Apostila de Trabalhos Extra Classe de Exercício e de Código (TEC) para Estudo Orientado de Octave

(Versão A2020M08D12)

Universidade Federal Fluminense

Alexandre Santos de la Vega

Departamento de Engenharia de Telecomunicações – TET

Escola de Engenharia – TCE

Universidade Federal Fluminense – UFF

Agosto – 2020

.

621.3192 (*) D278 (*) 2020	de la Vega, Alexandre Santos Apostila com Trabalhos Extra Classe de Exercício e de Código (TEC) para Estudo Orientado de Octave / Alexandre Santos de la Vega. – Niterói: UFF/TCE/TET, 2020. 61 (sem romanos) ou 81 (com romanos) p. (*) Apostila com Trabalhos Extra Classe de Exercício e de Código (TEC) – Graduação, Engenharia de Telecomunicações, UFF/TCE/TET, 2020. 1. Processamento de Sinais. 2. Processamento Digital de Sinais. 3. Telecomunicações. I. Título.

(*) OBTER INFO NA BIBLIOTECA, ATUALIZAR E PEDIR NOVO REGISTRO !!!

Aos meus alunos.

Prefácio

O trabalho em questão aborda os tópicos a serem apresentados na disciplina Fundamentos de Processamento Digital de Sinais.

O material completo encontra-se dividido em quatro volumes. O conteúdo teórico pode ser encontrado no volume entitulado Apostila de Teoria para Fundamentos de Processamento Digital de Sinais. O conteúdo prático pode ser encontrado no volume entitulado Apostila de Códigos de Programas Demonstrativos para Fundamentos de Processamento Digital de Sinais. As especificações dos trabalhos extra classe propostos na disciplina podem ser encontradas no volume entitulado Apostila de Trabalhos Extra Classe de Exercício e de Código (TEC) para Fundamentos de Processamento Digital de Sinais. Uma abordagem integradora dos tópicos de interesse da disciplina, de forma simples e direta, utilizando o sistema de média móvel como exemplo, pode ser encontrado no volume entitulado Tutorial sobre Sistema de Média Móvel para Fundamentos de Processamento Digital de Sinais.

Os documentos foram escritos com o intuito de servir como uma referência rápida para os alunos dos cursos de graduação e de mestrado em Engenharia de Telecomunicações da Universidade Federal Fluminense (UFF).

O material básico utilizado para o conteúdo teórico foram as minhas notas de aula, que, por sua vez, originaram-se em uma coletânea de livros sobre os assuntos abordados.

Os códigos de programas demonstrativos e as especificações dos trabalhos propostos são completamente autorais.

A motivação inicial para o desenvolvimento desse trabalho foi a de aumentar o dinamismo das aulas. Logo, deve ficar bem claro que os documentos produzidos não pretendem substituir os livros textos ou outros livros de referência. Pelo contrário, espera-se que eles sejam utilizados como ponto de partida para estudos mais aprofundados, utilizando-se a literatura existente.

Espero conseguir manter o presente texto em constante atualização e ampliação.

Correções e sugestões são sempre bem-vindas.

Alexandre Santos de la Vega
UFF / TCE / TET

Agradecimentos

Aos professores do Departamento de Engenharia de Telecomunicações (TET), da Escola de Engenharia (TCE), da Universidade Federal Fluminense (UFF), que colaboraram com críticas e sugestões bastante úteis à finalização da versão inicial deste trabalho.

Aos funcionários e ex-funcionários do TET, Arlei, Carmen Lúcia, Eduardo Wallace, Francisco e Jussara, pelo apoio constante.

Aos meus alunos, que, além de servirem de motivação principal, obrigam-me sempre a me tentar melhorar, em todos os sentidos.

Mais uma vez, e sempre, aos meus pais, por tudo.

Alexandre Santos de la Vega
UFF / TCE / TET

Apresentação do material didático

- O material aqui apresentado não é fruto de um projeto educacional envolvendo idealização, planejamento, pesquisa, estruturação, desenvolvimento, revisão e edição.

- Pelo contrário, ele nasceu, evoluiu e tem sido mantido de uma forma bem orgânica.

- Em 1995, o autor ingressou no Departamento de Engenharia de Telecomunicações (TET) da Universidade Federal Fluminense (UFF) e, desde então, tem sido responsável por diversas disciplinas oferecidas pelo TET para o Curso de Engenharia de Telecomunicações, da Escola de Engenharia da UFF (TCE/UFF), e para o Curso de Ciência da Computação, do Instituto de Computação da UFF (IC/UFF).

- Na época do seu ingresso, o Processamento Digital de Sinais já era um assunto presente na área de Telecomunicações. E com importância crescente. Apesar disso, ainda não era oferecida pelo TET uma disciplina formal sobre a matemática que o fundamenta.

- Com essa percepção, ele criou a disciplina optativa "Introdução ao Processamento Digital de Sinais", em 1998.

- Para dar suporte às aulas, foram elaboradas as primeiras notas de aula (manuscritas) para a disciplina optativa criada no TET. Nessa primeira tentativa de implantação da disciplina, foi usada a referência [Mit98] como livro texto.

- A disciplina optativa foi oferecida pelo autor apenas durante dois períodos letivos, em virtude do seu afastamento para finalização do seu doutoramento.

- Durante o afastamento, e mesmo algum tempo depois, a disciplina optativa foi oferecida por outro professor do TET. Nesse período, o autor lançou uma outra disciplina optativa, vinculada à primeira, tratando do Projeto de Filtros Digitais.

- Na primeira década de 2000, o TET realizou uma reforma curricular e a disciplina optativa "Introdução ao Processamento Digital de Sinais" tornou-se obrigatória, sob o nome de "Processamento Digital de Sinais".

- Tendo voltado a ministrar a disciplina, o autor decidiu ampliar as notas de aula manuscritas, baseando-se em diversos outros livros.

- Em 2008, com os objetivos iniciais de melhor organizar os manuscritos e de atender aos apelos dos alunos por cópia dos manuscritos, eles foram apenas transcritos para o Sistema de Preparação de Documentos LaTeX [KD04] , [MG04]. Assim, surgiu a primeira versão da apostila de teoria.

- A partir daí, com a maturação gradual que a disciplina foi ganhando a cada período letivo, novos conteúdos foram surgindo. Ora por curiosidade do autor, procurando incorporar um determinado tópico na disciplina. Ora por curiosidade dos alunos, por demandarem algum assunto em especial. Ora por necessidade pedagógica, pois, ao se perceberem dúvidas recorrentes dos alunos, novas formas de abordagem têm sido testadas.

- Além disso, como filosofia educacional do autor, as questões que fazem parte de toda e qualquer forma de avaliação formal da disciplina (testes, provas, trabalhos) são anexadas ao conteúdo, na forma de exercícios propostos.

- Também como filosofia educacional do autor, a apostila de teoria não apresenta figuras que ilustrem os assuntos abordados. Pelo contrário, é demandado aos alunos que eles gerem as suas próprias figuras, a partir de um aplicativo computacional adequado.

- Para incentivar os alunos a modificarem códigos existentes e a gerarem seus próprios códigos, a partir de 2011, uma nova apostila tem sido elaborada, contendo códigos de programas demonstrativos, relativos aos tópicos abordados na apostila de teoria, em sala de aula e/ou em alguma forma de avaliação formal da disciplina.

- A partir de 2016, com a incorporação de trabalhos semanais na prática da disciplina, uma nova apostila tem sido elaborada, contendo os trabalhos propostos a cada período letivo.

- No final da década de 2010, o TET realizou uma nova reforma curricular, a qual acarretou uma redução na quantidade e na carga horária das disciplinas. Isso provocou uma reformulação na abordagem dos tópicos da disciplina, que passou a ser denominada de "Fundamentos de Processamento Digital de Sinais".

- Em 2018, foi percebido que, utilizando o sistema de média móvel como exemplo, é possível abordar e integrar os tópicos de interesse da disciplina de forma simples e direta. Além disso, com ele, também é possível gerar exemplos, exercícios e aplicações práticas, com certa facilidade. Assim, teve início a elaboração do tutorial sobre o sistema de média móvel.

- Dessa forma, desde o início da sua confecção até o presente momento, sempre foram preparadas diversas versões de cada documento ao longo de um mesmo período letivo. Por essa razão, o identificador "Versão A<ano>M<mês>D<dia>" aparece logo abaixo do título de cada apostila.

- No tocante à apresentação do conteúdo teórico, os manuscritos originais continham apenas tópicos, destinados à abordagem do conteúdo programático durante as aulas. Pode-se dizer que tais manuscritos representavam apenas um roteiro de aula. Gradativamente, com a evolução da apostila de teoria, os tópicos têm sido trocados por textos dissertativos, relativos ao conteúdo abordado.

- No ponto de vista estrutural é que o aspecto dinâmico dos documentos mais se tem feito presente. Os mais diversos seccionamentos de texto (capítulos, seções, subseções, etc.) surgem, são mesclados e desaparecem, a cada nova versão.

- Por tudo isso, pode-se asseguradamente dizer que todo o material produzido encontra-se em constante atualização.

- Na preparação das aulas, têm sido utilizados os seguintes livros:
  - Livros indicados pela ementa da disciplina: [DdSN10], [Mit98].
  - Outros livros indicados: [Rob09], [PM06], [Jac96], [She95], [SK89], [Ant86], [SDD84], [OWY83], [PL76], [OS75], [Cad73].

Teoria abordada no material didático

- Introdução <2 horas>
  - Conceitos básicos: que busca contextualizar a disciplina no âmbito do curso e apresentar conceitos que serão necessários ao longo do texto. <2 horas>
  - Amostragem e interpolação: que apresenta um resumo das representações dos sinais analógicos no domínio da freqüência e aborda as duas formas de conexão entre os domínios analógico e digital. [Opcional]
- Sinais e sistemas (com tempo discreto) no domínio do tempo <10 horas>
  - Sinais no domínio do tempo: definições, classificações, operações, exemplos e caracterizações. <4 horas>
  - Seqüências exponenciais: características relevantes de exponenciais, funções com dependência exponencial, decomposição de funções usando exponenciais, amostragem de sinais contínuos no tempo. <2 horas>
  - Sistemas no domínio do tempo: definições, classificações, operações, exemplos e caracterizações. <4 horas>
- Representações de um Sistema Linear e Invariante ao Tempo (SLIT) <12 horas>
  - Resposta ao impulso.
  - Equação de diferença.
  - Diagramas de blocos de complexidade genérica.
  - Diagramas de sistema (ou estruturas ou realizações).
  - Operador de transferência.
  - Diagrama de pólos e zeros do operador de transferência.
  - Equações de estado.
  - Relações e mapeamentos entre as diversas representações.
- Respostas de um Sistema Linear e Invariante ao Tempo (SLIT) <10 horas>
  - Cálculos da resposta de um SLIT <8 horas>
    - Cálculo da resposta de um SLIT baseado na solução das equações de estado.
    - Cálculo da resposta de um SLIT baseado no uso do operador de transferência.
    - Cálculo da resposta de um SLIT baseado na solução convencional da equação de diferença.
    - Cálculo da resposta de um SLIT FIR (Resposta ao Impulso Finita) com entrada de comprimento indefinido.

  - Tipos de resposta de um SLIT <2 horas>
    * Resposta completa.
    * Resposta homogênea + resposta do sistema relaxado (resposta particular + resposta complementar).
    * Resposta ao estado + resposta à entrada.
    * Resposta natural + resposta forçada.
    * Resposta transitória + resposta permanente.

- Noções da representação em domínio transformado para sistemas de primeira ordem [Opcional]
  - Resposta em Freqüência: baseado no cálculo da resposta de um SLIT de primeira ordem, para um determinado tipo de sinal de entrada, pode-se identificar um novo tipo de representação para o sistema.
  - Função de Transferência: baseado no cálculo da resposta de um SLIT de primeira ordem, para um determinado tipo de sinal de entrada, pode-se identificar um novo tipo de representação para o sistema.

- Sinais e sistemas (com tempo discreto) no domínio da freqüência <20 horas>
  - Sinais <12 horas>
    * Motivações para a mudança de domínio de uma representação.
    * Revisão das representações em freqüência com tempo contínuo (Série de Fourier, Transformada de Fourier e Transformada de Laplace).
    * Série de Fourier de Tempo Discreto (DTFS).
    * Transformada de Fourier de Tempo Discreto (DTFT).
    * Transformada de Fourier Discreta (DFT).
    * Transformada Z.
    * Relações entre as diversas representações em freqüência, parâmetros e efeitos importantes.
  - Técnicas básicas para aceleração do cálculo da DFT. [Opcional]
  - SLIT de ordem qualquer <8 horas>
    * Tipos de respostas de um sistema.
    * Resposta completa em domínio transformado.
    * Resposta em Freqüência.
    * Seletividade em Freqüência.
    * Função de Transferência ou Função de Sistema.
    * Representações de um SLIT no domínio da freqüência.

- Aplicações: exemplos de aplicações são distribuídos ao longo do texto e exercitados na forma de trabalhos.

Objetivos da disciplina

- Apresentar a base matemática que fundamenta o Processamento Digital de Sinais.
- Trabalhar com sistemas que apresentem as seguintes características:
  - Sistema Linear e Invariante ao Tempo (SLIT).
  - Sistema *Single-Input Single-Output* (SISO).
  - Sistema operando com tempo discreto.
  - Sistema operando com sinais definidos em tempo discreto, quantizados (digitais) ou não (amostrados).
- Trabalhar com sinais básicos que sejam simultaneamente dependentes das variáveis tempo e freqüência, utilizando-os na composição dos demais sinais envolvidos.
- Discutir a análise de sistemas no domínio da variável tempo e no domínio da variável freqüência. No domínio do tempo, o foco está na FORMA que os sinais apresentam. No domínio da freqüência, o foco está na COMPOSIÇÃO que os sinais apresentam.
- Discutir a aplicação dos conceitos de Operador de Transferência (no domínio do tempo) e de Função de Transferência (no domínio da freqüência), bem como a relação existente entre ambos.
- Discutir a aplicação do conceito de estado de um sistema e da análise do sistema no espaço de estados.

Sumário

Prefácio **v**

Agradecimentos **vii**

Apresentação do material didático **ix**

Teoria abordada no material didático **xiii**

Objetivos da disciplina **xv**

Sumário **xvii**

I Introdução 1

1 O aplicativo computacional Octave **3**

1.1 Geral . . . 3
1.2 Interpretador de comandos . . . 3
1.3 Tipos de dados . . . 4
1.4 Estruturas de dados . . . 4
1.5 Funções dedicadas a matrizes . . . 5
1.6 *Download*, instalação e *help* . . . 5
1.6.1 *Download* e instalação . . . 5
1.6.2 *Help* . . . 5

2 Regras básicas **7**

II Trabalhos TEC básicos 9

3 Definição dos trabalhos TEC básicos manuais **11**

3.1 TEC-BM1: revisão de matrizes . . . 11
3.1.1 Definições . . . 11
3.1.2 Especificações . . . 11
3.1.3 Tarefas . . . 12
3.2 TEC-BM2: revisão de números complexos . . . 15
3.2.1 Definições . . . 15
3.2.2 Especificações . . . 15
3.2.3 Tarefas . . . 16
3.3 TEC-BM3: operações básicas sobre seqüências . . . 19
3.3.1 Definições . . . 19

3.3.2 Especificações 19
3.3.3 Tarefas 19

4 Definição dos trabalhos TEC básicos computacionais 23
4.1 TEC-BC1: elementos básicos da linguagem 23
4.1.1 Definições 23
4.1.2 Especificações 23
4.1.3 Tarefas 24
4.2 TEC-BC2: tópicos básicos sobre polinômios 24
4.2.1 Definições 24
4.2.2 Especificações 24
4.2.3 Tarefas 29
4.3 TEC-BC3: modelagem matricial para equações 30
4.3.1 Definições 30
4.3.2 Especificações 30
4.3.3 Tarefas 31
4.4 TEC-BC4: tópicos básicos em *scripts* e funções 32
4.4.1 Definições 32
4.4.2 Especificações 32
4.4.3 Tarefas 34

III Trabalhos TEC sobre aplicações básicas 37

5 Definição dos trabalhos TEC sobre gráficos básicos 39
5.1 TEC1: gráficos de curvas de funções unidimensionais 39
5.1.1 Definições 39
5.1.2 Especificações 39
5.1.3 Tarefas 40
5.2 TEC2: gráficos de funções unidimensionais complexas 42
5.2.1 Definições 42
5.2.2 Especificações 42
5.2.3 Tarefas 44
5.3 TEC3: gráficos de superfícieis complexas 46
5.3.1 Definições 46
5.3.2 Especificações 46
5.3.3 Tarefas 48

6 Definição dos trabalhos TEC sobre geração de sons 51
6.1 TEC1: geração de tons, atenuação e efeito Doppler 51
6.1.1 Especificações 51
6.1.2 Tarefas 52

7 Definição dos trabalhos TEC sobre imagens 53
7.1 TEC1: representação matricial de imagens 53
7.1.1 Especificações 53
7.1.2 Tarefas 56
7.2 TEC2: manipulação matricial de imagens 58
7.2.1 Especificações 58
7.2.2 Tarefas 59

Bibliografia **61**

Parte I

Introdução

Capítulo 1

O aplicativo computacional Octave

1.1 Geral

O Octave é um aplicativo computacional.

O Octave pode ser interpretado como uma máquina computacional virtual, que entende uma linguagem própria e a traduz para uma outra linguagem, que é entendida por uma outra máquina computacional, hierarquicamente abaixo dela.

O Octave trabalha com uma linguagem do tipo imperativa. Cada comando da linguagem representa uma assertiva específica, que deve ser traduzida e executada.

A tradução realizada pelo Octave é uma interpretação.

A parte do Octave que interage com o usuário é denominada de interpretador de comandos.

Um conjunto composto por comandos que sejam reconhecidos pelo aplicativo é denominado de programa ou código. Os comandos podem ser passados para o interpretador de comandos de duas formas básicas: interativamente ou por lote (*batch*). Na forma interativa, os comandos são passados individualmente para o interpretador de comandos. Na forma por lote, os comandos são organizados em um arquivo e o nome do arquivo é passado ao interpretador de comandos, que se encarrega de capturar cada um dos comandos do arquivo. De uma forma geral, os comandos são organizados em um arquivo do tipo TEXTO e esse arquivo recebe a denominação de *script*. Um conjunto de comandos que é muitas vezes utilizado por um programa e/ou é compartilhado por diversos programas pode ser organizado em uma função. Do ponto de vista do usuário, uma função é, basicamente, um *script* com uma sintaxe particular.

As estruturas de dados em Octave são dinamicamente tipadas. Não se faz necessária uma declaração antecipada, para a definição de tipo da estrutura e para a reserva de espaço em memória. Basta um simples comando de construção (construtor) ou uma simples atribuição de valores, em tempo de execução, para criá-las ou modificá-las.

1.2 Interpretador de comandos

O interpretador de comandos é a parte do aplicativo que interage com o usuário.

Caso uma expressão seja avaliada sem uma atribuição explícita, a variável padrão 'ans' (*answer* ou resposta) recebe o resultado da avaliação.

Alguns comandos úteis na interação com o interpretador de comandos são os seguintes: iskeyword, lookfor 'pattern', help 'name', who, whos, clear, close, clc.

Algumas funções úteis na interação com o interpretador de comandos são as seguintes: disp(), pause().

Alguns comandos úteis para identificar funções nativas (*built-in*) ou funções armazenadas em arquivos do tipo *M-file* são os seguintes: which 'function', exist 'function' ('built-in' = 5), builtin('function', arg1, arg2, ..., argN).

Algumas funções relacionadas com a versão do computador e o computador são as seguintes: version, computer.

1.3 Tipos de dados

O Octave admite os seguintes tipos de dados:

- logical;
- char;
- numérico inteiro (integer): uint8, uint16, uint32, uint64, int8, int16, int32, int64;
- numérico real (float): single, double.

Em cálculos numéricos, independentemente dos tipos de dados dos operandos, os cálculos são sempre efetuados na maior representação numérica, que é double.

Alguns valores associados com tipos numéricos são os seguintes: eps, realmax, realmin, intmax, intmin, inf, NaN, i ou j, pi.

Algumas funções relacionadas a números complexos são os seguintes: complex(), conj(), real(), imag(), abs(), angle(), unwrap().

Uma função para controle da exibição de dado numérico é a seguinte: format().

1.4 Estruturas de dados

A estrutura de dados básica no Octave é MATRIZ, que é uma estrutura bidimensional retangular. Todos os elementos em uma MATRIZ são do mesmo tipo.

Outras estruturas são definidas a partir de MATRIZ.

As estruturas mais simples são: escalar (matriz 1x1), vetor linha (matriz 1xC), vetor coluna (matriz Lx1), matriz retangular (matriz LxC), matriz quadrada (matriz MxM), matriz multidimensional (matriz D1xD2xD3x...).

No caso de uma matriz tridimensional (matriz D1xD2xD3), as dimensões D1, D2 e D3 são respectivamente denominadas de linha, coluna e página.

Uma dimensão Dn de valor unitário (Dn = 1) é denominada de SINGLETON.

As estruturas mais complexas são arranjos multidimensionais denominados genericamente de ARRAY. Dois tipos de ARRAY são definidos no Octave:

- CELL ARRAY: arranjo multidimensional de matrizes não necessariamente iguais.
- STRUCTURE ARRAY: arranjo multidimensional de uma estrutura de dados similar a STRUCTURE da linguagem C e a RECORD da linguagem Pascal, onde diferentes campos são definidos por um nome e por uma estrutura de dados.

1.5 Funções dedicadas a matrizes

Algumas funções mais comumentes utilizadas são as seguintes:

- Funções relacionadas com tipos especiais de matrizes:
  ones(), zeros(), eye(), diag(), magic(), rand(), randn().

- Funções relacionadas com a concatenação de matrizes:
  cat(), horzcat(), vertcat(), repmat(), blkdiag().

- Funções relacionadas com mudanças na forma de uma matriz:
  reshape(), rot90(), fliplr(), flipud(), flip(), transpose(), ctranspose(), permute().

- Funções relacionadas com deslocamento e ordenação de dados em uma matriz:
  circshift(), sort().

- Funções relacionadas com informações sobre uma matriz:
  ndims(), numel(), size(), length(), find().

1.6 *Download*, instalação e *help*

A seguir, são listadas algumas dicas para *download*, instalação e *help*.

1.6.1 *Download* e instalação

- **Windows**: Acessar a seguinte URL: https://www.gnu.org/software/octave/#install. Procurar arquivo para Windows 64 *bits* (octave-5.2.0_1-w64-installer.exe) ou arquivo para Windows 32 *bits* (octave-5.2.0_1-w32-installer.exe).

- **Linux**: Dependendo da sua distribuição, o *package* executável do Octave varia. As principais distribuições, tais como: Ubuntu, Debian, Fedora e Gentoo, possuem o Octave em suas lojas de aplicativos nativas. Uma outra opção é obter um *package* universal para sistemas linux (https://flathub.org/apps/details/org.octave.Octave).

- **MacOS**: Acessar a seguinte URL: http://wiki.octave.org/Octave_for_macOS. Procurar pelos seguintes arquivos: MacOS App Bundle of Octave 5.2.0 Beta (with GUI) ou macOS App Bundle of Octave 4.4.x (with GUI).

- Podem ser feitas instalações via código fonte (https://ftpmirror.gnu.org/octave).

- *Download* do Octave para todas as versões: https://www.gnu.org/software/octave/#install.

1.6.2 *Help*

- Apostila PET-Tele de *Introdução ao Octave/Matlab*: http://www.telecom.uff.br/pet/. Acessar a aba *Downloads* e, em seguida, escolher o item *Apostilas de Cursos*.
  Idioma: Português.

- *Help* da empresa Mathworks, fabricante do Matlab. Documentação, exemplos e funções: https://www.mathworks.com/support.html?s_tid=gn_supp. Idioma: Inglês.

- *Help* do Octave: https://octave.sourceforge.io/docs.php. Idioma: Inglês.

Capítulo 2

Regras básicas

Para todo e qualquer trabalho definido a seguir, sempre valerão as seguintes regras:

- Todo o material relativo ao trabalho deverá ser enviado simultaneamente para o professor e para o monitor da disciplina, na forma de anexos em uma mensagem de *e-mail*.
- Documentos deverão ser enviados na forma de arquivos no formato PDF.
- Itens não textuais (figuras, tabelas, imagens) deverão ser incluídos nos documentos apresentados.
- Serão ignorados os trabalhos entregues após a data de entrega definida.
- Serão ignorados os trabalhos entregues fora das normas definidas.
- Será atribuída nota nula aos trabalhos com indicação de cópia de outro(s) trabalho(s).

Parte II

Trabalhos TEC básicos

Capítulo 3

Definição dos trabalhos TEC básicos manuais

3.1 TEC-BM1: revisão de matrizes

3.1.1 Definições

- Tempo de execução: 1 semana.
- Título: Revisão de matrizes.
- Objetivo: Revisar tópicos básicos e tópicos importantes para a disciplina de DSP, relativos a matrizes.

3.1.2 Especificações

Definições

- Na área de Processamento Digital de Sinais, as matrizes são largamente empregadas e possuem grande importância, tanto na modelagem teórica quanto nos cálculos numéricos.
- Dados dois números L e C, naturais e não nulos, uma matriz L por C (ou L $\times$ C) é uma tabela M, formada por elementos m_{lc}, onde $1 \leq l \leq L$ e $1 \leq c \leq C$, distribuídos em L linhas e C colunas.
- Notações comuns são $[M]_{L\times C}$, $M = (m_{lc})_{L\times C}$ e, se $L = C = N$, $[M]_{N\times N} = [M]_N$.
- Algumas matrizes ganham denominações especiais, tais como:
  - Escalar: L=1 e C=1.
  - Matriz nula: $m_{lc} = 0$, $\forall\ l, c$.
  - Matriz linha: L=1 e C=N.
  - Matriz coluna: L=N e C=1.
  - Matriz retangular: L>C ou L<C.
  - Matriz quadrada de ordem N: L=C=N.
  - Matriz quadrada diagonal: $m_{lc} = 0$, para $l \neq c$.
  - Matriz quadrada unidade ou identidade ($[I]_N$): matriz quadrada diagonal, onde $m_{lc} = 1$, para $l = c$.

- Para matrizes quadradas de ordem N, duas estruturas se destacam:
  - Diagonal principal: $\{m_{lc} \mid l = c\} = \{m_{11}, m_{22}, m_{33}, \cdots, m_{NN}\}$.
  - Diagonal secundária: $\{m_{lc} \mid (l+c) = (N+1)\} = \{m_{1N}, m_{2(N-1)}, m_{3(N-2)}, \cdots, m_{N1}\}$.

Álgebra de matrizes

- Duas matrizes, $A = (a_{lc})_{L\times C}$ e $B = (b_{lc})_{L\times C}$, são ditas iguais (A=B) somente quando $a_{lc} = b_{lc}$, $\forall\ l, c$.
- A adição de duas matrizes, $A = (a_{lc})_{L\times C}$ e $B = (b_{lc})_{L\times C}$, é uma matriz $M = (m_{lc})_{L\times C}$, dada por $M = A + B$, onde $m_{lc} = a_{lc} + b_{lc}$, $\forall\ l, c$.
- Dada uma matriz $M = (m_{lc})_{L\times C}$, a matriz oposta (ou negativa) de M é dada por $M_{neg} = -M = (-m_{lc})_{L\times C}$.
- O produto de um número (ou escalar) k por uma matriz $A = (a_{lc})_{L\times C}$ é uma matriz $M = (m_{lc})_{L\times C}$, dada por $M = k * A$, onde $m_{lc} = k * a_{lc}$, $\forall\ l, c$.
- O produto de duas matrizes, $A = (a_{lj})_{L\times J}$ e $B = (b_{jc})_{J\times C}$, é uma matriz $M = (m_{lc})_{L\times C}$, dada por $M = A * B$, onde $m_{lc} = \sum_{j=1}^{J}\ a_{lj} \cdot a_{jc}$, $\forall\ l, c$.
- Dada uma matriz $M = (m_{lc})_{L\times C}$, a matriz transposta de M é dada por $M^T = (m_{cl})_{C\times L}$.
- Uma matriz quadrada $[M]_N$ é dita simétrica quando $M^T = M$.
- Uma matriz quadrada $[M]_N$ é dita anti-simétrica quando $M^T = -M$.
- Uma matriz quadrada $[M]_N$ é dita inversível (ou não singular) quando existe uma matriz $[M_{inv}]_N$, tal que $M * M_{inv} = M_{inv} * M = I_N$. Quando não existe M_{inv}, a matriz M é dita não inversível (ou singular). A matriz M_{inv} é chamada de matriz inversa de M.

3.1.3 Tarefas

Teoria

1. Denomine um conjunto ordenado de números $v = \{v_1, v_2, \cdots, v_V\}$ de vetor e discuta (pense, reflita e posicione-se) as seguintes afirmações:
   (a) Um número é uma matriz.
   (b) Um número é uma matriz linha.
   (c) Um número é uma matriz coluna.
   (d) Um número é uma matriz retangular.
   (e) Um número é uma matriz quadrada.
   (f) Um número é um vetor.
   (g) Um vetor é uma matriz.
   (h) Um vetor é uma matriz linha.
   (i) Um vetor é uma matriz coluna.

(j) Um vetor é uma matriz retangular.

(k) Um vetor é uma matriz quadrada.

(l) Uma matriz é um vetor.

(m) Uma matriz linha é uma matriz retangular.

(n) Uma matriz linha é uma matriz quadrada.

(o) Uma matriz coluna é uma matriz retangular.

(p) Uma matriz coluna é uma matriz quadrada.

(q) Uma matriz retangular é uma matriz quadrada.

(r) Uma matriz quadrada é uma matriz retangular.

(s) O número 0 é uma matriz diagonal.

(t) Um número é uma matriz diagonal.

(u) Uma matriz nula $N \times N$, $N > 0$, é uma matriz diagonal.

(2.0 pts)

2. Denomine um conjunto ordenado de números $v = \{v_1, v_2, \cdots, v_V\}$ de vetor e discuta (pense, reflita e posicione-se) as seguintes afirmações:

   (a) Uma variável matemática x pode ser representada por um vetor.

   (b) O produto cartesiano (x, y) pode ser representado por uma matriz.

   (c) Uma função $f(x)$ pode ser representada por um vetor.

   (d) Uma função $f(x, y)$ pode ser representada por uma matriz.

   (e) O gráfico $f(x) \times x$ pode ser representado por uma matriz.

   (f) O tempo pode ser representado por um vetor.

   (g) Um plano pode ser representado por uma matriz.

   (h) Uma superfície pode ser representada por uma matriz.

   (i) Um sinal de voz pode ser representado por um vetor.

   (j) Uma fotografia pode ser representada por uma matriz.

   (k) Um alfabeto pode ser representado por um vetor.

   (l) Uma palavra pode ser representada por um vetor.

   (m) Uma frase pode ser representada por um vetor.

   (n) Um parágrafo pode ser representado por um vetor.

   (o) Um texto pode ser representado por um vetor.

   (p) Uma página de livro pode ser representada por uma matriz.

   (2.0 pts)

Prática

Considere: o escalar $e = 3$; os vetores $v_1 = [1, 3, 5]$ e $v_2 = \begin{bmatrix} 2 \\ 6 \\ 10 \end{bmatrix}$;
as matrizes $M_r = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$, $M_q = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$, $P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ e $S = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

Calcule, manualmente, as seguintes operações:

1. $e * v_1$.
2. $v_1 * e$.
3. $e * v_2$.
4. $v_2 * e$.
5. $e * M_r$.
6. $M_r * e$.
7. $e * M_q$.
8. $M_q * e$.

9. $v_1 * v_2$.
10. $v_2 * v_1$.

11. $v_1 * M_q$.
12. $M_q * v_2$.
13. $v_1 * M_r$.
14. $M_q * M_r$.

15. $v_2^T * v_1^T$.
16. $v_1^T * v_2^T$.

17. $M_q^T * v_1^T$.
18. $v_2^T * M_q^T$.
19. $M_r^T * v_1^T$.
20. $M_r^T * M_q^T$.

21. $P * M_q$.
22. $M_q * P$.
23. $P * M_q * P$.

24. $S * M_q$.
25. $M_q * S$.
26. $S * M_q * S$.

(4.0 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Desenvolva a parte textual em um editor de texto. Desenvolva os cálculos manualmente, digitalize-os e incluaos no texto editado, na forma de imagens. (2.0 pts)

3.2 TEC-BM2: revisão de números complexos

3.2.1 Definições

- Tempo de execução: 1 semana.
- Título: Revisão de números complexos.
- Objetivo: Revisar tópicos básicos e tópicos importantes sobre números complexos para o seu uso em diversas aplicações.

3.2.2 Especificações

- Os números complexos são largamente empregados nas mais diversas áreas do conhecimento e possuem grande importância na modelagem teórica.
- Na sua definição básica, os números complexos são pares ordenados de números reais.
- Para tal, os pares ordenados devem obedecer a operações aritméticas complexas que são definidas a partir das operações aritméticas reais.
- Os números complexos também podem ser representados em uma forma algébrica (ou retangular).
- A partir da forma algébrica, são definidas as partes real e imaginária de um número complexo, bem como é definida a seguinte nomenclatura: número complexo, número real (puro) e número imaginário (puro).
- Alternativamente, os números complexos ainda podem ser interpretados a partir de uma visão geométrica.
- Com a visão geométrica, os números complexos podem ser diretamente associados com as funções trigonométricas e exponenciais.

3.2.3 Tarefas

Teoria

1. Considerando a definição básica de pares ordenados de números reais, apresente a definição de igualdade e das operações de adição e de multiplicação para números complexos.
2. Na representação geométrica alternativa para os números complexos, descreva as formas retangular (ou algébrica) e polar (ou trigonométrica).
3. Na representação polar, destaque as relações trigonométricas e exponencial (Equação de Euler) .
4. Apresente as operações de multiplicação, divisão, potenciação e radiciação, com base na representação geométrica.

Prática

1. Dados os números complexos $z_k = (a_k, b_k) = (a_k + j\ b_k)$, atenda aos seguintes itens:

   (a) Demonstre que $(z_1 + z_2) + z_3 = z_1 + (z_2 + z_3)$.

   (b) Demonstre que $(z_1 + z_2) = (z_2 + z_1)$.

   (c) Calcule o elemento neutro para a adição z_{en_A}, tal que $z_k + z_{en_A} = z_k$.

   (d) Calcule o elemento inverso para a adição z_{inv_A}, tal que $z_k + z_{inv_A} = z_{en_A}$.

   (1.0 pts)

2. Dados os números complexos $z_k = (a_k, b_k) = (a_k + j\ b_k)$, atenda aos seguintes itens:

   (a) Demonstre que $(z_1 \cdot z_2) \cdot z_3 = z_1 \cdot (z_2 \cdot z_3)$.

   (b) Demonstre que $(z_1 \cdot z_2) = (z_2 \cdot z_1)$.

   (c) Calcule o elemento neutro para a multiplicação z_{en_M}, tal que $z_k \cdot z_{en_M} = z_k$.

   (d) Calcule o elemento inverso para a multiplicação z_{inv_M}, tal que $z_k \cdot z_{inv_M} = z_{en_M}$, para $a_k \neq 0$ ou $b_k \neq 0$.

   (e) Demonstre que $z_1 \cdot (z_2 + z_3) = (z_1 \cdot z_2) + (z_1 \cdot z_3)$.

   (1.0 pts)

3. Considere que a subtração é definida por $z = z_1 - z_2$, de tal forma que $z_1 = z + z_2$. Dados os números complexos $z_k = (a_k, b_k) = (a_k + j\ b_k)$, onde $Re\{z_k\} = a_k$ e $Im\{z_k\} = b_k$, e os números complexos conjugados $z_k^* = (a_k, -b_k) = (a_k - j\ b_k)$ atenda aos seguintes itens:

   (a) Demonstre que: Se $z_k = z_k^*$, então $z \in \mathbb{R}$.

   (b) Demonstre que $(z_k + z_k^*) = 2\ Re\{z_k\}$.

   (c) Demonstre que $(z_k - z_k^*) = j\ 2\ Im\{z_k\}$.

   (d) Demonstre que $(z_k \cdot z_k^*) = a_k^2 + b_k^2$.

   (e) Demonstre que $(z_1 + z_2)^* = (z_1^* + z_2^*)$.

   (f) Demonstre que $(z_1 \cdot z_2)^* = (z_1^* \cdot z_2^*)$.

   (1.0 pts)

4. Considere que a subtração é definida por $z = z_1 - z_2$, de tal forma que $z_1 = z + z_2$. Considere que a divisão é definida por $z = z_1/z_2$, de tal forma que $z_1 = z \cdot z_2$. Dados os números complexos $z_k = (a_k, b_k) = (a_k + j\ b_k)$, atenda aos seguintes itens:

   (a) Calcule $(z_1 + z_2)$.
   (b) Calcule $(z_1 - z_2)$.
   (c) Calcule $(z_1 \cdot z_2)$.
   (d) Calcule (z_1/z_2).
   (e) Calcule $(z_1^* + z_2^*)$.
   (f) Calcule $(z_1^* - z_2^*)$.
   (g) Calcule $(z_1^* \cdot z_2^*)$.
   (h) Calcule (z_1^*/z_2^*).
   (i) Calcule $(z_1 + z_2)^*$.
   (j) Calcule $(z_1 - z_2)^*$.
   (k) Calcule $(z_1 \cdot z_2)^*$.
   (l) Calcule $(z_1/z_2)^*$.

   (1.0 pts)

5. Calcule os números complexos $z_k = j^k$, para $k = \{0, 1, 2, 3\}$, dado que $j = (0, 1)$, empregando as seguintes representações:

   (a) Par ordenado.
   (b) Forma algébrica (ou retangular).
   (c) Forma trigonométrica (ou polar).

   (1.0 pts)

6. Prove as seguintes relações:

   (a) $\left(1 - e^{-j2\theta}\right) e^{j\theta} = 2j\ sin(\theta)$.
   (b) $\left(e^{j2\theta} - 1\right) e^{-j\theta} = 2j\ sin(\theta)$.
   (c) $\left(1 + e^{-j2\theta}\right) e^{j\theta} = 2\ cos(\theta)$.
   (d) $\left(e^{j2\theta} + 1\right) e^{-j\theta} = 2\ cos(\theta)$.

   (1.0 pts)

7. Em um plano complexo, esboce o gráfico dos números complexos $z_{kl} = r_k \cdot e^{j\Theta_{N_l}}$, identificando cada um deles, dados $\boldsymbol{r} = \{r_1, r_2, r_3\} = \{0.5, 1, 2\}$ e $\Theta_{N_l} = \frac{2\pi}{N_l}$, onde $\boldsymbol{N} = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}\} = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 12\}$.

   (1.0 pts)

8. Esboce o gráfico, no plano complexo, do lugar geométrico definido por $|z| = 1$.

   (0.5 pts)

9. Dada a função (seqüência em n) complexa $x[n] = e^{j\Theta[n]}$, onde $\Theta[n] = n\Theta_N$, $\Theta_N = \frac{2\pi}{N}$, $-2N \leq n \leq (2N-1)$ e $N = \{3, 4, 6, 8\}$, atenda aos seguintes itens para cada valor de N:

   (a) Esboce o gráfico, no plano complexo, de $x[n]$. Indique o valor de n em cada ponto do gráfico.

   (b) Esboce o gráfico $|x[n]| \times n$.

   (c) Esboce o gráfico $\angle x[n] \times n$, usando:

      i. Valores absolutos (*unwrapped*).

      ii. Valores principais na faixa $[-\pi; \pi]$ (*wrapped around*).

   (d) Esboce o gráfico $Re\{x[n]\} \times n$.

   (e) Esboce o gráfico $Im\{x[n]\} \times n$.

   (1.5 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Desenvolva a parte textual em um editor de texto. Desenhe os gráficos manualmente, digitalize-os e inclua-os no texto editado, na forma de imagens. (1.0 pts)

3.3 TEC-BM3: operações básicas sobre seqüências

3.3.1 Definições

- Tempo de execução: 1 semana.
- Título: Operações básicas sobre seqüências.
- Objetivo: Identificar especificidades em algumas operações básicas sobre seqüências e manualmente realizar tais operações sobre seqüências fornecidas.

3.3.2 Especificações

- Em sistemas de processamento digital de sinais, os sinais de entrada, os sinais internos ao sistema e o sinal de saída, são seqüências ordenadas de valores numéricos.
- Algumas das operações básicas, realizadas por tais sistemas sobre as seqüências, são as seguintes:
  - Adaptação de comprimento do sinal, com adição de zeros.
  - Adição de sinais.
  - Multiplicação de sinal por constante (escalamento).
  - Deslocamento linear de sinal não periódico.
  - Deslocamento linear de sinal periódico.
  - Deslocamento circular de sinal não periódico.
  - Espelhamento linear de sinal não periódico.
  - Espelhamento linear de sinal periódico.
  - Espelhamento circular de sinal não periódico.
  - Extensão periódica.
  - Convolução linear (ou não periódica).
  - Convolução circular (ou periódica).

3.3.3 Tarefas

Teoria

1. Os somatórios dos produtos de duas seqüências, $h[n]$ e $x[k]$, definidos por

$$y[n] = \sum_{k=K_1}^{K_2} h[n-k] \; x[k] \tag{3.1}$$

e

$$y[n] = \sum_{k=K_1}^{K_2} x[n-k] \; h[k] \; , \tag{3.2}$$

podem ser descritos por relações matriciais.

2. Considere as seguintes faixas de valores: $-3 \leq n \leq 3$ e $-5 \leq k \leq 5$.

3. Considerando os vetores $\boldsymbol{y}[n]$ e $\boldsymbol{x}[n]$, respectivamente formados pelos valores das seqüências $y[n]$ e $x[n]$, descreva o somatório de produtos da Equação (3.1) pela relação matricial $\boldsymbol{y}[n] = \boldsymbol{H} \cdot \boldsymbol{x}[n]$, onde a matriz $\boldsymbol{H}$ é formada pelos valores da seqüência $h[n]$.

4. Considerando os vetores $\boldsymbol{y}[n]$ e $\boldsymbol{h}[n]$, respectivamente formados pelos valores das seqüências $y[n]$ e $h[n]$, descreva o somatório de produtos da Equação (3.2) pela relação matricial $\boldsymbol{y}[n] = \boldsymbol{X} \cdot \boldsymbol{h}[n]$, onde a matriz $\boldsymbol{X}$ é formada pelos valores da seqüência $x[n]$.

Prática

1. Considere os sinais $h[n] = [7, 5, 3, 0, -1, -2, 1]$, para $n = [0, 1, 2, 3, 4, 5, 6]$, $x[n] = [1, 2, 1]$, para $n = [2, 3, 4]$, e $h[n] = x[n] = 0$, para os demais valores de n.

2. Considere os sinais periódicos $\tilde{h}[n]$ e $\tilde{x}[n]$, respectivamente formados pelas extensões periódicas de $h[n]$ e $x[n]$, com $N_f = 7$.

3. Para a faixa $-12 \leq n \leq 12$, esboce, contendo todas as indicações pertinentes, os seguintes gráficos:

   (a) $x[n] \times n$.
   (b) $h[n] \times n$.
   (c) $x[-n] \times n$.
   (d) $h[-n] \times n$.
   (e) $h[n-3] \times n$.
   (f) $h[n+5] \times n$.
   (g) $h[-n+2] \times n$.
   (h) $h[-n-1] \times n$.

   (2.00 pts)

4. Para a faixa $-12 \leq n \leq 12$, esboce, contendo todas as indicações pertinentes, os seguintes gráficos:

   (a) $\tilde{x}[n] \times n$.
   (b) $\tilde{h}[n] \times n$.
   (c) $\tilde{x}[-n] \times n$.
   (d) $\tilde{h}[-n] \times n$.
   (e) $\tilde{h}[n-3] \times n$.
   (f) $\tilde{h}[n+5] \times n$.
   (g) $\tilde{h}[-n+2] \times n$.
   (h) $\tilde{h}[-n-1] \times n$.

   (2.00 pts)

5. Para a faixa $0 \leq n \leq 6$, esboce, contendo todas as indicações pertinentes, os seguintes gráficos:

   (a) $x[\langle n \rangle_7] \times n$.

   (b) $h[\langle n \rangle_7] \times n$.

   (c) $x[\langle -n \rangle_7] \times n$.

   (d) $h[\langle -n \rangle_7] \times n$.

   (e) $h[\langle n-3 \rangle_7] \times n$.

   (f) $h[\langle n+5 \rangle_7] \times n$.

   (g) $h[\langle -n+2 \rangle_7] \times n$.

   (h) $h[\langle -n-1 \rangle_7] \times n$.

   (2.00 pts)

6. Para o cálculo da convolução $y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[n-k]\ x[k]$, realize os seguintes passos:

   (a) Monte a operação na forma matricial, para $-2 \leq n \leq 12$.

   (b) Demonstre, por meio dos gráficos pertinentes, cada um dos cálculos efetuados na operação matricial.

   (2.00 pts)

7. Para o cálculo da convolução $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[n-k]\ h[k]$, realize os seguintes passos:

   (a) Monte a operação na forma matricial, para $-2 \leq n \leq 12$.

   (b) Demonstre, por meio dos gráficos pertinentes, cada um dos cálculos efetuados na operação matricial.

   (2.00 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Desenvolva a parte textual em um editor de texto. Desenhe os gráficos manualmente, digitalize-os e inclua-os no texto editado, na forma de imagens. (2.0 pts)

Capítulo 4

Definição dos trabalhos TEC básicos computacionais

4.1 TEC-BC1: elementos básicos da linguagem

4.1.1 Definições

- Tempo de execução: 1 semana.
- Título: Pesquisa e geração de exemplos simples envolvendo alguns elementos básicos (valores/comandos/funções) da linguagem do aplicativo Octave.
- Objetivo: Adquirir conhecimento e estabelecer domínio sobre alguns elementos básicos (valores/comandos/funções) da linguagem do aplicativo Octave.

4.1.2 Especificações

- O Octave é um aplicativo computacional, que trabalha com um processo de interpretação de uma linguagem imperativa.
- O trabalho em questão visa explorar alguns elementos básicos (valores/comandos/funções) da linguagem do aplicativo Octave.
- Suponha os seguintes valores definidos no Octave: eps, realmax, realmin, intmax, intmin, inf, NaN, i ou j, pi.
- Suponha os seguintes comandos definidos no Octave: iskeyword, lookfor 'pattern', help 'name', who, whos, clear, close, clc.
- Suponha as seguintes funções definidas no Octave:
  - disp(), pause().
  - complex(), conj(), real(), imag(), abs(), angle(), unwrap().
  - format().
  - ones(), zeros(), eye(), diag(), magic(), rand(), randn().
  - cat(), horzcat(), vertcat(), repmat(), blkdiag().
  - reshape(), rot90(), fliplr(), flipud(), flip(), transpose(), ctranspose(), permute().

– circshift(), sort().

– ndims(), numel(), size(), length(), find().

4.1.3 Tarefas

1. Pesquise e escreva uma breve descrição sobre cada um dos elementos listados acima. (1.0 + 1.0 + 2.0 pts)

2. Desenvolva exemplos simples de código Octave que ilustrem a operação dos elementos listados acima. (1.0 + 1.0 + 2.0 pts)

3. Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

4.2 TEC-BC2: tópicos básicos sobre polinômios

4.2.1 Definições

- Tempo de execução: 1 semana.
- Título: Tópicos básicos sobre polinômios.
- Objetivo: Trabalhar alguns dos tópicos básicos sobre polinômios.

4.2.2 Especificações

Conceitos básicos

- Do ponto de vista algébrico, polinômios p são entidades únicas, simples e fixas. Dado um símbolo abstrato x e as quantidades a_k, outros símbolos podem ser formados, empregando-se apenas as operações de adição, subtração e multiplicação, sobre x e a_k, tais como: $(a_1 x)$, $(a_2 x^2)$, $(a_3 x^3 + a_0)$. De uma forma geral, um polinômio $p(x)$ pode ser definido por

$$p(x) = a_N x^N + a_{N-1} x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0 = \sum_{k=0}^{N} a_k x^k \ . \tag{4.1}$$

- Do ponto de vista funcional, $p(x)$ pode ser interpretado como a função ou o mapeamento $x \to p(x)$, onde um valor da variável x é levado a um valor de $p(x)$, por meio da avaliação da equação de definição de $p(x)$ com o valor desejado de x. Nesse caso, as quantidades a_k são denominadas de coeficientes e as parcelas $a_k x^k$ são chamadas de termos de $p(x)$.

- A partir da lei de formação dada por (4.1), pode-se identificar um polinômio $p(x)$ apenas por uma sequência formada por seus coeficientes, de tal forma que

$$p(x) = [a_N, a_{N-1}, \cdots, a_2, a_1, a_0]$$

  ou

$$p(x) = [a_0, a_1, a_2, \cdots, a_{N-1}, a_N] \ .$$

- Na Equação (4.1), o coeficiente a_N é denominado de coeficiente dominante. Se $a_N \neq 0$, $p(x)$ é classificado como um polinômio de grau N. Se $a_N = 1$, $p(x)$ é classificado como um polinômio mônico. Se $a_k = 0$, $0 \leq k \leq N$, $p(x)$ é um polinômio nulo.

- A Equação (4.1) pode ser fatorada em

$$\begin{aligned} p(x) &= a_N x^N + a_{N-1} x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0 \\ &= a_N (x - r_1)(x - r_2) \cdots (x - r_{N-1})(x - r_N) \ , \end{aligned} \tag{4.2}$$

onde os valores r_k são denominados de raízes ou zeros de $p(x)$.

Operações básicas

- A adição de polinômios é definida por

$$p(x) = [f + g](x) = f(x) + g(x) = \sum_{k=0}^{N} a_k x^k + \sum_{k=0}^{N} b_k x^k = \sum_{k=0}^{N} (a_k + b_k) x^k = \sum_{k=0}^{N} c_k x^k \ ,$$

onde $c_k = (a_k + b_k)$.

- A subtração de polinômios é definida por

$$p(x) = [f - g](x) = f(x) - g(x) = \sum_{k=0}^{N} a_k x^k - \sum_{k=0}^{N} b_k x^k = \sum_{k=0}^{N} (a_k - b_k) x^k = \sum_{k=0}^{N} c_k x^k \ ,$$

onde $c_k = (a_k - b_k)$.

- A multiplicação de polinômios é definida por

$$p(x) = [f * g](x) = f(x) * g(x) = \sum_{k=0}^{M} a_k x^k * \sum_{k=0}^{N} b_k x^k = \sum_{k=0}^{M+N} c_k x^k \ ,$$

onde $c_k = \sum_{i=0}^{k} (a_i + b_{k-i})$.

- A divisão ou razão de polinômios é definida por meio da multiplicacão e da adição, empregando-se a relação

$$Dividendo = (divisor * quociente) + resto \ .$$

Assim, a razão polinomial $D(x)/d(x)$, $d(x) \neq 0$, é satisfeita pelas seguintes condições:

1. $D(x) = d(x) * q(x) + r(x)$.
2. grau de $r(x) <$ grau de $d(x)$ ou $r(x) = 0$.

Quando $r(x) = 0$, a divisão é dita exata.

Funções polinomiais racionais

- A divisão ou razão de dois polinômios $N_H(x)$ e $D_H(x)$ pode ser interpretada como um terceiro polinômio $H(x)$, definido pela função polinomial racional

$$\begin{aligned} H(x) &= \frac{N_H(x)}{D_H(x)} = \frac{\sum_{k=0}^{L} b_k x^k}{\sum_{k=0}^{N} a_k x^k} \\ &= \frac{b_L x^L + b_{L-1} x^{L-1} + \cdots + b_2 x^2 + b_1 x + b_0}{a_N x^N + a_{N-1} x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0} \\ &= \frac{b_L (x - z_1)(x - z_2) \cdots (x - z_{N-1})(x - z_L)}{a_N (x - p_1)(x - p_2) \cdots (x - p_{N-1})(x - p_N)} \\ &= K_H \, \frac{(x - z_1)(x - z_2) \cdots (x - z_{N-1})(x - z_L)}{(x - p_1)(x - p_2) \cdots (x - p_{N-1})(x - p_N)} \ , \end{aligned} \tag{4.3}$$

  onde $K_H = b_L / a_N$.

- O valor K_H é denominado ganho de $H(x)$.

- Os valores z_k são as raízes ou os zeros do polinômio numerador $N_H(x)$ e, portanto, são também as raízes ou os zeros de $H(x)$.

- Os valores p_k são as raízes ou os zeros do polinômio denominador $D_H(x)$ e, portanto, fazem com que os valores $H(p_k)$ tendam a infinito. Eles são chamados de pólos de $H(x)$.

- O conjunto de zeros e pólos é denominado de singularidades de $H(x)$. Considerando-se as singularidades com valores finitos e infinitos, o número total de zeros e o número total de pólos de $H(x)$ é sempre igual.

- Quando $L < N$, a razão é dita própria. Caso contrário, ela é uma razão imprópria. Efetuando-se uma divisão polinomial, uma razão imprópria pode ser decomposta na soma de um polinômio com uma razão própria, dada por

$$R_I(x) = \frac{\sum_{k=0}^{L} b_k x^k}{\sum_{k=0}^{N} a_k x^k} = \sum_{k=0}^{K} r_k x^k + \frac{\sum_{k=0}^{M} d_k x^k}{\sum_{k=0}^{N} c_k x^k} = P(x) + R_P(x) \ ,$$

  onde $L \geq N$ e $M < N$.

- Uma função polinomial racional própria ($L < N$) pode ser decomposta em frações parciais. Por exemplo, considerando-se que todos os pólos são simples (distintos) pode-se efetuar a seguinte decomposição:

$$\begin{aligned} H(x) &= \frac{(x - z_1)(x - z_2) \cdots (x - z_{N-1})(x - z_L)}{(x - p_1)(x - p_2) \cdots (x - p_{N-1})(x - p_N)} \\ &= \frac{K_1}{(x - p_1)} + \frac{K_2}{(x - p_2)} + \cdots + \frac{K_N}{(x - p_N)} \ , \end{aligned} \tag{4.4}$$

  onde a constante K_k é denominada de resíduo do pólo p_k.

Relações entre polinômios e matrizes

- Matematicamente, os polinômios e as matrizes são entidades naturalmente conectadas. Consequentemente, essa ligação aparece também nas suas aplicações, tais como: Teoria de Circuitos, Teoria de Sistemas, Teoria de Controle.

- Dada a matriz quadrada, de ordem N,

$$\boldsymbol{C_p} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{N-2} & -a_{N-1} \end{bmatrix} ,$$

  a matriz

$$\boldsymbol{M_C} = (x\boldsymbol{I} - \boldsymbol{C_p}) = \begin{bmatrix} x & -1 & 0 & \cdots & 0 & 0 \\ 0 & x & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ a_0 & a_1 & a_2 & \cdots & a_{N-2} & x + a_{N-1} \end{bmatrix} ,$$

  é denominada de matriz característica de $\boldsymbol{C_p}$.

- O polinômio

$$\begin{aligned} p(x) &= det(\boldsymbol{M_C}) = det(x\boldsymbol{I} - \boldsymbol{C_p}) \\ &= x^N + a_{N-1}x^{N-1} + \cdots + a_2x^2 + a_1x + a_0 \\ &= (x - r_1)(x - r_2)\cdots(x - r_{N-1})(x - r_N) \end{aligned} \tag{4.5}$$

  é chamado de polinômio característico de $\boldsymbol{C_p}$, onde $det(\boldsymbol{M_C})$ significa o determinante da matriz $\boldsymbol{M_C}$.

- Os valores r_k em (4.5) recebem duas denominações. Por um lado, eles são as raízes ou os zeros do polinômio $p(x)$. Além disso, eles são definidos como as raízes características ou os autovalores da matriz $\boldsymbol{C_p}$.

- Nesse contexto, $\boldsymbol{C_p}$ é dita a matriz companheira do polinômio $p(x)$.

Similaridade entre matrizes

- Dada uma matriz quadrada $\boldsymbol{M}$, de ordem N, um vetor $\boldsymbol{v}$, de comprimento N, e um escalar λ, para os quais vale a relação

$$\boldsymbol{M}\boldsymbol{v} = \lambda\boldsymbol{v} ,$$

  diz-se que λ é um valor característico ou autovalor de $\boldsymbol{M}$ e que $\boldsymbol{v}$ é o vetor característico ou autovetor de $\boldsymbol{M}$ associado a λ.

- Duas matrizes quadradas $\boldsymbol{A}$ e $\boldsymbol{B}$, ambas de ordem N, são ditas similares quando existe uma matriz $\boldsymbol{Q}$, não singular, tal que

$$\boldsymbol{B} = \boldsymbol{Q}^{-1}\boldsymbol{A}\boldsymbol{Q}$$

e

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^{-1} \\ &= (\boldsymbol{Q}^{-1})^{-1}\boldsymbol{B}(\boldsymbol{Q}^{-1}) \\ &= \boldsymbol{P}^{-1}\boldsymbol{B}\boldsymbol{P} \ , \end{aligned}$$

onde $\boldsymbol{P} = \boldsymbol{Q}^{-1}$. Esse processo de obtenção de $\boldsymbol{B}$ a partir de $\boldsymbol{A}$, e vice-versa, é denominado de transformação de similaridade.

- Dada uma matriz quadrada $\boldsymbol{M}$, de ordem N, pode-se encontrar uma matriz companheira $\boldsymbol{C_p}$, similar a M, tal que

$$\boldsymbol{C_p} = \boldsymbol{Q}^{-1}\boldsymbol{M}\boldsymbol{Q} \ .$$

- Dada uma matriz companheira $\boldsymbol{C_p}$, de ordem N, com N autovalores distintos λ_k e seus N autovetores associados $\boldsymbol{v}_k$, pode-se mostrar que ela possui uma matriz similar diagonal, dada por

$$\boldsymbol{D_p} = \boldsymbol{V}^{-1}\boldsymbol{C_p}\boldsymbol{V} = \boldsymbol{\lambda}\boldsymbol{I} \ ,$$

onde $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \ \cdots \ \lambda_N]$ e $\boldsymbol{V} = [\boldsymbol{v}_1 \ \boldsymbol{v}_2 \ \cdots \boldsymbol{v}_N]$.

- Portanto, dada uma matriz quadrada $\boldsymbol{M}$, de ordem N, com N autovalores distintos λ_k, pode-se mostrar que ela possui uma matriz similar diagonal, dada por

$$\boldsymbol{D_p} = \boldsymbol{V}^{-1}\boldsymbol{C_p}\boldsymbol{V} = \boldsymbol{V}^{-1}\boldsymbol{Q}^{-1}\boldsymbol{M}\boldsymbol{Q}\boldsymbol{V} = \boldsymbol{R}^{-1}\boldsymbol{M}\boldsymbol{R} = \boldsymbol{\lambda}\boldsymbol{I} \ ,$$

onde $\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{V}$.

- Pode-se mostrar que as matrizes similares $\boldsymbol{M}$, $\boldsymbol{C_p}$ e $\boldsymbol{D_p}$, acima definidas, possuem o mesmo polinômio característico $p(x)$.

- Empregando-se transformações de similaridade, a matriz companheira C_p pode assumir outras formas equivalentes.

Funções para polinômios no Octave

- O Octave oferece um conjunto de funções que operam sobre polinômios, tais como: roots(), poly(), polyval(), polyvalm(), polyout(), conv(), deconv(), polyreduce(), residue(), eig(), compan(), polyfit(), polyder(), polyint().

- Para interagir com tais funções, os polinômios são representados por vetores contendo seus coeficientes, organizados em uma dada ordem. Para algumas funções, os parâmetros de entrada, e de saída, são vetores do tipo linha ou do tipo coluna. Por sua vez, os coeficientes podem ser organizados em ordem crescente ou decrescente. Portanto, é sempre recomendável que se leia o manual (*help*) da função, antes de utilizá-la.

- Deve ser lembrado de que os coeficientes nulos devem ser explicitamente representados. Por exemplo: $p(x) = x^5 + 3x^2 + 2 \rightarrow v = [\ 1\ 0\ 0\ 3\ 0\ 2\ ]$.

4.2.3 Tarefas

Teoria

1. Dados um vetor genérico $\boldsymbol{v} = [\ v_1\ v_2\ v_3\ ]$, uma matriz quadrada genérica $\boldsymbol{M} = (m_{lc})_{3\times 3}$ e a matriz $Q = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$, atenda aos seguintes itens:
   - Calcule $\boldsymbol{Q}^{-1}$.
   - Calcule $\boldsymbol{Q}\boldsymbol{v}^T$, $\boldsymbol{v}\boldsymbol{Q}$, $\boldsymbol{Q}^{-1}\boldsymbol{M}$, $\boldsymbol{M}\boldsymbol{Q}$, $\boldsymbol{Q}^{-1}\boldsymbol{M}\boldsymbol{Q}$.
   - Compare o resultado de cada uma das multiplicações anteriores com as funções de *flip*, disponíveis no Octave.

   (1.0 pts)

Prática

1. Para todas as tarefas abaixo, considere que o usuário irá fornecer os dados de entrada. Para facilitar os testes, durante o desenvolvimento, use dados definidos internamente ao código.
2. Desenvolva um código Octave que receba os coeficientes de um polinômio e calcule o seu coeficiente dominante e as suas raízes. (0.5 pts)
3. Desenvolva um código Octave que receba o coeficiente dominante e as raízes de um polinômio e calcule os seu coeficientes. (0.5 pts)
4. Desenvolva um código Octave que receba os coeficientes de um polinômio e um conjunto de pontos e calcule o valor do polinômio nos pontos especificados. (0.5 pts)
5. Desenvolva um código Octave que receba os coeficientes dos polinômios divisor, quociente e resto, sem zeros à esquerda, e calcule os coeficientes do polinômio dividendo. (0.5 pts)
6. Desenvolva um código Octave que receba os coeficientes dos polinômios dividendo e divisor e calcule os coeficientes dos polinômios quociente e resto, sem zeros à esquerda. (0.5 pts)
7. Desenvolva um código Octave que receba os coeficientes dos polinômios numerador e denominador, de uma função polinomial racional, e calcule o ganho, os zeros e os pólos da função. (0.5 pts)
8. Desenvolva um código Octave que receba os coeficientes dos polinômios numerador e denominador, de uma função polinomial racional imprópria, e calcule o polinômio e a função polinomial racional própria, relacionados à função original. (1.0 pts)
9. Desenvolva um código Octave que receba os coeficientes de um polinômio e de uma função polinomial racional própria, e calcule os coeficientes dos polinômios numerador e denominador, da função polinomial racional imprópria, relacionada aos polinômios originais. (1.0 pts)
10. Desenvolva um código Octave que receba os coeficientes dos polinômios numerador e denominador, de uma função polinomial racional própria, com pólos simples, e calcule o seu desenvolvimento em frações parciais. (0.5 pts)

11. Desenvolva um código Octave que receba os coeficientes de um polinômio e calcule a sua matriz companheira. (0.5 pts)

12. Desenvolva um código Octave que receba uma matriz na forma companheira e calcule os coeficientes do seu polinômio característico. (0.5 pts)

13. Desenvolva um código Octave que receba uma matriz na forma companheira e calcule as raízes do seu polinômio característico. (0.5 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

4.3 TEC-BC3: modelagem matricial para equações

4.3.1 Definições

- Tempo de execução: 1 semana.

- Título: Modelagem matricial para equações.

- Objetivo: Elaborar um modelo matricial para um conjunto de equações e realizar os devidos cálculos.

4.3.2 Especificações

- Em Processamento Digital de Sinais, um par de equações de elevada importância é dado por

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \; e^{jk\left(\frac{2\pi}{N}\right)n}, \text{ para } 0 \leq n \leq N-1, \tag{4.6}$$

e

$$X[k] = \sum_{n=0}^{N-1} x[n] \; e^{-jk\left(\frac{2\pi}{N}\right)n}, \text{ para } 0 \leq k \leq N-1 \; . \tag{4.7}$$

- As equações acima realizam uma transformação da seqüência $x[n]$ na seqüência $X[k]$ e vice-versa.

- Para o cálculo das equações acima, o primeiro passo é a escolha de um valor para a constante N.

- Em seguida, o cálculo de um número finito de pontos pode ser efetuado por meio de um número finito de operações.

4.3.3 Tarefas

Teoria

1. Considere a constante $W_N = e^{-j\left(\frac{2\pi}{N}\right)}$. Reescreva as equações acima. (0.5 pts)

2. Considere os vetores coluna $x = [x[0], x[1], ..., x[N-1]]'$ e $X = [X[0], X[1], ..., X[N-1]]'$. Reescreva as equações contendo W_N na forma matricial dada por

$$x = D_{inv} \; X \tag{4.8}$$

   e

$$X = D_{dir} \; x \; . \tag{4.9}$$

   (1.0 pts)

3. Estabeleça uma relação entre as matrizes D_{inv} e D_{dir}. (0.5 pts)

Prática

1. Desenvolva um código Octave que receba um valor para a variável N e calcule as matrizes D_{inv} e D_{dir}. (3.0 pts)

2. Desenvolva um código Octave extra, a ser anexado ao código anterior, que receba um vetor x e calcule o vetor X correspondente ou que receba um vetor X e calcule o vetor x correspondente. Se o comprimento L do vetor recebido for menor que N, ele deverá ser completado com valores nulos de $L + 1$ até N (operação de *zero padding*). Se o comprimento L do vetor recebido for maior que N, deverão ser utilizados apenas os N primeiros valores. (3.0 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

4.4 TEC-BC4: tópicos básicos em *scripts* e funções

4.4.1 Definições

- Tempo de execução: 1 semana.
- Título: Tópicos básicos em *scripts* e funções.
- Objetivo: Pesquisar, estudar e trabalhar os tópicos básicos em *scripts* e funções.

4.4.2 Especificações

- O Octave opera com base em uma linguagem imperativa de comandos interpretados e em dados que são armazenados em variáveis com estrutura essencialmente matricial.
- A parte da memória onde são armazenadas as variáveis manipuladas por um determinado conjunto de comandos recebe o nome de *workspace*.
- A janela de comandos do Octave possui o seu próprio *workspace*, denominado de *base workspace*. Dependendo de como os comandos são organizados e executados, outros *workspaces* podem ser criados.
- Há duas maneiras básicas para executar um conjunto de comandos no Octave.
- No caso de um pequeno número de comandos, eles podem ser simplesmente digitados, e individualmente executados, na janela de comandos. Os dados manipulados por esses comandos são alocados no *base workspace*.
- Para um grande número de comandos, eles podem ser armazenados em arquivos, para uma futura execução. Tais arquivos executáveis devem possuir extensão ".m" e dividem-se em dois grupos, que são: *script* e função.
- Um *script* é apenas um agrupamento de comandos, sem qualquer estrutura extra. Assim, os comandos nele presentes são executados como se fossem simplesmente digitados na janela de comandos. Porém, as variáveis por ele manipuladas residem no *workspace* do código que o executou. Caso o *script* seja executado a partir da janela de comando, será utilizado o *base workspace*.
- As funções são blocos estruturados de comandos, com regras de construção bem definidas. Cada função possui um *workspace* individual, denominado de *function workspace*, onde residem as variáveis por ela manipuladas. Dependendo de como são definidas, as funções podem receber dados de entrada (argumentos) e podem gerar dados de saída (retornos). Uma vez que essas variáveis também são locais ao *function workspace*, os dados são passados por cópias dos conteúdos das variáveis.
- Alguns itens operacionais básicos são os seguintes:
  - *Scripts* podem ser executados a partir da janela de comando, de um outro *script* ou de uma função.
  - Funções podem ser chamadas a partir da janela de comando, de um *script* ou de uma outra função.
  - Originalmente, um arquivo de *script* não pode conter a declaração de uma função.

  - Um arquivo de função pode conter uma ou mais declarações de funções.
  - Quando um *script* ou uma função acionam um outro *script* ou uma outra função, o cógido acionador é interrompido até que o código acionado termine sua execução.

- Uma lista de regras gerais, para a construção de um arquivo de função, é a seguinte:

  - No caso da declaração de uma única função, o nome do arquivo deve ser o mesmo nome da função.
  - Quando forem declaradas várias funções no mesmo arquivo, ele deve ter o mesmo nome da função declarada no topo do arquivo. Tal função é denominada de função principal. As demais funções são chamadas de funções locais. Uma função local só é visível pela função principal do seu arquivo de declaração.
  - Caso o usuário não deseje declarar dados de entrada e de saída, a declaração da função será a seguinte:

    ```
    function nome_funcao
      codigo_funcao
    end
    ```

  - Caso o usuário deseje declarar apenas dados de entrada, a declaração da função será a seguinte:

    ```
    function nome_funcao(var_arg_1, ... , var_arg_M)
      codigo_funcao
    end
    ```

  - Caso o usuário deseje declarar apenas um dado de saída, a declaração da função será a seguinte:

    ```
    function var_ret = nome_funcao
      codigo_funcao
    end
    ```

  - Caso o usuário deseje declarar vários dados de saída, a declaração da função será a seguinte:

    ```
    function [ var_ret_1, ... , var_ret_N ] = nome_funcao
      codigo_funcao
    end
    ```

  - Portanto, a declaração mais genérica para uma função é a seguinte:

    ```
    function [ list_var_ret ] = nome_funcao(list_var_arg)
      codigo_funcao
    end
    ```

  - Para cada variável de retorno declarada, deve ser feita, pelo menos, uma atribuição de valor, durante a execução do código da função.

– Comentários, colocados entre a primeira linha da declaração e a primeira linha de código, serão identificados como instruções sobre a função e serão exibidos pelo comando *help nome_funcao*. Assim, a estrutura geral é a seguinte:

```
function [ list_var_ret ] = nome_funcao(list_var_arg)
% comentario_instrucao
% ...
% comentario_instrucao
  codigo_funcao
end
```

– Uma função retorna da sua chamada em duas situações, que são: quando ela acaba de executar o último comando do código ou quando ela encontra o comando *return*.

4.4.3 Tarefas

Teoria

1. Pesquise e escreva um pequeno texto sobre os seguintes conceitos, referentes a linguagens de programação:

   - Visibilidade de variáveis.
   - Escopo de variáveis.

   (1.0 pts)

2. Pesquise e escreva um pequeno texto sobre os conceitos de visibilidade e escopo de variáveis na plataforma Octave. (1.0 pts)

Prática

Atenda aos seguintes itens:

1. Execute cada um dos *scripts* 1 a 3, listados a seguir, separadamente. Empregando e retirando os comentários, isso deverá totalizar 7 execuções possíveis. Explique todos os resultados encontrados. (2.0 pts)

```
===== =====        ===== =====        ===== =====
script_1.m         script_2.m         script_3.m
===== =====        ===== =====        ===== =====
a = 3;             a =  7;            a = 13;
b = 5;             b = 11;            b = 19;
%script_2          %script_3
c = a * b          c = a - b          c = a + b
===== =====        ===== =====        ===== =====
```

2. Execute cada um dos *scripts* 4 a 6, listados a seguir, separadamente. Explique todos os resultados encontrados. (2.0 pts)

```
===== =====
script_4.m
===== =====

a = 3;
b = 5;

c = f5

d = a * b
===== =====
```

```
=====
f5.m
=====
function d = f5
  % Help de f5.

  a =  7;
  b = 11;

  d = a * b

end
=====
```

```
===== =====
script_5.m
===== =====

a = 3;
b = 5;

c = f6(a,b)

d = a * b
===== =====
```

```
=====
f6.m
=====
function d = f6(a,b)
  % Help de f6.

  a = a +  7;
  b = b + 11;

  d = a * b

end
=====
```

```
===== =====
script_6.m
===== =====

a = 3;
b = 5;

c = f7(a,b)

d = a * b
===== =====
```

```
=====
f7.m
=====
function d = f7(a,b)
  % Help de f7.

  a = a +  7;
  b = b + 11;
  d = a * b
  script_7
  d = a * b

end
=====
```

```
===== =====
script_7.m
===== =====

a = 13;
b = 19;

c = a + b

===== =====
```

3. Desenvolva 3 *scripts* e 1 função, de tal forma que:
    - O *script* 1 execute os *scripts* 2 e 3.
    - O *script* 2 execute o *script* 3.
    - O *script* 3 chame a função 4.
    - Durante a execução de cada um dos 4 arquivos executáveis, o código em questão deve ser sinalizar o seu início, o seu meio e o seu fim.

   (2.0 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

Parte III

Trabalhos TEC sobre aplicações básicas

Capítulo 5

Definição dos trabalhos TEC sobre gráficos básicos

5.1 TEC1: gráficos de curvas de funções unidimensionais

5.1.1 Definições

- Tempo de execução: 1 semana.
- Título: Gráficos de curvas de funções unidimensionais.
- Objetivo: Conhecer os elementos básicos usados na construção de gráficos de curvas de funções unidimensionais.

5.1.2 Especificações

- O Octave disponibiliza um conjunto de funções básicas que podem ser utilizadas na construção dos mais variados tipos de gráficos.
- Com a função figure(), pode-se criar uma figura ou, mais precisamente, uma *canvas* ou tela de desenho.
- É possível subdividir matricialmente uma *canvas* e definir uma posição para desenho, com a função subplot(). A posição pode ocupar uma única subdivisão matricial ou um conjunto delas.
- O gráfico da curva de uma função unidimensional é definido pela relação $y(x) \times x$, onde a variável independente x é denominada de abscissa e a variável dependente $y(x)$ é denominada de ordenada.
- Relações com abscissa discretizada $y[n] \times n$ são naturalmente representadas no Octave por meio de dois vetores, um para n e outro para $y[n]$. A função stem() constrói um gráfico discreto $y[n] \times n$.
- Relações analógicas $y(x) \times x$ são aproximadamente representadas no Octave por meio de dois vetores, um para valores discretos de x e outro para $y(x)$. A função plot() constrói um gráfico contínuo por partes, realizando uma interpolação linear entre cada dois pontos. Dada uma quantidade suficiente de pontos, a curva assume uma aparência contínua.

- A função plot() pode ser usada ainda para a marcação de pontos isolados no espaço, utilizando marcadores específicos.
- Com as funções xlabel(), ylabel() e title(), é possível identificar os eixos e o próprio gráfico.
- A função axis() possibilita descobrir os valores limites de abscissa e de ordenada que estejam sendo utilizados em um gráfico, bem como ajustá-los para valores desejados. Ela permite ainda ajustar o razão de aspecto (*aspect ratio*) na exibição do gráfico.
- Várias curvas podem ser superpostas em um mesmo gráfico, com o auxílio da função hold. É possível adicionar uma legenda no gráfico, por meio da função legend().
- A função grid controla a presença de um *grid* (reticulado) em um gráfico.
- Com a função plotyy() podem-se desenhar duas curvas com a mesma abscissa, mas com ordenadas diferentes, possibilitando o uso de diferentes escalas.
- Como o próprio nome indica, a função close é utilizada para fechar (suprimir, destruir) uma ou mais *canvas*.

5.1.3 Tarefas

Teoria

(4.0 pts) Pesquise e escreva um pequeno texto sobre a operação das seguintes funções:

1. figure(), subplot().
2. stem(), plot().
3. xlabel(), ylabel(), title().
4. axis(), hold, legend(), grid, close.

Prática

(4.0 pts) Atenda aos seguintes itens:

1. Para todos os gráficos especificados abaixo, cumpra as seguintes especificações comuns:
    - Reúna em um único *script* Octave todo o código proposto para gerar as figuras especificadas, sendo cada uma delas identificada por um número diferente.
    - Ative o *grid* em todos os gráficos.
    - Identifique cada abscissa, cada ordenada e cada gráfico.
    - Considerando X como abscissa e Y como ordenada, force, em todos os gráficos, o seguinte espaço de visualização: $X_{min} = -3$, $X_{max} = 3$, $Y_{min} = -10$, $Y_{max} = 10$.
2. Desenvolva um código Octave que gere uma figura contendo 1 gráfico discreto, com os seguintes pontos isolados:
    - Com marcadores “o”, em preto: (-2,0),(0,0),(1,-5) e (1,5).
    - Com marcadores “×”, em vermelho: (-1,-2.5),(-1,2.5),(0,0) e (2,0).

3. Desenvolva um código Octave que gere uma figura contendo 1 gráfico discreto, $y[n] \times n$, onde $y[n] = n^3$, para $-2 \leq n \leq 2$, com passo $\Delta n = 0.1$.

4. Desenvolva um código Octave que gere uma figura contendo 1 gráfico contínuo, $y(x) \times x$, onde $y(x) = x^3$, para $-2 \leq x \leq 2$, com passo $\Delta x = 0.5$.

5. Desenvolva um código Octave que gere uma figura contendo 1 gráfico contínuo, $y(x) \times x$, onde $y(x) = x^3$, para $-2 \leq x \leq 2$, com passo $\Delta x = 0.1$.

6. Desenvolva um código Octave que gere uma figura contendo 1 gráfico contínuo, $y(x) \times x$, com 3 curvas superpostas, de tal forma que:
   - As curvas são definidas por $y_1(x) = x$, $y_2(x) = x^2$ e $y_3(x) = x^3$.
   - As curvas $y_1(x)$, $y_2(x)$ e $y_3(x)$, devem ser desenhadas com as cores preto, vermelho e azul, respectivamente.
   - As curvas $y_1(x)$, $y_2(x)$ e $y_3(x)$, devem ser desenhadas com linha contínua, traços-e-pontos (-.-) e traços (- -), respectivamente.
   - Adicione uma legenda que identifique as curvas.

7. Desenvolva um código Octave que gere uma figura contendo 4 gráficos, organizados de forma matricial, de tal forma que:
   - Na primeira coluna da primeira linha, deverá ser gerado um gráfico, contendo a curva de $y_1(x)$, especificada acima.
   - Na segunda coluna da primeira linha, deverá ser gerado um gráfico, contendo a curva de $y_2(x)$, especificada acima.
   - Na terceira coluna da primeira linha, deverá ser gerado um gráfico, contendo a curva de $y_3(x)$, especificada acima.
   - Na segunda coluna da segunda linha, deverá ser gerado um gráfico, contendo as 3 curvas superpostas, conforme especificado acima.

8. Desenvolva um código Octave que gere uma figura contendo 4 gráficos, organizados de forma matricial, de tal forma que:
   - Na primeira linha da primeira coluna, deverá ser gerado um gráfico, contendo a curva de $y_1(x)$, especificada acima.
   - Na segunda linha da primeira coluna, deverá ser gerado um gráfico, contendo a curva de $y_2(x)$, especificada acima.
   - Na terceira linha da primeira coluna, deverá ser gerado um gráfico, contendo a curva de $y_3(x)$, especificada acima.
   - Ocupando as três linhas da segunda coluna, deverá ser gerado um gráfico, contendo as 3 curvas superpostas, conforme especificado acima.

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

5.2 TEC2: gráficos de funções unidimensionais complexas

5.2.1 Definições

- Tempo de execução: 1 semana.
- Título: Gráficos de funções unidimensionais complexas.
- Objetivo: Estudar e reproduzir os diversos gráficos (curvas bidimensionais e tridimensionais) associados a funções unidimensionais complexas, empregando, como exemplo, as funções exponenciais.

5.2.2 Especificações

Matemática

- Dados $A, b \in \mathbb{C}$, uma função exponencial pode ser definida por

$$x(t) = A \cdot b^t \ ,$$

para $-\infty < t < \infty$, onde $tin\mathbb{R}$.

- As constantes A e b podem ser reescritas como

$$A = |A| \cdot e^{j\angle A}$$

e

$$b = |b| \cdot e^{j\angle b} = e^{\sigma} \cdot e^{j\omega} = e^{\sigma + j\omega} = e^{s} \ ,$$

onde: $e \approx 2,718$ é o número de Euler, $|b| = e^{\sigma}$, $\angle b = \omega$ e $s = \sigma + j\omega$.

- Assim, a função exponencial torna-se

$$\begin{aligned} x(t) &= A \cdot b^t \\ &= \left(|A| \cdot e^{j\angle A}\right) \cdot (e^s)^t \ = \ \left(|A| \cdot e^{j\angle A}\right) \cdot e^{st} \\ &= \left(|A| \cdot e^{j\angle A}\right) \cdot e^{(\sigma + j\omega)t} \ = \ \left(|A| \cdot e^{j\angle A}\right) \cdot \left(e^{\sigma t} \cdot e^{j\omega t}\right) \\ &= \left(|A| \cdot e^{\sigma t}\right) \cdot e^{j(\omega t + \angle A)} \\ &= |x(t)| \cdot e^{j\angle x(t)} \ , \end{aligned} \tag{5.1}$$

onde $|x(t)| = (|A| \cdot e^{\sigma t})$ e $\angle x(t) = (\omega t + \angle A)$.

- Dois casos particulares, de grande interesse, são os seguintes:

  1. $x(t) = |A| \cdot e^{\sigma t}$, para $(\omega t + \angle A) = k \cdot 2\pi$, onde $k \in \mathbb{Z}$.
  2. $x(t) = e^{j(\omega t + \angle A)}$, para $(|A| \cdot e^{\sigma t}) = 1$.

- Dada a função $x(t) = |A| \cdot e^{-|\sigma| t}$, podem-se destacar alguns valores de interesse:
  1. Constante de tempo (*time constant*), dada por $\tau = 1/|\sigma|$ s, que é o tempo para a função chegar a $x(\tau) \approx 0,37\ |A|$, a partir de $t = 0$ s.
  2. Tempo de descida (*fall time*), dado por $T_{ft} \approx 2,2\tau$ s, que é o intervalo de tempo para a função descer de $x(0.11\tau) \approx 0,9\ |A|$ até $x(2.31\tau) \approx 0,1\ |A|$.
  3. Tempo de acomodação (*settling time*), dado por $T_{st} \approx 4\tau$ s, que é o tempo para a função chegar a $x(4\tau) \approx 0,02\ |A|$, a partir de $t = 0$ s.
- A Relação de Euler estabelece que

$$e^{\pm j\theta} = cos(\theta) \pm j\ sin(\theta)\ .$$

- Portanto, a função exponencial ainda pode ser escrita como

$$\begin{aligned} x(t) &= \left(|A| \cdot e^{\sigma t}\right) \cdot e^{j(\omega t + \angle A)} \\ &= \left(|A| \cdot e^{\sigma t}\right) \cdot [cos(\omega t + \angle A) + j\ sin(\omega t + \angle A)] \\ &= \left[\left(|A| \cdot e^{\sigma t}\right) \cdot cos(\omega t + \angle A)\right] + j\ \left[\left(|A| \cdot e^{\sigma t}\right) \cdot sin(\omega t + \angle A)\right] \\ &= Re\{x(t)\} + j\ Im\{x(t)\}\ , \end{aligned} \tag{5.2}$$

  onde $Re\{x(t)\} = (|A| \cdot e^{\sigma t}) \cdot cos(\omega t + \angle A)$ e $Im\{x(t)\} = (|A| \cdot e^{\sigma t}) \cdot sin(\omega t + \angle A)$.
- Uma função $f(t)$ é definida como periódica quando $f(t) = f(t \pm kT)$, para $k \in \mathbb{Z}$.
- As funções $e^{j\theta}$, $cos(\theta)$ e $sin(\theta)$, são periódicas, com período $\Theta_P = 2\pi$ rad.
- Definindo-se $\omega_P = 2\pi/T_P = 2\pi f_P$ rad/s, as funções $e^{j(\omega_P t)}$, $cos(\omega_P t)$ e $sin(\omega_P t)$, são periódicas, com período T_P s e frequência f_P Hz.

Geração de gráficos

- Embora a função $x(t)$ seja do tipo unidimensional, sendo dependente apenas da variável t, a avaliação da função em um ponto t_k retorna um valor complexo $x(t_k)$. Logo, não é possível gerar um gráfico bidimensional $x(t) \times t$.
- As soluções para a geração de gráficos associados à função complexa $x(t)$ são as seguintes:
  1. Gráfico com curva tridimensional: $Re\{x(t)\} \times Im\{x(t)\} \times t$.
  2. Gráficos projetivos com curvas bidimensionais: $Re\{x(t)\} \times t$, $Im\{x(t)\} \times t$ e $Re\{x(t)\} \times Im\{x(t)\}$.
  3. Gráficos com curvas bidimensionais: no caso particular de $x(t)$ real.

Octave

- A função plot() é capaz de gerar gráficos de curvas bidimensionais. Porém, para gerar gráficos de curvas tridimensionais, deve-se utilizar a função plot3().
- Assim como as funções xlabel() e ylabel() são usadas para identificar os eixos do gráfico de uma curva bidimensional, a função zlabel() também deve ser usada em gráficos de curvas tridimensionais.

- Para definir o ponto de vista no gráfico de uma curva tridimensional deve-se empregar a função view().
- Para controlar a relação de aspecto (*aspecto ratio*) de um gráfico, pode-se escolher uma das opções da função axis("option").

5.2.3 Tarefas

Teoria

Pesquise e escreva um pequeno texto sobre a operação das seguintes funções:
plot3(), zlabel(), view(), axis("square").
(1.0 pts)

Prática

Atenda aos seguintes itens:

1. Para todos os gráficos especificados abaixo, cumpra as seguintes especificações comuns:
   - Reúna em um único *script* Octave todo o código proposto para gerar as figuras especificadas, sendo cada uma delas identificada por um número diferente.
   - Ative o *grid* em todos os gráficos.
   - Identifique as abscissas, as ordenadas e os gráficos.
   - Todos os eixos da variável t devem assumir valores na faixa $0 \leq t \leq (NOP * Tp)$.
   - O código proposto deve obter do usuário todos os parâmetros empregados em todos os gráficos.
   - Para fins de testes e para a apresentação dos resultados, utilize os seguintes valores: $NOP = 4$ (número de períodos senoidais a serem visualizados), $|A| = 10$, $\angle A = 0$, $f_P = 1$ kHz, $T_{st} = [(NOP - 1) * Tp]$, $N_f = 56$ (número de pontos por período), azimute = 8 e elevação = 12.
2. Desenvolva um código Octave que gere uma figura contendo 6 gráficos, organizados de forma matricial, de tal forma que:
   - Os 6 gráficos deverão conter a curva $x(t) \times t$, onde $x(t) = M\ e^{\sigma t}$.
   - Na primeira linha: $M = |A|$,
   - Na segunda linha: $M = -|A|$,
   - Na primeira coluna: $\sigma < 0$ e ordenadas na faixa $[-2\ |A|\ ;\ 2\ |A|]$.
   - Na segunda coluna: $\sigma = 0$ e ordenadas na faixa $[-2\ |A|\ ;\ 2\ |A|]$.
   - Na terceira coluna: $\sigma > 0$ e ordenadas na faixa $[-max(|x(t)|)\ ;\ max(|x(t)|)]$, onde $max(x(t))$ é o valor máximo de x(t) na faixa de t considerada.

   (1.5 pts)

3. Desenvolva um código Octave que gere uma figura contendo 1 gráfico, com as seguintes curvas $x(t) \times t$:

   - Exponencial real (em azul): $x(t) = |A| \; e^{-|\sigma| t} = |A| \; e^{-\frac{t}{\tau}}$, onde $\tau = 1/|\sigma|$ s.
   - Reta 1 (em verde): $x(\tau)$.
   - Reta 2 (em preto): $x(0.11 \; \tau)$.
   - Reta 3 (em preto): $x(2.31 \; \tau)$.
   - Reta 4 (em vermelho): $x(4 \; \tau)$.
   - Usando a função stem(), marcar, na exponencial real, os pontos referentes aos quatro valores de t especificados nas 4 retas acima, com as respectivas cores.
   - Adicione uma legenda, referente às 4 retas acima.

   (1.5 pts)

4. Desenvolva um código Octave que gere uma figura contendo 3 gráficos, organizados de forma matricial, de tal forma que:

   - Na primeira linha, deverá ser gerado um gráfico, contendo a curva de $x_1(t) \times t$, onde $x_1(t) = |A| \; e^{-|\sigma| t}$, usando traços (- -) em vermelho, com ordenada na faixa $[0 \; ; \; |A|]$.
   - Na segunda linha, deverá ser gerado um gráfico, contendo a curva de $x_2(t) \times t$, onde $x_2(t) = cos(\omega_P t)$, usando linha contínua em azul, com ordenada na faixa $[-1 \; ; \; 1]$.
   - Na terceira linha, deverá ser gerado um gráfico, contendo a curva de $x(t) \times t$, onde $x(t) = x_1(t) \cdot x_2(t) = \left(|A| \; e^{-|\sigma| t}\right) \; cos(\omega_P t)$, usando linha contínua em preto, com ordenada na faixa $[-|A| \; ; \; |A|]$. De forma superposta, também devem ser desenhadas as curvas envoltórias de $x(t)$, usando traços (- -) em vermelho.

   (1.0 pts)

5. Desenvolva um código Octave que gere uma figura contendo 1 gráfico, com 1 curva tridimensional $Im\{x(t)\} \times Re\{x(t)\} \times t$, onde $x(t) = e^{j(w_P t + \Theta)}$ e $\Theta = \angle A$. Os limites de $Re\{x(t)\}$ e de $Im\{x(t)\}$ devem estar na faixa $[-1 \; ; \; 1]$.

   (1.5 pts)

6. Desenvolva um código Octave que gere uma figura contendo 3 gráficos, organizados de forma matricial, de tal forma que:

   - Considere a função $x(t) = e^{j(w_P t + \Theta)}$, onde $\Theta = \angle A$.
   - Na primeira linha da primeira coluna, deverá ser gerado um gráfico, contendo a curva projetiva $Im\{x(t)\} \times Re\{x(t)\}$. De forma superposta, também devem ser desenhados os vetores $e^{j(\theta_n n + \Theta)}$, onde $\theta_n = (2\pi/8)$ e $0 \leq n \leq 7$.
   - Na segunda linha da primeira coluna, deverá ser gerado um gráfico, contendo a curva projetiva $t \times Re\{x(t)\}$. A ordenada deverá crescer do topo para o fundo do gráfico. De forma superposta, também devem ser marcados os pontos referentes aos vetores desenhados no gráfico da primeira linha da primeira coluna.
   - Na primeira linha da segunda coluna, deverá ser gerado um gráfico, contendo a curva projetiva $Im\{x(t)\} \times t$. De forma superposta, também devem ser marcados os pontos referentes aos vetores desenhados no gráfico da primeira linha da primeira coluna.
   - Todos os 3 gráficos devem ter *aspect ratio* de 1 : 1.

   (1.5 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

5.3 TEC3: gráficos de superfícieis complexas

5.3.1 Definições

- Tempo de execução: 1 semana.
- Título: Gráficos de superfícieis complexas.
- Objetivo: Estudar e reproduzir os diversos gráficos associados a superfícies complexas.

5.3.2 Especificações

Matemática

- Na Teoria de Sistemas, que matematicamente fundamenta a Teoria de Circuitos e a Teoria de Controle, é comum que apareçam relações funcionais do tipo razão de polinômios.
- Considerando-se os sistemas analógicos, bem como as variáveis $\sigma, \omega \in \mathbb{R}$ e $s \in \mathbb{C}$, onde $s = \sigma + j\omega$, para $-\infty < (\sigma, \omega) < \infty$, alguns exemplos de tais funções polinomiais racionais são os seguintes:
$$H(j\omega) = \frac{N_H(j\omega)}{D_H(j\omega)} = \frac{b_0 + b_1(j\omega) + b_2(j\omega)^2 + \cdots + b_L(j\omega)^L}{a_0 + a_1(j\omega) + a_2(j\omega)^2 + \cdots + a_N(j\omega)^N} = \frac{\sum_{k=0}^{L} \; b_k \; (j\omega)^k}{\sum_{k=0}^{N} \; a_k \; (j\omega)^k}$$
e
$$H(s) = \frac{N_H(s)}{D_H(s)} = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_L s^L}{a_0 + a_1 s + a_2 s^2 + \cdots + a_N s^N} = \frac{\sum_{k=0}^{L} \; b_k \; s^k}{\sum_{k=0}^{N} \; a_k \; s^k} \; .$$
- Considerando-se os sistemas discretos/digitais, bem como as variáveis $r, \Omega \in \mathbb{R}$ e $z \in \mathbb{C}$, onde $z = re^{j\Omega}$, para $-\infty < (r, \Omega) < \infty$, alguns exemplos de tais funções polinomiais racionais são os seguintes:
$$H(e^{j\Omega}) = \frac{N_H(e^{j\Omega})}{D_H(e^{j\Omega})} = \frac{b_0 + b_1(e^{j\Omega}) + b_2(e^{j\Omega})^{-2} + \cdots + b_L(e^{j\Omega})^{-L}}{a_0 + a_1(e^{j\Omega}) + a_2(e^{j\Omega})^{-2} + \cdots + a_N(e^{j\Omega})^{-N}} = \frac{\sum_{k=0}^{L} \; b_k \; (e^{j\Omega})^{-k}}{\sum_{k=0}^{N} \; a_k \; (e^{j\Omega})^{-k}}$$
e
$$H(z) = \frac{N_H(z)}{D_H(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_L z^{-L}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} = \frac{\sum_{k=0}^{L} \; b_k \; z^{-k}}{\sum_{k=0}^{N} \; a_k \; z^{-k}} \; .$$
- Por serem funções polinomiais, as funções $H(s)$ e $H(z)$ podem ser fatoradas, o que fornece
$$H(s) = \frac{N_H(s)}{D_H(s)} = \left(\frac{b_L}{a_N}\right) \frac{(s - z_1)(s - z_2)\cdots(s - z_L)}{(s - p_1)(s - p_2)\cdots(s - p_N)} = K_s \; \frac{\prod_{k=1}^{L} \; (s - z_k)}{\prod_{k=1}^{N} \; (s - p_k)}$$
e
$$H(z) = \frac{N_H(z)}{D_H(z)} = \left(\frac{b_0}{a_0}\right) \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})\cdots(1 - z_L z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})\cdots(1 - p_N z^{-1})} = K_z \; \frac{\prod_{k=1}^{L} \; (1 - z_k z^{-1})}{\prod_{k=1}^{N} \; (1 - p_k z^{-1})} \; ,$$
onde $K_s = (b_L/a_N)$ e $K_z = (b_0/a_0)$.

- Os valores K_s e K_z são denominados de constantes de ganho. Os valores z_k são as raízes ou os zeros dos polinômios numeradores, sendo, portanto, os zeros da função polinomial racional. Os valores p_k são as raízes ou os zeros dos polinômios denominadores, sendo, portanto, os pólos da função polinomial racional. De forma geral, z_k e p_k são chamados de singularidades da função polinomial racional.

- Nas suas formas gerais, todas essas funções envolvem valores complexos. Avaliadas para os valores reais ω_m e Ω_m, as funções $H(j\omega)$ e $H(e^{j\Omega})$ retornam valores complexos. Por sua vez, avaliadas para os valores complexos s_m e z_m, as funções $H(s)$ e $H(z)$ retornam valores complexos.

- Assim, por se tratarem de funções complexas, dependentes de variáveis reais ou complexas, elas podem ser reescritas na forma polar, gerando

$$H(j\omega) = |H(j\omega)| \ e^{j\angle H(j\omega)} \ ,$$

$$H(s) = |H(s)| \ e^{j\angle H(s)} \ ,$$

$$H(e^{j\Omega}) = |H(e^{j\Omega})| \ e^{j\angle H(e^{j\Omega})}$$

e

$$H(z) = |H(z)| \ e^{j\angle H(z)} \ .$$

- A Região de Convergência (ROC) da função $H(s)$ é formada pelos valores de s para os quais a função é definida. Quando as funções $H(s)$ e $H(j\omega)$ são definidas e a ROC da função $H(s)$ engloba o eixo $s = j\omega$, pode-se garantir a seguinte relação:

$$H(s)|_{s=j\omega} = H(j\omega) \ .$$

- Da mesma forma, a ROC da função $H(z)$ é formada pelos valores de z para os quais a função é definida. Quando as funções $H(z)$ e $H(e^{j\Omega})$ são definidas e a ROC da função $H(z)$ engloba o círculo de raio unitário $z = e^{j\Omega}$, pode-se garantir a seguinte relação:

$$H(z)|_{z=e^{j\Omega}} = H(e^{j\Omega}) \ .$$

Geração de gráficos

- De modo geral, os valores das singularidades z_k e de p_k são complexos. Logo, eles podem ser marcados em gráficos bidimensionais $Im\{s\} \times Re\{s\}$ e $Im\{z\} \times Re\{z\}$.

- As funções $|H(j\omega)|$, $\angle H(j\omega)$, $|H(e^{j\Omega})|$ e $\angle H(e^{j\Omega})$, são do tipo real unidimensional. Logo, os gráficos $|H(j\omega)|\times\omega$, $\angle H(j\omega)\times\omega$, $|H(e^{j\Omega})|\times\Omega$ e $\angle H(e^{j\Omega})\times\Omega$, são curvas bidimensionais.

- Dado que s e z são variáveis complexas, $|H(s)|$, $\angle H(s)$, $|H(z)|$ e $\angle H(s)$, são funções reais bidimensionais. Logo, os gráficos $|H(s)| \times Im\{s\} \times Re\{s\}$, $\angle H(s) \times Im\{s\} \times Re\{s\}$, $|H(z)| \times Im\{z\} \times Re\{z\}$ e $\angle H(z) \times Im\{z\} \times Re\{z\}$, são superfícies tridimensionais.

- A função $e^{j\Omega}$ é periódica, com período $\Omega_P = 2\pi$. Portanto, os gráficos $|H(e^{j\Omega})| \times \Omega$ e $\angle H(e^{j\Omega}) \times \Omega$, são curvas periódicas, com o mesmo período.

Octave

- O Octave disponibiliza uma série de funções que são capazes de gerar gráficos de superfícies tridimensionais.

- De certa forma, as funções disponíveis são variações das funções básicas mesh() e surf().

- Os gráficos gerados são aproximações discretas de uma superfície, onde cada par de pontos consecutivos é ligado por um segmento de reta, gerando um reticulado que aproxima a superfície. Uma determinada variação de cores é utilizada para gerar a sensação de tridimensionalidade.

- Ao invés de receber os convencionais vetores $\boldsymbol{x}$ e $\boldsymbol{y}$, associados aos eixos dos gráficos, as funções de superfície esperam receber as matrizes $\boldsymbol{X}$ e $\boldsymbol{Y}$, definidas a seguir.

- Supondo a região delimitada pelos valores discretos $\boldsymbol{x} = [1, 2, 3, 4, 5]$ e $\boldsymbol{y} = [6, 7, 8]$, o conjunto de pontos (x_k, y_k) dessa região discretizada é definido por

6	(1,6)	(2,6)	(3,6)	(4,6)	(5,6)
7	(1,7)	(2,7)	(3,7)	(4,7)	(5,7)
8	(1,8)	(2,8)	(3,8)	(4,8)	(5,8)
	1	2	3	4	5

  de onde podem-se definir as seguintes matrizes

  $$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

  e

  $$\boldsymbol{Y} = \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 \\ 8 & 8 & 8 & 8 & 8 \end{bmatrix} .$$

- Dada a relação funcional $z = f(x, y)$, as matrizes $\boldsymbol{X}$ e $\boldsymbol{Y}$ também são necessárias para o cálculo da matriz $\boldsymbol{Z}$.

- O Octave disponibiliza a função meshgrid() com o objetivo de facilitar a montagem das matrizes $\boldsymbol{X}$ e $\boldsymbol{Y}$, a partir dos vetores $\boldsymbol{x}$ e $\boldsymbol{y}$.

5.3.3 Tarefas

Teoria

Pesquise e escreva um pequeno texto sobre a operação das seguintes funções:
num2str(), text(), axis("equal"), meshgrid(), mesh(), surf().
(1.5 pts)

Prática

Atenda aos seguintes itens:

1. Para todos os gráficos especificados abaixo, cumpra as seguintes especificações comuns:

   - Reúna em um único *script* Octave todo o código proposto para gerar as figuras especificadas, sendo cada uma delas identificada por um número diferente.
   - Ative o *grid* em todos os gráficos.
   - Identifique as abscissas, as ordenadas e os gráficos.
   - Utilize os seguintes coeficientes:

   $$\begin{aligned} \boldsymbol{b} &= [b_4, b_3, b_2, b_1, b_0] \\ &= [7.51 \times 10^{-4}, 0.0, 7.56 \times 10^{-5}, 0.0, 1.60 \times 10^{-6}] \end{aligned}$$

   e

   $$\begin{aligned} \boldsymbol{a} &= [a_5, a_4, a_3, a_2, a_1, a_0] \\ &= [1.0, 9.31 \times 10^{-2}, 1.75 \times 10^{-2}, 1.04 \times 10^{-3}, 6.64 \times 10^{-5}, 1.60 \times 10^{-6}] \ . \end{aligned}$$

   - Utilize os seguintes valores:
     - $Re\{s\} = \sigma$: $\Delta\sigma = 5.0 \times 10^{-3}$ e $-0.2 \le \sigma \le 0.2$.
     - $Im\{s\} = \omega$: $\Delta\omega = 2.5 \times 10^{-3}$ e $-0.3 \le \omega \le 0.3$.

2. Desenvolva um código Octave que gere uma figura contendo 1 gráfico, atendendo às seguintes especificações:

   - Calcule a constante de ganho K_s, os zeros z_k e os pólos p_k de $H(s)$, todos especificados acima.
   - Posicione os zeros no gráfico, com marcador "o", com dimensão 12, em vermelho.
   - Posicione os pólos no gráfico, com marcador "x", com dimensão 12, em preto.
   - Posicione o texto "$K_s = \pm N.NNe - NNN$" na posição (0.1,0.0) do gráfico.
   - Ajuste os eixos para terem as mesmas unidades de exibição.

   (1.5 pts)

3. Desenvolva um código Octave que gere uma figura contendo 4 gráficos, organizados de forma matricial, de tal forma que:

   - Na primeira coluna da primeira linha, deverá ser gerado um gráfico, contendo a curva de $|H(j\omega)| \times \omega$, especificada acima.
   - Na segunda coluna da primeira linha, deverá ser gerado um gráfico, contendo a curva de $|H(j\omega)|_{dB} \times \omega$, onde $|H(j\omega)|_{dB} = 20 \cdot log_{10}(|H(j\omega)|) \ dB$.
   - Em ambas as colunas da da segunda linha, deverá ser gerado um gráfico, contendo a curva de $\angle H(j\omega) \times \omega$, especificada acima, em graus, com variação contínua de valores (*unwrapped*).
   - Ajuste a visualização das abscissas para os limites especificados.

   (2.0 pts)

4. Desenvolva um código Octave que gere uma figura contendo 1 gráfico, atendendo às seguintes especificações:

    - Empregando a função mesh(), deverá ser gerado um gráfico, contendo a superfície de $|H(s)| \times Re\{s\} \times Im\{s\}$, especificada acima.
    - Superposta à superfície, deverá ser desenhada a curva $|H(j\omega)| \times \omega$, em vermelho.
    - Ajuste a visualização de $Re\{s\}$ e de $Im\{s\}$ para os limites especificados.
    - Ajuste a visualização de $|H(s)|$ para a faixa $0 \leq |H(s)| \leq 4$.
    - Ajuste o ponto de vista inicial para azimute = 70 e elevação = 35.

    (2.0 pts)

5. Desenvolva um código Octave que gere uma figura contendo 1 gráfico, atendendo às seguintes especificações:

    - Repita a geração da figura anterior.
    - Ajuste a visualização de $Re\{s\}$ para a faixa $-0.15 \leq Re\{s\} \leq 0.10$.
    - Ajuste a visualização de $Im\{s\}$ para a faixa $-0.30 \leq Im\{s\} \leq 0.30$.
    - Ajuste a visualização de $|H(s)|$ para a faixa $0 \leq |H(s)| \leq 1.1$.
    - Ajuste o ponto de vista inicial para azimute = 50 e elevação = 35.

    (1.0 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

Capítulo 6

Definição dos trabalhos TEC sobre geração de sons

6.1 TEC1: geração de tons, atenuação e efeito Doppler

- Tempo de execução: 1 semana.
- Título: Representação, manipulação e reprodução de sons com o auxílio de matrizes.
- Objetivo: Trabalhar noções básicas sobre a representação de sons por meio de matrizes, a realização de manipulações matriciais básicas e a reprodução de sons armazenados.

6.1.1 Especificações

- Um sinal é definido como uma grandeza matemática ou física que carrega uma informação.
- Ao se provocar ondas de pressão no ar, de tal forma que os valores da pressão sigam o perfil de um sinal matemático senoidal do tipo $x(t) = A\ cos(\omega t)$, onde $\omega = 2\pi f$, definido por uma intensidade A e uma determinada freqüência f, na faixa $20\ Hz < f < 20\ kHz$, gera-se um sinal que é identificado pelo sistema auditivo humano como um sinal sonoro, musicalmente associado a uma nota ou a um tom.
- A distância entre a fonte e o receptor do sinal sonoro produz uma atenuação na amplitude do sinal sonoro senoidal gerado (efeito de dissipação ou perda de energia).
- Quando a fonte e o receptor do sinal sonoro apresentam uma velocidade relativa entre si, isso provoca a sensação de variação na freqüência do sinal sonoro senoidal (efeito Doppler).
- O sinal sonoro de um carro de bombeiro pode ser modelado por meio da alternância entre dois sinais sonoros senoidais que possuem duas freqüências distintas (f_1 e f_2).
- A partir de amostras de sinais sonoros, que são armazenadas em matrizes, o Octave viabiliza a execução de sons, por meio de funções específicas, tais como: *sound()* e *soundsc()*. Estas funções enviam dados para o sistema de som do computador, para que o sinal sonoro seja gerado.

6.1.2 Tarefas

Teoria

1. Em relação ao modelo definido para um sinal sonoro de um carro de bombeiro real, pesquise e apresente os resultados encontrados, para cada um dos seguintes itens:
   - O padrão de freqüências utilizadas em cada um dos dois sinais senoidais.
   - O padrão de amplitudes relativas utilizadas em cada um dos dois sinais senoidais.
   - O tempo de execução de cada um dos dois sinais senoidais.

   (1.0 pts)
2. Desenvolva um modelo de atenuação de amplitude em função da distância entre a fonte e o receptor do sinal sonoro. (0.5 pts)
3. Apresente e explique as equações envolvidas no efeito Doppler aplicado a sons. (0.5 pts)

Prática

1. Para a geração dos sinais sonoros pelo Octave, utilize uma discretização temporal com $\Delta t = T_s = 1/F_s$ s, onde $F_s = 44.100$ Hz.
2. Desenvolva um código Octave que gere um sinal sonoro equivalente ao sinal de um carro de bombeiro que é escutado dentro do próprio carro. (2.0 pts)
3. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere um sinal sonoro equivalente ao sinal de um carro de bombeiro que é escutado por um pedestre parado na calçada, enquanto o carro de bombeiro se aproxima dele, passa por ele e se afasta dele, em um movimento contínuo. Nesse caso, devem ser levados em consideração os efeitos de atenuação e de efeito Doppler, de acordo com a velocidade do carro, fornecida pelo usuário.

   (4.0 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

Capítulo 7

Definição dos trabalhos TEC sobre imagens

7.1 TEC1: representação matricial de imagens

- Tempo de execução: 1 semana.
- Título: Representação e exibição de imagens com o auxílio de matrizes.
- Objetivo: Trabalhar noções básicas sobre a representação de imagens por meio de matrizes e a exibição de imagens armazenadas.

7.1.1 Especificações

- Representação gráfica discreta de função bidimensional:
  - Uma imagem discreta pode ser interpretada como um gráfico discreto de função bidimensional.
  - Um gráfico discreto de função bidimensional $z(x, y) = f(x, y)$ pode ser visto como uma sucessão de pontos $p = \{p_k\}$, formados por triplas $p_k = (x_k, y_k, z_k)$.
  - Podem ser destacadas duas diferenças básicas entre o gráfico de uma superfície e o gráfico de uma imagem:
    * Na superfície, o valor de z_k representa a distância do ponto p_k ao plano (x, y), gerando objetos tridimensionais. Por sua vez, na imagem, o valor de z_k representa a intensidade de cor do ponto p_k na posição (x_k, y_k), gerando objetos bidimensionais.
    * Na superfície, cada ponto é representado de forma adimensional. Para se obter uma aproximação de uma superfície contínua por uma superfície discreta, deve-se realizar alguma forma de interpolação entre seus pontos. Por sua vez, na imagem, cada ponto já é naturalmente associado a uma área finita bidimensional, preenchida com alguma cor, em torno da posição (x_k, y_k) do ponto p_k.
  - Cada ponto p_k de uma imagem representa o seu elemento básico constituinte e é denominado de pixel (*picture element*).
  - A quantidade de *bits* usados para armazenar informação sobre cada pixel (*bits per pixel*) é denominada de *bit depth*.

- Assim, o processo de se elaborar a imagem pode ser definido como:
  * Montar os vetores $x = \{x_k\}$, $y = \{y_k\}$ e $z = \{z_k\}$.
  * Montar triplas $p_k = (x_k, y_k, z_k)$.
  * Definir um padrão representativo para um pixel, representado por um ponto em um espaço bidimensional. Normalmente, costuma-se preencher uma área em torno de cada ponto desenhado.
  * Desenhar os pontos p_k (pixels) no espaço bidimensional.

- Representações de imagens no Octave:
  - Formatos padrões de armazenamento de imagens suportados:
    * BMP (*Microsoft Windows Bitmap*)
    * HDF (*Hierarchical Data Format*)
    * JPEG (*Joint Photographic Experts Group*)
    * PCX (*Paintbrush*)
    * PNG (*Portable Network Graphics*)
    * TIFF (*Tagged Image File Format*)
    * XWD (*X Window Dump*)
  - Tipo de dados usados para representar imagens: uint8, uint16, double.

- Estruturas de dados para imagens no Octave:
  - São utilizadas três estruturas de dados: imagem indexada, imagem de intensidade e imagem RGB.
  - Imagem indexada:
    * Utiliza duas matrizes por imagem:
      matrix de pixels ($L \times C$) e matriz de mapa de cores ($L_{map} \times 3$).
    * Matriz de pixels ($L \times C$)
      · Contém valores inteiros, que servem de índices para acesso ao mapa de cores.
      · Para dados uint8/16, a faixa de índices é [0;L_{map}-1].
      · Para dados double, a faixa de índices é [1;L_{map}].
      · O aplicativo não realiza operações matemáticas sobre dados uint8 e uint16. Para isso, podem-se empregar as seguintes conversões:

```
uint8  --> double:  M64 = double(M8)  + 1
uint16 --> double:  M64 = double(M16) + 1
double --> uint8 :  M8  = uint8 ( round(M64 - 1) )
double --> uint16:  M16 = uint16( round(M64 - 1) )
```

    * Matriz de mapa de cores ($L_{map} \times 3$)
      · Contém valores double, na faixa [0;1].
      · As colunas 1 a 3 representam, respectivamente, as intensidades das cores vermelho (*red* ou R), verde (*green* ou G) e azul (*blue* ou B).
      · A linha (0,0,0) representa a menor intensidade possível, o que significa a "cor" preto.
      · A linha (1,1,1) representa a maior intensidade possível, o que significa a "cor" branco.

  - Imagem de intensidade (ou escala de tons de cinza ou *grayscale*)
    * Utiliza uma matriz por imagem: matriz de pixels.
    * Matriz de pixels ($L \times C$): Contém valores de intensidade, que se encontram dentro de alguma faixa.
    * O aplicativo manipula a imagem de intensidade como se fosse uma imagem indexada.
    * A matriz de intensidade é utilizada como se fosse uma matriz de índices, associada a um mapa de cores especificado.
    * Normalmente, é utilizado um mapa com escala de tons de cinza (*grayscale*).
    * Os valores de intensidade são linearmente escalados para gerar os índices de acesso ao mapa de cores.
    * Os valores de intensidade podem ser uint8, uint16 ou double.
    * Valores típicos de intensidade:

```
Para dados uint8 , a faixa é [0;255].
Para dados uint16, a faixa de índices é [0;65535].
Para dados double, a faixa de índices é [0;1].
```

  - Imagem RGB ou *Truecolor*
    * Utiliza uma matriz por imagem: matriz de pixels.
    * Matriz de pixels ($L \times C \times 3$): Contém valores de intensidade, para cada uma das cores básicas (RGB).
    * Os valores de intensidade podem ser uint8, uint16 ou double.
    * Valores típicos de intensidade:

```
Para dados uint8 , a faixa é [0;255].
Para dados uint16, a faixa de índices é [0;65535].
Para dados double, a faixa de índices é [0;1].
```

- Funções comumente utilizadas:
  - Funções relacionadas com gráficos:
    figure(), subplot(), axis(), hold on/off,
    title(), xlabel(), ylabel(), zlabel(),
    linspace(), logspace().
  - Funções relacionadas com imagens:
    image(), imagesc(), colormap(),
    imread(), imwrite(), iminfo(),
    gray2ind(), ind2gray(), rgb2ind(), ind2rgb().
  - Identificações dos mapas de cores:
    * 1 cor: white.
    * 16 cores: vga.
    * 64 cores: hsv, hot, gray, bone, copper, pink, flag, lines,
      colorcube, jet, prism, cool, autumn, spring, winter, summer.

7.1.2 Tarefas

Teoria

1. Pesquise e responda as seguintes perguntas:

   - No *colormap gray*, qual o valor numérico para desenhar um pixel branco?
   - No *colormap gray*, qual o valor numérico para desenhar um pixel preto?

   (0.5 pts)

2. Pesquise e escreva uma breve descrição sobre cada uma das seguintes funções: colormap(), image(), axis(vector), axis("option"), disp(), pause().

   (0.5 pts)

Prática

1. Desenvolva um código Octave que gere uma figura contendo a imagem bidimensional de 1 pixel preto em um fundo branco, atendendo às seguintes especificações:

   - Ajuste a área de visualização da imagem para 1 unidade anterior e 1 unidade posterior, em cada dimensão, em relação à posição do pixel.
   - Pause a visualização.
   - Após a pausa, ajuste a imagem original para que o pixel tenha a aparência correta de um quadrado.

   (1.0 pts)

2. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere uma figura contendo a imagem bidimensional de 4 pixels pretos em um fundo branco, atendendo às seguintes especificações:

   - Os pixels deverão ser organizados matricialmente, em um arranjo do tipo 2×2.
   - Para cada dimensão, deve haver um espaço de 1 pixel entre cada 2 pixels pretos adjacentes.
   - Ajuste a área de visualização da imagem para 1 unidade anterior e 1 unidade posterior, em cada dimensão, em relação à posição dos pixels mais externos.
   - Pause a visualização.
   - Após a pausa, ajuste a imagem original para que cada pixel tenha a aparência correta de um quadrado.

   (1.5 pts)

3. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere uma figura contendo a imagem bidimensional de 9 pixels pretos em um fundo branco, atendendo às seguintes especificações:

   - Os pixels deverão ser organizados matricialmente, em um arranjo do tipo 3×3.
   - Para cada dimensão, deve haver um espaço de 1 pixel entre cada 2 pixels pretos adjacentes.

- Ajuste a área de visualização da imagem para 1 unidade anterior e 1 unidade posterior, em cada dimensão, em relação à posição dos pixels mais externos.
- Pause a visualização.
- Após a pausa, ajuste a imagem original para que cada pixel tenha a aparência correta de um quadrado.

(1.5 pts)

4. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere uma figura contendo a imagem bidimensional de 25 pixels pretos em um fundo branco, atendendo às seguintes especificações:
   - Os pixels deverão ser organizados matricialmente, em um arranjo do tipo 5×5.
   - Para cada dimensão, não deve haver espaço entre cada 2 pixels pretos adjacentes.
   - No canto esquerdo superior da imagem, deverá ser superposto um sinal "+" branco, composto por 3 pixels em cada dimensão.
   - Ajuste a área de visualização da imagem para 1 unidade anterior e 1 unidade posterior, em cada dimensão, em relação à posição dos pixels mais externos.
   - Pause a visualização.
   - Após a pausa, ajuste a imagem original para que a imagem tenha a aparência correta de um quadrado.

   (1.5 pts)

5. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere uma figura contendo a imagem bidimensional de uma moldura quadrada preta, em um fundo branco, atendendo às seguintes especificações:
   - A moldura deverá ser composta por 15 pixels em cada dimensão.
   - Na parte interna à moldura, deverão ser desenhados os sinais "+" e "-".
   - Os sinais "+" e "-" deverão ser desenhados nos cantos esquerdo superior e direito inferior, respectivamente.
   - O sinal "+" deverá ser composto por 3 pixels em cada dimensão. Ele deverá ser distanciado de 1 pixel em relação à moldura, em cada dimensão.
   - O sinal "-" deverá ser composto por 3 pixels. Ele deverá ser distanciado de 1 pixel em relação à borda vertical e de 2 pixels em relação à borda horizontal.
   - Ajuste a área de visualização da imagem para 1 unidade anterior e 1 unidade posterior, em cada dimensão, em relação à moldura.
   - Pause a visualização.
   - Após a pausa, ajuste a imagem original para que a imagem tenha a aparência correta de um quadrado.

   (1.5 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

7.2 TEC2: manipulação matricial de imagens

- Tempo de execução: 1 semana.
- Título: Representação, manipulação e exibição de imagens com o auxílio de matrizes.
- Objetivo: Trabalhar noções básicas sobre a representação de imagens por meio de matrizes, a realização de manipulações matriciais básicas e a exibição de imagens armazenadas.

7.2.1 Especificações

- Imagens podem ser interpretadas como representações gráficas de funções bidimensionais. Assim, cada par de coordenadas cartesianas (x, y) é mapeado em um valor $z = f(x, y)$, que carrega uma informação sobre a intensidade da imagem na posição (x, y). Nessa interpretação, a imagem representa o conjunto $p = \{p_k\}$ de todos os pontos formados por triplas $p_k = (x_k, y_k, z_k)$. Cada ponto p_k é denominado de *picture element* ou pixel. Por definição, uma imagem analógica contém um número infinito de pixels.
- A intensidade da imagem em cada pixel pode ser representada na forma de tons de cinza ou de combinações de cores.
- Uma imagem discreta pode ser interpretada como um gráfico discreto de função bidimensional. Os valores z_k dos pontos p_k podem ser armazenados em matrizes, onde as linhas e as colunas da matriz são associadas ao valores x_k e y_k, respectivamente. Com esse tipo de representação, as imagens podem ser matricialmente manipuladas.
- Na amostragem de imagens, quanto menor for o intervalo de amostragem (resolução) utilizado na geração da imagem discreta, maior será o número de pontos p_k (pixels) obtidos.
- Na exibição das imagens armazenadas em matrizes, considera-se que o ponto p_k (pixel) ocupa uma determinada área em torno da posição (x_k, y_k), a qual é preenchida com a intensidade definida pelo valor z_k da matriz. Quanto menor for a área ocupada por cada pixel, maior será a densidade superficial de pontos. Uma unidade comumente utilizada para expressar a densidade superficial de pontos é "Pontos Por Polegada" ou *Dots Per Inch* ou DPI.
- Levando-se em consideração a resolução usada na amostragem de uma imagem e a densidade superficial de pontos usada na sua reprodução, pode-se dizer que, independentemente do tamanho original da imagem, quanto menor for o valor da resolução e quanto maior for a densidade superficial, maior será a qualidade visual da reprodução da imagem.
- A quantidade de amostras de uma seqüência pode ser alterada por meio de duas operações básicas, conhecidas por: *downsampling* (redução sistemática do número de amostras) e *upsampling* (aumento sistemático do número de amostras). Como os próprios nomes sugerem, elas podem ser encaradas como uma variação da taxa de amostragem originalmente utilizada. No *downsampling*, amostras originais são descartadas. No *upsampling*, novas amostras devem ser geradas, por nova amostragem ou por interpolação entre as amostras existentes.

- Na tentativa de se ocupar menos espaço no armazenamento, pode-se aplicar a operação de *downsampling*. Por exemplo:
  - Aplicando-se um *downsampling* de 2 em x, elimina-se uma coluna entre cada duas antigas colunas.
  - Aplicando-se um *downsampling* de 2 em y, elimina-se uma linha entre cada duas antigas linhas.
- Por outro lado, na tentativa de se melhorar a resolução de uma imagem armazenada, diminuindo-se o seu valor, pode-se aplicar a operação de *upsampling*, seguida de algum método de interpolação. Por exemplo:
  - Aplicando-se um *upsampling* de 2 em x, abre-se uma nova coluna entre cada duas antigas colunas e novos valores devem ser calculados.
  - Aplicando-se um *upsampling* de 2 em y, abre-se uma nova linha entre cada duas antigas linhas e novos valores devem ser calculados.

  Um método de interpolação simples de se implementar é a interpolação linear, que, para um *upsampling* de 2, resume-se ao cálculo de uma média aritmética de dois valores.

7.2.2 Tarefas

Teoria

1. Discuta as seguintes afirmações:
   - Levando-se em consideração a resolução usada na amostragem de uma imagem e a densidade superficial de pontos usada na sua reprodução, pode-se dizer que, independentemente do tamanho original da imagem, quanto menor for o valor da resolução e quanto maior for a densidade superficial, maior será a qualidade visual da reprodução da imagem.
   - Diminuir o valor da resolução usada na amostragem de uma imagem e manter a densidade superficial de pontos usada na sua reprodução, produz um aumento no tamanho da imagem.
   - Manter o valor da resolução usada na amostragem de uma imagem e aumentar a densidade superficial de pontos usada na sua reprodução, produz uma diminuição no tamanho da imagem.

   (1.5 pts)

Prática

1. Desenvolva um código Octave que, recebendo uma matriz M_{org}, com dimensões $L_{org} \times C_{org}$, contendo os índices de cores de uma imagem, e um mapa de cores, realize um *upsampling* de 2, tanto em x quanto em y, na matriz M_{org}, gerando uma nova matriz M_{int}, e calcule o valor de cada novo ponto usando a média aritmética entre os dois pontos existentes imediatamente a cada lado do ponto em questão. (3.0 pts)

2. Desenvolva um código Octave extra, a ser anexado ao código anterior, que gere uma figura contendo 7 imagens, organizadas de forma matricial.

   Atenção: Todas as imagens deverão estar alinhadas com a origem $(x = 0, y = 0)$.

   Na primeira linha, deverão ser geradas três imagens, referentes à matriz M_{org}.

   - A primeira imagem deve ser gerada pela matriz M_{org}, em tamanho original, em uma área igual ao dobro do tamanho da imagem original.
   - A segunda imagem deve ser gerada pela matriz M_{org}, com o dobro do tamanho, em uma área igual ao dobro do tamanho da imagem original.
   - A terceira imagem deve ser gerada pela matriz M_{org}, realizando a superposição das duas imagens anteriores, em uma área igual ao dobro do tamanho da imagem original.

   Na segunda linha, deverão ser geradas três imagens, referentes à matriz M_{int}.

   - A primeira imagem deve ser gerada pela matriz M_{int}, em tamanho original, em uma área igual ao dobro do tamanho da imagem original.
   - A segunda imagem deve ser gerada pela matriz M_{int}, com a metade do tamanho, em uma área igual ao dobro do tamanho da imagem original.
   - A terceira imagem deve ser gerada pela matriz M_{int}, realizando a superposição das duas imagens anteriores, em uma área igual ao dobro do tamanho da imagem original.

   Na terceira linha, deverá ser gerada uma única imagem, referente às matrizes M_{org} e M_{int}, realizando a superposição das duas imagens, em tamanho original, em uma área igual ao dobro do tamanho da imagem original.

   (3.5 pts)

Relatório

- Escreva um relatório, baseado no modelo definido para a disciplina. Comente, quando necessário, trechos de código ao longo do texto. Coloque as listagens completas dos códigos desenvolvidos na seção Anexos, organizadas em subseções. (2.0 pts)

Referências Bibliográficas

[Ant86] A. Antoniou. *Digital Filters: Analysis and Design.* Tata McGraw-Hill, New Delhi, India, 2nd reprint edition, 1986.

[Cad73] J. A. Cadzow. *Discrete-Time Systems: An Introduction with Interdisciplinary Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[DdSN10] P. S. R. Diniz, E. A. B. da Silva, and S. Lima Netto. *Digital Signal Processing: System Analysis and Design.* Cambridge University Press, Cambridge, UK, 2nd edition, 2010.

[Jac96] L. B. Jackson. *Digital Filters and Signal Processing - with MATLAB exercises.* Kluwer Academic Publishers, 3rd edition, 1996.

[KD04] H. Kopka and P. W. Daly. *A Guide to LaTeX and Electronic Publishing.* Addison-Wesley, Harlow, England, 4th edition, 2004.

[MG04] F. Mittelbach and M. Goosens. *The LaTeX Companion.* Addison-Wesley, Boston, MA, USA, 2th edition, 2004.

[Mit98] S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill, New York, NY, 1998.

[OS75] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[OWY83] A. V. Oppenheim, A. S. Willsky, and I. T. Young. *Signals and Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1983.

[PL76] A. Peled and B. Liu. *Digital Signal Processing: Theory, Design and Implementation.* John Wiley, New York, NY, 1976.

[PM06] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications.* Prentice Hall, 4th edition, 2006.

[Rob09] M. J. Roberts. *Fundamentos em Sinais e Sistemas.* McGraw-Hill, São Paulo, SP, 2009.

[SDD84] W. D. Stanley, G. R. Dougherty, and R. Dougherty. *Signals and Systems.* Prentice-Hall, Reston, Virginia, 2nd edition, 1984.

[She95] K. Shenoi. *Digital Signal Processing in Telecommunications.* Prentice-Hall PTR, Upper Saddle River, NJ, 1995.

[SK89] R. D. Strum and D. E. Kirk. *First Principles of Discrete Systems and Digital Signal Processing.* Addison-Wesley, Massachusetts, 1989.