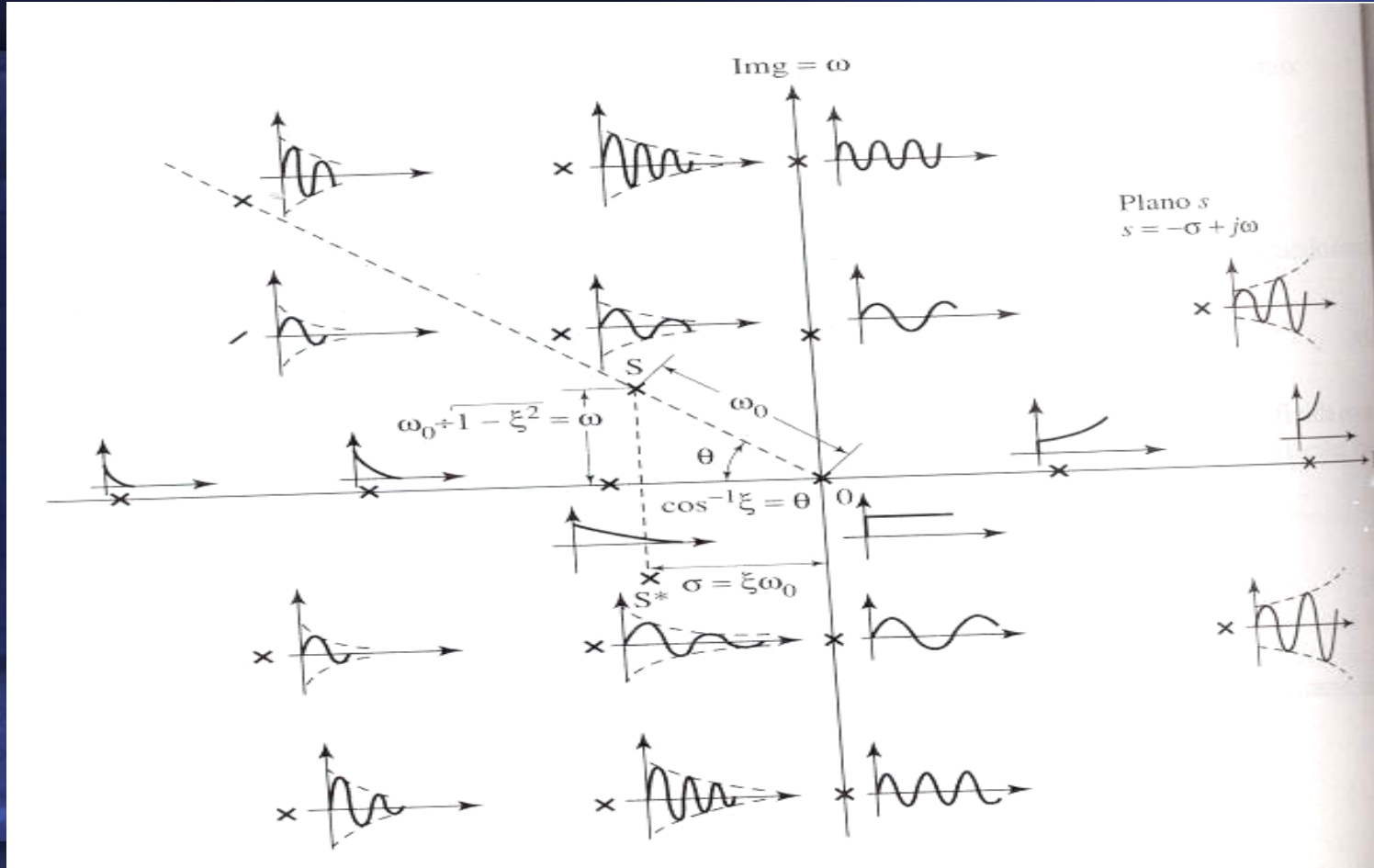
Definición de Estabilidad:

Se describe un sistema estable en el cual la solución complementaria decrece a cero conforme el tiempo se acerca al infinito. La presentación de la relación de Bode, se definió un sistema estable como aquel cuyos polos se encuentran en el semiplano izquierdo del plano s .

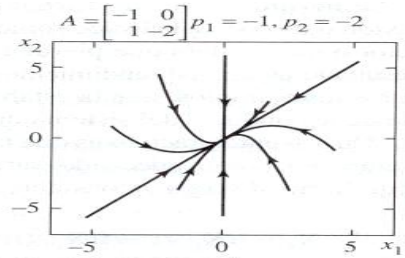




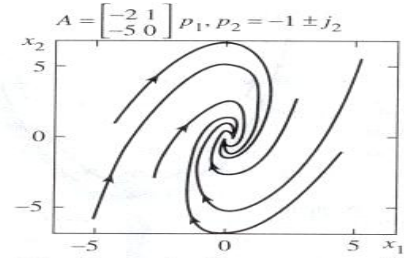
1. Un sistema lineal e invariante con el tiempo es estable si su salida está acotada para cada entrada acotada.
2. Un sistema lineal e invariante con el tiempo es estable si su función ponderada es absolutamente integrable en un intervalo de tiempo infinito, es decir, si $\int_0^{\infty} |w(t)| dt < \infty$, es decir es finita.
3. Un sistema lineal e invariante con el tiempo es estable si todos los polos de la función de transferencia en lazo cerrado $Y(s)/R(s)$ se encuentran en el semiplano izquierdo del plano s .



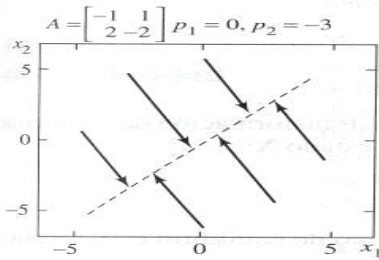
Conceptos de estabilidad
de estado



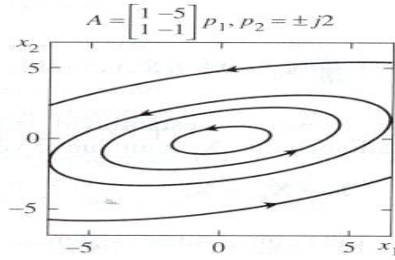
(a) Nodo asintóticamente estable



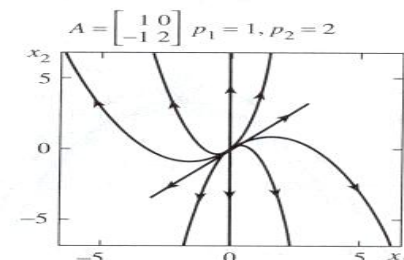
(b) Foco asintóticamente estable



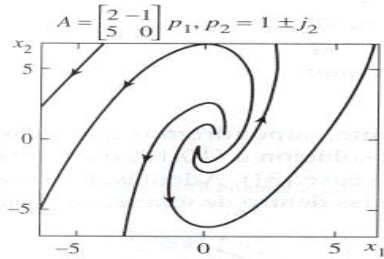
(c) Línea estable



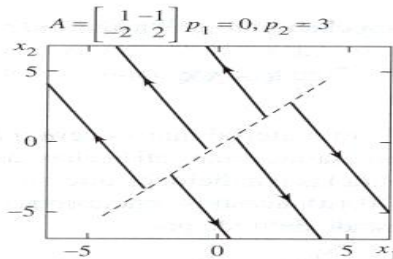
(d) Centro estable



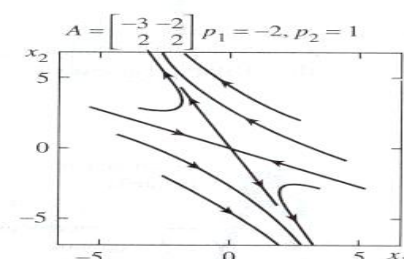
(e) Nodo inestable



(f) Foco inestable



(g) Línea inestable



(h) Montura inestable



1.2. Estabilidad

Dada una secuencia

$$x_{k+1} = f(x_k, k) \quad [1.4]$$

sean dos secuencias x_k y x_k^0 soluciones de [1.4]

Se dice que la secuencia x_k^0 es estable si dado

$$\|x_{k_0} - x_{k_0}^0\| < \delta \quad [1.5]$$

se cumple

$$\|x_k - x_k^0\| < \varepsilon \quad \forall k \geq k_0 \quad [1.6]$$

Se dice que la secuencia x_k^0 es *asintóticamente* estable si se cumple

$$\|x_k - x_k^0\| \rightarrow 0 \quad k \rightarrow \infty \quad [1.7]$$



1.2.1. Estabilidad de Sistemas Lineales

Sea el sistema

$$x^0_{k+1} = \Phi x^0_k \quad [1.8]$$

con

$$x^0_0 = a^0 \quad [1.9]$$

se cambia el valor inicial

$$x_0 = a \quad [1.10]$$

resultando

$$x_{k+1} = \Phi x_k \quad [1.11]$$

La diferencia entre ambas soluciones es

$$\bar{x}_{k+1} = x_{k+1} - x^0_{k+1} = \Phi \bar{x}_k \quad [1.12]$$

con

$$\bar{x}_0 = a - a^0 \quad [1.13]$$

esto implica que si x^0 es estable, toda otra solución será también estable.

Se deduce que la estabilidad es una característica del sistema y no de una solución determinada.



La solución de

$$\bar{x}_k = \Phi^k \bar{x}_0 \quad [1.14]$$

Si la matriz Φ se puede diagonalizar, la solución es una combinación lineal de los autovalores.

Si la matriz Φ no se puede diagonalizar, la solución es una combinación lineal del producto de polinomios por los autovalores.

Pero en ambos casos, para que la solución tienda a cero los autovalores deberán ser menor que 1.

Teorema 1. Un sistemas discreto, lineal, invariante en el tiempo es asintóticamente estable si todos los autovalores de Φ están dentro del círculo unidad.



5.2.- ANÁLISIS DE SISTEMAS DISCRETOS.

$$X_{(k+1)} = \Phi X_k + \Gamma U_k$$

Para lograr la estabilidad asintótica, todas las soluciones deben tender a cero cuando k tiende a infinito. Entonces los valores propios de Φ tienen la propiedad.

$$|\lambda_i| < 1$$

$$i = 1, 2, \dots, n$$



Teorema.- Estabilidad asintótica de sistemas lineales. Un sistema discreto, lineal e invariante es estable si y solo si todos los valores propios de Φ están, de manera estricta dentro del círculo unidad.

Cuando se emplea la palabra estable sin ningún calificativo posterior, significa normalmente estabilidad asintótica,

Teorema.- La estabilidad asintótica implica la estabilidad y la estabilidad entrada-acotada-salida-acotada (BIBO).

Estabilidad BIBO. Un sistema lineal invariante se dice que es estable entrada-acotada-salida-acotado (BIBO, del inglés bounded-input-bounded-output) si una entrada acotada produce una salida acotada para cualquier valor inicial.



1.2.2. Estabilidad de Sistemas con Entrada y Salida Acotadas(BIBO)

Un sistema cuya entrada es acotada, es estable si su salida también lo es.

La estabilidad asintótica es más restrictiva.

Ejemplo 1.1. *Oscilador Armónico*

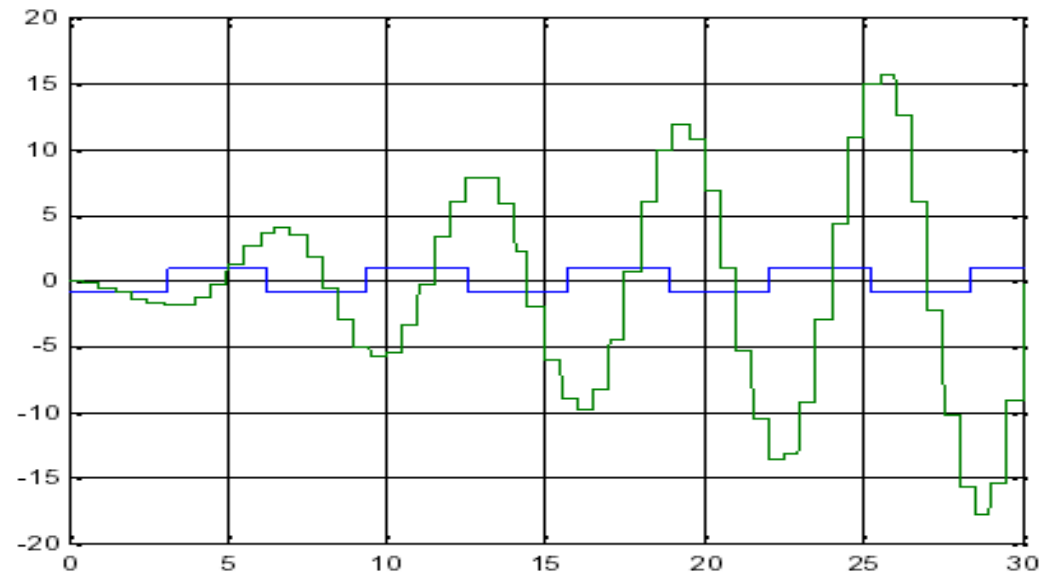
$$x_{k+1} = \begin{bmatrix} \cos(\omega T) & \text{sen}(\omega T) \\ -\text{sen}(\omega T) & \cos(\omega T) \end{bmatrix} x_k + \begin{bmatrix} 1 - \cos(\omega T) \\ \text{sen}(\omega T) \end{bmatrix} u_k$$

$$y_k = [1 \quad 0] x_k$$

Los autovalores son 1 .

Si la entrada es nula, el sistema es estable porque $\|x_k\| = \|x_0\|$

Pero si la entrada es una onda cuadrada de frecuencia ω la salida es



El sistema es estable pero no es estable en el sentido de entrada y salida acotada



PRUEBAS DE ESTABILIDAD.

Estas son algunas de las formas de determinar la estabilidad de un sistema discreto:

- Cálculo directo de los valores propios de Φ .
- Métodos basados en las propiedades de los polinomios característicos.
- Método del lugar de las raíces.
- Criterio de Nyquist.
- Método de Lyapunov.



CUV y TT - BUAP



En algunos casos es más fácil calcular el polinomio característico

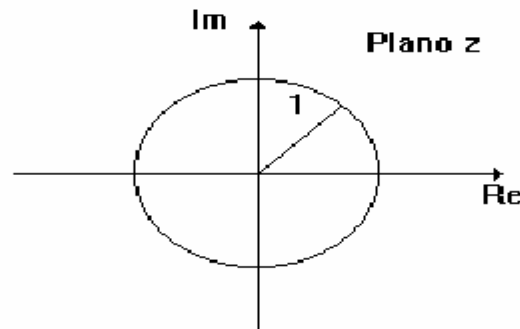
$$H(z) = \frac{B(z)}{A(z)}$$

$$A(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n$$

e investigar la ecuación característica

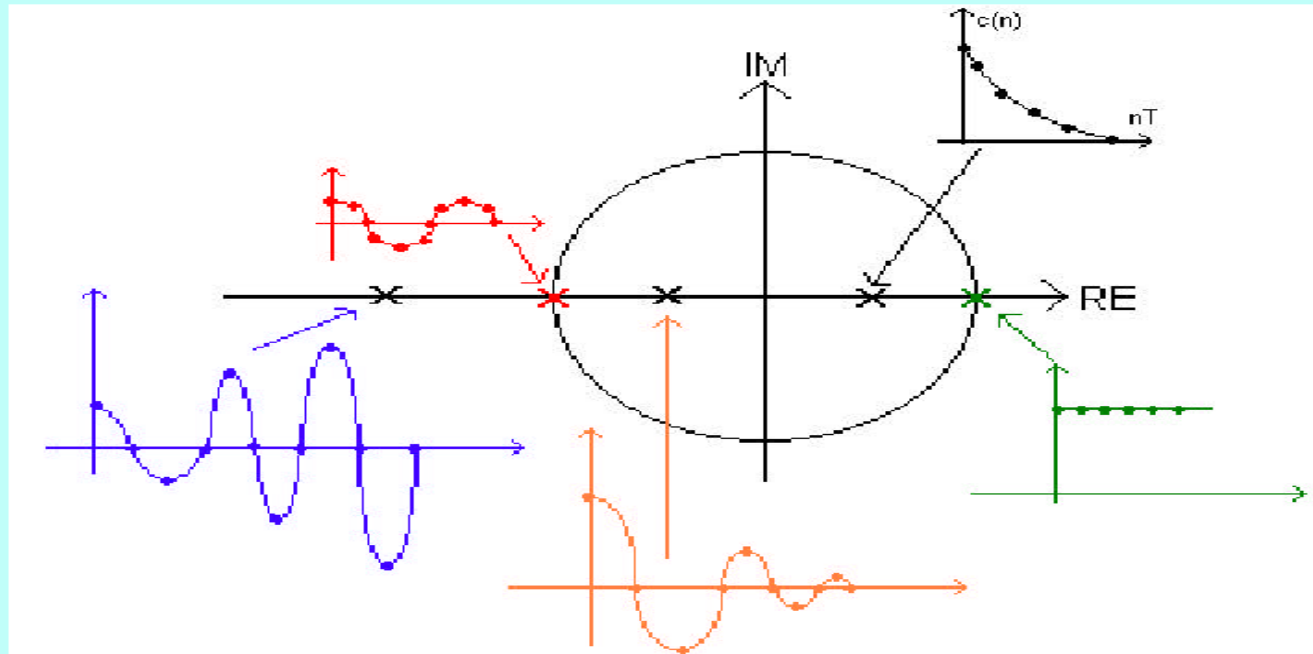
$$A(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n = 0$$

El polinomio característico es el polinomio denominador de la función de transferencia discreta. Las pruebas de estabilidad pueden obtenerse investigando las condiciones para que los polos del polinomio estén dentro del círculo unidad (<1 de las raíces).

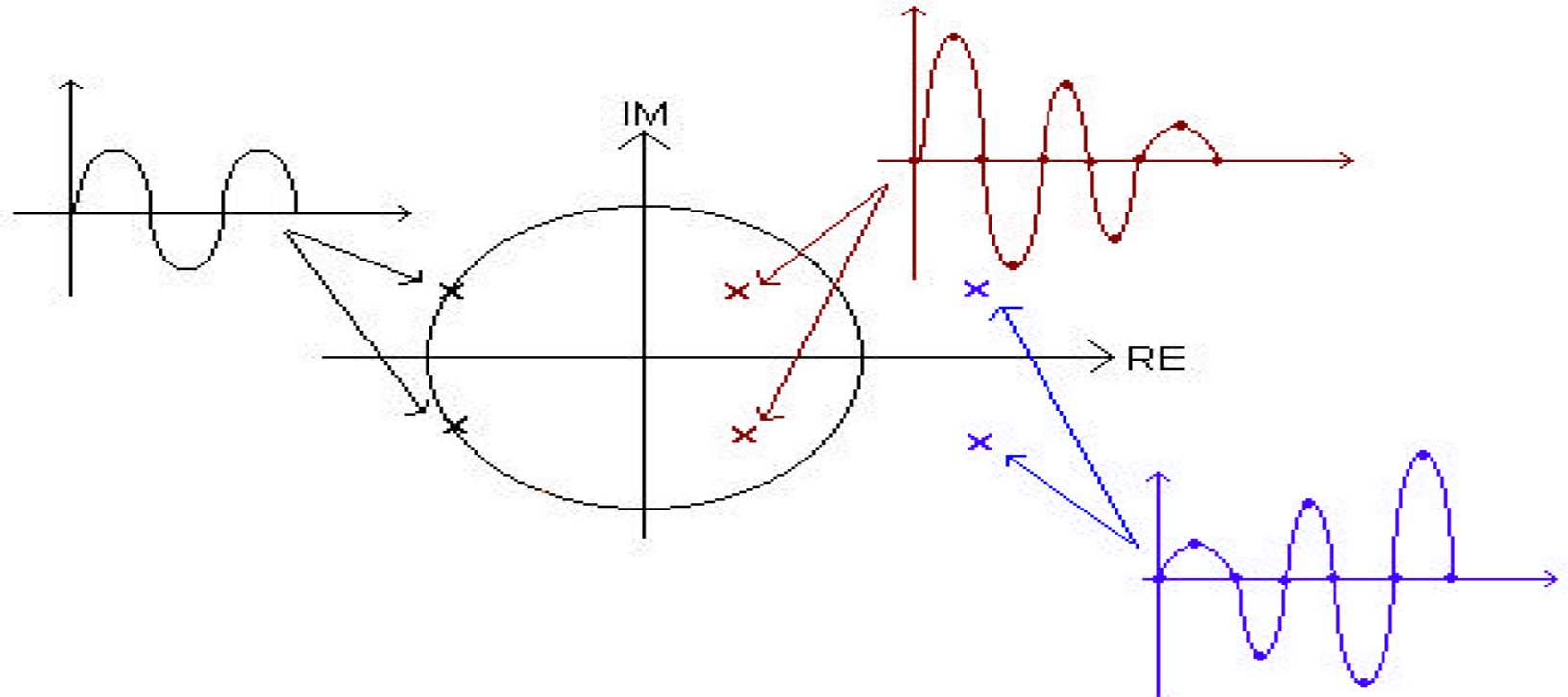


Estabilidad en el plano z

Polos reales:



Polos complejos





Conclusiones para estabilidad:

1. Si los polos tienen valor absoluto menor a la unidad el sistema es estable.
2. Si los polos están fuera del círculo unitario o existen polos repetidos sobre el círculo unitario el sistema es inestable.
3. Si se cumple la condición 1 y existe un polo sobre el círculo unitario el sistema es marginalmente estable.



Ejemplo:

$$z^2 + 1.5z + 0.9 = 0$$

Raíz	real	imaginaria
1	0.75	0.581
2	0.75	-0.581

Por lo tanto todas las raíces de esta ecuación están dentro del círculo unitario esto implica que el sistema es estable.



El criterio de ROUTH-HURWITZ es uno de los métodos existentes para determinar si un polinomio tiene todos sus ceros en el semiplano izquierdo. La transformación Möbius,

$$w = \frac{z+1}{z-1}$$

aplica el círculo unidad del plano z en el semiplano izquierdo del plano w .

Sin embargo, es útil disponer de condiciones que digan directamente si un polinomio tiene todos sus ceros dentro el círculo unidad. Schur, Cohn y Jury desarrollaron un criterio de este tipo, equivalente al criterio de ROUTH-HURWITZ.



CUVyTT-BUAP



CRITERIO DE JURY.

Para determinar si un polinomio como

$$A(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n$$

tienen todos sus ceros dentro del círculo unidad se puede emplear la siguiente prueba

a_0	a_1	a_{n-1}	a_n	
a_n	a_{n-1}	a_1	a_0	Orden n
b_0	b_1	b_{n-1}		
b_{n-1}	b_{n-2}	b_0		Orden n-1
c_0	c_1	...	c_{n-2}			
c_{n-2}	c_{n-3}	...	c_0			Orden n-2
						.
						.
						.
d_0						Orden 0



Numero de renglones $2n+1$.

$$\alpha_n = \frac{a_n}{a_0}$$

$$b_0 = a_0 - a_n \frac{a_n}{a_0}$$

$$b_1 = a_1 - a_{n-1} \frac{a_n}{a_0}$$

$$b_k = a_k - a_{n-k} \frac{a_n}{a_0}$$

$$c_0 = b_0 - b_{n-1} \frac{b_{n-1}}{b_0}$$

$$c_1 = b_1 - b_{n-2} \frac{b_{n-1}}{b_0}$$

$$c_k = b_k - b_{n-(k+1)} \frac{b_{n-1}}{b_0}$$

$$d_k = c_k - c_{n-(k+2)} \frac{c_{n-2}}{c_0}$$

con $a_0, b_0, c_0, \dots, d_0 > 0$ el sistema es estable.



EJEMPLO:

Sea la ecuación característica ;

$A(z) = z^2 + a_1z + a_2 = 0$ el esquema de Jury es:

1	a_1	a_2
a_2	a_1	1

b_0	b_1
$1 - a_2^2$	$a_1 - a_1a_2$
$a_1(1 - a_2)$	$1 - a_2^2$

c_0
$1 - a_2^2 - a_1^2(1 - a_2)/(1 + a_2)$

todas las raíces de la ecuación están dentro del círculo unidad si .

$$1 - a_2^2 > 0$$

$$1 - a_2^2 - a_1^2(1 - a_2)/(1 + a_2) > 0$$

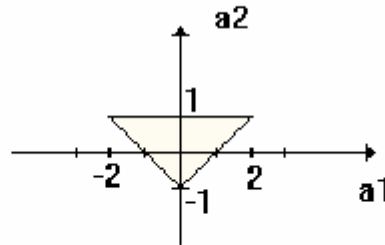
Esto da las condiciones

$$a_2 < 1$$

$$a_2 > -1 + a_1$$

$$a_2 > -1 - a_1$$

en la figura se da el área de estabilidad para la ecuación de segundo orden



$$a_0^0 = (1 - (a_2)^2) \left(1 - \frac{a_1}{1 + a_2} \right) > 0$$

esto implica

$$a_2 < 1$$

$$\left(1 - \frac{a_1}{1 + a_2} \right) = \left(\frac{1 + a_2 - a_1}{1 + a_2} \right) > 0$$

$$a_1 < 1 + a_2$$

$$a_1 > -1 - a_2$$



SEGUNDO METODO DE LYAPUNOV.

El *segundo método de Lyapunov* es una herramienta muy útil para determinar la estabilidad de los sistemas dinámicos no lineales. Lyapunov desarrolló la teoría para ecuaciones diferenciales, pero se puede obtener también una teoría equivalente para ecuaciones en diferencias.

Definición.- Función de Lyapunov. $V(x)$ es una función de Lyapunov para el sistema

$$x(k+1) = f(x(k)); \quad f(0) = 0$$

Si

1. $V(x)$ es continua en x y $V(0) = 0$.
2. $V(x)$ es definida positiva.
3. $\Delta V(x) = V(f(x)) - V(x)$ es definida negativa.



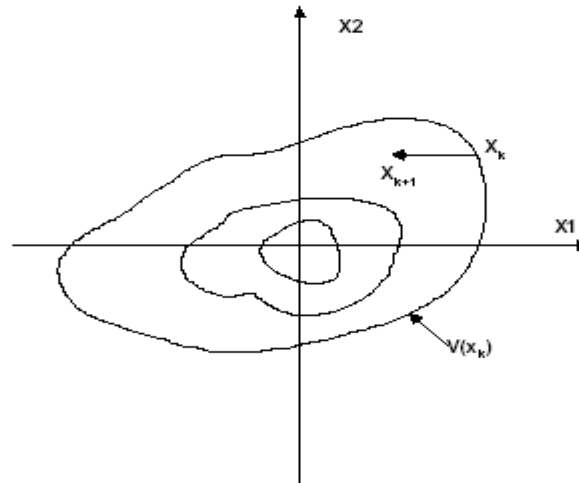
Una ilustración geométrica sencilla de la definición puede verse en la Figura. Las curvas de nivel de una función continua definida positiva V son curvas cerradas en un entorno del origen. Identifiquemos cada curva por el valor de la función. La condición 3 implica que la dinámica del sistema es tal que la solución siempre se mueve hacia las curvas con los valores más bajos. Todas las curvas de nivel rodean al origen y no se cortan entre sí. Entonces parece razonable que la existencia de una función de Lyapunov asegure la estabilidad asintótica. El siguiente teorema formula de forma precisa esta afirmación.

- Segundo Método de Lyapunov

Función de Lyapunov:

Sea el sistema

$$x_{k+1} = f(x_k) \quad f(0) = 0$$

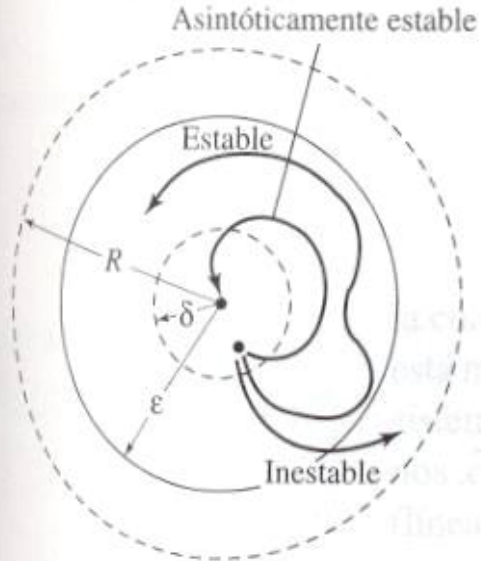


Las curvas de nivel de $V(x)$ son cerradas alrededor del origen

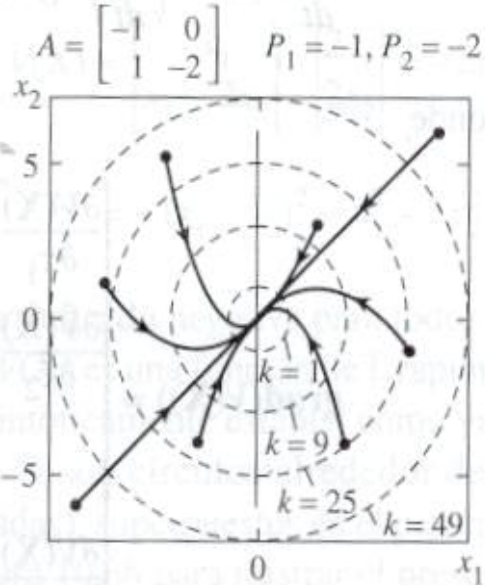
La tercera condición dice que la dinámica del sistema es tal que partiendo de un estado, el siguiente llevará a un valor de $V(x)$ menor o más cerca del origen.

$V(x)$ es una función de Lyapunov si

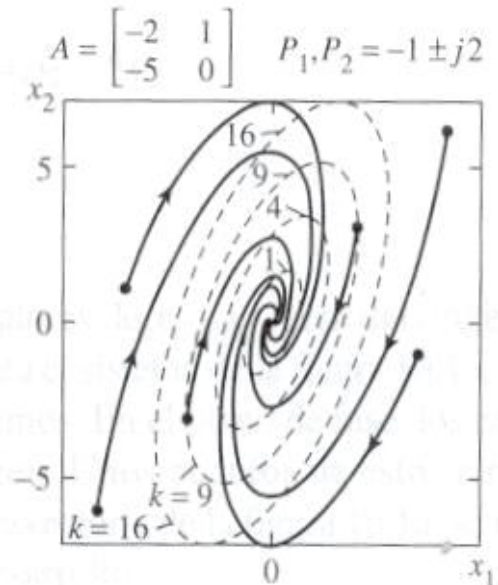
- $V(x)$ es continua en x y $V(0) = 0$
- $V(x)$ es definida positiva y
- $\Delta V(x) = V(f(x)) - V(x)$ es definida negativa



(a) Estabilidad en el sentido de Lyapunov



(b) $V(x) = x_1^2 + x_2^2$; (Círculos)



(c) $V(x) = 1.5x_1^2 - x_1x_2 + 0.5x_2^2$; (Elipse)



Teorema. Teorema de estabilidad de Lyapunov. La solución $x(k) = 0$ es asintóticamente estable si existe una función de Lyapunov para el sistema

$$x(k+1) = f(x(k))$$

Además, si

$$0 < \varphi(\|x\|) < V(x)$$

Donde $\varphi(\|x\|) \rightarrow \infty$ cuando $\|x\| \rightarrow \infty$, entonces la solución es asintóticamente estable para todas las condiciones iniciales.

El principal obstáculo para el empleo de la teoría de Lyapunov es encontrar una función de Lyapunov apropiada. En general, se trata de un problema difícil, sin embargo, para el sistema lineal, la determinación de funciones cuadráticas de Lyapunov es inmediato. Tomemos $V(x) = x^T P x$ como una posible función de Lyapunov. El incremento de V viene dado entonces por

$$\Delta V(x) = V(\Phi x) - V(x) = x^T \Phi^T P \Phi x - x^T P x$$

$$= x^T [\Phi^T P \Phi - P] x = -x^T Q x$$



Para que V sea una función de Lyapunov, es necesario y suficiente que exista una matriz definida positiva P que verifique la ecuación

$$\Phi^T P \Phi - P = -Q$$

Donde Q es definida positiva. La ecuación anterior se conoce como *Ecuación de Lyapunov*. Se demuestra que siempre hay una solución de la ecuación de Lyapunov cuando el sistema lineal es estable. La matriz P es definida positiva si Q lo es.



Teorema 4. Estabilidad

La solución $x_k = 0$ es asintóticamente estable si existe una función de Lyapunov para el sistema

Si además existe

$$0 < \varphi(\|x\|) < V(x) \quad [1.32]$$

y se cumple

$$\varphi(\|x\|) \rightarrow \infty \text{ cuando } \|x\| \rightarrow \infty \quad [1.33]$$

entonces, la solución es asintóticamente estable para cualquier condición inicial.

- El principal problema es encontrar la función de Lyapunov
- Para sistemas lineales, una función candidata es



$$V(x) = x^T P x$$

donde

$$\begin{aligned}\Delta V(x) &= V(\Phi x) - V(x) = x^T \Phi^T P \Phi x - x^T P x \\ &= x^T [\Phi^T P \Phi - P] x = -x^T Q x\end{aligned}$$

Para que $V(x)$ se una función de Lyapunov, debe existir una matriz P , que cumpla

$$\Phi^T P \Phi - P = -Q$$

con Q definida positiva

Esta es la *ecuación de Lyapunov*

La una matriz P es definida positiva si Q definida positiva



1.3.1. Controlabilidad y Alcanzabilidad

Sea el sistema

$$\begin{aligned}x_n &= \Phi^n x_0 + \Phi^{n-1} \Gamma u_0 + \dots + \Gamma u_{n-1} \\ &= \Phi^n x_0 + W_c U\end{aligned}\quad [1.37]$$

con

$$W_c = [\Gamma \quad \Phi\Gamma \quad \dots \quad \Phi^{n-1}\Gamma] \quad [1.38]$$

$$U = [u_{n-1}^T \quad \dots \quad u_0^T]^T \quad [1.39]$$

si W_c tiene rango n se pueden encontrar n valores de u para llevar al sistema a un valor deseado x_n .

Si hay más de una entrada la solución no es única.

- **Definición de Controlabilidad:**

Un sistema es controlable si se puede llevar desde cualquier punto al origen en tiempo finito.

- **Definición de Alcanzabilidad:**

Un sistema es alcanzable si se puede llevar desde cualquier punto a otro cualquiera en tiempo finito



1.3.1. Controlabilidad y Alcanzabilidad

Sea el sistema

$$\begin{aligned}x_n &= \Phi^n x_0 + \Phi^{n-1} \Gamma u_0 + \dots + \Gamma u_{n-1} \\ &= \Phi^n x_0 + W_c U\end{aligned}\quad [1.37]$$

con

$$W_c = [\Gamma \quad \Phi\Gamma \quad \dots \quad \Phi^{n-1}\Gamma] \quad [1.38]$$

$$U = [u^T_{n-1} \quad \dots \quad u^T_0]^T \quad [1.39]$$

si W_c tiene rango n se pueden encontrar n valores de u para llevar al sistema a un valor deseado x_n .

Si hay más de una entrada la solución no es única.

Un sistema es alcanzable si y solo si W_c es de rango n .

W_c se llama matriz de controlabilidad.

Ejemplo

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

no es alcanzable porque

$$W_c = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

si tuviera dos entrada con una matriz Γ no singular, el sistema sería alcanzable.



1.4. Observabilidad

Definición: El estado $x^0 \neq 0$ es *no observable* si existe un número finito $k_1 \geq n - 1$ en donde $y_k = 0 \forall 0 \leq k \leq k_1$ resultando $x_0 = x^0$ y $u_k = 0 \forall 0 \leq k \leq k_1$.

El sistema es observable si, conociendo k entradas y k salidas es suficiente para conocer el estado inicial.

En un sistema tal como el [1.1] es calculable el efecto de la entrada y no se pierde generalidad si se hace $u_k = 0$.

Se suponen conocidas las salidas y_0, y_1, \dots, y_{n-1}

Con esto se puede plantear



CUVYT-T-BUAP



$$y_0 = Cx_0$$

$$y_1 = Cx_1 = C\Phi x_0$$

$$\vdots$$

$$y_{n-1} = Cx_{n-1} = C\Phi^{n-1}x_0$$

vectorialmente

$$\begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix} x_0 = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

el estado inicial se puede reconstruir si la *matriz de observabilidad*

$$W_o = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix}$$

tiene rango n .



El sistema 5.1 es observable sii la *matriz de observabilidad* tiene rango n .

Ejemplo

Sea el sistema

$$x_{k+1} = \begin{bmatrix} 1,1 & -0,3 \\ 1 & 0 \end{bmatrix} x_k$$
$$y_k = [1 \quad 0,5] x_k$$

la matriz de observabilidad es

$$W_o = \begin{bmatrix} C \\ C\Phi \end{bmatrix} = \begin{bmatrix} 1 & -0,5 \\ 0,6 & -0,3 \end{bmatrix}$$

tiene rango l .

Implementación práctica del controlador PID discreto.

a) Acción proporcional discreta.

Continuo.

$$u(t) = K_p e(t) \Rightarrow D(s) = K_p$$

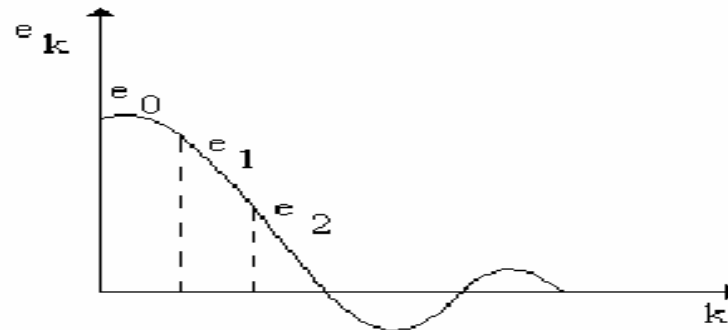
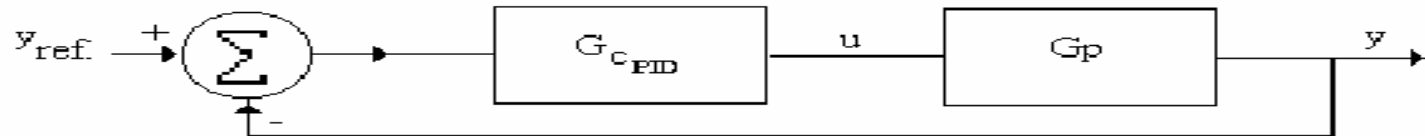
la forma discreta es:

$$u(k) = K_p e(k) \Rightarrow D(z) = K_p$$

donde $e(t)$ es el error

$$e(t) = y_{\text{ref.}} - y \Rightarrow$$

$$e_k = y_{\text{ref.}} - y(k-1)$$





b) La acción derivativa discreta.

Continua.

$$u(t) = K_p T_d \frac{de(t)}{dt} \quad \Rightarrow \quad D(s) = K_p T_d s$$

Discreto.

$$u(k) = K_p T_d \frac{e(k) - e(k-1)}{h} \quad \Rightarrow \quad D(z) = K_p T_d \frac{1 - z^{-1}}{h} = K_p T_d \frac{z-1}{h z}$$

otra forma:

$$\frac{u}{E}(z) = K_p T_d \frac{z-1}{h z} \quad \Rightarrow \quad u(k) = \frac{K_p T_d}{h} (e(k) - e(k-1))$$

$$u_d(k) = \frac{K_p T_d}{h} (e(k) - e(k-1))$$



c).- La acción integral discreta. Continuo

$$U(t) = \frac{K_p}{T_i} \int_{t_0}^t e(t) dt \Rightarrow D(s) = \frac{K_p}{T_i s}$$

Discreto

$$U(k) = U(k-1) + \frac{K_p h}{T_i} e(k) \Rightarrow D(z) = \frac{K_p h}{T_i (1 - z^{-1})} = \frac{K_p h z}{T_i (z - 1)}$$

otra forma

$$\frac{U}{E}(z) = \frac{K_p h (z + 1)}{2 T_i (z - 1)} = \frac{K_p h (1 + z^{-1})}{2 T_i (1 - z^{-1})}$$

multiplicando en ambos lados por $(1 - z^{-1})$ y dejando z^{-1}

$$U(k) - U(k-1) = \frac{K_p h}{2 T_i} (e(k) + e(k-1))$$

\Rightarrow

$$U_1(k) = U(k-1) + \frac{K_p h}{2 T_i} (e(k) + e(k-1))$$

$$U_1(k) = U(k-1) + \frac{K_p h}{T_i} e(k)$$



$$PID(k) = u_p(k) + u_i(k) + u_d(k) = P(k) + I(k) + D(k)$$

$$PID(k) = K_p e(k) + \frac{K_p T_d}{h} (e(k) - e(k-1)) + U_i(k-1) + \frac{K_p h}{T_i} e(k)$$

$$PID(k) = K_p e(k) + \frac{K_p}{h} (e(k) - e(k-1)) + U_i(k-1) + K_p h e(k)$$

$$U(k) = PID(k)$$

CONTROLADORES DIGITALES PID



La transformada de Laplace de un controlador PID continuo es :

$$G_c(s) = k \left[1 + \frac{1}{s T_I} + s T_d \right]$$

aplicando la transformacion bilineal (metodo de Tustin) donde

$$s = \frac{2}{h} \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$G_c(z) = k \left[1 + \frac{1}{\frac{2}{h} \frac{1 - z^{-1}}{1 + z^{-1}} T_I} + \frac{2}{h} \frac{1 - z^{-1}}{1 + z^{-1}} T_d \right]$$

$$G_c(z) = k \left[1 + \frac{h(1 + z^{-1})}{2(1 - z^{-1}) T_I} + \frac{2}{h} \frac{1 - z^{-1}}{1 + z^{-1}} T_d \right]$$



CUV_yTT-BUAP



$$G_c(z) = k \left[1 + \frac{h(1+z^{-1})}{2(1-z^{-1})T_1} + \frac{z-1}{hz} T_d \right]$$

Multiplicando por z:

$$G_c(z) = k \left[1 + \frac{h(z+1)}{2(z-1)T_1} + \frac{z-1}{hz} T_d \right]$$

Reduciendolo de la forma:

$$G_c(z) = k \frac{(b_0 + b_1 z + b_2 z^2)}{z(z-1)}$$

donde

$$b_0 = \frac{T_d}{h}$$

$$b_1 = \left(\frac{h}{2T_1} - \frac{2T_d}{h} - 1 \right)$$

$$b_2 = \left(\frac{h}{2T_1} + \frac{T_d}{h} + 1 \right)$$



Pseudo código de un control digital.

Inicio

inicializar variables y constantes.

$a1, a2, b1$

$e(n)=0$

$e(n-1)=0$

$u(n)=0$

$u(n-1)=0$

Capturar el valor de la variable

$c(nT)=ADC$

calcular el error:

$e(n)=r(nT)-c(nT)$

evaluar la acción de control:

$m(nT)=a_1 * e(n) + a_2 * e(n-1) + b_1 * u(n-1)$

Emitir la acción de control:

$DAC(m(n) + m(n-1))$

Actualizar variables:

$e(n-1)=e(n)$

$m(n-1)=m(n)$

GOTO captura valor de la variable

Considere el circuito mostrado en la figura 3.25. Este circuito es la implementación pasabajos de segundo grado. A continuación derivamos la función de transferencia como una forma de incrementar nuestra habilidad para analizar circuitos con amplificador operacional.

Se observa que para un amplificador operacional ideal $V_2(s) = V_o(s)$, escribimos una ecuación de corriente en el nodo 1, para obtener

$$\frac{V_i(s) - V_1(s)}{R_1} = \frac{V_1(s) - V_o(s)}{R_2} + [V_1(s) - V_o(s)](C_1s), \quad [3.32]$$

$$V_i(s)G_1 = [G_1 + G_2 + C_1s]V_1 - [G_2 + C_1s]V_o(s), \quad [3.33]$$

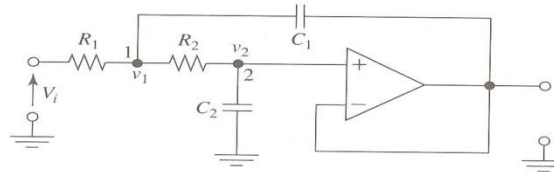


Figura 3.25 | Compensador pasabajos de segundo grado que utiliza un amplificador operacional.

donde por conveniencia se expresa R_1 y R_2 como admitancias. Una ecuación similar en el nodo 2 da

$$\frac{V_1(s) - V_o(s)}{R_2} = V_o(s)(sC_2), \quad [3.34]$$

donde se supuso que no fluye corriente hacia la terminal positiva del amplificador operacional. Por la ecuación [3.34] se ve que

$$V_1(s) = \left[\frac{1 + sC_2}{G_2} \right] V_o. \quad [3.35]$$

Sustituyendo la ecuación [3.35] en la ecuación [3.33] se obtiene

$$\begin{aligned} V_i(s)G_1 &= \left\{ [C_1s + (G_1 + G_2)] \left[\left(\frac{C_2}{G_2} \right) s + 1 \right] - [C_1s + G_2] \right\} V_o(s) \\ &= \left[\left(\frac{C_1C_2}{G_2} \right) s^2 + \left(\frac{C_2}{G_2} (G_1 + G_2) \right) s + G_1 \right] V_o(s). \end{aligned}$$



Entonces

$$\frac{V_o(s)}{V_i(s)} = \frac{(G_1 G_2 / C_1 C_2)}{s^2 + [(1/C_1)(G_1 + G_2)]s + (G_1 G_2)/(C_1 C_2)},$$

o, de manera equivalente

$$\frac{V_o(s)}{V_i(s)} = \frac{1/(R_1 C_1 R_2 C_2)}{s^2 + [(1/C_1)(1/R_1 + 1/R_2)]s + (1/R_1 C_1 R_2 C_2)}. \quad [3.36]$$

Aun cuando no es completamente obvio de acuerdo con la ecuación [3.36], se puede determinar por separado ambos coeficientes en el denominador de la función de transferencia. Suponga que deseamos que el denominador sea

$$(s + 1)(s + 10) = s^2 + 11s + 10.$$

Sea $R_2 = aR_1$ y $C_2 = bC_1$. Entonces

$$\frac{V_o(s)}{V_i(s)} = \frac{1/(ab)(R_1^2 C_1^2)}{s^2 + [(1/R_1 C_1)(1 + 1/a)]s + [1/(ab)R_1^2 C_1^2]}.$$

... de a y b. Sea



CILV y TT - RIAD

CALCULOS PARA LA FUNCION DE TRANSFERENCIA



Proponemos M_p y T_p de los siguientes valores:

$$\underline{M_p = .5}$$
$$\underline{T_p = 1s}$$

Entonces:

$$x = \sqrt{\frac{\ln(M_p)}{p^2 + \ln(M_p)}} = \sqrt{\frac{\ln(.5)}{p^2 + \ln(.5)}} = .21545$$

$$T_p = p/Wd \quad \Rightarrow \quad Wd = p = 3.1415$$

$$Wn = \frac{Wd}{\sqrt{1 - x^2}} = \frac{3.1415}{\sqrt{1 - .21545^2}} = 3.2172$$

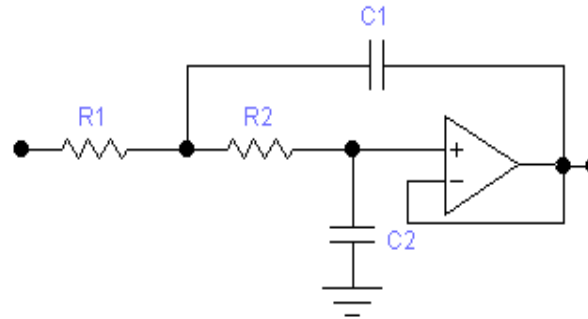
de la forma general de la función de transferencia

$$H(s) = \frac{Wn^2}{s^2 + 2xWns + Wn^2}$$

Entonces nos queda de la siguiente forma

$$H(s) = \frac{10.35}{s^2 + 1.386s + 10.35}$$

Ahora el circuito que elegimos que tiene una función de transferencia como la propuesta es el siguiente:



Donde:

$$H(s) = \frac{k\omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$$

Si hacemos que $K=1$ tenemos que:

$$\omega_0 = \omega_n = 3.2172$$

$$Q = \frac{1}{2\zeta} = \frac{1}{2(.21545)} = 2.32$$



Y sabemos que

$$W_o = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

Asi que tenemos:

$$W_o^2 = 10.35 = R_1 R_2 C_1 C_2$$

Si proponemos

$$R_1 = 100 \text{ kW}$$

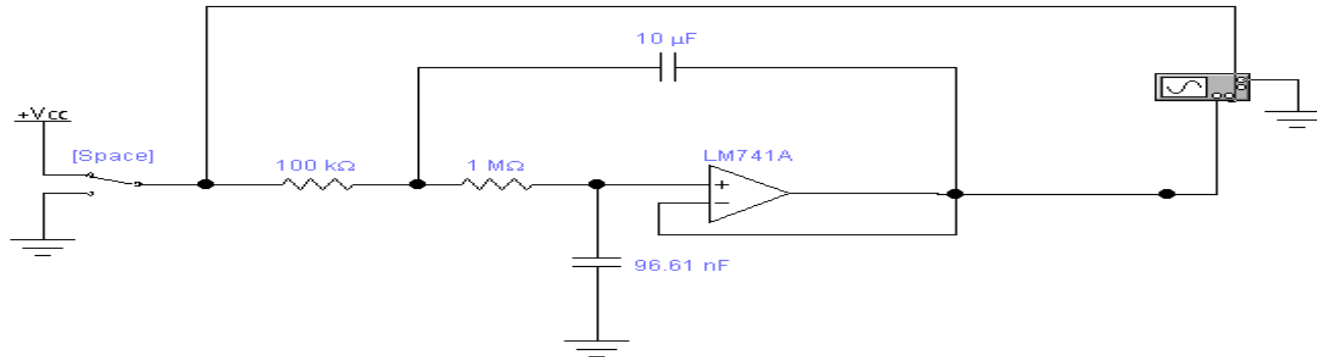
$$R_2 = 1 \text{ MW}$$

$$C_1 = 10 \text{ mF}$$

Entonces, despejando C_2 de la ecuación anterior tenemos

$$C_2 = \frac{10.35}{R_1 C_1 R_2} = \frac{10.35}{(100 \text{ K})(10 \text{ m})(1 \text{ M})} = 96.61 \text{ nF}$$

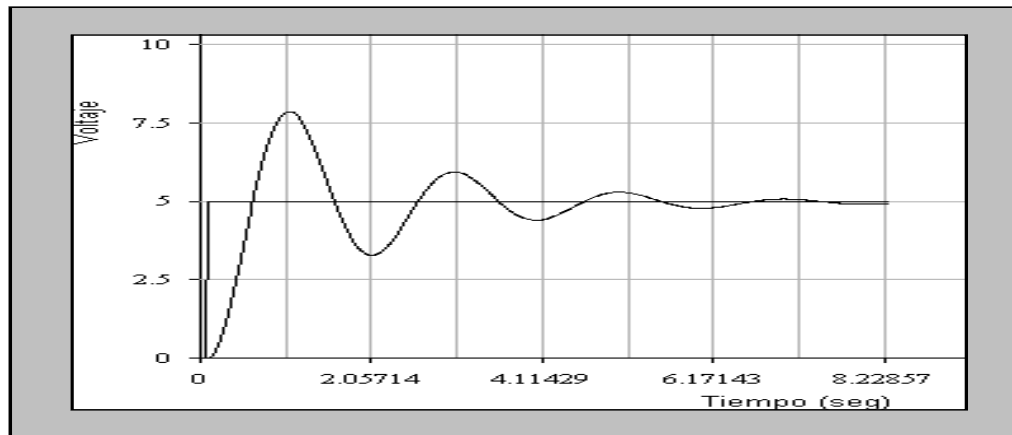
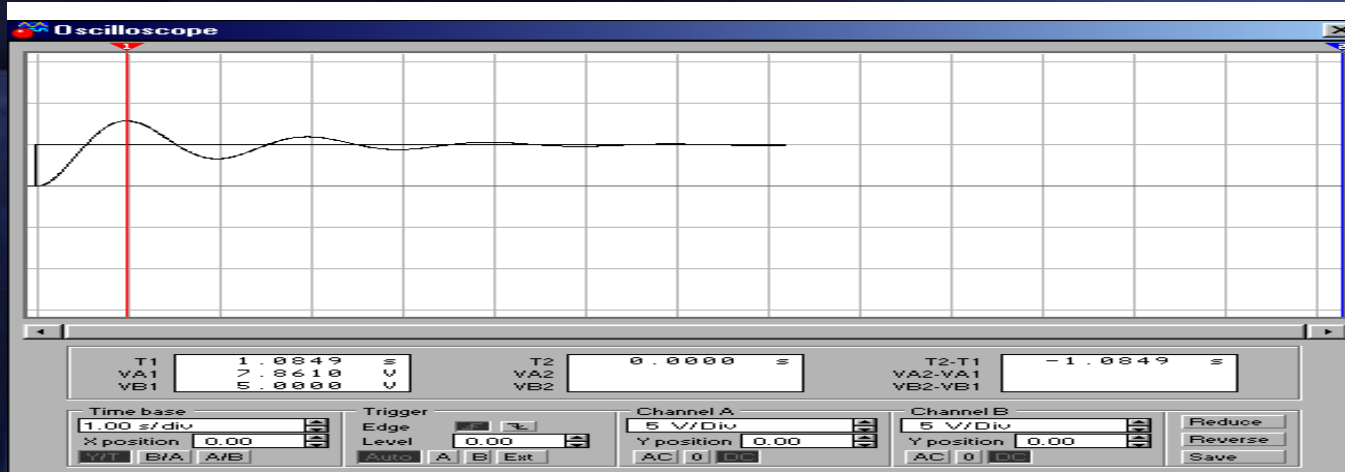
Entonces nuestro circuito queda como sigue:



haciendo la simulación en Electronics Workbench

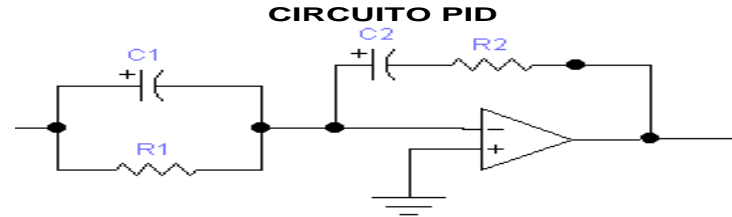


CUVyTT-BUAP



Como podemos observar se cumple que el tiempo de pico t_p es de 1 seg. y el sobreimpulso máximo es de aprox. .5 cumpliendo así con lo propuesto

Después conectando el PID como se muestra en la siguiente figura



sabemos que la función de transferencia tiene de manera general la siguiente forma.

$$K_p + \frac{K_i}{s} + K_d s$$

para encontrar la función de transferencia del circuito anterior tenemos que

$$\frac{E_o(s)}{E_i(s)} = -\frac{R_2 C_1}{R_1} s + \frac{R_2}{R_1} + \frac{C_1}{C_2} \frac{1}{s} + \frac{1}{(R_1 C_2) s^2}$$

de donde

$$K_p = \frac{R_2}{R_1} + \frac{C_1}{C_2}$$

$$K_i = \frac{1}{(R_1 C_2)}$$

$$K_d = (R_2 C_1)$$

propusimos los sig. valores

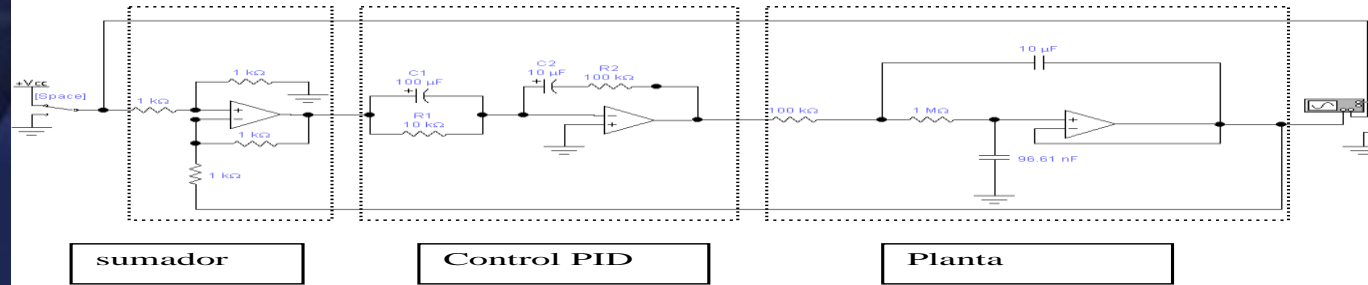
$$R_1 = 10k$$

$$R_2 = 100k$$

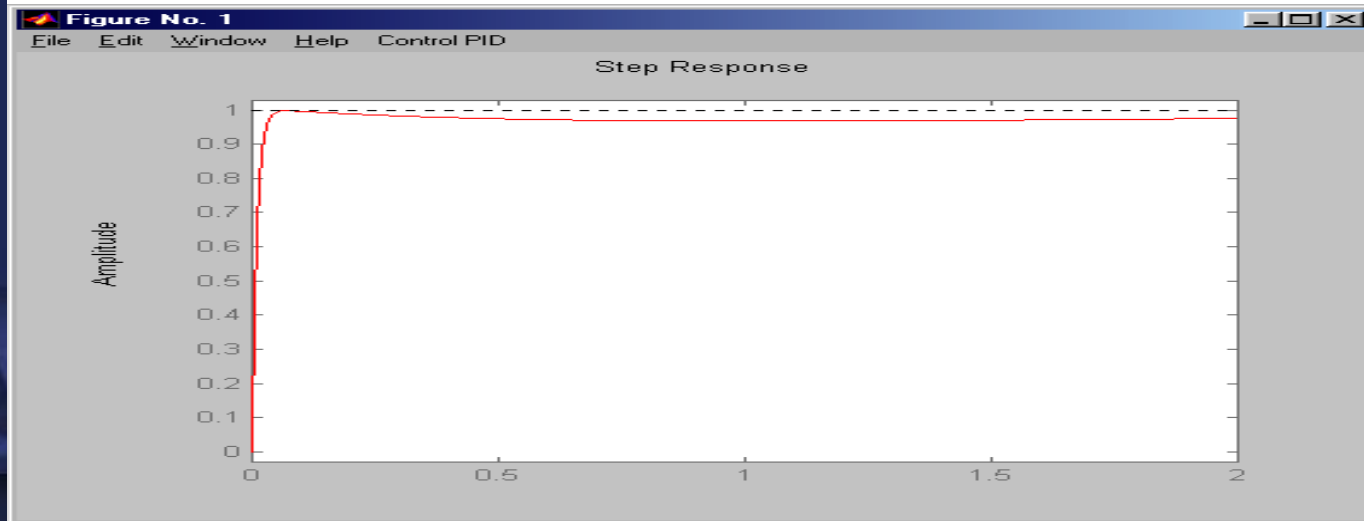
$$C_1 = 100m$$

$$C_2 = 10m$$

El circuito quedo como sigue



la grafica de la simulación en matlab fue la sig.





1.5.3. Realimentación con Relé

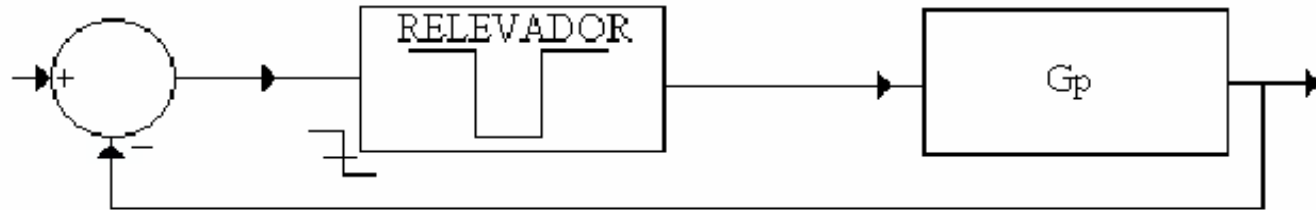
Desarrollando en serie de Fourier la salida de un relé, su primer armónico tiene una amplitud de:

$$A_r = \frac{4d}{\pi} \quad (1.5)$$

si la amplitud de la salida es a , la ganancia a esa frecuencia será

$$G\left(j\frac{2\pi}{t_c}\right) = -\frac{\pi a}{4d} \quad (1.5)$$

MÉTODO DEL RELEVADOR



$$G_p = \frac{y(k)}{u(k)} = -\frac{\Pi a}{4d} \Rightarrow K_c = \frac{4d}{\Pi a} \quad \text{Ganancia crítica}$$

donde :

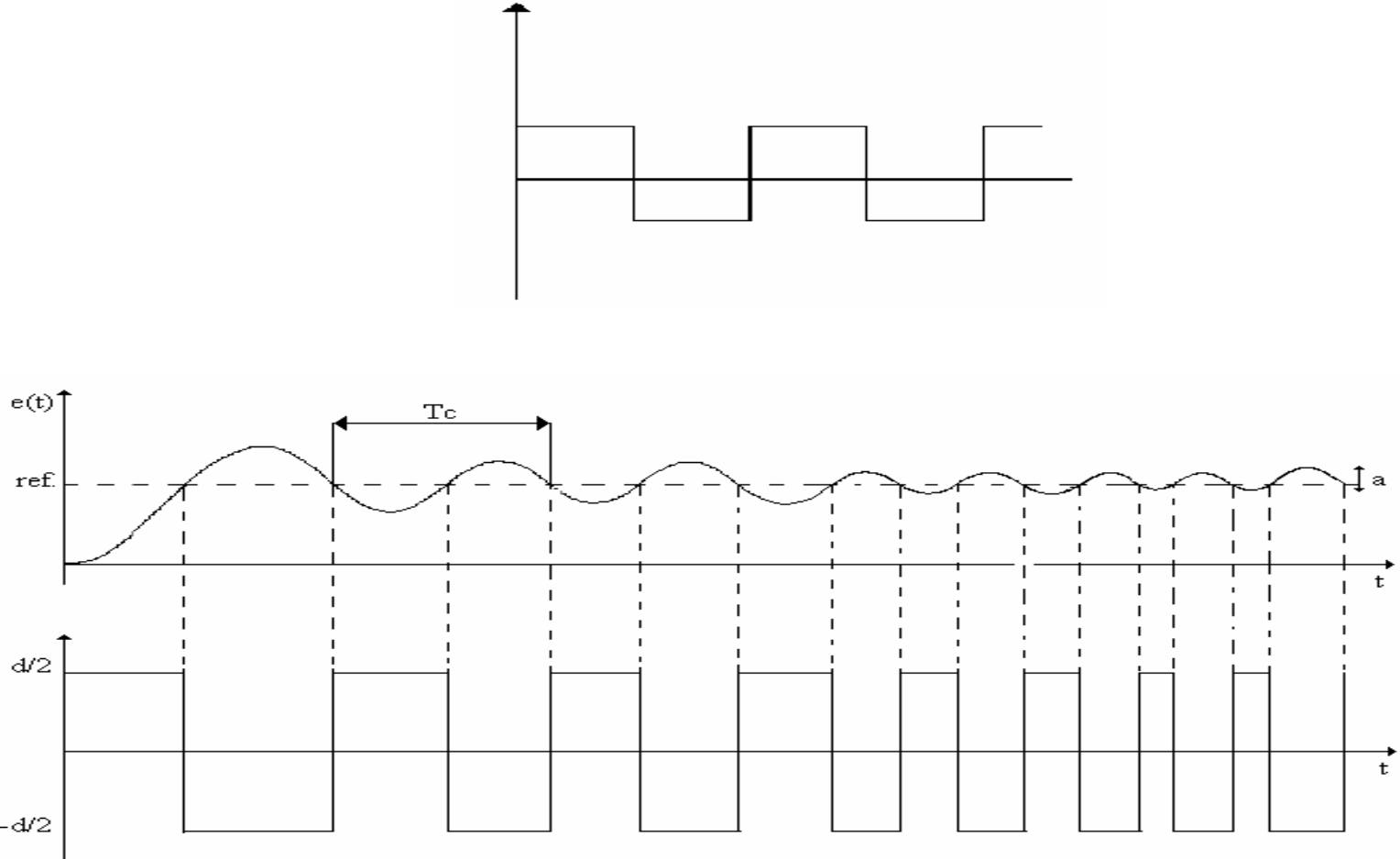
a = amplitud de la planta

d = amplitud del relevador ; $d = 0.1 \text{ ref.}$

Salida del relevador expandible en serie de fourier :

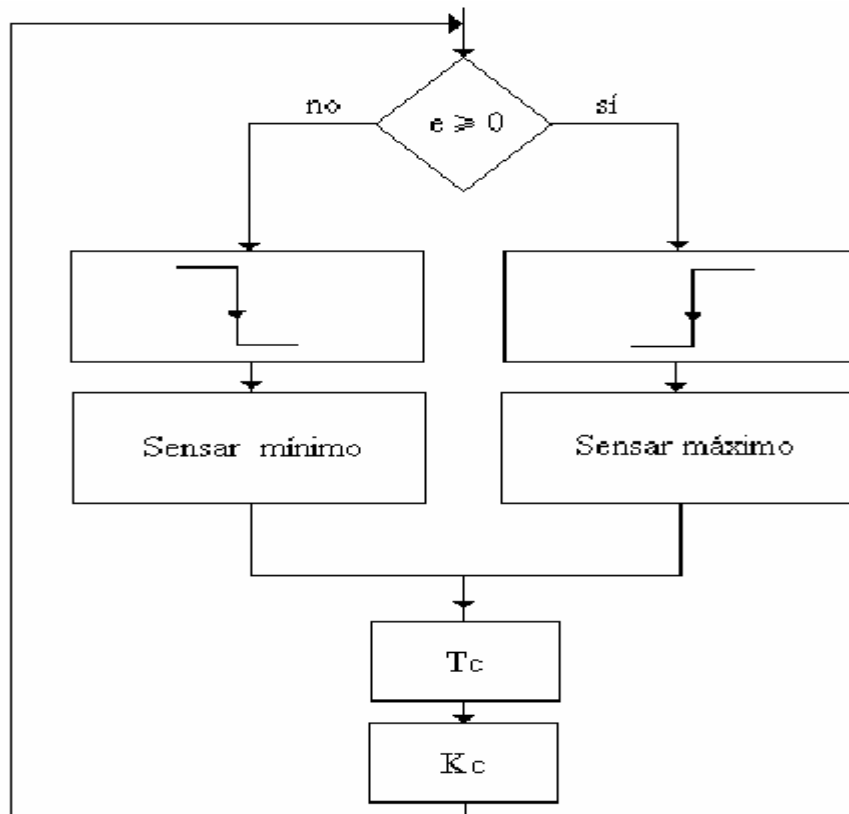


CUV_yTT-BUAP



8 oscilaciones repetidas para asegurar el $T_c =$ periodo crítico

$$T_c = \frac{2\pi}{\omega_c}$$



Parámetros recomendados por Ziegler - Nichols:

CONTROLADOR	K_p	T_i	T_d
P	$0.5 K_c$		
PI	$0.45 K_c$	$0.8 T_c$	
PID	$0.6 K_c$	$T_c/2$	$T_c/8$



1.5. Ajuste del PID

Existen muchos métodos para el cálculo de K , T_i y T_d ya sea en forma empírica o analítica. Los más clásicos son los métodos de Ziegler-Nichols (1942) tanto a partir de la respuesta al escalón como de la respuesta en frecuencia.

El método de Ziegler-Nichols basado en la respuesta en frecuencia o en cadena cerrada consiste en lo siguiente: Se realimenta la planta con un regulador proporcional aumentando su ganancia hasta que el sistema presente una oscilación no amortiguada.

Se mide la ganancia crítica K_c y el período t_c de las oscilaciones ajustando los parámetros del regulador según la tabla siguiente:

Controlador	K_p	T_i	T_d
P	$0,5K_c$		
PI	$0,45K_c$	$0,833t_c$	
PID	$0,6K_c$	$0,5t_c$	$0,125t_c$

Tabla 5-II Regulador PID en cadena cerrada



Ejercicios de tarea previos al examen parcial:

Del Libro de Karl J. Astrom
Computer- Controlled Systems

3.1

3.2

3.5

3.6

3.16

3.19

Alternativas en Automatización:

- ◆ **Computadoras**
- ◆ **PLCs**
- ◆ **Híbridos**

Ventajas de los Sistemas de Automatización Computarizados

- Enfasis en la Información
- Gran Flexibilidad
- Capacidades de Procesamiento y Almacenamiento Crecientes
- Sencillez de Operación
- Alta Conectividad
- Portabilidad
- Facilidad de Desarrollo Arquitectura Abierta
- Estandarización
- Servicio y Mantenimiento abiertos
- Bajo costo
- Actualizaciones y mejoras económicas

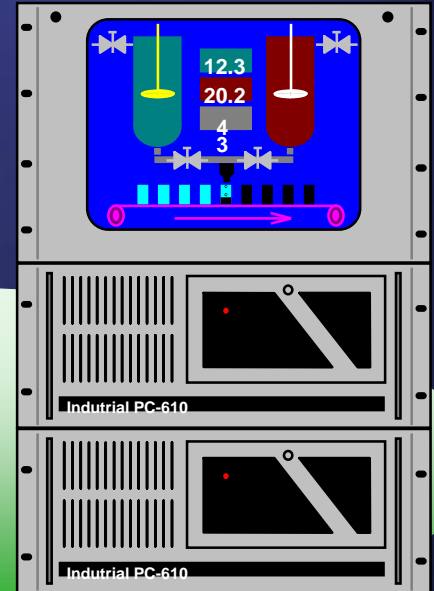
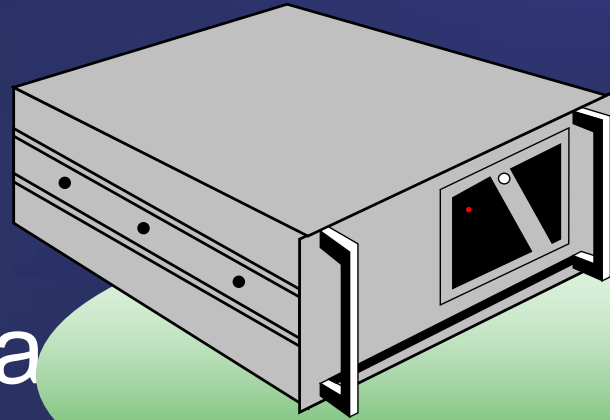
Componentes de un Sistema de Automatización Computarizado

- ◆ *Computadora Industrial*
- ◆ *Interfases de Adquisición de Datos y Control*
- ◆ *Software o programas*

Computadoras Industriales

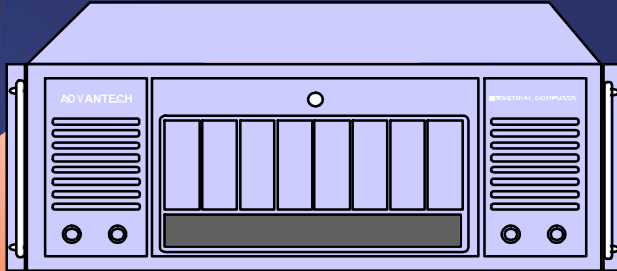
Características de Gabinetes (1)

✓ Robustos,
resistentes a
vibración e
impactos



Computadoras Industriales

Características de Gabinetes (2)

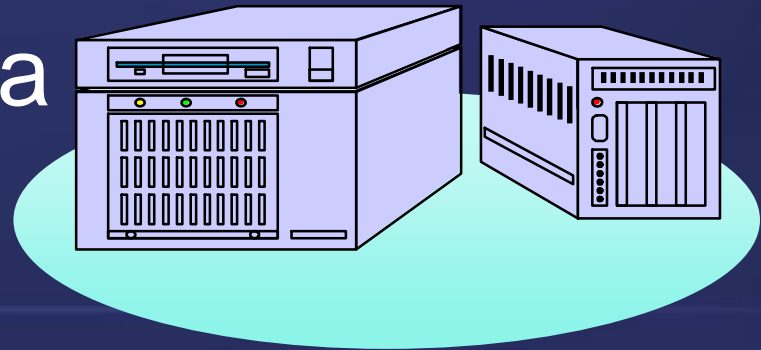


- ✓ Fuentes de gran capacidad

Computadoras Industriales

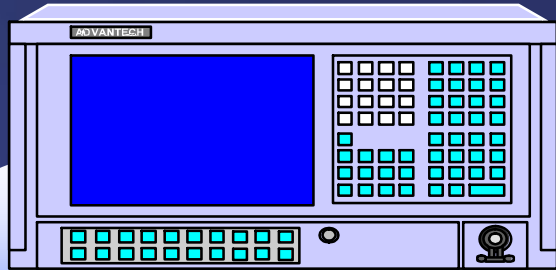
Características de Gabinetes (3)

- ✓ Tamaños apropiados para montaje industrial



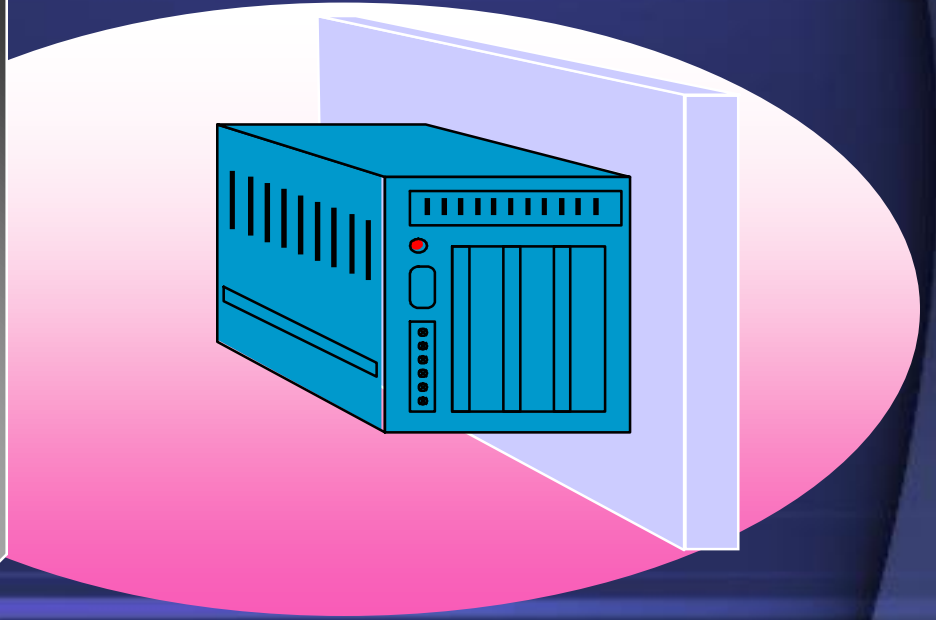
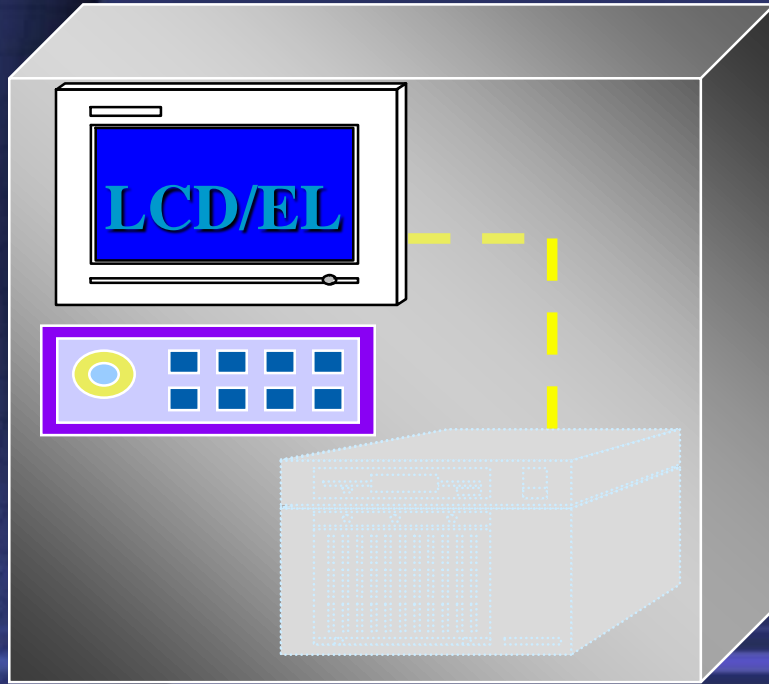
Computadoras Industriales

Características de Gabinetes (4)



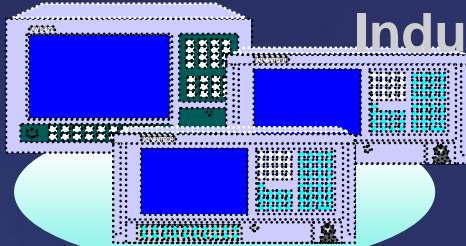
✓ Para
intemperie

Montajes típicos

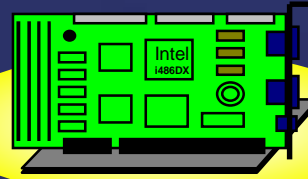
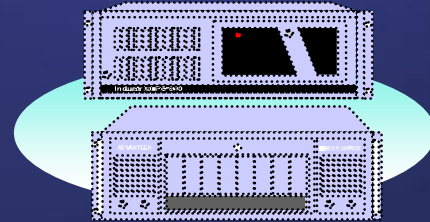


Plataformas PC para Automatización Industrial

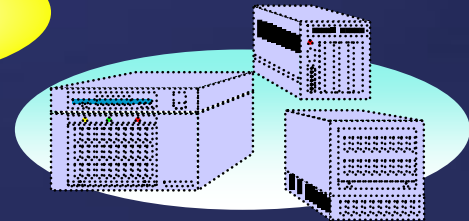
Estaciones de Trabajo
Industriales



IPCs para Rack de 19"



CPUs
Industriales



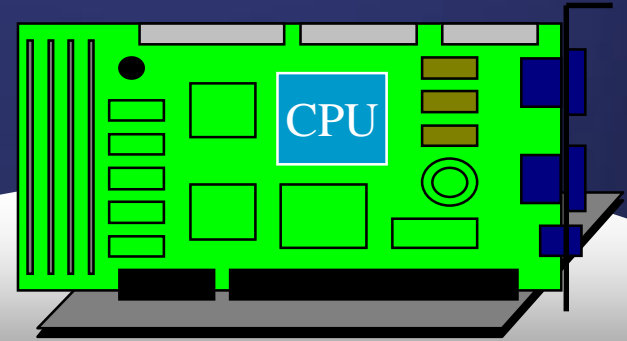
Mini Estaciones de Trabajo

IPCs para montaje
en pared

Computadoras Industriales

Características de CPUs (1)

- Calidad industrial en los componentes
- Procesadores de tipo TODO-EN-UNO
- Desde 286 hasta Pentium

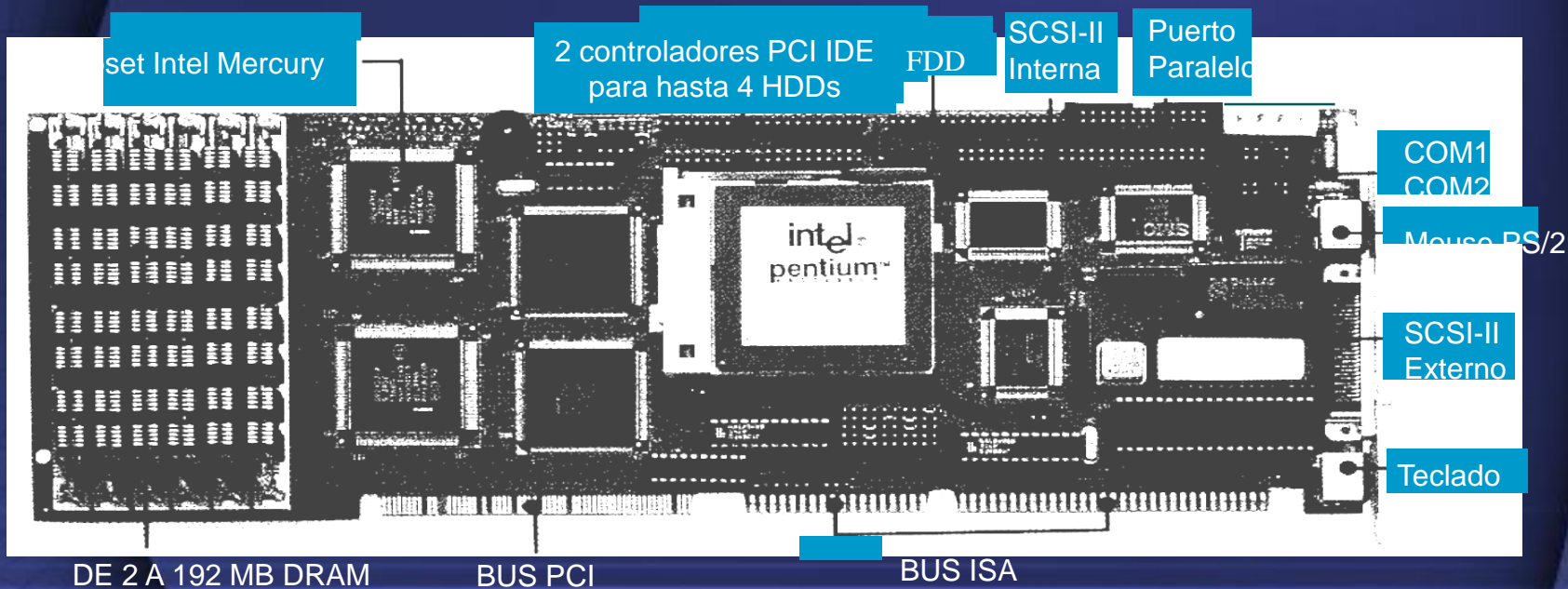


Computadoras Industriales

Características de CPUs (2)

- ❑ MTBF alto
- ❑ WATCHDOG Timer: protege a la IPC contra fallas de software
- ❑ Amplios rangos de operación en temperatura, humedad y vibración

PCA-6157 Procesador Pentium+PCI+ISA+SCSI



PCM-5860 Computadora Pentium en una Tarjeta

Intel Pentium
75/90/100/120
/133/150

Hasta 512 KB
de memoria cache

Conector
10BASE-2/5
(AUI)

Conector
10BASE-T

Hasta 64 MB
DRAM

Slot PCI

Chipset VIA

Controlador
VGA/pantalla plana

Hasta 1MB de
memoria de video

Conector de teclado
y Mouse PS2

Conector de Poder

Interfase de FDD

IDE mejorada de
Bus Maestro

Puerto de
impresora

Bidireccional
BIOS AWARD
Flash

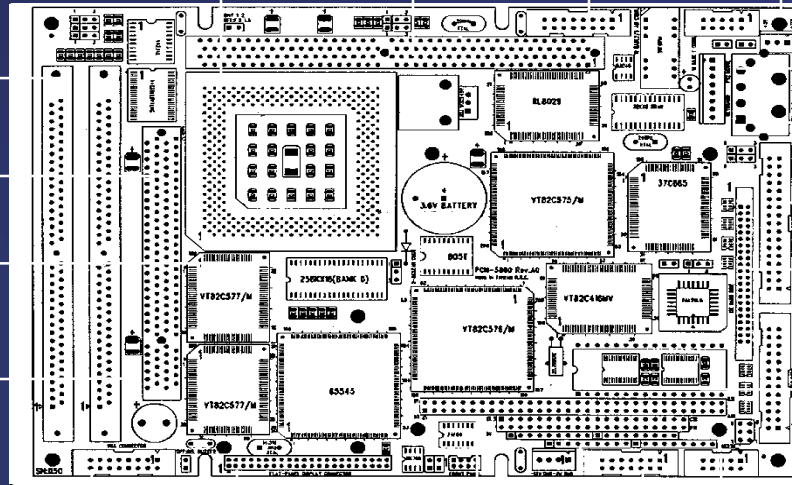
Conector para
CRT

Conectores
para Pantalla
Plana

PC/104
connector

COM2
RS-232
422/485

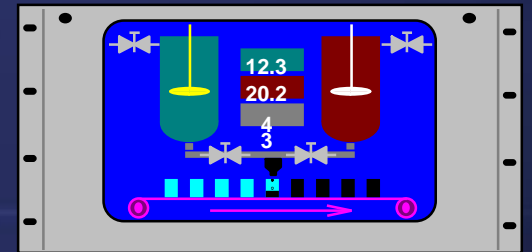
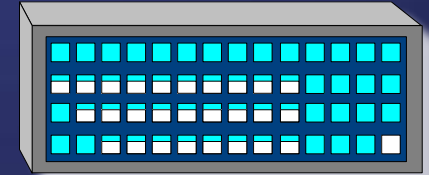
COM1
RS-232



Computadoras Industriales

Componentes Adicionales

- Teclados de Membrana
- Discos de Estado Sólido
- Mouse industrial
- Touchscreen

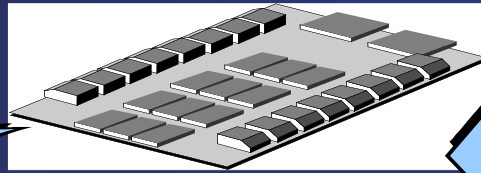
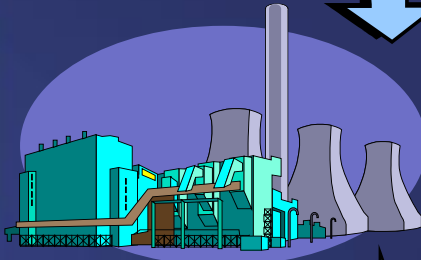


Interfases para Adquisición de Datos y Control

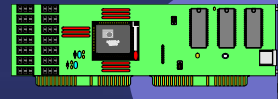
- ◆ **Interfases Locales**
- ◆ **Transmisores Remotos de
BUS**

Sistema de Adquisición de Datos y Control con Interfaces Locales

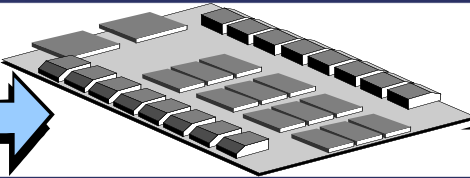
Acondicionamiento de Señales



D/A, D/O



Actuador/Sensor



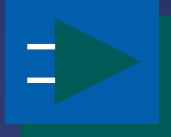
A/D, D/I

Acondicionamiento de Señales

Interfases Locales para Adquisición de Datos y Control

Entradas/Salidas Analógicas

Entradas Analógicas



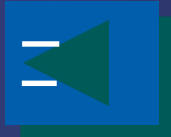
25 KHz a 330 KHz

12 a 16 bits

8 a 32 canales

Entradas Aisladas

Salidas Analógicas

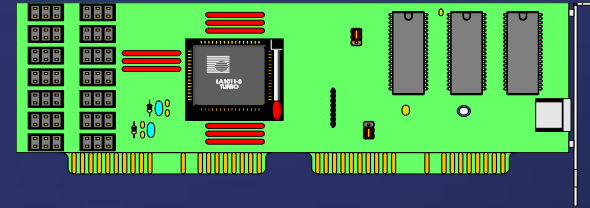


12 a 16 bits

Salida de Voltaje

Salida de Corriente

Salidas Aisladas



Acondicionamiento de Señales



Ganancia de Amplificadores
hasta por 1000

Termopares, RTDs

Multiplexores hasta

256 canales

Aislamiento de Señales

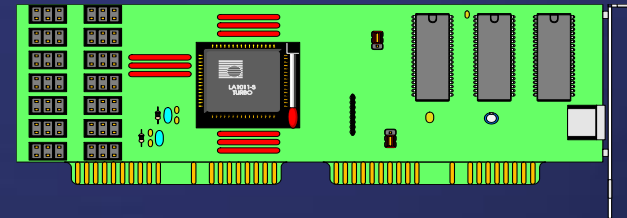
Interfases Locales para Adquisición de Datos y Control

Entradas/Salidas Digitales

Entradas Digitales

01010
10111
10001

>1Mhz, 24 a 144 bits
Entradas TTL y Aisladas
Interrupciones por Vector
Contadores



Salidas Digitales

10111
10011
00100

24 a 144 bits
Salidas TTL, a Relay y
Aisladas
Timers

Salidas a Relevador



De 30 VDC hasta 250 VAC
Formato A, Formato C y SSR
Compatibles con Opto-22

El Módulo de Adquisición de Datos y Control

Especificaciones de la tarjeta.

Entrada Analógica.

Canales:	8 entradas.
Resolución:	12 bits, aproximaciones sucesivas.
Rango de Entrada:	Bipolar, +/- 5V.
Convertidor:	AD574 o equivalente.
Tiempo de Conversión:	25 μ seg.
Exactitud:	0.015% leyendo +/- 1 LSB.
No linealidad:	+/- 1 bit.
Modo de disparo:	Disparo por software.
Transferencia de datos:	por programa.

El Módulo de Adquisición de datos y control.

Salida Analógica.

Canales:	1 canal.
Resolución:	12 bits.
Rango de Salida:	De 0 a +5V ó de 0 a +10V.
Tiempo de establecimiento:	30 μ seg.
Voltaje de Referencia:	Interno -5V y -10V (+/- 0.05V).
Convertidor:	AD7541KN, o equivalente.
No linealidad:	+/- 0.5 LSB.
Capacidad de Salida:	+/- 5mA máx.

El Módulo de adquisición de datos y control.

Entrada Digital.

Canales:	16 bits, TTL.
Voltaje de Entrada:	Bajo 0.8V máx; alto 2V mín.
Carga de entrada:	Bajo 0.4mA máx a 0.5V Alto 0.05mA máx a 2.7 V

Salida Digital.

Canales:	16 bits, TTL.
Carga de Salida:	Bajo 8mA a 0.5V máx. Alto 0.4mA a 2.4V mín.



Especificaciones generales.

Consumo de potencia

+5v: 100mA típico; max. 500mA.

+12V: 40mA típico; max. 100mA.

-12V: 20mA típico; max. 50mA.

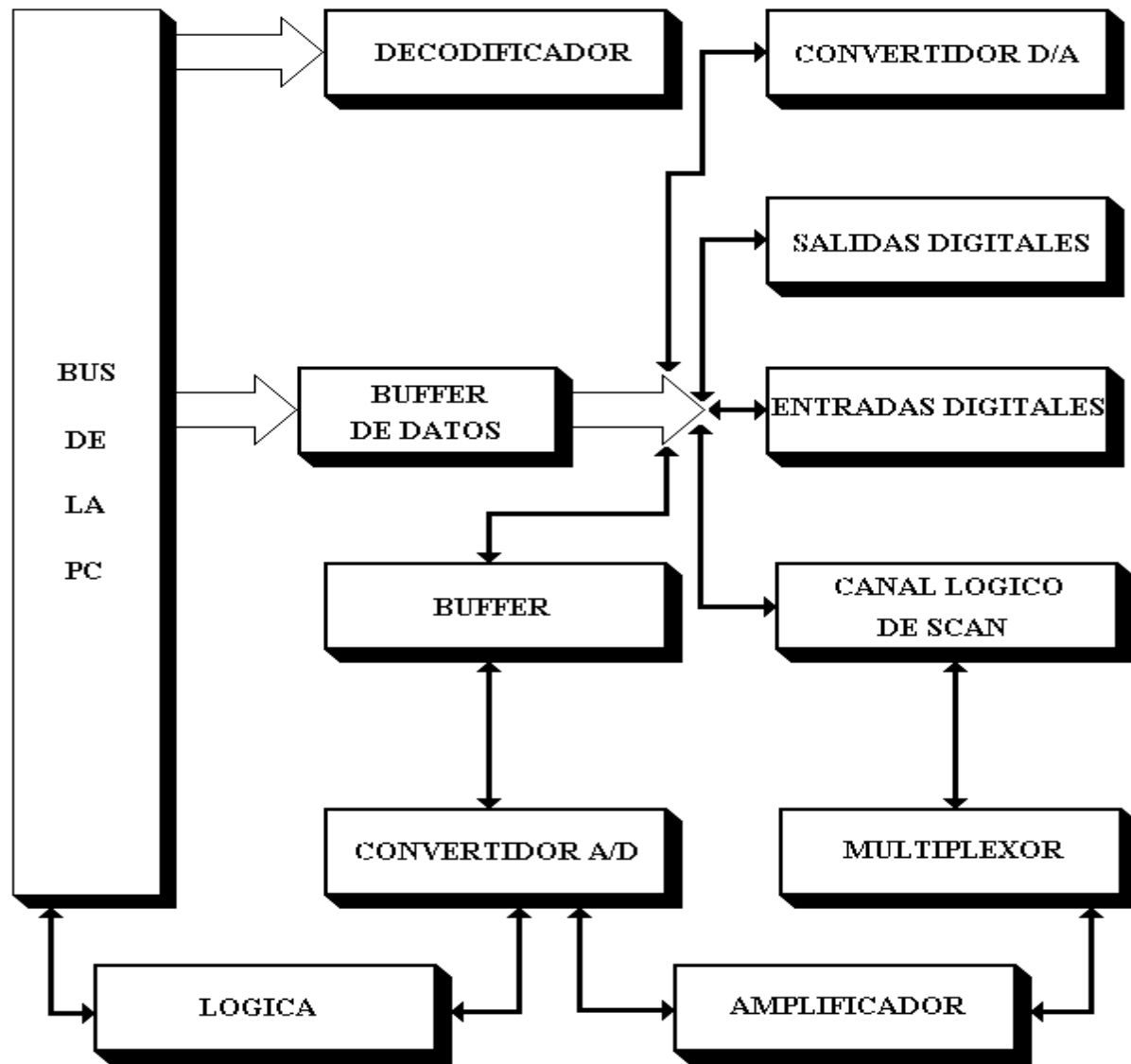
Conector I/O

Conector de 20 terminales para los puertos de entrada y salida analógicas/digitales.

Base Dirección I/O

Requiere 16 localidades consecutivas de dirección.

La dirección se define por los interruptores, en las líneas A9 - A4 (De fábrica es la 220 Hex.)





DIPLOMADO EN AUTOMATIZACIÓN



32	GND	-IOCHCK	1
33	RESDRV	D7	2
34	+5V	D6	3
35	IRQ9	D5	4
36	-5V	D4	5
37	DREQ2	D3	6
38	-12V	D2	7
39	-0WS	D1	8
40	+12V	D0	9
41	GND	IOCHRDY	10
42	-SMEMW	AEN	11
43	-SMEMR	A19	12
44	-IOW	A18	13
45	-IOR	A17	14
46	-DACK3	A16	15
47	DREQ3	A15	16
48	-DACK1	A14	17
49	DREQ1	A13	18
50	-REFSH	A12	19
51	SYSCLK	A11	20
52	IRQ7	D10	21
53	IRQ6	A9	22
54	IRQ5	A8	23
55	IRQ4	A7	24
56	IRQ3	A6	25
57	-DACK2	A5	26
58	TC	A4	27
59	ALE	A3	28
60	+5V	A2	29
61	14.3MHZ	A1	30
62	GND	A0	31

PC62
IBMJ1



7.2 TARJETA DE ADQUISICIÓN Y CONTROL.

Existe una tarjeta disponible comercialmente, la PC-MULTILAB modelo PCL-711 de Advantech Co. Ltd.; es una tarjeta fácil de usar, con múltiples funciones de adquisición de datos para IBM PC/XT/AT y compatibles. Las especificaciones de esta tarjeta hacen ideal el amplio rango de aplicaciones en la industria y ambientes de laboratorio. Esas aplicaciones incluyen adquisición de datos, control de procesos, pruebas automáticas y automatización de fábricas. Como se verá más adelante es viable reproducir esta tarjeta para que sea compatible con el software y el hardware de la PC-MultiLab, pero modificando la tarjeta PCL-711 con componentes disponibles en el mercado nacional.



DIPLOMADO EN AUTOMATIZACIÓN



7.2.2 SELECCIÓN DE LA DIRECCIÓN BASE

Muchos dispositivos periféricos e interfaces de tarjetas son controlados a través de los puertos de entrada-salida (I/O). Esos puertos son direccionados usando el espacio de direcciones de puertos de entrada-salida (I/O). La dirección base se va a seleccionar de la misma manera que la PC-MultiLab por medio de interruptores tipo DIP's. La dirección base de la presente tarjeta va a ser a partir de la 220H, aunque, sí se necesita otro rango de direcciones, sólo se modifican las posiciones del DIP, lo cual puede ayudar a que se puedan colocar diferentes dispositivos en direcciones apropiadas.

I/O Direcciones	POSICIÓN DEL SWITCH (SW1)					
Rango(Hex)	1	2	3	4	5	6
	A9	A8	A7	A6	A5	A4
000-00F	0	0	0	0	0	0
100-10F	0	1	0	0	0	0
200-20F	1	0	0	0	0	0
210-21F	1	0	0	0	0	1
220-22F	1	0	0	0	1	0
300-30F	1	1	0	0	0	0
3F0-3FF	1	1	1	1	1	1

Nota: *On = 0, Off = 1

*A4...A9 corresponden a las líneas del bus de direcciones de la PC.



DIPLOMADO EN AUTOMATIZACIÓN



7.2.3 ASIGNAMIENTO DE LAS TERMINALES DE LOS CONECTORES.

La PCL-compatible está equipada con tres conectores de 20 terminales, los cuales son accesibles desde las ventanas de los slots de la PC. Todos esos conectores pueden ser conectados con el mismo tipo de conector.

Los siguientes diagramas muestran el asignamiento de las terminales de cada conector:

A/D:	Entrada analógica
A.GND:	Tierra analógica.
D/A:	Salida analógica.
D/O:	Salida digital.
D/I:	Entrada digital.
D.GND:	Tierra digital.

Entradas analógicas y salida analógica:

A/D 0	1	2	A.GND
A/D 1	3	4	A.GND
A/D 2	5	6	A.GND
A/D 3	7	8	A.GND
A/D 4	9	10	A.GND
A/D 5	11	12	A.GND
A/D 6	13	14	A.GND
A/D 7	15	16	A.GND
D/A	17	18	A.GND
A.GND	19	20	A.GND

Salida digital:

D/O 0	1	2	D/O 1
D/O 2	3	4	D/O 3
D/O 4	5	6	D/O 5
D/O 6	7	8	D/O 7
D/O 8	9	10	D/O 9
D/O 10	11	12	D/O 11



DIPLOMADO EN AUTOMATIZACIÓN



7.2.4 ESTRUCTURA DE LOS REGISTROS Y FORMATO

Hay 16 registros direccionables a partir de la dirección base que es la 220H, esto se muestra a continuación:

Mapa de direcciones del puerto de entrada-salida (I/O)

Localización	Lectura	Escritura
Base+0	No usado	No usado
Base+1	No usado	No usado
Base+2	No usado	No usado
Base+3	No usado	No usado
Base+4	A/D bajo byte	D/A bajo byte
Base+5	A/D alto byte	D/A alto byte
Base+6	D/I bajo byte	No usado
Base+7	D/I alto byte	No usado
Base+8	No usado	No usado
Base+9	No usado	No usado
Base+10	No usado	Mux Scan channel
Base+11	No usado	No usado
Base+12	No usado	Software A/D trigger
Base+13	No usado	D/O bajo byte
Base+14	No usado	D/O alto byte
Base+15	No usado	No usado



7.2.5 REGISTRO DE DATOS DEL A/D

El byte bajo de datos esta en las posiciones D0(AD0) a D7(AD7) de BASE+4 y el byte alto esta en las posiciones D0(AD8) a D3(AD11) de BASE+5. El MSB es AD11 y el LSB es AD0. Los formatos son los siguientes:

Byte bajo del A/D

Base+4	D7	D6	D5	D4	D3	D2	D1	D0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Byte alto del A/D

Base+5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	STS	AD11	AD10	AD9	AD8

STS= 0 Dato leído

STS= 1 Dato no leído



7.2.6 REGISTROS DEL MUX SCAN

El registro mux scan es un registro solamente de escritura usando la dirección base+10. La parte baja de los datos suministra el número de canal del scan. El multiplexor del MUX cambia a un nuevo canal cuando se escribe a este registro.

Base+10	D7	D6	D5	D4	D3	D2	D1	D0
Scan								
Canal	X	X	X	X	X	CL2	CL1	CL0
Canal 0	X	X	X	X	X	0	0	0
Canal 1	X	X	X	X	X	0	0	1
Canal 2	X	X	X	X	X	0	1	0
Canal 3	X	X	X	X	X	0	1	1
Canal 4	X	X	X	X	X	1	0	0
Canal 5	X	X	X	X	X	1	0	1
Canal 6	X	X	X	X	X	1	1	0
Canal 7	X	X	X	X	X	1	1	1

Nota: CL2 a CL0 - Número de canal



DIPLOMADO EN AUTOMATIZACIÓN



7.2.7 REGISTROS DE I/O DIGITALES.

Este bloque ofrece 16 canales de entrada digital y 16 canales de salida digital. Los canales de I/O usan para la entrada las direcciones BASE+6 y BASE+7. Los puertos de salida son la dirección BASE+13 y BASE+14.

Las entradas y salidas de datos son digitales, a la entrada se leerá en formato de 16 bits para su posterior procesamiento y a la salida se escribirá en formato de 16 bits.

Base+6	D7	D6	D5	D4	D3	D2	D1	D0
	DI 7	DI 6	DI 5	DI 4	DI 3	DI 2	DI 1	DI 0
Base+7	D7	D6	D5	D4	D3	D2	D1	D0
	DI 15	DI 14	DI 13	DI 12	DI 11	DI 10	DI 9	DI 8
Base+13	D7	D6	D5	D4	D3	D2	D1	D0
	DO 7	DO 6	DO 5	DO 4	DO 3	DO 2	DO 1	DO 0
Base+14	D7	D6	D5	D4	D3	D2	D1	D0
	DO15	DO14	DO13	DO12	DO11	DO10	DO 9	DO 8



7.2.8 REGISTROS DEL CONVERTIDOR D/A

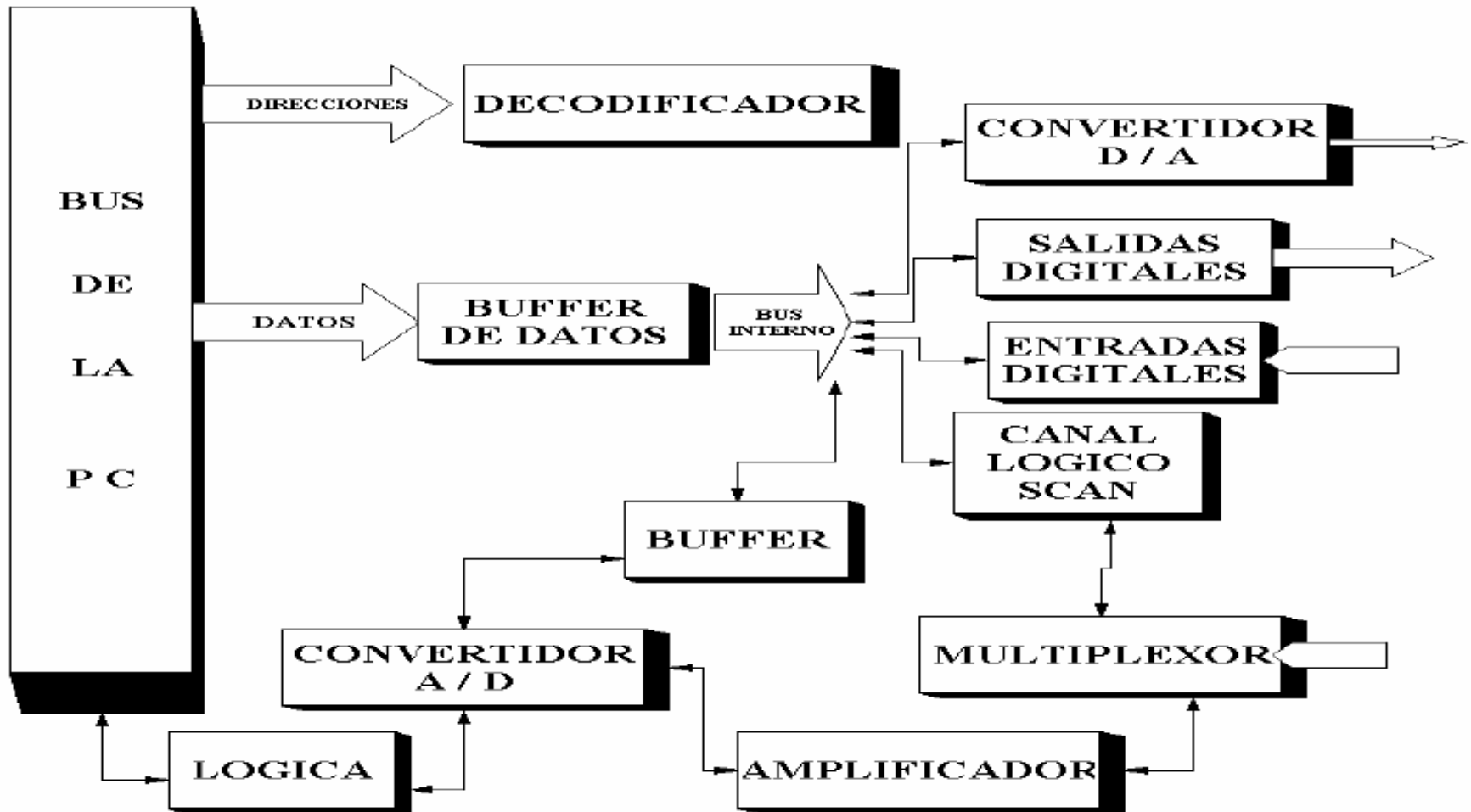
Los registros del convertidor D/A son registros de escritura, usan las direcciones $BASE+4$ y $BASE+5$. $DA0$ es el LBS y $DA11$ es el MSB. El formato es el siguiente:

Base+4	D7	D6	D5	D4	D3	D2	D1	D0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Base+5	D7	D6	D5	D4	D3	D2	D1	D0
	X	X	X	X	DA11	DA10	DA9	DA8



Los pasos a seguir para la conversión son los siguientes:

1. Poner el canal de entrada en el registro MUX scan.
2. Disparo por el puesto de escritura ($BASE+12$) con algún valor.
3. Esperar por SOS, leyendo el byte alto ($BASE+5$).
4. Lectura de datos $BASE+5$, $BASE+4$, primero debe leerse el byte alto.
5. Los datos binarios son convertidos a enteros.

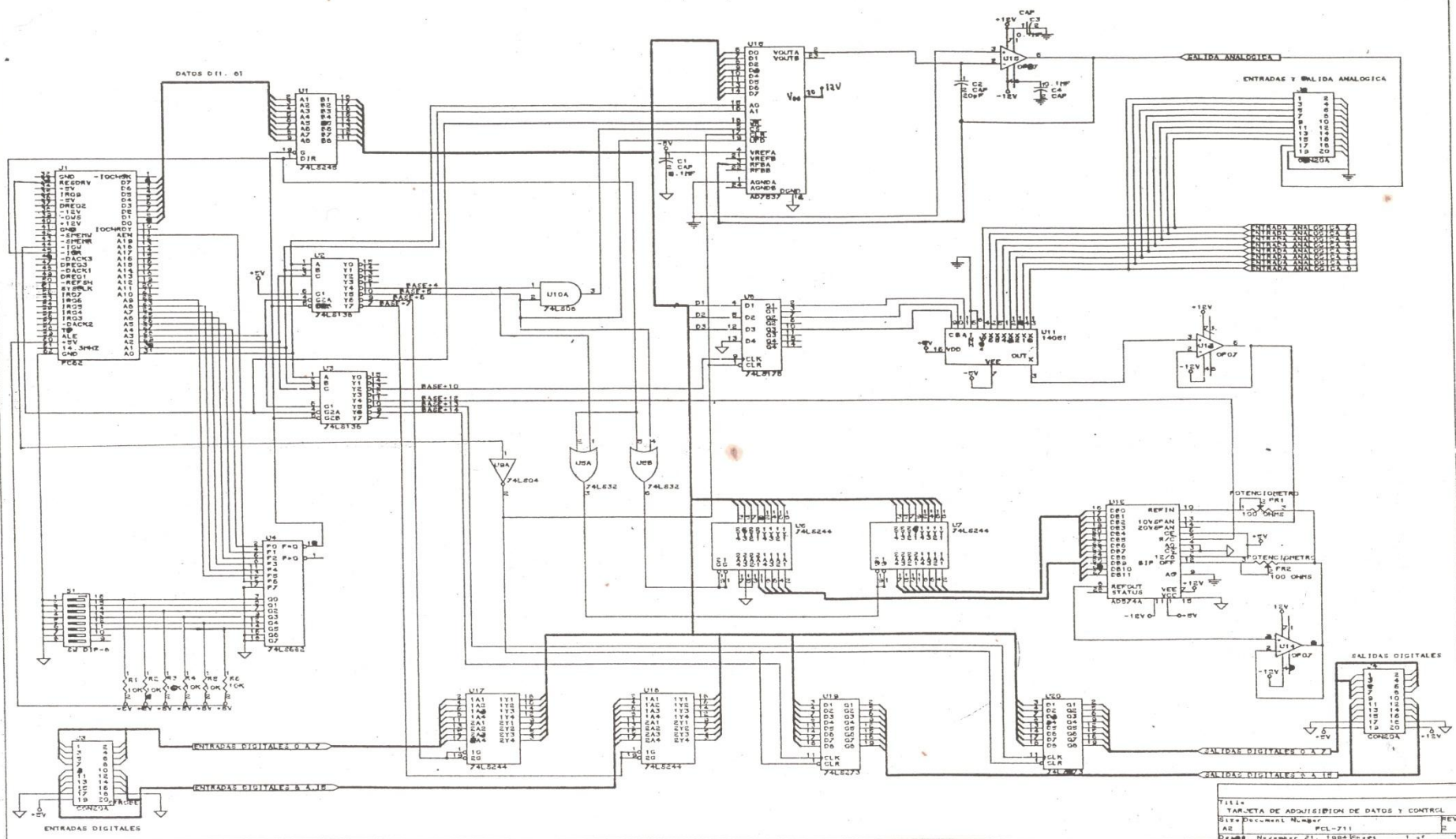




7.3.1 DESCRIPCIÓN DE CADA BLOQUE.

Las señales que se tomaron del bus de la PC a través de la ranura de expansión y las líneas de alimentación son:

- a).- 10 señales de direcciones, que son: A0, A1, ... A9.
- b).- 8 señales de datos, que son: D0, D1, ...D7.
- c).- 4 señales de control, que son: RESET DRV, -IOW, -IOR, AEN.
- d).- 4 líneas de alimentación, que son: GND, +5V, +12V, -12V.



TITULO: TARJETA DE ADQUISICION DE DATOS Y CONTROL
 Serie Document Number: PCL-711
 AE: PCL-711
 Fecha: Noviembre 21, 1982

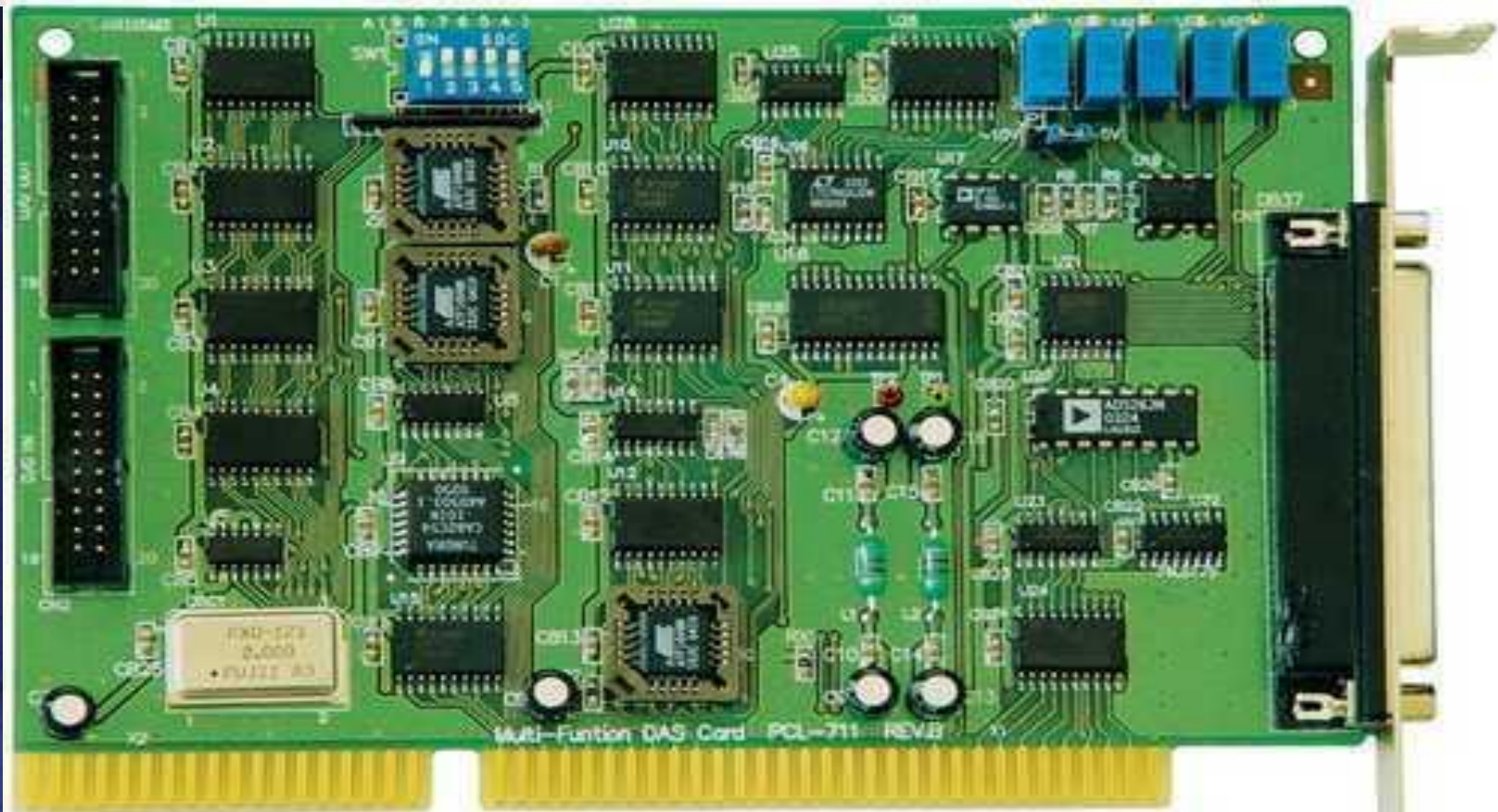


PCL-10120

ADAM-3920

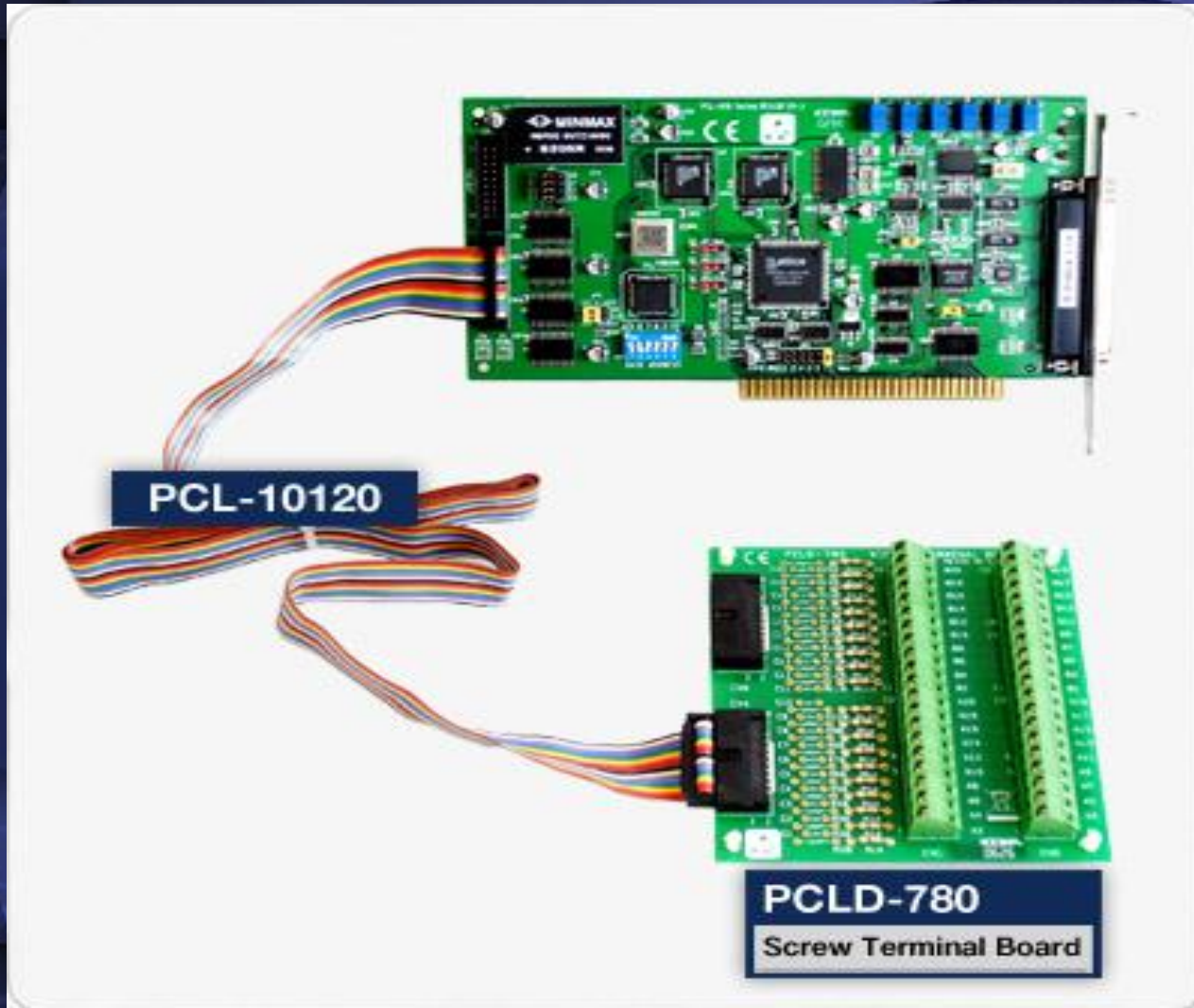
20-pin Flat Cable Wiring Terminal
for DIN-rail Mounting



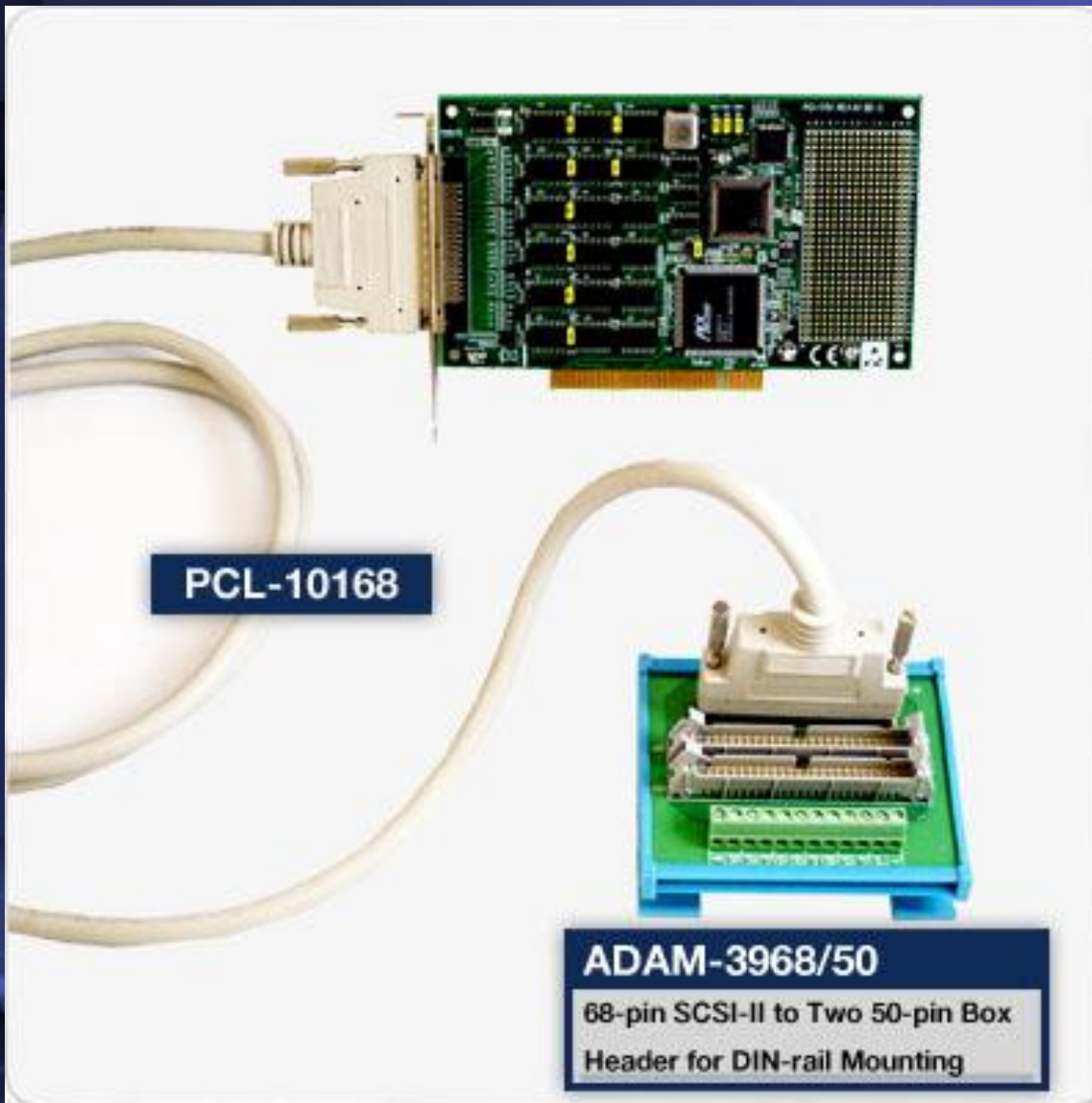


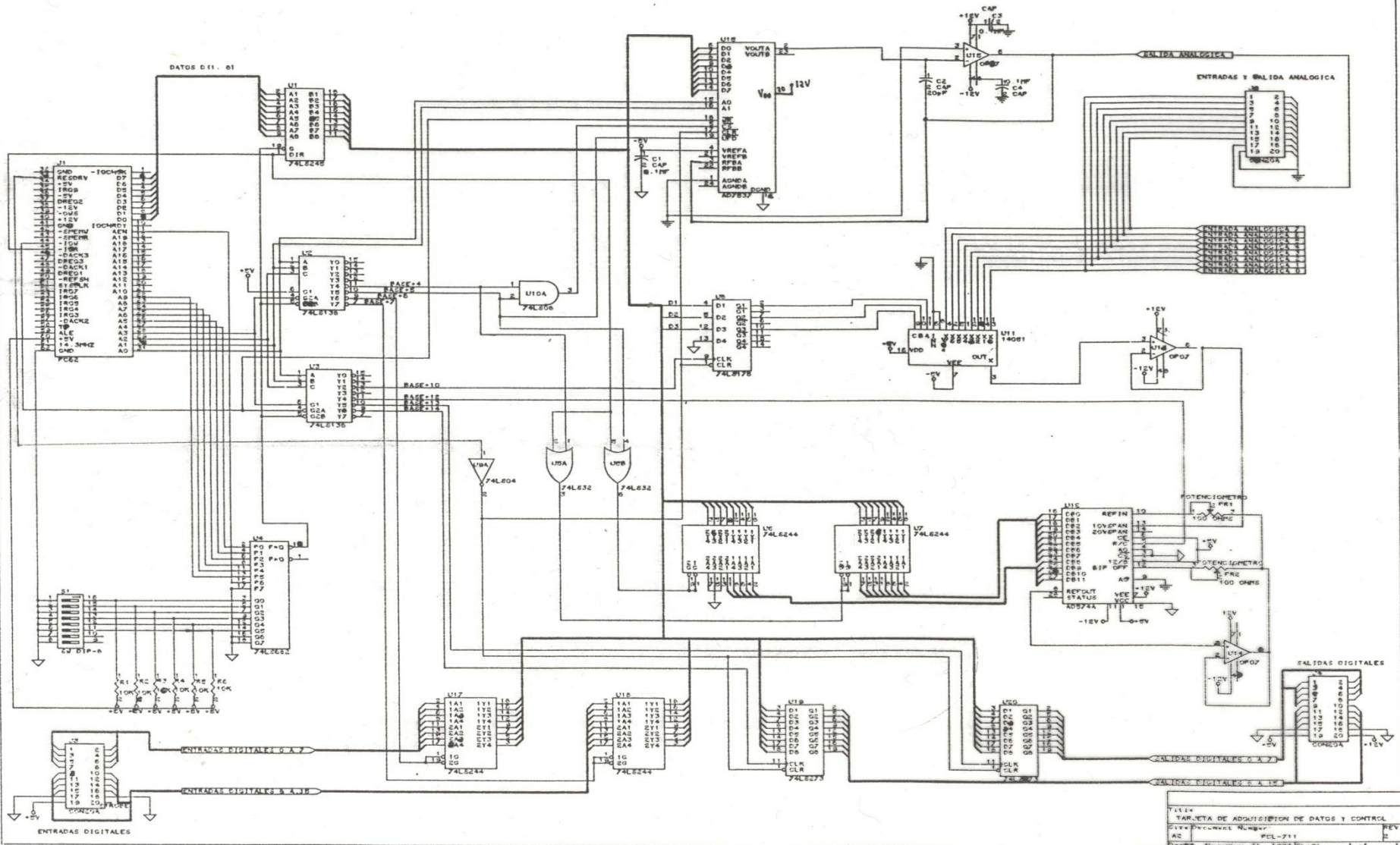
北京宏信博控科技有限公司











T-1114
 TARJETA DE ADQUISICION DE DATOS Y CONTROL
 Serie Proceso N.º 100
 AC PCL-711
 Rev. 01 - 1984

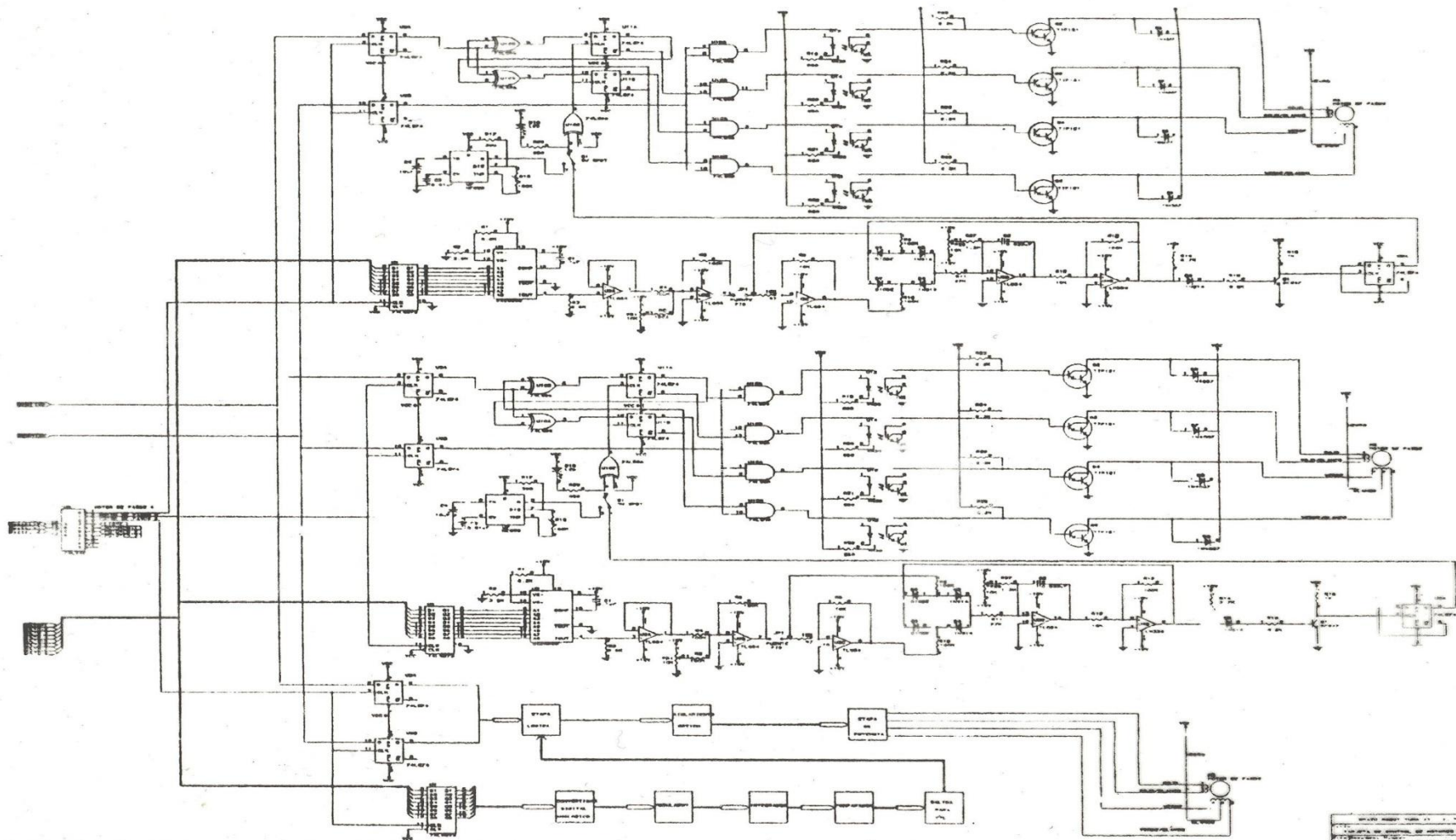


LISTA DE COMPONENTES DE LAS TARJETAS ELECTRÓNICAS



TARJETA DE ADQUISICION DE DATOS Y CONTROL Revised: January 23, 1995
PCL-711-951001-JAIME CID MONJARAZ Revision: 2
Bill Of Materials April 17, 1995 7:37:04 .Page. 1

Item	Quantity	Reference	Part
1	4	C1, C2, C3, C4	CAP 0.1uF
2	1	J1	PC62
3	3	J2, J3, J4	CON20A
4	2	PR1, PR2	100
5	1	RP1	10K
6	1	S1	SW DIP-8
7	1	U1	74LS245
8	2	U2, U3	74LS138
9	1	U4	74LS682
10	1	U5	74LS32
11	4	U6, U7, U17, U18	74LS244
12	1	U8	74LS175
13	1	U9	74LS04
14	1	U10	74LS08
15	1	U11	14051
16	1	U12	AD574A
17	3	U13, U14, U16	OP07
18	1	U15	AD7537
19	2	U19, U20	74LS273

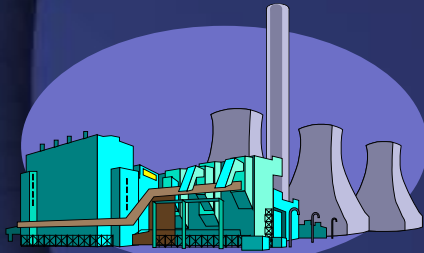


ESQUEMA DE LA TARJETA DE CONTROL DE MOTORES DE PASOS.

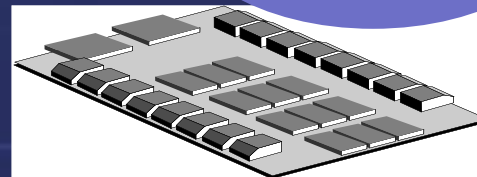
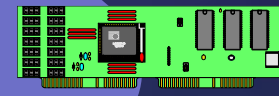
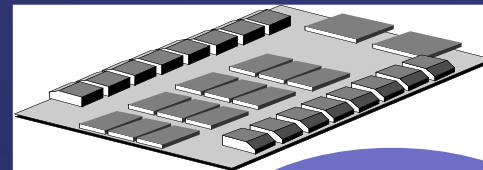
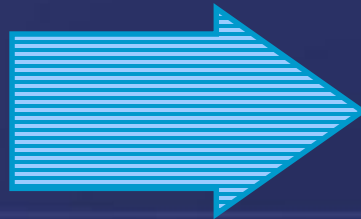
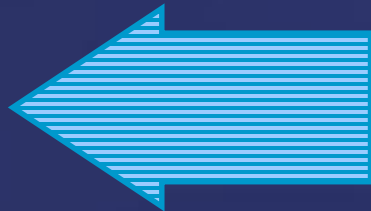
ESCUELA SUPERIOR DE INGENIERIA Y ARQUITECTURA
 DEPARTAMENTO DE ELECTRONICA Y SISTEMAS DE CONTROL
 LABORATORIO DE SISTEMAS DE CONTROL Y AUTOMATICA
 CARRERAS 10 y 11, No. 1000, Ciudad de Mexico, D.F.

Sistema de Adquisición de Datos y Control con Interfaces Locales

Cableado costoso



No modular



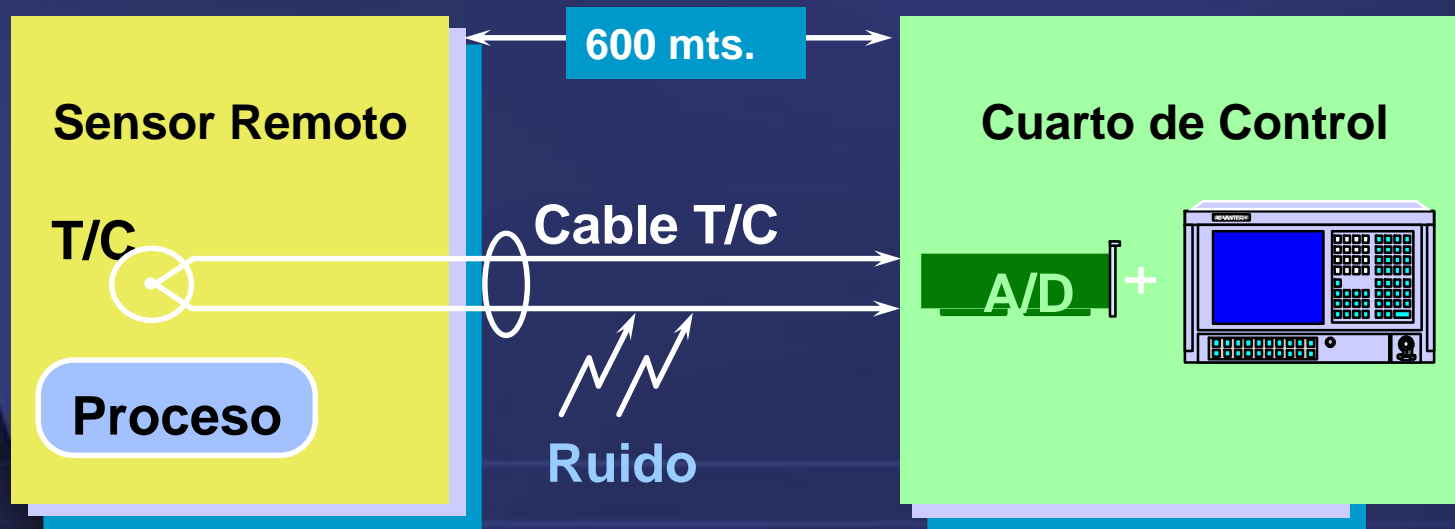
Transmisores Remotos vía Bus

Esquema típico



Transmisores Remotos vía Bus

Interfase cerca de la computadora



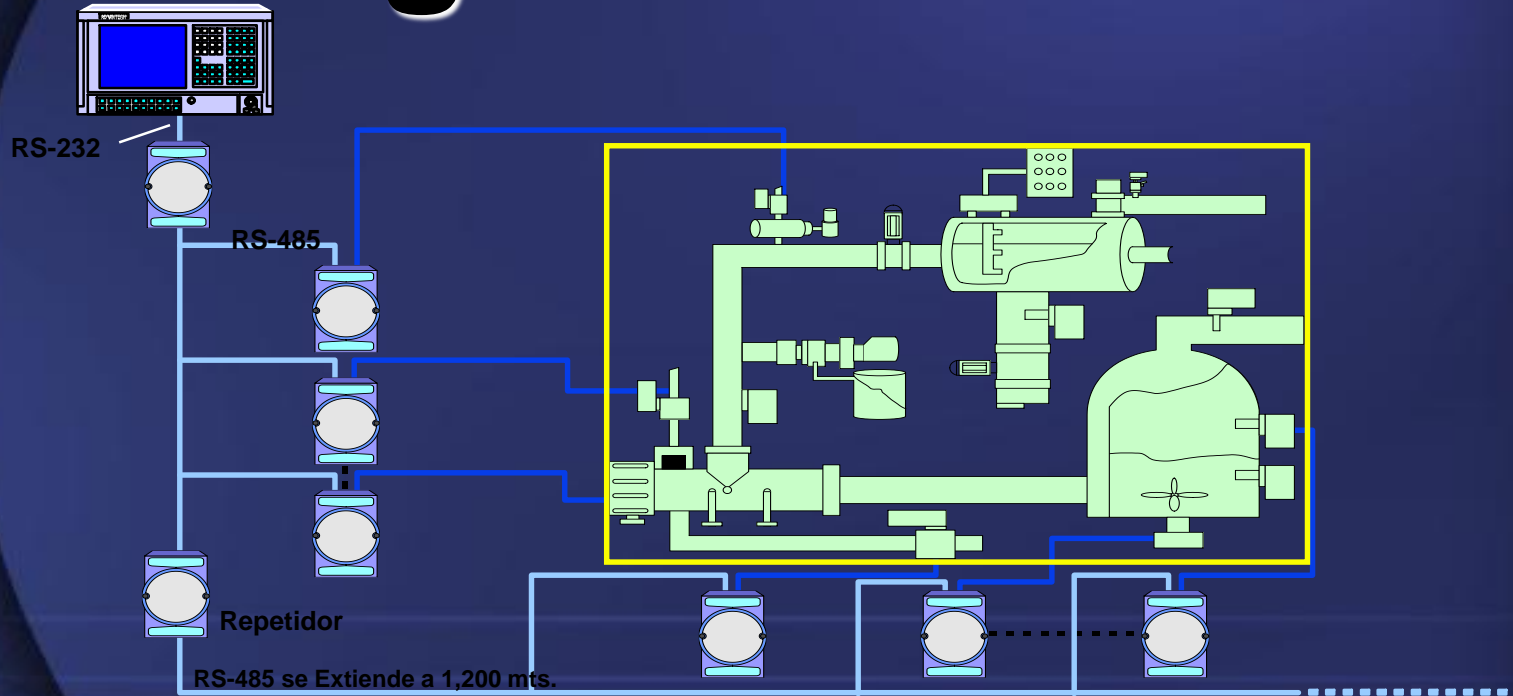
Transmisores

Remotos vía Bus

Interfase cerca del proceso



Transmisión Digital vía Bus

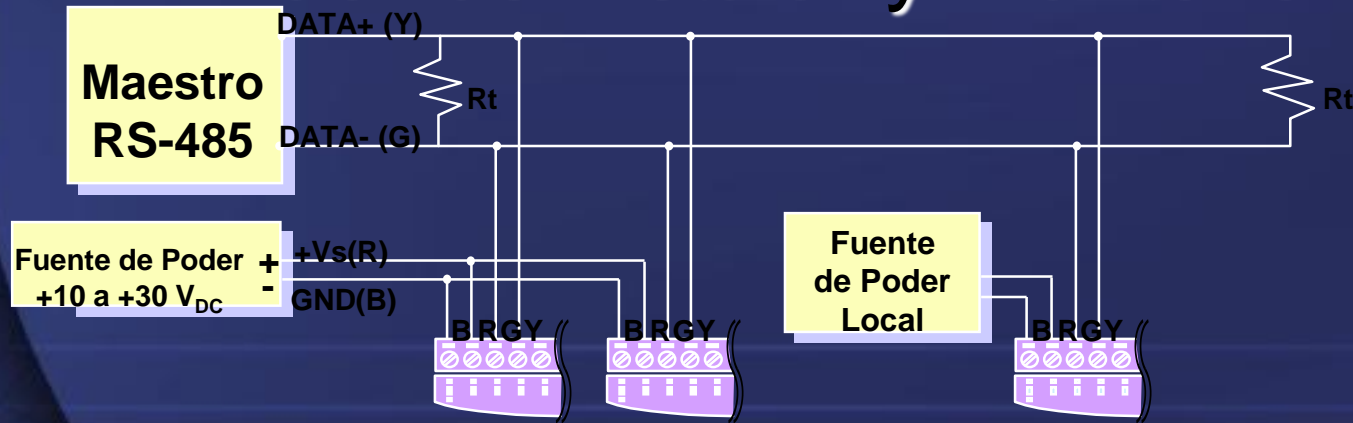


¿Porqué utilizar el bus RS-485?

- Es un estándar de Comunicación Industrial
- Reduce Costos de Cableado
 - *Comunicación por 2 hilos*
- Modular y Fácil de Expandir
 - *Red Multipunto*

Comunicación a 2 Hilos

- Costos de Cableado Bajos
- Fácil de Instalar y Mantener

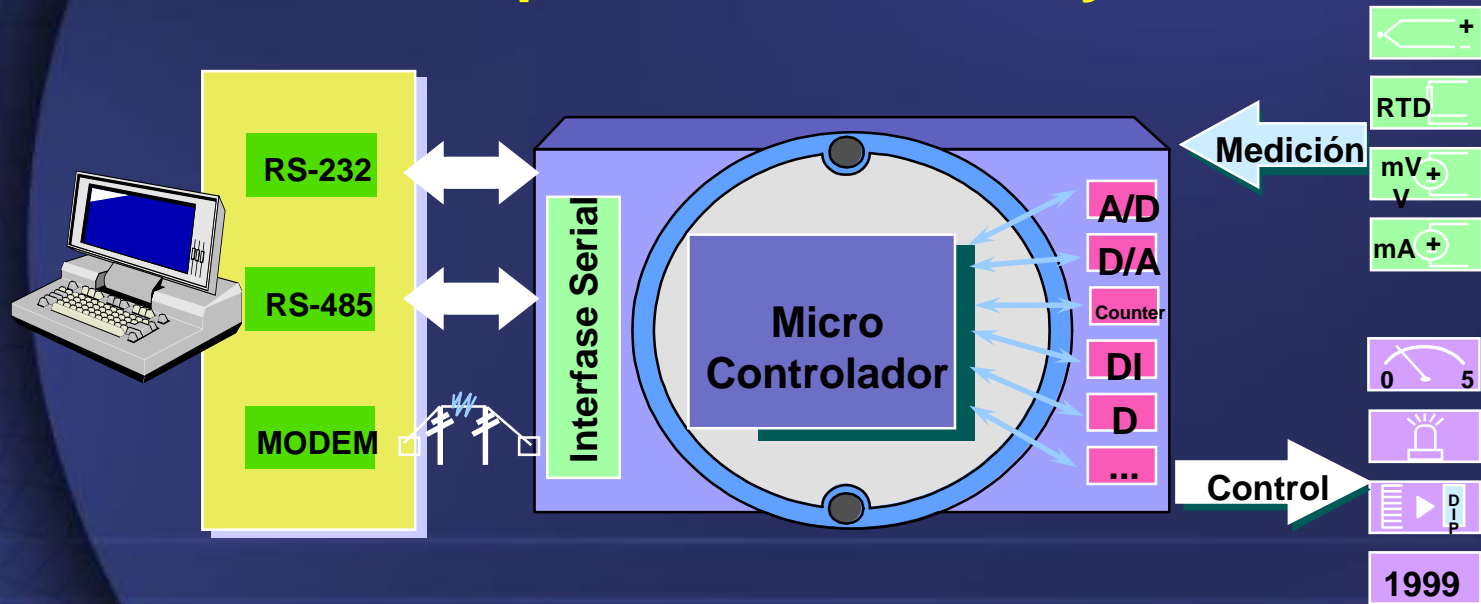


Transmisores Remotos ADAM Serie 4000

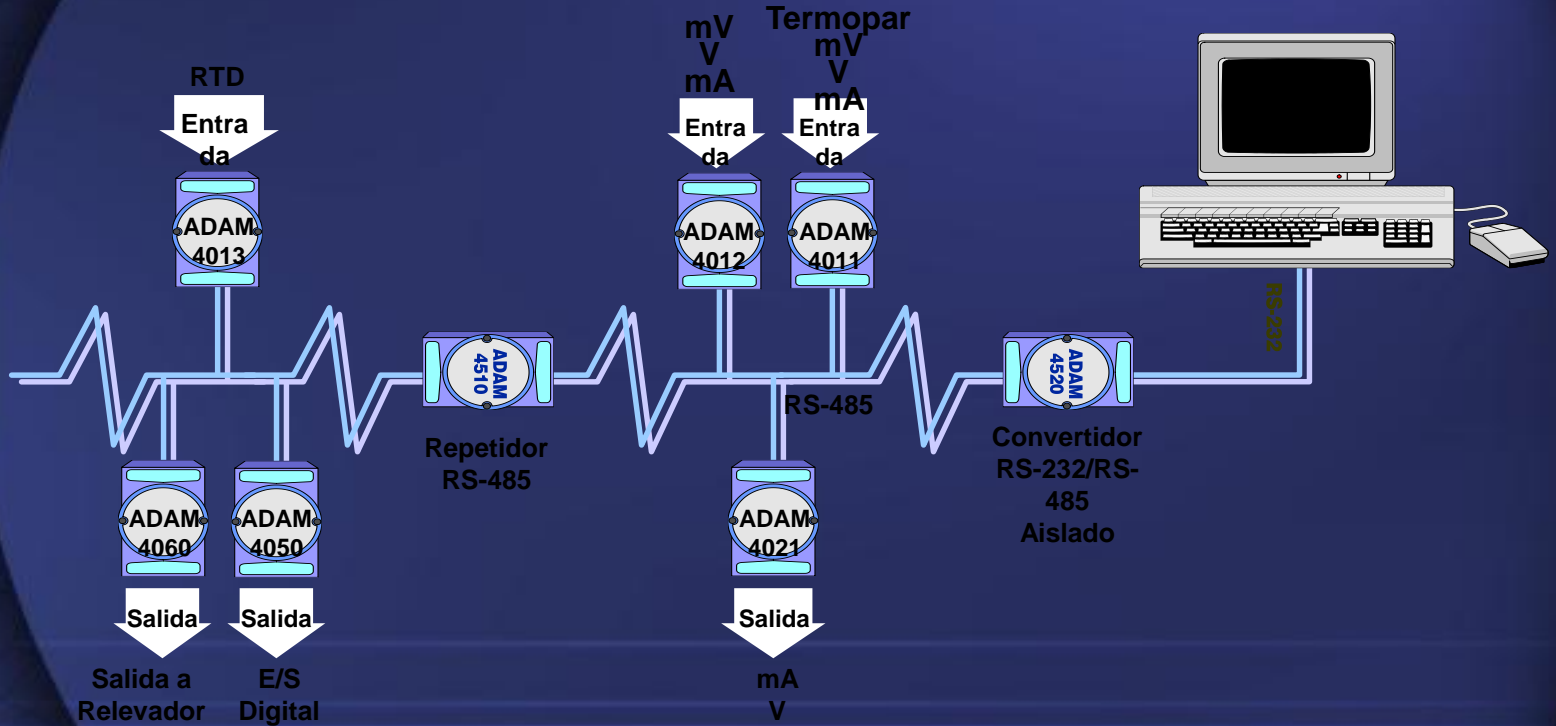
- ✓ Módulos de Entradas/Salidas Analógicas
- ✓ Módulos de Entradas/Salidas Digitales
- ✓ Módulos de Salida a Relevadores
- ✓ Módulos Contadores
- ✓ Módulos con Memoria Local
- ✓ Para ambientes Industriales

Módulos ADAM 4000

Para Adquisición de Datos y Control

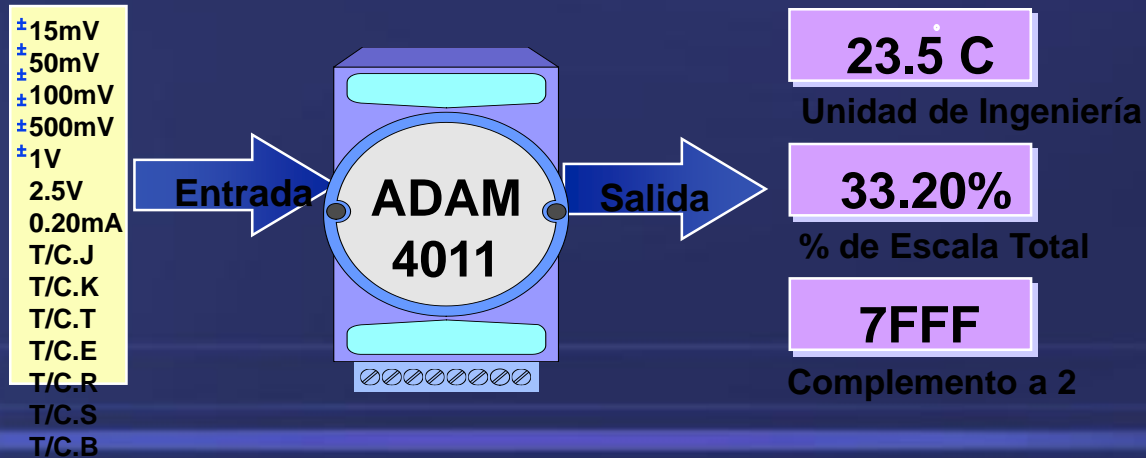


Red ADAM RS-485



Tipos y Rangos de Entradas/Salidas Programables

Sin potenciómetros ni Switches

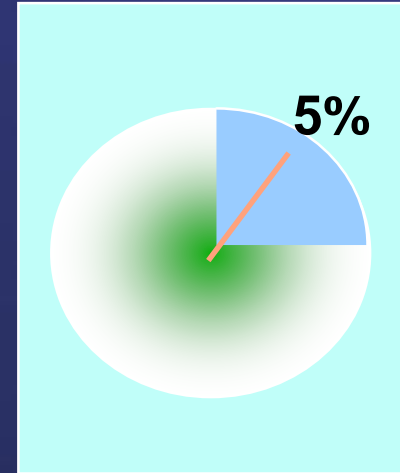


Capacidad de Operación sin Computadora

- ❖ Controlador ON/OFF Autónomo
- ❖ Alarmas Alta/Baja Configurables



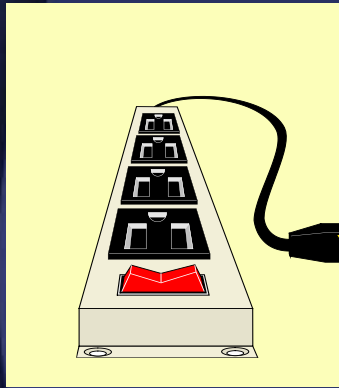
Listos para Trabajar en Ambientes Industriales



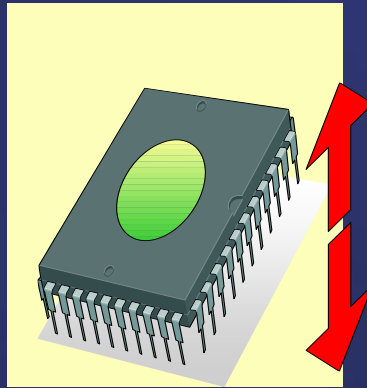
Temperatura de Operación: 0 a 70 o C

Humedad de Operación: 5%~95% (No condensante)

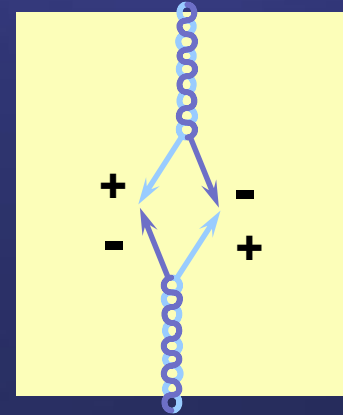
Diseñados para Aplicaciones Industriales



- ❖ Conexión/Desconexión en Línea



- ❖ Monitoreo de Voltaje para protección del EEPROM

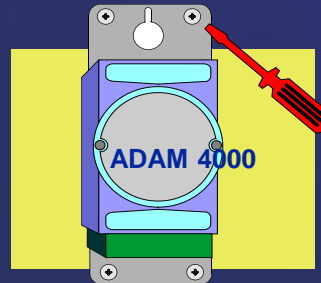


- ❖ Protección contra inversión de potencia

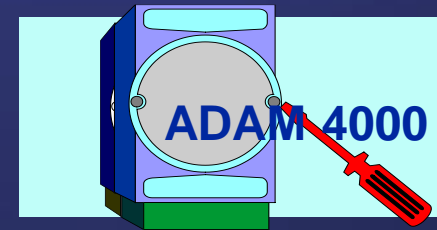
Instalación Flexible



Montaje en riel DIN



Montaje en
Pared o
Gabinete

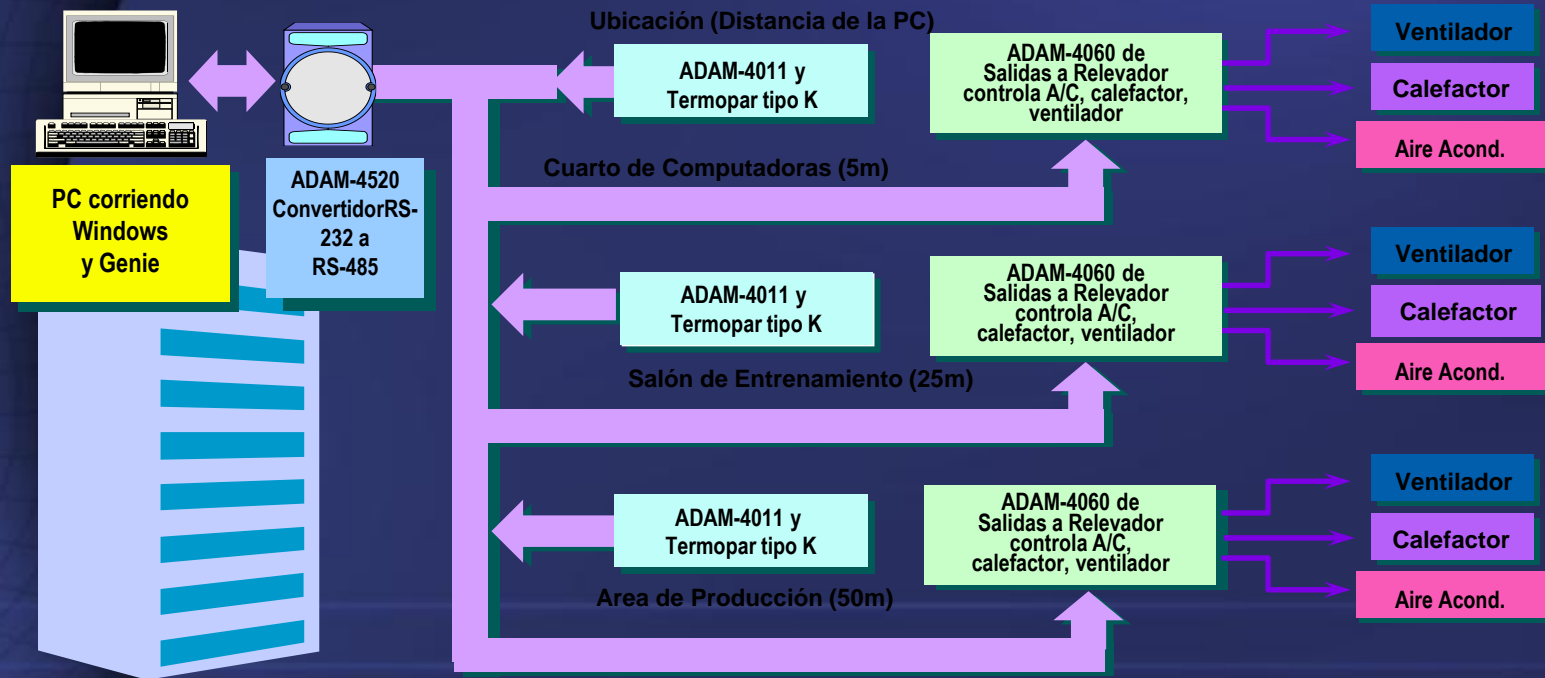


Apilamiento en
“piggyback”

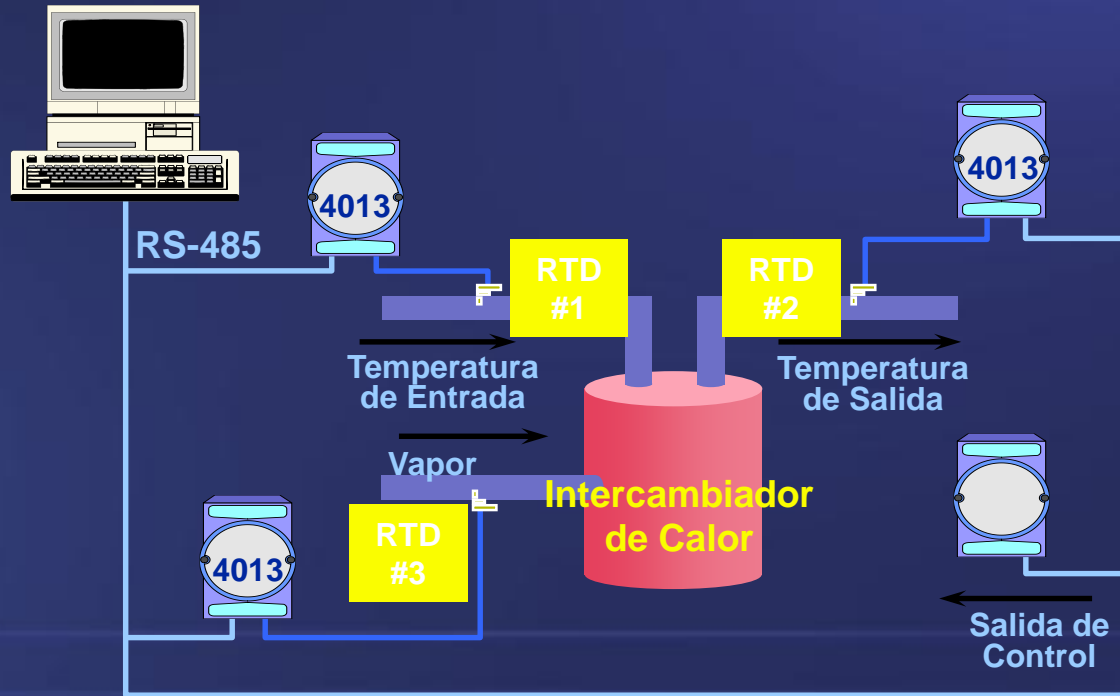
Aplicaciones

- ◆ Monitoreo y Control de Procesos
- ◆ Sistemas de Control Supervisorio y Adquisición de Datos (SCADA)
- ◆ Manejo de Energía
- ◆ Sistemas de Automatización de Edificios
- ◆ Registro de Datos Remoto
- ◆ Monitoreo y Control de Temperatura
- ◆ Pruebas de Productos

Sistema de Automatización de Edificio



Sistema de Monitoreo de Temperatura



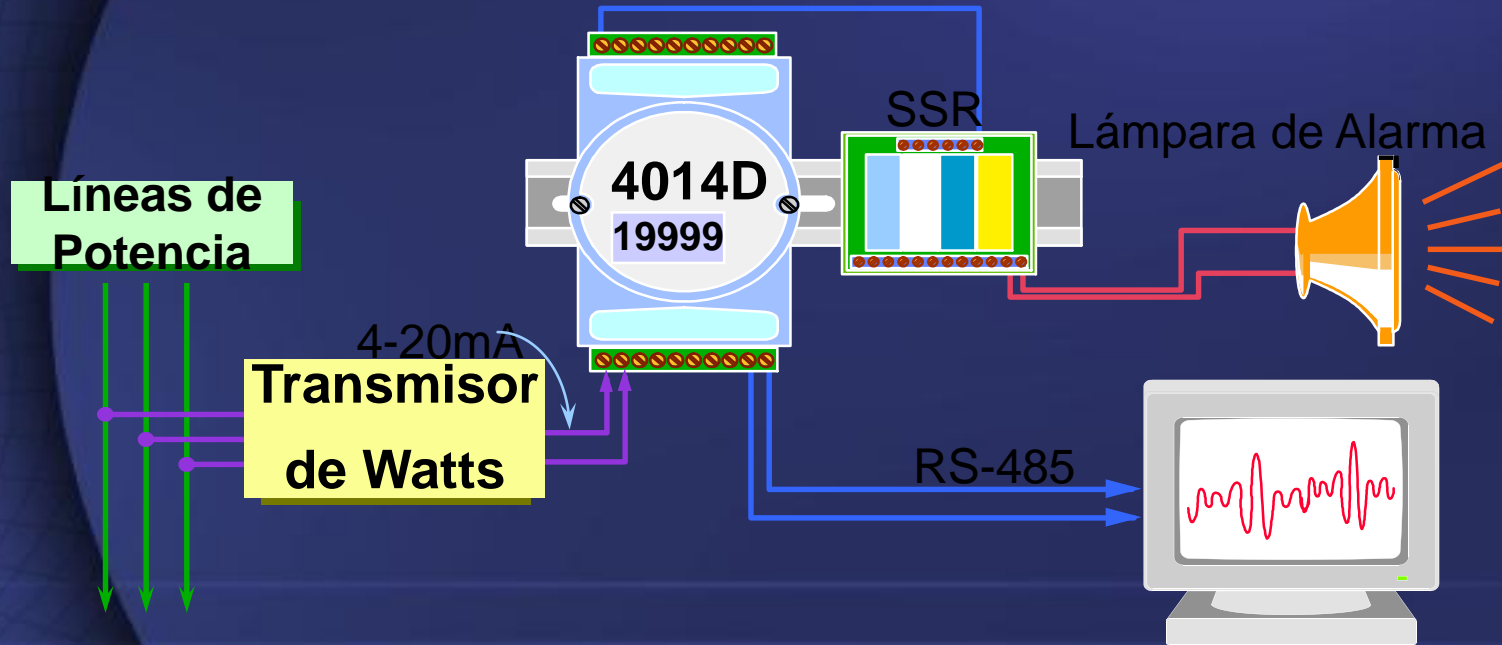
Sistema de Monitoreo y Alarma de Calidad del Aire



Cuarto de Control

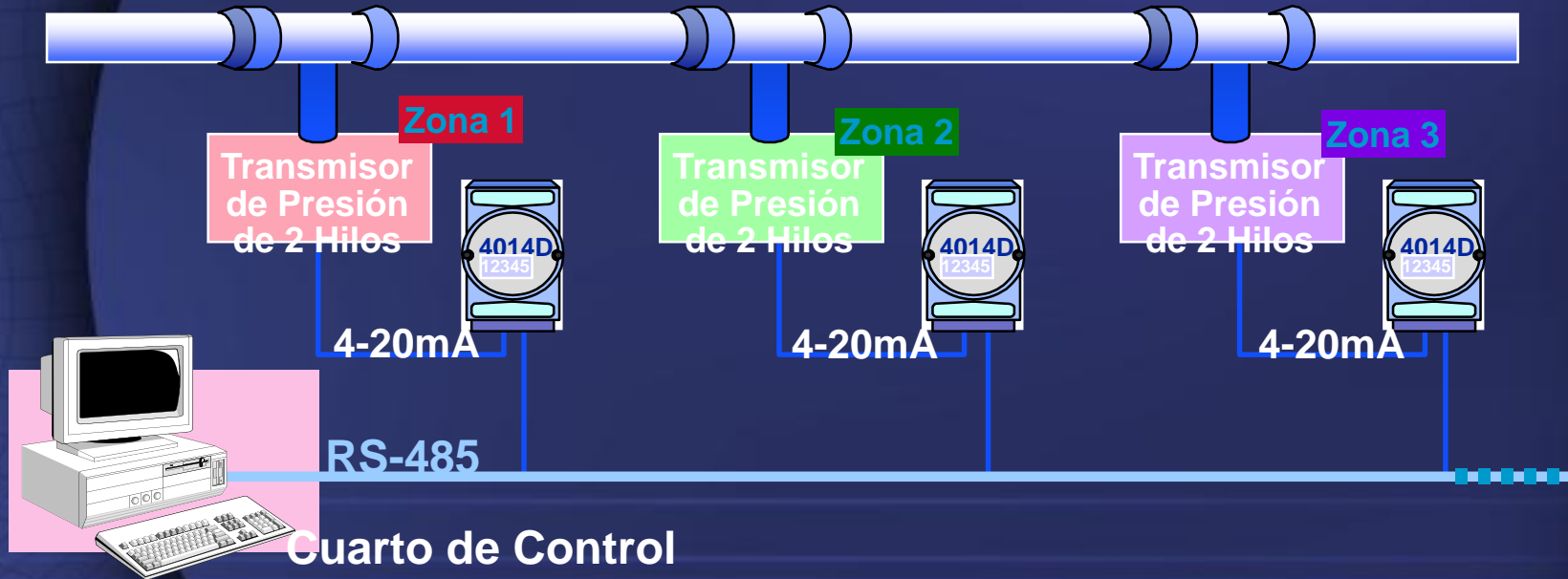


Sistema de Registro de Potencia



Sistema de Medición de Presión

Línea Presurizada







Reduce Your Sensor Wiring Costs with **ADAM** I/O Modules from Advantech

ADAM-5000 Series
Distributed Data Acquisition
and Control Solutions

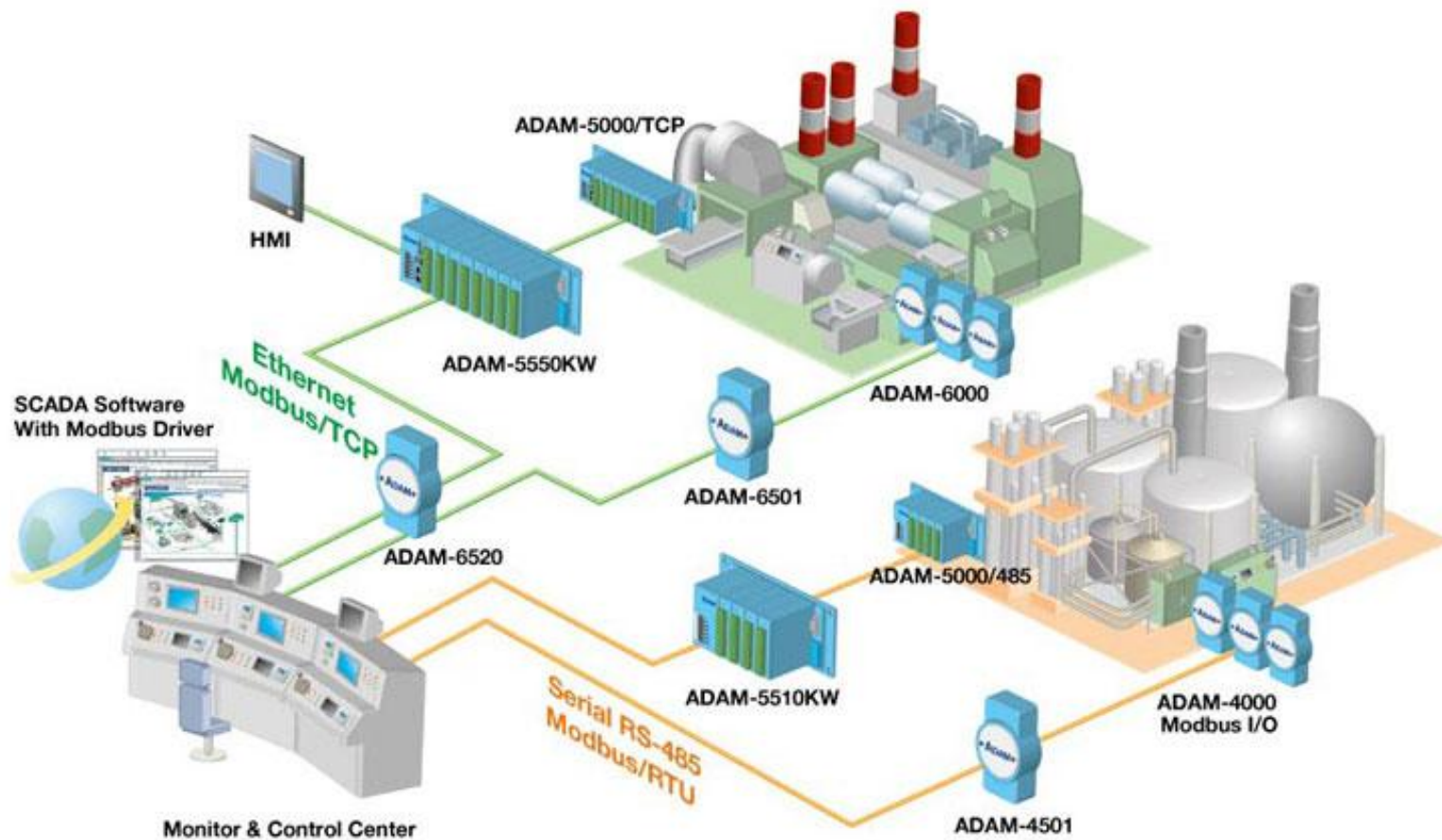


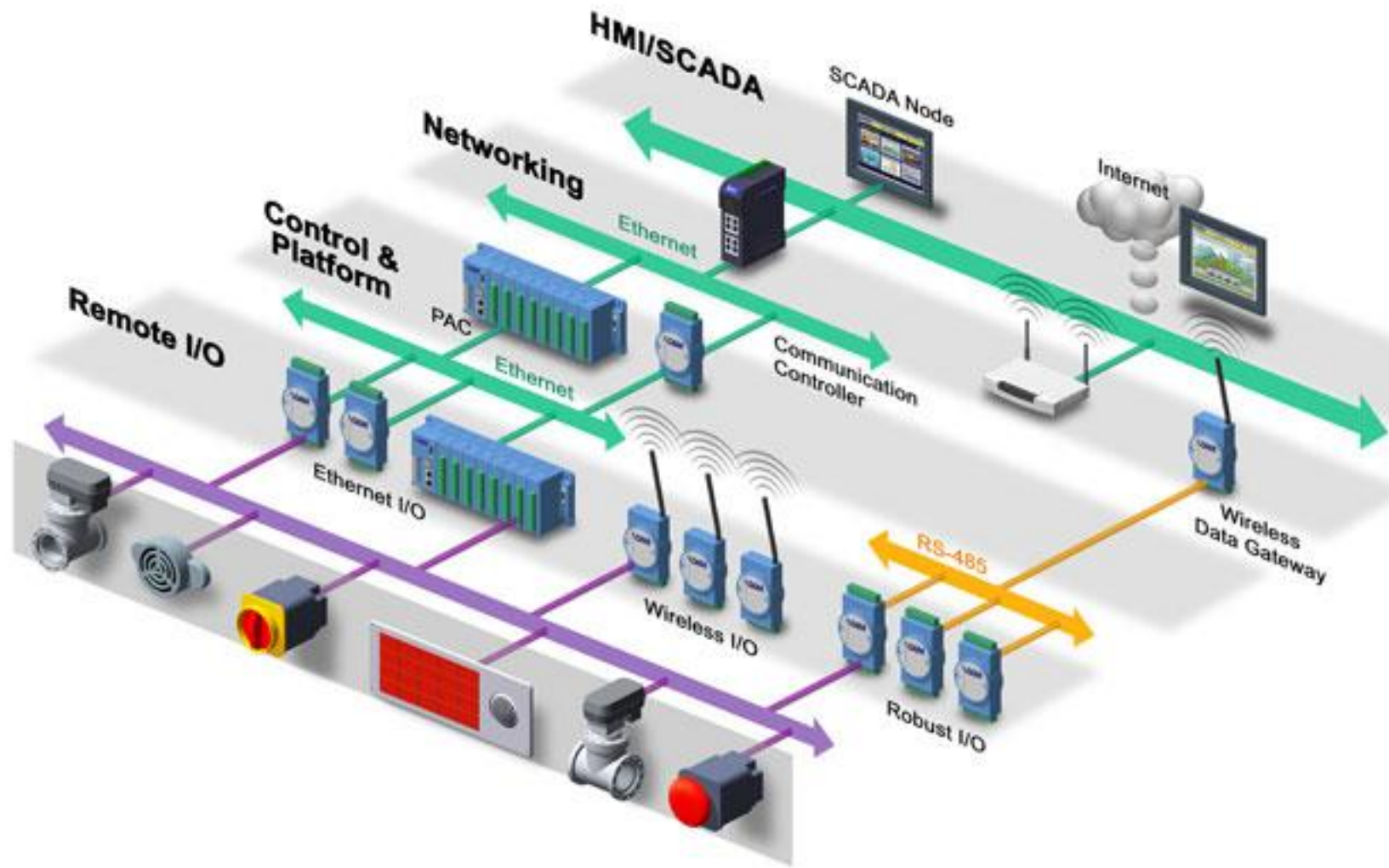
ADAM-4000 Series
Remote I/O Modules



ADAM-6000 Series
Smart Web I/O Modules







Comunicaciones con Equipos Inteligentes

- PLCs
- Indicadores Digitales
- Transmisores de Flujo
- Equipos Analíticos
- Computadoras Remotas
- Indicadores de Básculas
- Terminales de Datos
- Lectores de Código de Barras

Software

Para Adquisición de
Datos y Control

Software: Dos Alternativas Básicas

**De Paquete
vs.
A la Medida**

Software de Paquete vs. Software a la Medida

A LA MEDIDA

- ↓ Programación en lenguajes especializados
- ↓ Se requiere de programadores experimentados
- ↓ Ciclos de desarrollo prolongados
- ↑ Programas que operan exactamente como se desea

DE PAQUETE

- ↑ "Programación" simplificada
- ↑ "Programación" por personal no especializado
- ↑ Desarrollo acelerado
- ↓ Posibles limitaciones en la operación

Software de Paquete vs. Software a la Medida

A LA MEDIDA

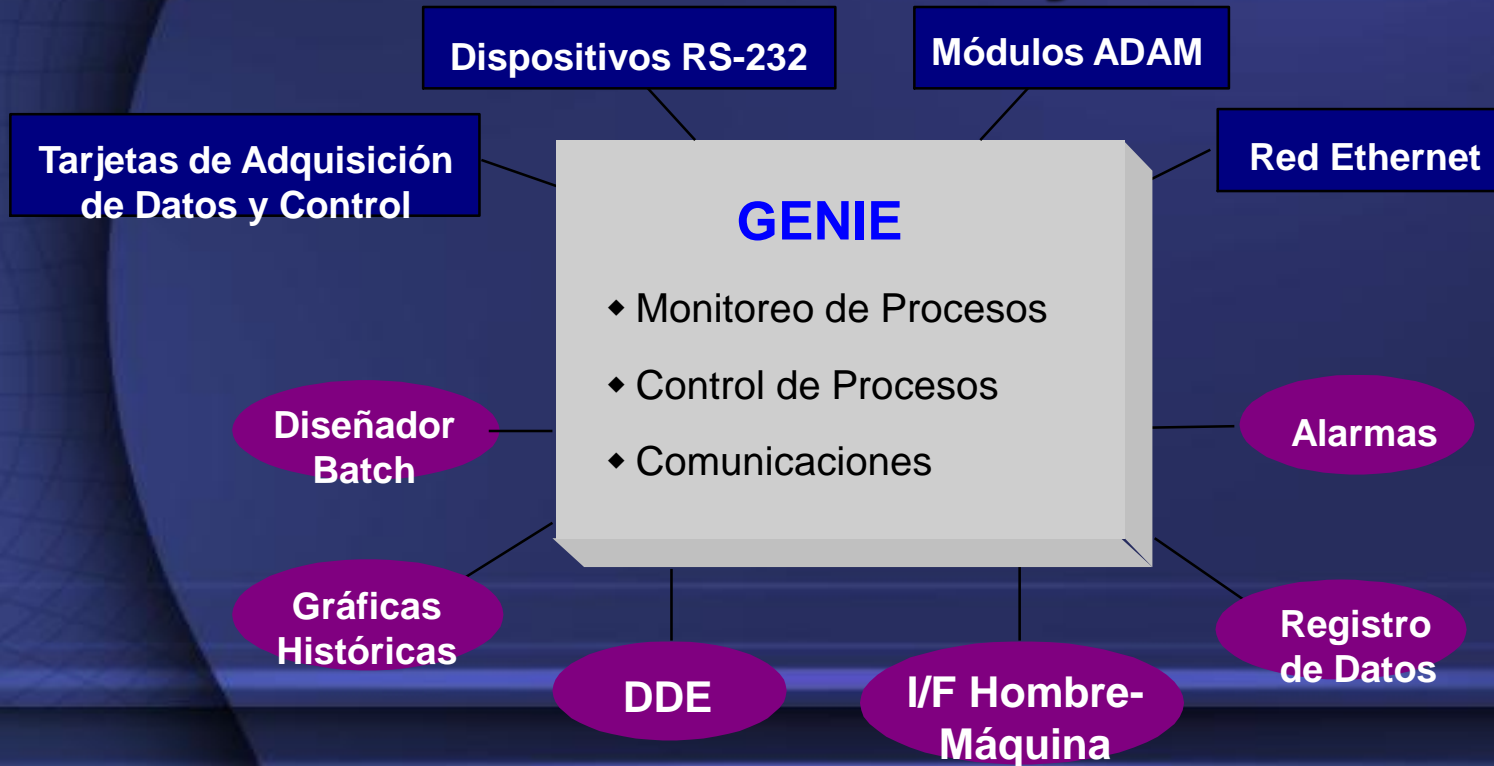
- ↓ Dependencia del desarrollador
- ↓ Actualizaciones costosas
- ↓ Costo proporcional al tamaño del sistema

DE PAQUETE

- ↑ Dependencia del soporte del fabricante
- ↑ Actualizaciones económicas
- ↑ Costo fijo

GENIE 3.2

Software para Adquisición de Datos y Control



Genie

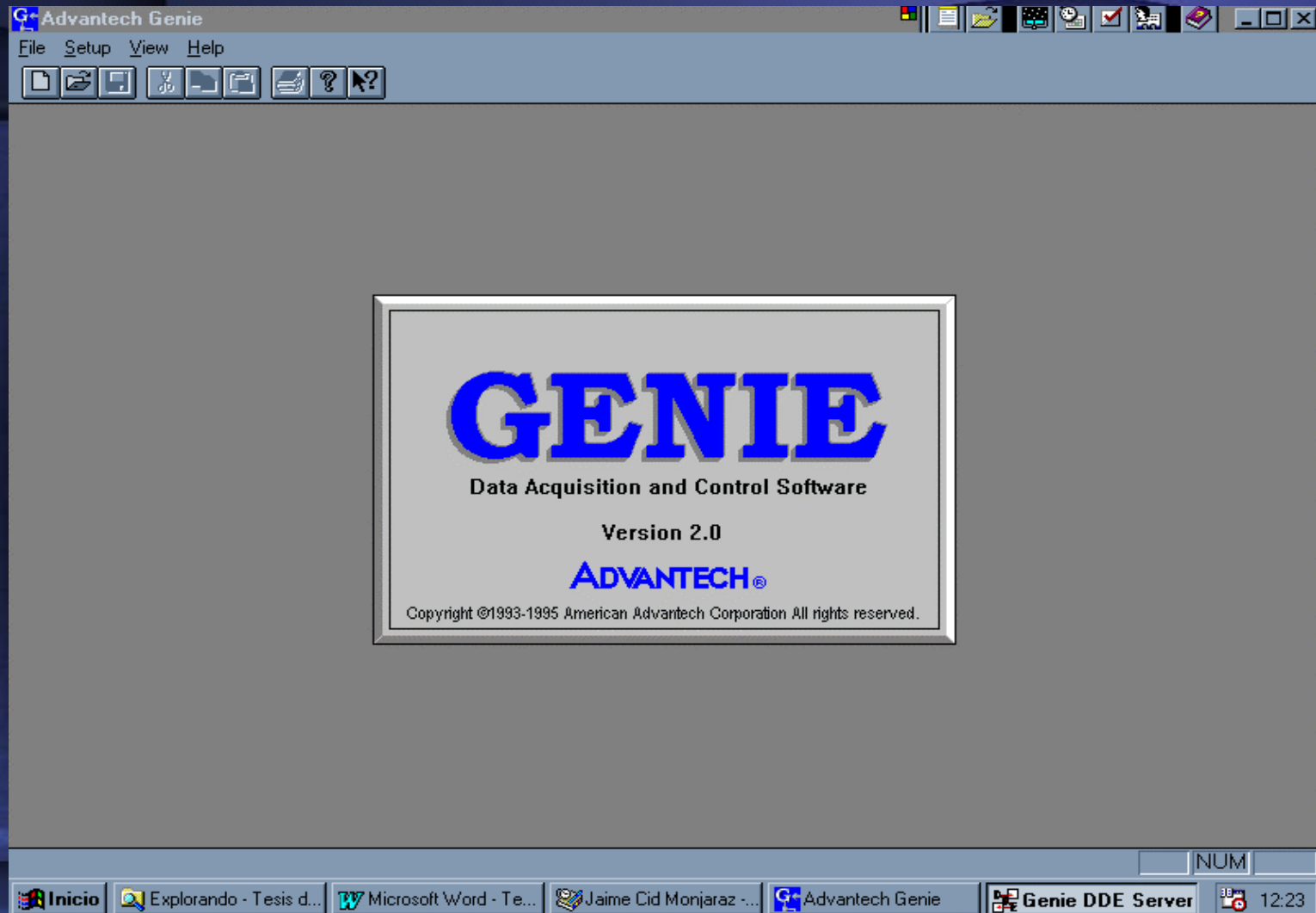
Características

- Soporta todo el Hardware de Advantech
- Almacenamiento de Datos en Archivos
- Despliegue de Pantallas con Múltiples Herramientas Visuales
- Registro y Reconocimiento de Alarmas/Eventos

Genie

Características

- Control ON/OFF y PID
- Operaciones de Cálculo
- Comunicación vía Red Ethernet
- Despliegue de Pantallas múltiples
- Despliegue de Mímicos
- Niveles de Seguridad Múltiples



El software de monitoreo y

Caja de herramientas

íconos



El editor de estrategias es un programa basado en íconos que trabaja bajo ambiente windows. Tiene asociada una caja de herramientas representativas de los posibles bloques de construcción de estrategias de automatización.

Editor de Estrategías

Advantech Genie - [Strategy Designer: TASK1]

File Edit Setup View Window Run Help

SOC12 NUM

Advantech Genie - [Strategy Designer: TASK1]

File Edit Setup View Window Run Help

Caja de herramientas

Ejemplo de estrategia

The screenshot displays the Advantech Genie Strategy Designer interface. At the top, the window title is "Advantech Genie - [Strategy Designer: TASK1]" and the menu bar includes "File", "Edit", "Setup", "View", "Window", "Run", and "Help". A toolbar with various icons is located below the menu bar. On the left side, a vertical "Caja de herramientas" (Toolbox) contains numerous control blocks such as "user prog", "FAI", "AI", "AO", "DI", "DO", "PID", "Run Avg", "DDE", "SOW", and others. The main workspace shows a central "DISP-1" controller block connected to several input blocks labeled "AI1" and "AI2", and two output blocks labeled "user prog PRG1" and "user prog PRG2". A red arrow points from the "Caja de herramientas" label to the toolbox, and another red arrow points from the "Ejemplo de estrategia" label to the central controller block. The status bar at the bottom left says "For Help, press F1" and the bottom right has a "NUM" indicator.

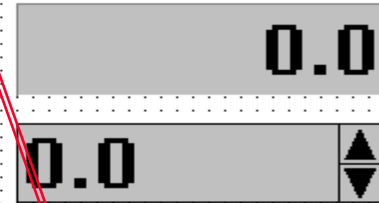
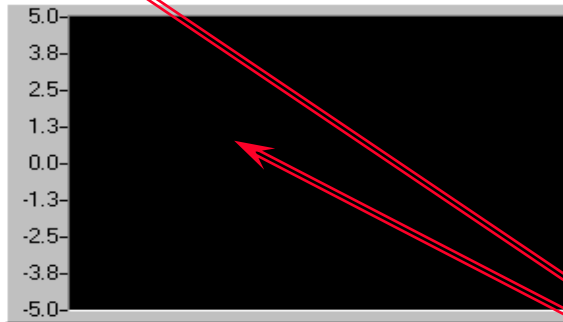
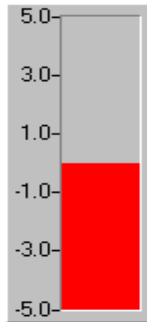
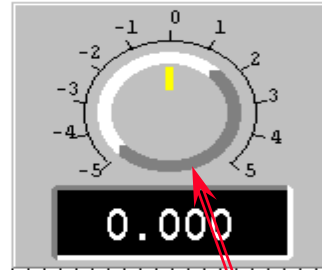
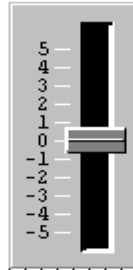
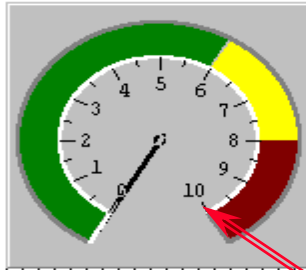
El software de monitoreo y control GENIE

Caja de herramientas

editor de pantallas

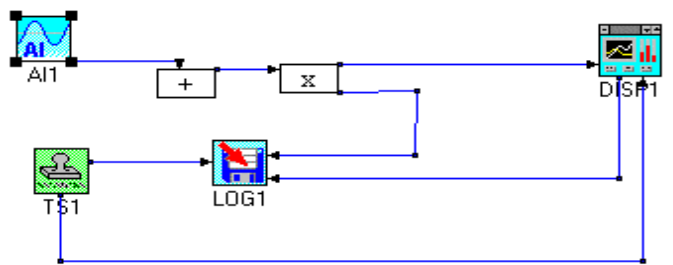
El editor de pantallas es un subprograma basado en íconos que trabaja bajo ambiente windows. Tiene asociada una caja de herramientas representativas de los posibles bloques de pantallas que se pueden utilizar para crear los paneles





Caja de herramientas

Ejemplos de pantallas



Analog Input Block

Tag: Description:

Device:

Channel:

Input Range:

Exp. Channel:

Board ID:

Establish DDE Link

Update Rate:

OK Cancel Help Scaling

Advantech Genie - COLOTLA2.GNI

File Edit Setup View Window Run Help

Strategy Designer: TASK1

Digital Input Block

Tag: Description:

Device:

Group (Byte):

Bit (s):

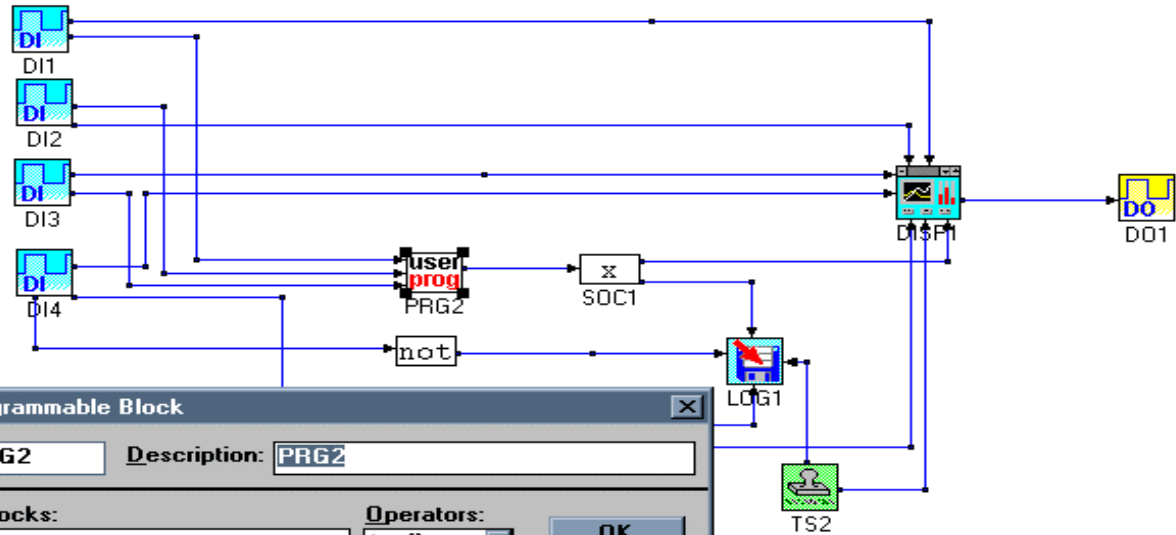
0
1
2
3
4
5
6
7

Establish DDE Link Update Rate:

DI4:DI4

NUM

Inicio Explorando - Tesis d... Microsoft Word - Te... Jaime Cid Monjaraz ... Advantech Geni... Genie DDE Server 12:53



User Programmable Block

Tag: PRG2 Description: PRG2

Input Blocks:
 DI1:DI1
 DI2:DI2
 DI3:DI3

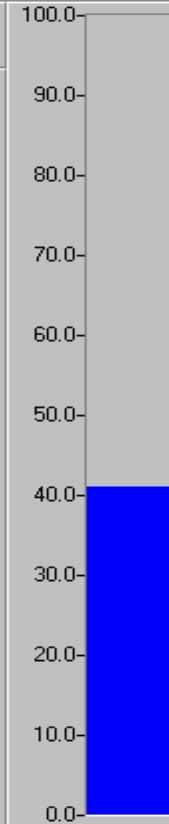
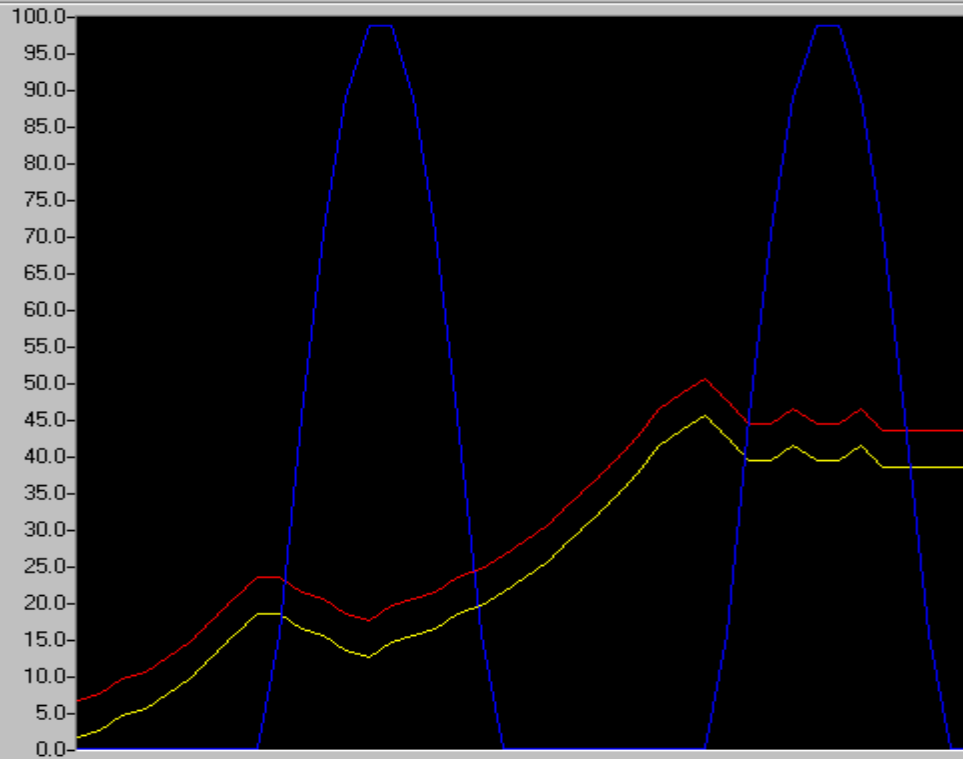
Operators:
 tan()
 rnd()
 abs()
 sqr()
 int()

Program:
`d = DI3*4 + DI2*2 + DI1;
 output d;`

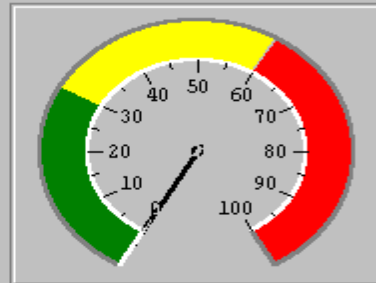
OK Cancel Help



MEDICION CON EL SENSOR LM35DZ



ESTADO DEL CONTROLADOR



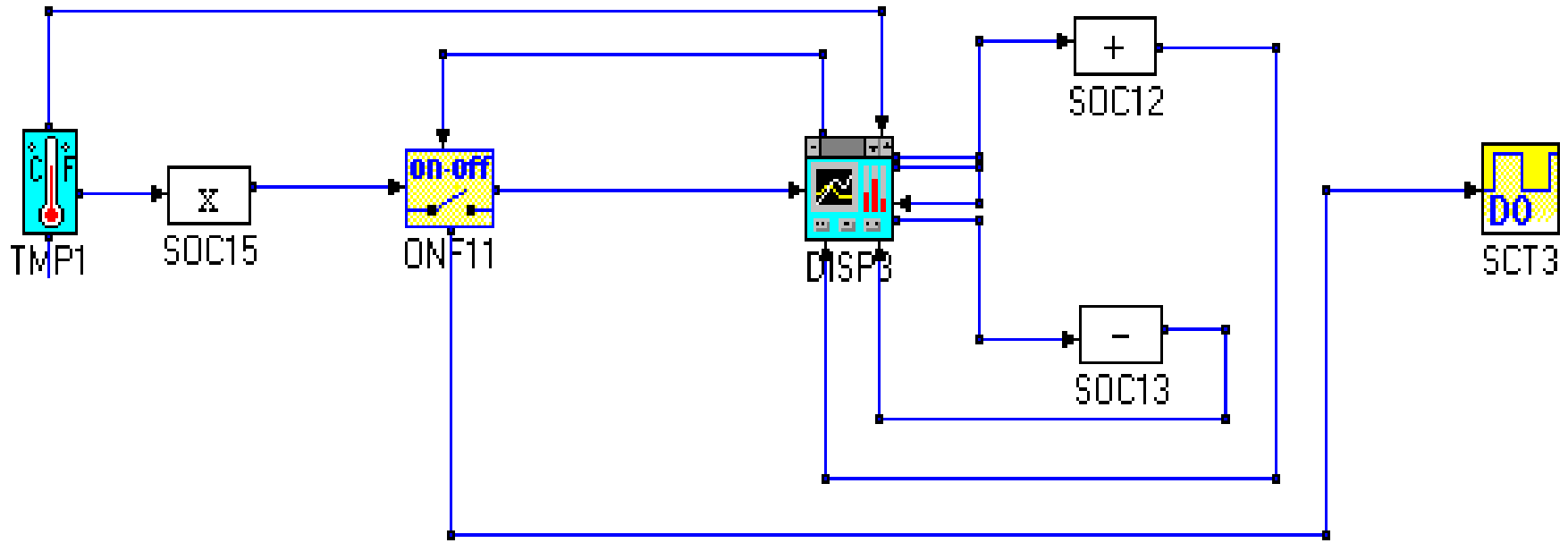
41.0

TEMPERATURA DESEADA (°C)

-45.4

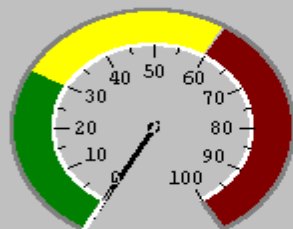
TEMPERATURA MEDIDA (°C)

Estrategia para el sensor de temperatura con termopar tipo J.

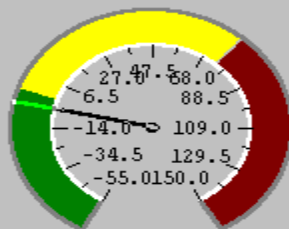




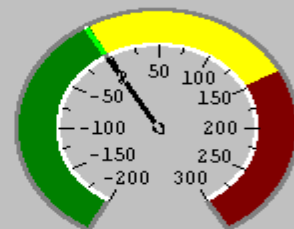
PANTALLA DE MONITOREO GENERAL



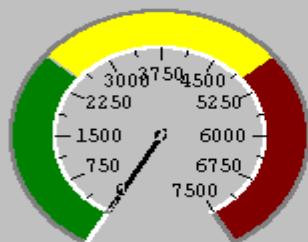
TEMPERATURA 1



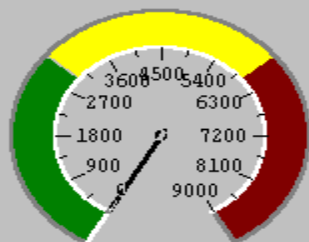
TEMPERATURA 2



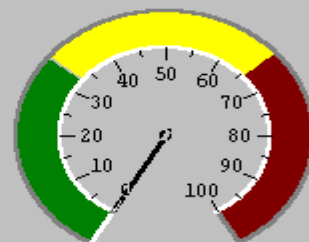
TEMPERATURA 3



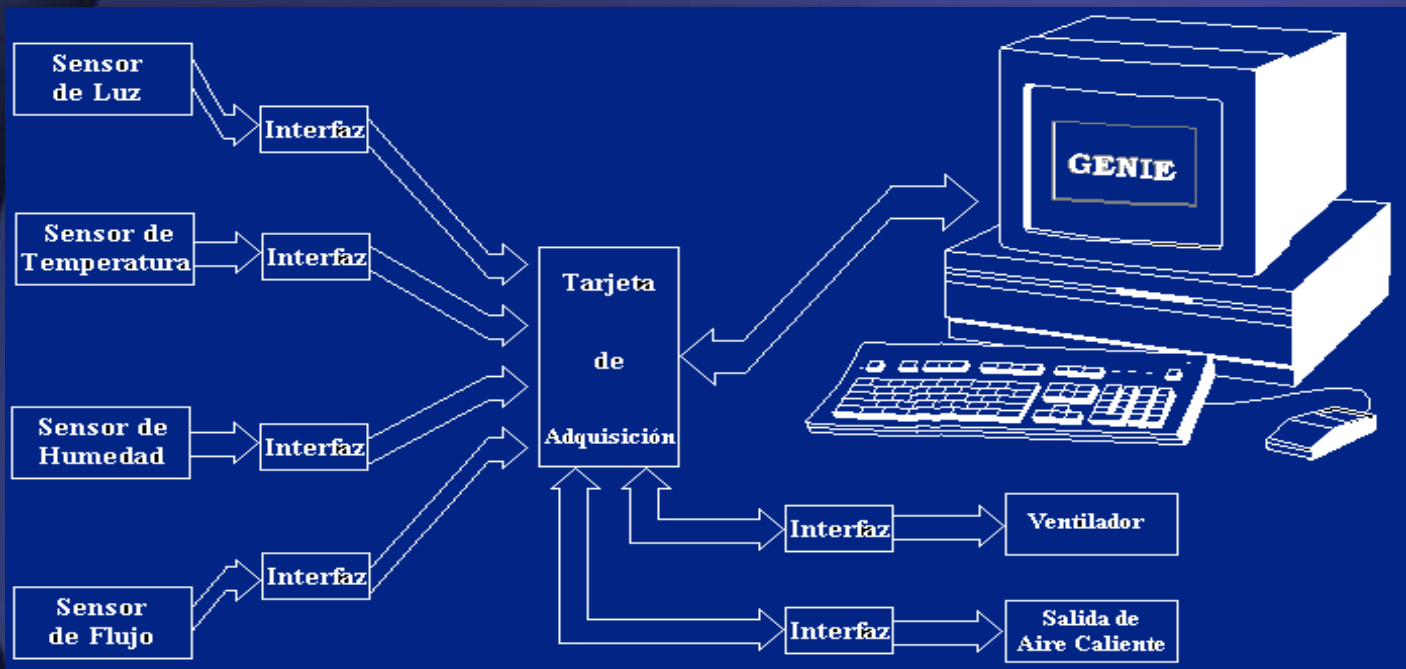
VELOCIDAD



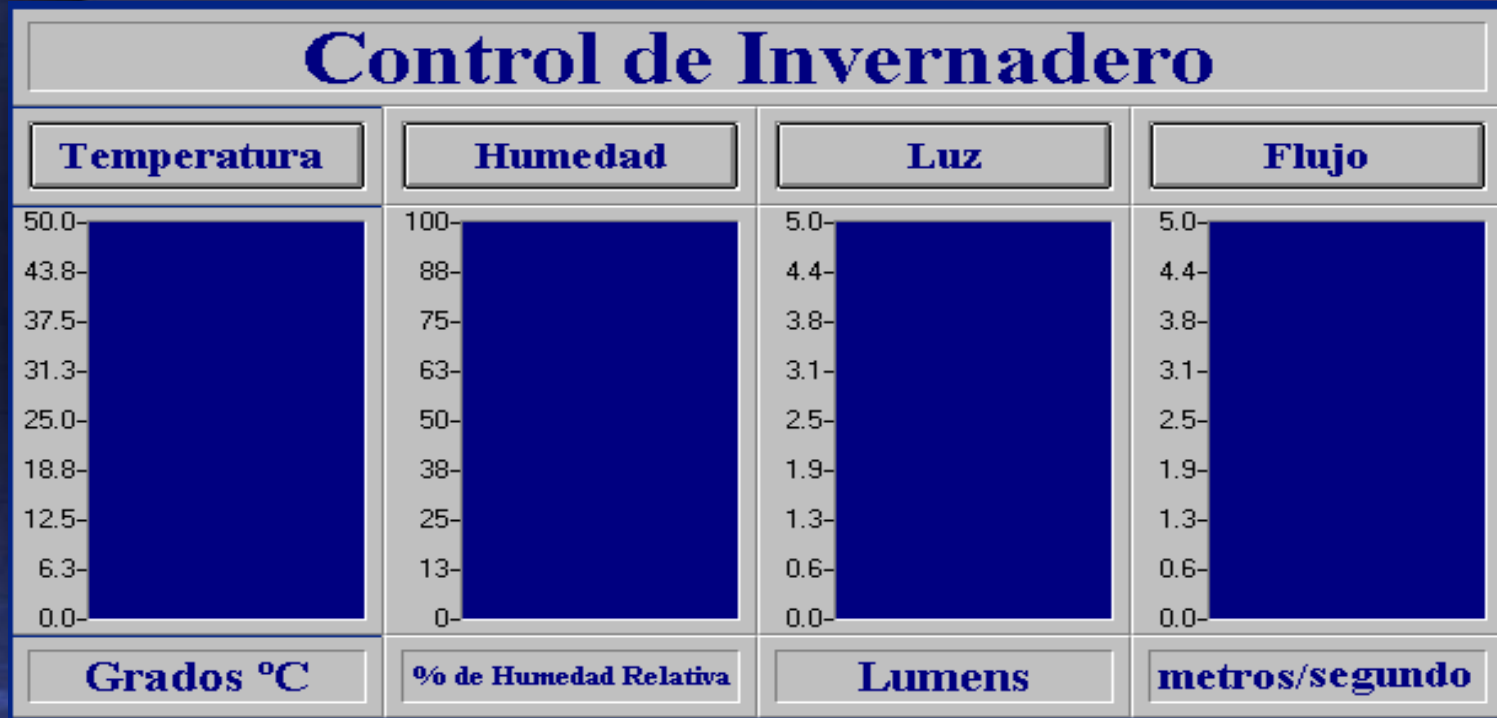
PRESION



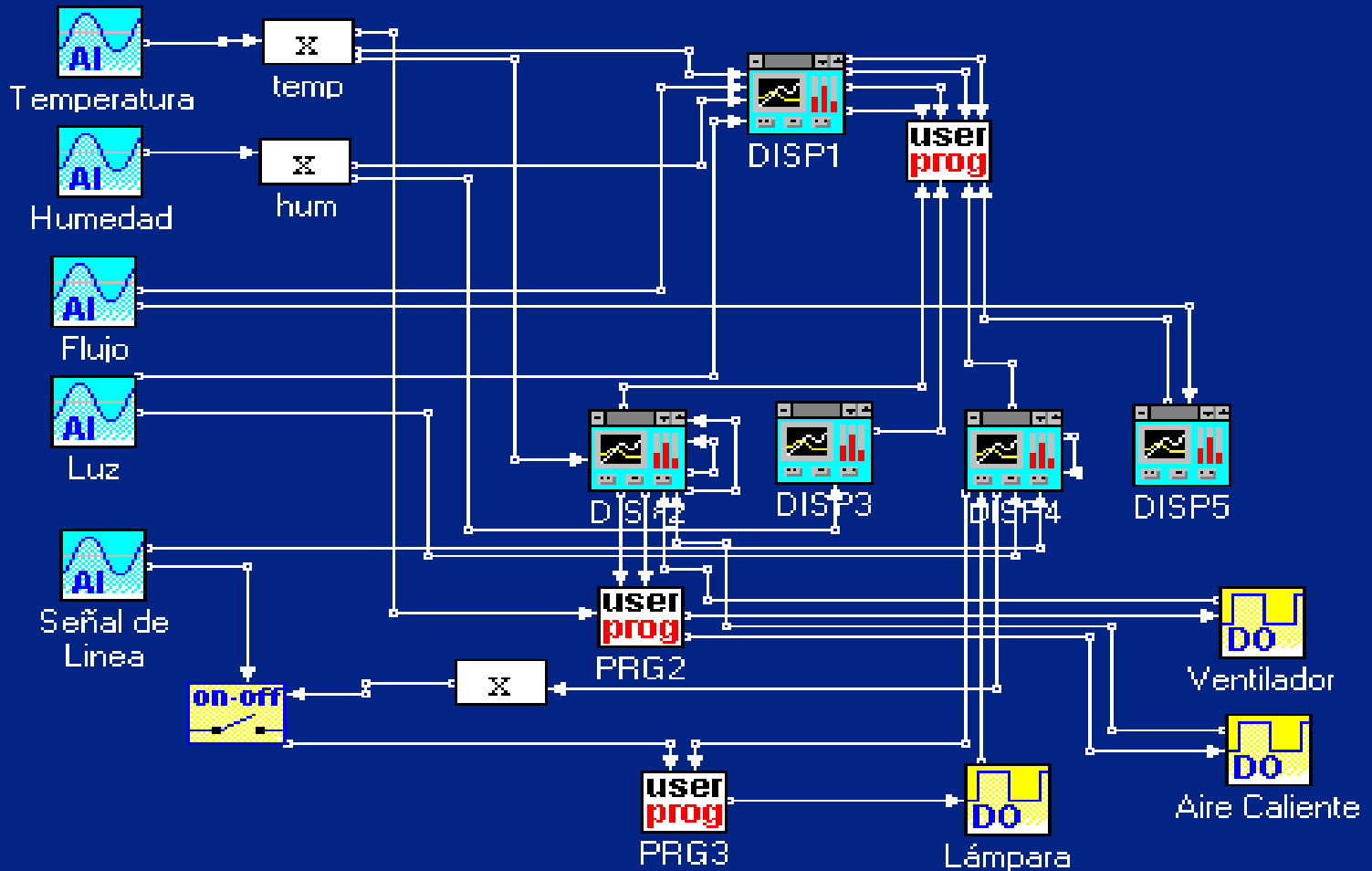
HUMEDAD



Estrategias De Control

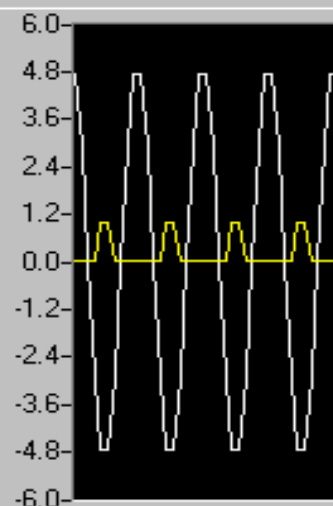
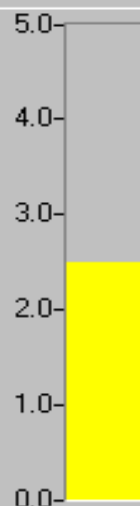
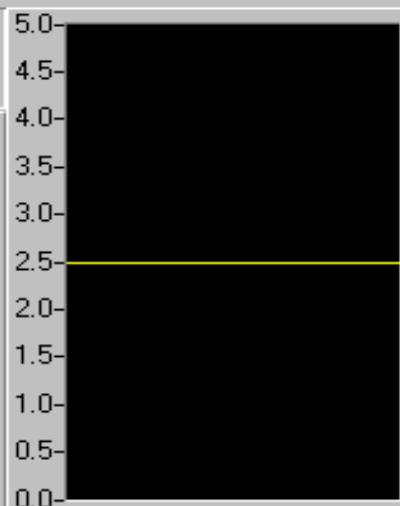
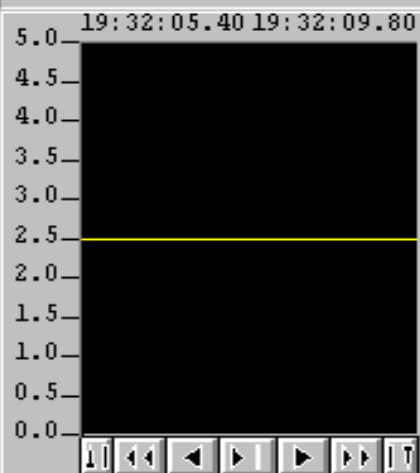


Pantalla Principal de Monitoreo



Control de intensidad de Luz

2.5



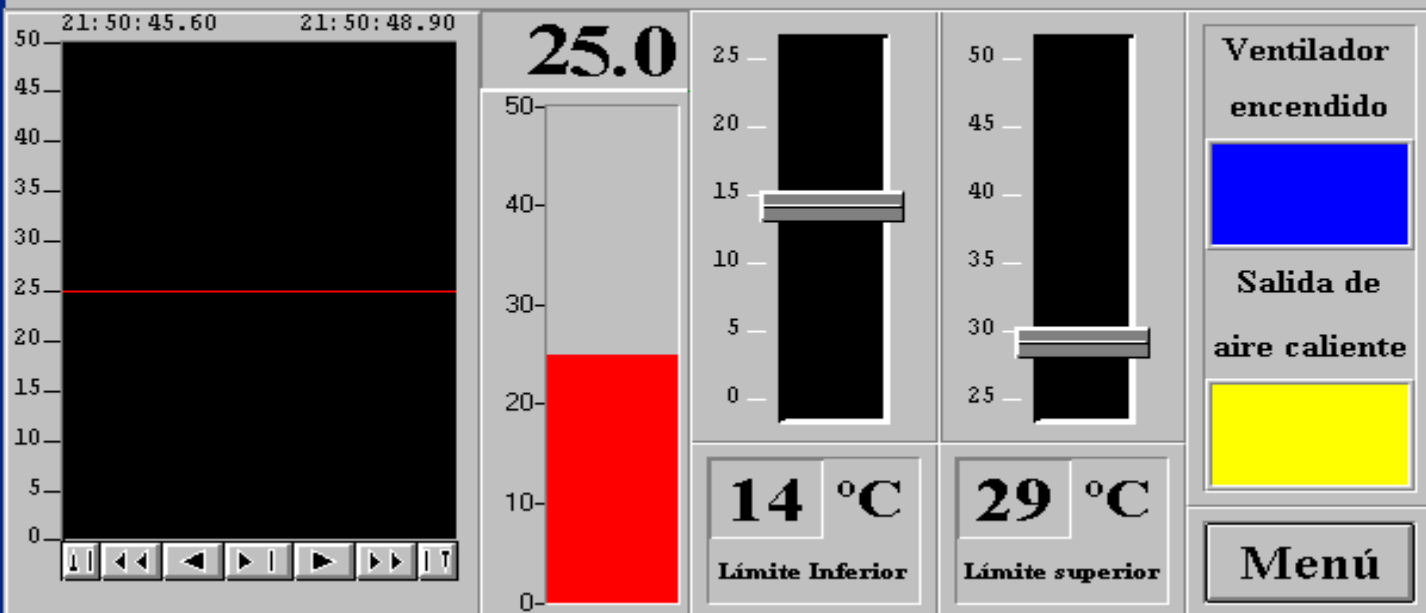
2.0

Encendido/Apagado Manual

Disparo

Menú

Temperatura °C



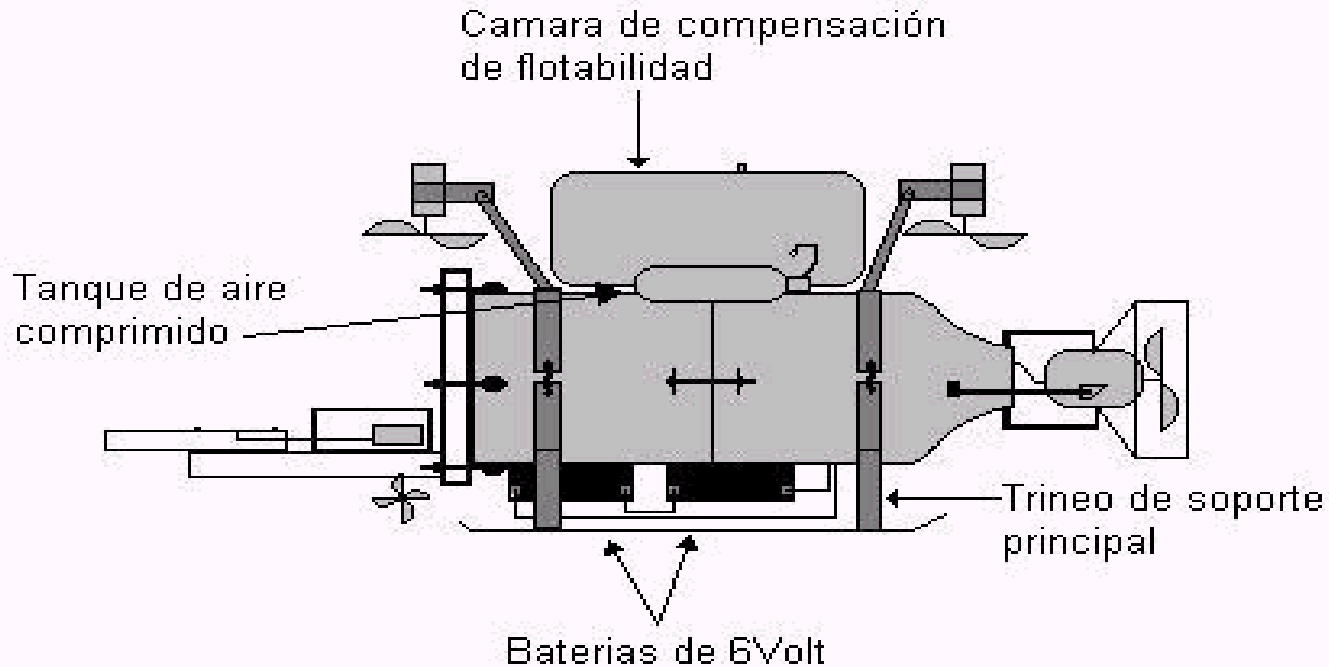


CUV_yTT-BUAP

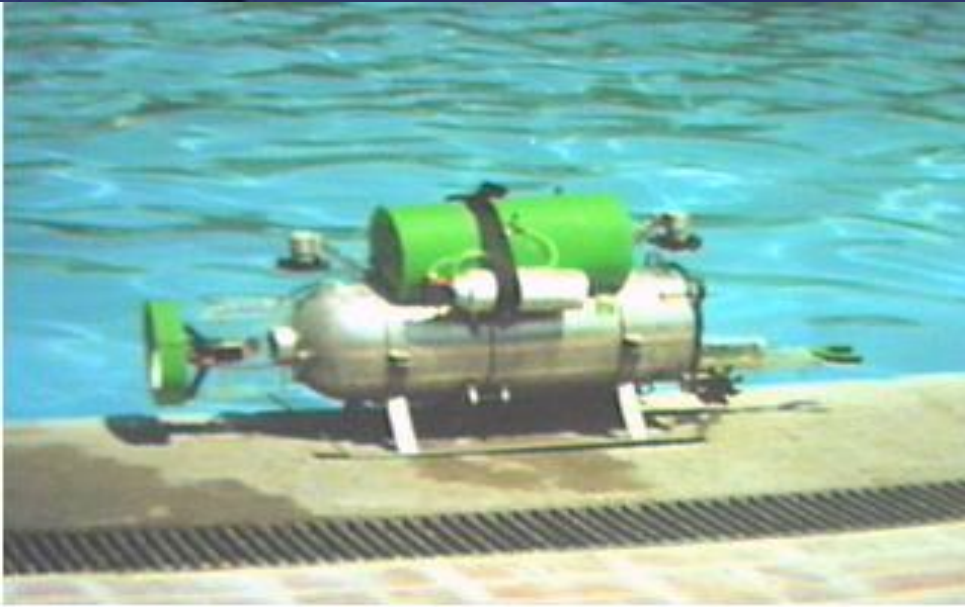


Construcción del Cuerpo Principal

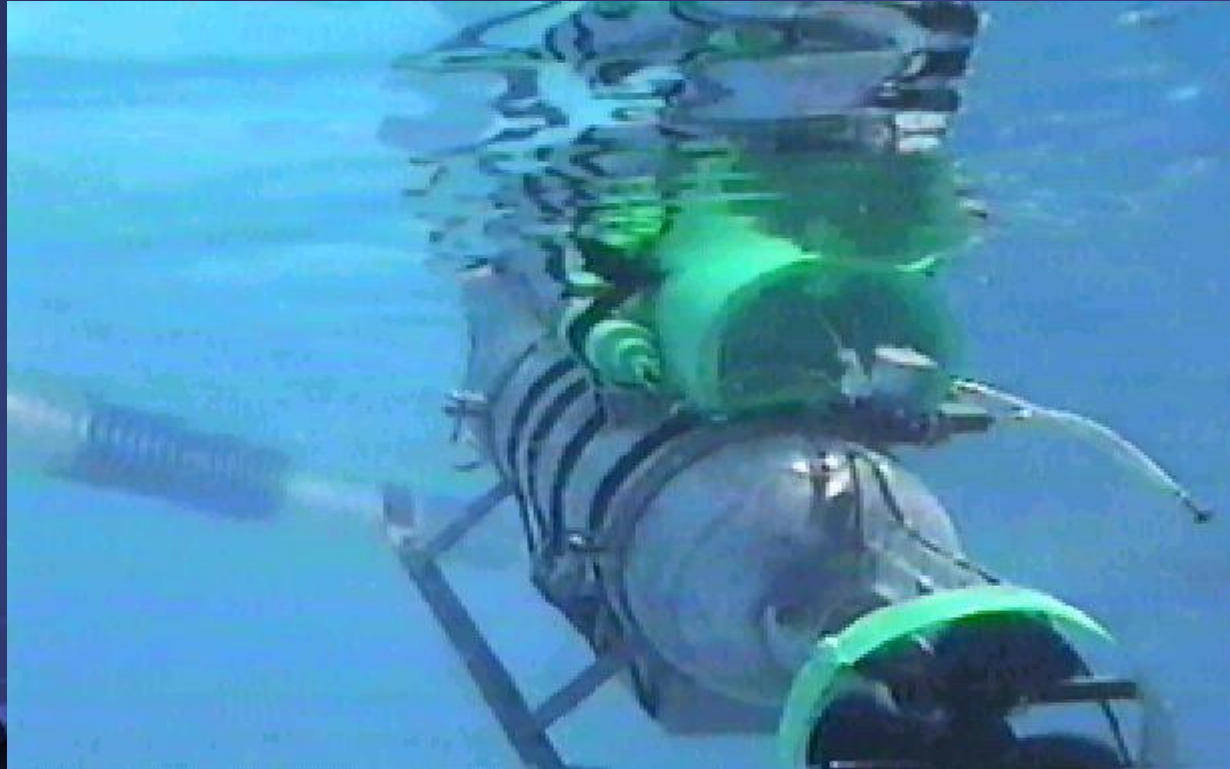
MONTAJE EXTERIOR



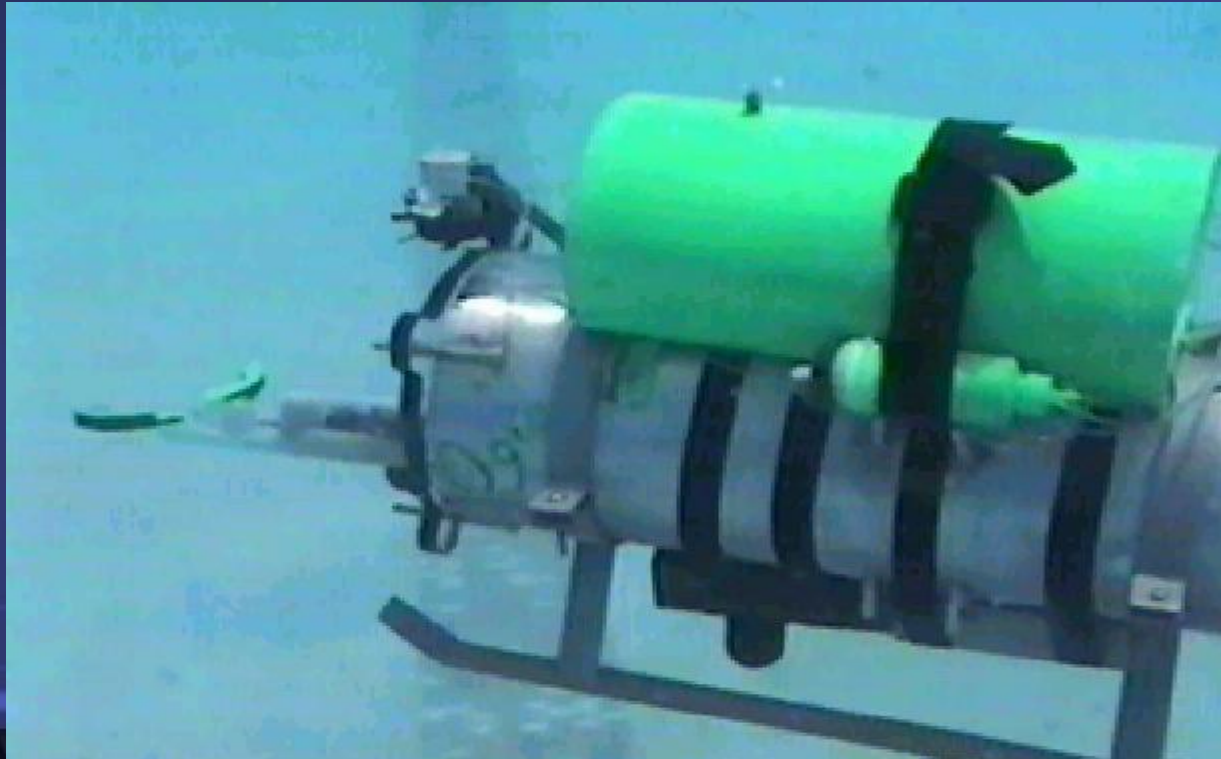
Pruebas en Campo



Inmersión del Mini-Submarino



Inmersión del Mini-Submarino



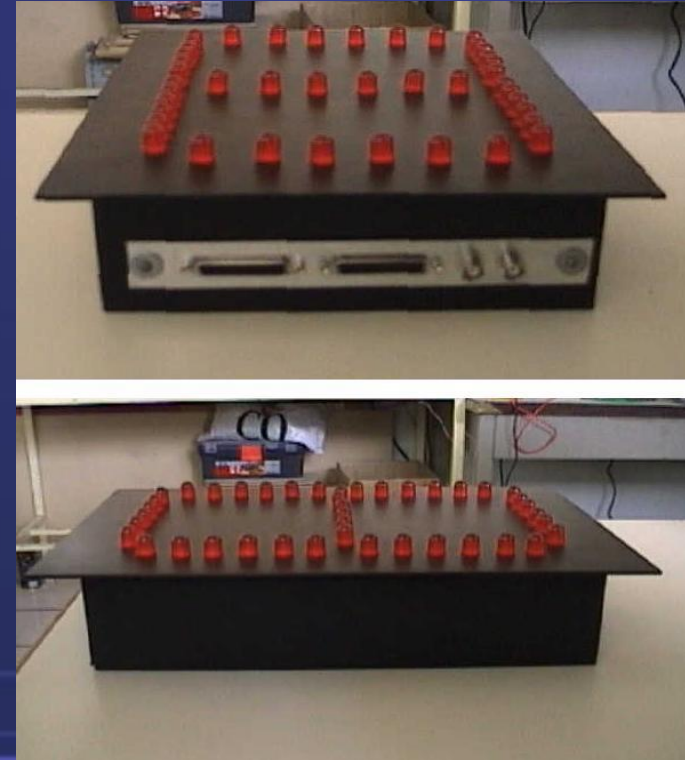
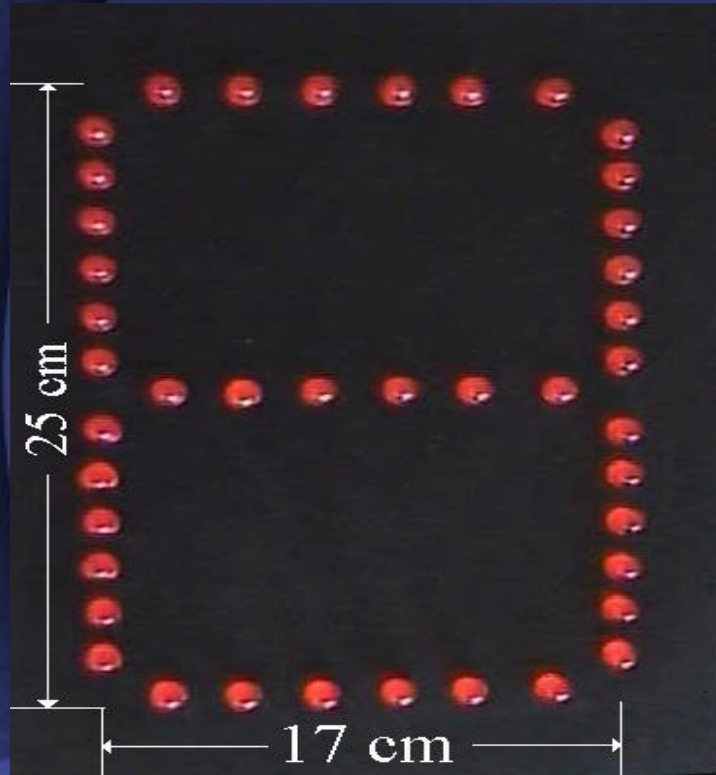
Inmersión del Mini-Submarino



Pruebas en Campo



Dimensiones del dígito



TIEMPO

13:30

LOCAL

VISITANTE

02

03

FALTAS

CUARTO

FALTAS

2

1

1

PANTALLA INTERACTIVA PARA EL USUARIO

Tablero_2_cards_letreros.vi

File Edit Operate Project Windows Help

VL

VL **Tablero Electrónico** **VL**

TIEMPO

Factor 1.00

00:17

Local **Visitante**

PUMAS **AMERICA**

15 **0**

Faltas L. Periodo **Faltas V.**

1 **2** **7**

Tiempo de duración (min)

15

Inicio

ON

Salir

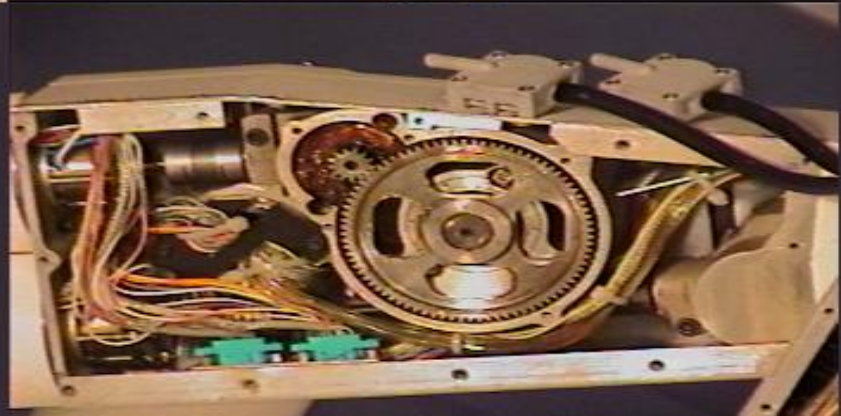
ON



Sistema PUMA

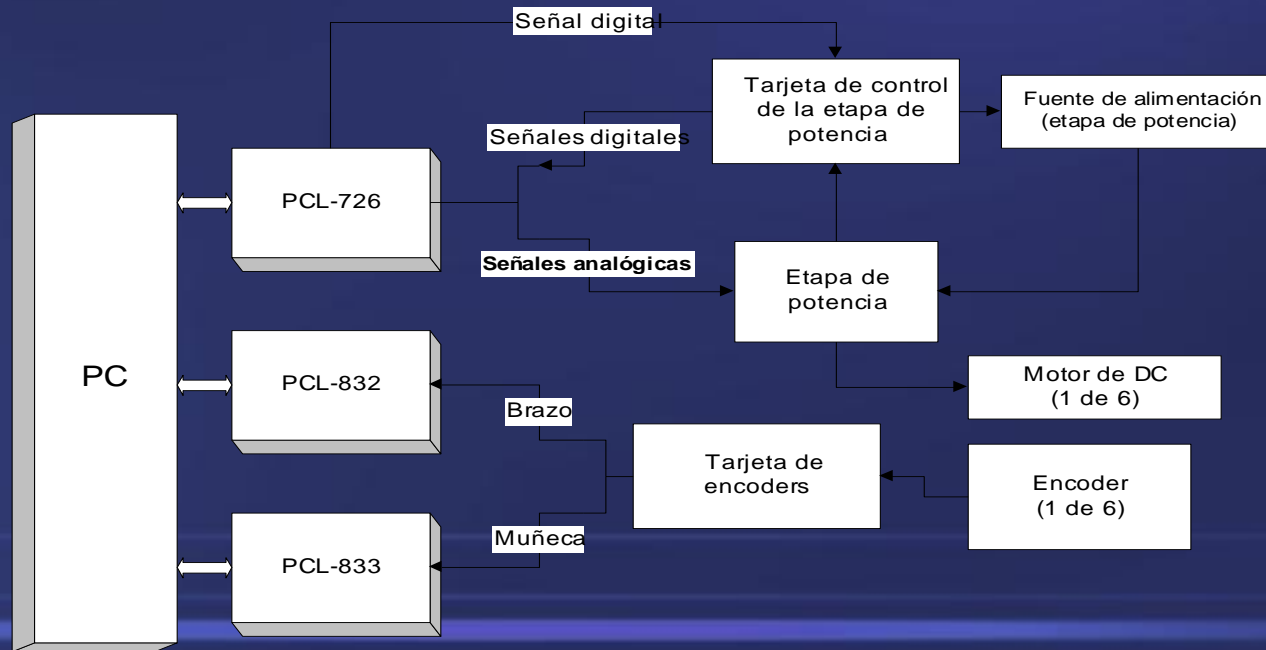
- Software
- Periféricos
- Control manual
- Brazo robot
- Controlador







Esquema De Conexión PC - PUMA 200





Proyecto PUMA200

PUMA 200 - Ayuda

Encendido

Posición de casa

Grados

Eje 1:	<input type="text" value="0"/>	Eje 4:	<input type="text" value="0"/>
Eje 2:	<input type="text" value="0"/>	Eje 5:	<input type="text" value="0"/>
Eje 3:	<input type="text" value="0"/>	Eje 6:	<input type="text" value="0"/>

Brazo Enc. **Error de Potencia**
Límite **Temp. Alta**

Paro de emergencia

Tiempo de muestreo: ms.

Configuración

Número de pulsos:

Valor de contador

Eje 1:	<input type="text" value="0"/>	Eje 2:	<input type="text" value="0"/>	Eje 3:	<input type="text" value="0"/>
Eje 4:	<input type="text" value="0"/>	Eje 5:	<input type="text" value="0"/>	Eje 6:	<input type="text" value="0"/>

Brazo

Cintura:

Hombro:

Codo:

Muñeca

Cabeceo:

Balanceo:

Giro:



DISEÑO Y CONSTRUCCION DE UN BRAZO DE ROBOT

SISTEMA RDC II

MENU DE FUNCIONES



MOVIMIENTO



EJE 1



EJE 1



EJE 2



EJE 2



EJE 3



EJE 3



EJE 4



EJE 4



EJE 5



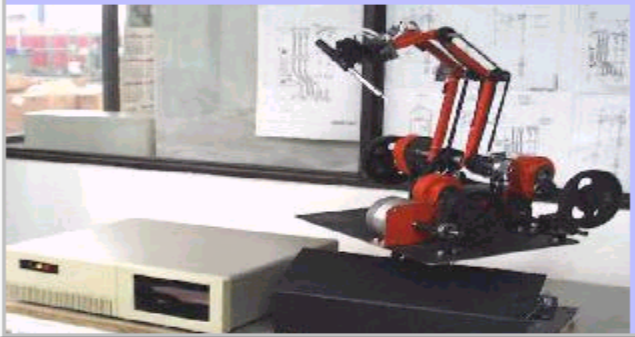
EJE 5



EJE 6



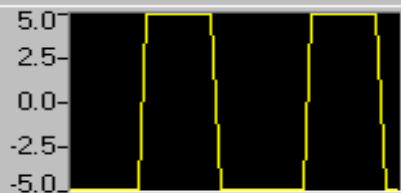
EJE 6





Display Designer: DISP3

UBICACIÓN DE TELESCOPIO EN COORDENADAS DESEADAS



Señal Encoder 1



Señal Encoder 2

ENC

A Principal

Coordenadas Deseadas

SECUENCIA DE MOTORES



polar



declinación

42

-30

Ubicando Coordenadas

26

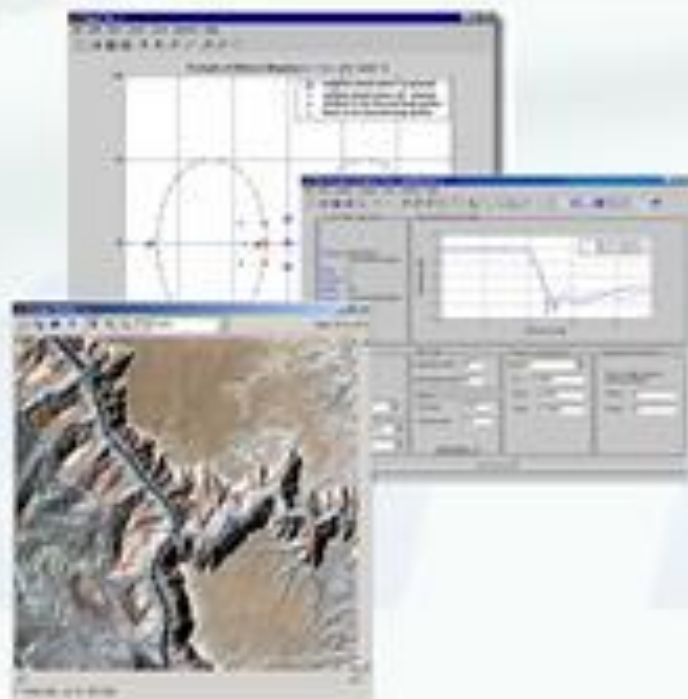
polar

-06

declina

Aplicaciones de MATLAB

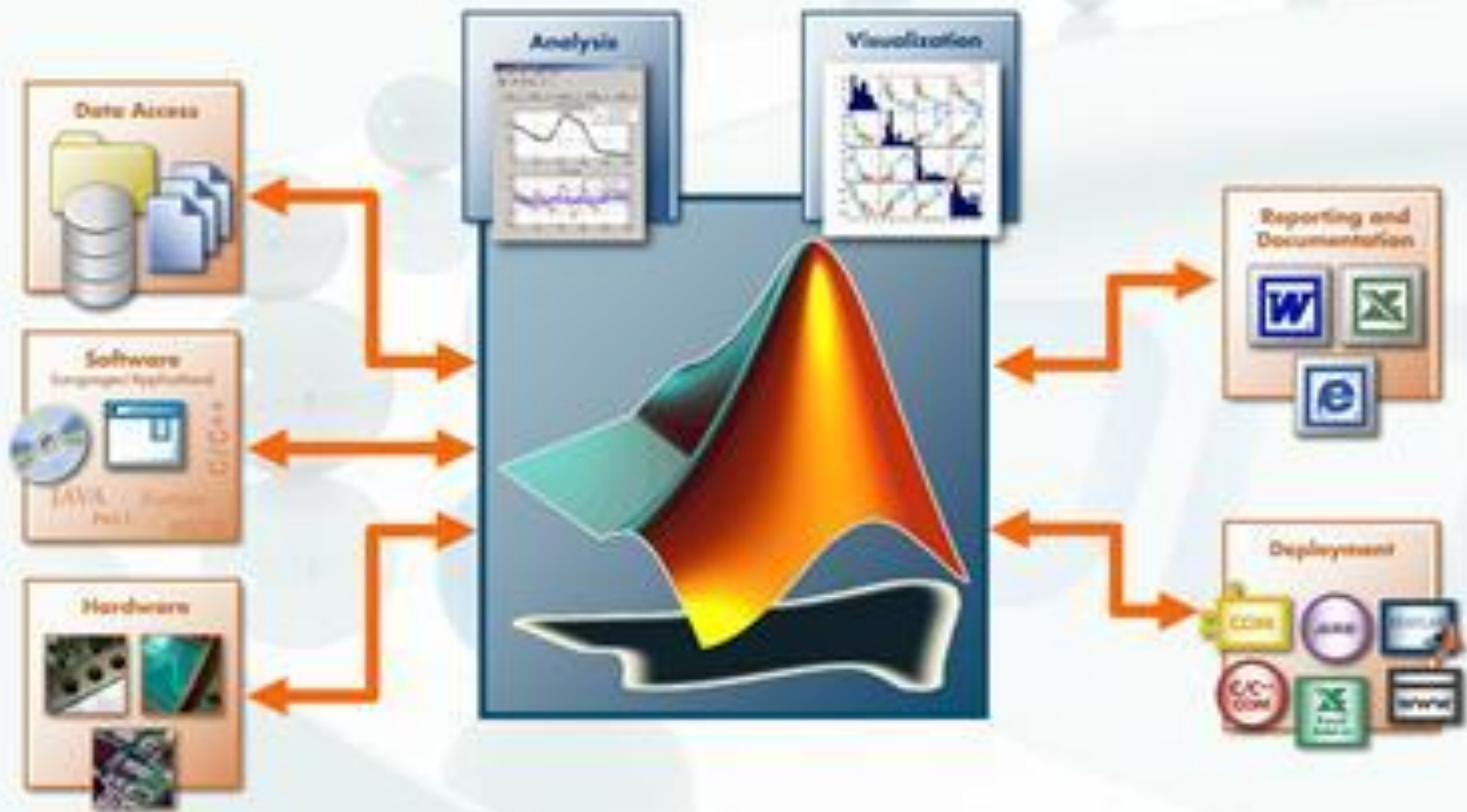
- Control
- Procesamiento de señales
- Procesamiento de imágenes
- Pruebas y Mediciones
- Finanzas
- Computo Técnico
 - Estadística
 - Optimización



Accesar Datos

Explorar

Compartir



Acceso a Datos

- Archivos
 - Excel, texto, or binario
 - Multimedia, científico
 - Web, XML
- Aplicaciones y lenguajes
 - COM, C/C++, Java, Fortran, DLL's
 - Bases de datos
- Instrumentos de medición
 - Hardware de adquisición de datos para señales o imágenes
 - Instrumentos individuales



Conectividad con Hardware

Data Acquisition Toolbox

Conectar tarjetas de adquisición de datos



Instrument Control Toolbox

Instrumentación electrónica y científica



Image Acquisition Toolbox

Instrumentos de captura de imágenes



MATLAB

Genéricos DLL

ActiveX/COM

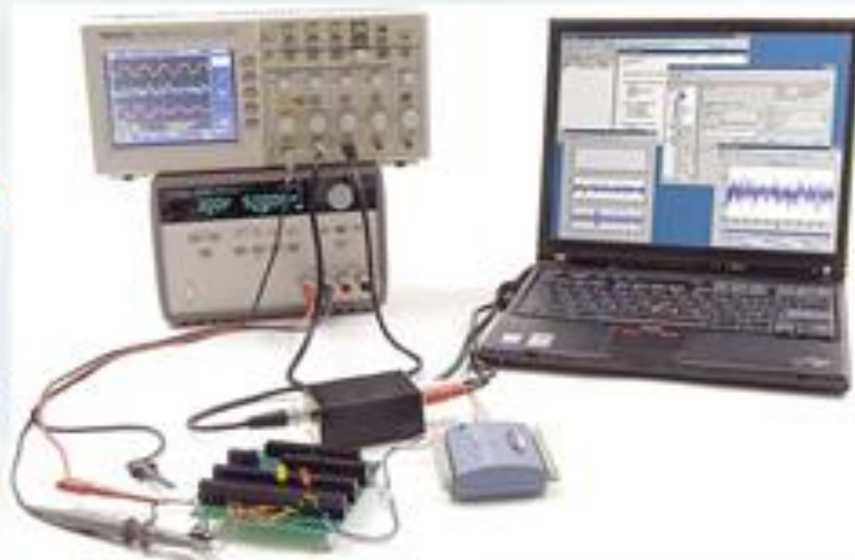
MEX

Interfaces para comunicarse con todo

Instrument Control Toolbox

Controle y comuníquese con instrumentos científicos

- Integra Instrumentos en aplicaciones de MATLAB y modelos de Simulink
- Herramientas interactivas para detectar y controlar instrumentos
- Generación automática de código para una fácil y rápida implementación
- Soporte para I/O, VXI *plug&play*, y controladores de instrumentos en MATLAB
- Soporte para protocolos de comunicación comunes



Adquisición de DATos

Retos – Mediciones Especializadas

Cantidad Física



Sensor/
Transductor



Hardware



MATLAB

- **Acelerómetros IEPE** (*vibración*)
- **Termocuplas, RTDs** (*temperatura*)
- **Sensores basados en puentes** (*galgas*)
- **Contadores/timers** (*eventos*)



Nuevas Características

R2011b

- **Data Acquisition Toolbox**
 - La nueva interface de MATLAB permite el acceso al hardware de NI directamente para este tipo de mediciones
- **Instrument Control Toolbox**
 - Soporte para comunicacion con Bluetooth desde MATLAB
 - Uso del perfil del puerto Serie.







Implementando en DSPs y FPGAs

MathWorks
Modeling Environment



Fixed Point Modeling

Generacion de Código

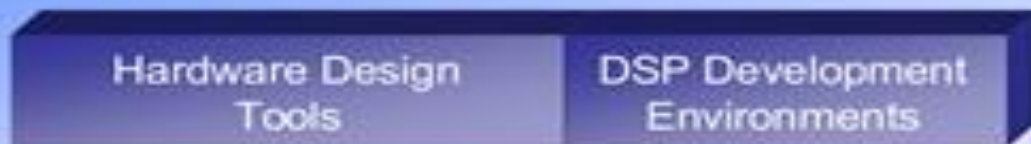


Verificación continua

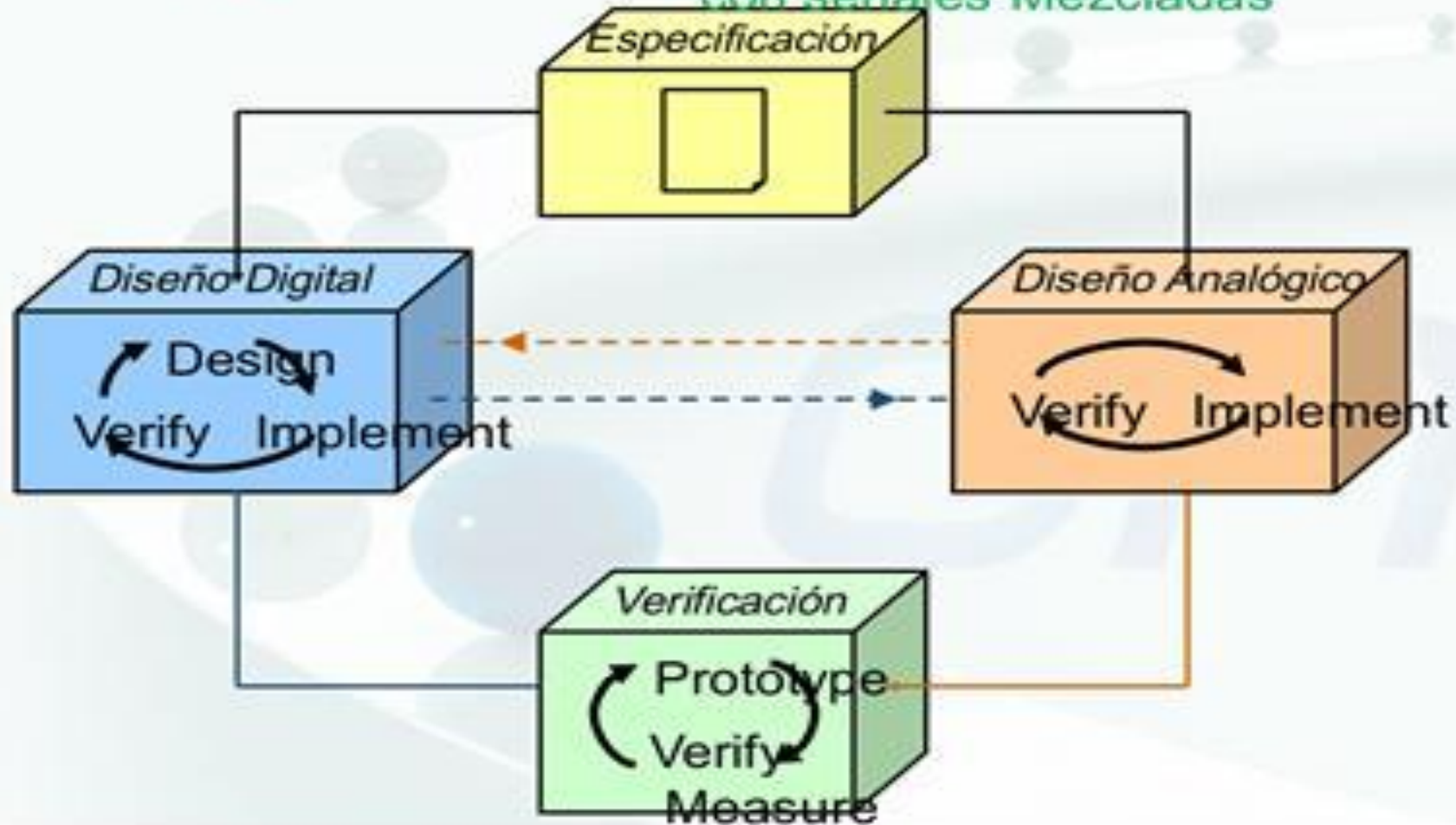


Embedded IDE Link™ and EDA Simulator Link™

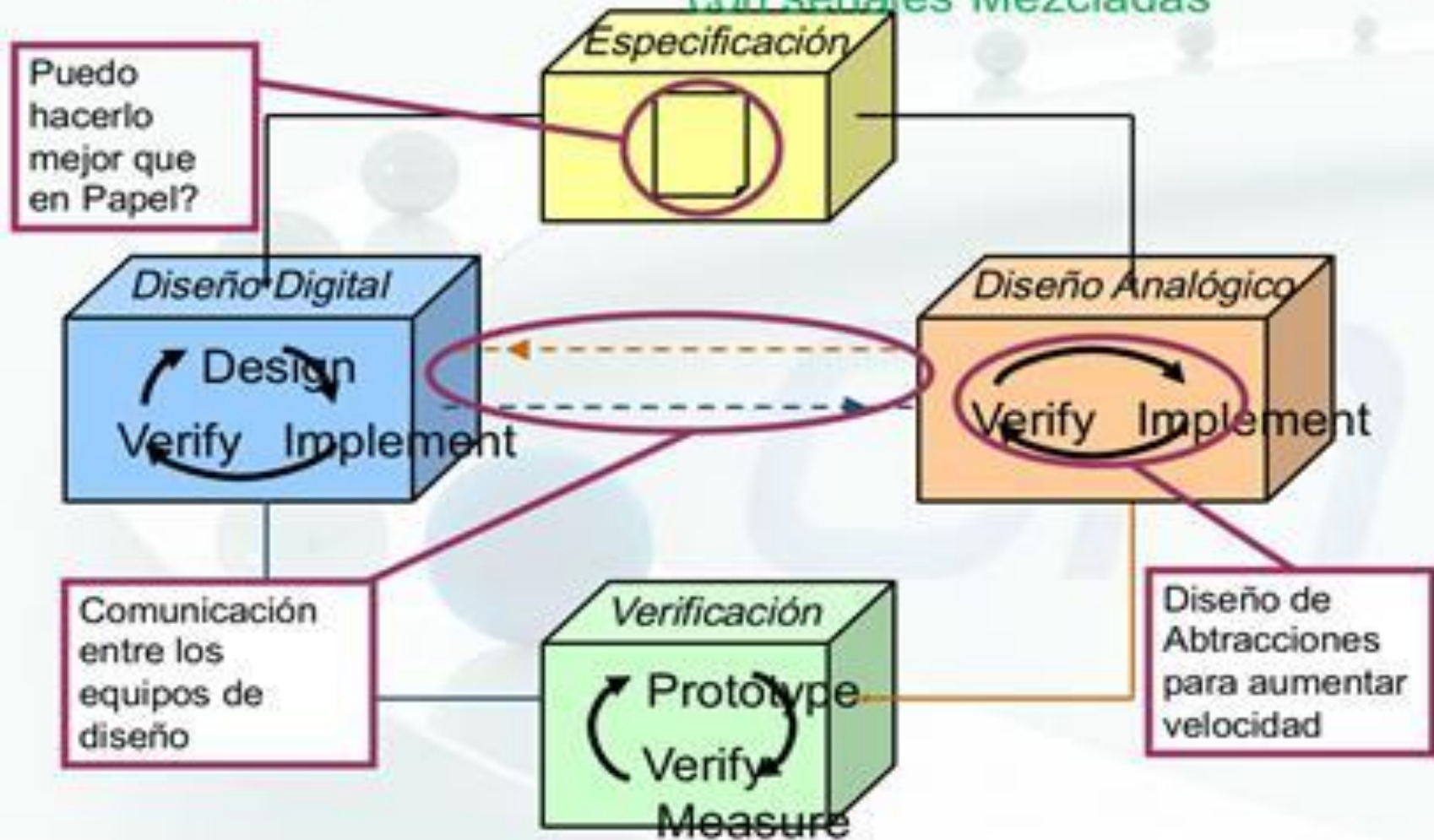
3rd Party Integrated Development Environments



Flujo de trabajo para diseño de sistemas con señales Mezcladas



Flujo de trabajo para diseño de sistemas con señales Mezcladas



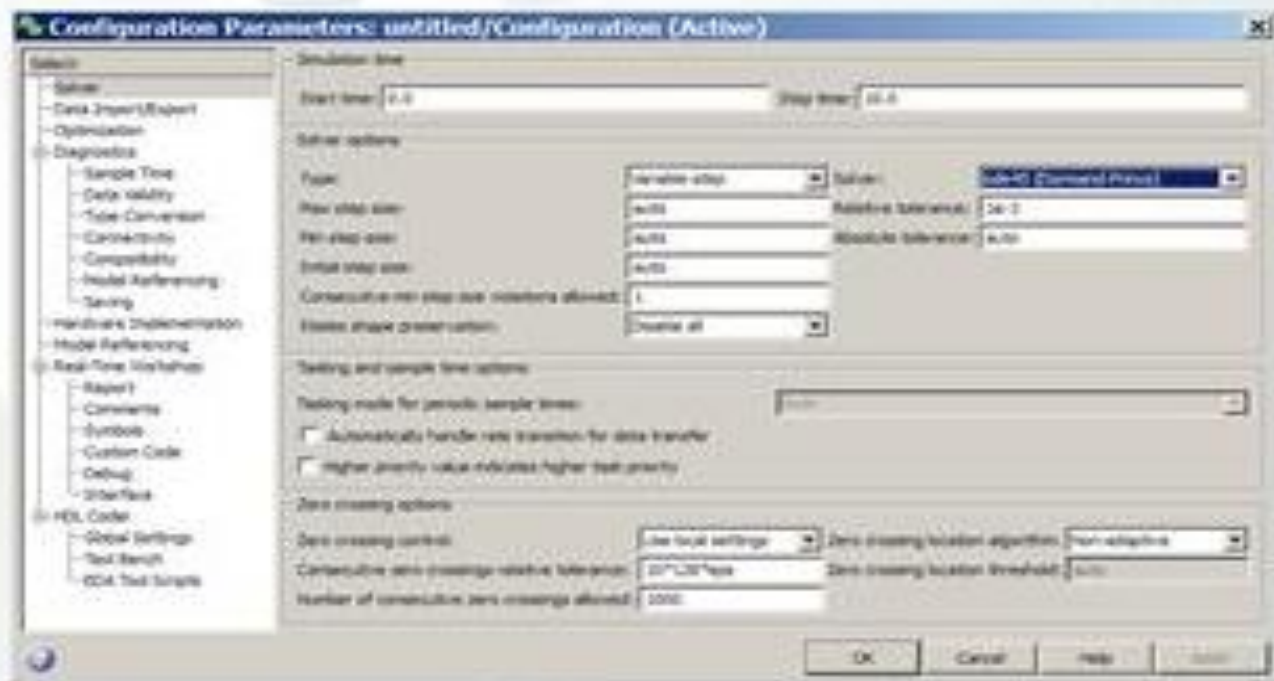
Diseño Rápido a través de la Abstracción

Tres Características de Simulink:

- Librerías de Abstracción
- La ayuda del paso variable (Lo analógico y lo digital en el mismo modelo)
- Simulación al nivel Requerido

ODE solver de paso Variable

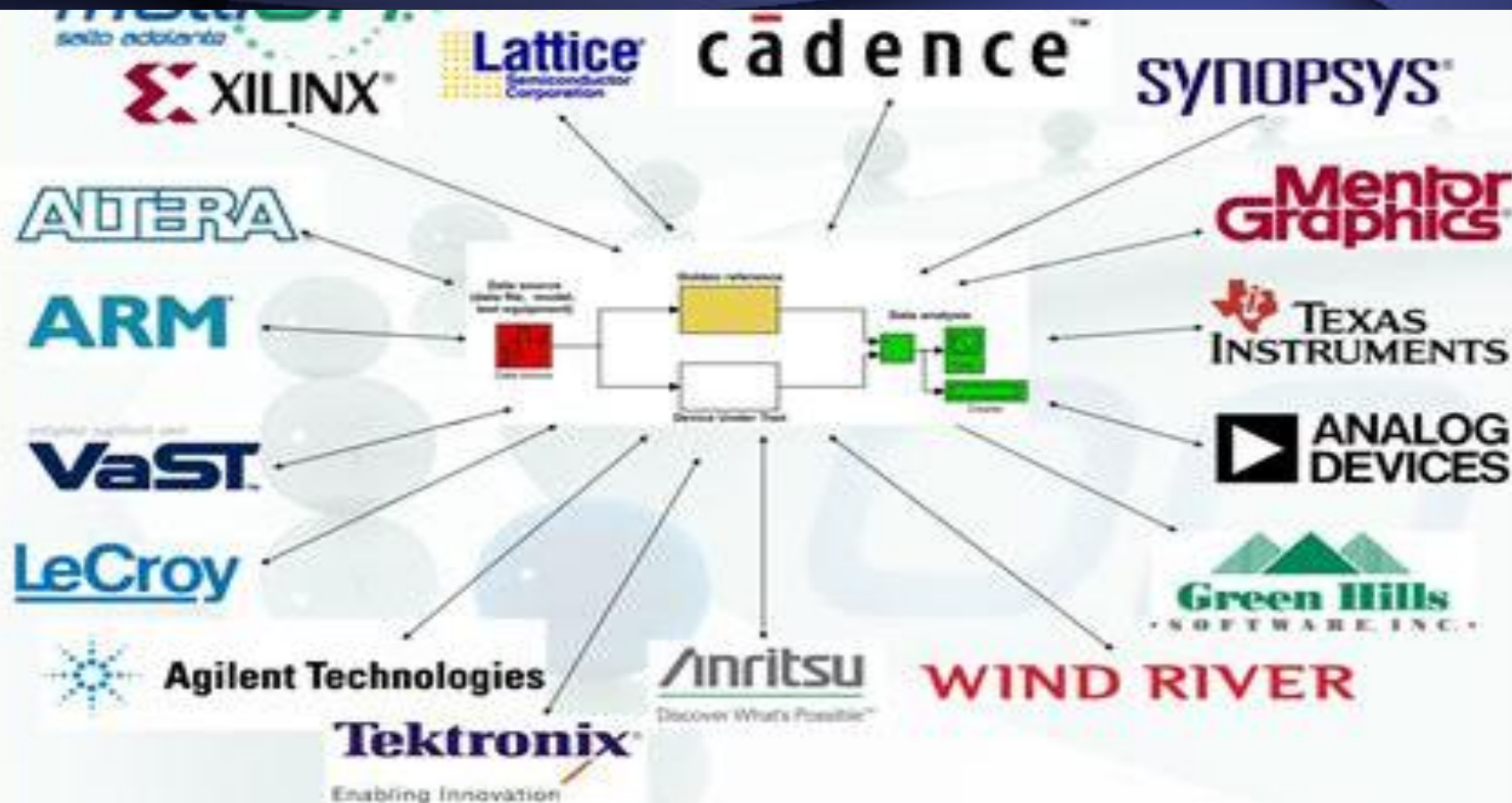
Un sofisticado tiempo de paso del solver es crucial para la simulación Analógica



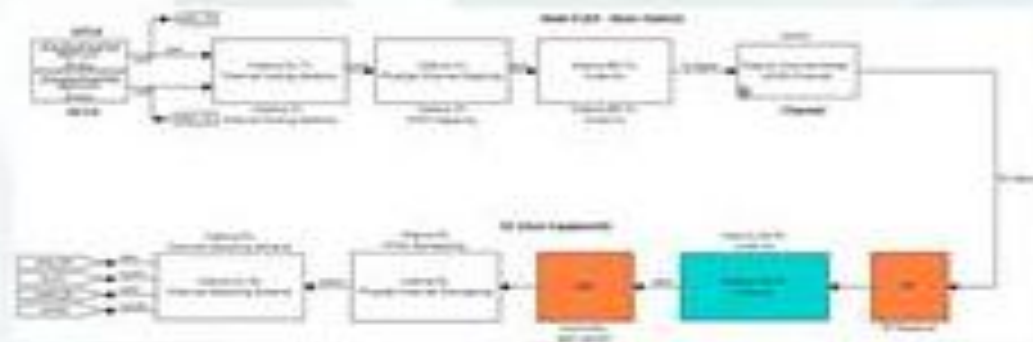
Sin lenguaje común







Simulink como lenguaje Común



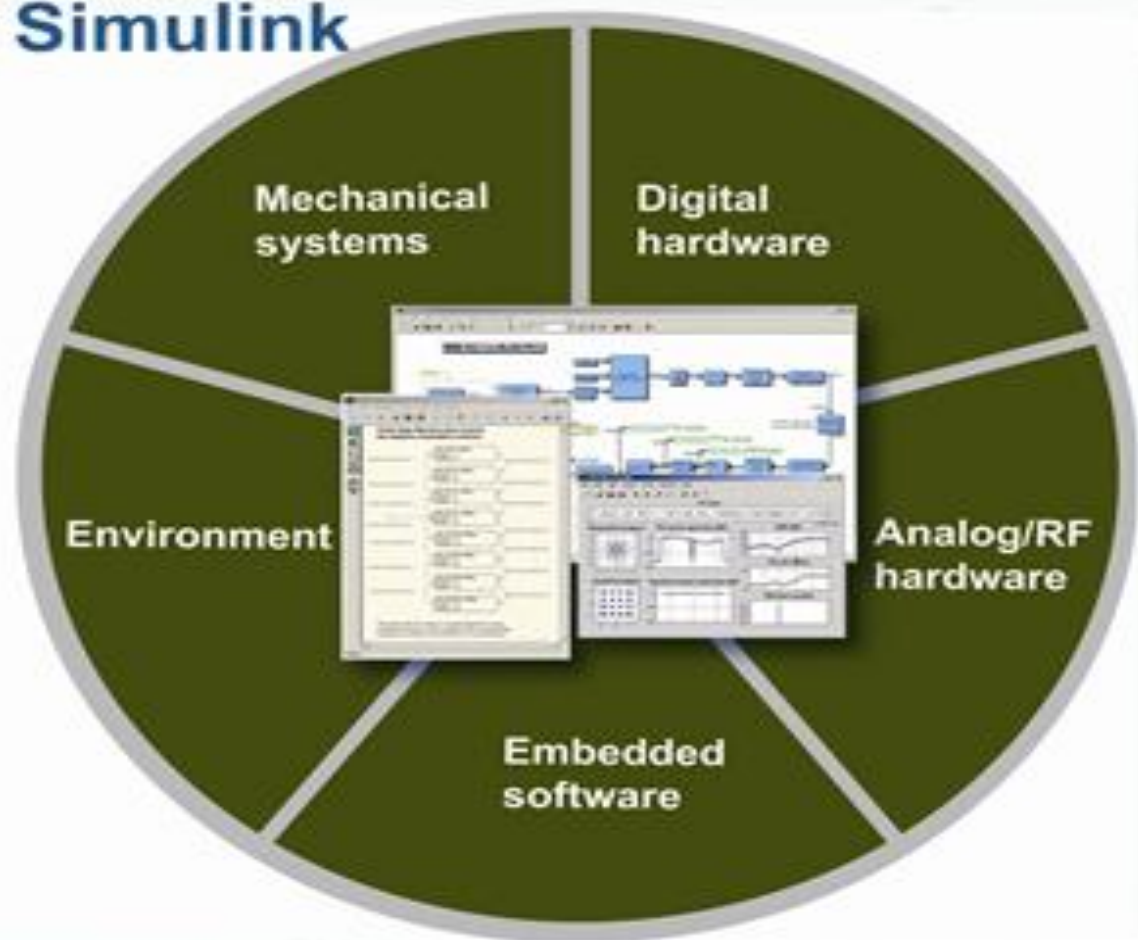
Linking tools and teams

- Enlace entre herramientas analógicas y digitales
- Visualiza el ambiente dinámico

Modelado de Sistemas multidominio en MATLAB y Simulink



Modelado de Sistemas multidominio en MATLAB y Simulink



Spectrum of Modeling Approaches for Analog

High speed ← High fidelity →

Laplace/ODEs

Behavioral models
of components

Implementation
level

Simulink

RF Blockset
SimPowerSystems

Co simulation
with third-party
circuit simulators

MATLAB 7.12.0 (R2011a)

File Edit Debug Paral

Program Files\MATLAB\R2011a\bin

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
fx >>
```

Workspace

Name	Value	Min	Max

Command History

```
-- 27/02/2012 09:56 a.m. --
-- 06/03/2012 01:43 p.m. --
```

Aerospace
 Bioinformatics
 Communications System
 Computer Vision System
 Control System
 Curve Fitting
 Data Acquisition
 Database
 Datafeed
 DSP System
 Econometrics
 Filter Design HDL Coder
 Financial
 Financial Derivatives
 Fixed-Income
 Fixed-Point
 Fuzzy Logic
 Global Optimization
 Image Acquisition
 Image Processing
 Instrument Control
 Mapping
 Model Predictive Control
 Neural Network
 Optimization
 More...

Test and Measurement Tool (tmtool)
 MATLAB Instrument Driver Editor (midedit)
 MATLAB Instrument Driver Testing Tool (midtest)
 Help
 Demos
 Product Page (Web)

m3registry
 registry
 util
 win64
 deploytool.bat
 insttype.ini
 lcdata.xml
 lcdata.xsd
 license.txt
 matlab.bat
 matlab.exe
 mbuild.bat
 mcc.bat
 mex.bat
 mex.pl
 mexext.bat
 mexsetup.pm
 mexutils.pm
 mw_mpiexec.bat
 ProductRoots
 workspace.bat
 MATLAB
 Toolboxes
 Simulink
 Shortcuts
 Desktop Tools
 Web
 Get Product Trials
 Check for Updates
 Preferences...
 Find Files...
 Help
 Demos

Test & Measurement

- Instrument Control Toolbox
 - Hardware
 - Serial
 - GPIB
 - VXI
 - GPIB-VXI
 - TCPIP (VXI-11)
 - USB**
 - TCPIP
 - UDP
 - Instrument Objects
 - Interface Objects
 - Device Objects
 - Instrument Drivers
 - MATLAB Instrument Drivers
 - aqilent_33120a.mdd
 - aqilent_34401a.mdd
 - aqilent_e3648a.mdd
 - qeneric_aqilent_33120a.mdd
 - qeneric_aqilent_e3648a.mdd
 - lecroy_8600a.mdd
 - lecroy_lt3441_ex.mdd
 - tektronix_tds2024.mdd
 - tektronix_tds210.mdd
 - VXIplug&play Drivers
 - IVI

USB

Vendor ^	Configuration	Identification

Last scan date: 06-Mar-2012 13:45:43

Scan

Help

Selecting a USB Instrument

Selecting the USB node enables you to scan for installed USB instruments.

1. Click **Scan** in the lower-right of the center panel.
2. Scanning returns the available USB instruments and adds a subnode for each instrument found.
3. Expand the USB node and select the instrument you want to use.
4. Establish communication with the instrument.

Simulink Library Browser

File Edit View Help

Enter search term

Libraries

- Commonly Used Blocks
- Continuous
- Discontinuities
- Discrete
- Logic and Bit Operations
- Lookup Tables
- Math Operations
- Model Verification
- Model-Wide Utilities
- Ports & Subsystems
- Signal Attributes
- Signal Routing
- Sinks
- Sources
- User-Defined Functions
- Additional Math & Discrete
- Aerospace Blockset
- Communications System Toolbox
- Computer Vision System Toolbox
- Control System Toolbox
- DSP System Toolbox
- EDA Simulator Link
- Embedded Coder
- Fuzzy Logic Toolbox
- Image Acquisition Toolbox
- Instrument Control Toolbox
- Model Predictive Control Toolbox
- Neural Network Toolbox
- Report Generator
- Robust Control Toolbox
- SimEvents
- SimPowerSystems
- SimRF
- Simscape
- Simulink 3D Animation
- Simulink Coder
- Simulink Control Design
- Simulink Design Optimization
- Simulink Design Verifier
- Simulink Extras
- Simulink Verification and Validation
- Stateflow
- System Identification Toolbox
- xPC Target

Library: Instrument Control Toolbox Search Results: (none) Most Frequently Us

- Query Instrument Serial Configuration
- Serial Receive Serial Send
- TCP/IP Receive TCP/IP Send
- To Instrument UDP Receive
- UDP Send

Showing: Instrument Control Toolbox

Workspace

Stack: Select data to plot

untitled *

File Edit View Simulation Format Tools Help

100 Normal

Ready 100% ode45

Command History

```

%-- 27/02/2012 09:56 a.m. --%
%-- 06/03/2012 01:43 p.m. --%

```

MATLAB R2011b

10/1/2012 1:24

Simulink Library Browser

File Edit View Help

Enter search term

Libraries

- Simulink
 - Commonly Used Blocks
 - Continuous
 - Discontinuous
 - Discrete
 - Logic and Bit Operations
 - Lookup Tables
 - Math Operations
 - Model Verification
 - Model-Wide Utilities
 - Ports & Subsystems
 - Signal Attributes
 - Signal Routing
 - Sinks
 - Sources
 - User-Defined Functions
 - Additional Math & Discrete
- Aerospace Blockset
- Arduino Target
- Communications System Toolbox
- Computer Vision System Toolbox
- Control System Toolbox
- DSP System Toolbox
- Data Acquisition Toolbox
- EDA Simulator Link
- Embedded Coder
- Fuzzy Logic Toolbox
- Gauges Blockset
- Image Acquisition Toolbox
- Instrument Control Toolbox
- Model Predictive Control Toolbox
- Neural Network Toolbox
- OPC Toolbox
- Real-Time Windows Target
- Robotics

Showing Simulink/Sinks

Library: Simulink/Sinks Search Results

- Display
- Floating Scope
- Out
- Scope**
- Stop Simulation
- Terminator
- To File
- To Workspace
- XY Graph

untitled

File Edit View Simulation Format Tools Help

Format

Scope

Time offset: 0

VariableStepDiscrete

link to using the solver 'VariableStepDiscrete' specifying a discrete solver in the solver tab of parameter selection' diagnostic to 'none' in the

Stack: Base - plot(bout)

Value	Size
<1000x1 double>	1000x1

Command History

Start

Quit

The screenshot displays the MATLAB Simulink environment. On the left, the Simulink Library Browser is open, showing various block categories. The 'Sinks' category is selected, and the 'Scope' block is highlighted. In the center, a block diagram shows a 'Scope' block connected to an 'Analog Input' block. A 'Scope' window is open, displaying a plot of a signal over time. The plot shows a signal that is mostly flat at zero, with some small fluctuations. The x-axis is labeled 'Time offset: 0' and ranges from 0 to 10. The y-axis ranges from -5 to 5. Below the plot, there is a message about the solver 'VariableStepDiscrete'.



2.6. Simulink

Simulink es un ambiente interactivo para modelar y simular una gran variedad de sistemas dinámicos, incluyendo sistemas lineales, no lineales, entre otros. Se trata de un entorno gráfico, el sistema se representa como una interconexión de bloques elementales, cada bloque lleva asociado un modelo matemático que representa su relación entrada/salida. Combina la potencia y facilidad del uso de un paquete de aplicación con la flexibilidad y extensibilidad de un lenguaje. Simulink permite seleccionar, adaptar y crear componentes de software y hardware para facilitar tus necesidades específicas. Simulink está constituido sobre el ambiente de técnicas computarizadas del Matlab y Simulink que proveen de un ambiente integrado ideal para desarrollar modelos ejecutando simulaciones de sistemas dinámicos y diseñando y probando nuevas ideas.

Dr. Jaime Cid Monjaraz

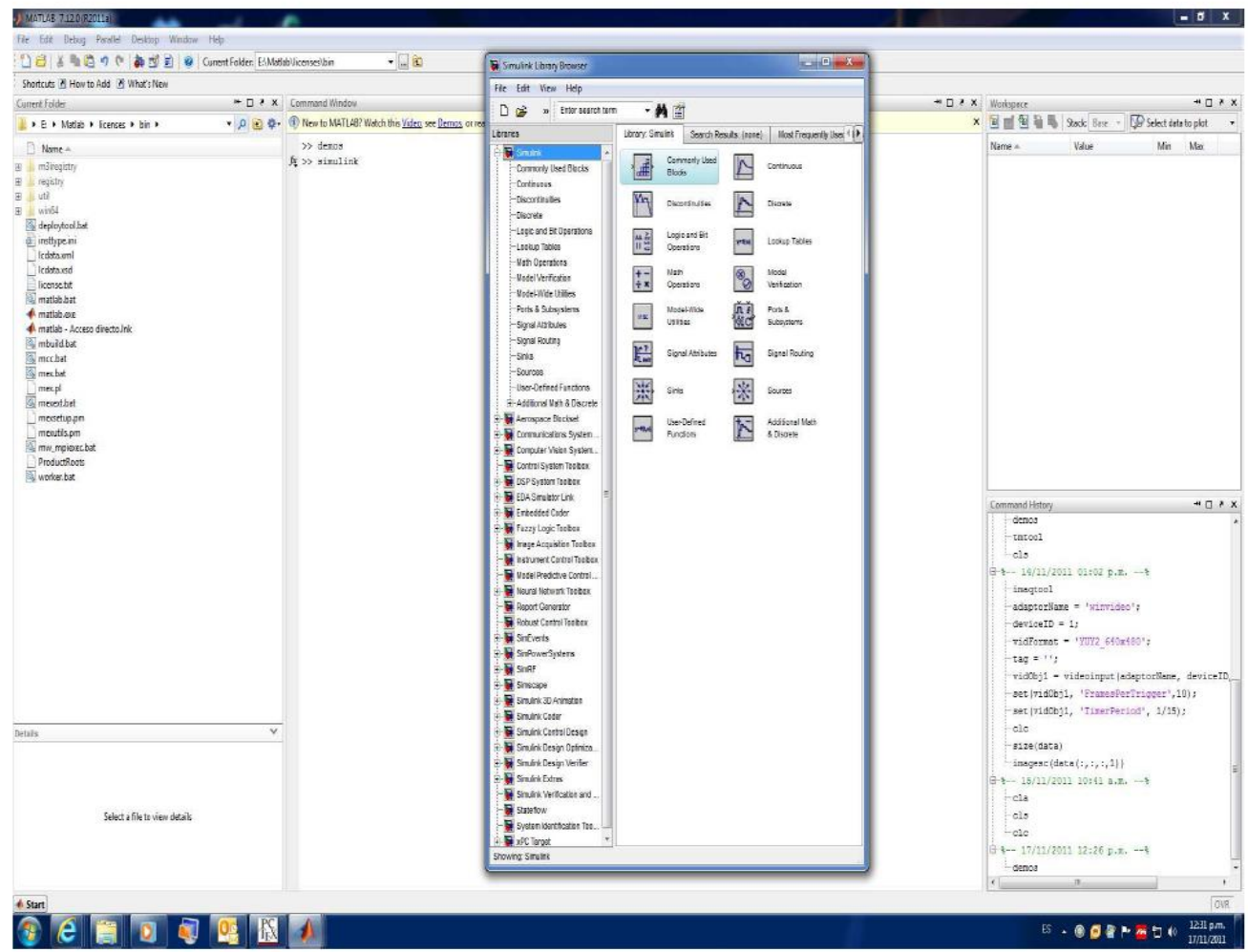


Figura 2.1: Inicio de Simulink

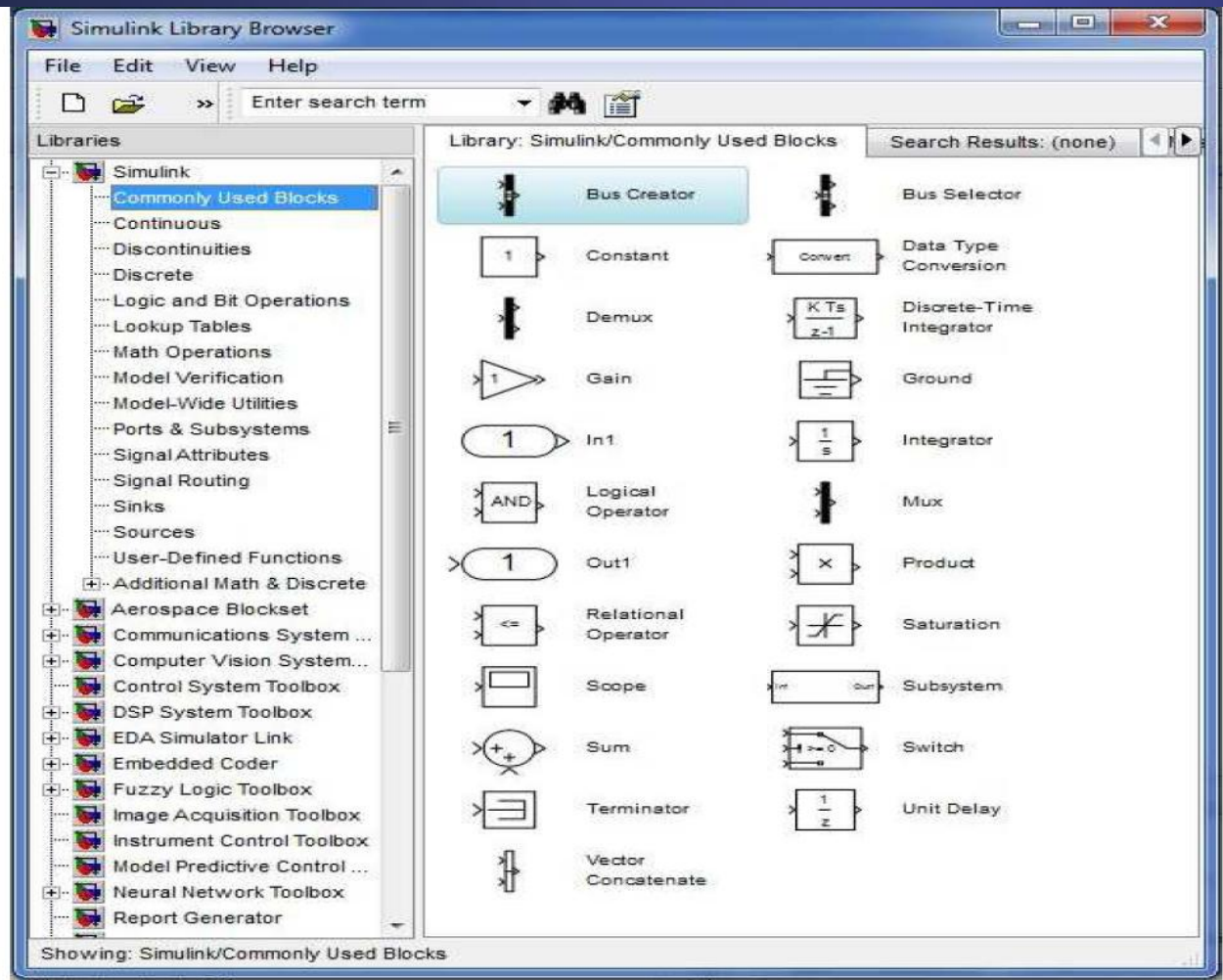


Figura 2.2: Navegador de librerías más comunes

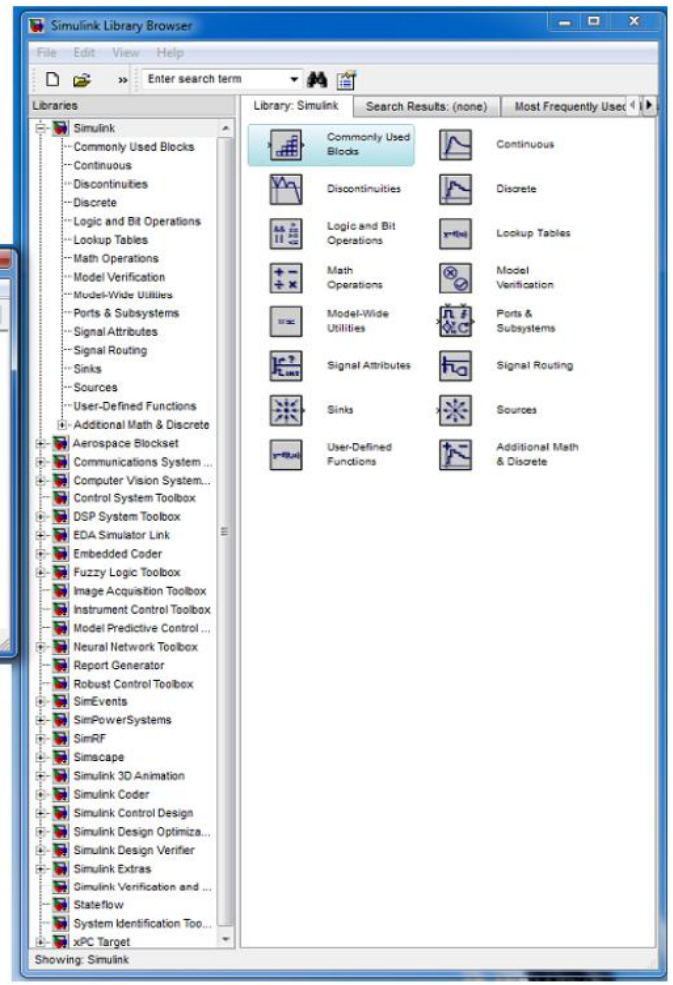
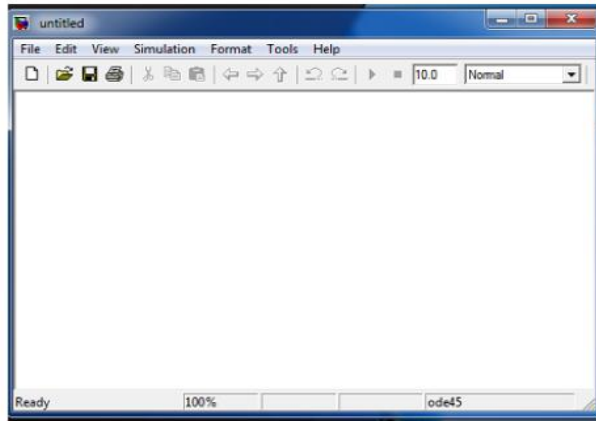


Figura 2.3: Nuevo modelo

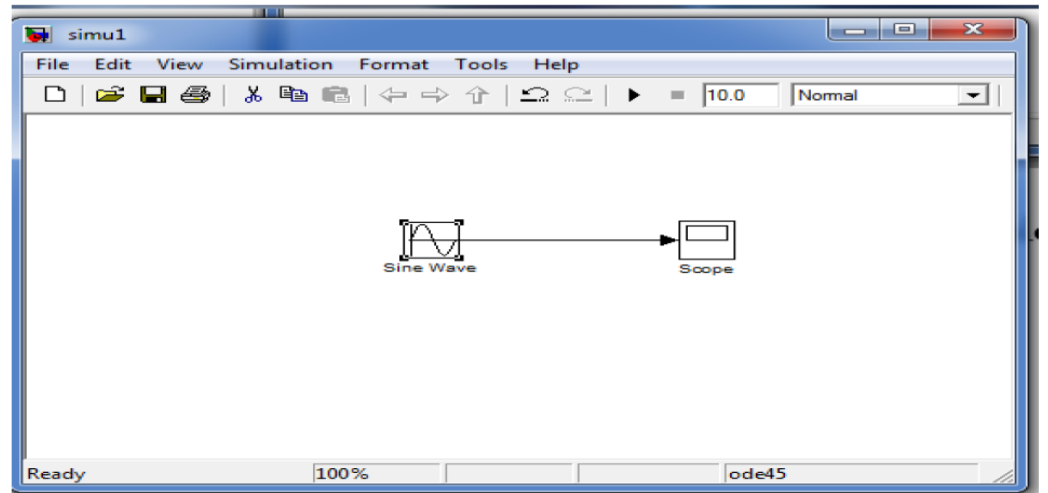


Figura 2.5: Conectando bloques

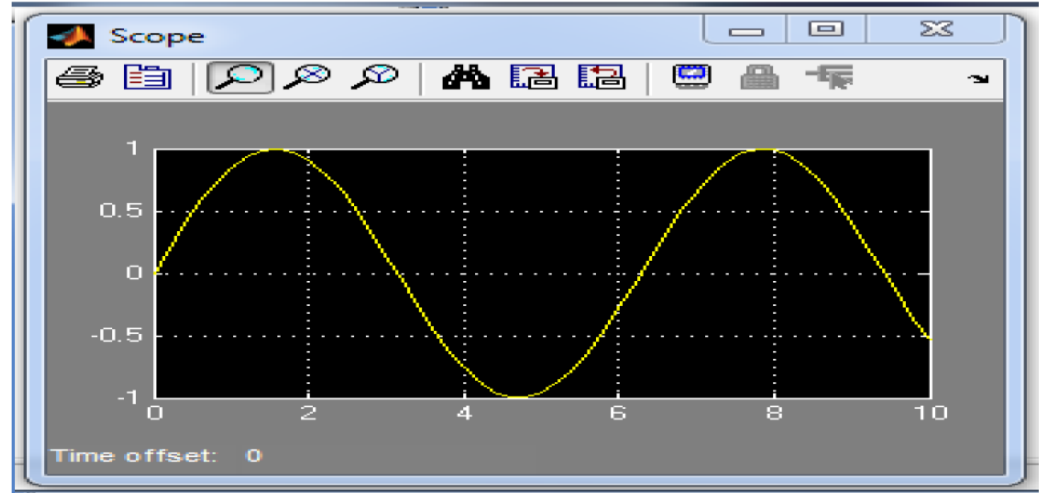


Figura 2.6: Resultado de la gráfica



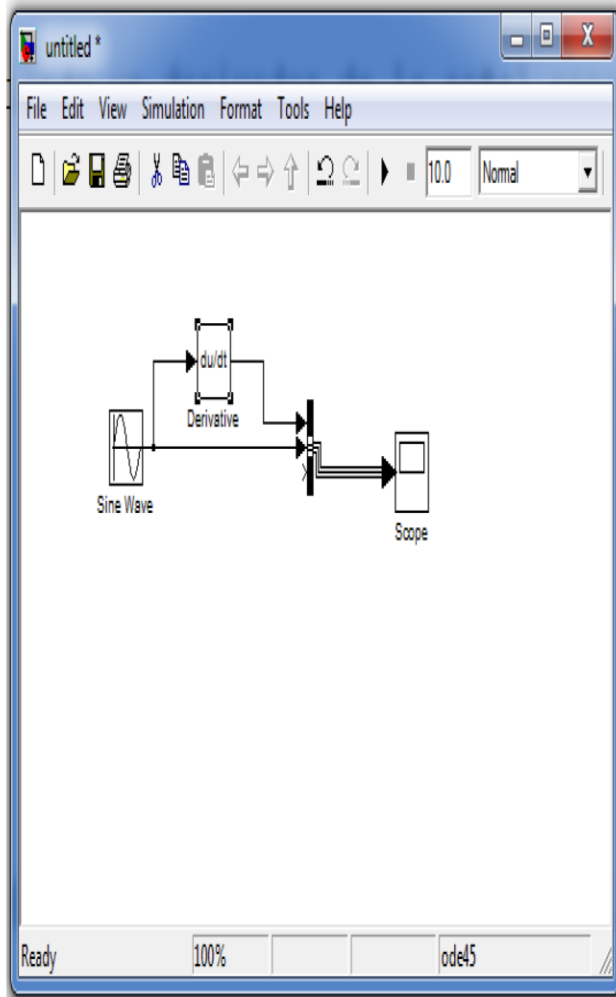


Figura 2.7: Modelo con un derivador de una señal senoidal

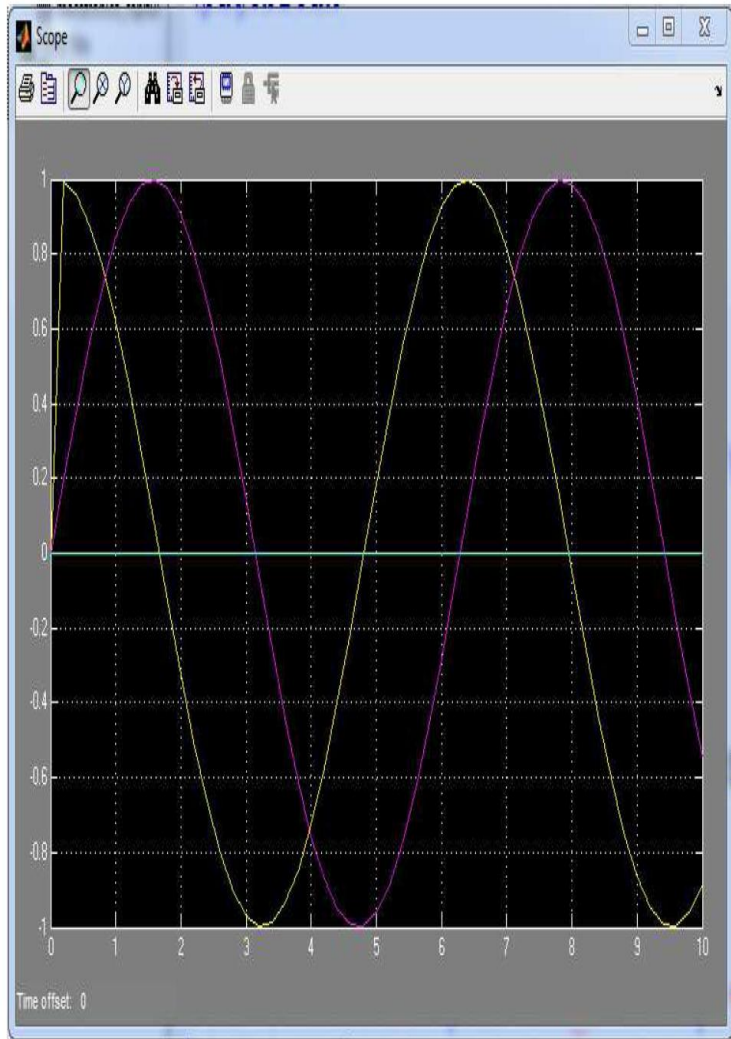


Figura 2.8: Resultado de la derivación de la señal

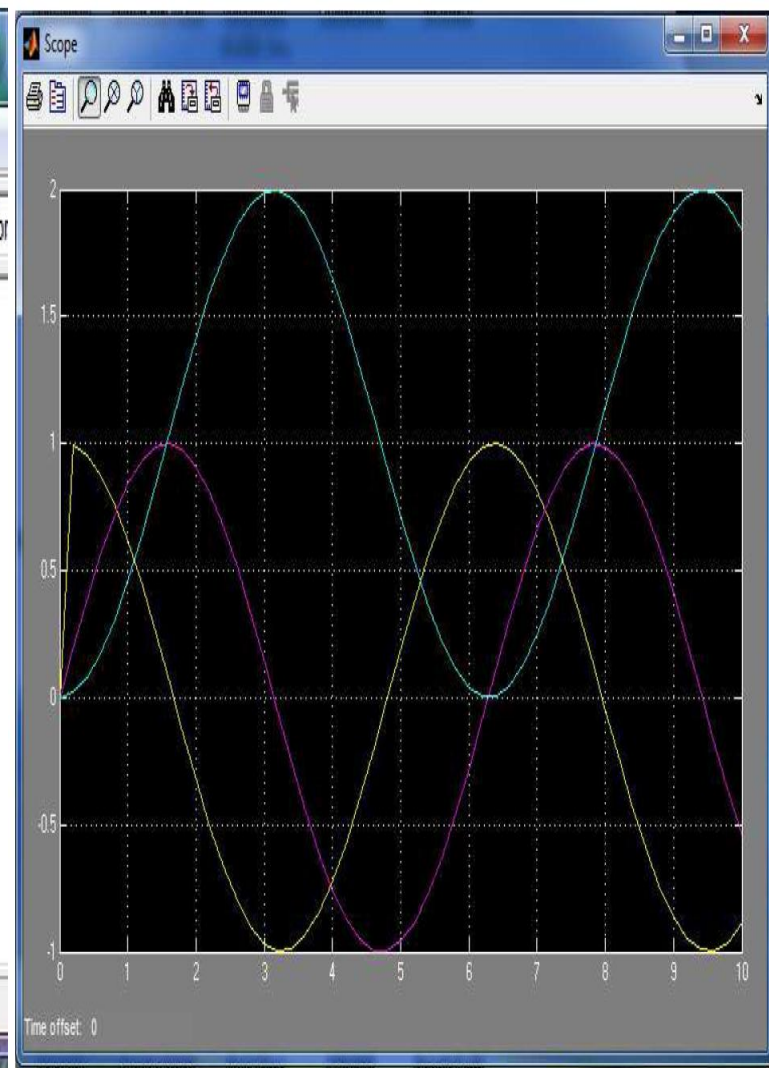
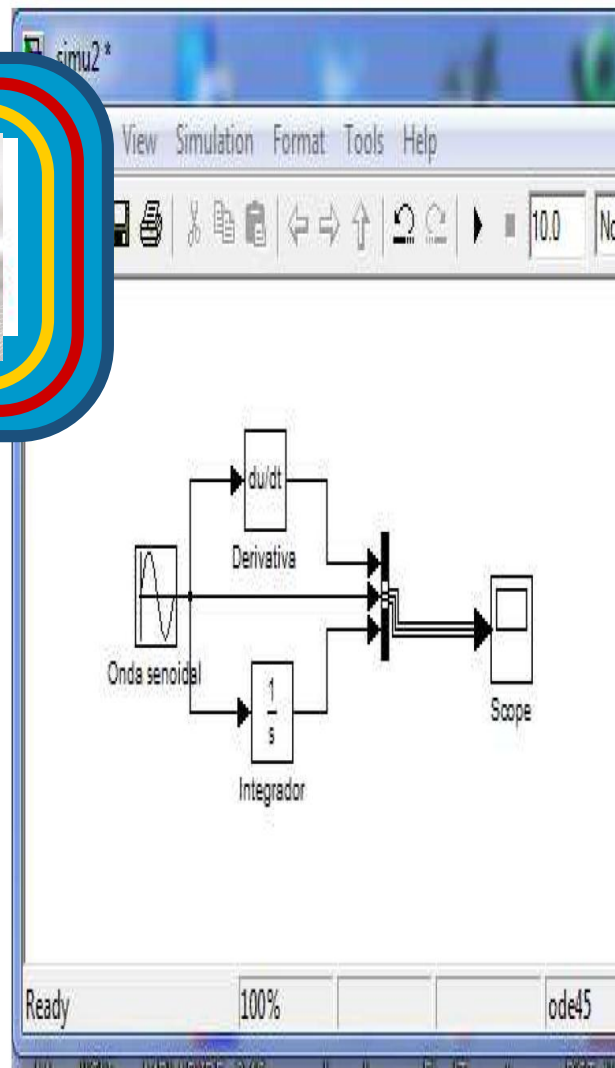
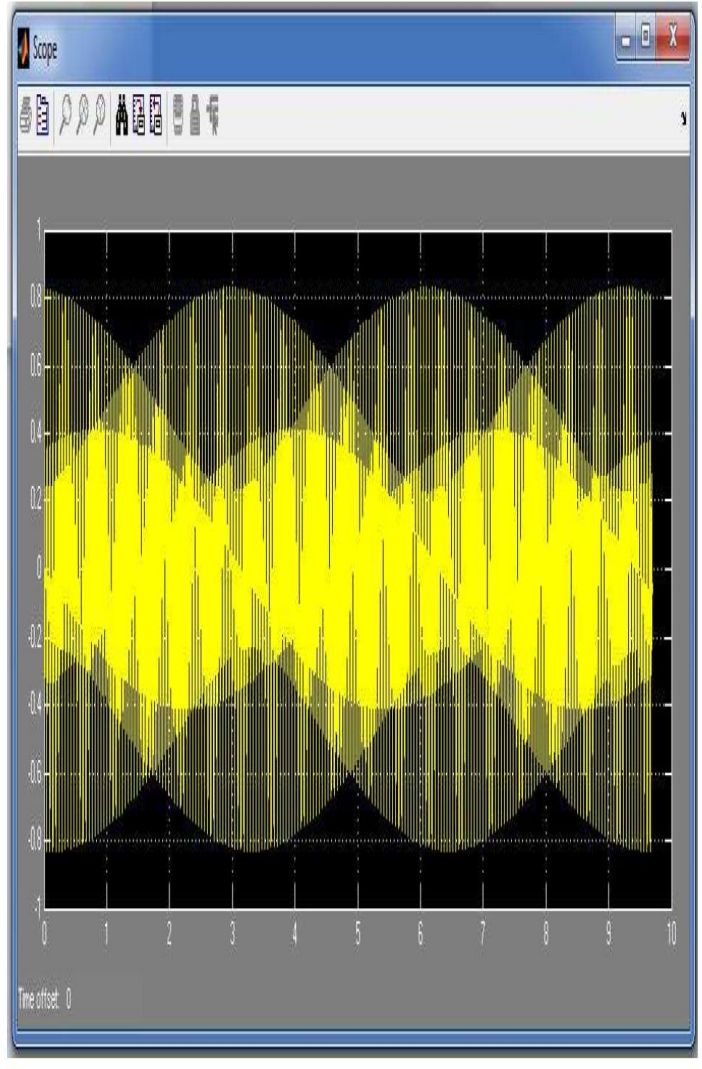
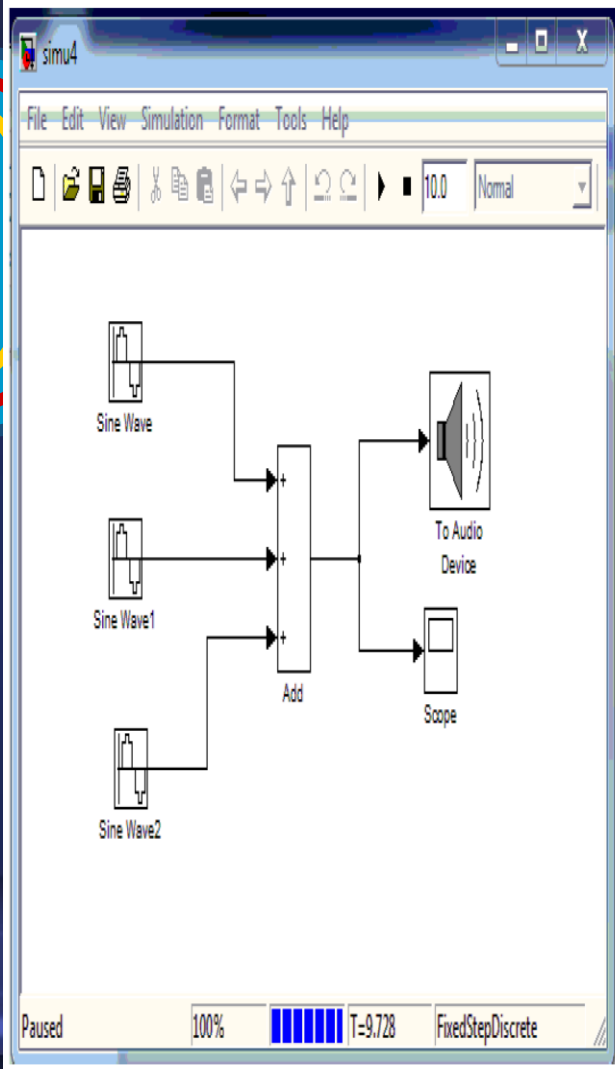


Figura 2.9: Modelo con un derivador e integrador a 2.10: Resultado de la integración y derivación de la señal



: Modelo con tres señales senoidales que tiene distintas frecuencias. Resultado de la suma de señales senoidales de distintas frecuencias

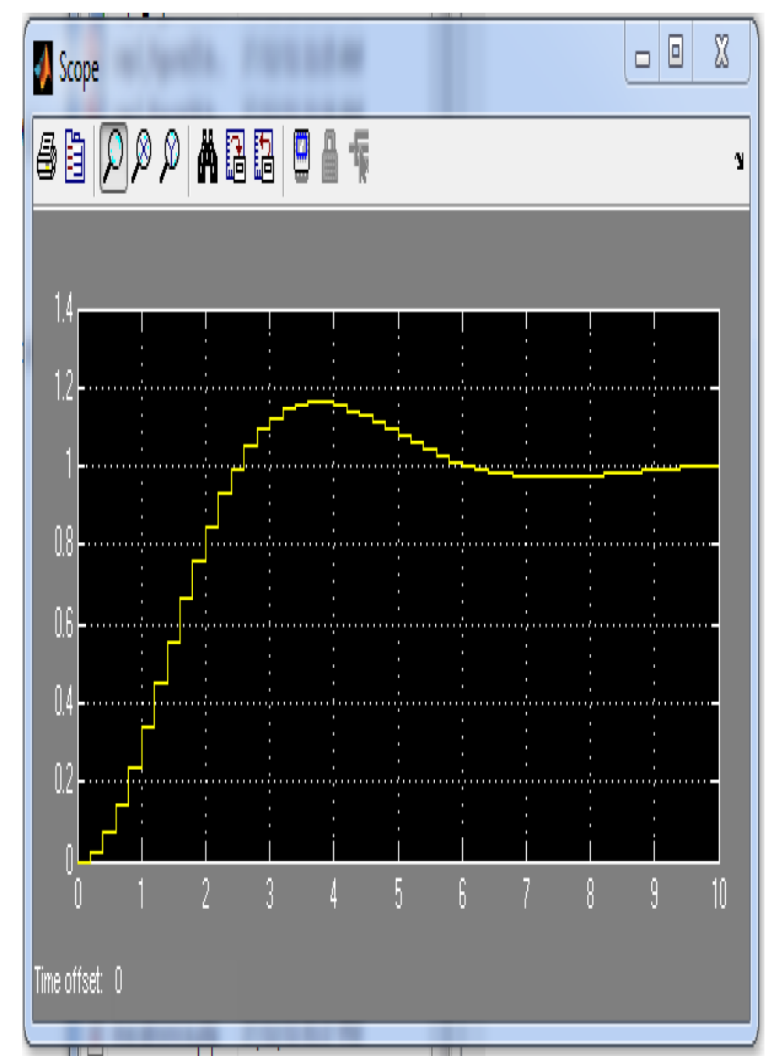
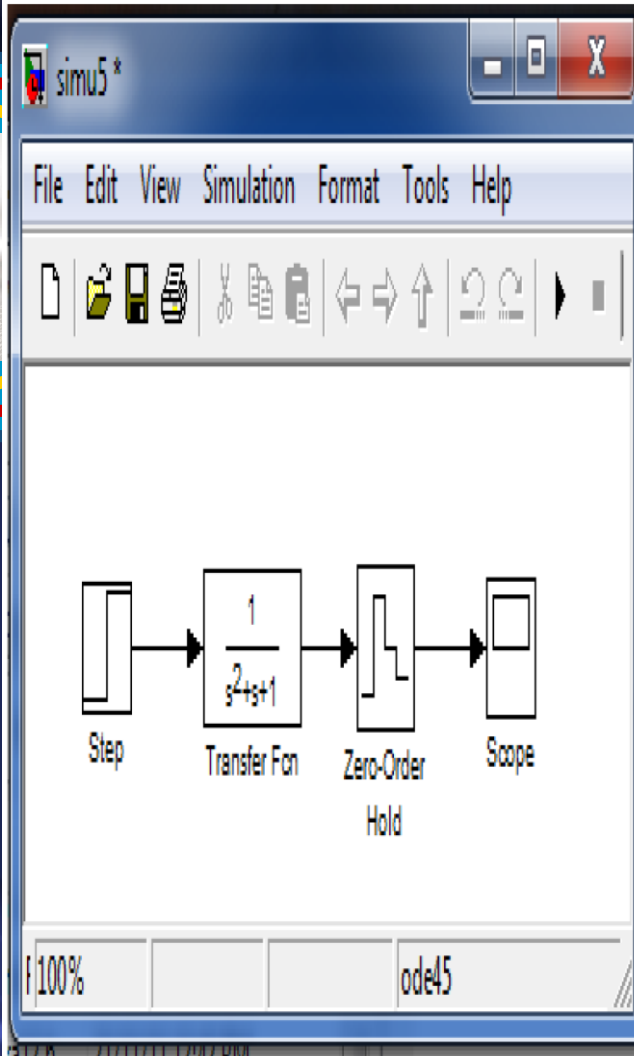
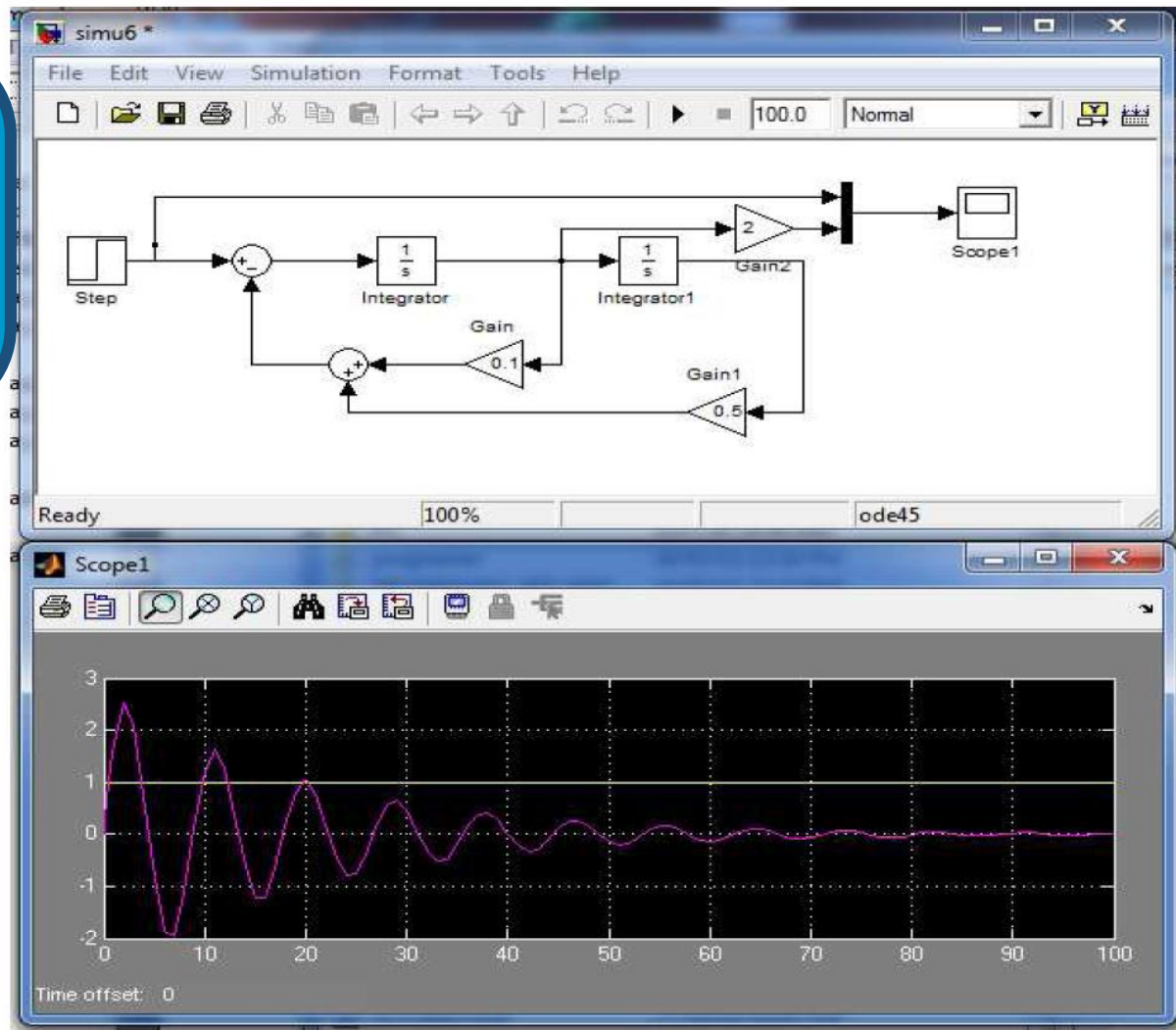


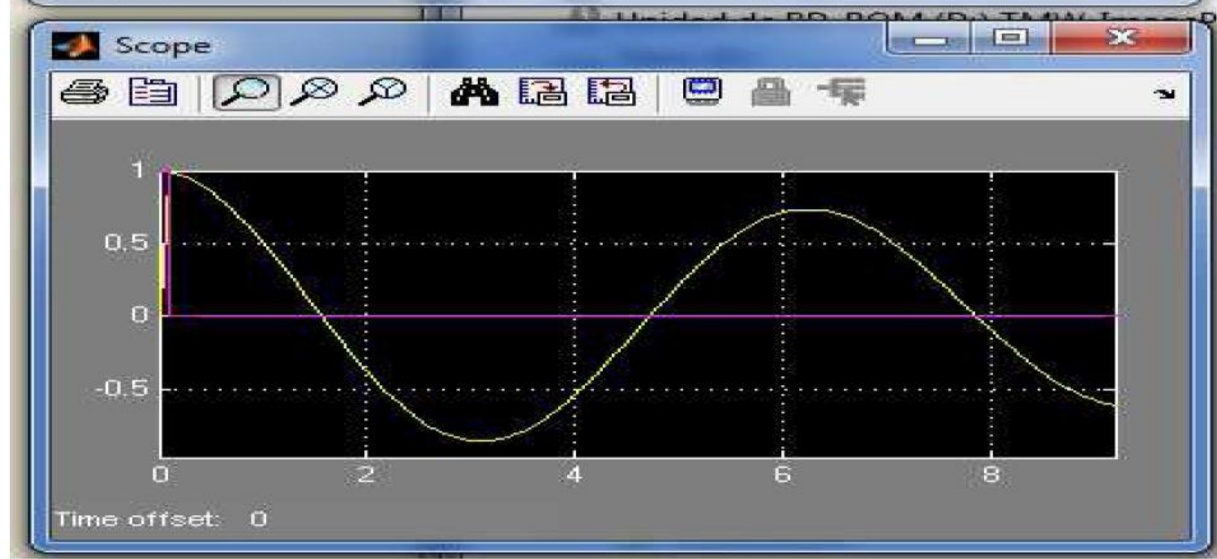
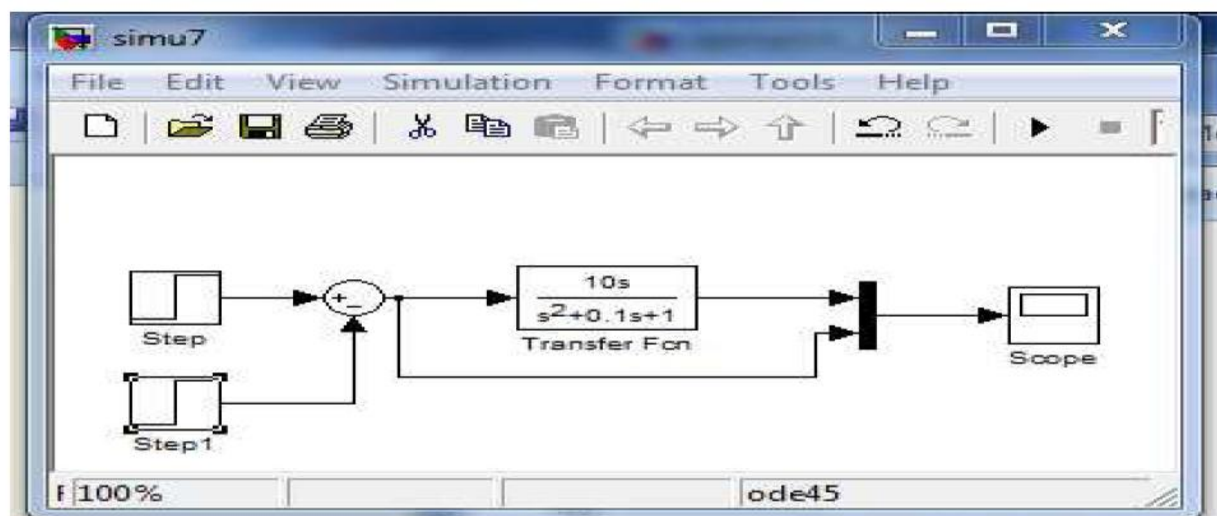
Figura 2.13: Modelo de función de transferencia en lazo abierto Figura 2.14: Resultado con el retenedor de orden cero



15: Modelo de ecuación diferencial y su resultado en la gráfica



FCE
Facultad de Ciencias
de la Electrónica



Modelo de entrada de impulso unitario para una función

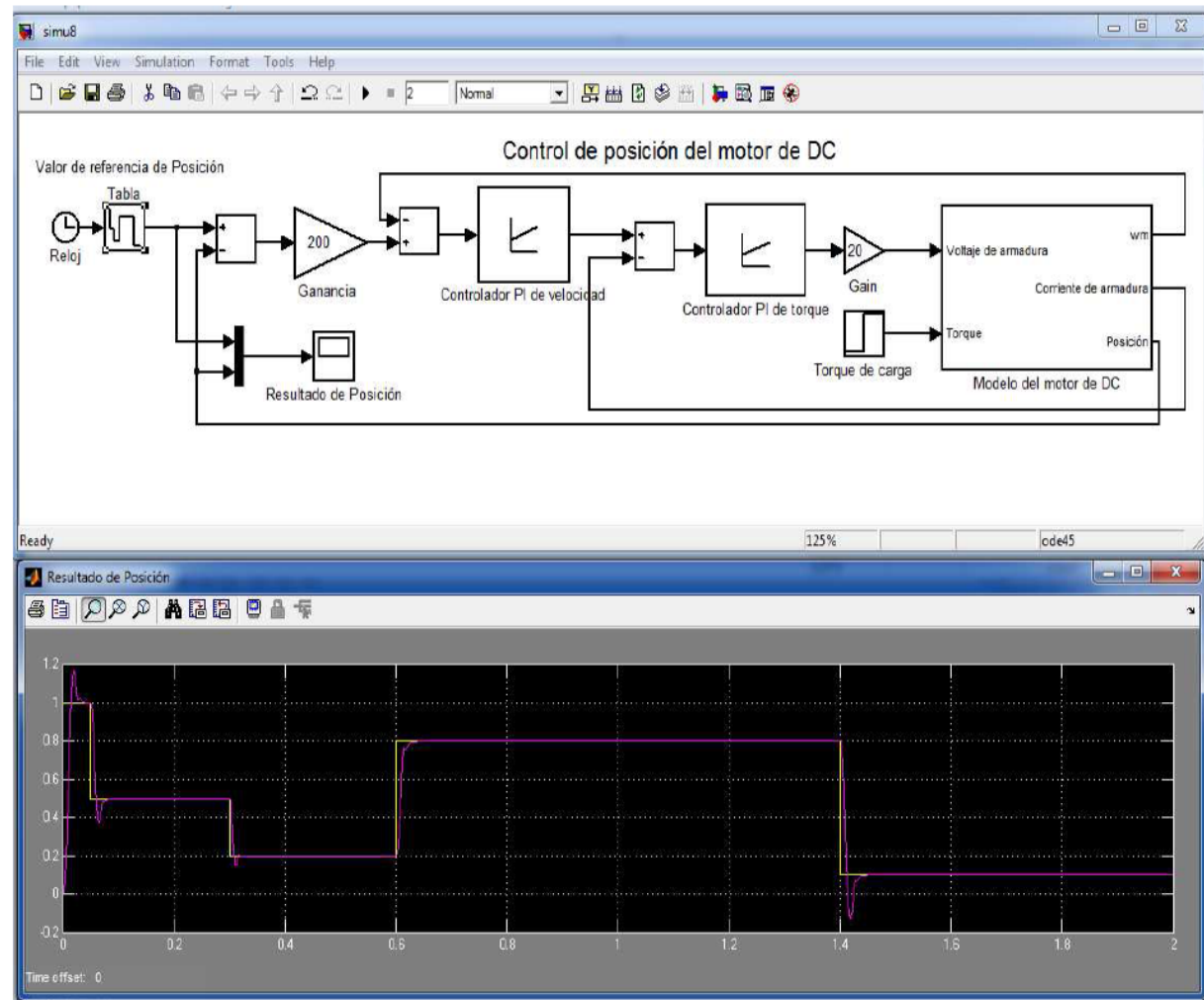


Figura 2.17: Modelo de control de posición de un motor de DC



file ^{img tool} → Generat M-file

→ micuadro.m ↗
.mat

vidobj = micuadro ↘

```
I = getsnapshot(vidobj);
```

```
imshow(I)
```

```
I_e = edge(I)
```

```
imshow(I_e)
```

```
imtool(I)
```

Guide
miguie

ipdemos →

```
plot(rand(10,1));
```

Traffic

2.7. Interfaz Gráfica de Usuario

La interfaz gráfica de usuario del Matlab recibe el nombre de Graphical User Interface Development Environment (GUIDE), es la manera en que un usuario puede interactuar con un programa de forma sencilla y sí se hace bien la interfaz poder trabajar intuitivamente. Contiene diferentes elementos gráficos tales como: botones, barras deslizantes, campo de texto, menús, gráficos, inspector de propiedades, etc.

2.7.1. El entorno de GUIDE

GUIDE proporciona un entorno gráfico al usuario que facilita diseño e interacción con el programa, al incluir una interfase sencilla para su uso. Esta herramienta proporciona un entorno gráfico que permite al usuario poner bloques para su uso. GUIDE usa los bloques para representar enlaces dinámicos, en lugar de dibujar los bloques individualmente, son copiados de una librería de bloques estándar suministrados por GUIDE. Para utilizar la interfaz gráfica de usuario existen diferentes modos de iniciar una sesión, se puede escribir

Al iniciar GUIDE, se obtiene una ventana como la que se muestra en la figura 2.18, que tiene dos opciones principales crear una nueva GUI o abrir una existente.

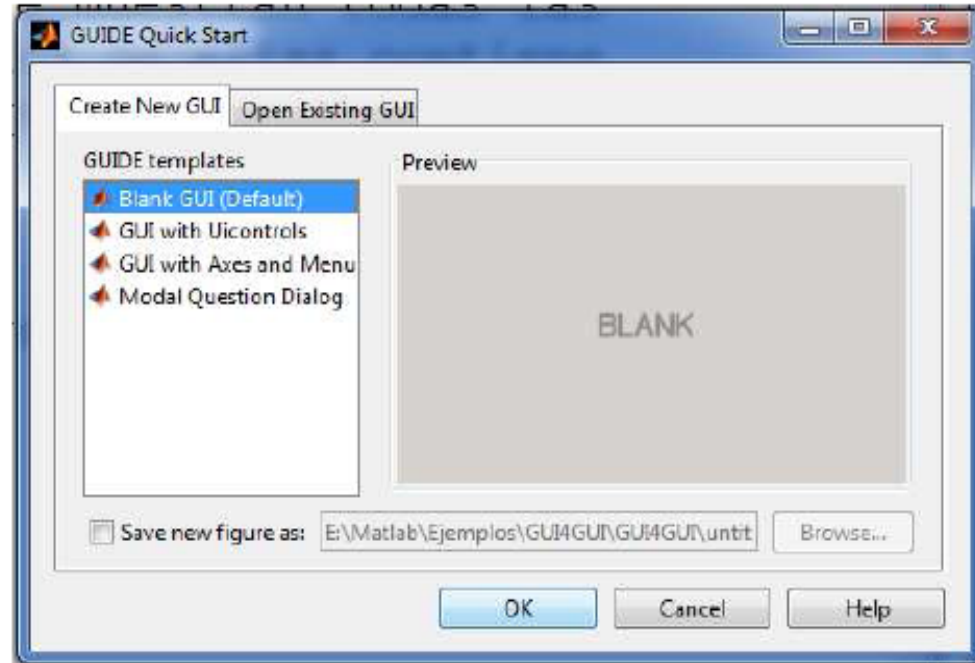


Figura 2.18: Inicio de GUIDE

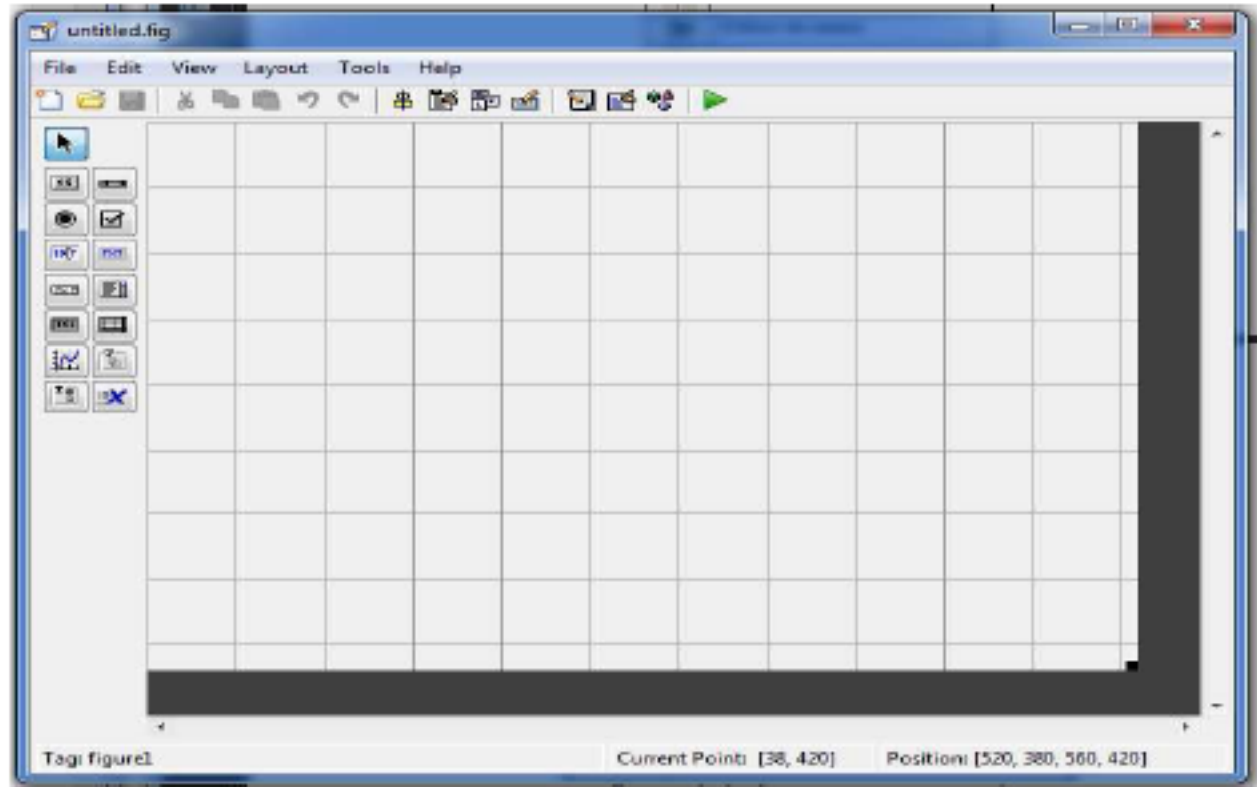


Figura 2.19: Inicio de Interfaz gráfica

Dr. Jaime Cid Monjaraz

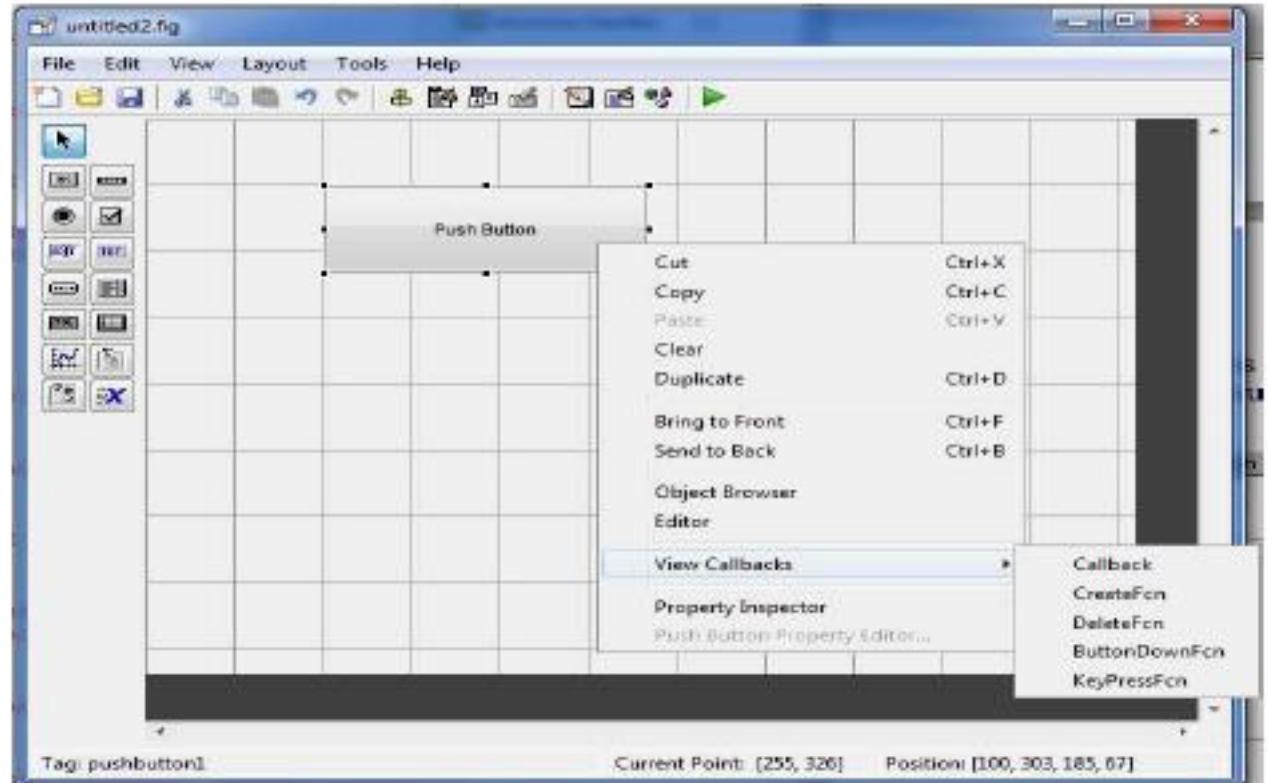


Figura 2.20: Propiedades del elemento

Dr. Jaime Cid Monjaraz

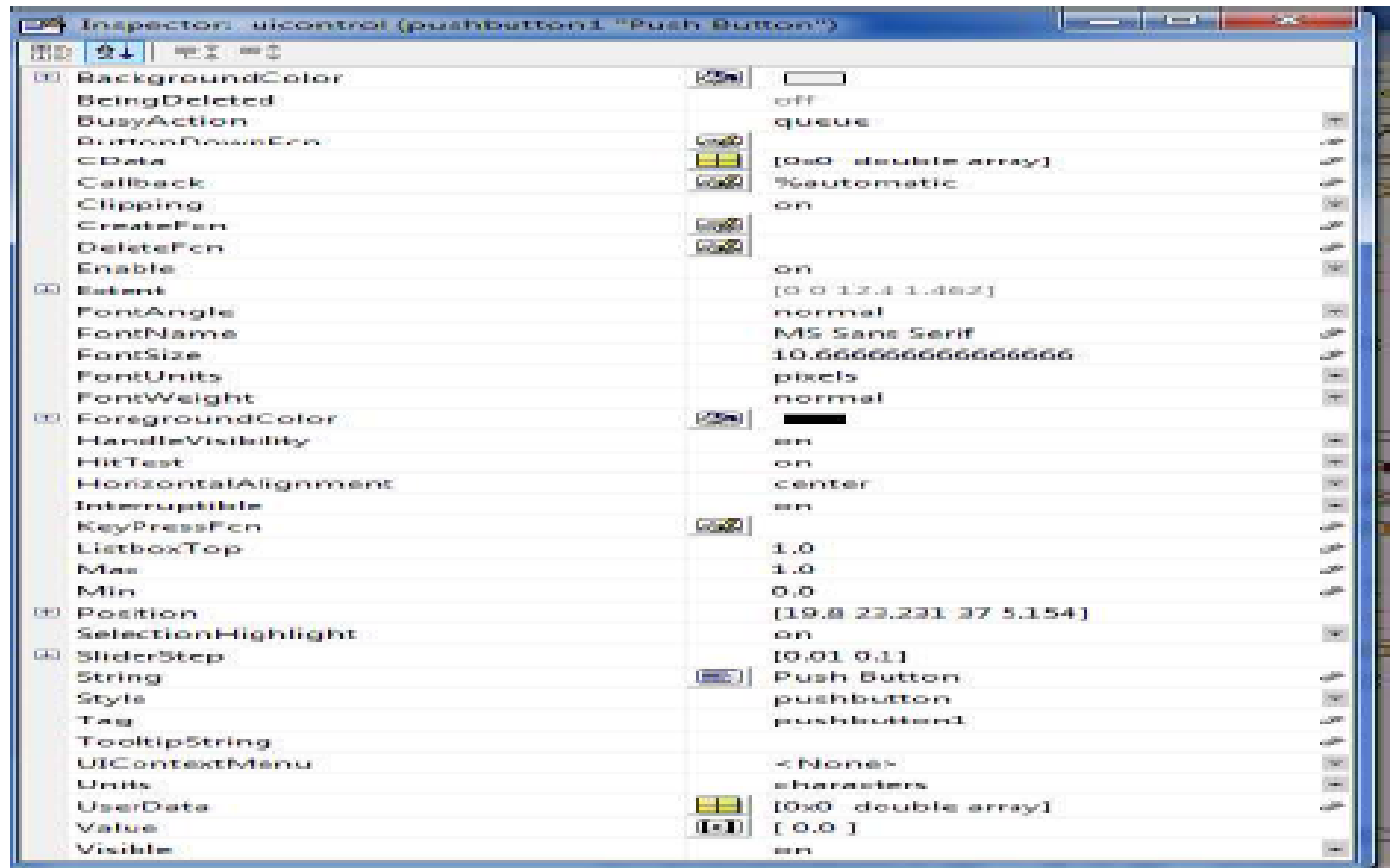


Figura 2.21: Editor de Inspector de Propiedades

vo. Por ejemplo si se guarda un archivo `gui1.fig` y `gui1.m` escribiendo `gui1` y presionando enter se ejecuta el programa de la figura 2.22.

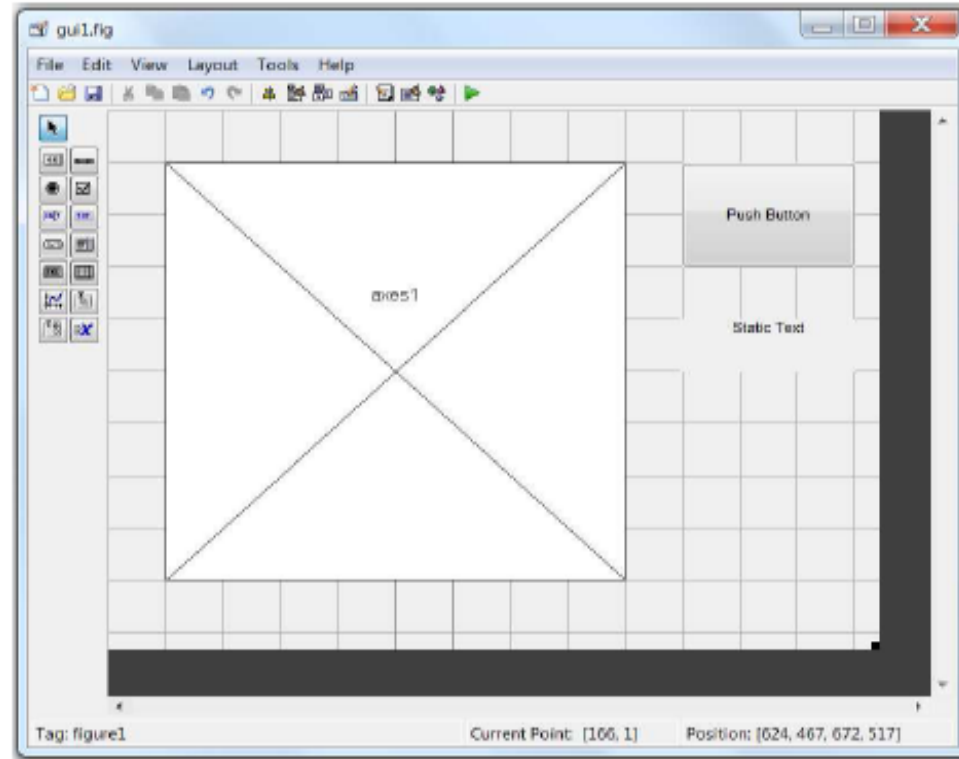


Figura 2.22: Primer Interfaz gráfica de usuario

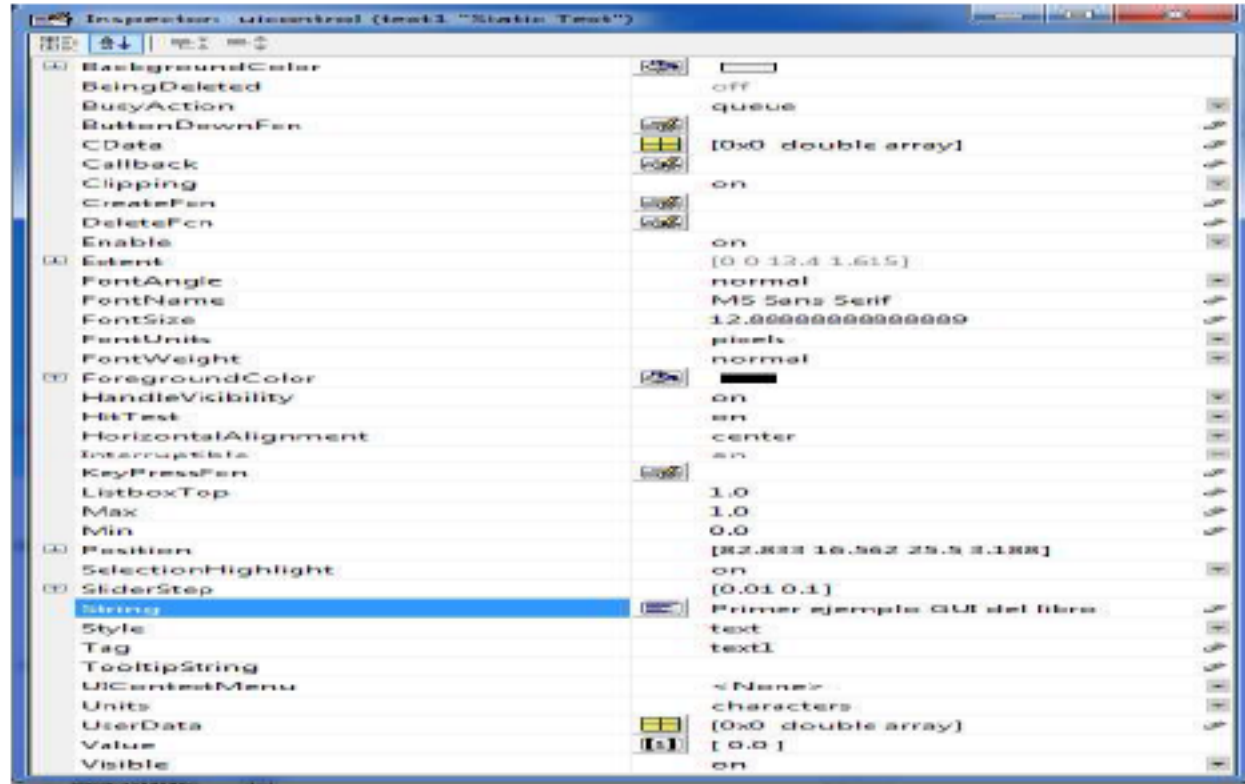


El archivo .m que se crea tiene una estructura predeterminada. Consta de un encabezado y a continuación viene el código correspondiente a las siguientes subrutinas. Por ejemplo una aplicación cuya figura tenga un botón, un gráfico y un cuadro de edición tendrá un archivo `gui1.fig` como se muestra en la figura 2.23 con la siguiente estructura inicial (todavía no se agrega el código de la subrutina) se presenta en el archivo generado `gui0.m` (quitando los comentarios) se puede observar en la imagen

Pulsando el botón derecho en cada uno de los iconos que se utilizan en la GUIDE se llama al inspector de propiedades y se puede cambiar su valores, por ejemplo el nombre del Static text, no posee una función asociada, pero sí una dirección, que se puede utilizar para escribir comentarios. Para saber cuál es esta dirección, también se puede hacer doble-click en esta componente, y se ubica la etiqueta Tag y String, como se muestra en la figura 2.24:

Si al componente de Push Button se pulsa el botón derecho del mouse aparece una opción de llamado al View Callbacks donde se abre el archivo `gui1.m` posee un campo donde se puede programar las condiciones iniciales que tendrá el diseño y se posiciona el cursor en el código relacionado con dicho componente, después de los comentarios que tiene se puede poner debajo de esta línea, el siguiente código:

Dr. Jaime Cid Monjaraz



2.24: Inspector de Propiedades para el componente Static Text

```

% TIMEOUT makes gui wait for user response (see UIRESUME)
% await(handles.figure);

% --- Outputs from this function are returned to the command line.
function varargout = gui_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Set default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.ejex=0*pi/360:2*pi;
y1=cos(2*pi*i*handles.ejex);
plot(handles.ejex,y1,'lineWidth',2);
axis([0 2*pi -1 1]);

```

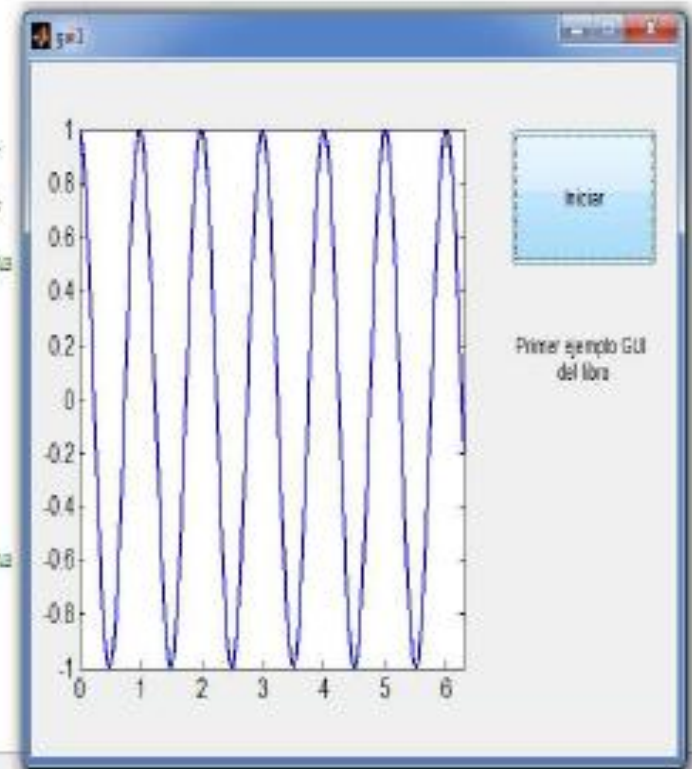


Figura 2.25: Resultado del primer ejemplo de GUI

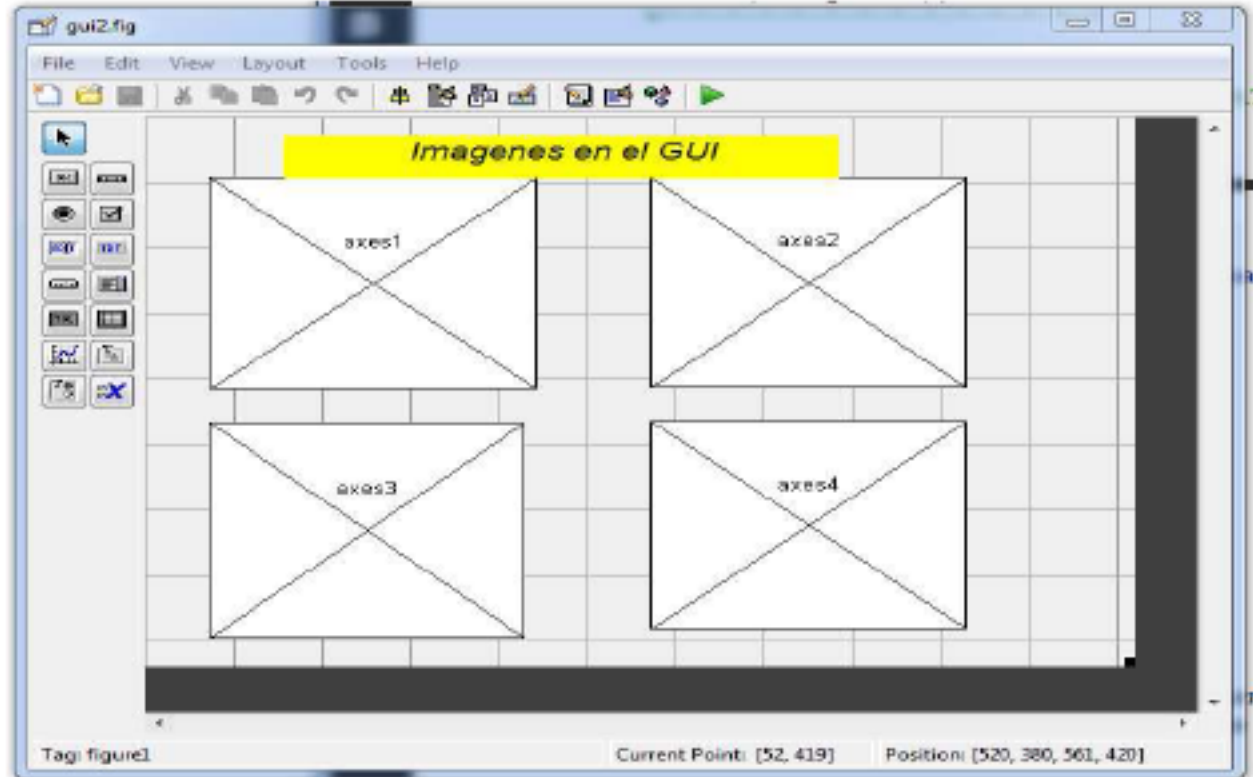


Figura 2.26: Programa GUI de imágenes

En la parte de inicialización del programa se edita lo siguiente:

```
varargout1 = handles.output;axes(handles.axes1) background = imread('love-Mex.jpg');  
axis off;
```

```
imshow(background);  
axes(handles.axes2) background = imread('escultura-inaugura.jpg');  
axis off;  
imshow(background);  
axes(handles.axes3)background = imread('familia.jpg');  
axis off;  
imshow(background);  
axes(handles.axes4) background = imread('ficp.jpg');  
axis off;  
imshow(background);
```

Los nombres de cada imagen son los que se muestran anteriormente. Se aclara que las imágenes deben estar en la misma carpeta donde está el programa `gui2.m`, obteniendo el siguiente resultado en la figura 2.27.



Figura 2.27: Programa GUI de imágenes

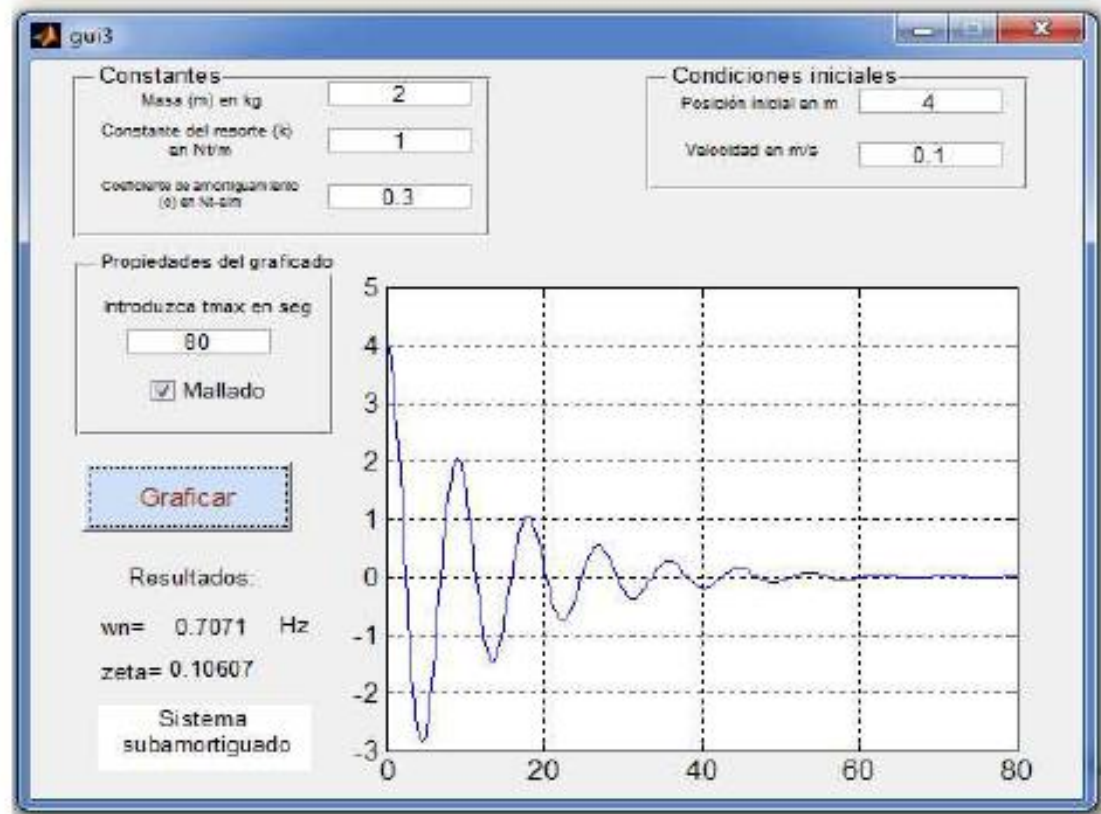


Figura 2.28: Programa GUI de sistema masa, resorte y amortiguador



Para finales de marzo de 2013 se aprueban los temas de los proyectos, a mediados de abril me entregaran un documento (3 0 4 paginas) con el siguiente contenido:

- Nombre del Proyecto, integrantes, carátula, etc
- Antecedentes (estado del Arte).
- Objetivo.
- Descripción funcional.
- Diagrama a bloques.
- Sistema Físico.
- Sistema Electrónico (Cámara web, sensores, actuadores, tarjeta de adquisición de datos y control, computadora, interfaz, etc).
- Programación (c, Matlab, Simulink, GUI, Arduino).
- Integración y Pruebas.
- Cronograma.

Dr. Jaime Cid Monjaraz