



Pontifícia Universidade Católica do Paraná
Centro de Ciências Exatas e de Tecnologia – CCET
Engenharia Elétrica – Ênfase Telecomunicações

Processamento Digital de Sinais (PDS)

Prof. Marcelo E. Pellenz - PPGIA – Tel. 3271-1690

<http://www.ppgia.pucpr.br/~marcelo>

e-mail: marcelo@ppgia.pucpr.br

Período: 6^o/8^o **Semestre:** 1^o./2^o **Ano:** 2005

1 Ementa

- Introdução ao processamento digital de sinais
- Processo de digitalização de sinais analógicos: conversão A/D, teorema de Nyquist amostragem, quantização, codificação e reconstrução do sinal analógico (Conversão D/A).
- Sinais e Seqüências Discretas: sinais senoidais discretos/contínuos, normalização de freqüências, periodicidade de seqüências discretas senoidais, interpretação de freqüências altas e baixas, interpretação da freqüência normalizada.
- Operações com seqüências discretas: seqüências discretas, impulso unitário, degrau unitário, seqüências exponenciais reais/complexas, seqüência par e ímpar.
- Representação de Sinais e Sistemas Discretos no Domínio do Tempo/Freqüência: sistemas discretos lineares invariantes no tempo (LTI), propriedades de sistemas LTI, convolução Linear, equação de diferenças.
- Transformada de Fourier em Tempo Discreto (DTFT)
- Transformada Discreta de Fourier (DFT/FFT)
- Transformada-Z
- Projeto de Filtros Digitais FIR e IIR
- Estrutura de Filtros Digitais e Aspectos de Implementação
- Aplicações do Processamento Digital de Sinais em Sistemas de Comunicação Digital

2 Objetivos

Capacitar o aluno para a caracterização, projeto e implementação de filtros digitais, análise espectral de sinais usando DFT e desenvolvimento de algoritmos para processamento digital de sinais aplicados em sistemas de comunicação.

3 Bibliografia

- Digital signal processing using MATLAB, V. K. Ingle and J. G. Proakis, Brooks/Cole, 2000
- Digital signal processing: principles, algorithms, and applications, J. G. Proakis and D. G. Manolakis, Prentice Hall, 1996
- Digital Signal Processing: A computer based approach, S. K. Mitra, McGraw-Hill, 1998
- Material didático (apostila) e técnico (artigos) fornecidos pelo professor

4 Procedimentos de Avaliação

- Três notas parciais no semestre, onde cada nota será composta pela nota da prova (peso 7) e da nota dos trabalhos de laboratório (peso 3) correspondentes ao período avaliado.
- O conteúdo das provas é acumulativo e também englobará as atividades e conceitos desenvolvidos em laboratório.
- Trabalhos serão desenvolvidos em sala de aula e em laboratório (Exercícios/Matlab/DSP), com a entrega dos resultados e/ou relatórios ao final do experimento.
- As datas das avaliações serão estabelecidas com um prazo mínimo de uma semana de antecedência.
- A média e freqüência mínimas para a aprovação serão regidas pelas regras da instituição.

5 Observações Importantes

- O desenvolvimento de alguns experimentos de laboratório será feito em grupo. Após a definição dos grupos, eles não podem ser modificados. Um aluno do grupo será escolhido pelo professor para defender o trabalho no momento da apresentação.
- Os relatórios devem ser entregues impreterivelmente na data e horário limite estipulado pelo professor. Não serão aceitos relatórios fora do prazo.
- Os alunos que faltarem as aulas de laboratório deverão realizar o experimento extra classe e entregar os resultados/relatórios impreterivelmente na próxima aula.

6 Material de Laboratório

- 2 fones de ouvido para desenvolvimento dos experimentos de laboratório no Matlab e no kit DSP

ÍNDICE

Introdução ao processamento digital de sinais	5
Digitalização de Sinais e Normalização de Freqüências	28
Periodicidade de Seqüências Discretas	30
Interpretação da Freqüência Normalizada	31
Sinais de Freqüências Distintas com o mesmo Período	32
Interpretação de Freqüências Altas e Baixas	33
Interpretação de Freqüências Altas e Baixas em Tempo Discreto	34
Operações com Seqüências Discretas	35
Função Impulso Unitário	35
Função Degrau Unitário	35
Pré-Lab 1 – Representação e Operações com Seqüências Discretas	36
Laboratório 1 – Representação e Operações com Sinais em Tempo Discreto	37
Pré-Relatório para o Laboratório 2	40
Decomposição em Seqüência Par e Ímpar	40
Sinais de Energia e Sinais de Potência	41
Escalonamento no Tempo (Down-Sampling)	41
Sistemas Discretos Lineares e Invariantes no Tempo (LTI)	41
Estabilidade de Sistemas LTI	42
Sistema LTI Causal	42
Correlação e Autocorrelação de Seqüências	42
Equação de Diferenças	43
Representação de sistemas usando diagrama de blocos	43
Pré-Lab 2 – Sistema de Discagem por Tons (DTMF)	43
Laboratório 2 – Sinais e Sistemas Discretos	44
Pré-Relatório para o Laboratório 3	46
Análise Espectral de Sinais Contínuos no Tempo	47
Análise Espectral de Sinais Discretos no Tempo	48
Transformada Discreta de Fourier (DFT)	53
Representação de Sistemas LTI no Domínio da Freqüência	56
Laboratório 3 – Transformada de Fourier Discreta (DFT)	57
Transformada de Fourier Discreta (DFT)	58
Transformada-Z	61
Região de Convergência (ROC)	61
Transformada-Z Inversa usando Frações Parciais	63
Relação da Função de Transferência com a Equação de Diferenças	64
Projeto de Filtros Digitais FIR	65
Laboratório 4 – Aplicações do Processamento Digital de Sinais	70
Laboratório 5 – Transformada-Z	74
Projeto de Filtros FIR	76

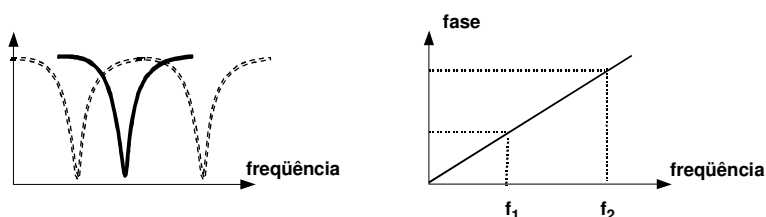
Prefácio

Este curso tem por objetivo apresentar os fundamentos da teoria de processamento digital de sinais aplicados à área de transmissão digital, demonstrando os conceitos e algoritmos através de simulação no software Matlab® e da implementação prática (em tempo real) destes algoritmos usando o kit DSP TMS320C5416. Os modernos dispositivos de comunicação (por exemplo: celulares/modems/rádios digitais/placas de rede local/placas de WLAN (WiFi)/Bluetooth) realizam todo o processamento do sinal na forma digital (modulação, demodulação, detecção de símbolo, recuperação de portadora, filtragem, sincronismo, equalização, detecção e correção de erros) utilizando DSPs. Os circuitos analógicos são utilizados apenas na conversão A/D-D/A, nos misturadores (mixers) para translação de frequência e nos estágios de amplificação de potência. Esta é a atual estratégia de desenvolvimento e implementação de dispositivos de comunicação digital.

1. Introdução ao Processamento Digital de Sinais

Porque Processamento Digital ?

- As técnicas de processamento digital de sinais são atualmente tão poderosas que é extremamente difícil, se não impossível, que o processamento analógico de sinais atinja desempenho similar.
- Exemplos:
 - Filtros FIR com fase linear
 - Filtros Adaptativos



AULA 1

1-1

Porque Processamento Digital ?

- O processamento analógico de sinais é feito através do uso de componentes analógicos:
 - Resistores.
 - Capacitores.
 - Indutores.
- As tolerâncias inerentes associadas com estes componentes, temperatura, variações de tensão e vibrações mecânicas podem afetar dramaticamente a eficiência dos circuitos analógicos.

AULA 1

1-2

Porque Processamento Digital ?



- **Com o processamento digital de sinais é fácil:**
 - Alterar aplicações.
 - Corrigir aplicações.
 - Atualizar aplicações.
- **Adicionalmente o processamento digital de sinais reduz:**
 - Suscetibilidade ao Ruído.
 - Tempo de Desenvolvimento.
 - Custo.
 - Consumo de Potência.

AULA 1

1-3

Programação

- **1 Hardware = Muitas Tarefas**



- **Atualização e Flexibilidade**
 - DSP: Desenvolvimento de Novo Código → Atualização do software
 - Sistema Analógico → Soldar novos componentes

AULA 1

1-4

Restrições ao uso do Processamento Digital ?

- Sinais de alta frequência não podem ser processados digitalmente devido a duas razões:
 - Conversores Analógico-para-Digital (ADC) não podem operar rápido o suficiente.
 - A aplicação pode ser também muito complexa para operar em tempo real.

AULA 1 _____ 1-5 _____

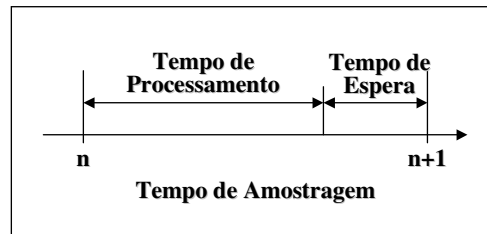
Processamento em Tempo Real

- Os DSPs precisam realizar tarefas em tempo real, portanto como podemos definir tempo real ?
- A definição de tempo real depende da aplicação.
- Exemplo: Um filtro digital FIR com 100 taps é implementado em tempo real se o DSP pode realizar e completar a seguinte operação entre duas amostras:

$$y(n) = \sum_{k=0}^{99} a(k)x(n-k)$$

AULA 1 _____ 1-6 _____

Processamento em Tempo Real



- Podemos dizer que temos uma aplicação em tempo real se:
 - Tempo de Espera ≥ 0

AULA 1 ————— 1-7

Porque Precisamos de DSPs ?

- Porque não utilizar um processador para aplicações gerais como um Pentium ao invés do DSP ?
 - Qual é o consumo de potência de um Pentium e de um DSP ?
 - Qual é o custo de um Pentium e de um DSP ?

AULA 1 ————— 1-8

Porque Precisamos de DSPs ?

- Use um DSP quando os seguintes requisitos forem necessários:
 - Redução de Custo.
 - Tamanho Reduzido.
 - Baixo consumo de potência.
 - Processamento de muitos sinais de alta frequência em tempo real.
- Use um processador genérico quando for necessário:
 - Grande memória.
 - Sistemas operacionais avançados.

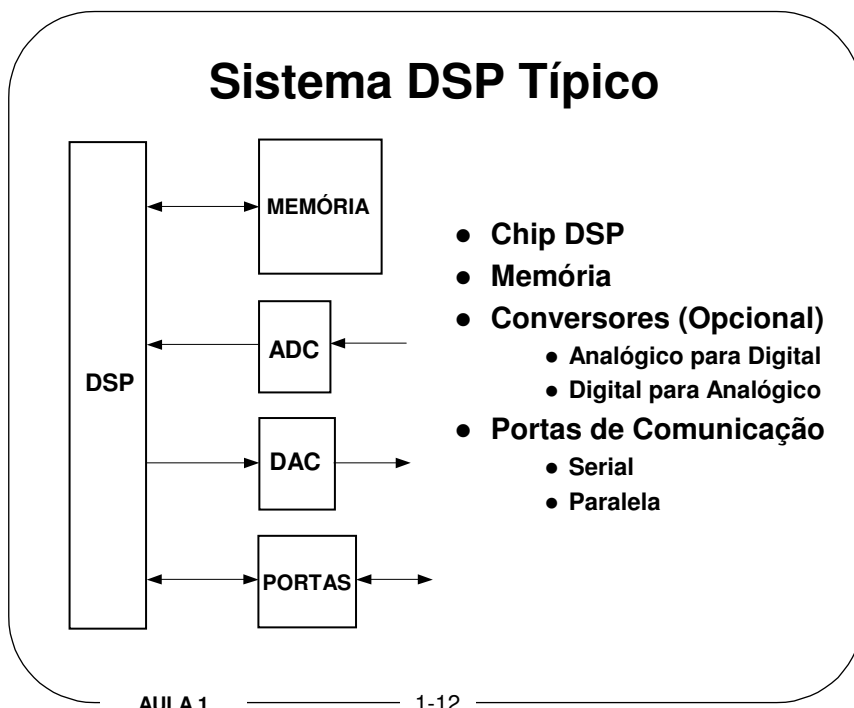
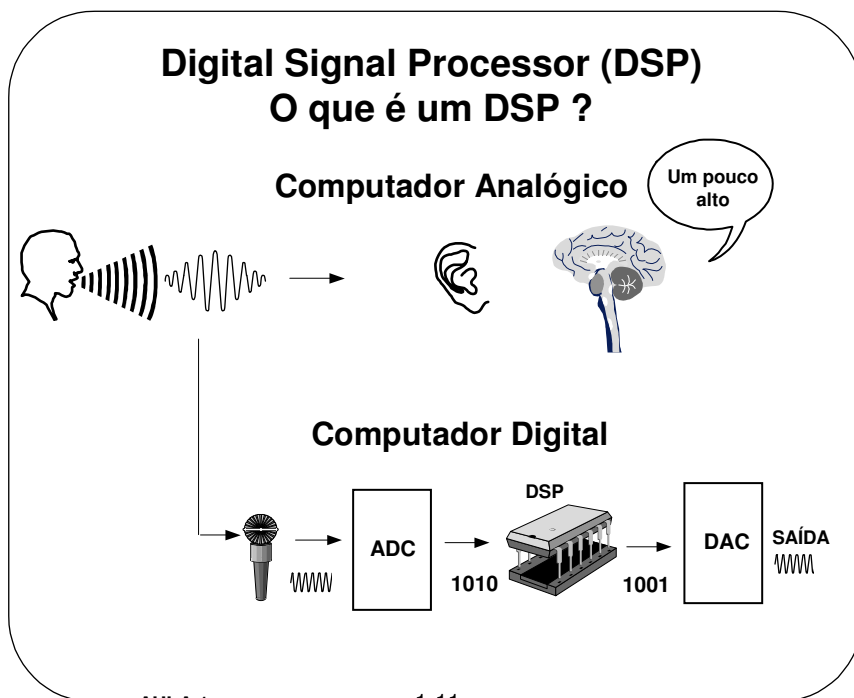
AULA 1 ————— 1-9

Quais são os algoritmos típicos de um DSP ?

- A soma de produtos é o elemento chave na maioria dos algoritmos DSP:

Algorithm	Equation
Finite Impulse Response Filter	$y(n) = \sum_{k=0}^M a_k x(n-k)$
Infinite Impulse Response Filter	$y(n) = \sum_{k=0}^M a_k x(n-k) + \sum_{k=1}^N b_k y(n-k)$
Convolution	$y(n) = \sum_{k=0}^N x(k)h(n-k)$
Discrete Fourier Transform	$X(k) = \sum_{n=0}^{N-1} x(n) \exp[-j(2\pi / N)nk]$
Discrete Cosine Transform	$F(u) = \sum_{x=0}^{N-1} c(u).f(x).\cos\left[\frac{\pi}{2N}u(2x+1)\right]$

AULA 1 ————— 1-10



Multiplicação e Adição

Adição $1+2 = 3$

```

    0001
  + 0010
  -----
    0011
    
```

Multiplicação $5*3 = 15$

0	x	8	x	0011	→	0000
1	x	4	x	0011	→	0011
0	x	2	x	0011	→	0000
1	x	1	x	0011	→	0011

↑ Deslocado e adicionado 5 vezes ↑ 3 =

Operações Mais Comuns no DSP

$$A = B * C + D$$

$$E = F * G + A$$

⋮

Multiplica, Adiciona, e Acumula
Instrução MAC

Operação MAC

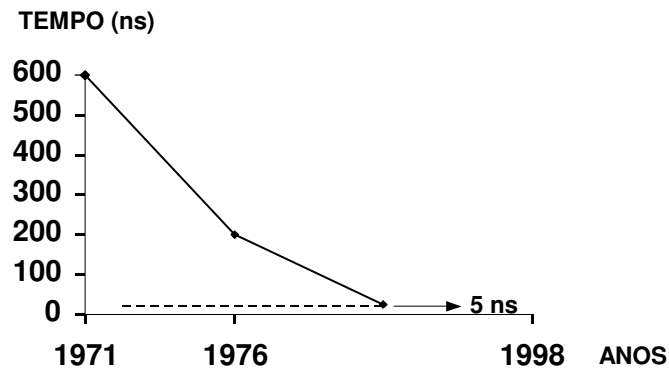
Tipicamente 70 ciclos de clock em processadores normais

Tipicamente 1 ciclo de clock em processadores digitais de sinais

AULA 1

1-13

Evolução nos Tempos de Multiplicação



AULA 1

1-14

Sistemas DSP Práticos

- Equipamentos Hi-Fi
- Brinquedos
- Videofones
- Modem/xDSL
- Sistemas Telefônicos
- Placas Gráficas 3D
- Processamento de Imagens
- Telefones Celulares
- Câmeras de Vídeo

AULA 1 ————— 1-15

Vantagens de Sistemas Analógicas

- **Baixo Custo e Simplicidade em Algumas Aplicações**
 - Atenuadores / Amplificadores
 - Filtros Simples
- Grande Largura de Banda (GHz)
- Sinais de Baixo Nível
- Taxa de Amostragem Efetiva Infinita
 - Infinita resolução em frequência
- Infinita Resolução em Amplitude
 - Sem ruído de quantização

AULA 1 ————— 1-16

Vantagens do Processamento de Sinais (DSP)

- **Invariabilidade**
 - Baixa sensibilidade para tolerâncias de componentes
 - Baixa sensibilidade para mudanças de temperatura
 - Baixa sensibilidade para efeitos de envelhecimento
 - Desempenho praticamente idêntico de unidade para unidade
 - Circuitos idênticos custam menos
- **Alta imunidade ao ruído**
- **Na maioria das aplicações o DSP oferece maior desempenho e baixo custo**

AULA 1 ————— 1-17

Circuitos Analógicos no Processamento Digital de Sinais

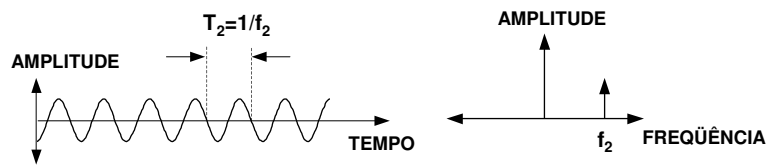
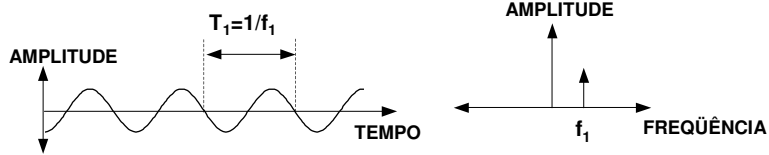
- **A maioria dos transdutores são analógicos por natureza**
 - Microfones, alto-falantes, etc.
- **Circuitos analógicos são necessários para pré-processar sinais de baixo nível antes do ADC**
- **Filtros analógicos podem ser usados para limitar a banda dos sinais**
 - Filtros *anti-aliasing* (antes do ADC) e filtros de reconstrução (depois do DAC)
- **Circuitos analógicos podem ser usados para acionar transdutores de saída**
 - Um amplificador de potência é necessário para habilitar o DAC à acionar um alto-falante

AULA 1 ————— 1-18

Sinais no Domínio do Tempo e da Frequência

Domínio do Tempo

Domínio da Frequência

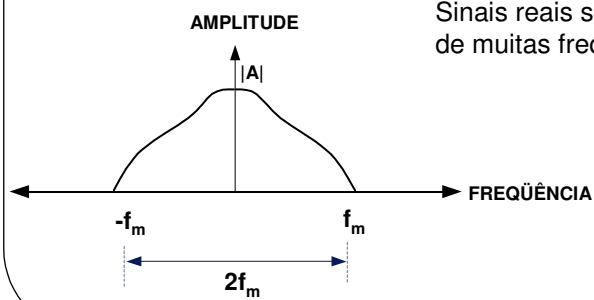
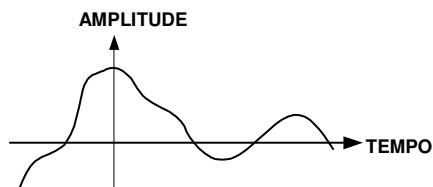


T = período f = frequência

AULA 1

1-19

Sinais Reais



Sinais reais são uma combinação de muitas frequências

AULA 1

1-20

Amostragem

- Pegar amostras de dados (sinais) com variação contínua
- O Período de amostragem é fixo
- This makes information understandable
- My share price hit its lowest point in week 4
- My share price reached its peak in week 9

→ ← **Período de Amostragem** – Período entre amostras
 → ← **Tempo de Amostragem** – Tempo para pegar uma amostra

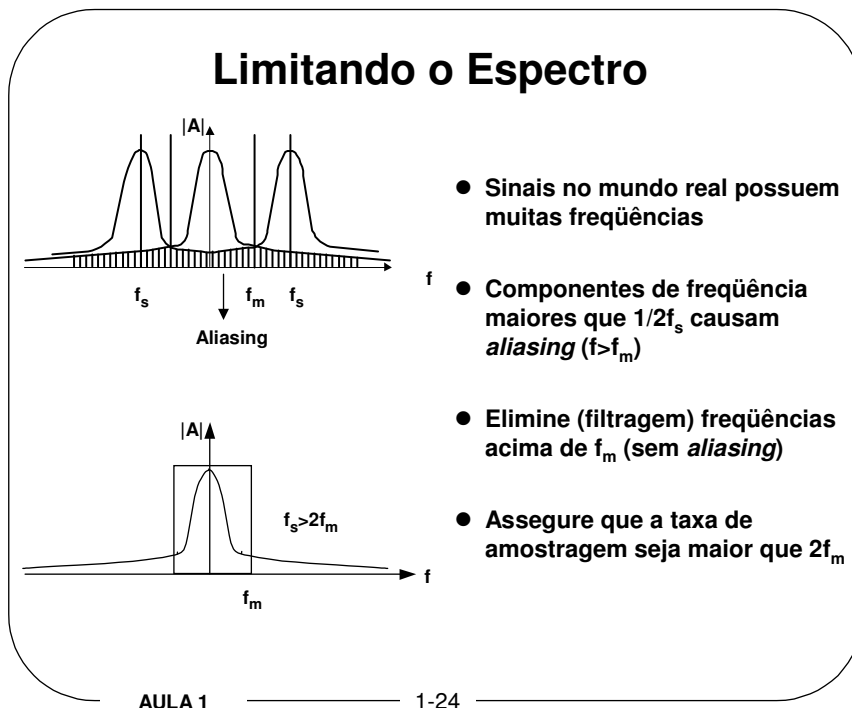
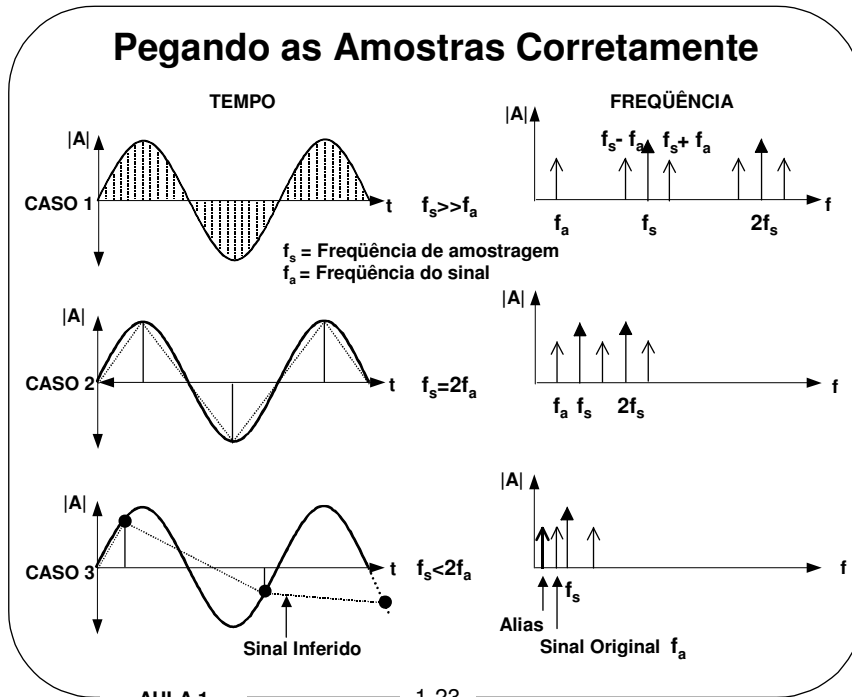
AULA 1 1-21

Informação Perdida

- **Amostras Não-Periódicas**
 - Pode perder informação
 - A queda de preços entre T3 e T4 fica imperceptível
 - A informação não pode ser interpretada facilmente
- **Amostragem Periódica**
 - Pode perder informação
 - A queda de preços entre T2 e T3 fica imperceptível
 - Fácil para interpretar

A chave é a frequência de amostragem.

AULA 1 1-22



Digitalizando o Sinal

- O objetivo é obter uma representação discreta
- Amostre o sinal periodicamente
- Retenha o valor amostrado até a próxima amostra
- Quantize o sinal
- Mais níveis de quantização significam melhor precisão

Sinal = 10 10 01 01 00 00 01

AULA 1 1-25

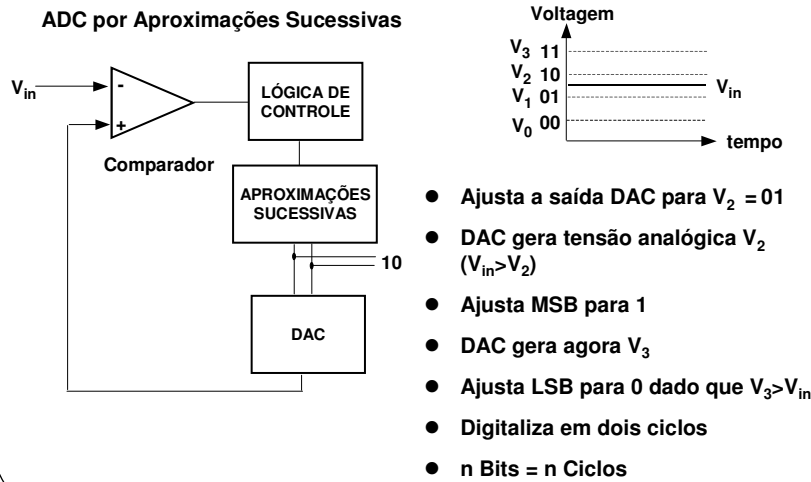
Erro de Quantização

- A quantização introduz erros
- Aumentar o número de níveis não é sempre a solução
- Quantização Não-Uniforme
 - Utiliza mais níveis onde existe maior variação
 - Utiliza menos níveis onde há menos variações

AULA 1 1-26

Conversor Analógico-Digital (ADC)

ADC por Aproximações Sucessivas

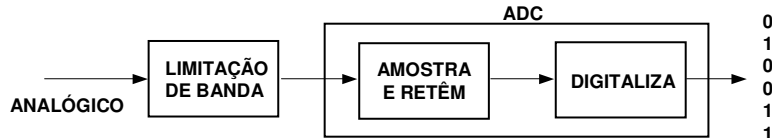


AULA 1

1-27

Outros Tipos de ADC

Digitalização Paralela



Dual Slope ADC

- Lento
- Caro

Flash ADC

- Requer componentes de precisão

Sigma Delta ADC

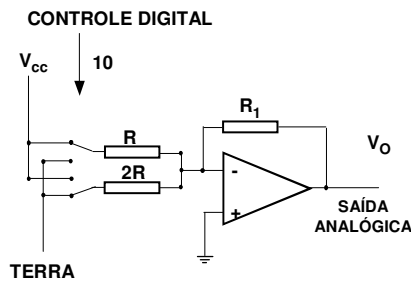
- Estável

AULA 1

1-28

Conversão Digital para o Analógico

“Voltage Source Multiplying DAC”



- Para Controle Digital 10:
Chavear em R para fonte (V_{cc})
2R para terra (GND)
Saída Analógica = $(R1/R) * V_{cc}$

- Ganho = $R1/\text{Resistência de Entrada}$
TENSÃO DE SAÍDA

$$V_O = - [V_{in} * (R_1/R) + V_{in} * (R_1/2R)]$$

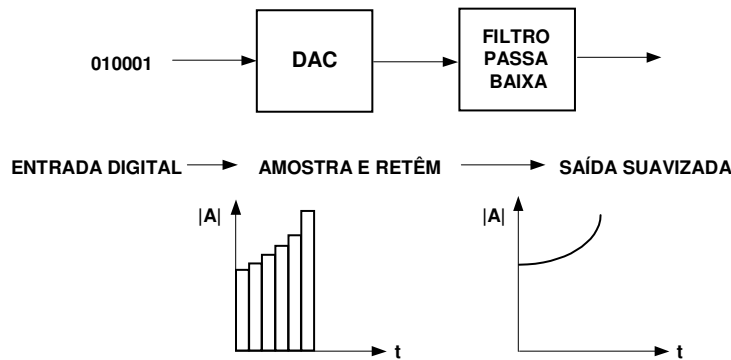
MSB = Bit mais significativo
LSB = Bit menos significativo

- Possíveis Entradas e Saídas para $R1 = R$

ENTRADA	SAÍDA
11	$1.5 V_{cc}$
10	V_{cc}
01	$0.5 V_{cc}$
00	0

AULA 1 1-29

Suavizando a Saída

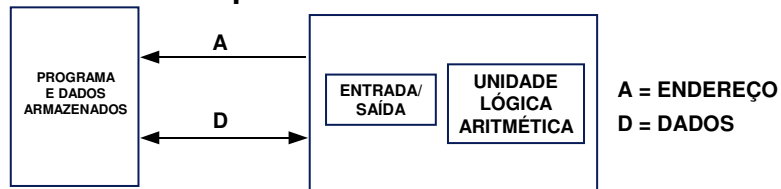


- Converta a entrada digital para valor analógico
- Mantenha o valor analógico até a próxima entrada digital
- Suavize o sinal de saída

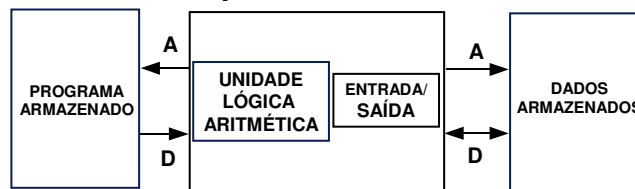
AULA 1 1-30

Computadores Digitais

Arquitetura Von Neuman



Arquitetura Harvard



AULA 1 ————— 1-31

Sistemas de Numeração

- Representação digital de números

Decimal	128	64	32	16	8	4	2	1
$2^{\text{Digit Number}}$	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Digit Number	7	6	5	4	3	2	1	0

- Qualquer número pode ser representado como uma série de 1s e 0s

- Decimal 3 em binário

Decimal	0	0	0	0	0	0	2+	1=	3
$2^{\text{Digit Number}}$	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Digit Number	7	6	5	4	3	2	1	0	
Binary	0	0	0	0	0	0	1	1	0000 0011

- Decimal 26 em binário

Decimal	0	0	0	16+	8+	0	2=	0	26
$2^{\text{Digit Number}}$	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Digit Number	7	6	5	4	3	2	1	0	
Binary	0	0	0	1	1	0	1	0	0001 1010

AULA 1 ————— 1-32

Binário e Hexadecimal

Decimal 0, 1, 2 ,, 9 16 Decimal → 0x10 Hex
Binário 0, 1 20 Decimal → 0x14 Hex
Hexadecimal 0, 1, 2 ,, A, B, C, D, E, F

- 4 bits do sistema binário são representados por um único dígito hexadecimal:

Decimal	8+	4+	2+	1=	15
2 ^{Digit Number}	2 ³	2 ²	2 ¹	2 ⁰	
Binary	1	1	1	1	1111
Hex	F				F

- Decimal 26 em binário e hexadecimal:

Decimal	0	0	0	16+	8+	0	2=	0	26
2 ^{Digit Number}	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
Digit Number	7	6	5	4	3	2	1	0	
Binary	0	0	0	1	1	0	1	0	0001 1010
Hex	1			A				1A	

Representação de Números Inteiros com Sinal

- Magnitude de números inteiros com sinal

Decimal	Signed		
	Hex	Binary	
		Sign	Number
2	00 00 00 02	0	000 0000 0000 0000 0000 0000 0010
3	00 00 00 03	0	000 0000 0000 0000 0000 0000 0011
-2	80 00 00 02	1	000 0000 0000 0000 0000 0000 0010
-3	80 00 00 03	1	000 0000 0000 0000 0000 0000 0011

Notação *Complemento de Dois* para Representação de Números Inteiros com Sinal

$2^{\text{Digit number}}$		-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Decimal		-128	64	32	16	8	4	2	1	
Binary two's complement	3	0	0	0	0	0	0	1	1	
Decimal calculation		0	0	0	0	0	0	+2	+1=	3
Binary two's complement	-2	1	1	1	1	1	1	1	0	
Decimal calculation		-128	+64	+32	+16	+8	+4	+2	+0=	-2

AULA 1 ————— 1-35

Notação *Complemento de Dois*

- Conversão para a notação *complemento de dois*

Action	Decimal	Binary							
		Sign	Number						
Signed integer	-2	1	000	0000	0000	0000	0000	0000	0010
Strip sign bit		0	000	0000	0000	0000	0000	0000	0010
Invert		1	111	1111	1111	1111	1111	1111	1101
Add one		1							1
Two's complement		1	111	1111	1111	1111	1111	1111	1110
	Hex		F	F	F	F	F	F	E

- Adição usando a notação *complemento de dois*

Decimal	Hex	Two's Complement Binary	
		Sign	Number
-2	FF FF FF FE	1	111 1111 1111 1111 1111 1111 1110
3	00 00 00 03	0	000 0000 0000 0000 0000 0000 0011
$-2 + 3 = 1$	00 00 00 01	0	000 0000 0000 0000 0000 0000 0001

AULA 1 ————— 1-36

Notação de Ponto-Fixo

Convenções:

- Faixa do número está entre 1 e -1
- O ponto decimal está sempre numa localização fixa (ex: 0.74, 0.34, etc.)
- Multiplicando uma fração por uma fração sempre resulta em uma fração e não irá produzir um *overflow* (ex: 0.99 x 0.9999 = menor que 1)
- Adições sucessivas podem causar *overflow*

Porque ?

- O processamento de sinais é intensivo em multiplicações
- A notação de ponto-fixo previne *overflow* (útil com uma pequena faixa dinâmica de variação)
- A notação de ponto-fixo é mais barata

Como a notação de ponto-fixo é realizada no DSP ?

- A maioria dos DSPs de ponto-fixo são de 16 bits
- A faixa de números que podem ser representados é de 32767 até -32768
- O formato mais comum de ponto-fixo é denominado Q15

Notação Q15

Bit 15	Bits 14 to 0
sign	two's complement number

Formato Q15

Faixa Dinâmica no Formato Q15

Number	Biggest	Smallest
Fractional number	0.999	-1.000
Scaled integer for Q15	32767	-32768

Representação do Número no Formato Q15

Decimal	Q15 = Decimal x 2 ¹⁵	Q15 Integer
0.5	0.5 x 32767	16384
0.05	0.05 x 32767	1638
0.0012	0.0012 x 32767	39

Regras de operação

- Evite operações com números maiores que 1
 $2.0 \times (0.5 \times 0.45) = (0.2 \times 0.5 \times 0.45) \times 10 = (0.5 \times 0.45) + (0.5 \times 0.45)$
- Escale os números antes da operação

$0.5 \text{ in Q15} = 0.5 \times 32767 = 16384$

Operações no Formato Q15

Adição

Decimal	Q15	Scale back Q15 / 32767
$0.5 + 0.05 = 0.55$	$16384 + 1638 = 18022$	0.55
$0.5 - 0.05 = 0.45$	$16384 - 1638 =$	

Multiplicação

$2 \times 0.5 \times 0.45 =$

Decimal	Q15	Back to Q15 Product / 32767	Scale back Q15 / 32767
$0.5 \times 0.45 = 0.225$	$16384 \times 14745 = 241584537$	7373	
$0.225 + 0.225 = 0.45$	$7373 + 7373 = 14746$		0.45

AULA 1 1-39

Formato de Ponto Flutuante do TMS

Formato de Ponto Flutuante

31	...	24	23	22	0	← No. do Bit
e			s	f			
8 bits			1 bit	23 bits			

e = o expoente é um número de 8 bits com sinal na representação complemento dois e determina a localização do ponto binário Q

s = sinal da mantissa (s = 0 positivo, s = 1 negativo)

f = parte fracionária da mantissa; um valor 1.0 é adicionado a esta fração mas não é alocado no campo de bit pois este valor está sempre presente

Equações de Conversão

	Binary	Decimal	Equation
s = 0	$X = 01.f \times 2^e$	$X = 01.f \times 2^e$	1
s = 1	$X = 10.f \times 2^e$	$X = (-2 + 0.f) \times 2^e$	2

Caso Especial

s = 0	X = 0	e = -128
-------	-------	----------

	Expoente (e)				
	Decimal	0	1	127	-1
Hex two's comp.	00	01	7F	FF	80

AULA 1 1-40

Números Ponto-Flutuante

Calcula 1.0e0

Em hex 00 00 00 00
 Em binária 0000 0000 0000 0000 0000 0000 0000 0000
 s = 0 → Equação 1: $X = 01.f \times 2^e$

$$\boxed{f = 0} \downarrow$$

$$01.0 \times 2^0 \leftarrow \boxed{e = 0} = 1.0$$

Calcula 1.5e01

Em hex 03 70 00 00
 Em binária 0011 0111 0000 0000 0000 0000 0000 0000
 s = 0 → Equação 1: $X = 01.f \times 2^e$

0011 → e = 3
 s111 ... → f = 0.5 + 0.25 + 0.125 = 0.875

$$X = 01.875 \times 2^3 = 15.0 \text{ decimal}$$

AULA 1 1-41

Ponto Flutuante

Calcula -2.0e0

Em hex 00 80 00 00
 Em binário 0000 0000 1000 0000 0000 0000 0000 0000
 s = 1 → Equação 2: $X = (-2.0 + 0.f) \times 2^e$

$$\boxed{f = 0} \downarrow$$

$$(-2.0 + 0.0) \times 2^0 \leftarrow \boxed{e = 0} = -2.0$$

Adição

$$1.5 + (-2.0) = 0.5$$

Multiplicação

$$1.5e00 \times 1.5e01 = 2.25e01 = 22.5$$

AULA 1 1-42

Faixa Dinâmica

Faixas dos Sistemas de Numeração

Numbers	Base 2	Decimal	Two's Complement Hex
Largest Integer	$2^{31} - 1$	2 147 483 647	7F FF FF FF
Smallest Integer	-2^{31}	-2 147 483 648	80 00 00 00
Largest Q15	$2^{15} - 1$	32 767	7F FF
Smallest Q15	-2^{15}	-32 768	80 00
Largest Floating Point	$(2 - 2^{-23}) \times 2^{127}$	3.402823×10^{38}	7F 7F FF FD
Smallest Floating Point	-2×2^{127}	-3.402823×10^{38}	83 39 44 6E

- A faixa dinâmica da representação de ponto flutuante é muito ampla
- Conclusão
 - Maior inteiro $\times (1.5 \times 10^{29}) \sim =$ maior ponto flutuante
 - Maior Q15 $\times (1.03 \times 10^{34}) \sim =$ maior ponto flutuante

AULA 1 ————— 1-43

Ponto Fixo versus Flutuante

- Dispositivos DSP são projetados para ponto fixo ou ponto flutuante
- Dispositivos ponto-fixo são usualmente 16-bits, Ex: TMS320C5x
- Dispositivos ponto-flutuante são usualmente 32-bits, Ex: TMS320C3x
- Dispositivos ponto-flutuante possuem usualmente um conjunto completo de instruções ponto-fixo
- Dispositivos ponto flutuante são mais fáceis de programar
- Dispositivos ponto-fixo podem emular operação ponto flutuante em software

Comparação

Characteristic	Floating point	Fixed point
Dynamic range	much larger	smaller
Resolution	comparable	comparable
Speed	comparable	comparable
Ease of programming	much easier	more difficult
Compiler efficiency	more efficient	less efficient
Power consumption	comparable	comparable
Chip cost	comparable	comparable
System cost	comparable	comparable
Design cost	less	more
Time to market	faster	slower

AULA 1 ————— 1-44

Família TMS320

Dispositivos 16-Bit Ponto Fixo

'C1x Controladores de *Hard-Disk*

'C2x Fax

'C2xx Controle

'C5x Processamento de Voz

'C54x Telefones Celulares Digitais

Dispositivos 32-Bit Ponto Flutuante

'C3x Videofones

'C4x Processamento Paralelo

Outros Dispositivos

'C6x **Processador Avançado**
Estações Base *Wireless*
Modems

'C8x Vídeo Conferência

2. Digitalização de Sinais e Normalização de Freqüências

Considere a digitalização de um sinal analógico, $x_c(t)$, usando uma freqüência de amostragem $F_s = 1/T_s$ Hz (amostras/s). A seqüência amostrada (vetor de amostras) é representada por

$$x[n] = x_c(nT_s)$$

onde $t = nT_s = \frac{n}{F_s}$ são os instantes de amostragem do sinal analógico. Como exemplo considere a digitalização de um coseno com freqüência, F ,

$$x_c(t) = A \cdot \cos(2\pi F t + \theta) = A \cdot \cos(\Omega t + \theta)$$

$$F \text{ (Hz)} \qquad \Omega = 2\pi F \text{ (rad/s)}$$

$$x[n] = x_c(nT_s) = A \cdot \cos(2\pi F nT_s + \theta) = A \cdot \cos\left(\frac{2\pi F n}{F_s} + \theta\right)$$

$$x[n] = A \cdot \cos(2\pi f + \theta) \qquad w = \Omega T_s = 2\pi f$$

$$f \text{ (ciclos/amostra)} \qquad w \text{ (rad/amostra)}$$

$\Omega, F \Rightarrow$ Notação de freqüência para sinais analógicos

$w, f \Rightarrow$ Notação de freqüência para sinais em tempo discreto (amostrados)

$$f = \frac{F}{F_s} \qquad w = \Omega T_s = 2\pi f \qquad \Rightarrow \text{Normalização de freqüências}$$

Considerando a digitalização na taxa de Nyquist, $F_s = 2F_{MAX}$, onde $F_{MAX} = F$, teremos os seguintes valores de freqüência normalizada:

$$f = \frac{F}{2F} = \frac{1}{2} \qquad w = 2\pi f = \pi \qquad \Rightarrow \text{Normalização de freqüências}$$

Portanto, a faixa de variação de valores para as freqüências do sinal em *tempo contínuo* e *tempo discreto* são:

$$-\infty < F < \infty \Leftrightarrow -1/2 < f < 1/2$$

$$-\infty < \Omega < \infty \Leftrightarrow -\pi < w < \pi$$

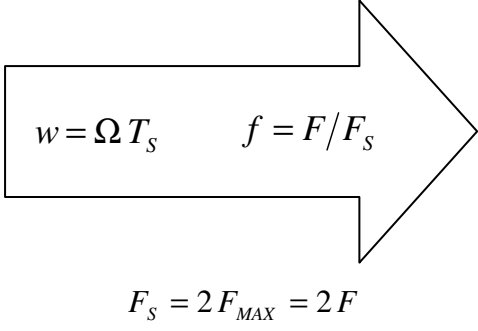
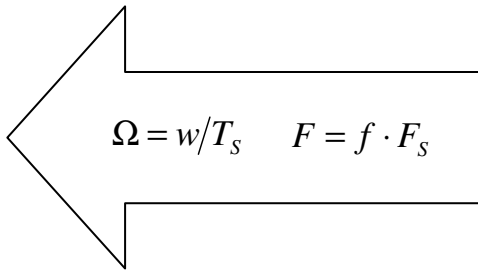
Relação entre Frequências do Sinal Contínuo e Discreto (Normalização) Interpretação Considerando a Digitalização de Sinais Senoidais		
Sinais em Tempo Contínuo		Sinais em Tempo Discreto
$\Omega = 2\pi F \text{ (rad/s)}$		$w = 2\pi f \text{ (rad/amostra)}$
$F \text{ (Hz)}$		$f \text{ (ciclos/amostra)}$
$-\infty < F < \infty$ $-\infty < \Omega < \infty$		$-1/2 < f < 1/2$ $-\pi < w < \pi$
$-\frac{F_s}{2} \leq F \leq \frac{F_s}{2}$ $-\frac{\pi}{T_s} \leq \Omega \leq \frac{\pi}{T_s}$		$-1/2 < f < 1/2$ $-\pi < w < \pi$
	$F_{MAX} = \frac{F_s}{2} = \frac{1}{2T_s}$ $\Omega_{MAX} = \pi F_s = \frac{\pi}{T_s}$	

TABELA 1

EXEMPLO 2.1 – ANÁLISE DO DO EFEITO DE ALIASING

$$x_1(t) = \cos(2\pi 10t) \quad F_1 = F_{MAX} = 10 \text{ Hz} \quad F_s = 40 \text{ Hz}$$

$$x_2(t) = \cos(2\pi 50t) \quad F_2 = F_{MAX} = 50 \text{ Hz} \quad F_s = 40 \text{ Hz}$$

$$x_1[n] = \cos\left[2\pi\left(\frac{10}{40}\right)n\right] = \cos\left(\frac{\pi}{2}n\right)$$

$$x_2[n] = \cos\left[2\pi\left(\frac{50}{40}\right)n\right] = \cos\left(\frac{5\pi}{2}n\right) = \cos\left(2\pi n + \frac{\pi}{2}n\right) = \cos\left(\frac{\pi}{2}n\right)$$

$$x_1[n] = x_2[n] \Rightarrow \text{ALIASING (indistinguível)}$$

$$F_2 \underset{\text{alias}}{\Rightarrow} F_1 \quad (F_1 + F_s \cdot k) \underset{\text{alias}}{\Rightarrow} F_1 \quad k = 1, 2, 3, \dots \text{ (inteiro)}$$

$$F_k = F_0 + k \cdot F_s \quad -\infty < k < \infty \quad \text{Frequências de alias}$$

EXERCÍCIOS

- 1) $x_c(t) = \text{sen}(2\pi 100t)$ $F_S = 400\text{Hz}$ Determine $x[n]$
- 2) $x[n] = \cos\left(\frac{\pi}{4}n\right)$ $F_S = 1000\text{Hz}$ Determine Ω dado que $x_c(t) = \cos(\Omega t)$
- 3) $x_c(t) = \cos(4000\pi t)$ $x[n] = \cos\left(\frac{\pi}{3}n\right)$ Determine F_S
- 4) $x_c(t) = \text{sen}(20\pi t) + \cos(40\pi t)$ $x[n] = \text{sen}\left(\frac{\pi}{5}n\right) + \cos\left(\frac{2\pi}{5}n\right)$ Determine F_S
- 5) Considere o sinal analógico $x_c(t) = 3 \cdot \cos(100\pi t)$
 - a) Determine a frequência de amostragem mínima para evitar aliasing
 - b) Sendo $F_S = 200\text{Hz}$, determine o sinal $x[n]$
 - c) Sendo $F_S = 75\text{Hz}$, determine o sinal $x[n]$

3. Periodicidade de Seqüências Discretas

Características dos sinais senoidais em tempo discreto:

a) Um sinal senoidal em tempo discreto é periódico, somente se a sua frequência normalizada, f , for um número racional (razão de inteiros)

$$x[n] = x[n + N] \Rightarrow \text{Condição de periodicidade para um sinal de período } N$$

Para um sinal senoidal (em tempo discreto) de frequência, f , ser periódico, tem que ser satisfeita a seguinte relação

$$x[n] = \cos(2\pi f n) = \cos(2\pi f (n + N)n)$$

$$x[n] = \cos(2\pi f n) = \cos(2\pi f n + 2\pi f N)$$

$$\cos(\theta) = \cos(\theta + 2\pi)$$

O termo, $2\pi f N$, deve ser múltiplo inteiro de 2π , ou seja,

$$2\pi f N = 2\pi \cdot k, \quad k \text{ inteiro}$$

$$f = \frac{k}{N} \quad f = \frac{w}{2\pi} = \frac{k}{N} \quad w = \frac{2\pi k}{N}$$

Podemos interpretar a relação $w = \frac{2\pi k}{N}$ da seguinte forma: existem N frequências distintas para as quais as seqüências correspondentes são periódicas com período N .

b) Sinais senoidais discretos, cujas frequências são separadas por um múltiplo inteiro de 2π , são idênticos

$$\cos((w + 2\pi)n + \theta) = \cos(wn + \theta)$$

EXEMPLO 3.1

$$x_1[n] = \cos\left(\frac{\pi}{4} n\right) \quad f = \frac{1}{8} \quad k = 1 \quad w = \frac{2\pi k}{8} = 2\pi \frac{1}{8}$$

$$w_k = \frac{2\pi k}{8} \quad k = 0, 1, 2, 3, 4, 5, 6, 7$$

Contudo existem 8 sinais com frequências diferentes, w_k , e com o mesmo período $N=8$:

$$\begin{array}{cccc} w_0 = 0 & w_1 = \frac{\pi}{4} & w_2 = \frac{\pi}{2} & w_3 = \frac{3\pi}{4} \\ w_4 = \pi & w_5 = \frac{5\pi}{4} & w_6 = \frac{3\pi}{2} & w_7 = \frac{7\pi}{4} \end{array}$$

4. Interpretação da Frequência Normalizada

A frequência normalizada, $f = \frac{k}{N}$, indica em quantos períodos do sinal analógico, k , estão contidos num período, N , do sinal em tempo discreto (amostrado).

EXEMPLO 4.1

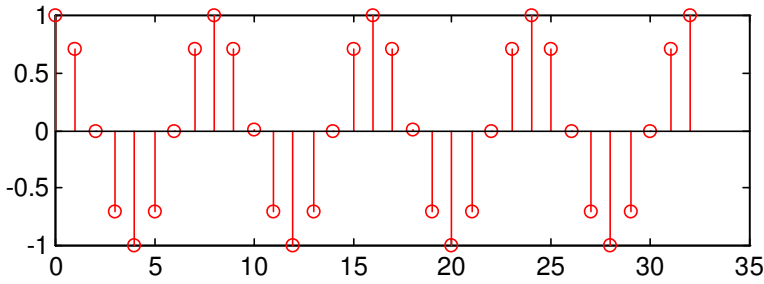
A seguir são apresentadas três seqüências discretas amostradas com frequências distintas, que possuem a mesma frequência normalizada:

$$x_1(t) = 1,5 \cdot \cos\left(2\pi \frac{3}{8}t\right) \quad F_s = 1 \quad x_1[n] = 1,5 \cdot \cos\left(2\pi \frac{3}{8}n\right)$$

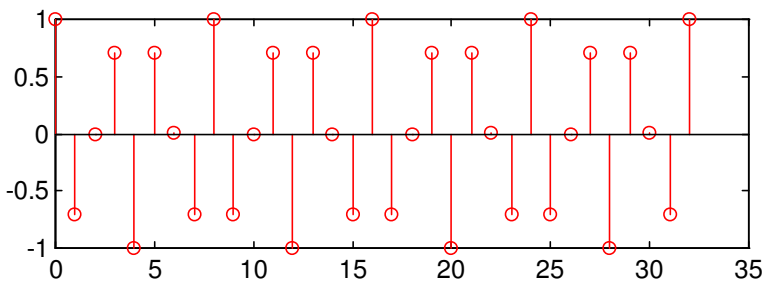
$$x_2(t) = 1,5 \cdot \cos\left(2\pi \frac{3}{16}t\right) \quad F_s = \frac{1}{2} \quad x_2[n] = 1,5 \cdot \cos\left(2\pi \frac{3}{8}n\right)$$

$$x_3(t) = 1,5 \cdot \cos\left(2\pi \frac{3}{4}t\right) \quad F_s = 2 \quad x_3[n] = 1,5 \cdot \cos\left(2\pi \frac{3}{8}n\right)$$

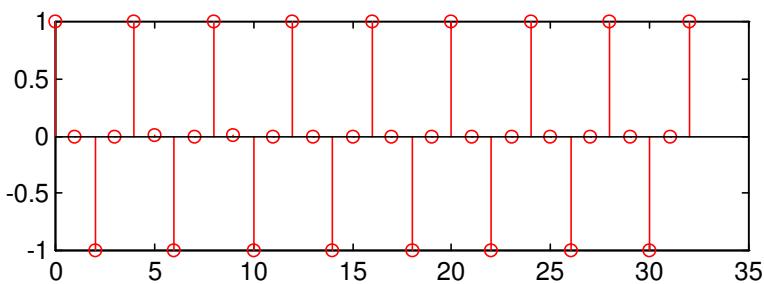
5. Sinais de Freqüências Distintas com o mesmo Período



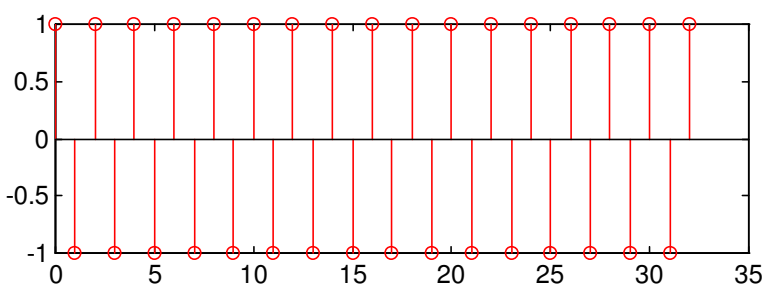
$$w_0 = 2\pi \frac{1}{8}$$



$$w_0 = 2\pi \frac{2}{8}$$



$$w_0 = 2\pi \frac{3}{8}$$



$$w_0 = 2\pi \frac{4}{8}$$

6. Interpretação de Frequências Altas e Baixas

Para sinais contínuos, quando aumentamos a frequência, aumentamos a oscilação.

$$x(t) = \cos(\Omega_0 t + \theta) \quad \uparrow \Omega_0 \quad \uparrow \text{OSCILAÇÃO}$$

Para sinais discretos, quando aumentamos a frequência de 0 para π , aumentamos a oscilação. Aumentando a frequência de π para 2π , a oscilação diminui.

$$x[n] = A \cdot \cos(w_0 n + \theta)$$

$$0 \leq w_0 \leq \pi \quad \uparrow w_0 \quad \uparrow \text{OSCILAÇÃO}$$

$$\pi \leq w_0 \leq 2\pi \quad \uparrow w_0 \quad \downarrow \text{OSCILAÇÃO}$$

Isso ocorre porque $w_0 = 0$ é indistinguível de $w_0 = 2\pi$, ou seja, frequências em torno de $w_0 = 0$ são indistinguíveis de frequências em torno de $w_0 = 2\pi$.

$$\text{Frequências Baixas} \quad \Rightarrow \quad w_0 = 2\pi k$$

$$\text{Frequências Altas} \quad \Rightarrow \quad w_0 = \pi + 2\pi k$$

Exemplo

$$w_0 = 0 = 2\pi$$

$$w_0 = \frac{\pi}{8} = 2\pi - \frac{\pi}{8} = \frac{15\pi}{8}$$

$$w_0 = \frac{\pi}{4} = 2\pi - \frac{\pi}{4} = \frac{7\pi}{4}$$

Exemplos de Sinais Periódicos:

$$x_1[n] = \cos\left(\frac{\pi}{4}n\right) \quad N = 8$$

$$x_2[n] = \cos\left(\frac{3\pi}{8}n\right) \quad N = 16$$

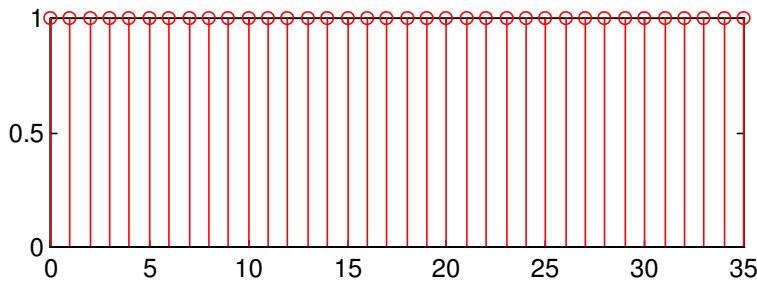
Neste caso aumentando a frequência de $w_0 = 2\pi/8$ para $w_0 = 3\pi/8$, o período também aumentou de $N = 8$ para $N = 16$. No caso de sinais contínuos, aumentando a frequência, o período diminui.

Exemplo de Sinal não Periódico:

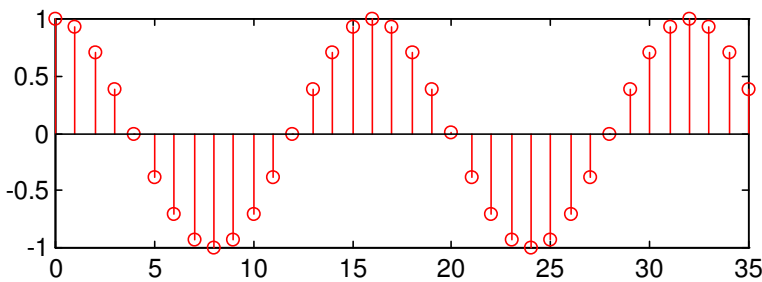
$$x_3[n] = \cos(n) \quad 1 = \frac{2\pi k}{N}$$

Não existe k inteiro tal que N seja inteiro.

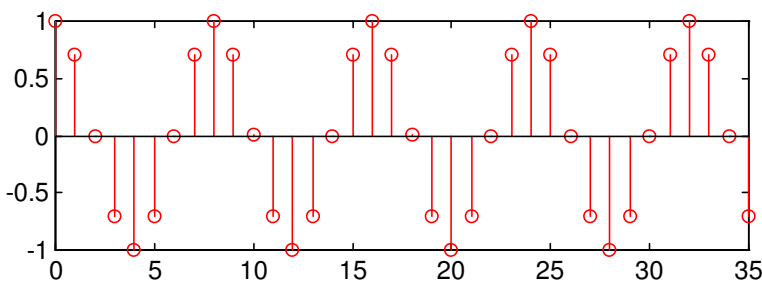
7. Interpretação de Frequências Altas e Baixas em Tempo Discreto



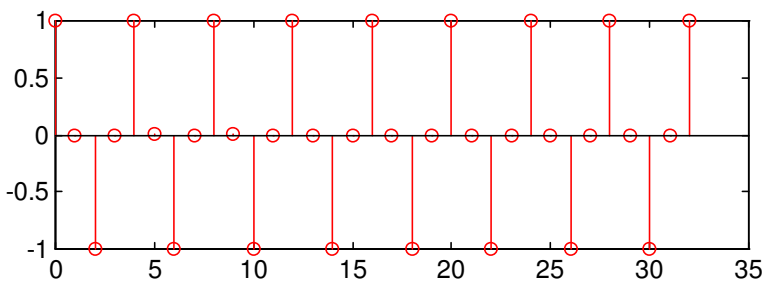
$$w_0 = 0 = 2\pi$$



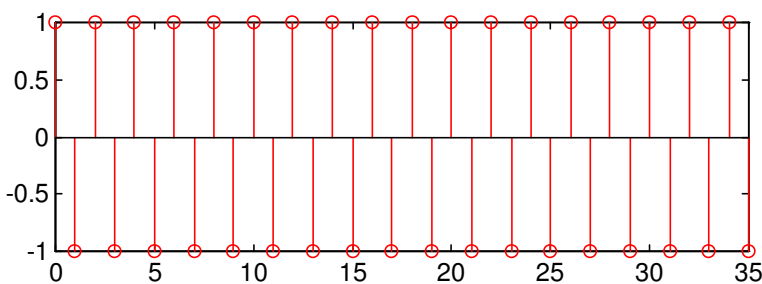
$$w_0 = \frac{\pi}{8}$$



$$w_0 = \frac{\pi}{4}$$



$$w_0 = \frac{\pi}{2}$$



$$w_0 = \pi$$

8. Operações com Seqüências Discretas

$$x[n] = \{\dots, x[-1], x[0], x[1], \dots\} \quad -\infty < n < \infty$$

$$x_1[n] = \{\dots, x_1[-1], x_1[0], x_1[1], \dots\}$$

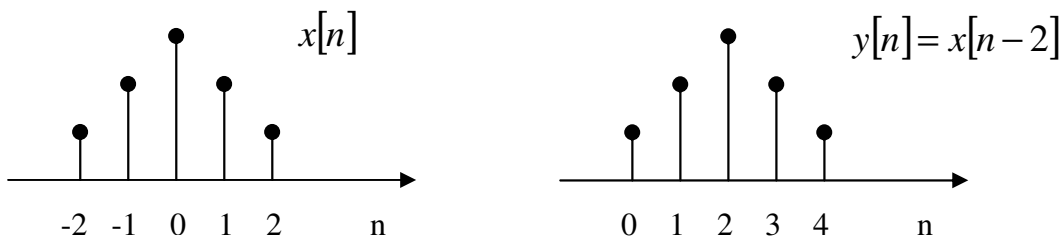
$$x_2[n] = \{\dots, x_2[-1], x_2[0], x_2[1], \dots\}$$

Multiplicação por escalar: $y[n] = \alpha \cdot x_1[n] = \{\dots, \alpha x_1[-1], \alpha x_1[0], \alpha x_1[1], \dots\}$

Multiplicação de seqüências: $y[n] = x_1[n] \cdot x_2[n] = \{\dots, x_1[-1] \cdot x_2[-1], x_1[0] \cdot x_2[0], x_1[1] \cdot x_2[1], \dots\}$

Adição de seqüências: $y[n] = x_1[n] + x_2[n] = \{\dots, x_1[-1] + x_2[-1], x_1[0] + x_2[0], x_1[1] + x_2[1], \dots\}$

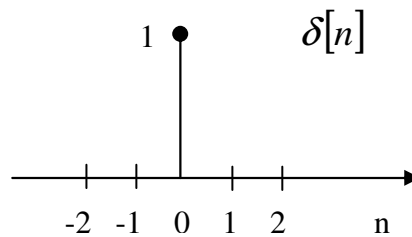
Deslocamento no Tempo: $y[n] = x[n - n_0]$, n_0 inteiro



9. Função Impulso Unitário

A função impulso em tempo discreto desempenha o mesmo papel da função “delta de Dirac” para sinais contínuos:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



Qualquer seqüência discreta, $x[n]$, pode ser escrita como um somatório de funções impulsos:

$$x[n] = \sum_{k=-\infty}^{+\infty} x[k] \cdot \delta[n - k]$$

10. Função Degrau Unitário

$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad u[n] = \sum_{k=0}^{+\infty} \delta[n - k] \quad \delta[n] = u[n] - u[n - 1]$$

PRÉ-LAB 1 – REPRESENTAÇÃO E OPERAÇÕES COM SEQUENCIAS DISCRETAS

- Leitura dos Capítulos 1 e 2
- Fazer o download do toolbox de funções do livro (trazer em disquete)

Link:

[HTTP://WWW.BROOKSCOLE.COM/CGI-WADSWORTH/COURSE_PRODUCTS_WP.PL?FID=M20B&PRODUCT_ISBN_ISSN=0534371744&DISCIPLINE_NUMBER=9](http://www.brookscoble.com/cgi-wadsworth/course_products_wp.pl?fid=M20B&product_isbn_issn=0534371744&discipline_number=9)

- Leitura do Capítulo 2 pg. 8-20
- Trazer uma página, escrita a mão, descrevendo as funções para geração e operação com seqüências discretas, com os respectivos parâmetros de passagem e retorno.

EXERCÍCIO 3 – Geração de seqüências exponenciais complexas

$$x[n] = e^{(\sigma + j\omega)n}, \forall n \quad \omega = \text{freqüência} \quad \sigma = \text{fator de atenuação}$$

Utilizando o programa em anexo, trace os gráficos das seqüências exponenciais complexas, para os seguintes parâmetros:

- $\sigma = 0.5 \quad \omega = \pi/6 \quad N = 50$
- $\sigma = 1 \quad \omega = \pi/4 \quad N = 100$
- $\sigma = -0.5 \quad \omega = \pi/4 \quad N = 30$
- $\sigma = 0 \quad \omega = \pi/6 \quad N = 60$

Como as seqüências são complexas (amostras complexas) devemos visualizar a parte real e imaginária ou as componentes em módulo e fase. Na análise espectral de sinais usando a Transformada Discreta de Fourier (DFT/FFT) trabalharemos com amostras complexas.

```
% Geração de uma seqüência exponencial complexa
clear; clc;
sigma=input('Digite o valor de sigma:');
w=input('Digite a frequencia: ');
N= input('Digite o comprimento da sequencia: ');
n=0:(N-1); c=sigma+w*i; x=exp(c*n);
figure(1)
subplot(2,1,1);
stem(n,real(x));
xlabel('n');
ylabel('Amplitude');
title('Parte Real');
subplot(2,1,2);
stem(n,imag(x));
xlabel('n');
ylabel('Amplitude');
title('Parte Imaginária');
figure(2)
subplot(2,1,1);
stem(n,abs(x));
xlabel('n');
ylabel('Módulo');
subplot(2,1,2);
stem(n,angle(x));
xlabel('n'); ylabel('Fase');
```

EXERCÍCIO 4 – Implemente um programa para gerar seqüências, $x[n] = \cos(\omega_0 n)$, para as seguintes freqüências:

$$\omega_0 = 0, \pi/4, 2\pi/4, 3\pi/4, \pi$$

$$\omega_0 = 5\pi/4, 6\pi/4, 7\pi/4, 2\pi$$

Interprete o conceito de freqüências altas e baixas. Se estas seqüências forem sintetizadas pelo conversor D/A para uma freqüência de amostragem especificada, F_s , quais as freqüências dos sinais analógicos gerados (Hz) ?

	$\omega_0 = \pi/4$	$\omega_0 = 2\pi/4$	$\omega_0 = 3\pi/4$	$\omega_0 = \pi$	$\omega_0 = 5\pi/4$
$F_s = 8 \text{ kHz}$					
$F_s = 16 \text{ kHz}$					

(Preencher Tabela)

EXERCÍCIO 5 – Implemente um programa para traçar a seguinte seqüência

$$x[n] = 5 \cdot \cos(0.5 \pi n + \pi/4) + 2 \cdot \sin(0.5 \pi n), \quad -20 \leq n \leq 20$$

EXERCÍCIO 6 – Analisar o funcionamento das seguintes funções, fornecidas com o livro *Digital Signal Processing using Matlab* (V. K. Ingle and J. G. Proakis):

A biblioteca de funções do livro está disponível no endereço:

[HTTP://WWW.BROOKSCOLE.COM/CGI-WADSWORTH/COURSE_PRODUCTS_WP.PL?FID=M20B&PRODUCT_ISBN_ISSN=0534371744&DISCIPLINE_NUMBER=9](http://www.brookscoble.com/cgi-wadsworth/course_products_wp.pl?fid=M20b&product_isbn_issn=0534371744&discipline_number=9)

Gerar uma função impulso, $x(t) = \delta(t - t_o)$ em tempo discreto, $x[n] = \delta(n - n_o)$, $n_1 \leq n \leq n_2$.

[x,n]=impseq(n0,n1,n2)

$$x[n] = \delta[n] \quad -20 \leq n \leq 20 \quad x[n] = \delta[n - 4] \quad -20 \leq n \leq 20$$

Gerar uma função degrau unitário, $x(t) = u(t - t_o)$ em tempo discreto, $x[n] = u(n - n_o)$, $n_1 \leq n \leq n_2$.

[x,n]=stepseq(n0,n1,n2)

$$x[n] = u[n] \quad -20 \leq n \leq 20 \quad x[n] = u[n - 4] \quad -20 \leq n \leq 20$$

Somar dois sinais em tempo discreto, $y[n] = x_1[n] + x_2[n]$: **[y,n]=sigadd(x1,n1,x2,n2)**

Multiplicar dois sinais em tempo discreto, $y[n] = x_1[n] \cdot x_2[n]$: **[y,n]=sigmult(x1,n1,x2,n2)**

$$x_1[n] = \{3, 0, 2, 1, 5, 7, 0, 0, 1, 1, 10\} \quad x_2[n] = \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$$

\uparrow
 \uparrow

Deslocar um sinal em tempo discreto, $y[n] = x[n - n_o]$: **[y,n]=sigshift(x,n,n0)**

$$x_1[n] = \{3, 0, 2, 1, 5, 7, 0, 0, 1, 1, 10\} \quad y[n] = x_1[n - 4]$$

\uparrow

Rebater um sinal em tempo discreto, $y[n] = x[-n]$: **[y,n]=sigfold(x,n)**

$$y[n] = x_1[-n]$$

EXERCÍCIO 7 – Implemente e trace o gráfico das seguintes seqüências utilizando as funções da biblioteca do livro *Digital Signal Processing using MATLAB* (Vinay K. Ingle and John G. Proakis).

a) $x[n] = 2 \delta[n + 2] - \delta[n - 4] \quad -20 \leq n \leq 20$

b) $x[n] = n \cdot \{u[n] - u[n - 10]\} + 10 \cdot \exp[-0.3 \cdot (n - 10)] \cdot \{u[n - 10] - u[n - 20]\} \quad 0 \leq n \leq 50$

11.

EXERCÍCIO 8 – Trace o gráfico das seguintes seqüências discretas e determine se estes sinais são periódicos.

$$x[n] = \cos\left(\frac{\pi}{4} n\right) \quad x[n] = \cos\left(\frac{3\pi}{8} n\right) \quad x[n] = \cos(n) \quad 0 \leq n \leq 30$$

PRÉ-RELATÓRIO PARA O LABORATÓRIO 2

Resolver os problemas P2.1 (itens b,c), P2.2 (itens a, b,c) da pág. 35 do livro *Digital Signal Processing using MATLAB* (Vinay K. Ingle and John G. Proakis).

OBS: O livro está disponível na biblioteca.

OBSERVAÇÃO SOBRE GERAÇÃO DE SEQÜÊNCIAS

$$x[n] = \{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$$

↑

$$x_1[n] = 2 \cdot x[n-5] - 3 \cdot x[n+4]$$

```
clear;
clc;
n = -2:10;
x = [1:7,6:-1:1];
[x11,n11] = sigshift(x,n,5);
[x12,n12] = sigshift(x,n,-4);
[x1,n1] = sigadd(2*x11,n11,-3*x12,n12);
figure(1);
subplot(2,1,1);
stem(n1,x1);
```

13. Decomposição em Seqüência Par e Ímpar

Uma seqüência discreta, $x[n]$, é classificada como par (simétrica) se $x_e[n] = x_e[-n]$.

Uma seqüência discreta, $x[n]$, é classificada como ímpar (assimétrica) se $x_o[n] = -x_o[-n]$.

Qualquer sinal real pode ser escrito como $x[n] = x_e[n] + x_o[n]$, onde

$$x_e[n] = \frac{1}{2}[x[n] + x[-n]] \quad x_o[n] = \frac{1}{2}[x[n] - x[-n]]$$

14. Sinais de Energia e Sinais de Potência

$$E_x = \sum_{n=-\infty}^{+\infty} |x[n]|^2 \quad E_N = \sum_{n=-N}^{+N} |x[n]|^2 \quad E_x = \lim_{N \rightarrow \infty} E_N$$

$$P_x = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N |x[n]|^2 = \lim_{N \rightarrow \infty} \frac{1}{2N+1} E_N$$

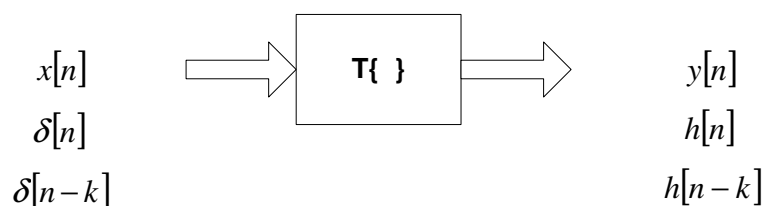
Para sinais periódicos

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$$

15. Escalonamento no Tempo (Down-Sampling)

$$y[n] = x[k \cdot n], \quad k \text{ inteiro, } (k > 0)$$

16. Sistemas Discretos Lineares e Invariantes no Tempo (LTI)



$$T\{a \cdot x_1[n] + b \cdot x_2[n]\} = a \cdot T\{x_1[n]\} + b \cdot T\{x_2[n]\}$$

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k] \quad \Rightarrow \quad y[n] = T\{x[n]\} = T\left\{ \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k] \right\}$$

Aplicando a propriedade da linearidade, temos

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot T\{\delta[n-k]\}$$

Aplicando a propriedade da invariância da resposta ao impulso, temos

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

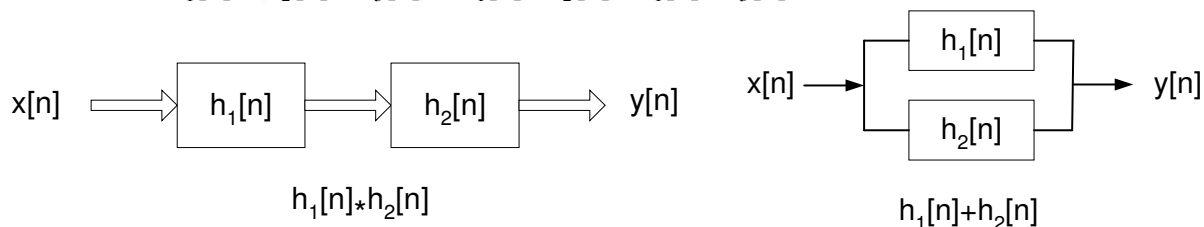
Este resultado representa a expressão de convolução (soma de convolução) para se calcular a resposta de um sistema LTI com resposta ao impulso $h[n]$ ao sinal de entrada, $y[n]$.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = x[n] * h[n]$$

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k] = h[n] * x[n]$$

PROPRIEDADES DA CONVOLUÇÃO

- a) Comutativa: $x[n] * h[n] = h[n] * x[n]$
- b) Associativa: $(x_1[n] * x_2[n]) * x_3[n] = x_1[n] * (x_2[n] * x_3[n])$
- c) Distributiva: $x_1[n] * (x_2[n] + x_3[n]) = x_1[n] * x_2[n] + x_1[n] * x_3[n]$



Exercício 1 – Realizar a convolução entre as duas seqüências discretas:

$$x[n] = [3, 0, 4, 0, 0, -2] \quad n = [-2, -1, 0, 1, 2, 3] \quad h[n] = [3, 2, 1] \quad n = [0, 1, 2]$$

Exercício 2 – Calcular a resposta ao impulso do sistema: $y[n] - a \cdot y[n-1] = x[n]$

17. Estabilidade de Sistemas LTI

Um sistema LTI é estável no sentido BIBO (Bounded Input–Bounded Output) se

$$S = \sum_{k=-\infty}^{\infty} |h[k]| < \infty$$

Um sistema BIBO implica que para uma seqüência de entrada limitada teremos uma saída limitada,

$$|x[n]| < \infty \Rightarrow |y[n]| < \infty \quad \forall x, y$$

18. Sistema LTI Causal

Um sistema LTI é causal se $h[n] = 0$ para $n < 0$.

19. Correlação e Autocorrelação de Seqüências

$$r_{x,y}[l] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n-l] = x[n] * y[-l] \quad \text{Correlação}$$

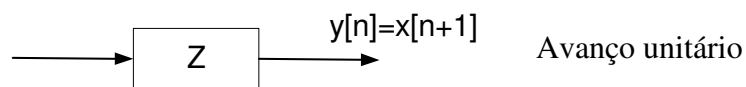
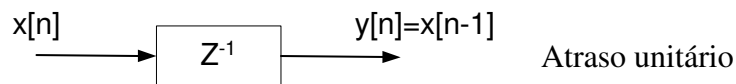
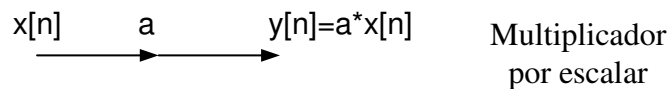
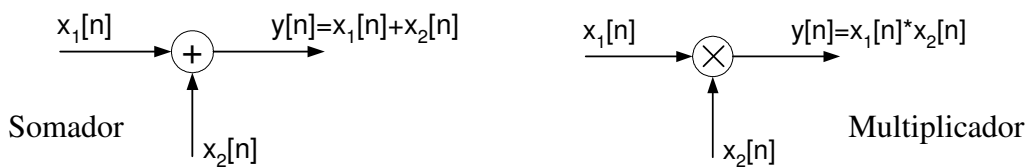
$$r_{x,x}[l] = \sum_{n=-\infty}^{\infty} x[n] \cdot x[n-l] = x[l] * x[-l] \quad \text{Autocorrelação}$$

20. Equação de Diferenças Linear com Coeficientes Constantes

$$\sum_{k=0}^N a_k \cdot y[n-k] = \sum_{k=0}^M b_k \cdot x[n-k]$$

$$a_0 \cdot y[n] + \sum_{k=1}^N a_k \cdot y[n-k] = \sum_{k=0}^M b_k \cdot x[n-k] \quad y[n] = -\sum_{k=1}^N \frac{a_k}{a_0} \cdot y[n-k] + \sum_{k=0}^M \frac{b_k}{a_0} \cdot x[n-k]$$

21. Representação de Sistemas usando Diagrama de Blocos



PRÉ-LAB 2 – SISTEMA DE DISCAGEM POR TONS DE FREQUÊNCIA (DTMF)

697 Hz	1	2	3	A	Implementar no Matlab® um programa para gerar os tons de frequência (DTMF) utilizados nos sistemas de discagem. Utilize uma frequência de amostragem de 8kHz. A duração de cada tom deve ser de 125ms. Utilize a função <i>sound</i> do Matlab para escutar os tons gerados.
770 Hz	4	5	6	B	
852 Hz	7	8	9	C	
941 Hz	*	0	#	D	
	1209 Hz	1336 Hz	1477 Hz	1633 Hz	

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
 LABORATÓRIO 2 – SINAIS E SISTEMAS DISCRETOS
 PROF. MARCELO PELLEZ

EXERCÍCIO 1 – Decomposição do Sinal em Componente Simétrica (Par) e Componente Assimétrica (Ímpar)

Qualquer seqüência discreta real, $x[n]$, pode ser decomposta em suas componentes par e ímpar:

$$x[n] = x_{par}[n] + x_{ímpar}[n]$$

$$x_{par}[n] = \frac{1}{2} \{ x[n] + x[-n] \} \quad x_{ímpar}[n] = \frac{1}{2} \{ x[n] - x[-n] \}$$

O programa a seguir utiliza a função **[xe,xo,m]=evenodd(x,n)** do *toolbox do livro* para decompor a seqüência discreta $x[n] = u[n] - u[n-10]$ em suas componentes par e ímpar. Verifique nos gráficos gerados, que se adicionarmos as seqüências $x_{par}[n] + x_{ímpar}[n]$ obteremos a seqüência original $x[n]$.

```
n = [0:10];
x = stepseq(0,0,10)-stepseq(10,0,10);
[xe,xo,m] = evenodd(x,n);
subplot(1,1,1)
subplot(2,2,1); stem(n,x); title('Pulso Retangular')
xlabel('n'); ylabel('x(n)'); axis([-10,10,0,1.2])
subplot(2,2,2); stem(m,xe); title('Componente Par')
xlabel('n'); ylabel('xe(n)'); axis([-10,10,0,1.2])
subplot(2,2,4); stem(m,xo); title('Componente Impar')
xlabel('n'); ylabel('xe(n)'); axis([-10,10,-0.6,0.6])
```

EXERCÍCIO 2 – Convolução Linear

Para sistemas lineares invariantes no tempo (LTI – Linear Time-Invariant Systems), podemos relacionar os sinais de entrada e saída do sistema através da soma de convolução, definida pela equação:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k]$$

Se conhecermos a resposta ao impulso do sistema (por exemplo um filtro digital) podemos calcular a saída $y[n]$ fazendo a convolução do sinal de entrada, $x[n]$, com a resposta ao impulso do sistema, $h[n]$. Utilize a função **y=conv(x,h)** do Matlab® para calcular a convolução linear entre duas seqüências discretas de duração finita. A função **conv** assume que as duas seqüências iniciam no instante **n=0**.

$$x[n] = [3 \ 11 \ 7 \ 0 \ -1 \ 4 \ 2] \text{ - Sinal de entrada do sistema}$$

$$h[n] = [2 \ 3 \ 0 \ -5 \ 2 \ 1] \text{ - Resposta ao impulso do sistema}$$

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \text{ - Saída do sistema}$$

No programa trace os gráficos das seqüências $x[n]$, $h[n]$ e $y[n]$. Interprete graficamente a operação de convolução.

EXERCÍCIO 3 – Convolução Linear

Para se fazer a convolução entre seqüências genéricas

$$x[n], \quad n_{xb} \leq n \leq n_{xe}$$

$$h[n], \quad n_{hb} \leq n \leq n_{he}$$

utilize a função `[y,ny]=conv_m(x,nx,h,nh)` do *toolbox do livro*. Esta função já retorna também um vetor para traçar o gráfico correto de $y[n]$. Faça a convolução linear entre as duas seqüências discretas abaixo:

$$x[n] = [3 \ 11 \ 7 \ 0 \ -1 \ 4 \ 2], \quad -3 \leq n \leq 3$$

$$h[n] = [2 \ 3 \ 0 \ -5 \ 2 \ 1], \quad -1 \leq n \leq 4$$

Trace o gráfico de $x[n]$, $h[n]$ e $y[n] = x[n] * h[n]$. Interprete graficamente a convolução.

EXERCÍCIO 4 – Cálculo de Correlação entre Seqüências

A correlação é uma operação muito usada em aplicações de processamento digital de sinais. É uma medida do grau de similaridade entre duas seqüências (sinais).

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n-l] = y[l] * x[-l]$$

A autocorrelação nos dá uma medida de auto-similaridade do sinal com versões deslocadas dele mesmo:

$$r_{xx}[l] = \sum_{n=-\infty}^{\infty} x[n] \cdot x[n-l] = x[l] * x[-l]$$

O cálculo da correlação e autocorrelação podem ser implementados usando a convolução. Considere a seqüência $x[n] = [3 \ 11 \ 7 \ 0 \ -1 \ 4 \ 2]$, $-3 \leq n \leq 3$ e a seqüência $y[n]$ como sendo uma versão deslocada e com ruído da seqüência $x[n]$, ou seja, $y[n] = x[n-2] + w[n]$.

```
% sequencia com ruído 1
x = [3, 11, 7, 0, -1, 4, 2]; nx=[-3:3]; % sinal x(n)
[y,ny] = sigshift(x,nx,2); % x(n-2)
w = randn(1,length(y)); nw = ny; % ruído w(n)
[y,ny] = sigadd(y,ny,w,nw); % y(n) = x(n-2) + w(n)
[x,nx] = sigfold(x,nx); % x(-n)
[rxy,nrxy] = conv_m(y,ny,x,nx); % autocorrelacao
subplot(1,1,1)
subplot(2,1,1);stem(nrxy,rxy)
axis([-4,8,-50,250]);xlabel('deslocamento l')
ylabel('rxy');title('Autocorrelacao: sequencia com ruído 1')
gtext('Maximo')
% sequencia com ruído 2
x = [3, 11, 7, 0, -1, 4, 2]; nx=[-3:3]; % sinal x(n)
[y,ny] = sigshift(x,nx,2); % x(n-2)
w = randn(1,length(y)); nw = ny; % ruído w(n)
[y,ny] = sigadd(y,ny,w,nw); % y(n) = x(n-2) + w(n)
[x,nx] = sigfold(x,nx); % x(-n)
[rxy,nrxy] = conv_m(y,ny,x,nx); % autocorrelacao
subplot(2,1,2);stem(nrxy,rxy)
gtext('Maximum')
axis([-4,8,-50,250]);xlabel('deslocamento l')
ylabel('rxy');title('Autocorrelacao: sequence com ruído 2')
```

O MATLAB[®] possui a função `xcorr(x,y)` e `xcorr(x)` para calcular a correlação e autocorrelação.

EXERCÍCIO 5 – Exemplo de Implementação de um Filtro Digital

Exemplo de aplicação do filtro de média móvel para filtragem de um sinal com ruído:

$$y[n] = \frac{1}{M} \sum_{i=0}^{M-1} x[n-k]$$

```
% Filtro de média móvel para M=3
```

```
clear; clc;
```

```
figure(1);
```

```
N=50;
```

```
d=rand(N,1)-0.5;
```

```
n=0:1:N-1;
```

```
s=2*n.*(0.9.^n);
```

```
subplot(2,1,1); stem(n,s); xlabel('n'); ylabel('Amplitude'); title('s[n]');
```

```
subplot(2,1,2); stem(n,d); xlabel('n'); ylabel('Amplitude'); title('d[n]');
```

```
N=50;
```

```
d=rand(N,1)-0.5;
```

```
n=0:1:N-1;
```

```
s=2*n.*(0.9.^n);
```

```
x=s+d';
```

```
figure(2);
```

```
plot(n,d,'r-',n,s,'b-',n,x,'g-'); xlabel('n'); ylabel('Amplitude'); legend('r-','d[n]','b-','s[n]','g-','x[n]');
```

```
M=3;
```

```
b=ones(M,1)/M;
```

```
y=filter(b,1,x);
```

```
figure(3);
```

```
plot(n,s,'r-',n,y,'b-'); legend('r-','s[n]','b-','y[n]'); xlabel('n'); ylabel('Amplitude');
```

EXERCÍCIO 6 – Equações Lineares com Coeficientes Constantes

Um sistema discreto linear pode ser caracterizado por uma equação linear de coeficientes constantes na forma

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

A função **y=filter(b,a,x)** do Matlab resolve a equação de diferenças. Considere a seguinte equação de diferenças

$$y[n] - y[n-1] + 0.9 y[n-2] = x[n]$$

Determine:

- Determine a resposta ao impulso do sistema, $h[n]$, no intervalo $-20 \leq n \leq 120$
- Determine a resposta ao degrau do sistema, $s[n]$, no intervalo $-20 \leq n \leq 120$

PRÉ-RELATÓRIO PARA O LABORATÓRIO 3

Resolver os problemas P2.3, P2.4 (para itens a, b, c do P2.2), P2.5, P2.6 e P2.7 do livro *Digital Signal Processing using MATLAB* (Vinay K. Ingle and John G. Proakis).

Análise Espectral de Sinais Contínuos no Tempo

1 – Série de Fourier

Sinais contínuos periódicos (sinais de potência) podem ser expressos pela série complexa de Fourier,

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \cdot e^{j2\pi k F_0 t} \quad F_0 = \frac{1}{T_0}$$

$$c_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) \cdot e^{-j2\pi k F_0 t} dt \quad k = 0, \pm 1, \pm 2, \dots$$

$$c_k = |c_k| \cdot e^{j\theta_k} \quad \text{Coeficientes complexos da série de Fourier}$$

2 – Transformada de Fourier

Sinais contínuos aperiódicos (sinais de energia)

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(e^{jw}) \cdot e^{jw t} dw$$

$$X(e^{jw}) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-jw t} dt$$

Análise Espectral de Sinais Discretos no Tempo

1 – Série de Fourier

Sinais discretos periódicos, $x[n] = x[n + N]$, podem ser expressos pela série complexa de Fourier,

$$x[n] = \sum_{k=0}^{N-1} c_k \cdot e^{j2\pi kn/N} = \sum_{k=0}^{N-1} c_k \cdot e^{j2\pi f n} \quad f = \frac{k}{N}$$

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1$$

$$e^{j2\pi kn/N} = e^{jw_k n} \quad w_k = \frac{2\pi k}{N}$$

O espectro do sinal $x[n]$ é também uma seqüência periódica com período N .

$$0 \leq w_k < 2\pi \quad 0 \leq k \leq N-1 \quad 0 \leq F \leq F_s$$

$$-\pi \leq w_k < \pi \quad -\frac{N}{2} < k \leq \frac{N}{2}$$

2 – Transformada de Fourier em Tempo Discreto (DTFT)

Sinais discretos não periódicos

$$X(e^{jw}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-jwn}$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{jw}) \cdot e^{jwn} dw$$

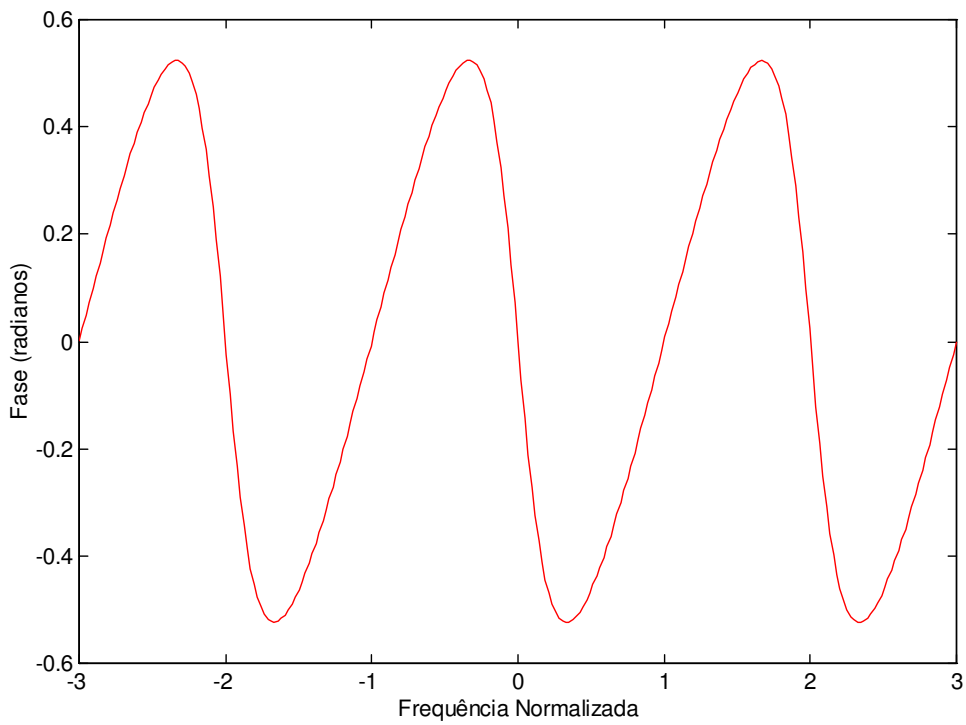
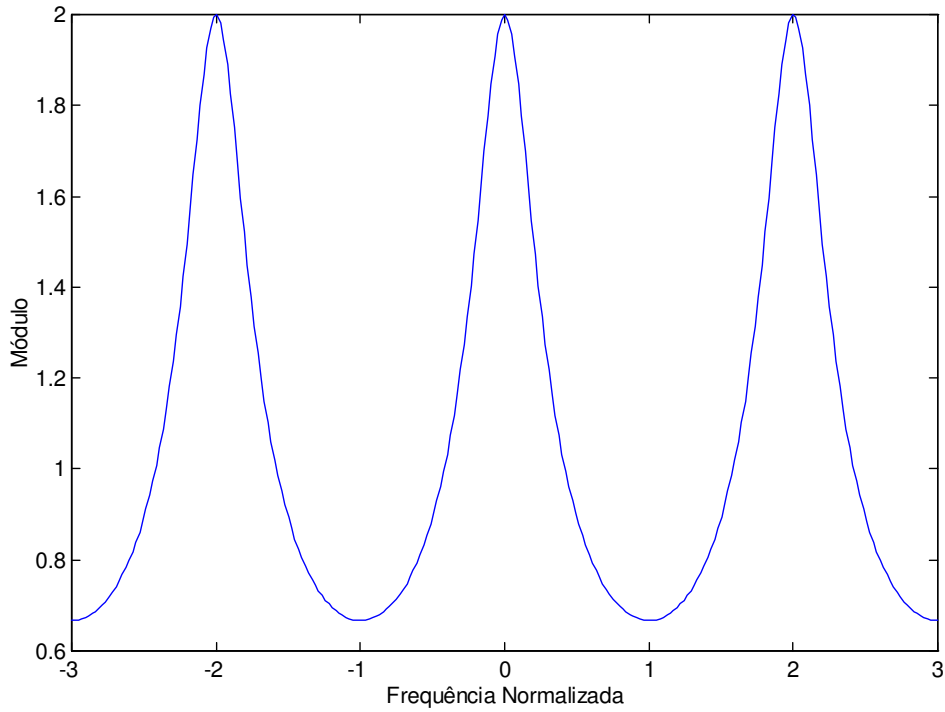
$$X(e^{jw}) = X_{\text{Re}}(w) + j X_{\text{Im}}(w) = |X(e^{jw})| \cdot e^{j\theta(w)}$$

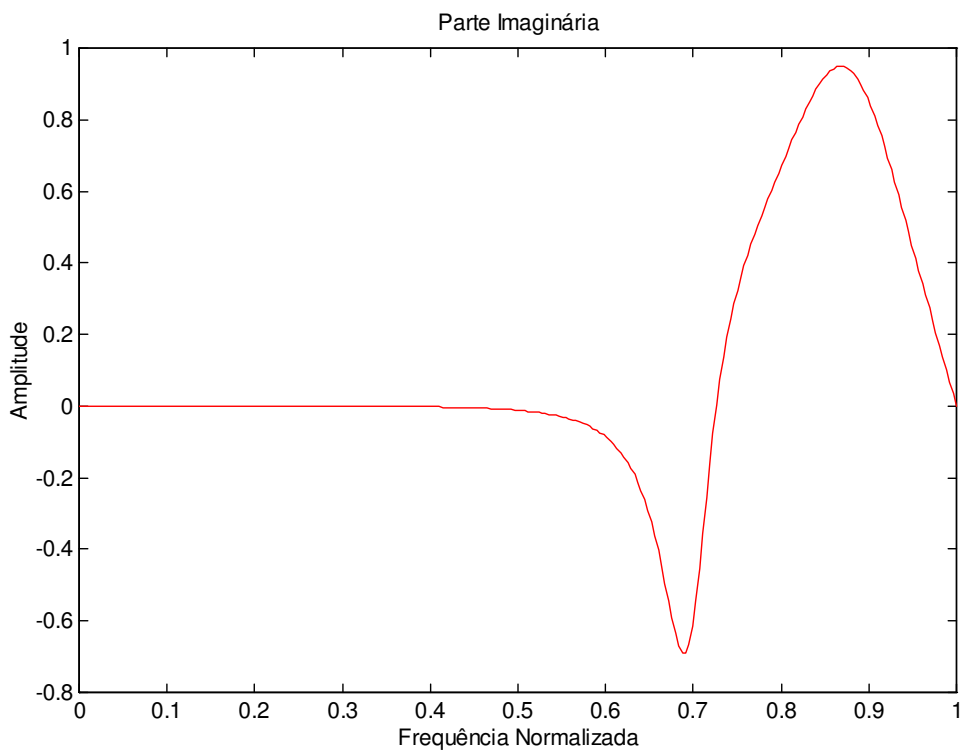
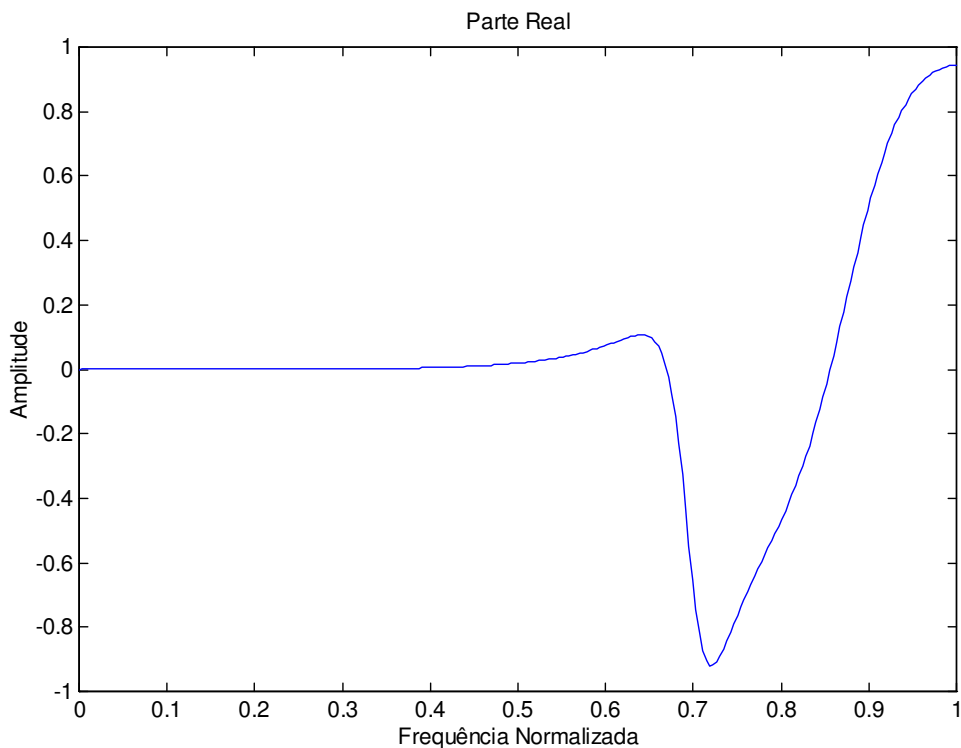
Seqüência	DTFT
$\delta[n]$	1
$u[n]$	$\frac{1}{1 - e^{-jw}} + \sum_{k=-\infty}^{\infty} \pi \cdot \delta[w + 2\pi k]$
$e^{jw_0 n}$	$\sum_{k=-\infty}^{\infty} 2\pi \cdot \delta[w - w_0 + 2\pi k]$
$\alpha^n \cdot u[n] \quad \alpha < 1$	$\frac{1}{1 - \alpha \cdot e^{-jw}}$

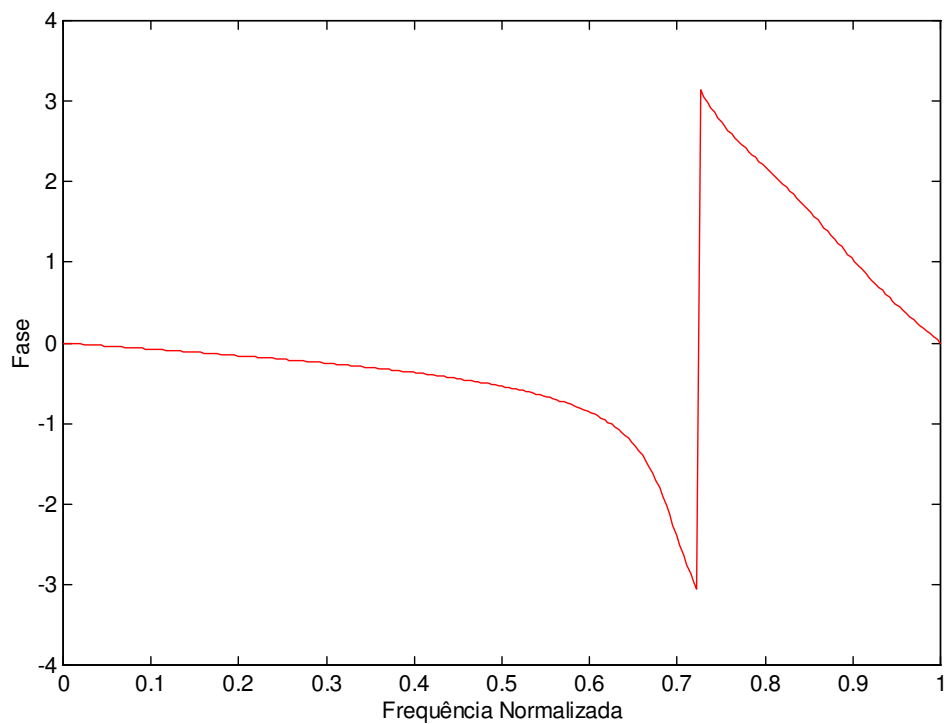
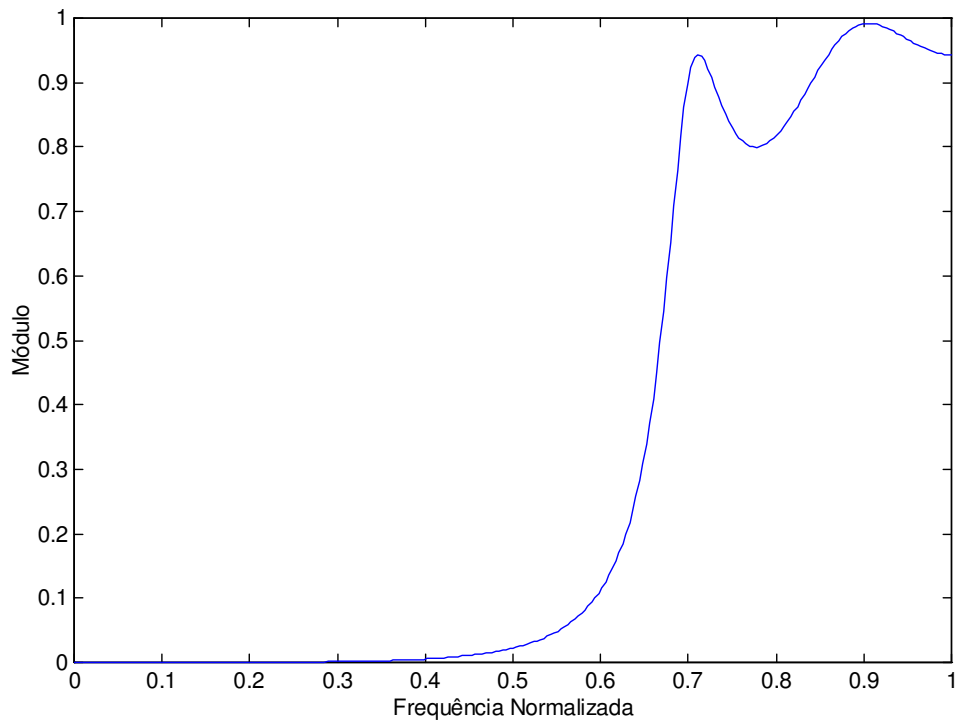
TABELA 1 – Pares da transformada de Fourier em tempo discreto (DTFT)

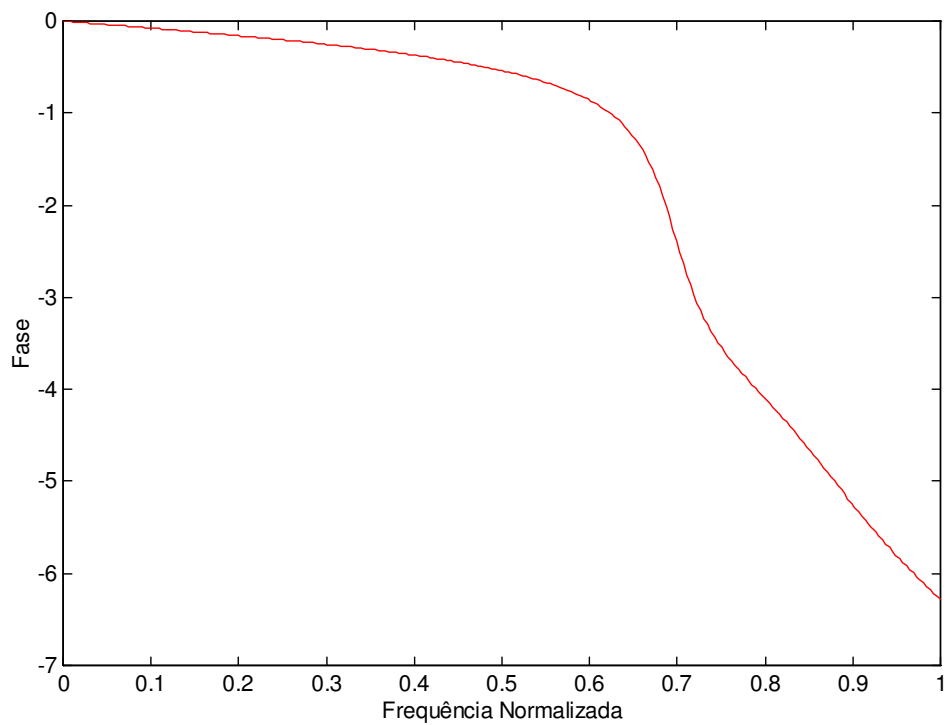
Exemplo

$$x[n] = (0,5)^n \cdot u[n] \Rightarrow X(e^{j\omega}) = \frac{1}{1 - 0,5 \cdot e^{-j\omega}}$$









TRANSFORMADA DISCRETA DE FOURIER (DFT)

$$x[n], 0 \leq n \leq N-1 \Rightarrow X(e^{jw})$$

Precisamos apenas de N pontos de frequência distintos para representar unicamente $x[n]$ e $X(e^{jw})$. Os valores de frequência são calculados em $w = w_k, k = 0, 1, \dots, N-1$. A DFT é aplicável apenas para seqüências finitas.

$$X[k] = X(e^{jw})_{w=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi kn}{N}} \quad k = 0, 1, \dots, N-1$$

$$x[n] = \frac{1}{N} \cdot \sum_{k=0}^{N-1} X[k] \cdot W_N^{-kn} \quad n = 0, 1, \dots, N-1$$

A seqüência $X[k]$ é chamada de DFT. Podemos definir $W_N = e^{-j\frac{2\pi}{N}}$

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} \quad k = 0, 1, \dots, N-1 \quad (\text{DFT})$$

$$x[n] = \frac{1}{N} \cdot \sum_{k=0}^{N-1} X[k] \cdot W_N^{-kn} \quad n = 0, 1, \dots, N-1 \quad (\text{IDFT})$$

Representação Matricial

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} \quad k = 0, 1, \dots, N-1$$

$$\bar{X} = D_N \cdot \bar{x} \quad \text{DFT de N pontos}$$

$$\bar{x} = D_N^{-1} \cdot \bar{X} \quad \text{IDFT de N pontos}$$

$$\bar{X} = [X[0] \quad X[1] \quad \dots \quad X[N-1]]^T$$

$$\bar{x} = [x[0] \quad x[1] \quad \dots \quad x[N-1]]^T$$

$$D_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \quad \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} = D_N^{-1} \cdot \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} \quad D_N^{-1} = \frac{1}{N} \cdot D_N^*$$

Cálculo Numérico da DTFT usando a DFT

Seja $X(e^{j\omega})$ a DTFT de uma seqüência $x[n]$ de comprimento N . Se desejarmos calcular $X(e^{j\omega})$ numa escala densa de freqüências $\omega_k = \frac{2\pi k}{M}$, $k = 0, 1, \dots, M - 1$, onde $M \gg N$, teremos

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\omega_k n} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \frac{2\pi k n}{M}}$$

Definindo $x_e[n] = x[n] + (M - N)$ amostras nulas,

$$X(e^{j\omega}) = \sum_{n=0}^{M-1} x_e[n] \cdot e^{-j \frac{2\pi k n}{M}} \quad \text{DFT de } M \text{ pontos}$$

A DFT de uma seqüência de comprimento M é calculada eficientemente usando o algoritmo de FFT para M inteiro, potência de dois.

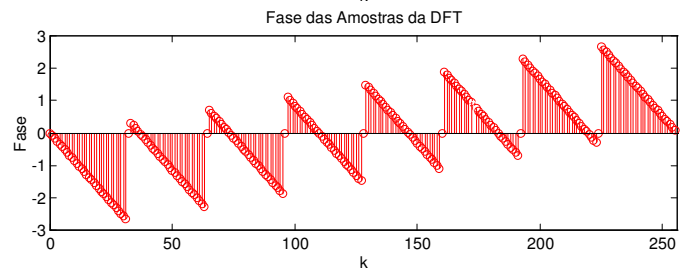
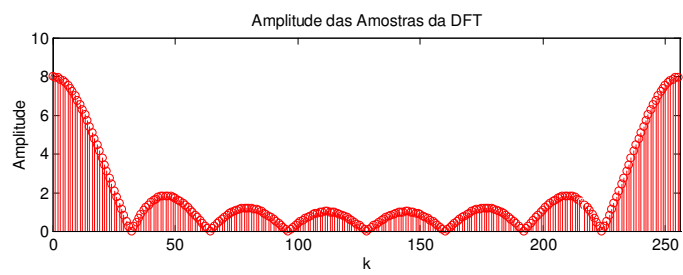
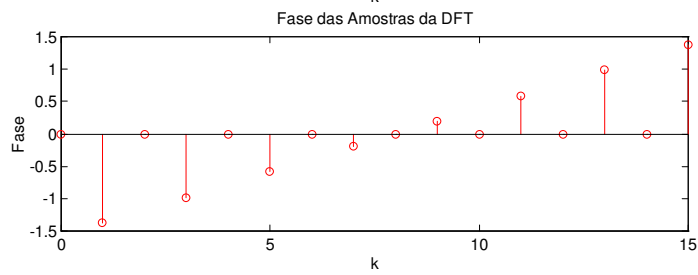
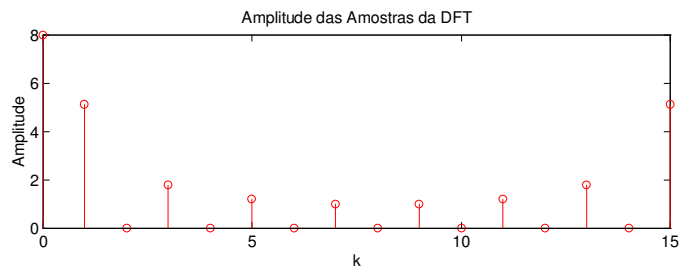
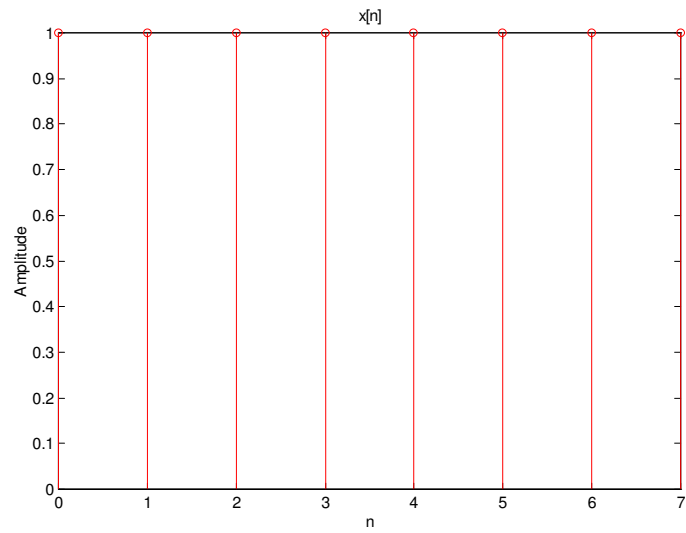
Propriedades da DFT

Seqüência de comprimento N	DFT de N pontos
$x[n]$	$X[k]$
$h[n]$	$H[k]$
$\alpha \cdot x[n] + \beta \cdot h[n]$	$\alpha \cdot X[k] + \beta \cdot H[k]$
$x[\langle n - n_0 \rangle_N]$	$W_N^{k n_0} \cdot X[k]$
$\sum_{k=0}^{N-1} x[k] \cdot h[\langle n - k \rangle_N]$	$G[k] \cdot H[k]$

Convolução Circular versus Convolução Linear

$$y_L[n] = \sum_{k=0}^{N-1} x[k] \cdot h[n - k] = x[n] * h[n] \quad n = 0, 1, \dots, 2N - 2$$

$$y_C[n] = \sum_{k=0}^{N-1} x[k] \cdot h[\langle n - k \rangle_N] = x[n] \circ h[n] \quad n = 0, 1, \dots, 2N - 2$$



Representação de Sistemas LTI no Domínio da Frequência

A resposta em frequência, $H(e^{j\omega})$, é uma função complexa com período 2π :

$$H(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} h[n] \cdot e^{-j\omega n} = |H(e^{j\omega})| \cdot e^{j\theta(\omega)}$$

$$G(\omega) = 20 \cdot \log_{10} |H(e^{j\omega})| \quad (\text{dB}) \quad \text{Função de Ganho}$$

Para $h[n]$ real, $|H(e^{j\omega})|$ é uma *função par* e $\theta(\omega)$ é uma *função ímpar*. A derivada da resposta em fase é denominada de *atraso de grupo*, $\tau(\omega)$. A função de atraso de grupo fornece uma medida da linearidade da função de fase. O atraso de grupo é dado em segundos. Se a resposta em fase $\theta(\omega)$ for linear, o atraso de grupo será constante.

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega}$$

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 3 – TRANSFORMADA DE FOURIER DISCRETA (DFT)
PROF. MARCELO PELLEZ

Para o cálculo do espectro de freqüências de sinais contínuos, utilizamos a série de Fourier (sinais periódicos/potência) e a transformada de Fourier (sinais não periódicos/energia). A transformada de Fourier é de particular interesse porque na prática utilizamos sinais de energia para transmissão digital. As expressões da transformada e transformada inversa de Fourier são:

$$X(\omega) = F\{x(t)\} = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j\omega t} dt$$

$$x(t) = F^{-1}\{X(f)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) \cdot \exp(j\omega t) df$$

Quando aplicamos a transformada de Fourier numa seqüência em tempo discreto, $x[n]$, as expressões da transformada se simplificam para

$$X(\omega) = F\{x[n]\} = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\omega n}$$

$$x[n] = F^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(\omega) \cdot e^{j\omega n} d\omega$$

Exercício 1 – Determine analiticamente a transformada de Fourier em tempo discreto (DTFT) das seqüências abaixo e trace o espectro (módulo/fase/parte real/parte imaginária) num intervalo $-6\pi \leq \omega \leq 6\pi$.

a) $x[n] = (0,5)^n \cdot u[n]$

b) $x[n] = \{1, 2, 3, 4, 5\}$
 $\quad \quad \quad \uparrow$

Analise a periodicidade do gráfico do espectro e responda as seguintes questões:

- Entre que faixa de valores de freqüência normalizada, $\omega = 2\pi f$, devemos analisar os gráficos de espectro
- Se desejarmos a informação do espectro em hertz, como podemos proceder ?
- Através das observações dos gráficos de espectro, classifique o módulo, fase, parte real e parte imaginária como função par ou ímpar.

Exercício 2 – Considere as seguintes DTFTs

$$X(e^{j\omega}) = \frac{1}{1 - 0.5e^{-j\omega}}$$

$$X(e^{j\omega}) = \frac{0.008 - 0.033e^{-j\omega} + 0.05e^{-j2\omega} - 0.033e^{-j3\omega} + 0.008e^{-j4\omega}}{1 + 2.37e^{-j\omega} + 2.7e^{-j2\omega} + 1.6e^{-j3\omega} + 0.41e^{-j4\omega}}$$

Desenhe os gráficos da parte real, imaginária, módulo e fase das DTFTs, para k pontos de freqüência. Considere $k = 256, 512, 1024, 2048$.

Transformada de Fourier Discreta (DFT)

Como podemos observar, o cálculo do espectro de uma seqüência em tempo discreto deve ser feito analiticamente através da DTFT. Para termos um cálculo não analítico (computacional), o que implementa é a Transformada Discreta de Fourier (DFT), que são amostras da DTFT.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \dots, N-1 \quad \text{DFT}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad n = 0, 1, \dots, N-1 \quad \text{IDFT}$$

$$W_N = e^{-j2\pi/N}$$

Exercício 2 - Determine a DFT de M pontos, $U[k]$, da seguinte seqüência discreta de comprimento N ,

$$u[n] = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{fora} \end{cases}$$

Utilize $N = 8$, $M = 8$, $M = 16$, $M = 64$, $M = 256$.

```
% Exemplo do calculo computacional da DFT

N=input('Digite o comprimento N da sequencia = ');
M=input('Digite o comprimento da DFT = ');

u=[ones(1,N)];
U=fft(u,M);

n=0:1:N-1;
figure(1);
stem(n,u);
title('x[n]');
ylabel('Amplitude');
xlabel('n');

figure(2);
subplot(2,1,1);
k=0:1:M-1;
stem(k,abs(U));
axis([0 256 0 10]);
title('Amplitude das Amostras da DFT');
ylabel('Amplitude');
xlabel('k');

subplot(2,1,2);
stem(k,angle(U));
axis([0 256 -3 3]);
title('Fase das Amostras da DFT');
ylabel('Fase');
xlabel('k');
```

Exercício 3 - Determine a DTFT $X(e^{j\omega})$ de uma seqüência $x[n] = \cos\left(2\pi \frac{3}{16}n\right)$ de comprimento 16, usando uma DFT de 512 pontos.

```
% Calculo numerico da DTFT usando a DFT
clear; clc;
k=0:15;
x=cos(2*pi*k*3/16);
X=fft(x);
XE=fft(x,512);
L=0:511;
plot(L/M,abs(XE),'b');
hold
plot(k/N,abs(X),'or');
xlabel('Frequencia Angular Normalizada');
ylabel('Amplitude');
```

Exercício 4 – Filtro FIR Passa Alta – O programa a seguir implementa um filtro FIR de comprimento 3, definido pela equação de diferenças,

$$y[n] = -6.76195 \cdot x[n] + 13.456335 x[n-1] - 6.76195 \cdot x[n-2].$$

O sinal de entrada do filtro é $x[n] = \{\cos(0.1n) + \cos(0.4n)\}u[n]$.

```
% Filtro Digital FIR Passa Alta
% Especifica os coeficientes do filtro
b=[-6.76195 13.456335 -6.76195];
% Gera as duas sequencias senoidais
n=0:99;
x1=cos(0.1*n);
x2=cos(0.4*n);
% Gera a sequencia de saida do filtro
y=filter(b,1,x1+x2);
% Mostra as sequencias de entrada e saida
plot(n,y,'r-',n,x2,'b-',n,x1,'g-',n,x1+x2,'c-'); grid;
axis([0 100 -2 4]);
ylabel('Amplitude');
xlabel('n');
legend('r-', 'y[n]', 'b-', 'x2[n]', 'g-', 'x1[n]', 'c-', 'x1[n]+x2[n]');
```

Exercício 5 – Filtro FIR Passa Baixa – Utilize o mesmo programa do Exercício 4 substituindo apenas os novos coeficientes do filtro, $b=[6.7619 \ -12.4563 \ 6.7619]$.

Exercício 6 – O programa a seguir mostra a resposta em frequência (módulo/fase) do filtro FIR implementado no Exercício 4.

$$H(e^{j\omega}) = (2\alpha \cos \omega + \beta) e^{-j\omega} \quad \alpha = -6.76195 \quad \beta = 13.456635$$

```
% Resposta em frequencia do filtro FIR
clear; clc;
% Coeficientes do filtro
h=[-6.76195 13.456335 -6.76195];
w=0:pi/1000:pi;
%Calcula a resposta em frequencia
```

```
H=freqz(h,1,w);
%Calcula e mostra a resposta em amplitude
m=abs(H);
figure(1);
plot(w,m,'r-');
axis([0 0.4 0 1]);
ylabel('Amplitude');
xlabel('w');
figure(2);
plot(w/pi,m,'r-');
ylabel('Amplitude');
xlabel('w/pi (normalizado)');
%Calcula e mostra as respostas em fase
pha=angle(H);
figure(3);
plot(w/pi,pha);
axis([0 1 -4 4]);
ylabel('Fase em Graus');
xlabel('Frequência Angular Normalizada');
```

Transformada-Z

Nas seções anteriores estudamos a *Transformada de Fourier em Tempo Discreto* (DTFT) para representar sinais discretos usando seqüências exponenciais complexas. A DTFT de um sistema linear invariante no tempo, com resposta ao impulso $h[n]$, é denominada de resposta em freqüência do sistema, $H(e^{jw})$. Contudo, existem duas restrições para a abordagem da DTFT. A primeira é que para muitos sinais úteis na prática como $u[n]$ e $n \cdot u[n]$ a DTFT não existe (não converge). A segunda é que a resposta transitória de um sistema devido a condições iniciais ou alterações da entrada não podem ser calculadas usando a abordagem da DTFT. Portanto utilizamos uma extensão da DTFT denominada de transformada-Z:

$$H(Z) = \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n} \quad \text{Transformada-Z}$$

onde $z = r \cdot e^{jw}$ é uma variável complexa. O conjunto de valores de Z para o qual $H(z)$ existe é chamado de região de convergência (ROC), $R_{x-} < |z| < R_{x+}$, para números positivos R_{x-} e R_{x+} . A transformada Z inversa para uma seqüência complexa $H(Z)$ é dada por

$$h[n] = \frac{1}{2\pi j} \oint_C H(Z) z^{n-1} dz \quad \text{Transformada-Z Inversa}$$

A variável complexa Z é chamada de freqüência complexa dada por $z = |z| \cdot e^{jw}$, onde $|z|$ é a atenuação e w é a freqüência real. Dado que a região de convergência é definida em termos da magnitude $|z|$, o formato da ROC é um anel. Note que R_{x-} pode ser igual a zero e/ou R_{x+} pode ser inclusive ∞ . Se $R_{x+} < R_{x-}$, então a ROC é um espaço nulo e a transformada Z não existe. A função $|z|=1$ ($z = e^{jw}$) é um círculo de raio unitário no plano Z e é denominado *círculo unitário*. Se a ROC contém o círculo unitário, a transformada z se reduz a transformada de Fourier em tempo discreto (DTFT), $X(e^{jw})$. A DTFT converge se a região de convergência da transformada Z incluir o círculo de raio unitário,

$$H(z) \Big|_{z=e^{jw}} = H(e^{jw}).$$

Exercício – Calcule a transformada-Z da seqüência $x[n] = a^n \cdot u[n]$, $0 < |a| < \infty$.

1 – A região de convergência é *sempre limitada por um círculo* pois a condição de convergência é sobre o $|z|$.

2 – Para uma seqüência limitada a esquerda (zero para $n < n_0$), a região de convergência é sempre a *parte externa* de um círculo de raio R_{x-} . Para $n_0 \geq 0$ a seqüência é causal.

$$x[n] = a^n \cdot u[n]$$

3 – Para uma seqüência limitada a direita (zero para $n > n_0$), a região de convergência é sempre a *parte interna* do círculo de raio R_{x+} .

$$x[n] = -b^n \cdot u[-n-1]$$

4 – Para uma seqüência não limitada a direita nem à esquerda, a região de convergência é um anel,

$$R_{x-} < |z| < R_{x+}$$

5 – Seqüências que são zero para $n < n_1$ e $n > n_2$ são chamadas de seqüências com duração finita. A região de convergência para tais seqüências é *todo o plano Z*. Se $n_1 < 0$, então $z = \infty$ não está na região de convergência. Se $n_2 > 0$, então $z = 0$ não está na região de convergência

6 – A região de convergência não pode incluir nenhum pólo pois $H(Z)$ converge uniformemente nesta região.

7 – Existe pelo menos um pólo no limite da região de convergência. A ROC é limitada por pólos.

Seqüência Discreta	Transformada Z	ROC
$\delta[n]$	1	$\forall z$
$u[n]$	$\frac{1}{1-z^{-1}}$	$ z > 1$
$-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$ z < 1$
$a^n u[n]$	$\frac{1}{1-a z^{-1}}$	$ z > a $
$-b^n u[-n-1]$	$\frac{1}{1-b z^{-1}}$	$ z < b $
$[a^n \cdot \text{sen}(w_0 n)] \cdot u[n]$	$\frac{[a \cdot \text{sen}(w_0)] \cdot z^{-1}}{1 - [2a \cos(w_0)]z^{-1} + a^2 z^{-2}}$	$ z > a $
$[a^n \cdot \cos(w_0 n)] \cdot u[n]$	$\frac{1 - [a \cdot \cos(w_0)] \cdot z^{-1}}{1 - [2a \cos(w_0)]z^{-1} + a^2 z^{-2}}$	$ z > a $
$n a^n u[n]$	$\frac{a z^{-1}}{(1-a z^{-1})^2}$	$ z > a $
$-n b^n u[-n-1]$	$\frac{b z^{-1}}{(1-b z^{-1})^2}$	$ z < b $

Região de Convergência (ROC)

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{a_0 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}}$$

$$H(z) = \frac{b_0}{a_0} z^{(N-M)} \frac{\prod_{l=1}^M (z - z_l)}{\prod_{l=1}^N (z - p_l)}$$

Zeros $\Rightarrow z = z_l \Rightarrow H(z_l) = 0$

Pólos $\Rightarrow z = p_l \Rightarrow H(p_l) = \infty$

Se $N > M$ teremos $(N - M)$ zeros em $z = 0$

Se $N < M$ teremos $(N - M)$ pólos em $z = 0$

Transformada-Z Inversa usando Frações Parciais

A transformada-Z inversa pode ser obtida através da expansão em frações parciais:

$$H(z) = \sum_{k=1}^N \frac{R_k}{1 - p_k z^{-1}} + \underbrace{\sum_{k=0}^{M-N} C_k \cdot z^{-k}}_{M \geq N}$$

$$R_k = \left. \frac{\tilde{b}_0 + \tilde{b}_1 z^{-1} + \dots + \tilde{b}_{N-1} z^{-(N-1)}}{1 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \cdot (1 - p_k z^{-1}) \right|_{z=p_k}$$

$$h[n] = \sum_{l=1}^N R_k \cdot Z^{-1} \left\{ \frac{1}{1 - p_k z^{-1}} \right\} + \underbrace{\sum_{k=0}^{M-N} C_k \cdot \delta(n-k)}_{M \geq N}$$

Teorema 1 - Um sistema LTI é estável se e somente se o círculo de raio unitário está dentro da região de convergência (ROC) de $H(z)$

Teorema 2 - Um sistema LTI causal é estável se e somente se a função de transferência $H(z)$ possui todos os pólos dentro do círculo de raio unitário.

Relação da Função de Transferência com a Equação de Diferenças

$$\sum_{k=0}^N d_k \cdot y[n-k] = \sum_{k=0}^M p_k \cdot x[n-k]$$

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{\sum_{k=0}^M p_k \cdot e^{-j\omega k}}{\sum_{k=0}^N d_k \cdot e^{-j\omega k}}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M p_k \cdot z^{-k}}{\sum_{k=0}^N d_k \cdot z^{-k}}$$

Exercício 1 – Determine a função de transferência de um sistema com a seguinte equação de diferenças:

$$y[n] = 2 \cdot x[n] + 2,6 \cdot x[n-1] + 0,78 \cdot x[n-2] + 0,054 \cdot x[n-3] \\ + 1,5 \cdot y[n-1] - 0,47 \cdot y[n-2] - 0,063 \cdot y[n-3]$$

Exercício 2 – Determine a função de transferência do sistema com a seguinte equação de diferenças:

$$y[n] = 3 \cdot x[n] + 1,2 \cdot x[n-1] - 5,22 \cdot x[n-2] - 1,548 \cdot x[n-3] + 0,359 \cdot x[n-4] \\ - 0,6 \cdot y[n-1] - 0,68 \cdot y[n-2] - 0,5 \cdot y[n-3] + 0,477 \cdot y[n-4]$$

Exercício 3 – Determine os primeiros 3 termos da resposta ao impulso, $h[n]$, do filtro IIR causal com a seguinte função de transferência:

$$H(z) = \frac{(2 + 0,4 \cdot z^{-1})(0,5 + 2 \cdot z^{-1})}{(1 - 2 \cdot z^{-1} + 2 \cdot z^{-2})(1 - 0,5 \cdot z^{-1})}$$

Projeto de Filtros Digitais FIR (Finite Impulse Response)

Vantagens dos filtros FIR:

- A resposta em fase pode ser exatamente linear
- São fáceis de implementar porque não há problemas de estabilidade
- A DFT pode ser usada na implementação

Vantagens da resposta em fase linear:

- A implementação envolve apenas aritmética real e não complexa
- Não existe distorção de atraso e sim apenas um atraso fixo entre entrada/saída
- Para filtros de comprimento N (ordem $N-1$) o número de operações é da ordem de $N/2$

Metodologia de Projeto – Método das Janelas

A ênfase do método está em selecionar uma janela apropriada e um filtro ideal:

- a) Selecionar um filtro ideal (não causal), que possui resposta ao impulso infinita
- b) Truncar a resposta ao impulso (usando uma janela) para obter um filtro FIR causal
- c) Deslocar o início do vetor de amostras, $h[n]$, para iniciar em $n=0$
(equivale a associar uma fase linear)

OBSERVAÇÃO: A notação utilizada para o comprimento da janela é N , onde $N=2M+1$

Resposta ao Impulso dos Filtros Ideais

1 - Filtro Passa Baixa Ideal

$$H_{LP}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < \omega \leq \pi \end{cases} \quad \text{Resposta em frequência do filtro ideal}$$

$$h_{LP}[n] = \frac{\text{sen}(\omega_c n)}{\pi \cdot n} \quad -\infty \leq n \leq \infty \quad \text{Resposta ao impulso do filtro ideal}$$

Truncando a resposta ao impulso entre $-M \leq n \leq M$ obtemos uma seqüência finita não causal de comprimento $N=2M+1$, que quando deslocada para a direita torna-se os coeficientes do filtro FIR causal, com fase linear.

Resposta ao impulso do filtro projetado (real):

$$\hat{h}_{LP}[n] = h_{LP}[n] \cdot w[n] = \frac{\text{sen}(\omega_c (n-M))}{\pi \cdot (n-M)} \quad 0 \leq n \leq N-1$$

$$w[n] = \text{janela simétrica em relação a } M \text{ entre } 0 \leq n \leq N-1$$

Resposta em frequência do filtro projetado:

$$\hat{H}_{LP}(e^{j\omega}) = \{H_{LP}(e^{j\omega}) * W(e^{j\omega})\} \cdot e^{-jM\omega} \quad (\text{convolução})$$

2 - Filtro Passa Alta Ideal

$$h_{HP}[n] = \begin{cases} 1 - \frac{w_c}{\pi}, & n = 0 \\ -\frac{\text{sen}(w_c \cdot n)}{\pi \cdot n}, & |n| > 0 \end{cases} \quad \text{Resposta ao impulso ideal}$$

3 - Filtro Passa Faixa Ideal

$$h_{BP}[n] = \frac{\text{sen}(w_{c2} \cdot n)}{\pi \cdot n} - \frac{\text{sen}(w_{c1} \cdot n)}{\pi \cdot n}, \quad |n| \geq 0 \quad \text{Resposta ao impulso ideal}$$

4 - Filtro Rejeita Faixa Ideal

$$h_{BS}[n] = \begin{cases} 1 - \frac{(w_{c2} - w_{c1})}{\pi}, & n = 0 \\ \frac{\text{sen}(w_{c1} \cdot n)}{\pi \cdot n} - \frac{\text{sen}(w_{c2} \cdot n)}{\pi \cdot n}, & |n| > 0 \end{cases} \quad \text{Resposta ao impulso ideal}$$

Estes métodos são para filtros com 2 níveis de amplitude. O método pode ser generalizado para filtros FIR multiníveis (multibanda).

Tipos de Janelas Fixas

1 – Janela Retangular

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases} \quad W(e^{jw}) = \left[\frac{\text{sen}\left(\frac{w \cdot N}{2}\right)}{\text{sen}\left(\frac{w}{2}\right)} \right] \cdot e^{-jw \frac{N-1}{2}} = W_r(w) \cdot e^{-jw \frac{N-1}{2}}$$

2 – Janela de Bartlett

$$w[n] = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} \leq n \leq N-1 \end{cases}$$

3 – Janela de Hanning

$$w[n] = \begin{cases} 0,5 \cdot \left[1 - \cos\left(\frac{2\pi n}{N-1}\right) \right], & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

4 – Janela de Hamming

$$w[n] = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

5 – Janela de Blackman

$$w[n] = \begin{cases} 0,42 - 0,5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

Os parâmetros das janelas que definem o desempenho do filtro FIR são interpretados analisando-se o espectro da janela.

Considere os seguintes parâmetros do espectro da janela:

Δ_{ML} = largura do lóbulo principal do espectro da janela

A_{SL} = diferença em dB entre as amplitudes do lóbulo lateral mais largo e o lóbulo principal (nível relativo dos lóbulos laterais)

- Para larguras pequenas do lóbulo principal temos transições mais rápidas na resposta do filtro.
- Para reduzirmos o ripple (oscilação) na banda passante e na banda de corte, a área sob os lóbulos laterais deve ser pequena.
- Para as janelas fixas (retangular/hanning/hamming/blackman/bartlett) o valor do ripple não depende do comprimento da janela.
- A frequência de corte, w_c , é essencialmente constante
- A largura da faixa (banda) de transição do filtro é dada por

$$\Delta w \approx \frac{c}{M} \quad c = \text{constante}$$

A tabela abaixo ilustra o desempenho entre as janelas fixas para $w_c = 0,4\pi$ e $M = 128$:

Tipo da Janela	Largura do Lóbulo Principal Δ_{ML}	Nível Relativo do Lóbulo Lateral A_{SL}	Atenuação Mínima Fora da Faixa	$\Delta w = w_p - w_s \approx \frac{c}{M}$
Retangular	$4\pi/(2M + 1)$	13,3 dB	20,9 dB	$0,92\pi/M$
Hanning	$8\pi/(2M + 1)$	31,5 dB	43,9 dB	$3,11\pi/M$
Hamming	$8\pi/(2M + 1)$	42,7 dB	54,5 dB	$3,32\pi/M$
Blackman	$12\pi/(2M + 1)$	58,1 dB	75,3 dB	$5,56\pi/M$

Método de Projeto do Filtro FIR

- 1) Determinar a frequência de corte $w_c = \frac{w_p + w_s}{2}$
- 2) Estimar o valor de M usando a relação $\Delta w \approx \frac{c}{M}$

Janelas Ajustáveis

Nas janelas fixas o ripple é fixo, mas com as janelas ajustáveis é possível controlar o ripple.

1 – Janela de Dolph-Chebyshev

$$w[n] = \frac{1}{2M+1} \left[\frac{1}{\gamma} + 2 \cdot \sum_{k=1}^M T_k \left(\beta \cdot \cos \left(\frac{k\pi}{2M+1} \right) \right) \cdot \cos \left(\frac{2\pi kn}{2M+1} \right) \right]$$

$$\gamma = \frac{\text{Amplitude do lóbulo lateral}}{\text{Amplitude do lóbulo principal}} \quad \beta = \cosh \left(\frac{1}{2M} \cdot \cosh^{-1} \left(\frac{1}{\gamma} \right) \right)$$

$$T_l(x) = \begin{cases} \cos(l \cdot \cos^{-1} x), & |x| \leq 1 \\ \cosh(l \cdot \cosh^{-1} x), & |x| > 1 \end{cases}$$

2 – Janela de Kaiser

$$w[n] = \frac{I_0 \left\{ \beta \sqrt{1 - (n/M)^2} \right\}}{I_0(\beta)} \quad -M \leq n \leq M$$

$I_0(u)$ = função de Bessel modificada de ordem zero

$$I_0(u) = 1 + \sum_{r=1}^{\infty} \left[\frac{(u/2)^r}{r!} \right]^2$$

β = parâmetro ajustável

$$\beta = \begin{cases} 0,1102 (\alpha_s - 8,7) & \alpha_s > 50 \\ 0,5842 (\alpha_s - 21)^{0,4} + 0,07886 (\alpha_s - 21) & 21 \leq \alpha_s \leq 50 \\ 0 & \alpha_s < 21 \end{cases}$$

$$N \approx \begin{cases} \frac{\alpha_s - 7,95}{14,36 \Delta f} & \alpha_s > 21 \\ \frac{0,9222}{\Delta f} & \alpha_s \leq 21 \end{cases}$$

$$w_c = \frac{(w_p + w_s)}{2}$$

$$\Delta f = \frac{(w_s - w_p)}{2\pi}$$

$$\alpha_p = -20 \cdot \log_{10}(1 - \delta_p) \text{ dB}$$

$$\alpha_s = -20 \cdot \log_{10}(\delta_s) \text{ dB}$$

α_s = atenuação mínima na banda de corte (dB)

δ_s = ripple na banda de corte

α_p = ripple na banda passante (dB)

$1 - \delta_p$ = ripple na banda passante

Exercício - Projete um filtro digital FIR passa baixa, usando a **janela de Kaiser**. As especificações do filtro são $w_p = 0,3\pi$, $w_s = 0,4\pi$ e $\alpha_s = 50$ dB.

EXEMPLOS DE APLICAÇÕES

Filtro de Eco: Um filtro de eco possui uma equação de diferenças no formato:

$$y[n] = x[n] + a \cdot x[n - D]$$

A função de transferência e a resposta ao impulso do filtro são:

$$H(z) = 1 + a \cdot z^{-D}$$

$$h[n] = \delta[n] + a \cdot \delta[n - D]$$

Escolha o número de amostras, D , que o sinal deve ser atrasado, de tal forma que o sinal seja atrasado de 0,125s a especifique $a=0,5$.

Trace a resposta em frequência do filtro usando o comando *freqz* do Matlab.

Execute o experimento com valores de D cada vez menores e escute o sinal com eco ate que não seja mais perceptível distinguir do sinal original. Calcule a quantos milisegundos de atraso este valor corresponde.

Filtro Pente (Comb Filter):

Implemente um filtro que adicione 3 ecos sucessivos, que possui a equação de diferenças no formato:

$$y[n] = x[n] + a \cdot x[n - D] + a^2 \cdot x[n - 2D] + a^3 \cdot x[n - 3D]$$

A função de transferência e a resposta ao impulso do filtro são:

$$H(z) = 1 + a \cdot z^{-D} + a^2 \cdot z^{-2D} + a^3 \cdot z^{-3D}$$

Implemente a forma não recursiva deste filtro escolhendo D tal que o atraso seja 0,2s. Utilize $a=0,5$ and trace a resposta em frequência deste filtro. Mantendo o mesmo atraso, D , escolha $a=0,2$ e rode o programa novamente. Que efeito você escuta ? Repita para $a=0,1$.

Filtro de Reverberação:

Se uma quantidade infinita de ecos for adicionada ao sinal, o efeito imita a reverberação natural de uma sala e fornece o filtro pente (comb filter) IIR ou reverberador. A equação de diferenças e a função de transferências são:

$$y[n] = x[n] + a \cdot y[n - D]$$

$$H(z) = \frac{1}{1 - a \cdot z^{-D}}$$

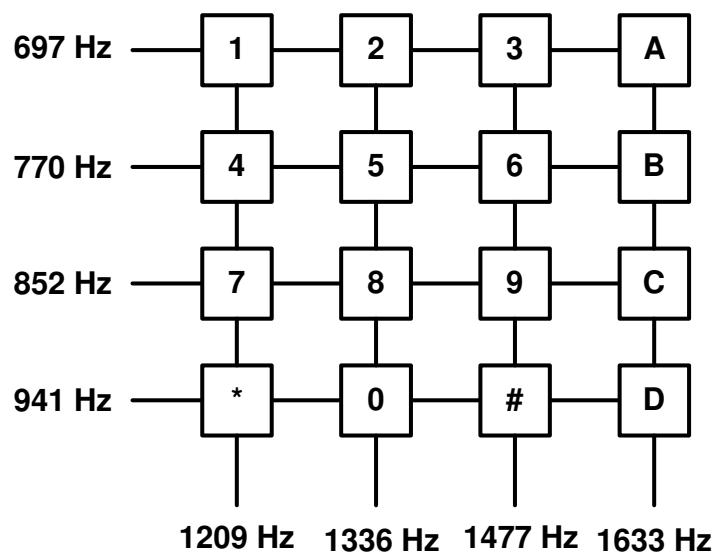
Novamente, escolha D tal que o atraso seja 0,125s, especifique $a=0,5$ e trace o gráfico da resposta em frequência do filtro reverberador. Execute o programa com o novo coeficiente $a=0,9$. Trace a resposta ao impulso. Repita para $a=0,25$. Discuta o efeito de aumentar ou diminuir a .

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 4 – APLICAÇÕES DO PROCESSAMENTO DIGITAL DE SINAIS
SÍNTESE E DETECÇÃO DTMF
PROF. MARCELO PELLEZ

(DTMF) DUAL-TONE MULTIFREQUENCY

N=205		F _T = 8 kHz	
Tom de frequência (Hz)	Valor exato de k	Valor inteiro de k mais próximo	Erro absoluto em k
697	17,861	18	0,139
770	19,731	20	0,269
852	21,833	22	0,167
941	24,113	24	0,113
1209	30,981	31	0,019
1336	34,235	34	0,235
1477	37,848	38	0,152
1633	41,846	42	0,154

N=201		F _T = 8 kHz	
Segunda harmônica (Hz)	Valor exato de k	Valor inteiro de k mais próximo	Erro absoluto em k
1394	35,024	35	0,024
1540	38,692	39	0,308
1704	42,813	43	0,187
1882	47,285	47	0,285
2418	60,752	61	0,248
2672	67,134	67	0,134
2954	74,219	74	0,219
3266	82,058	82	0,058



Fórmulas do cálculo da DFT de N pontos:

$$X(e^{j\omega}) \Rightarrow \text{DTFT}$$

$$X[k] = X(e^{j\omega}) \Big|_{\omega_k = \frac{2\pi k}{N}} \quad k = 0, 1, \dots, N-1$$

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} \quad k = 0, 1, \dots, N-1 \quad W_N = \exp\left(-\frac{j2\pi}{N}\right)$$

1 – No sistema DTMF são utilizadas duas frequências distintas para representar cada dígito. O programa a seguir calcula a DFT de cada frequência. Podemos observar que existe um espalhamento da energia em torno da frequência central (leakage).

$$f_k = \frac{k F_s}{N}, \quad k = 0, 1, \dots, N-1$$

```
% Amostras selecionadas da DFT para
% cada um dos tons de sinais DTMF
clear; clc;
close all;
fs=8000;
n=0:204;
f=[697 770 852 941 1209 1336 1447 1633];
escala=[[10 25]; [10 25]; [15 30]; [15 30]; [25 40]; [25 40]; [35 50]; [35 50]];
for i=1:8
figure(i);
x=sin(2*pi*f(i)*n*(1/fs));
X=fft(x);
stem(n,abs(X));
label=num2str(f(i));
title(strcat(label,' Hz'));
ylabel('|X[k]|');
xlabel('n')
axis([escala(i,1) escala(i,2) 0 110]);
end
```

2 – O programa a seguir sintetiza, analisa e decodifica os tons de frequência de discagem. Os tons transmitidos são enviados para o conversor D/A e podem ser ouvidos através da placa de som.

```
% Funcao para calcular 1 amostra da DFT
% XF=gfft(x,N,k)
% x = sequencia de entrada de comprimento <= N
% N = comprimento da DFT
% k = numero da amostra
% XF = amostra desejada da DFT

function XF=gfft(x,N,k)
if length(x)<N
    xe=[x zeros(1,N-length(x))];
else
    xe=x;
end

x1=[xe 0];
d1=2*cos(2*pi*k/N); W=exp(-i*2*pi*k/N);
y=filter(1,[1 -d1 1],x1);
XF=y(N+1)-W*y(N);
```



```
% DUAL-TONE MULTIFREQUENCY TONE DETECTION

clear; clc;

fs=8000;
n=0:204;
n1=0:319;
loop=0;

while loop==0,

    d=input('Digite Numero:');
    symbol=abs(d);

    if symbol==1
        f1=697; f2=1209;
    end
    if symbol==2
        f1=697; f2=1336;
    end
    if symbol==3
        f1=697; f2=1477;
    end
    if symbol==4
        f1=770; f2=1209;
    end
    if symbol==5
        f1=770; f2=1336;
    end
    if symbol==6
        f1=770; f2=1477;
    end
    if symbol==7
        f1=852; f2=1209;
    end
    if symbol==8
        f1=852; f2=1336;
    end
    if symbol==9
        f1=852; f2=1477;
    end
    if symbol==0
        f1=941; f2=1336;
    end

    y=sin(2*pi*f1*n1*(1/fs))+sin(2*pi*f2*n1*(1/fs));
    sound(y, fs);

    x=sin(2*pi*f1*n*(1/fs))+sin(2*pi*f2*n*(1/fs));

    k=[18 20 22 24 31 34 38 42];

    for m=1:8;
        Fx(m)=gfft(x, 205, k(m));
        val=abs(Fx);
    end
    stem(k, val); grid;
    limit=8100;

    if val(1)*val(5)>limit
        disp('Tecla Apertada = 1');
    end
end
```

```

if val(1)*val(6)>limit
    disp('Tecla Apertada = 2');
end

if val(1)*val(7)>limit
    disp('Tecla Apertada = 3');
end

if val(2)*val(5)>limit
    disp('Tecla Apertada = 4');
end

if val(2)*val(6)>limit
    disp('Tecla Apertada = 5');
end

if val(2)*val(7)>limit
    disp('Tecla Apertada = 6');
end

if val(3)*val(5)>limit
    disp('Tecla Apertada = 7');
end

if val(3)*val(6)>limit
    disp('Tecla Apertada = 8');
end

if val(3)*val(7)>limit
    disp('Tecla Apertada = 9');
end

if val(4)*val(6)>limit
    disp('Tecla Apertada = 0');
end

end

```

3 – No programa a seguir analisamos o efeito do comprimento da DFT na análise espectral. O sinal analisado no domínio da frequência é

$$x[n] = 0,5 \sin(2\pi f_1 n) + \sin(2\pi f_2 n), \quad 0 \leq n \leq N-1$$

$$f_1 = 0,22 \quad f_2 = 0,34$$

Calcule a DFT do sinal de comprimento $N=16$. Utilize os seguintes comprimentos para a DFT:
22. R=16 R=32 R=64 R=128

```

clear; clc;
close all;
N=input('Comprimento do Sinal: ');
R=input('Comprimento da DFT: ');
f1=0.22; f2=0.34;
n=0:N-1;
x=0.5*sin(2*pi*f1*n)+sin(2*pi*f2*n);
X=fft(x,R);
figure(1);
k=0:R-1;
stem(k,abs(X)); grid;
title(['N = ', num2str(N), ' R = ', num2str(R)]);
xlabel('k'); ylabel('|X[k]|');

```

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 5 – TRANSFORMADA-Z
PROF. MARCELO PELLEZ

A Transformada de Fourier em Tempo Discreto (DTFT) de um sistema linear invariante no tempo, com resposta ao impulso $h[n]$, é denominada de resposta em frequência do sistema, $H(e^{j\omega})$. Contudo, existem duas restrições para a abordagem da DTFT. A primeira é que para muitos sinais úteis na prática como $u[n]$ e $n \cdot u[n]$ a DTFT não existe. A segunda é que a resposta transitória de um sistema devido a condições iniciais ou alterações da entrada não podem ser calculadas usando a abordagem da DTFT. Portanto utilizamos uma extensão da DTFT denominada de transformada Z:

$$H(Z) = \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n} \quad \text{Transformada-Z}$$

$$h[n] = \frac{1}{2\pi j} \oint_C H(Z) z^{n-1} dz \quad \text{Transformada-Z Inversa}$$

A transformada z pode ser escrita como uma função racional em z^{-1} , ou seja, como uma razão de polinômios em z^{-1} :

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{a_0 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \quad R_{h^-} < |z| < R_{h^+}$$

$$H(z) = \frac{\tilde{b}_0 + \tilde{b}_1 z^{-1} + \dots + \tilde{b}_{N-1} z^{-(N-1)}}{1 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} + \underbrace{\sum_{k=0}^{M-N} C_k \cdot z^{-k}}_{M \geq N}$$

A transformada-z inversa pode ser obtida através da expansão em frações parciais:

$$H(z) = \sum_{k=1}^N \frac{R_k}{1 - p_k z^{-1}} + \underbrace{\sum_{k=0}^{M-N} C_k \cdot z^{-k}}_{M \geq N}$$

$$R_k = \left. \frac{\tilde{b}_0 + \tilde{b}_1 z^{-1} + \dots + \tilde{b}_{N-1} z^{-(N-1)}}{1 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \cdot (1 - p_k z^{-1}) \right|_{z=p_k}$$

$$h[n] = \sum_{l=1}^N R_l \cdot Z^{-1} \left\{ \frac{1}{1 - p_l z^{-1}} \right\} + \underbrace{\sum_{k=0}^{M-N} C_k \cdot \delta(n-k)}_{M \geq N}$$

Teorema 1: Um sistema LTI é estável se e somente se o círculo de raio unitário está dentro da região de convergência (ROC) de $H(z)$

Teorema 2: Um sistema LTI causal é estável se e somente se a função de transferência $H(z)$ possui todos os pólos dentro do círculo de raio unitário.

Relação entre a **função de transferência** e a **equação de diferenças**:

$$Y(Z) = H(Z) \cdot X(Z)$$

$$y[n] + \sum_{k=1}^N a_k \cdot y[n-k] = \sum_{l=1}^M b_l \cdot x[n-l] \quad \text{Equação de Diferenças}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{l=0}^M b_l z^{-l}}{1 + \sum_{k=0}^N a_k z^{-k}} = \frac{B(z)}{A(z)} \quad \text{Função de Transferência}$$

$$H(z) = \frac{b_0 z^{-M} \left(z^M + \dots + \frac{b_M}{b_0} \right)}{z^{-N} \left(z^N + \dots + a_N \right)} \quad H(z) = b_0 z^{(N-M)} \frac{\prod_{l=1}^{N-1} (z - z_l)}{\prod_{k=1}^N (z - p_k)}$$

Exercício 1 – Determine a expansão em frações parciais da transformada-z, para a seguinte seqüência:

$$H(z) = \frac{z}{3z^2 - 4z + 1}$$

Utilize a função **residuez** do Matlab.

Exercício 2 – Dado um sistema LTI causal com a seguinte função de transferência:

$$H(z) = \frac{z + 1}{z^2 - 0,9z + 0,81}$$

Determine:

- Expressão da resposta em frequência
- Expressão da equação de diferenças
- Escreva $H(z)$ na forma fatorada utilizando expansão em frações parciais (**residuez**)
- Determine os zeros e pólos (**roots**)
- Faça o diagrama de pólos e zeros (**zplane**)

Exercício 3 – Um sistema LTI causal é descrito pela seguinte equação de diferenças:

$$y[n] = 0,81 \cdot y[n-2] + x[n] - x[n-2]$$

Determine:

- Expressão da função de transferência, $H(z)$
- Traçar o gráfico da resposta em frequência, $H(e^{j\omega})$ (módulo/fase) (**freqz**)
- Escreva $H(z)$ na forma fatorada utilizando expansão em frações parciais (**residuez**)
- Traçar o diagrama de pólos e zeros (**zplane**)

Exercício 4 – Expresse a seguinte transformada- z na forma fatorada (produto de seções de segunda ordem), trace o diagrama de pólos e zeros e determine as regiões de convergência.

$$H(z) = \frac{2z^4 + 16z^3 + 44z^2 + 56z + 32}{3z^4 + 3z^3 - 15z^2 + 18z - 12}$$

Função do Matlab: `tf2sos`

$$sos = \begin{bmatrix} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & a_{0L} & a_{1L} & a_{2L} \end{bmatrix} \quad H(z) = \prod_{k=1}^L \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{a_{0k} + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

Exercício 5 – Determine a forma racional da transformada- z a partir da localização de seus zeros e pólos.

- Os zeros são: $z_1 = 0.21$, $z_2 = 3.14$, $z_3 = -0.3 + j0.5$ e $z_4 = -0.3 - j0.5$.
- Os pólos são: $p_1 = -0.45$, $p_2 = 0.67$, $p_3 = 0.81 + j0.72$ e $p_4 = 0.81 - j0.72$
- O ganho constante é $k = 2.2$

Função do Matlab: `zp2tf`

**PROJETO DE FILTROS FIR
TIPOS DE FASE LINEAR PARA FILTROS FIR**

Fase Linear \Rightarrow Resposta ao impulso simétrica

TIPO 1 – SIMÉTRICO - COMPRIMENTO ÍMPAR

$$h[n] = h[N - n], \quad 0 \leq n \leq N$$

$$H(z) = \sum_{n=0}^N h[n] z^{-n} \quad \Rightarrow \text{Função de Transferência}$$

Exemplo com $N=8$

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8}$$

$$\left. \begin{array}{l} h[0] = h[8] \\ h[1] = h[7] \\ h[2] = h[6] \\ h[3] = h[5] \end{array} \right\} \text{SIMETRIA}$$

$$H(z) = z^{-4} \{ h[0] (z^4 + z^{-4}) + h[1] (z^3 + z^{-3}) + h[2] (z^2 + z^{-2}) + h[3] (z + z^{-1}) + h[4] \}$$

$$H(e^{jw}) = e^{-j4w} \cdot \{ 2h[0] \cos(4w) + 2h[1] \cos(3w) + 2h[2] \cos(2w) + 2h[3] \cos(w) + h[4] \}$$

$$\tau(w) = 4$$

$$H(e^{jw}) = e^{-j\frac{Nw}{2}} \left\{ \sum_{n=0}^{N/2} a[n] \cos(wn) \right\}$$

$$a[0] = h\left[\frac{N}{2}\right] \quad a[n] = 2h\left[\frac{N}{2} - n\right], \quad 1 \leq n \leq \frac{N}{2}$$

TIPO 2 – SIMÉTRICO - COMPRIMENTO PAR

Exemplo para $N=7$

$$H(z) = h[0] (1 + z^{-7}) + h[1] (z^{-1} + z^{-6}) + h[2] (z^{-2} + z^{-5}) + h[3] (z^{-3} + z^{-4})$$

$$H(z) = z^{-7/2} \{h[0] (z^{7/2} + z^{-7/2}) + h[1] (z^{5/2} + z^{-5/2}) + h[2] (z^{3/2} + z^{-3/2}) + h[3] (z^{1/2} + z^{-1/2})\}$$

$$H(e^{jw}) = e^{-j\frac{7w}{2}} \left\{ 2h[0] \cos\left(\frac{7w}{2}\right) + 2h[1] \cos\left(\frac{5w}{2}\right) + 2h[2] \cos\left(\frac{3w}{2}\right) + 2h[3] \cos\left(\frac{w}{2}\right) \right\}$$

$$\tau(w) = \frac{7}{2}$$

$$H(e^{jw}) = e^{-jNw/2} \left\{ \sum_{n=1}^{(N+1)/2} b[n] \cos\left[w \left(n - \frac{1}{2} \right) \right] \right\}$$

$$b[n] = 2h \left[\frac{N+1}{2} - n \right] \quad 1 \leq n \leq \frac{N+1}{2}$$

TIPO 3 – ANTISIMÉTRICO – COMPRIMENTO ÍMPAR

Características:

- Atraso de grupo constante
- Resposta em fase quase linear

$$h[n] = -h[N - n] \quad 0 \leq n \leq N$$

Exemplo para $N=8$

$$H(z) = z^{-4} \{h[0](z^4 - z^{-4}) + h[1](z^3 - z^{-3}) + h[2](z^2 - z^{-2}) + h[3](z^1 - z^{-1})\}$$

$$h[N/2] = 0$$

$$H(e^{jw}) = e^{-j4w} e^{j\frac{\pi}{2}} \{2h[0] \text{sen } 4w + 2h[1] \text{sen } 3w + 2h[2] \text{sen } 2w + 2h[3] \text{sen } w\}$$

$$\tau(w) = 4$$

$$H(e^{jw}) = e^{-jNw/2} \cdot e^{j\pi/2} \cdot \left\{ \sum_{n=1}^{N/2} c[n] \cdot \text{sen}(wn) \right\}$$

$$c[n] = 2h\left[\frac{N}{2} - n\right], \quad 1 \leq n \leq \frac{N}{2}$$

TIPO 4 – ANTISIMÉTRICO – COMPRIMENTO PAR

Exemplo para $N=7$

$$H(z) = z^{-\frac{7}{2}} \left\{ h[0] \left(z^{\frac{7}{2}} - z^{-\frac{7}{2}} \right) + h[1] \left(z^{\frac{5}{2}} - z^{-\frac{5}{2}} \right) + h[2] \left(z^{\frac{3}{2}} - z^{-\frac{3}{2}} \right) + h[3] \left(z^{\frac{1}{2}} - z^{-\frac{1}{2}} \right) \right\}$$

$$H(e^{jw}) = e^{-j\frac{7w}{2}} e^{j\frac{\pi}{2}} \left\{ 2h[0] \text{sen} \left(\frac{7w}{2} \right) + 2h[1] \text{sen} \left(\frac{5w}{2} \right) + 2h[2] \text{sen} \left(\frac{3w}{2} \right) + 2h[3] \text{sen} \left(\frac{w}{2} \right) \right\}$$

$$\tau(w) = \frac{7}{2}$$

$$H(e^{jw}) = e^{-j\frac{Nw}{2}} e^{j\frac{\pi}{2}} \left\{ \sum_{n=1}^{(N+1)/2} d[n] \text{sen} \left(w \left(n - \frac{1}{2} \right) \right) \right\}$$

$$d[n] = 2 \cdot h \left[\frac{N+1}{2} - n \right] \quad 1 \leq n \leq \frac{N+1}{2}$$

$$H(e^{jw}) = e^{-j\frac{Nw}{2}} \cdot e^{j\beta} \cdot \tilde{H}(w) \quad \beta = 0 \text{ ou } \frac{\pi}{2}$$

$\tilde{H}(w)$ = Resposta em Amplitude

LAB. DSP 1 – Introdução ao Kit TMS320C5416

Documentação para o Laboratório

Nos experimentos de laboratório utilizaremos o TMS320C5416 DSP STARTER KIT da Texas Instruments (TI).

As cópias eletrônicas destes documentos são encontradas abaixo.

NÃO IMPRIMA estes documentos, apenas utilize na forma eletrônica. Para maiores informações sobre a família de

processadores TMS320C54xx, consulte o site da TI. (www.ti.com)

Documentação Técnica

TMS320C5416 Digital Signal Processor Data Sheet (C:\ti\docs\hlp\C5416DSK.HLP)

TMS320C54x DSP Functional Overview ([SPRU307A](#))

TMS320C54x Code Composer Studio Tutorial ([SPRU327C](#))

TMS320C54x Code Composer Studio User's Guide ([SPRU328B](#))

TMS320C54x Assembly Language Tools User's Guide ([SPRU102E](#))

TMS320C54x Evaluation Module Technical Reference ([SPRU135](#))

TMS320C54x Optimizing C/C++ Compiler ([SPRU103f](#))

TMS320C54x DSP Ref. Set Vol. 2: Mnemonic Instruction Set ([SPRU172C](#))

Atividades de Laboratório

Atividade 1 - Abrir o arquivo de help <c:\ti2\docs\hlp\c5416dsk.hlp>

Ler a descrição geral do kit TMS320C5416 e executar os exemplos do tutorial que utilizam o DSP/BIOS (Real-Time Multitasking Kernel):

a) O exemplo do LED reside no diretório <c:\ti2\examples\dsk5416\bsl\led>

Faça uma cópia de todo o diretório para o diretório de trabalho <c:\ti2\myprojects> a partir do qual você deve trabalhar.

b) O exemplo de geração de TOM reside no diretório <c:\ti2\examples\dsk5416\bsl\tone>

Faça uma cópia de todo o diretório para o diretório de trabalho <c:\ti2\myprojects> a partir do qual você deve trabalhar.

b.1) Identificar a frequência de amostragem utilizada pelo A/D - D/A

b.2) Calcular a frequência normalizada " $w=2*\pi*f$ "

b.3) Calcular o período do sinal em tempo discreto em número de amostras

b.4) Calcular o número de amostras por período do sinal analógico

b.5) Identifique no **Codec API Summary** quais as frequências de amostragem padrões que podem ser utilizadas

Vá para o item Software Overview --> BSL API Summary --> Codec API

Este tópico é importante para os trabalhos futuros de laboratório para entender como utilizar a

API

para escrever programas com I/O de streams de áudio para aplicações do DSP.

Atividade 2 - Ler os capítulos 2, 3, 4 e 6 do help do CCS DSK: Code Composer Studio Overview.

Capítulo 2 - Developing a Simple Program

Capítulo 3 - Developing a DSP/BIOS Program

Capítulo 4 - Testing Algorithms and Data from a file

Capítulo 6 - Analyzing Real-Time Behavior

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 6 – PROJETO DE FILTROS DIGITAIS FIR
 PROF. MARCELO PELLEZZ

Vantagens dos filtros FIR:

- A resposta em fase pode ser exatamente linear
- São fáceis de implementar porque não há problemas de estabilidade
- A DFT pode ser usada na implementação

Vantagens da resposta em fase linear:

- A implementação envolve apenas aritmética real e não complexa
- Não existe distorção de atraso e sim apenas um atraso fixo entre entrada/saída
- Para filtros de comprimento N (ordem $N-1$) o número de operações é da ordem de $N/2$

Metodologia de Projeto – Método das Janelas

A ênfase do método está em selecionar uma janela apropriada e um filtro ideal:

- a) Selecionar um filtro ideal (não causal), que possui resposta ao impulso infinita
- b) Truncar a resposta ao impulso (usando uma janela) para obter um filtro FIR causal
- c) Deslocar o início do vetor de amostras, $h[n]$, para iniciar em $n=0$
(equivale a associar uma fase linear)

OBSERVAÇÃO: A notação utilizada para o comprimento da janela é N , onde $N=2M+1$

Resposta ao Impulso dos Filtros Ideais

1 - Filtro Passa Baixa Ideal

$$H_{LP}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < \omega \leq \pi \end{cases} \quad \text{Resposta em frequência do filtro ideal}$$

$$h_{LP}[n] = \frac{\text{sen}(\omega_c n)}{\pi \cdot n} \quad -\infty \leq n \leq \infty \quad \text{Resposta ao impulso ideal}$$

Truncando a resposta ao impulso entre $-M \leq n \leq M$ obtemos uma seqüência finita não causal de comprimento $N=2M+1$, que quando deslocada para a direita torna-se os coeficientes do filtro FIR causal, com fase linear.

Resposta ao impulso do filtro projetado (real)

$$\hat{h}_{LP}[n] = h_{LP}[n] \cdot w[n] = \frac{\text{sen}(\omega_c (n-M))}{\pi \cdot (n-M)} \quad 0 \leq n \leq N-1$$

$w[n]$ = janela simétrica em relação a M entre $0 \leq n \leq N-1$

Resposta em frequência do filtro projetado

$$\hat{H}_{LP}(e^{j\omega}) = \{H_{LP}(e^{j\omega}) * W(e^{j\omega})\} \cdot e^{-jM\omega}$$

2 - Filtro Passa Alta Ideal

$$h_{HP}[n] = \begin{cases} 1 - \frac{w_c}{\pi}, & n = 0 \\ -\frac{\text{sen}(w_c \cdot n)}{\pi \cdot n}, & |n| > 0 \end{cases} \quad \text{Resposta ao impulso ideal}$$

3 - Filtro Passa Faixa Ideal

$$h_{BP}[n] = \frac{\text{sen}(w_{c2} \cdot n)}{\pi \cdot n} - \frac{\text{sen}(w_{c1} \cdot n)}{\pi \cdot n}, \quad |n| \geq 0 \quad \text{Resposta ao impulso ideal}$$

4 - Filtro Rejeita Faixa Ideal

$$h_{BS}[n] = \begin{cases} 1 - \frac{(w_{c2} - w_{c1})}{\pi}, & n = 0 \\ \frac{\text{sen}(w_{c1} \cdot n)}{\pi \cdot n} - \frac{\text{sen}(w_{c2} \cdot n)}{\pi \cdot n}, & |n| > 0 \end{cases} \quad \text{Resposta ao impulso ideal}$$

Estes métodos são para filtros com 2 níveis de amplitude. O método pode ser generalizado para filtros FIR multiníveis.

Tipos de Janelas Fixas**1 - Janela Retangular**

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

2 - Janela de Bartlett

$$w[n] = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} \leq n \leq N-1 \end{cases}$$

3 - Janela de Hanning

$$w[n] = \begin{cases} 0,5 \left[1 - \cos\left(\frac{2\pi n}{N-1}\right) \right], & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

4 - Janela de Hamming

$$w[n] = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

5 – Janela de Blackman

$$w[n] = \begin{cases} 0,42 - 0,5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{fora} \end{cases}$$

Os parâmetros das janelas que definem o desempenho do filtro FIR são interpretados analisando-se o espectro da janela.

Considere os seguintes parâmetros do espectro da janela:

Δ_{ML} = largura do lóbulo principal do espectro da janela

A_{SL} = diferença em dB entre as amplitudes do lóbulo lateral mais largo e o lóbulo principal (nível relativo dos lóbulos laterais)

Para larguras pequenas do lóbulo principal temos transições mais rápidas na resposta do filtro.

Para reduzirmos o ripple na banda passante e na banda de corte, a área sob os lóbulos laterais deve ser pequena.

Para as janelas fixas (retangular/hanning/hamming/blackman/Bartlett) o valor do ripple não depende do comprimento da janela.

A frequência de corte, w_c , é essencialmente constante

A largura de faixa de transição é dada por

$$\Delta w \approx \frac{c}{M} \quad c = \text{constante}$$

Comandos do MATLAB para o projeto das janelas:

`w=blackman(N)`

`w=hamming(N)`

`w=hanning(N)`

`w=chebwin(N,R)`

`w=kaiser(N,BETA)`

N=comprimento da janela

L = N-1 = ordem do filtro

b=fir1(L,Wn)

b=fir1(L,Wn,'filtertype')

b=fir1(L,Wn>window)

b=fir1(L,Wn,'filtretype',window)

Wn = Frequência de Corte Normalizada (0 <= Wn <=1)

1 – Neste exemplo você irá projetar apenas a janela (tipo blackman), $w[n]$, e visualizar o seu espectro:

```
clear; clc;
N=input('Comprimento da Janela:');
w=blackman(N)
disp(w)
[h,omega]=freqz(w,1,256);
mag=20*log10(abs(h));
figure(1);
plot(omega/pi,mag); grid;
ylabel('Frequencia Normalizada');
xlabel('Ganho (dB)'); title('Janela');
```

Analise todos os tipos de janelas fixas e anote os seguintes parâmetros:

Δ_{ML} = largura do lóbulo principal do espectro da janela

A_{SL} = diferença em dB entre a amplitude do lóbulo lateral e o lóbulo principal

TAREFAS

a) Analisar o espectro das janelas e anotar numa tabela os valores da largura do lóbulo principal, Δ_{ML} e a atenuação relativa do lóbulo lateral, A_{SL} .

b) Projetar os seguintes filtros digitais FIR, considerando $F_s=8\text{kHz}$:

- Filtro passa baixa $F_c=2\text{kHz}$ $N=17$
- Filtro passa alta $F_c=2\text{kHz}$ $N=17$
- Filtro passa faixa $F_{c1}=2\text{kHz}$ $F_{c2}=3\text{kHz}$ $N=17$
- Filtro rejeita faixa $F_{c1}=2\text{kHz}$ $F_{c2}=3\text{kHz}$ $N=17$

Traçar o gráfico da resposta em frequência de cada filtro (módulo/fase)

Entregar no final da aula para cada filtro projetado, $h[n]$, $H(z)$ e a equação de diferenças.

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 7 – PROJETO DE FILTROS DIGITAIS FIR
 PROF. MARCELO PELLEZ

Projetar os seguintes filtros digitais e converter os coeficientes para a notação de 16 bits (Q15) utilizada pelo DSP. Os coeficientes devem estar no formato final hexadecimal. Os números negativos devem ser representados utilizando a notação complemento de 2.

Aluno	Tipo de Filtro	Tipo de Janela	Ordem do Filtro	Frequência de Corte	Frequência de Amostragem

n	h[n]	h[n]*32767	Formato Binário	Formato Hexadecimal
0				
1				
:				
N-1				

Ao final da aula devem ser entregues as tabelas com os coeficientes projetados no formato final para implementação no kit DSP TMS320C5416.

PROJETO DE FILTROS DIGITAIS FIR TÉCNICA DA AMOSTRAGEM EM FREQUÊNCIA

A função de transferência é obtida das amostras $H[k]$ da resposta em frequência $H(e^{j\omega})$. Considere a resposta ao impulso, $h[n]$, de um filtro FIR com M amostras. A sua DFT, $H[k]$ também terá M pontos.

$$H(z) = \sum_{n=0}^{M-1} h[n] z^{-n} = \frac{1-z^{-M}}{M} \cdot \sum_{k=0}^{M-1} \frac{H[k]}{1-z^{-1} \cdot e^{j2\pi k/M}}$$

$$H[k] = \sum_{n=0}^{M-1} h[n] e^{-j\frac{2\pi kn}{M}} \quad h[n] = \frac{1}{M} \sum_{k=0}^{M-1} H[k] e^{j\frac{2\pi kn}{M}}$$

$$H(e^{j\omega}) = \frac{1-e^{-j\omega M}}{M} \cdot \sum_{k=0}^{M-1} \frac{H[k]}{1-e^{-j\omega} \cdot e^{j2\pi k/M}}$$

$$H\left(e^{j\frac{2\pi k}{M}}\right) = H[k] = \begin{cases} H[0], & k = 0 \\ H^*[M-k], & k = 1, \dots, M-1 \end{cases}$$

Para filtros FIR com fase linear,

$$h[n] = \pm h[M-1-n], \quad n = 0, 1, \dots, M-1$$

$$H[k] = H_r[k] \cdot e^{j\angle H[k]}$$

$$H_r[k] = \begin{cases} H_r[0], & k = 0 \\ H_r[M-k], & k = 1, \dots, M-1 \end{cases}$$

Para filtros tipo 1 e 2:

$$\angle H[k] = \begin{cases} -\left(\frac{M-1}{2}\right) \cdot \left(\frac{2\pi k}{M}\right), & k = 0, \dots, \left\lfloor \frac{M-1}{2} \right\rfloor \\ +\left(\frac{M-1}{2}\right) \frac{2\pi}{M} (M-k), & k = \left\lfloor \frac{M-1}{2} \right\rfloor + 1, \dots, M-1 \end{cases}$$

Para filtros tipo 3 e 4:

$$\angle H[k] = \begin{cases} \left(\pm \frac{\pi}{2}\right) - \left(\frac{M-1}{2}\right) \cdot \left(\frac{2\pi k}{M}\right), & k = 0, \dots, \left\lfloor \frac{M-1}{2} \right\rfloor \\ -\left(\pm \frac{\pi}{2}\right) + \left(\frac{M-1}{2}\right) \frac{2\pi}{M} (M-k), & k = \left\lfloor \frac{M-1}{2} \right\rfloor + 1, \dots, M-1 \end{cases}$$

$$h[n] = \text{IDFT}\{H[k]\}$$

Dada a resposta ideal do filtro, $H_d(e^{j\omega})$, especificar o valor de M e amostras $H_d(e^{j\omega})$ em M frequências igualmente espaçadas entre 0 e 2π . A resposta do filtro obtido, $H(e^{j\omega})$, é a interpolação das amostras $H[k]$.

- O erro de aproximação, ou seja, a diferença entre a resposta ideal e a projetada, é zero nas frequências de amostragem.
- O erro de aproximação nas outras frequências depende do formato da resposta ideal. Quanto mais abrupto a resposta ideal, maior o erro.
- O erro é maior próximo dos limites das faixas de transição e menor dentro das faixas.

Verificar Gráfico

Métodos de Construção (duas abordagens) {

- Aplicação direta do método (sem considerar o erro de aproximação)
- Projeto ótimo, onde tenta - se minimizar o erro na banda de corte variando - se os valores das amostras na banda de transição.

PROCESSAMENTO DIGITAL DE SINAIS (PDS)
LABORATÓRIO 8 – PROJETO DE FILTROS DIGITAIS FIR
 PROF. MARCELO PELLEZ

Metodologia de Projeto usando Janelas Ajustáveis

A ênfase do método está em selecionar uma janela apropriada e um filtro ideal:

- a) Selecionar um filtro ideal (não causal), que possui resposta ao impulso infinita
- b) Truncar a resposta ao impulso (usando uma janela) para obter um filtro FIR causal
- c) Deslocar o início do vetor de amostras, $h[n]$, para iniciar em $n=0$
(equivale a associar uma fase linear)

OBSERVAÇÃO: A notação utilizada para o comprimento da janela é N , onde $N=2M+1$

Janela Ajustável de kaiser

1 – Considere o projeto de um filtro digital FIR passa baixa, usando a **janela de Kaiser**. As especificações do filtro são $w_p = 0.3\pi$, $w_s = 0.4\pi$ e $\alpha_s = 50$ dB.

A frequência de corte é dada por

$$w_c = \frac{(w_p + w_s)}{2} \quad \Delta f = \frac{(w_s - w_p)}{2\pi} \quad \alpha_s = -20 \cdot \log_{10}(\delta_s) \text{ dB}$$

$$\beta = \begin{cases} 0.1102 (\alpha_s - 8.7) & \alpha_s > 50 \\ 0.5842 (\alpha_s - 21)^{0.4} + 0.07886 (\alpha_s - 21) & 21 \leq \alpha_s \leq 50 \\ 0 & \alpha_s < 21 \end{cases} \quad N \approx \begin{cases} \frac{\alpha_s - 7.95}{14.36 \Delta f} & \alpha_s > 21 \\ \frac{0.9222}{\Delta f} & \alpha_s \leq 21 \end{cases}$$

```
% Geracao da Janela de Kaiser
clear; clc;
N=input('Comprimento da Janela:');
beta=input('Valor de Beta:');
w=kaiser(N,beta);
disp(w)
[h,omega]=freqz(w,1,256);
mag=20*log10(abs(h));
figure(1);
plot(omega/pi,mag); grid;
ylabel('Frequencia Normalizada');
xlabel('Ganho (dB)');
title('Janela de Kaiser');
```

2 – Projete um filtro FIR passa baixa usando a **janela de kaiser** para os mesmos parâmetros do programa anterior.

```
% Projeto do Filtro FIR usando a janela de Kaiser
clear; clc;
N=input('Comprimento da Janela:');
beta=input('Valor de Beta:');
wc==input('Frequencia de corte:');
kw=kaiser(N,beta);
b=fir1(N-1,wc,kw);
[h,omega]=freqz(b,1,256);
mag=20*log10(abs(h));
figure(1);
plot(omega/pi,mag); grid;
ylabel('Frequencia Normalizada');
xlabel('Ganho (dB)');
title('Filtro FIR Passa Baixa');
```

Método de Projeto de Filtros FIR baseado na Amostragem em Frequência

$$b = \text{fir2}(L, f, m)$$

$$b = \text{fir2}(L, f, m, \text{window})$$

$$b = \text{remez}(N, f, m)$$

$$b = \text{remez}(N, f, m, \text{wt})$$

$$b = \text{remez}(N, f, m, \text{'ftype'})$$

$$b = \text{remez}(N, f, m, \text{wt}, \text{'ftype'})$$

3 – Considere o projeto de um filtro FIR de ordem 100 com 3 níveis diferentes de amplitude:

- 0.3 na faixa de frequência de 0 a 0.28
- 1 na faixa de frequência de 0.3 a 0.5
- 0.7 na faixa de frequência de 0.52 a 1

```
% Projeto do Filtro FIR multifaixa usando a janela de Hamming
clear; clc;
fpts=[0 0.28 0.3 0.5 0.52 1];
mval=[0.3 0.3 1 1 0.7 0.7];
b=fir2(100,fpts,mval);
[h,omega]=freqz(b,1,512);
mag=abs(h);
figure(1);
plot(omega/pi,mag); grid;
ylabel('Frequencia Normalizada');
xlabel('Ganho (dB)');
title('Filtro FIR Passa Baixa');
```

4 – Determine a função de transferência e trace a resposta em frequência de um filtro digital FIR passa faixa, com fase linear, utilizando o método da amostragem em frequência. A ordem do filtro deve ser 27, a faixa de passagem de 0.3 a 0.5 e as faixas de corte de 0 a 0.25 e de 0.55 a 1.

$$f = [0 \ 0.25 \ 0.3 \ 0.5 \ 0.55 \ 1]$$

$$m = [0.01 \ 0.01 \ 1 \ 1 \ 0.01 \ 0.01]$$

```

% Projeto de um filtro digital FIR passa faixa
clear; clc;
N=input('Ordem do Filtro:');
f=input('Faixas de Frequencia:');
m=input('Valores de Amplitude:');
b=remez(N,f,m);
disp(b)
[h,w]=freqz(b,1,256);
mag=20*log10(abs(h));
fase=angle(h);
figure(1);
plot(w/pi,mag); grid;
ylabel('Frequencia Normalizada');
xlabel('Ganho (dB)');
title('Filtro Digital de FIR Passa Faixa');
figure(2);
plot(w/pi,fase); grid;
ylabel('Frequencia Normalizada');
xlabel('Fase');
title('Filtro Digital de FIR Passa Faixa');

```

5 – Projete um filtro digital FIR, passa baixa, usando o método da série de Fourier truncada, com as seguintes especificações:

$$\Omega_p = 1 \text{ rad/s} \quad \Omega_s = 2 \text{ rad/s}$$

$$\alpha_p = 0.1 \text{ dB} \quad \alpha_s = 45 \text{ dB}$$

$$\Omega_T = 2\pi F_T = 10 \text{ rad/s} \quad \rightarrow \quad \text{Frequência de amostragem}$$

Use o comando **fir1** do Matlab. Utilize as seguintes janelas: Hamming, Hann, Blackman e Kaiser. Mostre os coeficientes da resposta ao impulso e trace a resposta em frequência dos filtros projetados para cada caso.

LAB. 1 – Introdução ao kit DSP TMS320C5416

Documentação para o Laboratório

Nos experimentos de laboratório utilizaremos o TMS320C5416 DSP STARTER KIT da Texas Instruments (TI).

As cópias eletrônicas destes documentos são encontradas abaixo.

NÃO IMPRIMA estes documentos, apenas utilize na forma eletrônica. Para maiores informações sobre a família de processadores TMS320C54xx, consulte o site da TI. (www.ti.com)

Documentação Técnica

TMS320C5416 Digital Signal Processor Data Sheet (C:\ti\docs\hlp\C5416DSK.HLP)

TMS320C54x DSP Functional Overview ([SPRU307A](#))

TMS320C54x Code Composer Studio Tutorial ([SPRU327C](#))

TMS320C54x Code Composer Studio User's Guide ([SPRU328B](#))

TMS320C54x Assembly Language Tools User's Guide ([SPRU102E](#))

TMS320C54x Evaluation Module Technical Reference ([SPRU135](#))

TMS320C54x Optimizing C/C++ Compiler ([SPRU103f](#))

TMS320C54x DSP Ref. Set Vol. 2: Mnemonic Instruction Set ([SPRU172C](#))

Atividades de Laboratório

Atividade 1 - Abrir o arquivo de help <c:\ti2\docs\hlp\c5416dsk.hlp>

Ler a descrição geral do kit TMS320C5416 e executar os exemplos do tutorial que utilizam o DSP/BIOS (Real-Time Multitasking Kernel):

a) O exemplo do LED reside no diretório <c:\ti2\examples\dsk5416\bs\led>

Faça uma cópia de todo o diretório para o diretório de trabalho <c:\ti2\myprojects> a partir do qual você deve trabalhar.

b) O exemplo de geração de TOM reside no diretório <c:\ti2\examples\dsk5416\bs\lstone>

Faça uma cópia de todo o diretório para o diretório de trabalho <c:\ti2\myprojects> a partir do qual você deve trabalhar.

b.1) Identificar a frequência de amostragem utilizada pelo A/D - D/A

b.2) Calcular a frequência normalizada " $w=2*\pi*f$ "

b.3) Calcular o período do sinal em tempo discreto em número de amostras

b.4) Calcular o número de amostras por período do sinal analógico

b.5) Identifique no **Codec API Summary** quais as frequências de amostragem padrões que podem ser utilizadas

Vá para o item Software Overview --> BSL API Summary --> Codec API

Este tópico é importante para os trabalhos futuros de laboratório para entender como utilizar a API para escrever programas com I/O de streams de áudio para aplicações do DSP.

Atividade 2 - Ler os capítulos 2, 3, 4 e 6 do help do CCS DSK: Code Composer Studio Overview.

Capítulo 2 - Developing a Simple Program

Capítulo 3 - Developing a DSP/BIOS Program

Capítulo 4 - Testing Algorithms and Data from a file

Capítulo 6 - Analyzing Real-Time Behavior

Lab. 2 – TMS320C5416 Laboratório de Efeitos de Áudio

Este laboratório trata de efeitos de áudio. Os algoritmos são implementados usando o DSP de maneira que os arquivos de áudio possam ser processados em tempo real. Vamos implementar um circuito fechado de áudio neste laboratório e escutar um trecho de música do CD player. Na parte 1 usaremos o Matlab para implementação e processamento de áudio (não em tempo real) usando arquivos WAV. Isso permite um bom entendimento dos efeitos de áudio e seus algoritmos. Na parte 2 os algoritmos serão implementados em tempo real usando o KIT DSP TMS320C5416. Estes efeitos de áudio podem ser combinados para gerar efeitos complexos de áudio. Você pode gravar a sua própria voz para testar os algoritmos. Grave no modo mono com 8 bits de resolução na frequência de amostragem de 8kHz e que o arquivo seja salvo no formato WAV.

Equipamentos Necessários:

- PC com Matlab e Toolbox de Processamento Digital de Sinais
- Microfone e alto-falante (fone de ouvido)
- CD player portátil
- KIT DSP TMS320C5416

PARTE 1 – EFEITOS DIGITAIS DE AUDIO NO MATLAB

Filtro de Eco:

Implemente um filtro de eco que possui uma equação de diferenças no formato:

$$y[n] = x[n] + a \cdot x[n - D]$$

A função de transferência e a resposta ao impulso do filtro são:

$$H(z) = 1 + a \cdot z^{-D}$$

$$h[n] = \delta[n] + a \cdot \delta[n - D]$$

Escolha o número de amostras, D , que o sinal deve ser atrasado, de tal forma que o sinal seja atrasado de 0,125s a especifique $a=0,5$.

Trace a resposta em frequência do filtro usando o comando *freqz* do Matlab.

Execute o experimento com valores de D cada vez menores e escute o sinal com eco até que não seja mais perceptível distinguir do sinal original. Calcule a quantos milissegundos de atraso este valor corresponde.

Filtro Pente (Comb Filter):

Implemente um filtro que adicione 3 ecos sucessivos, que possui a equação de diferenças no formato:

$$y[n] = x[n] + a \cdot x[n - D] + a^2 \cdot x[n - 2D] + a^3 \cdot x[n - 3D]$$

A função de transferência e a resposta ao impulso do filtro são:

$$H(z) = 1 + a \cdot z^{-D} + a^2 \cdot z^{-2D} + a^3 \cdot z^{-3D}$$

Implemente a forma não recursiva deste filtro escolhendo D tal que o atraso seja 0,2s. Utilize $a=0,5$ and trace a resposta em frequência deste filtro. Mantendo o mesmo atraso, D , escolha $a=0,2$ e rode o programa novamente. Que efeito você escuta ? Repita para $a=0,1$.

Filtro de Reverberação:

Se uma quantidade infinita de ecos for adicionada ao sinal, o efeito imita a reverberação natural de uma sala e fornece o filtro pente (comb filter) IIR ou reverberador. A equação de diferenças e a função de transferências são:

$$y[n] = x[n] + a \cdot y[n - D]$$

$$H(z) = \frac{1}{1 - a \cdot z^{-D}}$$

Novamente, escolha D tal que o atraso seja 0,125s, especifique $a=0,5$ e trace o gráfico da resposta em frequência do filtro reverberador. Execute o programa com o novo coeficiente $a=0,9$. Trace a resposta ao impulso. Repita para $a=0,25$. Discuta o efeito de aumentar ou diminuir a .

PARTE 2 – LOOP BACK DE AUDIO NO KIT DSP

Esta seção do laboratório e uma introdução à programação em tempo real no KIT DSP TMS320C5416 da Texas Instruments. O objetivo e tornar o aluno familiar com o hardware e o software usado no curso e introduzir os conceitos de processamento digital de sinais em tempo real. Cada sinal em tempo real deve ser recuperado, digitalizado (quantizado), processado pelo chip DSP C5416 e finalmente devera ser convertido para um sinal analógico.

Seção 2a

1 – Modelo para o projeto de áudio: Configura o CODEC, as 4 chaves para configurar o programa e 4 leds para funcionar como um gráfico de barras. Este template pode ser usado como um modelo para outros projetos.

2 – Atrasos e Ecos: Simula o atraso em redes de comunicação e reflexão do som escutada em um canyon. Introduz buffer circular e a configuração usada para filtros de resposta ao impulso finita (FIR)

3 – Reverberação: Simula uma e múltiplas reflexões das paredes de uma sala. Introduz a configuração usada em filtros de resposta ao impulso infinita (IIR)

4– Crossover eletrônico: Divide um sinal de áudio em bass e treble e usa para acionar os diferentes alto-falantes. Usa um filtro FIR e o usuário muda para selecionar 15 diferentes frequências de crossover entre 100Hz e 6,4kHz

Seção 2b – Exemplo de Loop Back de Audio

O DSK usa um codec stereo Burr Brown PCM3002 para entrada e saída de sinais de áudio. O codec amostra os sinais analógicos no microfone e *line in* e converte eles para dados digitais de tal forma que ele possa ser processado no DSP. Quando o DSP termina o processamento ele usa o codec para converter as amostras de volta para sinais analógicos no line out e alto-falantes para que o usuário possa ouvir a saída.

A biblioteca de periféricos on-board fornece suporte de alto nível para as aplicações do DSP rodando no C5416 DSK. Esta biblioteca habilita o usuário a desenvolver aplicações que podem controlar e operar os periféricos da plataforma DSK. O suporte dos periféricos on-board inclui:

- Inicialização on-board do periférico
- Acesso aos registradores
- Movimento de dados

Lab. 3 – TMS320C5416 DSK: Instalando Aplicações

O seguinte procedimento foi escrito para instalar a aplicação chamada *template*, que está armazenada no arquivo *template.zip*. Para outros projetos, modifique o nome *template* para aquela do arquivo *zip* que esta sendo instalado, por exemplo, *electronic_crossover*:

- 1- Inicie o software *CCS (Code Composer Studio)* para o TMS320C5416 DSK.
- 2- Selecione *Project -> New*. Para o nome do projeto digite a palavra *template*. Selecione o botão *Finish*. O novo projeto será criado. Este procedimento é mostrado na Figura 1. Faça uma anotação do diretório em que o projeto está sendo armazenado então minimize a janela do *CCS*. Os projetos devem ser criados em diretórios específicos, dentro do diretório *myprojects*.
- 3- Usando o *Windows Explorer* selecione o diretório onde o novo projeto foi criado, por
- 4- exemplo, *c:\ti\myprojects\template*.
- 5- Copie o arquivo *template.zip* para o diretório *c:\ti\myprojects\template*.
- 6- Descompacte o arquivo *template.zip* usando WinZip no diretório corrente, na opção *Extract Here*.
- 7- Retorne ao *CCS*. Selecione *Project -> Add Files to Project*. Selecione a pasta *c:\ti\myprojects\template*. Selecione todos os arquivos **.c* e aperte a tecla *Open*.
- 8- Selecione *Project -> Add Files to Project*. Selecione a pasta *c:\ti\myprojects\template*. Na caixa tipo de arquivo selecione *Configuration File (*.cdb)*. Selecione o arquivo *template.cdb* e tecla *Open* para adicionar ao projeto.
- 9- Selecione *Project -> Add Files to Project*. Selecione a pasta *c:\ti\myprojects\template*. Na caixa tipo de arquivo selecione *Linker Command File (*.cmd)*. Selecione o arquivo *template.cmd* e pressione a tecla *Open*.
- 10- Selecione *Project -> Add Files to Project*. Selecione a pasta *c:\ti\myprojects\template*. Na caixa tipo de arquivo selecione os arquivos *(*a, *.s)*. Selecione os arquivos *(.asm)*, se existirem, então pressione a tecla *Open*. Caso contrário pressione *Cancel*.
- 11- Apenas necessário quando estiver usando a biblioteca *dsplib*. Selecione *Project -> Add Files to Project*. Selecione a pasta *c:\ti\myprojects\template*. Na caixa tipo de arquivo selecione os arquivos *(*o, *.lib)*. Selecione os arquivos *(.lib)*, se existirem, então pressione a tecla *Open*. Caso contrário pressione *Cancel*.
- 12- Não há nenhuma necessidade de adicionar os arquivos *(*h)* ao projeto. Isto é feito automaticamente.
- 13- Selecione *Project -> Build Options*. Selecione a pasta *Compiler*. Selecione *Advanced* na janela *Category*. Habilite a opção *Calls (-mf) (548 and higher)* como indicado na Figura 2.
- 14- Ainda na mesma janela, selecione a subpasta *Linker*. Na linha *Include Libraries (-l)* digite *dsk5416f.lib*, a seguir selecione *OK*. Este procedimento é ilustrado na Figura 3.

- 15- Selecione *Project -> Rebuild All*. Irá aparecer uma mensagem de *Error* e um *Warning*. Os novos arquivos *templatecfg.s54*, *templatecfg_c.c* e *templatecfg.h*, devem ter sido criados no diretório do projeto.
- 16- Selecione *Project -> Add Files to Project*. Na caixa tipo de arquivo selecione os arquivos (*.cmd). Selecione o arquivo *templatecfg.cmd* e aperte *Open*. Haverá uma mensagem de erro indicando que um arquivo (*.cmd) já existe no projeto. Substitua o arquivo *template.cmd* pelo novo arquivo *templatecfg.cmd*, selecionando a opção *Yes*.
- 17- Selecione *Option -> Customize -> Program Load Options*. Habilite a opção *Load Program after Build*, conforme ilustrado na Figura 4.
- 18- Selecione *Project -> Build*. Desta vez o projeto deve ser compilado com sucesso e o arquivo *template.out* deve ser carregado automaticamente para o DSP.
- 19- Selecione *Debug -> Go Main* para mostrar o código principal. Selecione *Debug -> Run* para executar o projeto.

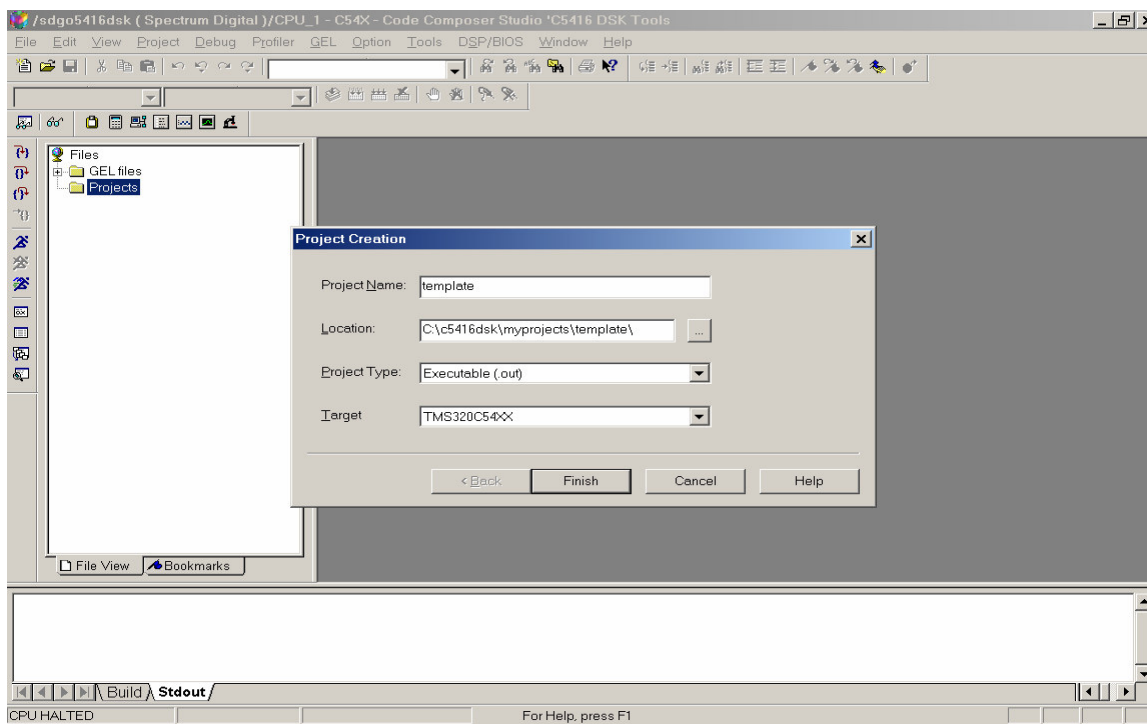


Figura 1 – Criando um novo projeto

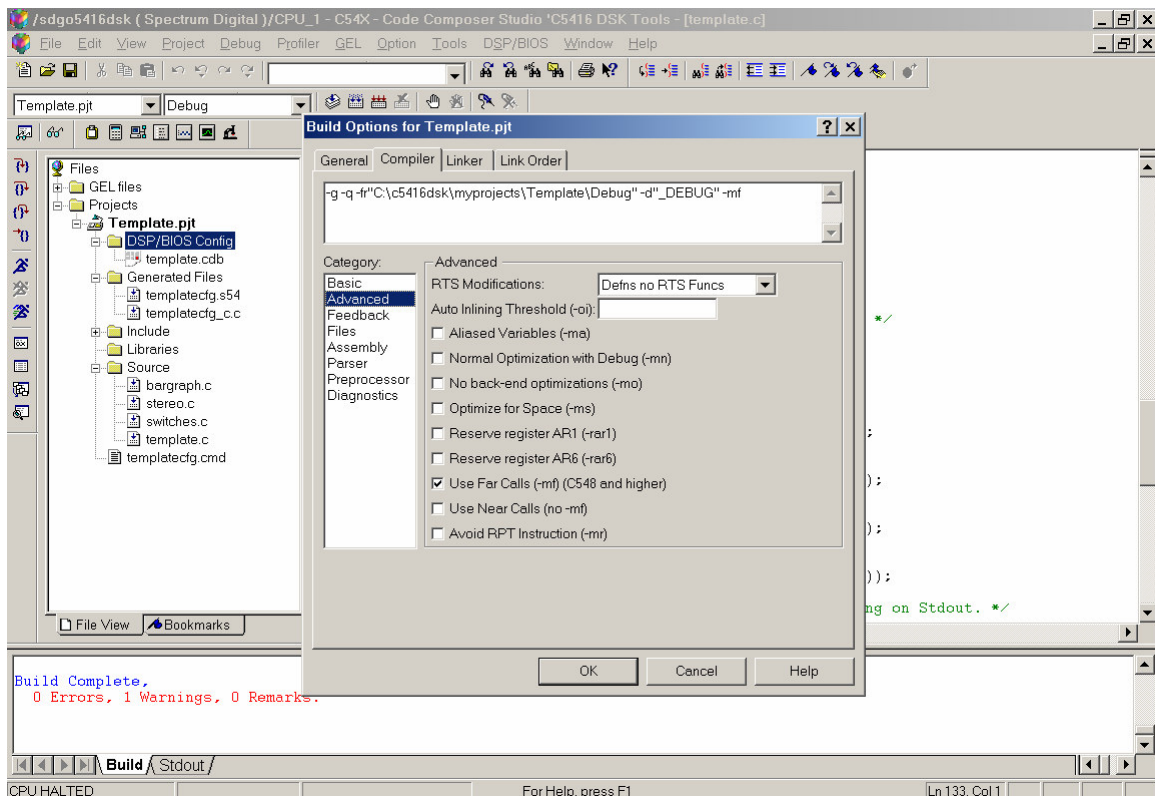


Figura 2 – Selecionando as opções de *Compiler* da etapa 11

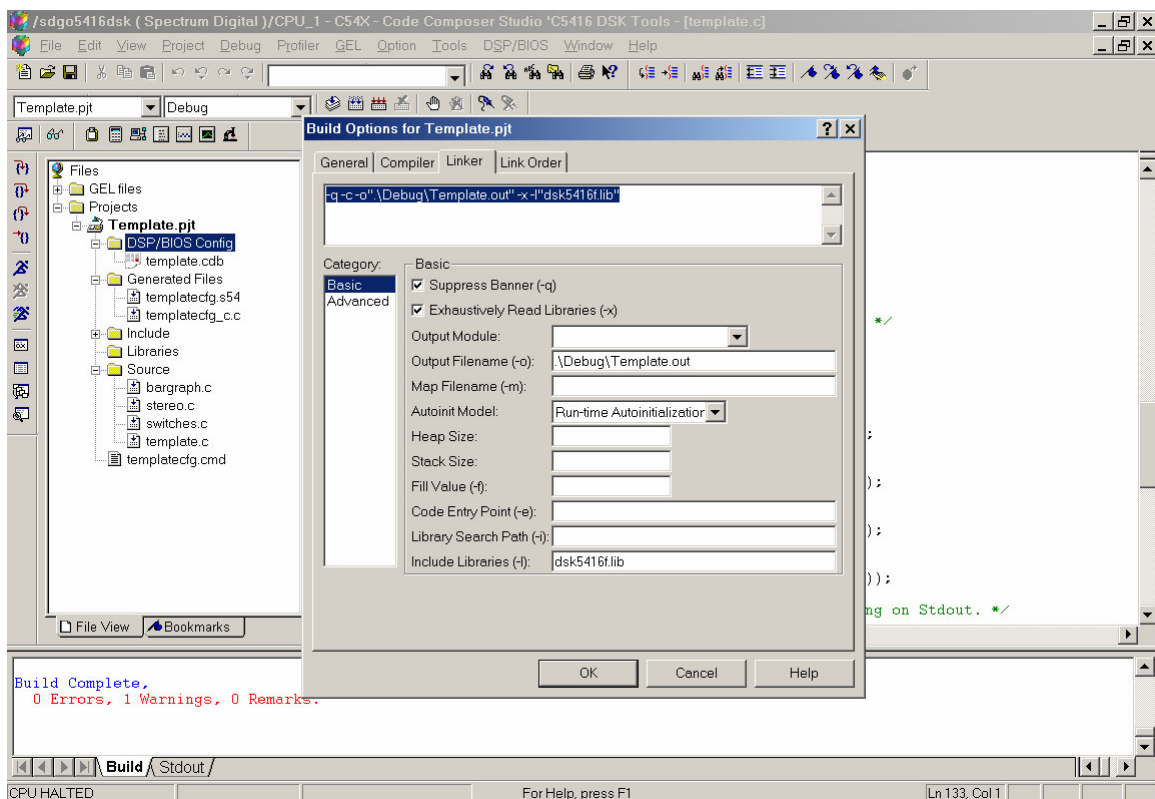


Figura 3 – Configurando as opções de *Linker* da etapa 11

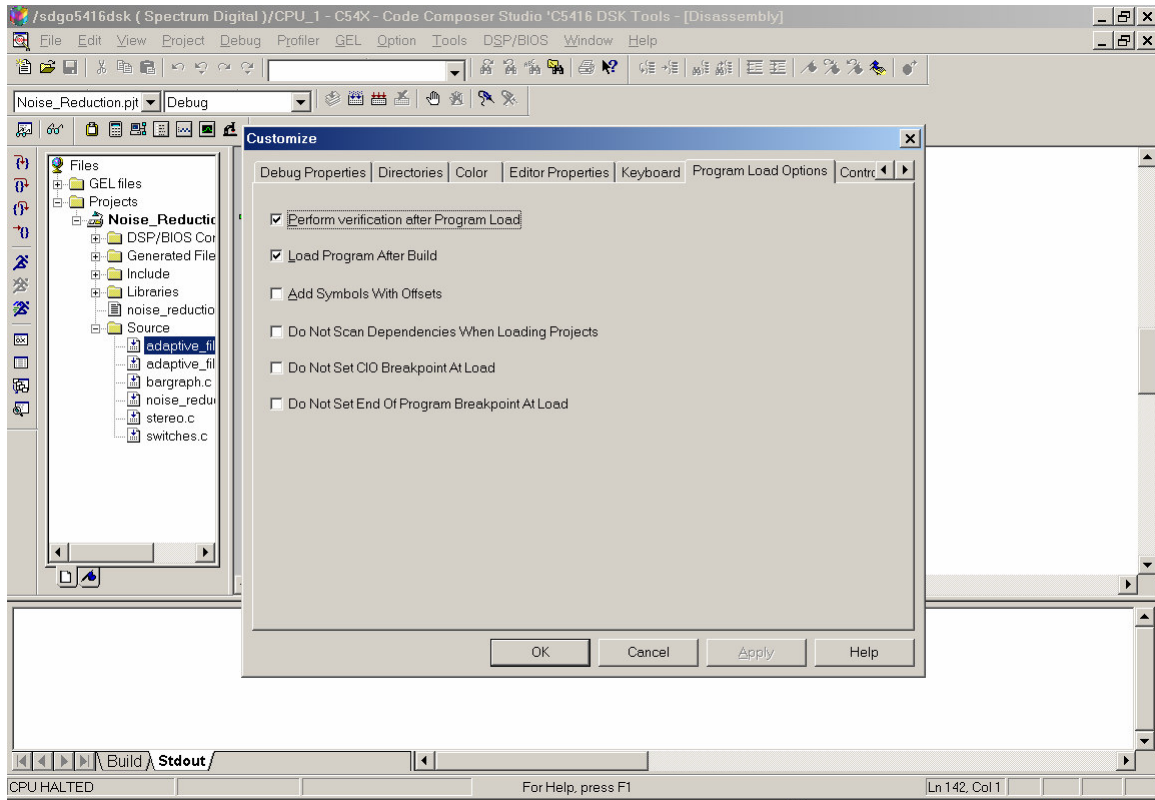


Figura 4 – Configurando a opção de *Load Program*

LAB. 4 DSP – Implementação de Filtros Digitais FIR

Neste laboratório deverá ser utilizado como base o projeto *electronic_crossover.zip*, a partir do qual serão implementados os filtros digitais.

- a) Utilizar o software MATLAB[®] para projetar os seguintes filtros digitais FIR:

Fs=24kHz							
Tipo	Frequência de Corte (Fc)	Comprimento (N)			Ordem (N-1)		
Filtro Passa Baixa (FPB)							
Filtro Passa Alta (FPA)							
Filtro Passa Faixa (FPF)							
Filtro Rejeita Faixa (FRF)							
Filtro Multibanda (Equalizador)							

- b) Implementar os filtros projetados no DSP e testar em tempo real utilizando como sinal de entrada uma forma de onda senoidal analógica, com frequência ajustada através do gerador de funções
- c) Ajustar no programa a frequência correta de amostragem (Fs) dos conversores A/D e D/A (codec)

LAB. 5 – Implementação de Filtros Digitais IIR

Neste laboratório deverá ser utilizado como base o projeto *electronic_crossover.zip*, a partir do qual serão implementados os filtros digitais.

Projeto Final da Disciplina de Processamento Digital de Sinais

Tópico	Descrição
Modulador AM-DSB	Implementar um modulador AM utilizando o DSP. Sintetizar via software a frequência de portadora de 6kHz e gerar o sinal modulado AM para um sinal analógico de informação (sinal modulante) obtido do conversor A/D. O sinal analógico não deve ter frequência superior a 2kHz. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Modulador AM DSB-SC	Implementar um modulador AM utilizando o DSP. Sintetizar via software a frequência de portadora de 12kHz e gerar o sinal modulado AM para um sinal analógico de informação (sinal modulante) obtido do conversor A/D. O sinal analógico não deve ter frequência superior a 2kHz. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Modulador 4-ASK	Sintetizar o sinal de portadora de 8kHz via software. Escolher uma seqüência de símbolos de informação a serem modulados (10 símbolos) e gerar o sinal modulado 4-ASK. A seqüência de informação escolhida deve ser modulada periodicamente, por questões de visualização no osciloscópio. Considerar uma taxa de transmissão de 200 baud. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Modulador 4-PSK	Sintetizar o sinal de portadora de 3kHz via software. Escolher uma seqüência de símbolos de informação a serem modulados (10 símbolos) e gerar o sinal modulado 4-PSK. A seqüência de informação escolhida deve ser modulada periodicamente, por questões de visualização no osciloscópio. Considerar uma taxa de transmissão de 500 baud. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Modulador 4-FSK	Sintetizar as frequências do modulador via software: 1,5kHz 1,7kHz 1,9kHz 2,1kHz Escolher uma seqüência de símbolos de informação a serem modulados (10 símbolos) e gerar o sinal modulado 4-FSK. A seqüência de informação escolhida deve ser modulada periodicamente, por questões de visualização no osciloscópio. Considerar uma taxa de transmissão de 300 baud. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Modulador 8-QAM	Sintetizar via software as frequências das portadoras que formam os sinais base (seno/coseno) de 5kHz e gerar o sinal modulado 8-QAM. A seqüência de informação escolhida deve ser modulada periodicamente, por questões de visualização no osciloscópio. Considerar uma taxa de transmissão de 1000 baud. O sinal modulado deve ser convertido para analógico (D/A) e visualizado no osciloscópio. (Fs=48kHz)
Analizador de Espectro	
Modulador AM SSB	
Costas Loop	
Squaring Loop	

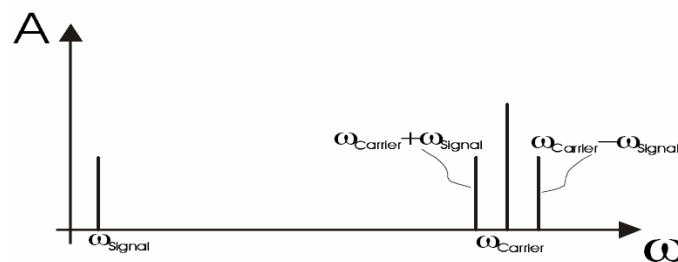
LAB. 6 – Modulação em Amplitude (AM/ASK) usando DSP

1. Introdução

Por muitos anos a modulação em amplitude foi o esquema de modulação padrão devido ao baixo custo e fácil implementação. Para a modulação e demodulação em amplitude são utilizados multiplicadores, que são implementados com semicondutores. Uma abordagem mais moderna da modulação em amplitude é o uso de técnicas de processamento digital de sinais. A geração da portadora e também a multiplicação pelo sinal de informação são feitos pelo software do DSP. As técnicas de modulação em amplitude (AM/ASK) são ainda muito utilizadas devido à economia em largura de banda.

2. Sistemas de Modulação em Amplitude

Na modulação AM/ASK significa simplesmente o deslocamento da frequência do sinal de informação para outra frequência (usualmente maior). O conteúdo espectral do sinal de informação (sinal modulante) é trasladado para outra frequência, em torno da portadora. A translação em frequência é feita pela multiplicação dos dois sinais no domínio do tempo. Multiplicação no domínio do tempo corresponde ao deslocamento no domínio da frequência, conforme ilustrado na Figura 1.



a) Justificativa Matemática para o Deslocamento em Frequência

O deslocamento em frequência pode ser provado aplicando-se a transformada de Fourier numa função $f(t)$ multiplicada por uma função coseno.

$$f(t) \cdot \cos(\omega_0 t) = \frac{1}{2} (f(t) \cdot e^{j\omega_0 t} + f(t) \cdot e^{-j\omega_0 t})$$

$$f(t) \cdot \cos(\omega_0 t) \Leftrightarrow \frac{1}{2} (F(\omega - \omega_0) + F(\omega + \omega_0))$$

Isso mostra que a multiplicação do sinal $f(t)$ pela frequência da portadora ω_0 desloca o espectro $F(\omega)$ por $\pm \omega_0$. Este tipo de modulação é chamado de modulação em amplitude.

b) Modulação em Amplitude Padrão

A modulação em amplitude balanceada resulta na perda do sinal de portadora, que transporta apenas informação redundante. Contudo a portadora é transmitida por diferentes razões na modulação em amplitude padrão. Portanto um *offset* é adicionado a portadora e ela é transmitida igualmente.

$$\begin{aligned}
 V_m &= A \cdot \text{sen}(w_c t) \cdot (1 + m \cdot \cos(w_{\text{mod}} t)) \\
 V_m &= A \cdot \left(-\frac{1}{2j} \right) (e^{jw_c t} - e^{-jw_c t}) \left(1 + m \frac{1}{2} (e^{jw_{\text{mod}} t} - e^{-jw_{\text{mod}} t}) \right) \\
 V_m &= -\frac{A}{2j} (e^{jw_c t} - e^{-jw_c t}) - \frac{Am}{4j} ((e^{jw_{\text{mod}} t} + e^{-jw_{\text{mod}} t}) (e^{jw_c t} - e^{-jw_c t})) \\
 V_m &= -\frac{A}{2j} (e^{jw_c t} - e^{-jw_c t}) - \frac{Am}{4j} (e^{j(w_{\text{mod}}+w_c)t} - e^{-j(w_{\text{mod}}+w_c)t} + e^{j(w_{\text{mod}}-w_c)t} - e^{-j(w_{\text{mod}}-w_c)t})
 \end{aligned}$$

onde m é chamado de índice de modulação (razão entre o pico do sinal modulante e o pico da portadora), e A é a amplitude do sinal de portadora. Sinais usando modulação em amplitude padrão podem ser demodulados por meio de diodos.

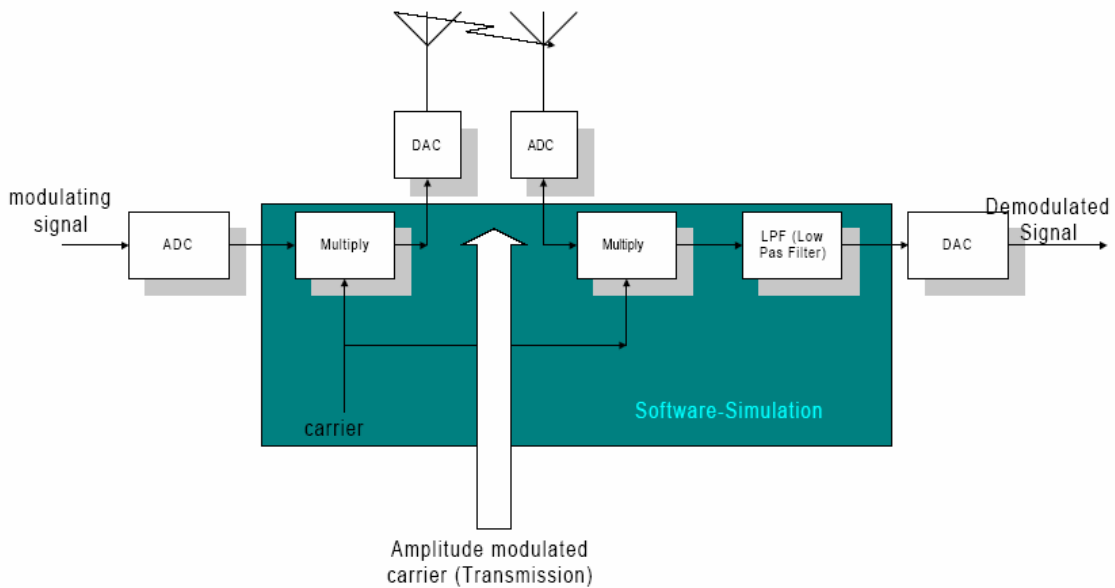
c) Modulação em Amplitude Balanceada

Nesta modulação o sinal de portadora é suprimido pois não contém informação útil e consome muita energia na transmissão.

$$\begin{aligned}
 V_m &= A \cdot \text{sen}(w_c t) \cdot m \cdot \cos(w_{\text{mod}} t) \\
 V_m &= A \cdot \left(-\frac{1}{2j} \right) (e^{jw_c t} - e^{-jw_c t}) \frac{m}{2} (e^{jw_{\text{mod}} t} + e^{-jw_{\text{mod}} t}) \\
 V_m &= -\frac{Am}{4j} (e^{j(w_c+w_{\text{mod}})t} - e^{-j(w_c+w_{\text{mod}})t} + e^{j(w_c-w_{\text{mod}})t} - e^{-j(w_c-w_{\text{mod}})t}) \\
 V_m &= -\frac{Am}{4j} \left(-\frac{1}{2j} \text{sen}(w_c + w_{\text{mod}}) - \frac{1}{2j} \text{sen}(w_c - w_{\text{mod}}) \right) \\
 V_m &= -\frac{Am}{2j} (-j \text{sen}(w_c + w_{\text{mod}}) - j \text{sen}(w_c - w_{\text{mod}})) \\
 V_m &= -\frac{Am}{2} (\text{sen}(w_c + w_{\text{mod}}) + \text{sen}(w_c - w_{\text{mod}}))
 \end{aligned}$$

4. Modulação em Amplitude usando DSP

Em sistemas de modulação em amplitude usando métodos de processamento digital de sinais a geração da portadora e a aritmética (multiplicação e geração de portadora) são feitos por meio de um processador digital de sinais. O sinal modulante é alimentado através do ADC para o sistema. A transmissão (canal) pode ser digital ou analógico. Para uma transmissão analógica a portadora digital modulada em amplitude deve ser convertida para analógica e de volta para digital no receptor. O objetivo do laboratório é projetar o sistema modulador e demodulador em amplitude em linguagem C. O sinal modulante foi implementado via uma função seno e o sinal de portadora foi implementado através de um filtro IIR de segunda ordem. O filtro passa baixa para remover as componentes indesejáveis foi implementado através de um filtro digital FIR.



a) Oscilador Digital

O oscilador para o sinal de portadora foi implementado por meio de um filtro digital IIR (recursivo) de segunda ordem. A equação diferencial de um oscilador analógico de segunda ordem é dada por:

$$\frac{d^2 y(t)}{dt^2} + w_0^2 \cdot y(t) = w_0^2 \cdot x(t)$$

$$\frac{[y(nT) - y(nT - T)]/T - [y(nT - T) - y(nT - 2T)]/T}{T} + w_0^2 \cdot y(nT) = w_0^2 \cdot x(nT)$$

$$\frac{y(n) - 2y(n-1) + y(n-2)}{T^2} + w_0^2 \cdot y(n) - w_0^2 \cdot x(n) = 0$$

$$y(n) - 2y(n-1) + y(n-2) + w_0^2 T^2 y(n) - w_0^2 T^2 x(n) = 0$$

$$y(n) - \frac{2y(n-1)}{1 + w_0^2 T^2} + \frac{y(n-2)}{1 + w_0^2 T^2} - \frac{w_0^2 T^2 x(n)}{1 + w_0^2 T^2} = 0$$

$$y(n) = \frac{1}{1 + w_0^2 T^2} x(n) + \frac{2}{1 + w_0^2 T^2} y(n-1) - \frac{1}{1 + w_0^2 T^2} y(n-2)$$

onde $x[n]$ é usualmente um impulso para iniciar o sistema. O problema da aproximação de diferenças recursiva é que ela muda a frequência do oscilador e a capacidade de oscilar sem perda do sinal devido a alteração dos coeficientes (Transforms in Signals and Systems, Peter Kraniiasukas, Addison-Wesley, pg. 321). Um bom cálculo dos coeficientes resulta na seguinte equação (C Algorithms for Real-Time DSP, Paul M. Embree, Prentice Hall, 1995, pg. 178)

$$y[n] = c_1 \cdot y[n-1] - c_2 \cdot y[n-2] + x[n]$$

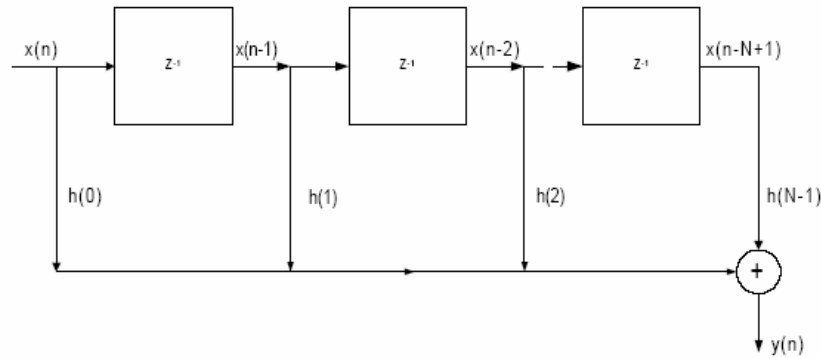
onde $c_1 = 2 \cdot e^{-d \cdot T} \cdot \cos(wT)$, $c_2 = e^{-2 \cdot d \cdot T}$, $T = \frac{1}{F_s}$, $w = 2\pi F_s$ e $d = 0 = \text{damping}$.

Para a capacidade de oscilação os pólos da transformada-Z devem estar sobre o círculo de raio unitário, o que não é dado pela aproximação de diferenças recursiva. A frequência da portadora do sinal pode ser facilmente verificada comparando com a frequência da modulação. A razão deve ser 1/10.

c) Filtro Passa Baixa

A função de transferência do filtro FIR, $H(z)$, é dada por:

$$H(z) = \sum_{n=0}^{N-1} h[n] \cdot z^{-n}$$



Os coeficientes do filtro foram calculados usando o método da Série de Fourier Truncada. As especificações são $f_{cutoff}=1\text{kHz}$, $f_{sample}=50\text{kHz}$, sem especificação da banda de transição e atenuação na banda de corte. Assim uma função de janela retangular foi usada.

$$\Delta\omega_c = 2\pi \frac{1000}{50000} = 0,04\pi$$

A resposta em frequência do filtro ideal é:

$$H(e^{j\omega}) = \begin{cases} e^{-j\alpha\omega}, & |\omega| \leq 0,04\pi \\ 0, & 0,04\pi < |\omega| \leq \pi \end{cases} \quad \text{onde } \alpha = \frac{N-1}{2}$$

Aplicando-se a IDTFT na resposta em frequência obtemos a resposta ao impulso:

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) \cdot e^{j\omega n} d\omega$$

$$h[n] = \frac{1}{2\pi} \int_0^{0,04\pi} e^{-j\alpha\omega} \cdot e^{j\omega n} d\omega + \frac{1}{2\pi} \int_0^{0,04\pi} 0 \cdot e^{j\omega n} d\omega$$

$$h[n] = \frac{1}{2\pi} \int_0^{0,04\pi} e^{-j\alpha\omega} \cdot e^{j\omega n} d\omega = \frac{1}{2\pi} \int_0^{0,04\pi} e^{j\omega(n-\alpha)} d\omega$$

$$h[n] = \frac{1}{2\pi} \left[\frac{e^{j\omega(n-\alpha)}}{j(n-\alpha)} \right]_0^{0,04\pi} = \frac{1}{2\pi j(n-\alpha)} (e^{j0,04\pi(n-\alpha)} - 1)$$

$$h[n] = \frac{1}{\pi(n-\alpha)} \text{sen}[0,04\pi(n-\alpha)]$$

A partir desta equação os coeficientes podem ser diretamente calculados pois nenhuma função de janela foi usada. Para calcular o coeficiente central do filtro aplicamos o teorema de L'Hospital para resolver a indeterminação (0/0):

$$h[n] = \frac{\sin[A\pi(n-\alpha)]}{\pi(n-\alpha)} = \frac{0}{0} \quad \text{quando } n = \alpha \text{ diferencie separadamente o numerador e o denominador}$$

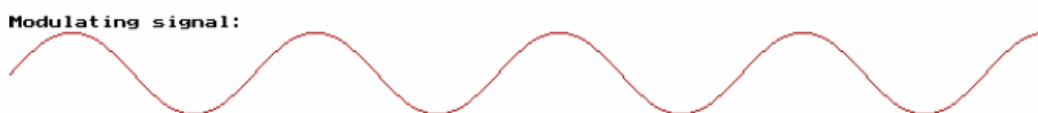
$$h[n] = \frac{A\pi[\cos(A\pi n - A\pi\alpha)]}{\pi}$$

$$h[n] = A = 0,04$$

5. Realização usando C

a) O sinal de modulação

Na modulação usual do sinal em amplitude, onde voz ou sinais equivalentes são modulados, o sinal modulante é alimentado para o DSP via um conversor A/D. No laboratório de simulação o sinal modulante foi gerado usando uma função seno com frequência 1kHz. A realização em C é dada por $y[n] = \sin(n \cdot w_{n1})$, onde w_{n1} é a frequência normalizada do sinal de 1kHz.



b) O Oscilador Digital

O oscilador foi projetado baseado na seguinte equação de diferenças derivada através da aproximação de diferenças recursivas. Apenas os coeficientes precisam ser determinados para preencher os requisitos de frequência de oscilação de 10kHz para uma frequência de amostragem de 50kHz.

$$y[n] = c_1 \cdot y[n-1] - c_2 \cdot y[n-2] + x[n]$$

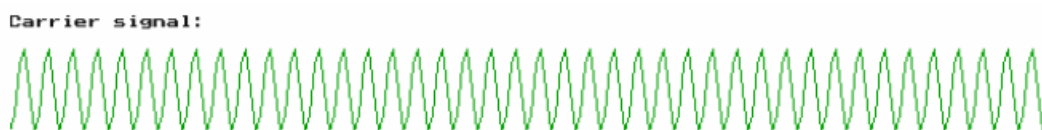
onde $c_1 = 2 \cdot e^{-d \cdot T} \cdot \cos(wT)$, $c_2 = e^{-2 \cdot d \cdot T}$, $T = \frac{1}{F_s}$, $w = 2\pi F_s$ e $d = 0 = \text{damping}$.

$$c_1 = 2 \cdot 1 \cdot \cos\left(\frac{2\pi 10\text{kHz}}{50\text{kHz}}\right) = 0,6180339$$

$$c_2 = 1$$

O código resultante é o seguinte:

```
x[0]=0.6180339*x[1]-x[2]+i;
x[2]=x[1]; // shifting oscillator array
x[1]=x[0];
```

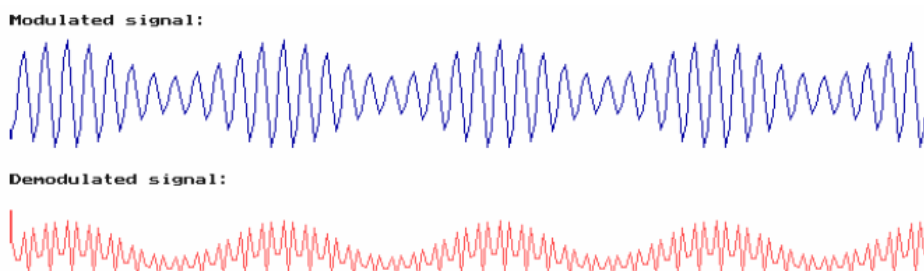


c) Modulação e Demodulação

A modulação e demodulação é baseada numa simples multiplicação:

```
// Modulating
mod=(ampmod+(2-ampmod*1.5)*signal)*carrier;
// Demodulating
demod=-mod*carrier;
```

A variável *ampmod* é usada para alterar entre modulação em amplitude normal e balanceada. O valor é especificado como 0 para modulação balanceada e 1 para modulação não balanceada. O fator 1 e 0.5 ($=2-1.5$) são escolhidos para obter-se resultados apropriados de modulação, porque a razão entre sinal modulado e portadora deve ser entre $0 < m < 1$, para obtermos resultados visíveis da modulação AM padrão. Os valores para modulação balanceada foram $A=2$ e $m=1$:



d) Filtragem Passa Baixa

O filtro passa baixa de ordem 51 é implementado usando convolução:

```
// Calculating Convolver Filter coefficients
for(n=0; n<=25 ; n++)
{
h[n]=sin((0.04)*pi*(n-25))/(pi*(n-25));
h[(50)-n]=h[n];
// Remove comments for printing filter coefficients
// printf("%d=>%f | %d=>%f \n",n,h[n],(50-n),h[50-n]);
}
h[25]=0.04; // Set mid filter coefficient (from L'Hospital)
```

Programa de filtragem que implementa a convolução:

```
// Filtering via convolver -----
filter_out=0;
for(k=0; k<=50; k++)
{
filter_out+=xstore[k]*h[k]; // convolving
xstore[k]=xstore[k+1]; // shifting array
}
xstore[50]=demod; /* writing new xstore(n) value */
// END OF CONVOLVER -----
```

a) Código fonte completo do software de modulação

```

#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define pi 3.14159
#define end 640
//
// Amplitude modulation - DSP simulation using C
// Dirk Becker, BENG 2/3, 9801351
//
// Balanced and Standard Amplitude Modulation
//
float x(int n);
void difference(int ypos, int scale, int ampmod);
int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode, scale=3;
    char key;
    int modtype=1;
    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLACK);
    do {
        /* start */
        difference(40, scale, modtype);
        setcolor(LIGHTBLUE);
        moveto(0, 447); lineto(639, 447);
        outtextxy(5, 450, "+- for scaling or Esc for exit, B for toggling
        modulating mode");
        key=getch();
        if (key=='+') scale++;
        if (key=='-') scale--;
        if ((key=='b') || (key=='B')) modtype=abs(modtype-1); // toggle
        if (scale<=0) {scale=1;};
        cleardevice();
        clrscr;
    } while (key!=27);
    /* closegraph (clean up) */
    closegraph();
    return 0;
}
/* ----- END OF MAIN FUNCTION ----- */
/* ----- */
/* ----- Oscillators, modulator, demodulator and filter ----- */
void difference(int ypos, int scale, int ampmod)
{
    int n, k, y_old, x_old, y_old2, xplot, yplot, yplot2, z_old, zplot,
    f_old=500, fplot; // Vars for line drawing
    // (storing old points)
    int m_old=290, mplot; // ( -"- )
    float xvalue, y[1]={0}, x[5]={0}, mod, demod, i=1;
    float fsignal1=1000, fsignal2=10000; // signal and oscillator
    float fsample=50000; // and sample frequencies
    float wn1=2*pi*fsignal1/fsample; // normalisation of them and
    float wn2=2*cos(2*pi*fsignal2/fsample); // coefficients of difference
    // equation
    float carrier, signal; // outputs vars
    float filter_out, xstore[52]={0}, h[52]={0}; // filter vars
    setcolor(RED);
    int scaley=25; // scaling in y direction
    float carriermax=0, signalmax=0, modmax=0, demodmax=0; // max-min
    calcs
    // Calculating Convolver Filter coefficients
    for(n=0; n<=25; n++)
    {
        h[n]=sin((0.04)*pi*(n-25))/(pi*(n-25));
        h[(50)-n]=h[n];
    }
}

```

```

// Remove comments for printing filter coefficients
// printf("%d=>%f | %d=>%f \n",n,h[n],(50-n),h[50-n]);
}
h[25]=0.04; // Set mid filter coefficient (from L'Hospital)
moveto(0,ypos);
i=1;
y_old=ypos;y_old2=ypos+70;z_old=ypos+200; // preset storings
// for line function
x_old=0;
for(n=0; n<=end; n++) // main loop
{
// Calculating frequency 1 (signal)
y[0]=sin(n*wn1);
signal=y[0]; // for normalisation to 1 (already is)
if (signal>=signalmax) signalmax=signal;
// Calculating frequency 2 (carrier)
x[0]=wn2*x[1]-x[2]+i;
carrier=x[0]; // for normalisation to 1 ( - " - )
if (carrier>=carriermax) carriermax=carrier;
// Modulating Ymod=A*f1*(1+m*f2) and change between
// Balanced and Standard modulation
// Standard Modulation: A=1(ampmod) , m=1.5
// Balanced Modulation: A=2 , m=1
//
mod=(ampmod+(2-ampmod*1.5)*signal)*carrier;
if (mod>=modmax) modmax=mod;
// Demodulating
demod=-mod*carrier;
if (demod<=demodmax) demodmax=demod;
// Filtering via convolver -----
filter_out=0;
for(k=0; k<=50; k++)
{
filter_out+=xstore[k]*h[k]; // convolving
xstore[k]=xstore[k+1]; // shifting array
}
xstore[50]=demod; /* writing new xstore(n) value */
// END OF CONVOLVER -----
x[2]=x[1]; // shifting oscillator array
x[1]=x[0];
// Drawing the waveforms
// the old points are always stored in the _old vars
setcolor(RED); // Raw signal
xplot=n*scale;
yplot=-signal*scale+ypos;
line(x_old, y_old, xplot, yplot);
y_old=yplot;
setcolor(GREEN); // Carrier frequency
yplot2=carrier*scale+ypos+70;
line(x_old, y_old2, xplot, yplot2);
y_old2=yplot2;
setcolor(BLUE); // Modulated signal
zplot=mod*scale+ypos+170;
line(x_old, z_old, xplot, zplot);
z_old=zplot;
setcolor(LIGHTRED); // Demodulated (multiplied)
mplot=demod*scale+ypos+295;
line(x_old, m_old, xplot, mplot);
m_old=mplot;
setcolor(LIGHTGREEN); // Filtered via convolver
if ((filter_out>400)|(filter_out<-400)) filter_out=0;
fplot=filter_out*(50+100*ampmod)+ypos+(370+60*ampmod);
line(x_old, f_old, xplot, fplot);
f_old=fplot;
x_old=xplot;
i=0; /* End of impulse */
}
// Adding text to curves
//
setcolor(WHITE);
outtextxy(0,5,"Modulating signal:");

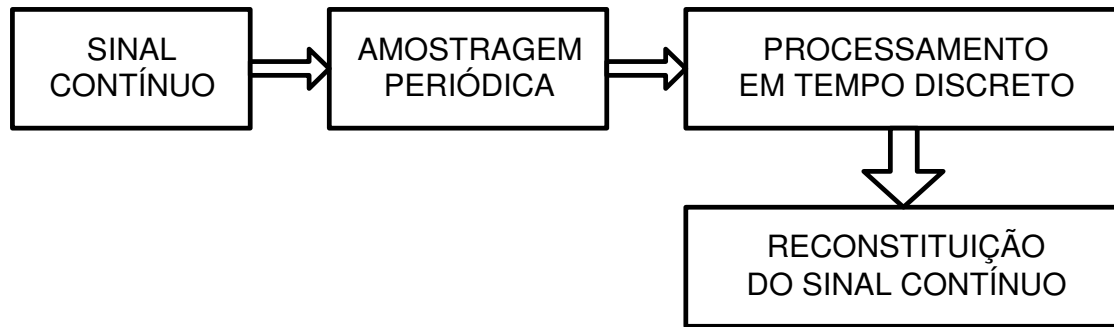
```

```
outtextxy(0,65,"Carrier signal:");
outtextxy(0,155,"Modulated signal:");
outtextxy(0,265,"Demodulated signal:");
outtextxy(0,340,"Demodulated and filtered signal:");
// Printing max values
gotoxy(60,1);
printf("sigmax: %1.2f \n",signalmax);
gotoxy(60,5);
printf("carmax: %1.2f \n",carriermax);
gotoxy(60,11);
printf("modmax: %1.2f \n",modmax);
gotoxy(60,17);
printf("denmax: %1.2f \n",demodmax);
}
```

Anexo A

Teorema de Nyquist

1 – AMOSTRAGEM PERIÓDICA



Considere um sinal contínuo $x_c(t)$. O sinal discreto no tempo pode ser obtido através da amostragem periódica do sinal $x_c(t)$,

$$x[n] = x_c(nT), \quad n = 0, \pm 1, \pm 2, \dots$$

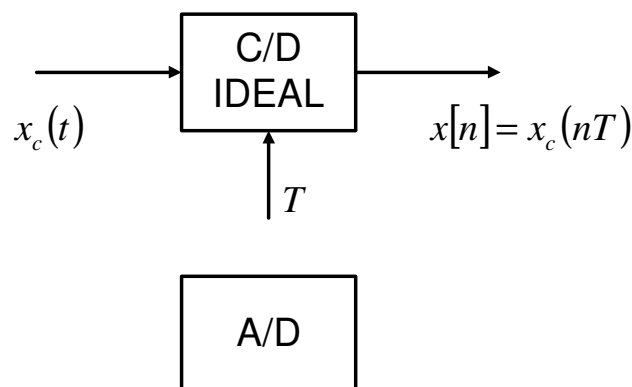
$$\dots, x[-2], x[-1], x[0], x[1], x[2], \dots$$

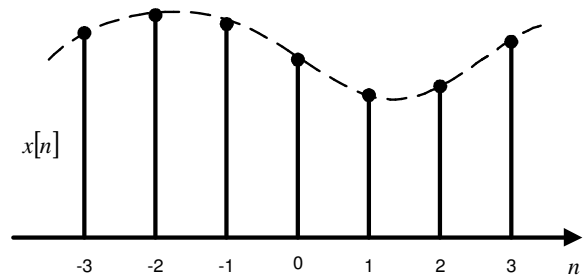
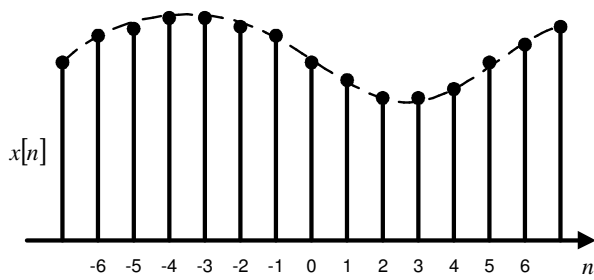
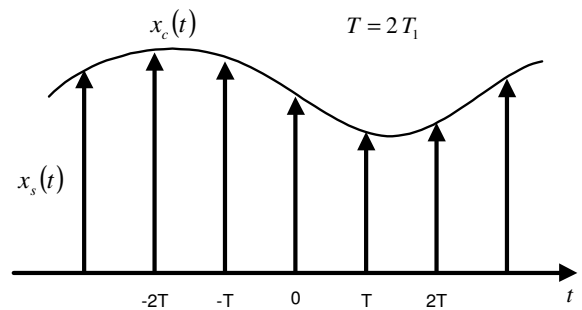
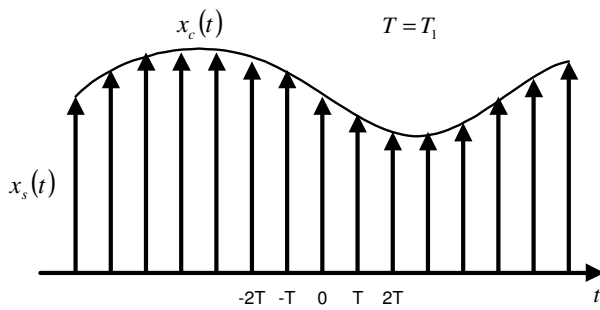
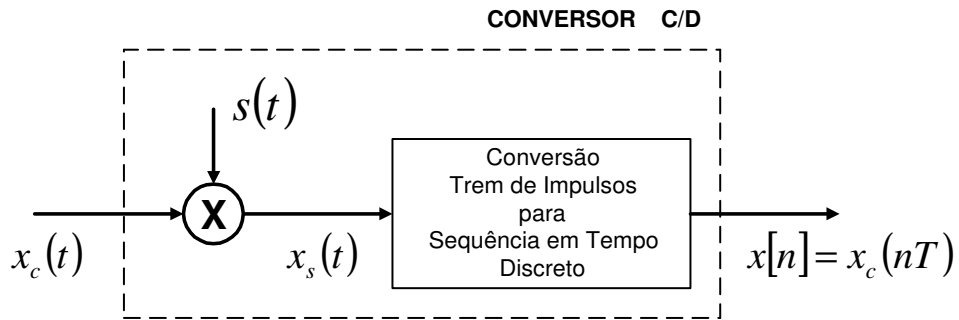
onde

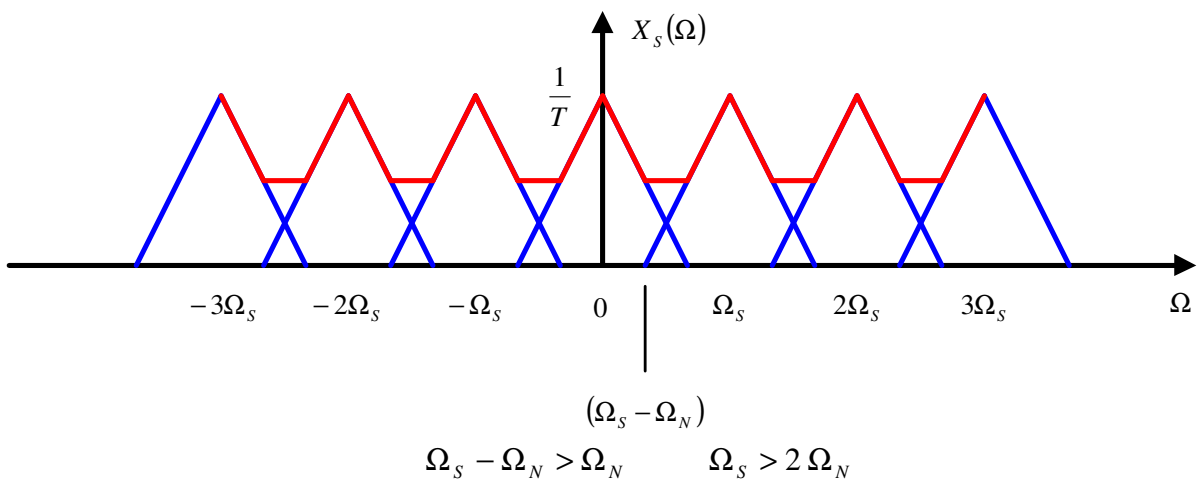
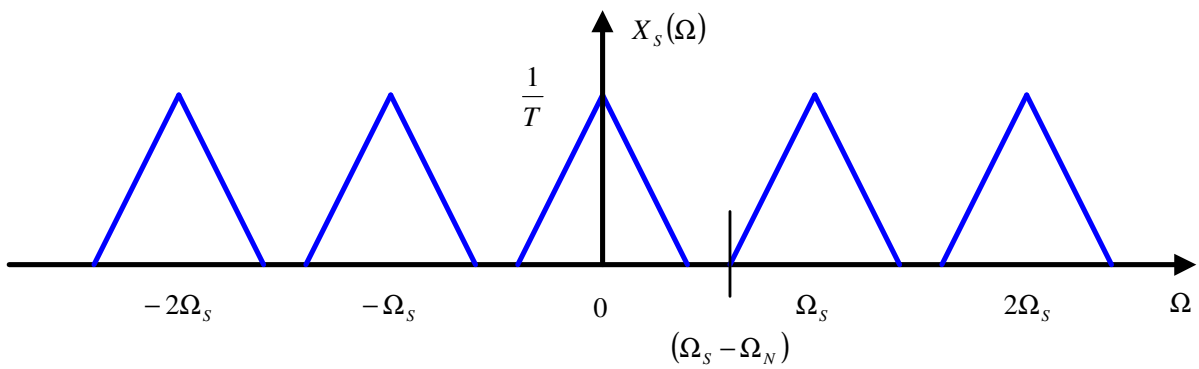
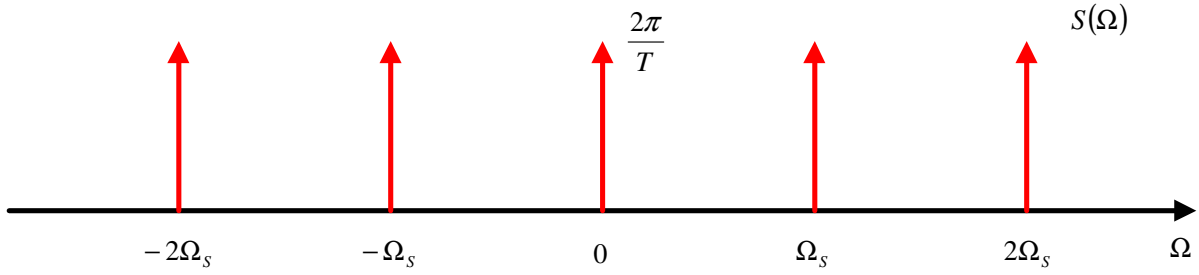
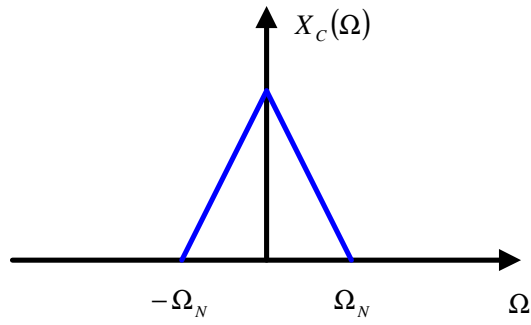
$T_s =$ Período de amostragem

$f_s = 1/T_s =$ Frequência de amostragem

$\Omega_s = 2\pi/T_s =$ Frequência angular (radianos/s)







2 – REPRESENTAÇÃO DA AMOSTRAGEM NO DOMÍNIO DA FREQUÊNCIA

$$s(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$

$\delta(t)$ = função delta de Dirac

$$x_s(t) = x_c(t) s(t)$$

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x_c(nT) \delta(t - nT)$$

$$X(\Omega) = \int_{-\infty}^{+\infty} x(t) \exp(-j\Omega t) dt$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\Omega) \exp(j\Omega t) d\Omega$$

$$x_1(t) * x_2(t) \Rightarrow X_1(\Omega) X_2(\Omega)$$

$$x_1(t) x_2(t) \Rightarrow \frac{1}{2\pi} X_1(\Omega) * X_2(\Omega)$$

$$S(\Omega) = \frac{2\pi}{T} \sum_{n=-\infty}^{+\infty} \delta(\Omega - n\Omega_s)$$

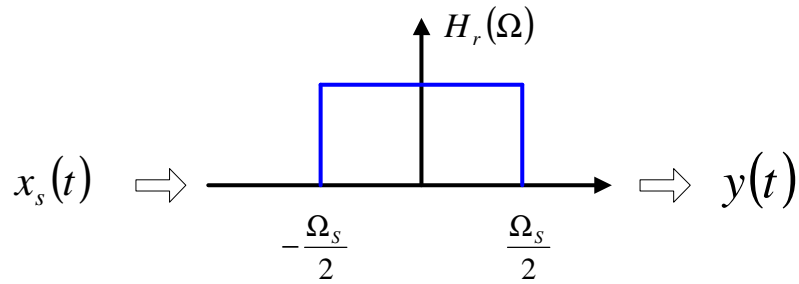
$$X_s(\Omega) = \frac{1}{2\pi} X_c(\Omega) * S(\Omega)$$

$$X_s(\Omega) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} X_c(\Omega - n\Omega_s)$$

3 – RECONSTRUÇÃO DO SINAL DE BANDA LIMITADA A PARTIR DE SUAS AMOSTRAS

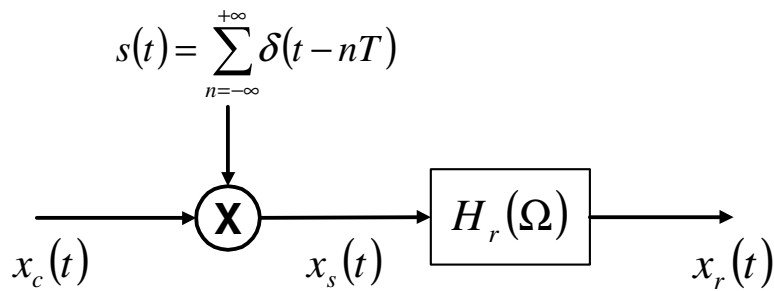
$$x_s(t) = \sum_{n=-\infty}^{+\infty} x[n] \delta(t - nT)$$

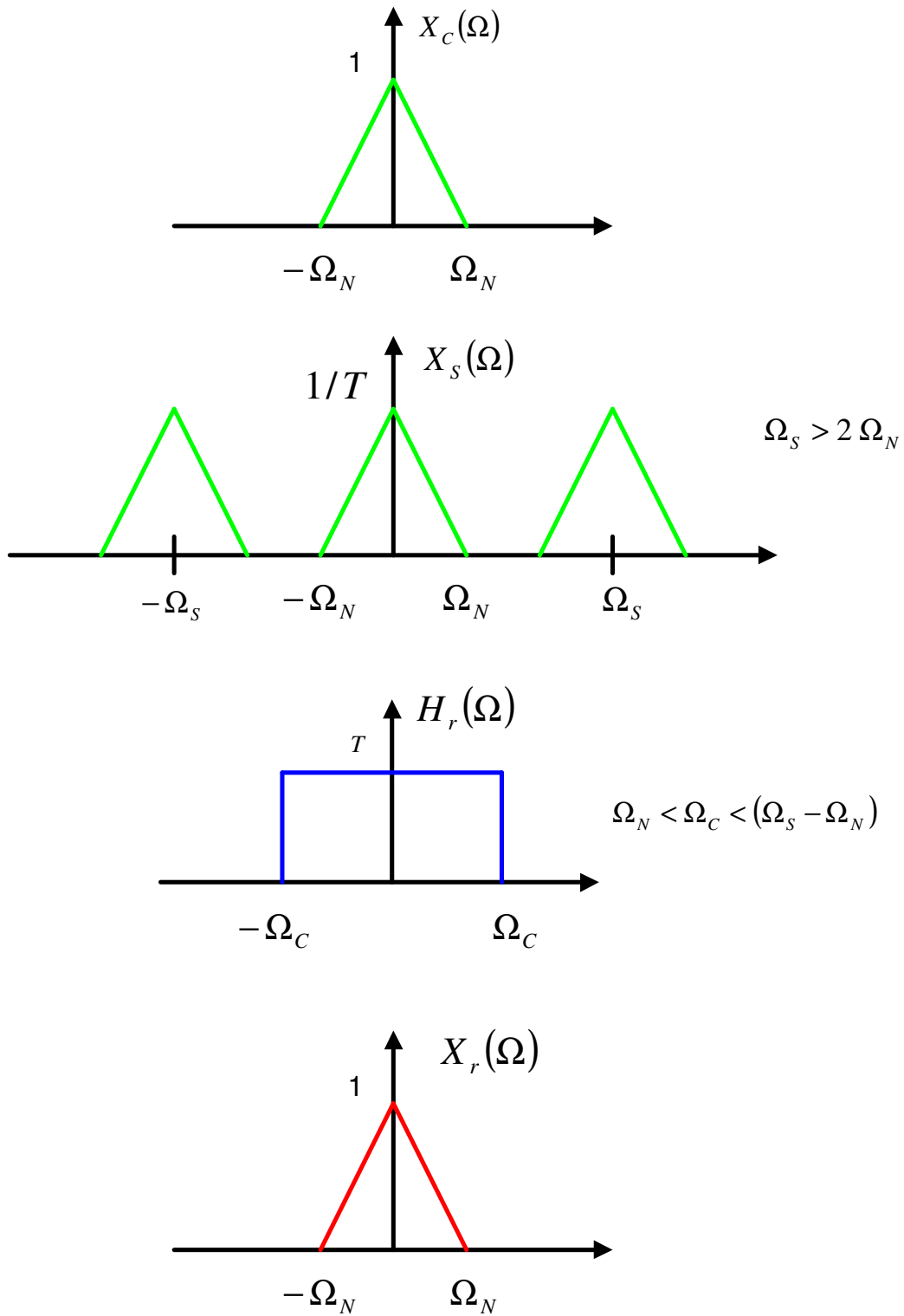
$$y(t) = x_s(t) * h_r(t) = x_r(t)$$



$$h_r(t) = \frac{\text{sen}(\pi t / T)}{\pi t / T}$$

$$x_r(t) = \sum_{n=-\infty}^{+\infty} x[n] \frac{\text{sen}[\pi(t - nT) / T]}{\pi(t - nT) / T}$$





4 – TEOREMA DE NYQUIST

$$x_c(\Omega) = 0 \quad \text{para } |\Omega| \geq \Omega_N$$

$x_c(t)$ é um sinal de faixa limitada

$$x[n] = x_c(nT) \quad n = 0, \pm 1, \pm 2, \dots$$

$$\Omega_s = \frac{2\pi}{T} \geq 2 \Omega_N$$

Ω_N = Frequência de Nyquist

$2 \Omega_N$ = Taxa de Nyquist