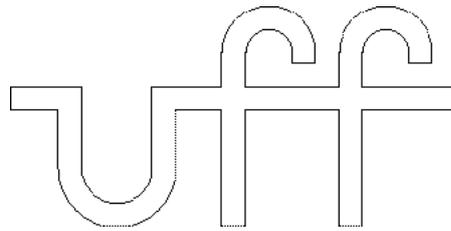


**Tutorial
sobre
Sistema de Média Móvel
para
Fundamentos de
Processamento Digital de Sinais**

(Versão A2025M03D17)



Universidade Federal Fluminense

Alexandre Santos de la Vega

Departamento de Engenharia de Telecomunicações – TET

Escola de Engenharia – TCE

Universidade Federal Fluminense – UFF

Março – 2025

621.3192	de la Vega, Alexandre Santos
(*)	
D278	Tutorial sobre Sistema de Média Móvel para Fundamentos de Processamento Digital de Sinais / Alexandre Santos de la Vega. – Niterói: UFF/TCE/TET, 2025.
(*)	
2025	114 (sem romanos) ou 136 (com romanos) p. (*)
	Tutorial – Graduação, Engenharia de Telecomunicações, UFF/TCE/TET, 2025.
	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, dividido em apostilas e tutoriais, encontra-se dividido nos seguintes volumes:

1. O conteúdo teórico pode ser encontrado no volume intitulado Apostila de Teoria para Fundamentos de Processamento Digital de Sinais.
2. O conteúdo prático pode ser encontrado no volume intitulado Apostila de Códigos de Programas Demonstrativos para Fundamentos de Processamento Digital de Sinais.
3. As especificações dos trabalhos extra classe propostos na disciplina podem ser encontradas no volume intitulado Apostila de Trabalhos Extra Classe de Exercício e de Código (TEC) para Fundamentos de Processamento Digital de Sinais.
4. Conteúdos matemáticos básicos, necessários à disciplina em questão, são abordados no volume intitulado Apostila de Nivelamento para Fundamentos de Processamento Digital de Sinais.
5. 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 intitulado Tutorial sobre Sistema de Média Móvel para Fundamentos de Processamento Digital de Sinais.
6. Um conteúdo associado com Teoria de Sistemas, Teoria de Controle, Teoria de Circuitos e Teoria de Processamento de Sinais, envolvendo aspectos teóricos e códigos de programas demonstrativos, pode ser encontrado no volume intitulado Tutorial sobre Influência dos zeros e dos pólos da Função de Transferência no formato da função Resposta em Frequência 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. Por outro lado, 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

Àqueles 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 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

Metodologia de construção

- O material aqui apresentado não é fruto de um projeto educacional envolvendo idealização, planejamento, pesquisa, estudo, estruturação, desenvolvimento, revisão e edição.
- Pelo contrário, ele nasceu, evoluiu e tem sido mantido de uma forma bem orgânica.

Histórico

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

- 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 L^AT_EX [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.
- 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”.
- Ainda 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.
- Desde 2011, objetivando incentivar os alunos a modificarem códigos existentes e a gerarem seus próprios códigos, uma nova apostila tem sido elaborada, contendo códigos para 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 extra classe (TEC) propostos a cada período letivo.
- 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.
- Em 2019, foi iniciado um tutorial sobre a influência dos zeros e dos pólos da Função de Transferência no formato da função Resposta em Frequência, buscando atender a uma série de motivações, listadas no documento em questão.
- A partir do isolamento social, imposto pela Pandemia de COVID-19, nos anos de 2020 e 2021, foi notada uma deficiência relativa aos conteúdos matemáticos básicos de Matrizes, Números complexos e Polinômios, necessários à disciplina em questão.
- A princípio, tais conteúdos foram apenas abordados em aulas iniciais de revisão. Em seguida, um texto inicial foi anexado à apostila de teoria, na forma de apêndices. Por fim, a partir de 2025, uma apostila de nivelamento, foi incorporada ao conjunto dos materiais autorais produzidos.

Comentários gerais:

- Conforme foi exposto acima, 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. Buscando uma melhor apresentação dos tópicos abordados, os diversos seccionamentos de texto (capítulos, seções, subseções, etc.) comumente 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>
 - Conexão entre os modelos analógico e discreto/digital: que apresenta um resumo das representações dos sinais analógicos no domínio da frequência e aborda as duas formas de conexão entre os domínios analógico e digital. [Opcional]
- Sinais e sistemas (com tempo discreto) no domínio do tempo <12 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. <4 horas>
 - Sistemas no domínio do tempo: definições, classificações, operações, exemplos e caracterizações. <4 horas>
- Representações de um Sistema Linear e Invariante ao Tempo (SLIT) <10 horas>
 - Resposta ao impulso. <1 hora>
 - Diagramas de blocos de complexidade genérica. <1 hora>
 - Equação de diferença. <1 hora>
 - Diagramas de sistema (ou estruturas ou realizações). <2 horas>
 - Operador de transferência. <1 hora>
 - Diagrama de pólos e zeros do operador de transferência. <2 horas>
 - Equações de estado. <2 horas>
 - Relações e mapeamentos entre as diversas representações. <distribuído ao longo da apresentação do conteúdo e exercitado na forma de trabalhos>
- 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. <1 hora>
 - * Cálculo da resposta de um SLIT baseado no uso do operador de transferência. <1 hora>

- * Cálculo da resposta de um SLIT baseado na solução convencional da equação de diferença. <4 horas>
- * Cálculo da resposta de um SLIT FIR (Resposta ao Impulso Finita) com entrada de comprimento indefinido. <2 horas>
- 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 Frequência: 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 frequência <20 horas>
 - Sinais <10 horas>
 - * Motivações para a mudança de domínio de uma representação. <1/2 hora>
 - * Revisão das representações em frequência com tempo contínuo (Série de Fourier, Transformada de Fourier e Transformada de Laplace). <1/2 hora>
 - * Série de Fourier de Tempo Discreto (DTFS). <1 hora>
 - * Transformada de Fourier de Tempo Discreto (DTFT). <2 horas>
 - * Transformada de Fourier Discreta (DFT). <2 horas>
 - * Transformada Z. <2 horas>
 - * Relações entre as diversas representações em frequência, parâmetros e efeitos importantes. <2 horas>
 - Técnicas básicas para aceleração do cálculo da DFT. [Opcional]
 - Aplicações básicas da Transformada Z em sinais e sistemas. <6 horas>
 - * Transformada Z de alguns sinais importantes.
 - * Transformada Z aplicada na convolução.
 - * Transformada Z aplicada em sinais deslocados.
 - * Transformada Z aplicada na equação de diferença.
 - * Transformada Z no cálculo de respostas de um SLIT.

- SLIT de ordem qualquer <4 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 frequência, utilizando-os na composição dos demais sinais envolvidos.
- Discutir a análise de sistemas no domínio da variável tempo e no domínio da variável frequência. No domínio do tempo, o foco está na FORMA que os sinais apresentam. No domínio da frequência, 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 frequência), 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	xvii
Sumário	xix
I Sistema de Média Móvel	1
1 Sistema de Média Móvel	3
1.1 Introdução	3
1.2 Definição do sistema de média móvel	3
1.3 Sistemas não recursivos e o SMM	4
II Teoria	5
2 Sinais e sistemas no domínio do tempo	7
2.1 Introdução	7
2.2 Operações sobre sequências	7
2.3 Classificações de sistemas	7
2.4 Tipos de implementações	9
3 Representações de SLIT no domínio do tempo	11
3.1 Introdução	11
3.2 Resposta ao impulso	11
3.3 Associações de sistemas	12
3.4 Equação de diferença	13
3.5 Classificação de um SLIT quanto à saída	14
3.5.1 Forma de cálculo da saída	14
3.5.2 Resposta ao impulso	14
3.5.3 Relação entre as classificações	14
3.5.4 Condição inicial da equação de diferença	14
3.5.5 Sistemas SLIT FIR com coeficientes simétricos	15
3.6 Diagrama de sistema ou realização ou estrutura	15

3.7	Operador de transferência	16
3.8	Conjunto ZPK do operador de transferência	17
3.9	Equações de estado	17
3.9.1	Definição	17
3.9.2	Exemplos de equações genéricas	18
3.9.3	Exemplos de equações para um SMM	18
4	Respostas de um SLIT no domínio do tempo	21
4.1	Introdução	21
5	Sinais no domínio da frequência	23
5.1	Introdução	23
5.2	DTFS (<i>Discrete-Time Fourier Series</i>)	23
5.3	DTFT (<i>Discrete-Time Fourier Transform</i>)	24
5.4	DFT (<i>Discrete Fourier Transform</i>)	24
5.5	Transformada Z	25
5.6	Relação entre a Transformada Z, a DTFT e o Círculo Unitário	26
6	Representações de SLIT no domínio da frequência	27
6.1	Introdução	27
6.2	Função Resposta em Frequência	27
6.3	Função de Transferência	28
6.4	Conjunto ZPK da Função de Transferência	29
6.5	Relação entre a Função de Transferência, a Resposta em Frequência e o Círculo Unitário	30
7	Respostas de um SLIT no domínio da frequência	31
III	Projeto	33
8	Projeto de filtros por amostragem na frequência	35
8.1	Introdução	35
8.2	Conceitos básicos	35
8.2.1	<i>Comb filter</i>	36
8.2.2	<i>Resonator</i>	37
8.2.3	Conexão série de <i>comb filter</i> com <i>resonator</i>	39
8.3	Projeto de filtros FIR por amostragem na frequência	42
8.3.1	Projeto de filtro de ordem 1	43
8.3.2	Projeto de filtro de ordem 2	43
8.3.3	Projeto de filtro de ordem M_T	43
IV	Prática	45
9	Códigos para Octave	47
9.1	Introdução	47
9.2	Operações sobre sequências	47
9.2.1	Deslocamentos, soma e escalamento	47
9.2.2	Razão de funções senoidais com abscissa escalada	52

9.3	Diagrama de Pólos e Zeros (DPZ)	58
9.3.1	Função para geração de DPZ (com marcação de singularidades múltiplas)	58
9.3.2	ZPK e DPZ do SMM	62
9.4	Respostas de um SLIT no domínio do tempo	63
9.4.1	Equação recursiva, resposta à entrada e resposta ao estado	63
9.4.2	Equação recursiva, estrutura em cascata e resposta ao impulso	66
9.5	Função Resposta em Frequência	70
9.5.1	Curvas do SMM para diferentes valores de M_T	70
9.5.2	DPZ, módulo e ângulo de fase, da Resposta em Frequência do SMM, para diferentes valores de M_T	74
9.5.3	N -point DFT da resposta ao impulso	79
9.5.4	Comparação entre SMM e filtro projetado com $fir(N, \Omega_p)$	84
9.5.5	Relação entre a Função de Transferência, a Resposta em Frequência e o Círculo Unitário	98
10	Códigos para <i>kit</i> Arduino	107
10.1	Introdução	107
10.2	Sensor LDR	107
10.3	Medidor de vazão	109
	Referências bibliográficas	113

Parte I

Sistema de Média Móvel

Capítulo 1

Sistema de Média Móvel

1.1 Introdução

O conteúdo da disciplina Fundamentos de Processamento Digital de Sinais trata da área matemática de Sinais e Sistemas, discretos e digitais. Tal conteúdo possui, naturalmente, vários tópicos independentes, embora relacionados entre si. Por serem independentes, os tópicos são apresentados isoladamente uns dos outros, embora sejam relacionados e integrados à medida que são abordados.

O sistema de média móvel é de fácil formulação, de simples entendimento e de implementação direta. Na implementação por *software*, o código é extremamente simples, tanto em linguagens de programação de uso geral quanto em ambientes de simulação matemática, bem como em *kits* de desenvolvimento. Na implementação por *hardware*, os circuitos digitais necessários são blocos funcionais básicos.

Portanto, 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.

Nesse sentido, foi elaborado um documento que procura reunir e unificar tópicos relevantes da disciplina, utilizando o sistema de média móvel como elemento de integração.

Além dos aspectos teóricos, também foram incluídos códigos para o aplicativo matemático Octave e para uma aplicação física com o *kit* de desenvolvimento Arduino.

1.2 Definição do sistema de média móvel

De uma forma geral, um sistema de média móvel (SMM) tem a função de calcular o n -ésimo valor da sequência de saída $y[n]$ por meio da média simples de M_T valores da sequência de entrada $x[n]$. Isso é matematicamente definido por

$$y[n] = (x[n - M_2] + x[n - (M_2 - 1)] + \cdots + x[n - (M_1 + 1)] + x[n - M_1]) / M_T \quad (1.1)$$

ou

$$y[n] = \frac{1}{M_T} \sum_{k=M_1}^{M_2} x[n - k], \quad (1.2)$$

onde $M_T = (M_2 - M_1 + 1)$, $M_2 > M_1$ e $\{M_2, M_1\} \in \mathbb{Z}$.

Em um caso particular, a média pode ser calculada em torno do n -ésimo valor da sequência de entrada. Nesse caso, o sistema é definido por

$$y[n] = \frac{(x[n - M_P] + x[n - (M_P - 1)] + \cdots + x[n] + \cdots + x[n + (M_F - 1)] + x[n + M_F])}{M_T} \quad (1.3)$$

ou

$$y[n] = \frac{1}{M_T} \sum_{k=-M_F}^{M_P} x[n - k], \quad (1.4)$$

onde $M_T = (M_P + M_F + 1)$ e $\{M_P, M_F\} \in \mathbb{N}$.

Em um outro caso particular, com $M_1 = M_F = 0$, a média pode ser calculada com valores não futuros da sequência de entrada. Aqui, o sistema é definido por

$$y[n] = \frac{(x[n - M_P] + x[n - (M_P - 1)] + \cdots + x[n])}{M_T} \quad (1.5)$$

ou

$$y[n] = \frac{1}{M_T} \sum_{k=0}^{M_P} x[n - k], \quad (1.6)$$

onde $M_T = (M_P + 1)$ e $M_P \in \mathbb{N}^+$.

1.3 Sistemas não recursivos e o SMM

Sistemas não recursivos são aqueles onde o cálculo da sua variável de saída $y[n]$ não emprega a própria variável de saída. Portanto, um SMM pode ser interpretado como um caso particular de sistemas não recursivos.

Assim, a Equação 1.2 pode ser vista como um caso particular de

$$y[n] = \sum_{k=M_1}^{M_2} b_k x[n - k]. \quad (1.7)$$

Por sua vez, a Equação 1.4 pode ser vista como um caso particular de

$$y[n] = \sum_{k=-M_F}^{M_P} b_k x[n - k]. \quad (1.8)$$

Finalmente, a Equação 1.6 pode ser vista como um caso particular de

$$y[n] = \sum_{k=0}^{M_P} b_k x[n - k]. \quad (1.9)$$

Em todos os equacionamentos, o SMM é o caso particular onde $b_k = 1/M_T$.

Parte II

Teoria

Capítulo 2

Sinais e sistemas no domínio do tempo

2.1 Introdução

A seguir, são apresentados alguns conceitos básicos sobre sinais e sistemas no domínio do tempo.

2.2 Operações sobre sequências

As equações de definição do SMM podem ser interpretadas de diversas formas. Do ponto de vista de operações básicas sobre sequências, pode-se dizer que o sistema realiza uma primeira etapa de deslocamento, seguida de uma fase de soma e, finalmente, de um escalamento de amplitude. Por exemplo, na Equação (1.3), tais etapas são definidas por

$$v_k[n] = x[n - k] ,$$

$$w[n] = \sum_{k=-M_F}^{M_P} v_k[n]$$

e

$$y[n] = \frac{1}{M_T} w[n] .$$

2.3 Classificações de sistemas

Quanto ao tipo de variáveis manipuladas pelo sistema, o SMM aqui abordado é do tipo amostrado ou de tempo discreto.

Quanto ao número de entradas e saídas, o sistema é do tipo SISO (*Single-Input Single-Output*).

Um vez que o SMM calcula uma média de valores, ele precisa armazená-los para realizar o cálculo. Sendo assim, ele deve possuir memória e, por isso, é um sistema dinâmico.

Quanto à linearidade, suponha-se que o SMM relaxado gera as saídas $y_1[n]$ e $y_2[n]$, para as

entradas $x_1[n]$ e $x_2[n]$, respectivamente. Logo, a partir da Equação (1.4), tem-se que

$$\begin{aligned}
 y[n] &= \frac{1}{M_T} \sum_{k=-M_F}^{M_P} (c_1 x_1[n-k] + c_2 x_2[n-k]) \\
 &= \frac{1}{M_T} \sum_{k=-M_F}^{M_P} (c_1 x_1[n-k]) + \frac{1}{M_T} \sum_{k=-M_F}^{M_P} (c_2 x_2[n-k]) \\
 &= c_1 \left(\frac{1}{M_T} \sum_{k=-M_F}^{M_P} x_1[n-k] \right) + c_2 \left(\frac{1}{M_T} \sum_{k=-M_F}^{M_P} x_2[n-k] \right) \\
 &= c_1 y_1[n] + c_2 y_2[n],
 \end{aligned}$$

o que prova que o SMM é um sistema linear.

Quanto à invariância ao tempo (ou ao deslocamento), supondo-se que a entrada deslocada $x_D[n] = x[(n - N_D)]$, com $N_D \in \mathbb{Z}$, é aplicada ao SMM, a sua saída pode ser calculada por

$$\begin{aligned}
 y_D[n] &= \frac{1}{M_T} \sum_{k=-M_F}^{M_P} x_D[n-k] \\
 &= \frac{1}{M_T} \sum_{k=-M_F}^{M_P} x[(n - N_D) - k] \\
 &= y[(n - N_D)],
 \end{aligned}$$

o que prova que o SMM é um sistema invariante ao tempo (ou ao deslocamento).

Quanto à causalidade, a classificação depende dos limites envolvidos no cálculo da média. Se, nas Equações (1.1) a (1.4), os limites forem $M_1 < 0$, $M_2 < 0$ ou $M_F > 0$, então o SMM será não causal. Caso contrário, o SMM será causal.

Sem perda de generalidade, um sistema não causal pode ser descrito por um sistema causal, por meio de uma troca de variáveis. Matematicamente, isso é feito da seguinte forma:

$$\begin{aligned}
 y[n] &= \frac{1}{M_T} \sum_{k=-M_F}^{M_P} x[n-k] \\
 &= \frac{1}{M_T} \sum_{l=0}^{M_P+M_F} x[n - (l - M_F)] \\
 &= \frac{1}{M_T} \sum_{l=0}^{M_T-1} v[n-l],
 \end{aligned} \tag{2.1}$$

onde

$$v[n] = x[n + M_F] \longleftrightarrow v[n - M_F] = x[n]. \tag{2.2}$$

As Equações (2.1) e (2.2) mostram que, dado um sistema causal, com entrada $v[n]$ e saída $y[n]$, basta adicionar uma linha de retardo com M_F atrasos para implementar um sistema não causal, com entrada $x[n]$ e saída $y[n]$. Nesse caso, o atraso de valor M_F é denominado de **latência**, em relação à entrada $v[n]$, dado que a saída $y[n]$ só é calculada depois que M_F valores de $v[n]$ são recebidos. Por outro lado, o fluxo de dados (*throughput*) não é alterado, uma vez que, para cada novo valor de $v[n]$, é gerado um novo valor de $y[n]$.

2.4 Tipos de implementações

A implementação de um SMM pode ser efetuada por meio das seguintes alternativas:

- Programação com linguagens de alto nível.
- Programação em ambientes de simulação matemática.
- Programação de processadores (microprocessador, microcontrolador, DSP).
- Emprego de circuitos digitais (CPLD/EPLD, FPGA).

Em qualquer das formas acima citadas, a implementação é de extrema simplicidade, uma vez que deverão ser implementadas apenas as operações de armazenamento, adição e escalamento.

No caso de implementação em circuito digital, a operação mais custosa é o escalamento. Porém, se for possível utilizar um escalamento por uma potência de dois, o escalador pode ser substituído por uma simples operação de deslocamento de *bits*. No caso extremo, onde o escalador possa ser evitado, a implementação torna-se ainda mais simples.

Capítulo 3

Representações de SLIT no domínio do tempo

3.1 Introdução

A seguir, são abordadas algumas representações de um SLIT, no domínio do tempo, utilizando um SMM como exemplo.

3.2 Resposta ao impulso

Considerando-se um SMM relaxado (estado inicial nulo), com entrada $x[n] = \delta[n]$, a resposta ao impulso $h[n]$ é dada por

$$\begin{aligned} h[n] &= \frac{1}{M_T} \sum_{k=M_1}^{M_2} \delta[n-k] \\ &= \begin{cases} \frac{1}{M_T} & , \quad M_1 \leq n \leq M_2 \\ 0 & , \quad \text{caso contrário} \end{cases} \\ &= \frac{1}{M_T} (u[n-M_1] - u[n-(M_2+1)]) \\ &= \frac{1}{M_T} (\delta[n-M_1] - \delta[n-(M_2+1)]) * u[n] \\ &= \frac{1}{M_T} (h_1[n] * h_2[n]) \quad , \end{aligned} \tag{3.1}$$

onde: $M_T = (M_2 - M_1 + 1)$, $h_1[n] = (\delta[n - M_1] - \delta[n - (M_2 + 1)])$ e $h_2[n] = u[n]$.

No caso em que $M_1 = 0$ e $M_2 = (M_T - 1)$, obtém-se

$$\begin{aligned}
 h[n] &= \frac{1}{M_T} \sum_{k=0}^{M_T-1} \delta[n-k] \\
 &= \begin{cases} \frac{1}{M_T} & , \quad 0 \leq n \leq (M_T - 1) \\ 0 & , \quad \text{caso contrário} \end{cases} \\
 &= \frac{1}{M_T} (u[n] - u[n - M_T]) \\
 &= \frac{1}{M_T} (\delta[n] - \delta[n - M_T]) * u[n] \\
 &= \frac{1}{M_T} (h_1[n] * h_2[n]) , \tag{3.2}
 \end{aligned}$$

onde: $h_1[n] = (\delta[n] - \delta[n - M_T])$ e $h_2[n] = u[n]$.

As Equações (3.1) e (3.2) mostram que resposta ao impulso do SMM é um *gate* retangular com amplitude $A_G = 1/M_T$.

3.3 Associações de sistemas

As Equações (3.1) e (3.2) indicam que um SMM pode ser interpretado como um sistema composto por três subsistemas em cascata.

Considerando-se um SMM causal, definido pela Equação (3.2), o primeiro sistema é um escalador, definido por

$$v[n] = K x[n] ,$$

com ganho $K = 1/M_T$.

O segundo sistema, conhecido como *Comb Filter*, apresenta resposta ao impulso

$$h_1[n] = \delta[n] - \delta[n - M_T] = h_{11}[n] - h_{12}[n] ,$$

o que indica que o mesmo é formado pela conexão paralela de dois subsistemas deslocadores,

$$w[n] = w_{11}[n] - w_{12}[n] ,$$

definidos por

$$w_{1k}[n] = v[n - N_{D_k}] ,$$

com deslocamentos $N_{D_1} = 0$ e $N_{D_2} = M_T$, respectivamente.

Finalmente, o terceiro sistema apresenta resposta ao impulso

$$h_2[n] = u[n] ,$$

o que indica que ele é definido por

$$y[n] = \sum_{k=-\infty}^n w[k] ,$$

que é a equação de um sistema acumulador, também conhecido como *Resonator*, a qual pode ser reescrita como

$$y[n] = \sum_{k=-\infty}^n w[k] = \sum_{k=-\infty}^{n-1} w[k] + w[n] = y[n-1] + w[n] .$$

3.4 Equação de diferença

Dada a equação de diferença genérica de um sistema causal

$$y[n] = - \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^L b_k x[n-k] , \quad (3.3)$$

a equação de diferença de um SMM causal, definido pela Equação (3.2), é

$$y[n] = \sum_{k=0}^L b_k x[n-k] = \sum_{k=0}^{M_T-1} \frac{1}{M_T} x[n-k] . \quad (3.4)$$

Por outro lado, considerando-se que o mesmo sistema é formado pela conexão em cascata de três subsistemas, a sua equação de diferença assume a forma

$$\begin{aligned} y[n] &= - \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^L b_k x[n-k] \\ &= -a_1 y[n-1] + b_0 x[n] + b_{M_T} x[n-M_T] \\ &= y[n-1] + \frac{1}{M_T} x[n] - \frac{1}{M_T} x[n-M_T] \\ &= y[n-1] + v[n] - v[n-M_T] , \end{aligned} \quad (3.5)$$

onde $v[n] = \frac{1}{M_T} x[n]$.

A Equação 3.5 também pode ser demonstrada a partir da própria definição do SMM. Da Equação 1.6, pode-se dizer que

$$\begin{aligned} y[n-1] &= \frac{1}{M_T} \sum_{l=0}^{M_T-1} x[n-1-l] \\ &= \frac{1}{M_T} \sum_{l=0}^{M_T-1} x[n-(l+1)] \\ &= \frac{1}{M_T} \sum_{k=1}^{M_T} x[n-k] , \end{aligned} \quad (3.6)$$

onde $k = (l+1)$. Manipulando-se a Equação 1.6 e empregando-se a Equação 3.6, obtém-se

$$\begin{aligned} y[n] &= \frac{1}{M_T} \sum_{k=0}^{M_T-1} x[n-k] \\ &= \frac{1}{M_T} x[n] + \frac{1}{M_T} \sum_{k=1}^{M_T-1} x[n-k] \\ &= \frac{1}{M_T} x[n] + \frac{1}{M_T} \sum_{k=1}^{M_T-1} x[n-k] + \frac{1}{M_T} x[n-M_T] - \frac{1}{M_T} x[n-M_T] \\ &= \frac{1}{M_T} x[n] + \frac{1}{M_T} \sum_{k=1}^{M_T} x[n-k] - \frac{1}{M_T} x[n-M_T] \\ &= \frac{1}{M_T} x[n] + y[n-1] - \frac{1}{M_T} x[n-M_T] , \end{aligned}$$

que é a equação de diferença definida em (3.5).

3.5 Classificação de um SLIT quanto à saída

3.5.1 Forma de cálculo da saída

Dado um SLIT definido pela Equação 3.3, de acordo com a forma pela qual a sua saída é calculada, o sistema pode ser classificado como: recursivo ou não recursivo. Em um sistema recursivo, o valor da saída atual depende de valores da entrada e de valores passados da própria saída. Em um sistema não recursivo, o valor da saída atual depende apenas de valores da entrada.

Baseado na classificação acima, Equação 3.5 indica uma forma de cálculo recursiva.

Por sua vez, Equação 3.4 mostra que não é necessário que o cálculo da saída seja efetuado de forma recursiva, ao definir que ele pode ser realizado de forma não recursiva.

Portanto, o SMM é um exemplo claro de que, embora um sistema possa apresentar uma definição recursiva, o mesmo pode possuir uma dinâmica de operação não recursiva.

3.5.2 Resposta ao impulso

A resposta ao impulso de um sistema é a saída calculada quando a entrada é um impulso unitário e o estado inicial é nulo.

Quando a resposta ao impulso de um sistema possui um número finito de valores não nulos, ele é classificado como FIR (*Finite Impulse Response*). Caso contrário, o sistema é classificado como IIR (*Infinite Impulse Response*).

A Equação 3.4 mostra claramente que o SMM é um sistema do tipo FIR.

Portanto, o SMM é um exemplo claro de que, embora um sistema possa apresentar uma definição recursiva, o mesmo pode ser do tipo FIR.

3.5.3 Relação entre as classificações

A princípio, as classificações acima são independentes, por se basearem em parâmetros independentes. Mas, por razões estruturais, existe uma relação entre elas.

Por um lado, demonstra-se que os sistemas com definição não recursiva são do tipo FIR.

Por sua vez, a natureza recursiva do cálculo da saída, tende a sustentar uma saída não nula após a aplicação do impulso. Assim, boa parte dos sistemas recursivos é do tipo IIR. Porém, isso não necessariamente é verdade. O próprio SMM é um contraexemplo: embora exiba uma formulação recursiva, o mesmo possui uma dinâmica de operação não recursiva, sendo classificado como FIR.

3.5.4 Condição inicial da equação de diferença

Para calcular a resposta do SMM, a partir da forma recursiva da sua equação de diferença, é necessário que se defina uma condição inicial $y[-1]$. Porém, dado que o SMM é do tipo FIR, bem como assumindo-se que a entrada é nula para $n < 0$, conclui-se que a condição inicial deve ser nula ($y[-1] = 0$).

3.5.5 Sistemas SLIT FIR com coeficientes simétricos

Há um grande interesse prático na classe de sistemas do tipo SLIT FIR que apresentam simetrias par ou ímpar nos seus coeficientes.

Considerando-se, sem perda de generalidade, o sistema causal

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

os coeficientes b_k podem assumir as seguintes configurações:

- N par e simetria par: $\mathbf{b} = [b_0, b_1, b_2, b_3, b_4, b_5, b_6] = [b_0, b_1, b_2, b_3, b_2, b_1, b_0]$.
- N par e simetria ímpar: $\mathbf{b} = [b_0, b_1, b_2, b_3, b_4, b_5, b_6] = [b_0, b_1, b_2, 0, -b_2, -b_1, -b_0]$.
- N ímpar e simetria par: $\mathbf{b} = [b_0, b_1, b_2, b_3, b_4, b_5] = [b_0, b_1, b_2, b_2, b_1, b_0]$.
- N ímpar e simetria ímpar: $\mathbf{b} = [b_0, b_1, b_2, b_3, b_4, b_5] = [b_0, b_1, b_2, -b_2, -b_1, -b_0]$.

Do ponto de vista temporal, tais sistemas permitem uma redução na quantidade de circuitos multiplicadores, que são os blocos com maiores custos.

Do ponto de vista frequencial, esses sistemas podem ser projetados para apresentarem um perfil de seletividade em frequência controlado, além de apresentarem a característica de ângulo de fase linear e, portanto, um atraso de grupo constante.

Portanto, de acordo com tal classificação, o SMM é um caso particular da classe de sistemas do tipo SLIT FIR com coeficientes que apresentam simetria par, uma vez que $b_k = 1/M_T$.

3.6 Diagrama de sistema ou realização ou estrutura

A Equação (3.4) indica que um SMM pode ser descrito por uma estrutura não recursiva padrão, composta por uma linha de retardo de $(M_T - 1)$ atrasadores, M_T escaladores e $(M_T - 1)$ somadores.

Reescrevendo-se a Equação (3.4) como

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

pode-se propor uma estrutura composta por um único escalador em cascata com uma estrutura não recursiva otimizada, composta por uma linha de retardo de $(M_T - 1)$ atrasadores e $(M_T - 1)$ somadores.

Por sua vez, a Equação (3.5) indica que um SMM pode ser descrito por uma estrutura recursiva, composta por uma linha de retardo de $(M_T + 1)$ atrasadores, 2 escaladores e 2 somadores.

Reescrevendo-se a Equação (3.5) como

$$y[n] = y[n-1] + \frac{1}{M_T} (x[n] - x[n - M_T]) ,$$

pode-se propor uma outra estrutura recursiva, otimizada, composta por três subsistemas: um escalador, em cascata com uma linha de retardo de M_T atrasadores e 1 subtrator, em cascata com 1 atrasador e 1 somador.

3.7 Operador de transferência

Aplicando-se o operador de deslocamento $D^{-N_D}\{\cdot\}$, definido por

$$w[n] = v[n - N_D] = D^{-N_D}\{v[n]\} = (D^{-N_D})v[n]$$

na Equação (3.3), obtém-se o operador de transferência $T(D)$, definido por

$$y[n] = T(D) x[n] = \left(\frac{\sum_{k=0}^L b_k D^{-k}}{1 + \sum_{k=1}^N a_k D^{-k}} \right) x[n].$$

Aplicando-se o operador de deslocamento na Equação (3.4), obtém-se o operador de transferência

$$\begin{aligned} y[n] &= T(D) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\sum_{k=0}^{M_T-1} D^{-k} \right) x[n] \\ &= \left(\frac{1}{M_T} \right) (1 + D^{-1} + D^{-2} + \dots + D^{-(M_T-1)}) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\frac{1}{D^{M_T-1}} \right) (D^{M_T-1} + \dots + D^2 + D + 1) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\frac{1}{D^{M_T-1}} \right) \left(\sum_{k=0}^{M_T-1} D^k \right) x[n]. \end{aligned} \quad (3.7)$$

Aplicando-se o operador de deslocamento na Equação (3.5), obtém-se o operador de transferência

$$\begin{aligned} y[n] &= T(D) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\frac{1 - D^{-M_T}}{1 - D^{-1}} \right) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\frac{1}{D^{M_T}} \right) \left(\frac{D^{M_T} - 1}{D - 1} \right) (D) x[n] \\ &= \left(\frac{1}{M_T} \right) \left(\frac{1}{D^{M_T-1}} \right) \left(\frac{D^{M_T} - 1}{D - 1} \right) x[n]. \end{aligned} \quad (3.8)$$

Os operadores de transferência definidos nas Equações (3.7) e (3.8) podem ser diretamente relacionados ao se observar que

$$\begin{aligned} \frac{D^{M_T} - 1}{D - 1} &= D^{M_T-1} + \frac{D^{M_T-1} - 1}{D - 1} \\ &= D^{M_T-1} + D^{M_T-2} + \frac{D^{M_T-2} - 1}{D - 1} \\ &= D^{M_T-1} + D^{M_T-2} + \dots + D^1 + \frac{D^1 - 1}{D - 1} \\ &= D^{M_T-1} + D^{M_T-2} + \dots + D^1 + D^0 + \frac{D^0 - 1}{D - 1} \\ &= D^{M_T-1} + D^{M_T-2} + \dots + D + 1. \end{aligned} \quad (3.9)$$

Portanto, embora a Equação (3.8) indique que o SMM é um sistema recursivo, a terceira linha da Equação (3.9) mostra que ocorre um cancelamento pólo-zero em $D = 1$, transformando-o no sistema não recursivo definido na Equação (3.7).

3.8 Conjunto ZPK do operador de transferência

O operador de transferência definido na Equação (3.7) pode ser reescrito como

$$\begin{aligned} T(D) &= \left(\frac{1}{M_T} \right) \frac{(D^{M_T-1} + \dots + D^2 + D + 1)}{(D^{M_T-1})} \\ &= (K_C) \frac{(D - z_1)(D - z_2) \cdots (D - z_{M_T-1})}{(D - p)^{M_T-1}}, \end{aligned} \quad (3.10)$$

onde $K_C = (1/M_T)$, $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ e $\mathbf{p} = [p_1, p_2, \dots, p_{M_T-1}] = [0, 0, \dots, 0]$, são, respectivamente, o ganho, o vetor de zeros e o vetor de pólos do operador de transferência.

Por sua vez, a Equação (3.8) acusa a existência de M_T zeros e M_T pólos. O denominador da equação indica um pólo em $p_0 = 1$ e os demais $(M_T - 1)$ pólos em $p_k = 0$. O numerador da equação representa as M_T raízes complexas de 1. Logo, considerando-se $z_0 = 1$, os M_T zeros de $T(D)$, $\mathbf{z} = [z_0, z_1, z_2, \dots, z_{M_T-1}]$, encontram-se igualmente espaçados ao longo do círculo unitário do plano complexo. Caso M_T seja ímpar, os zeros $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ ocorrerão em pares complexos conjugados. Porém, se M_T for par, além dos pares complexos conjugados e do zero em $z_0 = 0$, haverá um zero em $z_{\frac{M_T}{2}} = \pi$.

Deve-se notar que, ainda de acordo com a Equação (3.8), ocorre um cancelamento entre o zero $z_0 = 1$ e o pólo $p_0 = 1$, o que estabelece um relacionamento consistente entre todas as representações.

3.9 Equações de estado

3.9.1 Definição

Um SLIT MIMO (*Multiple-Input Multiple-Output*) pode ser representado pelas equações de estado

$$\mathbf{x}[n+1] = \mathbf{A} \cdot \mathbf{x}[n] + \mathbf{B} \cdot \mathbf{r}[n] \quad (3.11)$$

e

$$\mathbf{y}[n] = \mathbf{C} \cdot \mathbf{x}[n] + \mathbf{D} \cdot \mathbf{r}[n], \quad (3.12)$$

onde:

- Vetor de sequências de entrada: $\mathbf{r}[n] = [r_1[n] \ r_2[n] \ \cdots \ r_R[n]]^T$.
- Vetor de sequências de saída: $\mathbf{y}[n] = [y_1[n] \ y_2[n] \ \cdots \ y_Y[n]]^T$.
- Vetor de sequências de variáveis de estado: $\mathbf{x}[n] = [x_1[n] \ x_2[n] \ \cdots \ x_X[n]]^T$.
- Vetor de estado inicial: $\mathbf{x}[0]$.
- Matrizes que definem o sistema: \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} .

3.9.2 Exemplos de equações genéricas

Dado um SLIT SISO genérico, com $N = L = 4$, descrito por

$$y[n] = - \sum_{k=1}^4 a_k y[n-k] + \sum_{k=0}^4 b_k r[n-k] , \quad (3.13)$$

pode-se utilizar a estrutura Forma Direta II e reescrever as Equações (3.11) e (3.12) como

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \\ x_4[n+1] \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [r[n]] \quad (3.14)$$

e

$$[y[n]] = [(b_1 - b_0 a_1) \quad (b_2 - b_0 a_2) \quad (b_3 - b_0 a_3) \quad (b_4 - b_0 a_4)] \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + [b_0] [r[n]] . \quad (3.15)$$

Por sua vez, empregando-se a estrutura Forma Direta II Transposta, pode-se reescrever as Equações (3.11) e (3.12) como

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \\ x_4[n+1] \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & 0 \\ -a_2 & 0 & 1 & 0 \\ -a_3 & 0 & 0 & 1 \\ -a_4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + \begin{bmatrix} (b_1 - b_0 a_1) \\ (b_2 - b_0 a_2) \\ (b_3 - b_0 a_3) \\ (b_4 - b_0 a_4) \end{bmatrix} [r[n]] \quad (3.16)$$

e

$$[y[n]] = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + [b_0] [r[n]] . \quad (3.17)$$

3.9.3 Exemplos de equações para um SMM

Para a estrutura Forma Direta, associada à Equação (3.4), com $N = L = (M_T - 1) = 4$ e $r[n] = x[n]$, as Equações (3.14) e (3.15) tornam-se

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \\ x_4[n+1] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [\frac{1}{M_T} r[n]] \quad (3.18)$$

e

$$[y[n]] = [1 \quad 1 \quad 1 \quad 1] \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + [1] [\frac{1}{M_T} r[n]] . \quad (3.19)$$

Por sua vez, para a estrutura Forma Direta Transposta, associada à Equação (3.4), com $N = L = (M_T - 1) = 4$ e $r[n] = x[n]$, as Equações (3.16) e (3.17) tornam-se

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \\ x_4[n+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \left[\frac{1}{M_T} r[n] \right] \quad (3.20)$$

e

$$\begin{bmatrix} y[n] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \left[\frac{1}{M_T} r[n] \right]. \quad (3.21)$$

Dado que o SMM é do tipo FIR, bem como assumindo-se que a entrada é nula para $n < 0$, conclui-se que o estado inicial deve ser nulo ($\mathbf{x}[0] = \mathbf{0}$).

Capítulo 4

Respostas de um SLIT no domínio do tempo

4.1 Introdução

Um dos principais objetivos a serem alcançados quando se modela um sistema é o cálculo da sua saída, a qual é interpretada como uma resposta do sistema a uma energia aplicada externamente e/ou a uma energia internamente armazenada.

Na definição do problema de cálculo da resposta de um sistema, podem ser utilizados os seguintes dados:

- Equação de diferença ou operador de transferência ou equações de estado: que representam o sistema.
- Condições iniciais ou estado inicial: que representam a energia interna do sistema.
- Entradas ou excitações: que representam a energia externamente aplicada ao sistema.

Para uma determinada entrada, podem-se sugerir os seguintes procedimentos para o cálculo da resposta de um SLIT no domínio do tempo:

- Cálculo numérico iterativo, empregando implementação do sistema em *software* ou em *hardware* dedicado: solução baseada na iteração da equação de diferença ou das equações de estado, a partir das condições iniciais ou do estado inicial.
- Cálculo analítico:
 - Solução baseada em algumas iterações das equações de estado, a partir do estado inicial, e posterior tentativa de inferência de uma equação geral.
 - Solução baseada em algumas iterações da equação de diferença, a partir das condições iniciais, e posterior tentativa de inferência de uma equação geral.
 - Solução convencional da equação de diferença, por meio do cálculo das respostas homogênea $y_h[n]$, particular $y_p[n]$ e complementar $y_c[n]$, com o auxílio das condições iniciais.
 - Solução baseada no operador de transferência, por meio do cálculo das respostas ao estado $y_{est}[n]$ e à entrada $y_{ent}[n]$, com o auxílio das condições iniciais.

- Casos particulares:
 - Cálculo da resposta ao impulso $h[n]$ a partir da resposta homogênea $y_h[n]$:
 $h[0] = y_{ent}[0]$ e $h[n] = y_h[n]$, para $n > 0$ e condição inicial $h[0]$.
 - Cálculo da resposta ao impulso $h[n]$ a partir da resposta ao degrau $y_u[n]$:
 $\delta[n] = u[n] - u[n - 1] \rightarrow h[n] = y_u[n] - y_u[n - 1]$.
 - Cálculo da resposta à entrada $y_{ent}[n]$, dado um SLIT FIR relaxado, com resposta ao impulso $h[n]$: $y[n] = y_{ent}[n] + y_{est}[n] = y_{ent}[n] = x[n] * h[n] = h[n] * x[n]$.

Capítulo 5

Sinais no domínio da frequência

5.1 Introdução

A seguir, são apresentados alguns conceitos básicos sobre sinais e sistemas no domínio da frequência.

5.2 DTFS (*Discrete-Time Fourier Series*)

Dada uma sequência periódica $\tilde{x}[n]$, com período fundamental N , pode-se representá-la pela seguinte combinação de sequências exponenciais:

$$\text{DTFS: } \begin{cases} \tilde{x}[n] = \sum_{k=\langle N \rangle} \tilde{X}[k] e^{jk(\frac{2\pi}{N})n} \\ \tilde{X}[k] = \frac{1}{N} \sum_{n=\langle N \rangle} \tilde{x}[n] e^{-jk(\frac{2\pi}{N})n} \end{cases},$$

onde $\tilde{X}[k] = \tilde{X}[k \pm lN]$ e $\langle N \rangle$ representa uma faixa de N valores consecutivos.

A DTFS também pode ser expressa da seguinte forma:

$$\text{DTFS: } \begin{cases} \tilde{x}[n] = \sum_{k=\langle N \rangle} \tilde{X}[k] W_N^{-kn} \\ \tilde{X}[k] = \frac{1}{N} \sum_{n=\langle N \rangle} \tilde{x}[n] W_N^{kn} \end{cases},$$

onde $W_N = e^{-j(\frac{2\pi}{N})}$.

Considerando-se a faixa $\langle N \rangle = 0, 1, 2, \dots, (N-1)$, bem como os vetores

$$\tilde{\mathbf{x}}[n] = [\tilde{x}[0] \ \tilde{x}[1] \ \dots \ \tilde{x}[(N-1)]]^T$$

e

$$\tilde{\mathbf{X}}[k] = [\tilde{X}[0] \ \tilde{X}[1] \ \dots \ \tilde{X}[(N-1)]]^T,$$

a DTFS também pode ser expressa, matricialmente, por

$$\text{DTFS: } \begin{cases} \tilde{\mathbf{x}}[n] = \mathbf{W}_N^{-1} \tilde{\mathbf{X}}[k] \quad , \quad 0 \leq n \leq (N-1) \\ \tilde{\mathbf{X}}[k] = \mathbf{W}_N \tilde{\mathbf{x}}[n] \quad , \quad 0 \leq k \leq (N-1) \end{cases},$$

onde:

$$\mathbf{W}_N^{-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix},$$

$$\mathbf{W}_N = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix},$$

$$\mathbf{W}_N^{-1} = N \mathbf{W}_N^*$$

e

$$\mathbf{W}_N = \frac{1}{N} (\mathbf{W}_N^{-1})^*.$$

5.3 DTFT (*Discrete-Time Fourier Transform*)

Dada uma sequência não periódica $x[n]$, sem variação exponencial de amplitude, pode-se representá-la pela seguinte combinação de sequências exponenciais:

$$\text{DTFT: } \begin{cases} x[n] = \frac{1}{2\pi} \int_{\Omega=\langle 2\pi \rangle} X(e^{j\Omega}) e^{j\Omega n} d\Omega \\ X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \end{cases}.$$

Deve-se ressaltar que:

- Ω é uma variável contínua, escalar e angular.
- $X(e^{j\Omega})$ é uma função analógica e complexa.
- $X(e^{j\Omega})$ é uma função periódica, com período fundamental $\Omega_F = 2\pi$ rad.

5.4 DFT (*Discrete Fourier Transform*)

Dada uma sequência não periódica $x[n]$, sem variação exponencial de amplitude, a sua DTFT pode ser aproximada pela seguinte combinação de sequências exponenciais:

$$N\text{-point DFT: } \begin{cases} x[n] = \sum_{k=0}^{(N-1)} \tilde{X}[k] e^{jk(\frac{2\pi}{N})n} & , \quad 0 \leq n \leq (N-1) \\ \tilde{X}[k] = \frac{1}{N} \sum_{n=0}^{(N-1)} x[n] e^{-jk(\frac{2\pi}{N})n} & , \quad 0 \leq k \leq (N-1) \end{cases},$$

onde $\tilde{X}[k]$ representa a DTFS de um sinal periódico $\tilde{x}[n]$, auxiliar e intermediário, criado como extensão periódica de $x[n]$.

Observando-se que

$$X[k] = N\tilde{X}[k] = X(e^{j\Omega}) \Big|_{\Omega=k(\frac{2\pi}{N})},$$

para $0 \leq k \leq (N - 1)$, a DFT pode ser redefinida como

$$N\text{-point DFT: } \begin{cases} x[n] = \frac{1}{N} \sum_{k=0}^{(N-1)} X[k] e^{jk(\frac{2\pi}{N})n} & , \quad 0 \leq n \leq (N - 1) \\ X[k] = \sum_{n=0}^{(N-1)} x[n] e^{-jk(\frac{2\pi}{N})n} & , \quad 0 \leq k \leq (N - 1) \end{cases}$$

A DFT também pode ser expressa da seguinte forma:

$$N\text{-point DFT: } \begin{cases} x[n] = \frac{1}{N} \sum_{k=0}^{(N-1)} X[k] W_N^{-kn} & , \quad 0 \leq n \leq (N - 1) \\ X[k] = \sum_{n=0}^{(N-1)} x[n] W_N^{kn} & , \quad 0 \leq k \leq (N - 1) \end{cases}$$

onde $W_N = e^{-j(\frac{2\pi}{N})}$.

Considerando-se os vetores

$$\mathbf{x}[n] = [x[0] \ x[1] \ \cdots \ x[(N - 1)]]^T$$

e

$$\mathbf{X}[k] = [X[0] \ X[1] \ \cdots \ X[(N - 1)]]^T$$

a DFT também pode ser expressa, matricialmente, por

$$N\text{-point DFT: } \begin{cases} \mathbf{x}[n] = \mathbf{D}_N^{-1} \mathbf{X}[k] & , \quad 0 \leq n \leq (N - 1) \\ \mathbf{X}[k] = \mathbf{D}_N \mathbf{x}[n] & , \quad 0 \leq k \leq (N - 1) \end{cases}$$

onde:

$$\mathbf{D}_N^{-1} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix},$$

$$\mathbf{D}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix},$$

$$\mathbf{D}_N^{-1} = \frac{1}{N} \mathbf{D}_N^*$$

e

$$\mathbf{D}_N = N (\mathbf{D}_N^{-1})^* .$$

5.5 Transformada Z

Dada uma sequência não periódica $x[n]$, com variação exponencial de amplitude, pode-se representá-la pela seguinte combinação de sequências exponenciais:

$$\text{Transformada Z: } \begin{cases} x[n] = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \\ X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \end{cases},$$

onde C é um caminho fechado dentro da região de convergência (ROC) de $X(z)$, em torno da origem $z = 0$.

5.6 Relação entre a Transformada Z, a DTFT e o Círculo Unitário

Dada uma sequência $x[n]$ e supondo-se que existam tanto a sua DTFT, $X(e^{j\Omega})$, quanto a sua Transformada Z, $X(z)$, pode-se estabelecer uma relação entre elas.

Quando a ROC de $X(z)$ inclui o círculo de raio unitário, vale a identidade

$$X(z)|_{|z|=1} = X(e^{j\Omega}) .$$

Geometricamente, a identidade acima indica que, ao se calcular $X(z)$ ao longo de $|z| = 1$, percorre-se, na superfície $X(z)$, a curva $X(e^{j\Omega})$.

Capítulo 6

Representações de SLIT no domínio da frequência

6.1 Introdução

A seguir, são abordadas algumas representações de um SLIT, no domínio da frequência, utilizando um SMM como exemplo.

6.2 Função Resposta em Frequência

A relação entre a função Resposta em Frequência $H(e^{j\Omega})$ e a resposta ao impulso $h[n]$ de um SLIT é dada por

$$H(e^{j\Omega}) = DTFT\{h[n]\} , \quad (6.1)$$

onde $DTFT\{\cdot\}$ representa a *Discrete-Time Fourier Transform* (Transformada de Fourier em Tempo Discreto), cujo par de equações é definido por

$$DTFT: \begin{cases} x[n] = \frac{1}{2\pi} \int_{\Omega=\langle 2\pi \rangle} X(e^{j\Omega}) e^{j\Omega n} d\Omega \\ X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \end{cases} .$$

De (3.1), onde $M_T = (M_2 - M_1 + 1)$, tem-se que

$$h[n] = \begin{cases} \frac{1}{M_T} & , \quad M_1 \leq n \leq M_2 \\ 0 & , \quad \text{caso contrário} \end{cases} .$$

Logo, a $DTFT\{h[n]\}$ é dada por

$$H(e^{j\Omega}) = \frac{1}{M_T} \sum_{n=M_1}^{M_2} e^{-j\Omega n} , \quad (6.2)$$

que, por meio da identidade

$$\sum_{n=M_1}^{M_2} a^n = \frac{a^{M_1} - a^{(M_2+1)}}{1 - a} , \quad \text{para } M_2 \geq M_1 ,$$

pode ser reescrita como

$$\begin{aligned}
H(e^{j\Omega}) &= \frac{1}{M_T} \frac{e^{-j\Omega M_1} - e^{-j\Omega(M_2+1)}}{1 - e^{-j\Omega}} \\
&= \frac{1}{M_T} \frac{1 - e^{-j\Omega(M_2-M_1+1)}}{1 - e^{-j\Omega}} e^{-j\Omega M_1} \\
&= \frac{1}{M_T} \frac{e^{j\frac{\Omega}{2}(M_2-M_1+1)} - e^{-j\frac{\Omega}{2}(M_2-M_1+1)}}{e^{j\frac{\Omega}{2}} - e^{-j\frac{\Omega}{2}}} \frac{e^{-j\frac{\Omega}{2}(M_2-M_1+1)}}{e^{-j\frac{\Omega}{2}}} e^{-j\Omega M_1} \\
&= \frac{1}{M_T} \frac{e^{j\frac{\Omega}{2}(M_2-M_1+1)} - e^{-j\frac{\Omega}{2}(M_2-M_1+1)}}{e^{j\frac{\Omega}{2}} - e^{-j\frac{\Omega}{2}}} e^{-j\frac{\Omega}{2}(M_2+M_1)} \\
&= \frac{1}{M_T} \frac{e^{j\frac{\Omega}{2}M_T} - e^{-j\frac{\Omega}{2}M_T}}{e^{j\frac{\Omega}{2}} - e^{-j\frac{\Omega}{2}}} e^{-j\frac{\Omega}{2}(M_2+M_1)} \\
&= \frac{1}{M_T} \frac{\sin\left(M_T \frac{\Omega}{2}\right)}{\sin\left(\frac{\Omega}{2}\right)} e^{-j\frac{(M_2+M_1)}{2}\Omega} \\
&= \text{Diric}(\Omega, M_T) e^{-j\frac{(M_2+M_1)}{2}\Omega} \\
&= \text{Drcl}\left(\frac{\Omega}{2\pi}, M_T\right) e^{-j\frac{(M_2+M_1)}{2}\Omega} .
\end{aligned} \tag{6.3}$$

No caso de um SMM causal, onde $M_1 = 0$ e $M_T = (M_2 + 1)$, a Equação 6.3 torna-se

$$\begin{aligned}
H(e^{j\Omega}) &= \frac{1}{M_T} \frac{\sin\left(M_T \frac{\Omega}{2}\right)}{\sin\left(\frac{\Omega}{2}\right)} e^{-j\frac{(M_T-1)}{2}\Omega} \\
&= \text{Diric}(\Omega, M_T) e^{-j\frac{(M_T-1)}{2}\Omega} \\
&= \text{Drcl}\left(\frac{\Omega}{2\pi}, M_T\right) e^{-j\frac{(M_T-1)}{2}\Omega} .
\end{aligned} \tag{6.4}$$

6.3 Função de Transferência

A relação entre a Função de Transferência $H(z)$ e a resposta ao impulso $h[n]$ de um SLIT é dada por

$$H(z) = Z\{h[n]\} , \tag{6.5}$$

onde $Z\{\cdot\}$ representa a Z Transform (Transformada Z), cujo par de equações, para o caso bilateral, é definido por

$$\text{Transformada Z Bilateral: } \begin{cases} x[n] = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \\ X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \end{cases} .$$

Aplicando-se a Transformada Z Bilateral na Equação (3.4), obtém-se a Função de Transferência

$$\begin{aligned}
H(z) &= Z\{h[n]\} \\
&= \frac{1}{M_T} \sum_{k=0}^{M_T-1} z^{-k} \\
&= \frac{1}{M_T} (1 + z^{-1} + z^{-2} + \dots + z^{-(M_T-1)}) \\
&= \frac{1}{M_T} \frac{1}{D^{M_T-1}} (z^{M_T-1} + \dots + z^2 + z + 1) \\
&= \frac{1}{M_T} \frac{1}{z^{M_T-1}} \sum_{k=0}^{M_T-1} z^k.
\end{aligned} \tag{6.6}$$

Aplicando-se a Transformada Z Bilateral na Equação (3.5), obtém-se a Função de Transferência

$$\begin{aligned}
H(z) &= Z\{h[n]\} \\
&= \frac{1}{M_T} \frac{1 - z^{-M_T}}{1 - z^{-1}} \\
&= \frac{1}{M_T} \frac{1}{z^{M_T}} \frac{z^{M_T} - 1}{z - 1} z \\
&= \frac{1}{M_T} \frac{1}{z^{M_T-1}} \frac{z^{M_T} - 1}{z - 1}.
\end{aligned} \tag{6.7}$$

As Funções de Transferência definidas nas Equações (6.6) e (6.7) podem ser diretamente relacionados ao se observar que

$$\begin{aligned}
\frac{z^{M_T} - 1}{z - 1} &= z^{M_T-1} + \frac{z^{M_T-1} - 1}{z - 1} \\
&= z^{M_T-1} + z^{M_T-2} + \frac{z^{M_T-2} - 1}{z - 1} \\
&= z^{M_T-1} + z^{M_T-2} + \dots + z^1 + \frac{z^1 - 1}{z - 1} \\
&= z^{M_T-1} + z^{M_T-2} + \dots + z^1 + z^0 + \frac{z^0 - 1}{z - 1} \\
&= z^{M_T-1} + z^{M_T-2} + \dots + z + 1.
\end{aligned} \tag{6.8}$$

Portanto, embora a Equação (6.7) indique que o SMM é um sistema recursivo, a terceira linha da Equação (6.8) mostra que ocorre um cancelamento pólo-zero em $z = 1$, transformando-o no sistema não recursivo definido na Equação (6.6).

6.4 Conjunto ZPK da Função de Transferência

A Função de Transferência definida na Equação (6.6) pode ser reescrita como

$$\begin{aligned}
H(z) &= \left(\frac{1}{M_T} \right) \frac{(z^{M_T-1} + \dots + z^2 + z + 1)}{(z^{M_T-1})} \\
&= (K_C) \frac{(z - z_1)(z - z_2) \dots (z - z_{M_T-1})}{(z - p)^{M_T-1}},
\end{aligned} \tag{6.9}$$

onde $K_C = (1/M_T)$, $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ e $\mathbf{p} = [p_1, p_2, \dots, p_{M_T-1}] = [0, 0, \dots, 0]$, são, respectivamente, o ganho, o vetor de zeros e o vetor de pólos da Função de Transferência.

Por sua vez, a Equação (6.7) acusa a existência de M_T zeros e M_T pólos. O denominador da equação indica um pólo em $p_0 = 1$ e os demais $(M_T - 1)$ pólos em $p_k = 0$. O numerador da equação representa as M_T raízes complexas de 1. Logo, considerando-se $z_0 = 1$, os M_T zeros de $T(D)$, $\mathbf{z} = [z_0, z_1, z_2, \dots, z_{M_T-1}]$, encontram-se igualmente espaçados ao longo do círculo unitário do plano complexo. Caso M_T seja ímpar, os zeros $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ ocorrerão em pares complexos conjugados. Porém, se M_T for par, além dos pares complexos conjugados e do zero em $z_0 = 0$, haverá um zero em $z_{\frac{M_T}{2}} = \pi$.

Deve-se notar que, ainda de acordo com a Equação (6.7), ocorre um cancelamento entre o zero $z_0 = 1$ e o pólo $p_0 = 1$, o que estabelece um relacionamento consistente entre todas as representações.

6.5 Relação entre a Função de Transferência, a Resposta em Frequência e o Círculo Unitário

As Equações 6.6 a 6.9 mostram que:

- A Função de Transferência $H(z)$ do SMM possui M_T pólos e M_T zeros.
- Um pólo encontra-se em $p_0 = 1$.
- Os demais $(M_T - 1)$ pólos encontram-se em $p_k = 0$, para $1 \leq k \leq (M_T - 1)$.
- Os M_T zeros encontram-se igualmente espaçados ao longo do círculo unitário do plano complexo, nos pontos $z_k = e^{j\Omega_k} = e^{jk\Omega_0}$, onde $\Omega_0 = \left(\frac{2\pi}{M_T}\right)$, para $0 \leq k \leq (M_T - 1)$.
- Portanto, o pólo $p_0 = 1$ é cancelado pelo zero $z_0 = 1$.
- Se M_T for ímpar, os zeros $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ ocorrerão em pares complexos conjugados.
- Se M_T for par, além dos pares complexos conjugados e do zero em $z_0 = 0$, haverá um zero em $z_{\frac{M_T}{2}} = \pi$.

Uma vez que todos os pólos não cancelados de $H(z)$ possuem módulo menor que a unidade, a ROC de $H(z)$ inclui o círculo de raio unitário. Consequentemente, vale a identidade

$$H(e^{j\Omega}) = H(z)|_{|z|=1}$$

e, ao se calcular $H(z)$ ao longo de $|z| = 1$, percorre-se, na superfície $H(z)$, a curva $H(e^{j\Omega})$.

Em relação à Resposta em Frequência $H(e^{j\Omega})$ do SMM, as Equações 6.3 a 6.4 mostram que:

- $|H(e^{j\Omega})|_{\Omega=0} = 1$.
- $|H(e^{j\Omega})|_{\Omega=k\Omega_0} = 0$, para $0 \leq k \leq (M_T - 1)$, onde $\Omega_0 = \left(\frac{2\pi}{M_T}\right)$.
- Para M_T ímpar, $|H(e^{j\Omega})|_{\Omega=k2\pi} = 1$.
- Para M_T par, $|H(e^{j\Omega})|_{\Omega=k2\pi} = (-1)^k$.

Capítulo 7

Respostas de um SLIT no domínio da frequência

Para uma determinada entrada, podem-se sugerir os seguintes procedimentos para o cálculo da resposta de um SLIT no domínio da frequência:

- Cálculo da resposta à entrada completa: solução baseada na aplicação da Transformada Z na relação temporal

$$y_{ent}[n] = h[n] * x[n] ,$$

produzindo a relação frequencial

$$Y_{ent}(z) = H(z) \cdot X(z) .$$

- Cálculo da resposta à entrada em regime permanente: solução baseada na aplicação da DTFT na relação temporal

$$y_{ent}[n] = h[n] * x[n] ,$$

produzindo a relação frequencial

$$Y_{ent}(e^{j\Omega}) = H(e^{j\Omega}) \cdot X(e^{j\Omega}) .$$

- Cálculo da resposta completa: solução baseada na aplicação da Transformada Z na relação temporal

$$y[n] = - \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^L b_k x[n-k] ,$$

com condições iniciais $y[-1], y[-2], \dots, y[-N]$, produzindo a relação frequencial

$$Y_{tot}(z) = Y_{ent}(z) + Y_{est}(z) = H(z) \cdot X(z) + \frac{(-1)}{D_H(z)} \left\{ \sum_{k=1}^N P_{-k}(z) y[-k] \right\} ,$$

onde:

$$Y_{ent}(z) = H(z) \cdot X(z) ,$$

$$H(z) = \frac{N_H(z)}{D_H(z)} ,$$

$$Y_{est}(z) = \frac{(-1)}{D_H(z)} \left\{ \sum_{k=1}^N P_{-k}(z) y[-k] \right\}$$

e

$$P_{-k}(z) = \sum_{l=k}^N a_l z^{-(l-k)} .$$

Parte III

Projeto

Capítulo 8

Projeto de filtros FIR com fase linear e estrutura de amostragem na frequência

8.1 Introdução

A descrição de um Sistema de Média Móvel (SMM) por meio da conexão em cascata de um escalador, um FIR e um acumulador, pode servir de ponto inicial na proposta de uma técnica de projeto para filtros FIR. O cancelamento de um dos zeros do FIR pelo pólo do acumulador, bem como a introdução de um ganho arbitrário no ponto de cancelamento, induz a ideia de inclusão de vários acumuladores em paralelo, a fim de que se realizem outros cancelamentos com outros zeros do FIR, bem como outras introduções de ganhos nos pontos de cancelamento. Se os valores dos ganhos introduzidos forem iguais aos valores de pontos específicos de uma função de transferência, pode-se pensar no projeto de um filtro por amostragem na frequência. Essa ideia é brevemente apresentada a seguir.

8.2 Conceitos básicos

A Equação (6.7) define a Função de Transferência de um Sistema de Média Móvel (SMM) causal como

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

que pode ser interpretada como

$$\begin{aligned} H(z) &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left(\frac{1}{1 - z^{-1}} \right) \\ &= H_C(z) H_{R,0}(z), \end{aligned}$$

onde

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

e

$$H_{R,0}(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}.$$

As Funções de Transferência $H_C(z)$ e $H_{R,0}(z)$ representam, respectivamente, os sistemas conhecidos como *comb filter* e acumulador. Este último pode ser visto como um caso particular de um sistema denominado de *resonator*. Ambos são discutidos a seguir.

8.2.1 Comb filter

O sistema conhecido como *comb filter* não recursivo é definido por

$$y[n] = \sum_{k=0}^L b_k x[n-k] = \frac{1}{M_T} (x[n] - x[n - M_T]) .$$

A sua resposta ao impulso é calculada por

$$h[n] = \frac{1}{M_T} (\delta[n] - \delta[n - M_T]) .$$

A sua Função de Transferência é dada por

$$\begin{aligned} H_C(z) &= \frac{1}{M_T} (1 - z^{-M_T}) \\ &= \frac{1}{M_T} \frac{z^{M_T} - 1}{z^{M_T}} \\ &= \frac{1}{M_T} \frac{(z - z_0)(z - z_1) \cdots (z - z_{M_T-1})}{z^{M_T}} , \end{aligned}$$

a qual apresenta as seguintes características:

- Há M_T pólos, $\mathbf{p} = [p_0, p_1, \dots, p_{M_T-1}]$, e M_T zeros, $\mathbf{z} = [z_0, z_1, \dots, z_{M_T-1}]$.
- Os pólos encontram-se em $p_k = 0$, para $0 \leq k \leq (M_T - 1)$.
- Os M_T zeros encontram-se igualmente espaçados ao longo do círculo unitário do plano complexo, nos pontos $z_k = e^{j\Omega_k} = e^{jk\Omega_{M_T}}$, para $0 \leq k \leq (M_T - 1)$, onde $\Omega_k = k\Omega_{M_T}$ e $\Omega_{M_T} = \left(\frac{2\pi}{M_T}\right)$.
- Para $k = 0$, tem-se um zero em $z_0 = 0$.
- Se M_T for ímpar, os zeros $\mathbf{z} = [z_1, z_2, \dots, z_{M_T-1}]$ ocorrerão em pares complexos conjugados.
- Se M_T for par, além dos pares complexos conjugados e do zero em $z_0 = 0$, haverá um zero em $z_{\frac{M_T}{2}} = -1$.

Uma vez que todos os pólos possuem módulo menor que a unidade, a ROC de $H(z)$ inclui o círculo de raio unitário. Consequentemente, vale a identidade

$$\begin{aligned} H(e^{j\Omega}) &= H(z)|_{|z|=1} \\ &= \frac{1}{M_T} (1 - e^{-j\Omega M_T}) \\ &= \frac{1}{M_T} \frac{e^{j\Omega M_T} - 1}{e^{j\Omega M_T}} \\ &= \frac{1}{M_T} (2j) \sin\left(M_T \frac{\Omega}{2}\right) e^{-jM_T \frac{\Omega}{2}} \end{aligned} \tag{8.1}$$

e, ao se calcular $H(z)$ ao longo de $|z| = 1$, percorre-se, na superfície $H(z)$, a curva $H(e^{j\Omega})$.

Em relação à Resposta em Frequência $H(e^{j\Omega})$, pode-se dizer, de forma geral, que:

- $H(e^{j\Omega})$ é periódica, com período fundamental $\Omega_f = 2\pi$.
- $|H(e^{-j\Omega})| = |H(e^{j\Omega})|$.
- $\angle H(e^{-j\Omega}) = -\angle H(e^{j\Omega})$.
- $|H(e^{j\Omega})|_{\Omega=k\Omega_{M_T}} = 0$.
- $|H(e^{j\Omega})|_{\Omega=\frac{(2k+1)}{2}\Omega_{M_T}} = \frac{2}{M_T}$.

Já nos casos particulares, para $-\infty \leq k \leq \infty$, tem-se que:

- $|H(e^{j\Omega})|_{\Omega=0} = 0$.
- Para M_T ímpar, $|H(e^{j\Omega})|_{\Omega=(2k+1)\pi} = |H(e^{j\Omega})|_{\Omega=k2\pi} = \frac{2}{M_T}$.
- Para M_T par, $|H(e^{j\Omega})|_{\Omega=(2k+1)\pi} = |H(e^{j\Omega})|_{\Omega=k2\pi} = 0$.

Ao se traçar a curva $|H(e^{j\Omega})|$, observa-se que ela lembra a figura de um pente. Daí a denominação de *comb filter*, que é usada para identificar o sistema.

8.2.2 Resonator

O sistema conhecido como *resonator* é definido por

$$\begin{aligned} y[n] &= -a_1 y[n-1] + b_0 x[n] \\ &= e^{j\Omega_k} y[n-1] + G_k x[n], \end{aligned} \quad (8.2)$$

onde $\Omega_k = k\Omega_{M_T}$ e $\Omega_{M_T} = \left(\frac{2\pi}{M_T}\right)$.

A sua resposta ao impulso é calculada por

$$y[n] = e^{j\Omega_k} y[n-1] + G_k \delta[n].$$

A sua Função de Transferência é dada por

$$\begin{aligned} H_{R,k}(z) &= G_k \frac{1}{(1 - e^{j\Omega_k} z^{-1})} \\ &= G_k \frac{z}{z - e^{j\Omega_k}}, \end{aligned}$$

a qual apresenta as seguintes características:

- Há um zero em $z = 0$.
- Há um pólo em $p_k = e^{j\Omega_k}$.
- Por definição, $\Omega_k = k\Omega_{M_T}$ e $\Omega_{M_T} = \left(\frac{2\pi}{M_T}\right)$.

Nitidamente, dois pontos merecem destaque: a estabilidade do sistema e os coeficientes da sua equação de diferença.

Uma vez que o pólo p_k tem módulo unitário, o sistema é do tipo marginalmente estável ou oscilador. Portanto, não faz sentido definir-se uma função Resposta em Frequência para ele. Uma aplicação típica para tal sistema é na implementação de osciladores, com frequência fundamental $\Omega_f = k\Omega_{M_T}$. Porém, utilizando a técnica de cancelamento pólo-zero, o mesmo também pode ser utilizado, em conjunto com o sistema *comb filter*, para implementar sistemas do tipo SLIT FIR com fase linear. Esse tipo de projeto é abordado a seguir.

O coeficiente $a_1 = -e^{jk\Omega_{M_T}}$ só assume valores reais para os casos onde $k = 0$ e, se M_T for par, onde $k = M_T/2$, quando ele passa a valer $a_1 = -1$ e $a_1 = 1$, respectivamente. Porém, pensando-se na composição de uma Função de Transferência por frações parciais, pode-se criar um sistema de segunda ordem com coeficientes reais, por meio da conexão paralela de dois *resonators*. Definindo-se $1 \leq k \leq (M_T/2) - 1$, para M_T par, $1 \leq k \leq (M_T - 1)/2$, para M_T ímpar, $G_k = Re(G_k) + jIm(G_k)$ e $G_{(M_T-k)} = G_k^*$, bem como observando-se que $e^{j(M_T-k)\Omega_{M_T}} = e^{-jk\Omega_{M_T}}$, pode-se definir a Função de Transferência do sistema *resonator* de segunda ordem como

$$\begin{aligned}
 H_{R2,k}(z) &= H_{R,k}(z) + H_{R,(M_T-k)}(z) \\
 &= \frac{G_k}{(1 - e^{jk\Omega_{M_T}} z^{-1})} + \frac{G_{(M_T-k)}}{(1 - e^{j(M_T-k)\Omega_{M_T}} z^{-1})} \\
 &= \frac{G_k}{(1 - e^{jk\Omega_{M_T}} z^{-1})} + \frac{G_k^*}{(1 - e^{-jk\Omega_{M_T}} z^{-1})} \\
 &= \frac{G_k (1 - e^{-jk\Omega_{M_T}} z^{-1}) + G_k^* (1 - e^{jk\Omega_{M_T}} z^{-1})}{(1 - e^{jk\Omega_{M_T}} z^{-1}) (1 - e^{-jk\Omega_{M_T}} z^{-1})} \\
 &= \frac{(G_k + G_k^*) - (G_k e^{-jk\Omega_{M_T}} + G_k^* e^{jk\Omega_{M_T}})}{1 - (e^{jk\Omega_{M_T}} + e^{-jk\Omega_{M_T}}) z^{-1} + z^{-2}} \\
 &= \frac{2 Re(G_k) - 2 Re(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}} .
 \end{aligned} \tag{8.3}$$

Consequentemente, a sua equação de diferença é dada por

$$\begin{aligned}
 y[n] &= -a_1 y[n-1] - a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] \\
 &= 2 \cos(k\Omega_{M_T}) y[n-1] - y[n-2] + 2 Re(G_k) x[n] - 2 Re(G_k e^{-jk\Omega_{M_T}}) x[n-1] .
 \end{aligned} \tag{8.4}$$

Generalizando-se a ideia de se compor uma Função de Transferência por meio de frações parciais ou, equivalentemente, por meio de *resonators* conectados em paralelo, obtém-se

$$\begin{aligned}
 H_R(z) &= H_{R,0}(z) + H_{R,\frac{M_T}{2}}(z) + \sum_{k=1}^{\frac{M_T}{2}-1} H_{R2,k}(z) \\
 &= \frac{G_0}{(1 - z^{-1})} + \frac{G_{M_T/2}}{(1 + z^{-1})} + \sum_{k=1}^{\frac{M_T}{2}-1} \frac{2 Re(G_k) - 2 Re(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}} ,
 \end{aligned} \tag{8.5}$$

para M_T par, e

$$\begin{aligned}
H_R(z) &= H_{R,0}(z) + \sum_{k=1}^{\frac{(M_T-1)}{2}} H_{R2,k}(z) \\
&= \frac{G_0}{(1-z^{-1})} + \sum_{k=1}^{\frac{(M_T-1)}{2}} \frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}}, \quad (8.6)
\end{aligned}$$

para M_T ímpar.

8.2.3 Conexão série de *comb filter* com *resonator*

Uma vez que, tanto os zeros do sistema *comb filter* quanto os pólos do sistema *resonator*, encontram-se ao longo do círculo de raio unitário, a conexão série de tais sistemas pode promover cancelamentos zero-pólo. Isso é tratado a seguir, considerando-se um *comb filter* de comprimento $L = M_T$.

Cancelamento simples em $k = 0$

Realizando-se uma conexão série de um *comb filter* com um *resonator*, em $k = 0$, promove-se um cancelamento zero-pólo em $\Omega_k = k\Omega_{M_T} = \Omega_0 = 0$ rad.

A Função de Transferência total é dada por

$$H(z) = H_C(z) H_{R,0}(z) = \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left(\frac{G_0}{1 - z^{-1}} \right),$$

que, no caso onde $G_0 = 1$, representa o Sistema de Média Móvel.

Por sua vez, a função Resposta em Frequência é

$$H(e^{j\Omega}) = \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left(\frac{G_0}{1 - e^{-j\Omega}} \right).$$

Calculando-se o valor de $H(e^{j\Omega})$ em $\Omega_0 = 0$ rad, encontra-se uma indeterminação do tipo $\left(\frac{0}{0}\right)$. Aplicando-se o Teorema “Regra de L’Hôpital”, obtém-se

$$\begin{aligned}
H(e^{j\Omega_0}) &= \left[\frac{1}{M_T} (-1) (-jM_T) e^{-j\Omega_0 M_T} \right] \left[\frac{G_0}{(-1)(-j)e^{-j\Omega_0}} \right] \\
&= \left[\frac{1}{M_T} (M_T) e^{-j\Omega_0 M_T} \right] \left(\frac{G_0}{e^{-j\Omega_0}} \right) \\
&= G_0.
\end{aligned}$$

Portanto, o cancelamento zero-pólo, em $\Omega_0 = 0$ rad, produz as seguintes consequências:

- Remoção do pólo simples, sobre o círculo de raio unitário, o que garante a estabilidade do sistema total.
- Remoção do zero simples, sobre o círculo de raio unitário, evitando um zero de transmissão $|H(e^{j\Omega_0})| = 0$ no sistema total.
- A inserção do ganho $|H(e^{j\Omega_0})| = |G_0|$ no sistema total.
- Os demais zeros não sofrem qualquer alteração.

Cancelamento simples em $k = M_T/2$

Realizando-se uma conexão série de um *comb filter*, de M_T par, com um *resonator*, em $k = M_T/2$, promove-se um cancelamento zero-pólo em $\Omega_k = k\Omega_{M_T} = \Omega_{M_T/2} = \pi$ rad.

A Função de Transferência total é dada por

$$H(z) = H_C(z) H_{R, \frac{M_T}{2}}(z) = \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left(\frac{G_{M_T/2}}{1 + z^{-1}} \right).$$

Por sua vez, a função Resposta em Frequência é

$$H(e^{j\Omega}) = \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left(\frac{G_{M_T/2}}{1 + e^{-j\Omega}} \right).$$

Calculando-se o valor de $H(e^{j\Omega})$ em $\Omega_{M_T/2} = \pi$ rad, onde M_T é par, encontra-se uma indeterminação do tipo $\left(\frac{0}{0}\right)$. Aplicando-se o Teorema “Regra de L’Hôpital”, obtém-se

$$\begin{aligned} H(e^{j\Omega_{M_T/2}}) &= \left[\frac{1}{M_T} (-1) (-jM_T) e^{-j\Omega_{M_T/2} M_T} \right] \left[\frac{G_{M_T/2}}{(-j)e^{-j\Omega_{M_T/2}}} \right] \\ &= \left[\frac{1}{M_T} (-M_T) e^{-j\Omega_{M_T/2} M_T} \right] \left(\frac{G_{M_T/2}}{e^{-j\Omega_{M_T/2}}} \right) \\ &= G_{M_T/2}. \end{aligned}$$

Portanto, o cancelamento zero-pólo, em $\Omega_{M_T/2} = \pi$ rad, produz as seguintes consequências:

- Remoção do pólo simples, sobre o círculo de raio unitário, o que garante a estabilidade do sistema total.
- Remoção do zero simples, sobre o círculo de raio unitário, evitando um zero de transmissão $|H(e^{j\Omega_{M_T/2}})| = 0$ no sistema total.
- A inserção do ganho $|H(e^{j\Omega_{M_T/2}})| = |G_{M_T/2}|$ no sistema total.
- Os demais zeros não sofrem qualquer alteração.

Cancelamento duplo em $k \neq 0$ e $k \neq M_T/2$

Realizando-se uma conexão série de um *comb filter* com dois *resonators*, nos valores genéricos k e $(M_T - k)$, promovem-se dois cancelamentos zero-pólo, um em $\Omega_k = k\Omega_{M_T}$ rad e outro $\Omega_{(M_T-k)} = (M_T - k)\Omega_{M_T}$ rad. Nesse caso, tem-se que $1 \leq k \leq (M_T/2) - 1$, para M_T par, e $1 \leq k \leq (M_T - 1)/2$, para M_T ímpar.

A Função de Transferência total é dada por

$$\begin{aligned} H(z) &= H_C(z) H_{R2,k}(z) \\ &= H_C(z) [H_{R,k}(z) + H_{R,(M_T-k)}(z)] \\ &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left[\frac{G_k}{(1 - e^{jk\Omega_{M_T}} z^{-1})} + \frac{G_{(M_T-k)}}{(1 - e^{j(M_T-k)\Omega_{M_T}} z^{-1})} \right] \\ &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left[\frac{G_k}{(1 - e^{jk\Omega_{M_T}} z^{-1})} + \frac{G_k^*}{(1 - e^{-jk\Omega_{M_T}} z^{-1})} \right] \\ &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \left[\frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}} \right]. \end{aligned}$$

Por sua vez, a função Resposta em Frequência é

$$\begin{aligned} H(e^{j\Omega}) &= \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left[\frac{G_k}{(1 - e^{jk\Omega M_T} e^{-j\Omega})} + \frac{G_k^*}{(1 - e^{-jk\Omega M_T} e^{-j\Omega})} \right] \\ &= \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left[\frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega M_T}) e^{-j\Omega}}{1 - 2 \cos(k\Omega M_T) e^{-j\Omega} + e^{-j2\Omega}} \right]. \end{aligned}$$

Calculando-se o valor de $H(e^{j\Omega})$ em $\Omega_k = k\Omega_{M_T}$ rad, obtém-se $H(e^{j\Omega_k}) = 0$ na combinação do *comb filter* com o *resonator* $\{R, (M_T - k)\}$, bem como encontra-se uma indeterminação do tipo $\left(\frac{0}{0}\right)$, na combinação do *comb filter* com o *resonator* $\{R, k\}$. Aplicando-se o Teorema “Regra de L’Hôpital”, obtém-se

$$\begin{aligned} H(e^{j\Omega_k}) &= \left[\frac{1}{M_T} (-1) (-jM_T) e^{-j\Omega_k M_T} \right] \left[\frac{G_k}{(-1)e^{jk\Omega_{M_T}} (-j)e^{-j\Omega_k}} \right] \\ &= \left[\frac{1}{M_T} (M_T) e^{-j\Omega_k M_T} \right] \left(\frac{G_k}{e^{jk\Omega_{M_T}} e^{-j\Omega_k}} \right) \\ &= G_k. \end{aligned}$$

Por sua vez, calculando-se o valor de $H(e^{j\Omega})$ em $\Omega_{(M_T-k)} = (M_T - k)\Omega_{M_T} = -k\Omega_{M_T}$ rad, obtém-se $H(e^{j\Omega_{(M_T-k)}}) = 0$ na combinação do *comb filter* com o *resonator* $\{R, k\}$, bem como encontra-se uma indeterminação do tipo $\left(\frac{0}{0}\right)$, na combinação do *comb filter* com o *resonator* $\{R, (M_T - k)\}$. Aplicando-se o Teorema “Regra de L’Hôpital”, obtém-se

$$\begin{aligned} H(e^{j\Omega_{(M_T-k)}}) &= \left[\frac{1}{M_T} (-1) (-jM_T) e^{-j\Omega_{(M_T-k)} M_T} \right] \left[\frac{G_k^*}{(-1)e^{-jk\Omega_{M_T}} (-j)e^{-j\Omega_{(M_T-k)}}} \right] \\ &= \left[\frac{1}{M_T} (M_T) e^{-j\Omega_{(M_T-k)} M_T} \right] \left(\frac{G_k^*}{e^{-jk\Omega_{M_T}} e^{-j\Omega_{(M_T-k)}}} \right) \\ &= G_k^*. \end{aligned}$$

Portanto, os dois cancelamentos zero-pólo, em Ω_k rad e em $\Omega_{(M_T-k)}$ rad, produzem as seguintes consequências:

- Remoção do par de pólos complexos conjugados, sobre o círculo de raio unitário, o que garante a estabilidade do sistema total.
- Remoção do par de zeros complexos conjugados, sobre o círculo de raio unitário, evitando os zeros de transmissão $|H(e^{j\Omega_k})| = 0$ e $|H(e^{j\Omega_{(M_T-k)}})| = 0$ no sistema total.
- A inserção dos ganhos $|H(e^{j\Omega_k})| = |G_k|$ e $|H(e^{j\Omega_{(M_T-k)}})| = |G_k^*| = |G_k|$ no sistema total.
- Os demais zeros não sofrem qualquer alteração.

Cancelamento global

Realizando-se uma conexão série de um *comb filter* com um somatório de *resonators* $\{R, k\}$, para $0 \leq k \leq (M_T - 1)$, promove-se um cancelamento zero-pólo para cada zero do *comb filter*.

A Função de Transferência total é dada por

$$\begin{aligned}
 H(z) &= H_C(z) \left[H_{R,0}(z) + H_{R,\frac{M_T}{2}}(z) + \sum_{k=1}^{\frac{M_T}{2}-1} H_{R2,k}(z) \right] \\
 &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \\
 &\quad \left[\frac{G_0}{(1 - z^{-1})} + \frac{G_{M_T/2}}{(1 + z^{-1})} + \sum_{k=1}^{\frac{M_T}{2}-1} \frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}} \right], \quad (8.7)
 \end{aligned}$$

para M_T par, e

$$\begin{aligned}
 H(z) &= H_C(z) \left[H_{R,0}(z) + \sum_{k=1}^{\frac{(M_T-1)}{2}} H_{R2,k}(z) \right] \\
 &= \left[\frac{1}{M_T} (1 - z^{-M_T}) \right] \\
 &\quad \left[\frac{G_0}{(1 - z^{-1})} + \sum_{k=1}^{\frac{(M_T-1)}{2}} \frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega_{M_T}}) z^{-1}}{1 - 2 \cos(k\Omega_{M_T}) z^{-1} + z^{-2}} \right], \quad (8.8)
 \end{aligned}$$

para M_T ímpar.

Portanto, cada cancelamento zero-pólo, em Ω_k rad, produz as seguintes consequências:

- Remoção do pólo simples, sobre o círculo de raio unitário, o que garante a estabilidade do sistema total.
- Remoção do zero simples, sobre o círculo de raio unitário, evitando um zero de transmissão $|H(e^{j\Omega_k})| = 0$ no sistema total.
- A inserção do ganho $|H(e^{j\Omega_k})| = |G_0|$ no sistema total.

8.3 Projeto de filtros FIR por amostragem na frequência

Uma técnica de projeto simples, para filtros do tipo FIR, pode ser definida a partir da ideia de se realizar uma conexão série de um *comb filter* com um somatório de *resonators* $\{R, k\}$, para $0 \leq k \leq (M_T - 1)$, promovendo-se um cancelamento zero-pólo para cada zero do *comb filter* e inserindo-se um determinado ganho G_k na frequência do cancelamento. Para tal, os valores da função Resposta em Frequência desejada $H_D(e^{j\Omega})$ são calculados nos pontos de cancelamento zero-pólo e são utilizados no lugar dos ganhos G_k . Nesse sentido, definindo-se $\Omega_{M_T} = \left(\frac{2\pi}{M_T}\right)$ e $\Omega_k = k\Omega_{M_T}$, para $0 \leq k \leq (M_T - 1)$, deve-se calcular os ganhos

$$G_k = H_D(e^{j\Omega_k}) = H_D(e^{jk\Omega_{M_T}})$$

e, a partir deles, formar os respectivos *resonators*, os quais são resumidos a seguir.

Em função do mecanismo utilizado, essa técnica é denominada de projeto de filtros FIR por amostragem na frequência.

8.3.1 Projeto de filtro de ordem 1

Para $\Omega_k = k\Omega_{M_T} = \Omega_0 = 0$ rad, realiza-se uma conexão série de um *comb filter* com um *resonator* $\{R, k\}$, em $k = 0$, promovendo-se um cancelamento zero-pólo na frequência Ω_0 .

A função Resposta em Frequência é dada por

$$H(e^{j\Omega}) = \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left(\frac{G_0}{1 - e^{-j\Omega}} \right),$$

onde

$$G_0 = H_D(e^{j\Omega_0}).$$

Para $\Omega_k = k\Omega_{M_T} = \Omega_{M_T/2} = \pi$ rad, realiza-se uma conexão série de um *comb filter* com um *resonator* $\{R, k\}$, em $k = M_T/2$, promovendo-se um cancelamento zero-pólo na frequência $\Omega_{M_T/2}$.

A função Resposta em Frequência é dada por

$$H(e^{j\Omega}) = \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left(\frac{G_{M_T/2}}{1 + e^{-j\Omega}} \right),$$

onde

$$G_{M_T/2} = H_D(e^{j\Omega_{M_T/2}}).$$

8.3.2 Projeto de filtro de ordem 2

Para $\Omega_k = k\Omega_{M_T}$ rad e $\Omega_{(M_T-k)} = (M_T - k)\Omega_{M_T}$ rad, realiza-se uma conexão série de um *comb filter* com dois *resonators*, nos valores genéricos k e $(M_T - k)$, promovendo-se dois cancelamentos zero-pólo, um em $\Omega_k = k\Omega_{M_T}$ rad e outro $\Omega_{(M_T-k)} = (M_T - k)\Omega_{M_T}$ rad. Nesse caso, tem-se que $1 \leq k \leq (M_T/2) - 1$, para M_T par, e $1 \leq k \leq (M_T - 1)/2$, para M_T ímpar.

A função Resposta em Frequência é dada por

$$\begin{aligned} H(e^{j\Omega}) &= \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left[\frac{G_k}{(1 - e^{jk\Omega_{M_T}} e^{-j\Omega})} + \frac{G_k^*}{(1 - e^{-jk\Omega_{M_T}} e^{-j\Omega})} \right] \\ &= \left[\frac{1}{M_T} (1 - e^{-j\Omega M_T}) \right] \left[\frac{2 \operatorname{Re}(G_k) - 2 \operatorname{Re}(G_k e^{-jk\Omega_{M_T}}) e^{-j\Omega}}{1 - 2 \cos(k\Omega_{M_T}) e^{-j\Omega} + e^{-j2\Omega}} \right], \end{aligned}$$

onde

$$G_k = H_D(e^{j\Omega_k})$$

e

$$G_k^* = H_D^*(e^{j\Omega_k}) = H_D(e^{-j\Omega_k}).$$

8.3.3 Projeto de filtro de ordem M_T

Realizando-se uma conexão série de um *comb filter* com um somatório de *resonators* $\{R, k\}$, para $0 \leq k \leq (M_T - 1)$, promove-se um cancelamento zero-pólo para cada zero do *comb filter*.

A Função de Transferência total é dada por (8.7), para M_T par, e por (8.8), para M_T ímpar, onde

$$G_k = H_D(e^{j\Omega_k}).$$

Parte IV

Prática


```

%
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% define os limites do SMM
M_p = 4; % passado
M_f = 3; % futuro
M_t = (M_p + M_f + 1); % total

% define a entrada, em funcao dos limites do SMM
x = [ (1:(M_t - 1)), M_t, ((M_t - 1):(-1):1) ];
x_len = length(x);

% define o comprimento da saida
y_len = (x_len + M_t - 1);

% define a matriz com vetores de entrada deslocados
m_xd = zeros(M_t,y_len);
%
for k=1:M_t
    m_xd (k , [(1:x_len)+k-1]) = x;
end

% calcula a soma das entradas deslocadas
sum_xd = sum(m_xd);

% calcula a saida = soma escalada das entradas deslocadas
y = sum_xd / M_t;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% define parametros para os graficos
%

% define variavel de numeracao dos graficos
Fignbr = 0;

% define valores do indice da saida
ny = 0:(y_len - 1);
n = (ny-M_f);

% define limites de visualizacao
n_min = min(n);
n_max = max(n);

% define limites de visualizacao

```

```

x_max      = max(m_xd((M_f+1),:));
sum_xd_max = max(sum_xd);

%%%%%%%%%%%% %%%%%%%%%%%%%

% grafico do sinal de entrada

%
Fignbr = Fignbr + 1;
figure(Fignbr)
%
stem(n,m_xd((M_f+1),:),'k')
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = x_max;
axis(vax)
%
title('Sinal de entrada')
ylabel('x [n]')
xlabel('n')

%%%%%%%%%%%% %%%%%%%%%%%%%

% grafico dos sinais deslocados

%
Fignbr = Fignbr + 1;
figure(Fignbr)
%
for k=1:M_t
    %
    subplot(M_t,1,k)
    stem(n,m_xd(k,:),'k')
    %
    vax = axis;
    vax(1) = n_min;
    vax(2) = n_max;
    vax(4) = x_max;
    axis(vax)
    %
    if (k==1) title('Deslocamentos do sinal de entrada'); endif
    %
    lbl_str = ['x [n - (', int2str(-M_f + k-1), ')]]'];
    ylabel(lbl_str)
end

%%%%%%%%%%%% %%%%%%%%%%%%%

```

```

% grafico da soma das entradas deslocadas

%
Fignbr = Fignbr + 1;
figure(Fignbr)
%
stem(n,sum_xd,'b')
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = sum_xd_max;
axis(vax)
%
title('Soma das entradas deslocadas')
ylabel('x [n+Mf] + ... + x [n-Mp]')
xlabel('n')

%%%%%%%%%%%% %%%%%%%%%%%%%%

% grafico da saida
% (soma escalada das entradas deslocadas)

%
Fignbr = Fignbr + 1;
figure(Fignbr)
%
stem(n,y,'r')
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = x_max;
axis(vax)
%
tit_str = ...
    cellstr(
        ['Resposta do sistema';
         '(escalamento da soma das entradas deslocadas)']
    )
title(tit_str)
ylabel('y [n]')
xlabel('n')

%%%%%%%%%%%% %%%%%%%%%%%%%%

% graficos da entrada e da saida
% separados

%
Fignbr = Fignbr + 1;

```

```

figure(Fignbr)
%
subplot(2,1,1)
stem(n,m_xd((M_f+1),:),'k')
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = x_max;
axis(vax)
%
title('Sinal de entrada')
ylabel('x [n]')

%
subplot(2,1,2)
stem(n,y,'r')
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = x_max;
axis(vax)
%
title('Resposta do sistema')
ylabel('y [n]')
xlabel('n')

%%%%%%%%%%%% %%%%%%%%%%%%%

% graficos da entrada e da saida
% sobrepostos

%
Fignbr = Fignbr + 1;
figure(Fignbr)
%
stem(n,m_xd((M_f+1),:),'kx')
hold on
stem(n,y,'ro')
%
vax = axis;
vax(1) = n_min;
vax(2) = n_max;
vax(4) = x_max;
axis(vax)
%
title('Sinal de entrada x [n] e resposta do sistema y [n]')
ylabel('x [n] e y [n]')
xlabel('n')
%
```

```
legend('x [n]', 'y [n]')
```

```
%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
```

```
%
% EOF
%
```

9.2.2 Razão de funções senoidais com abscissa escalada

Abaixo, é apresentado um código para Octave, que calcula e exibe, para valores par e ímpar de M_T , as curvas de funções senoidais com abscissa escalada, bem como a sua razão, que representam seqüências constituintes da função Resposta em Frequência do SMM.

```
%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
```

```
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Sequencias constituintes
% da funcao Resposta em Frequencia,
% para valores par e impar de Mt.
%
% - Sequencias:
% sin(Mt W/2) , sin(W/2) e sin(Mt W/2) / sin (W/2).
%
% - Funcao Resposta em Frequencia:
% Diric(Mt,W) = (1/Mt) sin(Mt W/2) / sin (W/2).
%
%
% Versao: 2018_10_15.
%
% Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
```

```
%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
```

```
%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
```

```
% limpa o ambiente de trabalho
%
clear all
close all
%
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define os valores de Omega
Nw = 360;
%
Wstep = (2*pi/Nw);
%
Wmin = Wstep;
%Wmin = -2*pi;
%Wmax = 2*pi;
Wmax = 4*pi;
%
W = [ Wmin:Wstep:(-Wstep) , (Wstep):Wstep:Wmax ];

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define Mt
Mt_odd = 5;
Mt_even = (Mt_odd + 1);

% calcula a partes da RF
%
% sinal basico
f = sin(W);
%
% denominador
fden = sin(W/2);
%
% numerador
fnum_odd = sin(Mt_odd * W / 2);
fnum_even = sin(Mt_even * W / 2);
%
% razoes
ratio_odd = (fnum_odd ./ fden);
ratio_even = (fnum_even ./ fden);

% calcula a RF
%H_odd = ratio_odd / Mt_odd ;
%H_even = ratio_even / Mt_even;

% calcula modulo e fase da RF
%
%H_odd_mod = abs(H_odd);
%H_odd_arg = angle(H_odd)*180/pi;
%
%H_even_mod = abs(H_even);

```

```

%H_even_arg = angle(H_even)*180/pi;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% desenha graficos
%

% parametros basicos
%
FigNbr = 0;
%
Wnorm = W/pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos isolados
%
% sequencia basica e numerador
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
%
subplot(3,1,1)
plot(Wnorm,f)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (\Omega)')
title(cellstr(['H(e^{j\Omega}) do Sistema de Média Móvel';
               'Sinal básico'])))

%
subplot(3,1,2)
plot(Wnorm,fnum_odd)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (M_T \Omega/2)')
title(cellstr(['Numerador de H(e^{j\Omega})']));

```

```

                'M_T = ' , num2str(Mt_odd]))
%
subplot(3,1,3)
plot(Wnorm,fnum_even)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (M_T \Omega/2)')
title(cellstr(['Numerador de H(e^{j\Omega})';
                'M_T = ' , num2str(Mt_even)]))
%
xlabel('\Omega_{norm}')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos isolados
%
% sequencia basica e denominador
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
%
subplot(3,1,1)
plot(Wnorm,f)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (\Omega)')
title(cellstr(['H(e^{j\Omega}) do Sistema de Média Móvel';
                'Sinal básico']))
%
subplot(3,1,2)
plot(Wnorm,fden)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (\Omega/2)')
title('Denominador de H(e^{j\Omega})')
%
subplot(3,1,3)
plot(Wnorm,1./fden)
v=axis;

```

```

v(3)= -10;
v(4)= 10;
axis(v)
ylabel('1 / sin (\Omega/2)')
title('Denominador de H(e^{j\Omega})')
%
xlabel('\Omega_{norm}')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos isolados
%
% numerador, denominador e razao num/den
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
%
subplot(4,2,1)
plot(Wnorm,fnum_odd)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (M_T \Omega/2)')
title(cellstr(['Sistema de Média Móvel (M_T = ' , num2str(Mt_odd) , ')';
               'Numerador de H(e^{j\Omega})'])))

%
subplot(4,2,3)
plot(Wnorm,fden)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (\Omega/2)')
title('Denominador de H(e^{j\Omega})')

%
subplot(4,2,5)
plot(Wnorm,ratio_odd)
v = axis;
v(3) = -Mt_odd;
v(4) = Mt_odd;
axis(v)
ylabel('M_T * H(e^{j\Omega})')
title('Num(e^{j\Omega}) / Den(e^{j\Omega})')
%
```

```

subplot(4,2,7)
plot(Wnorm,abs(ratio_odd))
v = axis;
v(3) = 0;
v(4) = Mt_odd;
axis(v)
ylabel('M_T * | H(e^{j\Omega}) |')
title('| Num(e^{j\Omega}) / Den(e^{j\Omega}) |')
%
xlabel('\Omega_{norm}')

%
%
subplot(4,2,2)
plot(Wnorm,fnum_even)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (M_T \Omega/2)')
title(cellstr(['Sistema de Média Móvel (M_T = ', num2str(Mt_even) , ')';
              'Numerador de H(e^{j\Omega})']))
%
subplot(4,2,4)
plot(Wnorm,fden)
v=axis;
v(3)= -1;
v(4)= 1;
axis(v)
ylabel('sin (\Omega/2)')
title('Denominador de H(e^{j\Omega})')
%
subplot(4,2,6)
plot(Wnorm,ratio_even)
v = axis;
v(3) = -Mt_even;
v(4) = Mt_even;
axis(v)
ylabel('M_T * H(e^{j\Omega})')
title('Num(e^{j\Omega}) / Den(e^{j\Omega})')
%
subplot(4,2,8)
plot(Wnorm,abs(ratio_even))
v = axis;
v(3) = 0;
v(4) = Mt_even;
axis(v)
ylabel('M_T * | H(e^{j\Omega}) |')
title('| Num(e^{j\Omega}) / Den(e^{j\Omega}) |')

```

```
%
xlabel('\Omega_{norm}')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
% EOF
%
```

9.3 Diagrama de Pólos e Zeros (DPZ)

9.3.1 Função para geração de DPZ (com marcação de singularidades múltiplas)

Abaixo, é apresentado um código para Octave, que desenha um diagrama de pólos e zeros (DPZ), indicando a multiplicidade das singularidades.

```
function ASV_my_dpz_tracer(k,z,p,sing_dim)
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
%   ASV_my_dpz_tracer(k,z,p,sing_dim)
%
%   Recebe : escalar k, vetor z e vetor p (obrigatorios).
%           escalar sing_dim (opcional).
%
%   Retorna: desenha um diagrama de polos e zeros (DPZ),
%           indicando a multiplicidade das singularidades.
%
%   Versao: 2018_10_08.
%
%   Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% gerencia as singularidades simples e multiplas
%
%   zeros
%
%   seleciona os diferentes valores de zeros
z_unq = unique(z);
%
%   separa os zeros simples dos multiplos
```

```

% e
% identifica as multiplicidades
%
z_smp = [];
z_mlt = [];
z_mlt_m = [];
%
for ind = 1:length(z_unq)
    unq_pos = find( z == z_unq(ind) );
    m = length(unq_pos);
    if ( m == 1 )
        z_smp = [ z_smp ; z_unq(ind) ];
    else
        z_mlt = [ z_mlt ; z_unq(ind) ];
        z_mlt_m = [ z_mlt_m ; m ];
    endif
endfor
%
clear unq_pos m

% polos
%
% seleciona os diferentes valores de polos
p_unq = unique(p);

% separa os polos simples dos multiplos
% e
% identifica as multiplicidades
%
p_smp = [];
p_mlt = [];
p_mlt_m = [];
%
for ind = 1:length(p_unq)
    unq_pos = find( p == p_unq(ind) );
    m = length(unq_pos);
    if ( m == 1 )
        p_smp = [ p_smp ; p_unq(ind) ];
    else
        p_mlt = [ p_mlt ; p_unq(ind) ];
        p_mlt_m = [ p_mlt_m ; m ];
    endif
endfor
%
clear unq_pos m

%
%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
```

```

%
% cria grafico
%

% calculos comuns a todos os graficos
%
re_z_unq = real(z_unq);
im_z_unq = imag(z_unq);
%
re_p_unq = real(p_unq);
im_p_unq = imag(p_unq);
%
mult_offset_x = .025;
mult_offset_y = .050;

% define valor default
% da dimensao das singularidades
if (!exist ('sing_dim', 'var')) sing_dim = 8; endif

% calcula raio de visualizacao,
% baseado na posicao das singularidades (zeros e polos)
%
if ( isempty(z) && isempty(p) )
    dpz_limit = 1;
else
    max_re_sing = max( [max(abs(re_z_unq)) , max(abs(re_p_unq))] );
    max_im_sing = max( [max(abs(im_z_unq)) , max(abs(im_p_unq))] );
    max_sing    = max( [max_re_sing    , max_im_sing    ] );
    dpz_limit = fix(max_sing) + 1;
endif

% cria canvas
%
% define numero da figura
FigNbr = get(0,'CurrentFigure');
if(isempty(FigNbr))
    FigNbr = 1;
else
    FigNbr = FigNbr + 1;
end
%
hf=figure(FigNbr);
%
set(hf,'papertype','a4')

% define o circulo de raio unitario
N = 360;
Omega = 2*pi/N;

```

```

n = 0:N;
unit_circ = exp(j*Omega*n);

% desenha circulo de raio unitario
plot(real(unit_circ),imag(unit_circ),'b')

% mantem todos os graficos na mesma figura
hold on

% desenha os zeros simples
plot(real(z_smp),imag(z_smp),'ok','MarkerSize',sing_dim)

% desenha os zeros multiplos
plot(real(z_mlt),imag(z_mlt),'ok','MarkerSize',sing_dim)
for ind = 1:length(z_mlt_m)
    text( (real(z_mlt(ind)) + mult_offset_x) ,
          (imag(z_mlt(ind)) + mult_offset_y) ,
          num2str(z_mlt_m(ind))
          )
endfor

% desenha os polos simples
plot(real(p_smp),imag(p_smp),'xk','MarkerSize',sing_dim)

% desenha os polos multiplos
plot(real(p_mlt),imag(p_mlt),'xk','MarkerSize',sing_dim)
for ind = 1:length(p_mlt_m)
    text( (real(p_mlt(ind)) + mult_offset_x) ,
          (imag(p_mlt(ind)) + mult_offset_y) ,
          num2str(p_mlt_m(ind))
          )
endfor

% escreve constante de ganho
% em posicao definida pelos limites do grafico
K_str = ['K = ', sprintf("%.3f",k)];
K_str_offset = dpz_limit / 8;
text( ((-dpz_limit) + K_str_offset) ,
      (( dpz_limit) - K_str_offset) ,
      K_str
      )

% identifica o grafico
title('Diagrama de polos e zeros (DPZ)')
ylabel('Im\{.\}')
xlabel('Re\{.\}')

% habilita o grid na figura
grid on

% Controla formato do grafico
% Deve vir ANTES do controle dos limites !!!

```



```

M_p = 6; % passado
M_f = 0; % futuro
M_t = (M_p + M_f + 1); % total

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

% define o numerador do operador de transferencia
num = ones (1,M_t);
% define o denominador do operador de transferencia
den = zeros(1,M_t);
den(1) = 1;

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

% calcula o conjunto ZPK
z = roots(num);
p = roots(den);
k = 1 / M_t;

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

% invoca funcao que desenha um DPZ
ASV_my_dpz_tracer(k,z,p);

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

%
% EOF
%
```

9.4 Respostas de um SLIT no domínio do tempo

9.4.1 Equação recursiva, resposta à entrada e resposta ao estado

Abaixo, é apresentado um código para Octave, que calcula a resposta do SMM, a partir da sua equação recursiva, utilizando a função *filter()*. São calculadas as seguintes respostas: resposta à entrada, resposta ao estado (estado nulo) e resposta ao estado (estado não nulo, definido pelo usuário).

```

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Equacao recursiva.
%
% - Calculo da resposta usando a funcao filter().
```

```

%
%   - Calculo da resposta aa entrada.
%
%   - Calculo da resposta ao estado,
%       para estado nao nulo e definido pelo usuario.
%
%
%   Versao: 2018_11_12.
%
%   Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
% limpa o ambiente de trabalho
%
clear all
close all

%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
%
% obtem estado inicial a partir do usuario
fflush (stdout);
xi = input('Vetor de estados ([x0, x1, ... , xMt]) = ');

%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
%
% define comprimento do SMM
Mt = length(xi);
%Mt = 5;

%
% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
%
%
% define coeficientes ak (den) e bk (num)
% da equacao recursiva do SMM
num = zeros(1,(Mt+1));
den = num;
%
num(1) = 1/Mt;

```

```

num(end) = -1/Mt;
%
den(1) = 1;
den(2) = -1;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define quantidade de amostras
% para simulacao do SMM
NOIS = 3*Mt; % Nbr Of Inp Smpls
n = 0:(NOIS-1);

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define entrada nula
r_null = zeros(1,NOIS);

% define entrada
r = r_null;
r(1) = 1;
%r = r+1;
%r = n;

% define estado inicial nulo
xi_null = zeros(1,(length(den)-1));

% define estado inicial
%xi = xi_null;
%xi = xi+1;
%xi(1) = 1;
%xi(end) = 1;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% calcula saida e estado final
% para estado nulo (yr) e entrada nula (yx)
%
[yr,xfr] = filter(num,den,r,xi_null);
%
[yxx_null,xfxx_null] = filter(num,den,r_null,xi_null);
%
[yx,xfx] = filter(num,den,r_null,xi);

```

```

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%
figure(1)
%
subplot(3,1,1)
stem(n,yr)
title(cellstr([
'SMM com Equação Recursiva e Estrutura em Cascata (M_T = ',num2str(Mt),')'];
'Resposta à entrada']))
%title('Resposta à entrada do SMM')
ylabel('y_{ent} [n]')
v = axis;
v(2) = n(end);
axis(v);
%
subplot(3,1,2)
stem(n,yxx_null)
title('Resposta ao estado (estado nulo)')
ylabel('y_{est} [n]')
v = axis;
v(2) = n(end);
axis(v);
%
subplot(3,1,3)
stem(n,yx)
title('Resposta ao estado (estado definido pelo usuário)')
ylabel('y_{est} [n]')
xlabel('n')
v = axis;
v(2) = n(end);
axis(v);

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%
% EOF
%
```

9.4.2 Equação recursiva, estrutura em cascata e resposta ao impulso

Abaixo, é apresentado um código para Octave, que calcula a resposta ao impulso do SMM, a partir da sua equação recursiva, considerando uma estrutura em cascata de um escalador, um FIR e um acumulador. São calculadas as seguintes respostas: resposta na saída do escalador, resposta na saída do FIR, e resposta na saída do acumulador.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Equacao recursiva.
%
% - Estrutura em cascata: escalador, FIR e acumulador.
%
% - Calculo da resposta ao impulso.
%
%
% Versao: 2018_11_12.
%
% Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% limpa o ambiente de trabalho
%
clear all
close all

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% obtem comprimento do SMM a partir do usuario
fflush (stdout);
Mt = input('Comprimento Mt do SMM = ');
%Mt = 5;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% SMM definido por equacao recursiva,
% com uma estrutura em cascata,
% composta por: escalador, FIR, acumulador.
%

```

```

% define valor do escalador
Ksmm = 1/Mt;

% define coeficientes ak (den) e bk (num)
% do FIR
numFIR = zeros(1,(Mt+1));
denFIR = numFIR;
%
numFIR(1) = 1/Mt;
numFIR(end) = -1/Mt;
%
denFIR(1) = 1;

% define coeficientes ak (den) e bk (num)
% do acumulador
numACC = [1, 0];
denACC = [1, -1];

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define quantidade de amostras
% para simulacao do SMM
NOIS = 2*Mt; % Nbr Of Inp Smpls
n = 0:(NOIS-1);

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% define entrada nula
r_null = zeros(1,NOIS);

% define entrada
r = r_null;
r(1) = 1;
%r = r+1;
%r = n;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% calcula saida
% do escalador
%
y_esc = Ksmm * r;

```



```
%
% EOF
%
```

9.5 Função Resposta em Frequência

9.5.1 Curvas do SMM para diferentes valores de M_T

Abaixo, é apresentado um código para Octave, que calcula e exibe, para diferentes valores de M_T , as curvas de módulo e de argumento da função Resposta em Frequência do SMM.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Curvas de modulo e de argumento,
% da funcao Resposta em Frequencia,
% para diferentes valores de Mt.
%
% Versao: 2018_10_10.
%
% Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
% limpa o ambiente de trabalho
%
clear all
close all
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
%
% define os valores de Omega
Nw = 360;
Wstep = (2*pi/Nw);
W = 0:Wstep:pi;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%% %%%%%%%%%
```

```

%

% inicializa os vetores de valores da RF
H3 = zeros(size(W));
H2 = H3;
H1 = H3;

% define Mt
N3 = 20;
N2 = 10;
N1 = 5;

% calcula a RF
for k = 0:(N3-1)
    H3 = H3 + (exp(-j*k*W));
    if(k<N2)
        H2 = H2 + (exp(-j*k*W));
    endif
    if(k<N1)
        H1 = H1 + (exp(-j*k*W));
    endif
end

% normaliza a RF
H3 = H3/N3;
H2 = H2/N2;
H1 = H1/N1;

% calcula modulo e fase da RF
%
H3mod = abs(H3);
H3arg = angle(H3)*180/pi;
%
H2mod = abs(H2);
H2arg = angle(H2)*180/pi;
%
H1mod = abs(H1);
H1arg = angle(H1)*180/pi;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% parametros basicos
FigNbr = 0;
Wnorm = W/pi;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos isolados
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
%
subplot(2,3,1)
plot(Wnorm,H1mod)
ylabel('| H(e^{j\Omega}) |')
title(cellstr(['Sistema de Média Móvel';
               'M_T = ' , num2str(N1)]))
%
subplot(2,3,2)
plot(Wnorm,H2mod)
ylabel('| H(e^{j\Omega}) |')
title(cellstr(['Sistema de Média Móvel';
               'M_T = ' , num2str(N2)]))
%
subplot(2,3,3)
plot(Wnorm,H3mod)
ylabel('| H(e^{j\Omega}) |')
title(cellstr(['Sistema de Média Móvel';
               'M_T = ' , num2str(N3)]))
%
%
subplot(2,3,4)
plot(Wnorm,H1arg)
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')
%
subplot(2,3,5)
plot(Wnorm,H2arg)
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')
%
subplot(2,3,6)

```


9.5.2 DPZ, módulo e ângulo de fase, da Resposta em Frequência do SMM, para diferentes valores de M_T

Abaixo, é apresentado um código para Octave, que calcula e exhibe, para diferentes valores de M_T , os Diagramas de Pólos e Zeros (DPZ), bem como as curvas de módulo e de ângulo de fase, da função Resposta em Frequência do SMM.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
%
% Sistema de Media Movei (SMM):
%
% - Mt = 2 a 10.
%
% - Funcao Resposta em Frequencia:
%   H (e^jW) = (1/Mt) Sum_k_0_(Mt-1) exp(-j k W).
%
% - Diagrama de Polos e Zeros.
%
% - Modulo da funcao Resposta em Frequencia.
%   - Indicacao do valor maximo do 2o lobo.
%
% - Angulo de fase da funcao Resposta em Frequencia.
%
%
% Versao: 2018_10_23.
%
% Autor : Alexandre Santos de la Vega.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% limpa o ambiente de trabalho
%
clear all
close all

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% define os valores de Omega
%
Nw = 360;

```

```

%
Wstep = (2*pi/Nw);
%
Wmin = Wstep;
%Wmin = -2*pi;
%
Wmax = 2*pi;
%Wmax = 4*pi;
%
if (Wmin < 0)
    Wneg = Wmin:Wstep:(-Wstep);
else
    Wneg = [];
end
%
%
if (Wmax > 0)
    Wpos = (Wstep):Wstep:Wmax;
else
    Wpos = [];
end
%
W = [ Wneg , Wpos ];

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% define o circulo de raio unitario
unit_circ = exp(j*W);

% define Mt
Mt = 2:10;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% parametros comuns a todos os graficos
%
FigNbr = 0;
%
Wnorm = W/pi;
%
dpz_limit = 1.4;
%
sing_dim = 8;
mult_offset_x = .200;
mult_offset_y = .200;

```

```

% cria canvas 1
FigNbr = FigNbr + 1;
hf=figure(FigNbr);
set(hf,'papertype','a4')
%
% Apenas para induzir as dimensoes finais
% dos subplots da Fig 1 ...
subplot(3,3,9)

% cria canvas 2
FigNbr = FigNbr + 1;
hf=figure(FigNbr);
set(hf,'papertype','a4')

% cria canvas 3
FigNbr = FigNbr + 1;
hf=figure(FigNbr);
set(hf,'papertype','a4')

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%
% loop de calculos e graficos
%

% parametros de controle dos graficos
%
g_max = length(Mt);
%
col_max = 3;
line_max = fix(g_max / col_max);

%
for g = 1:g_max

%
% calcula funcoes
%

%
k = 1:(Mt(g)-1);
z_smm = exp(j*k*2*pi/Mt(g));

% calcula a RF
Hw = zeros(size(W));
for k = 0:(Mt(g)-1)

```

```

    Hw = Hw + (exp(-j*k*W));
end
%
Hw = Hw / Mt(g);

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%
% cria graficos
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% cria DPZ
%

% seleciona canvas
figure(1)

% cria subfigura
subplot(line_max,col_max,g)

% desenha circulo de raio unitario
plot(real(unit_circ),imag(unit_circ),'k')

% mantem todos os graficos na mesma figura
hold on

% desenha os zeros simples
plot(real(z_smm),imag(z_smm),'ok','MarkerSize',sing_dim)

% desenha os polos simples
plot(0,0,'xk','MarkerSize',sing_dim)
text( mult_offset_x , mult_offset_y , num2str((Mt(g)-1)) )

% escreve constante de ganho
% em posicao definida pelos limites do grafico
K_str = 'K = 1';
K_str_offset = dpz_limit / 8;
text( ((-dpz_limit) + K_str_offset) ,
      (( dpz_limit) - K_str_offset) ,
      K_str
      )

% identifica o grafico
%title('Diagrama de polos e zeros (DPZ)')
title(['SMM: DPZ' , ' (M_T=' , num2str(Mt(g)) , ')'])

```



```

%
% cria angulo de fase da RF
%

% seleciona canvas
figure(3)

% cria subfigura
subplot(line_max,col_max,g)

% desenha o modulo da RF
plot(Wnorm,angle(Hw)*108/pi,'k')

% identifica o grafico
title(['SMM: RF' , ' (M_T=' , num2str(Mt(g)) , ')'])
%title(cellstr(['Sistema de Média Móvel';
%               'M_T = ' , num2str(Mt(g))]))
ylabel('\angle H(e^{j\Omega}) ')
xlabel('\Omega_{norm}')

% Controla formato do grafico
% Deve vir ANTES do controle dos limites !!!
axis("normal")

% controla limites
v = axis;
v(3) = -180;
v(4) = 180;
axis(v)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% finaliza o FOR
end

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%
% EOF
%
```

9.5.3 *N-point* DFT da resposta ao impulso

Abaixo, é apresentado um código para Octave, que calcula e exhibe, para valores escolhidos pelo usuário, as curvas de módulo e de argumento da *N-point* DFT $H[k]$ da resposta ao impulso $h[n]$ do SMM. Também é apresentada uma superposição das curvas de $H[k]$ com as curvas da Resposta em Frequência $H(e^{j\Omega})$.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Calculo da N-point DFT H[k]
% da resposta ao impulso h[n].
%
% - Calculo da Resposta em Frequencia H(e^jW).
%
%
% Versao: 2018_11_12.
%
% Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% limpa o ambiente de trabalho
%
clear all
close all

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% define comprimento do SMM
fflush(stdout);
Mt = input('Comprimento Mt do SMM = ');
%Mt = 5;

% define num de pts por mod Mt
fflush(stdout);
NOPPM = input('Número de pontos por módulo Mt = ');
%NOPPM = 10;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% define a resposta ao impulso do SMM
hn = ones(1,Mt) / Mt;

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% define parametros para os graficos
%
FigNbr = 0;
%
k = 0:(Nfft - 1);
%
m = k * (2*pi/Nfft);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
stem(k,Hk_mod,'r')
title(cellstr(['Sistema de Média Móvel (M_T = ' , num2str(Mt) , ')';
               num2str(Nfft) , '-point DFT da resposta ao impulso h[n]']))
ylabel('| H [k] |')
%
v(1) = 0;
v(2) = k(end);
v(3) = 0;
v(4) = 1.01;
axis(v);
%
subplot(2,1,2)
stem(k,Hk_arg,'r')
ylabel('\angle H [k]')
xlabel('k')
%
v(1) = 0;
v(2) = k(end);
v(3) = -200;
v(4) = 200;
axis(v);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(W,Hw_mod,'k')
title(cellstr(['Sistema de Média Móvel (M_T = ' , num2str(Mt) , ')'];

```

```

                'Resposta em frequência H (e^{j \Omega})')]))
ylabel('| H (e^{j \Omega}) |')
%
v(1) = 0;
v(2) = W(end);
v(3) = 0;
v(4) = 1.01;
axis(v);
%
subplot(2,1,2)
plot(W,Hw_arg,'k')
ylabel('\angle H (e^{j \Omega})')
xlabel('\Omega _{norm}')
%
v(1) = 0;
v(2) = W(end);
v(3) = -200;
v(4) = 200;
axis(v);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(W,Hw_mod,'k')
hold on
stem(m,Hk_mod,'r')
title(cellstr(['Sistema de Média Móvel (M_T = ', num2str(Mt) , ')';
                'Resposta em frequência H (e^{j \Omega})';
                '( ', num2str(Nfft) , '-point DFT da resposta ao impulso h[n] )']))
ylabel('| H (e^{j \Omega}) |')
%
v(1) = 0;
v(2) = W(end);
v(3) = 0;
v(4) = 1.01;
axis(v);
%
subplot(2,1,2)
plot(W,Hw_arg,'k')
hold on
stem(m,Hk_arg,'r')
ylabel('\angle H (e^{j \Omega})')
xlabel('\Omega _{norm}')
%
v(1) = 0;

```



```

% calcula as funcoes RF
% dos filtros projetados
%
Hlp = [];
for f = 1:length(Nsmm)
    Hc = 0;
    for c = 1:Nsmm(f)
        Hc = Hc + ( b_lp{f}(c) * exp(-j*(c-1)*W) );
    end
    Hlp = [Hlp ; Hc];
end

% calcula modulo e argumento
% das funcoes de transferencia
% dos filtros projetados
%
Hlpmod = abs(Hlp);
HlpmodDB = 20*log10(Hlpmod);
Hlparg = angle(Hlp)*180/pi;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% calculo as funcoes de transferencia
% do SMM

% inicializa os vetores de valores da RF
H3 = zeros(size(W));
H2 = H3;
H1 = H3;

% define Mt
N3 = Nsmm(3);
N2 = Nsmm(2);
N1 = Nsmm(1);

% calcula a RF
for k = 0:(N3-1)
    H3 = H3 + (exp(-j*k*W));
    if(k<N2)
        H2 = H2 + (exp(-j*k*W));
    endif
    if(k<N1)
        H1 = H1 + (exp(-j*k*W));
    endif
end

% normaliza a RF

```

```

H3 = H3/N3;
H2 = H2/N2;
H1 = H1/N1;

% calcula modulo e fase da RF
%
H3mod = abs(H3);
H3modDB = 20*log10(H3mod);
H3arg = angle(H3)*180/pi;
%
H2mod = abs(H2);
H2modDB = 20*log10(H2mod);
H2arg = angle(H2)*180/pi;
%
H1mod = abs(H1);
H1modDB = 20*log10(H1mod);
H1arg = angle(H1)*180/pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% Graficos
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% parametros basicos
FigNbr = 0;
Wnorm = W/pi;
ModMin = -100;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes projetadas com fir1()
% grafico de modulo linear
%

%
FigNbr = FigNbr+1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,Hlpmod(1,:),'-.',
      Wnorm,Hlpmod(2,:), '--',
      Wnorm,Hlpmod(3,:), '-')

```

```

legend(['fir1(' , num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'],
       ['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
       "location", "east")
title('H_{lp} projetada com fir1(N,Wp)')
ylabel('| H_{lp} |')
%
subplot(2,1,2)
plot(Wnorm,Hlparg(1,:),'-.',
      Wnorm,Hlparg(2,:),'--',
      Wnorm,Hlparg(3,:),'-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
ylabel('\angle H_{lp}')
xlabel('\Omega_N')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes projetadas com fir1()
% grafico de modulo em dB
%

%
FigNbr = FigNbr+1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,HlpmodDB(1,:),'-.',
      Wnorm,HlpmodDB(2,:),'--',
      Wnorm,HlpmodDB(3,:),'-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'],
       ['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
       "location", "southwest")
title('H_{lp} projetada com fir1(N,Wp)')
ylabel('| H_{lp} |_{dB}')
%
subplot(2,1,2)
plot(Wnorm,Hlparg(1,:),'-.',
      Wnorm,Hlparg(2,:),'--',
      Wnorm,Hlparg(3,:),'-')

```

```

v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
ylabel('\angle H_{lp}')
xlabel('\Omega_N')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes projetadas com SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,H1mod,'-.', Wnorm,H2mod,'--', Wnorm,H3mod,'-')
legend(['M_T = ' , num2str(N1)],
       ['M_T = ' , num2str(N2)],
       ['M_T = ' , num2str(N3)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr(['Sistema de Média Móvel';
              ['M_T = ' , num2str(N1), ' , ' , num2str(N2), ' e ' , num2str(N3) ]
              ]))
%
subplot(2,1,2)
plot(Wnorm,H1arg,'-.', Wnorm,H2arg,'--', Wnorm,H3arg,'-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes projetadas com SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;

```

```

figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,H1modDB,'-.', Wnorm,H2modDB,'--', Wnorm,H3modDB,'-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['M_T = ' , num2str(N1)],
       ['M_T = ' , num2str(N2)],
       ['M_T = ' , num2str(N3)],
       "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr(['Sistema de Média Móvel';
  ['M_T = ' , num2str(N1), ' , ' , num2str(N2), ' e ' , num2str(N3) ]
  ])))
%
subplot(2,1,2)
plot(Wnorm,H1arg,'-.', Wnorm,H2arg,'--', Wnorm,H3arg,'-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 1 projetadas com fir1() e SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,H1pmod(1,:), '--',
     Wnorm,H1mod,'-')
legend(['fir1(' , num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'],
       ['M_T = ' , num2str(N1)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr([
  ['H_{lp} projetada com fir1(' , ...
  num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'];
  ['Sistema de Média Móvel com M_T = ' , num2str(N1)]

```

```

        ]) )
%
subplot(2,1,2)
plot(Wnorm,Hlparg(1,:), '--',
      Wnorm,H1arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 1 projetadas com fir1() e SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,HlmodDB(1,:), '--',
      Wnorm,H1modDB, '-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'],
       ['M_T = ' , num2str(N1)],
       "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr([
        'H_{lp} projetada com fir1(' , ...
        num2str(Nfir(1)), ' , ' , num2str(Wp(1)) , ')'];
       ['Sistema de Média Móvel com M_T = ' , num2str(N1)]
       ]))

%
subplot(2,1,2)
plot(Wnorm,Hlparg(1,:), '--',
      Wnorm,H1arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')

```

```

ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 2 projetadas com fir1() e SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,Hlpm(2,:), '--',
      Wnorm,H2mod, '-')
legend(['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['M_T = ' , num2str(N2)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr([
           'H_{lp} projetada com fir1(' , ...
           num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'];
       ['Sistema de Média Móvel com M_T = ' , num2str(N2)]
       ]))

%
subplot(2,1,2)
plot(Wnorm,Hlparg(2,:), '--',
      Wnorm,H2arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 2 projetadas com fir1() e SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%

```

```

subplot(2,1,1)
plot(Wnorm,HlpmodDB(2,:), '--',
      Wnorm,H2modDB, '-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['M_T = ' , num2str(N2)],
       "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr([
          ['H_{lp} projetada com fir1(' , ...
          num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'];
          ['Sistema de Média Móvel com M_T = ' , num2str(N2)]
        ]))
%
subplot(2,1,2)
plot(Wnorm,Hlparg(2,:), '--',
      Wnorm,H2arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 3 projetadas com fir1() e SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,Hlpmod(3,:), '--',
      Wnorm,H3mod, '-')
legend(['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
       ['M_T = ' , num2str(N3)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr([
          ['H_{lp} projetada com fir1(' , ...
          num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'];

```

```

        ['Sistema de Média Móvel com M_T = ' , num2str(N3)]
    ]) )

%
subplot(2,1,2)
plot(Wnorm,Hlparg(3,:), '--',
      Wnorm,H3arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes e projetadas com fir1() e SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,HlpmodDB(3,:), '--',
      Wnorm,H3modDB, '-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
       ['M_T = ' , num2str(N3)],
       "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr([
        ['H_{lp} projetada com fir1(' , ...
        num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'];
        ['Sistema de Média Móvel com M_T = ' , num2str(N3)]
    ]))

%
subplot(2,1,2)
plot(Wnorm,Hlparg(3,:), '--',
      Wnorm,H3arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)

```

```

xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 2 e 1 projetadas com fir1() e SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,Hlpm(2,:), '--',
      Wnorm,Hlmod, '-')
legend(['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['M_T = ' , num2str(N1)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr([
          'H_{lp} projetada com fir1(' , ...
          num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'];
        ['Sistema de Média Móvel com M_T = ' , num2str(N1)]
      ]))

%
subplot(2,1,2)
plot(Wnorm,Hlparg(2,:), '--',
      Wnorm,Hlarg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 2 e 1 projetadas com fir1() e SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)

```

```

%
subplot(2,1,1)
plot(Wnorm,HlpmodDB(2,:), '--',
      Wnorm,H1modDB, '-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'],
       ['M_T = ' , num2str(N1)],
       "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr([
          'H_{lp} projetada com fir1(' , ...
          num2str(Nfir(2)), ' , ' , num2str(Wp(2)) , ')'];
        ['Sistema de Média Móvel com M_T = ' , num2str(N1)]
      ]))

%
subplot(2,1,2)
plot(Wnorm,Hlparg(2,:), '--',
      Wnorm,H1arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 3 e 2 projetadas com fir1() e SMM
% grafico de modulo linear
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,Hlpmod(3,:), '--',
      Wnorm,H2mod, '-')
legend(['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
       ['M_T = ' , num2str(N2)],
       "location", "east")
ylabel('| H(e^{j\Omega}) |')
title(cellstr([
          'H_{lp} projetada com fir1(' , ...

```

```

        num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'];
        ['Sistema de Média Móvel com M_T = ' , num2str(N2)]
    ]) )

%
subplot(2,1,2)
plot(Wnorm,Hlparg(3,:), '--',
      Wnorm,H2arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;
axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% graficos superpostos
% funcoes 3 e 2 projetadas com fir1() e SMM
% grafico de modulo em dB
%

%
FigNbr = FigNbr + 1;
figure(FigNbr)
%
subplot(2,1,1)
plot(Wnorm,HlpmodDB(3,:), '--',
      Wnorm,H2modDB, '-')
v = axis;
v(3) = ModMin;
v(4) = 0;
axis(v)
legend(['fir1(' , num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'],
        ['M_T = ' , num2str(N2)],
        "location", "southwest")
ylabel('| H(e^{j\Omega}) |_{DB}')
title(cellstr([
        'H_{lp} projetada com fir1(' , ...
        num2str(Nfir(3)), ' , ' , num2str(Wp(3)) , ')'];
        ['Sistema de Média Móvel com M_T = ' , num2str(N2)]
    ]) )

%
subplot(2,1,2)
plot(Wnorm,Hlparg(3,:), '--',
      Wnorm,H2arg, '-')
v = axis;
v(3) = -200;
v(4) = 200;

```

```

axis(v)
xlabel('\Omega_{norm}')
ylabel('\angle H(e^{j\Omega})')

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

%
% EOF
%
```

9.5.5 Relação entre a Função de Transferência, a Resposta em Frequência e o Círculo Unitário

Abaixo, é apresentado um código para Octave, que gera a superfície de módulo da Função de Transferência, a curva da função Resposta em Frequência sobre a superfície e a curva projetada (círculo de raio unitário). É possível escolher entre um SMM, que é naturalmente um sistema do tipo passa-baixa, e um sistema do tipo passa-alta, definido a partir de um SMM.

```

%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%

%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
% Sistema de Media Movel (SMM):
%
% - Relacionamento entre
% a Funcao de Transferencia (ou Funcao do Sistema),
% a funcao Resposta em Frequencia
% e
% o circulo de raio unitario.
%
% - Elaboracao da superficie de modulo da Funcao de Transferencia,
% da curva da funcao Resposta em Frequencia sobre a superficie
% e
% da curva projetada (circulo de raio unitario).
%
% - Geracao de um sistema do tipo passa-alta,
% a partir de um SMM,
% que eh naturalmente um sistema do tipo passa-baixa.
%
%
% Versao: 2018_10_10.
%
% Autor : Alexandre Santos de la Vega.
%
% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%% %%%%%%%%%%%%%
%
```

```

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% limpa o ambiente de trabalho
%
clear all
close all

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% define parametros

% seleciona o tipo do sistema
%
Sel_type = 'LP'; % lowpass
%Sel_type = 'HP'; % highpass

% define o comprimento do polinomio
%
%Mt = 6; % odd !!!
Mt = 5; % odd !!!

% calcula os coeficientes do polinomio
%
b = ones(1,Mt) / Mt;

% taxa de amostragem das variaveis independentes
%
%Z_smp = 0.01;
%Z_smp = 0.02;
Z_smp = 0.03;
%Z_smp = 0.05;
%Z_smp = 0.10;

% inicializa o contador de figuras
FigNbr = 0;

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% calcula os pontos da superficie com limites genericos

% define os valores das variaveis independentes
%
```

```

z_lim = 1.25;
%
Re_z = [-z_lim:Z_smp:-Z_smp Z_smp:Z_smp:z_lim];
Im_z = [-z_lim:Z_smp:-Z_smp Z_smp:Z_smp:z_lim];

% calcula os valores do plano de variaveis independentes
%
[Mr,Mi] = meshgrid(Re_z,Im_z);

% calcula os valores da variavel complexa
%
Mz = Mr + j*Mi;

% calcula os pontos da superficie
%
Hz = b(1) * ones(size(Mz));
for k = 2:length(b)
    Hz = Hz + ( b(k) .* ( Mz .^ (-(k-1)) ) );
end
%
if (Sel_type == 'HP')
    Hz = (1 - Hz);
endif

%%%%%%%%%%%%%

% calcula os pontos da curva sobre a superficie

% calcula os valores do angulo
%
Omega = 0:Z_smp:(2*pi);

% define os valores das variaveis independentes
%
Re_z_line = cos(Omega);
Im_z_line = sin(Omega);

% calcula os valores da variavel complexa
%
z_circ = Re_z_line + j*Im_z_line;

% calcula os pontos da curva sobre a superficie
%
Hz_line = b(1) * ones(size(z_circ));
for k = 2:length(b)
    Hz_line = Hz_line + ( b(k) .* ( z_circ .^ (-(k-1)) ) );
end
%
```

```

if (Sel_type == 'HP')
    Hz_line = (1 - Hz_line);
endif

%%%%%%%%%%%%

% desenha:
%   a superfície, a curva sobre a superfície e a curva projetada

% cria canvas
FigNbr = FigNbr + 1;
figure(FigNbr)

% desenha a superfície
%
mesh(Mr,Mi,abs(Hz))
%
%surf(Mr,Mi,abs(Hz))
%
%surf(Mr,Mi,abs(Hz),'FaceColor','blue','EdgeColor','none')
%camlight left; lighting phong

% garante superposicao de curvas
%
hold on

% desenha a curva sobre a superfície
%
plot3(Re_z_line,Im_z_line,abs(Hz_line),'r')

% desenha a curva projetada
%
plot(Re_z_line,Im_z_line,'k')

% coloca labels
%
if (Sel_type == 'LP')
    tit_Hz = {'Sistema de Média Móvel';
             'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)'$ ;
             'Curva: Resposta em Frequência = DTFT  $\{ h[n] \} = H(e^{j \Omega})'$ };
else
    tit_Hz = {'Uso do Sistema de Média Móvel como subsistema';
             'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)'$ ;
             'Curva: Resposta em Frequência = DTFT  $\{ h[n] \} = H(e^{j \Omega})'$ };
endif
title(tit_Hz)
xlabel('Re\{z\}')
ylabel('Im\{z\}')
zlabel('|H(z)|')

```



```

% a superficie, a curva sobre a superficie e a curva projetada

% cria canvas
FigNbr = FigNbr + 1;
figure(FigNbr)

% desenha a superficie
%
mesh(Mr,Mi,abs(Hz))
%
%surf(Mr,Mi,abs(Hz))
%
%surf(Mr,Mi,abs(Hz),'FaceColor','blue','EdgeColor','none')
%camlight left; lighting phong

% garante superposicao de curvas
%
hold on

% desenha a curva sobre a superficie
%
plot3(Re_z_line,Im_z_line,abs(Hz_line),'r')

% desenha a curva projetada
%
plot(Re_z_line,Im_z_line,'k')

% coloca labels
%
if (Sel_type == 'LP')
    tit_Hz = {'Sistema de Média Móvel';
        'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)$ ';
        'Curva: Resposta em Frequência = DTFT  $\{ h[n] \} = H(e^{j \Omega})$ '};
else
    tit_Hz = {'Uso do Sistema de Média Móvel como subsistema';
        'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)$ ';
        'Curva: Resposta em Frequência = DTFT  $\{ h[n] \} = H(e^{j \Omega})$ '};
endif
title(tit_Hz)
xlabel('Re\{z\}')
ylabel('Im\{z\}')
zlabel('|H(z)|')

% controla area visivel
%
v = axis;
v(5) = 0;
%v(6) = 1.2;
v(6) = 2.0;

```

```

axis(v)

% define ponto de vista
%
%%view(15,10)
%%view(-45,30)
%view(-45,65)
view(22,15)

%%%%%%%%%%

% desenha:
%   a superficie, a curva sobre a superficie e a curva projetada

% cria canvas
FigNbr = FigNbr + 1;
figure(FigNbr)

% desenha a superficie
%
%mesh(Mr,Mi,abs(Hz))
%
%surfc(Mr,Mi,abs(Hz))
%
surf(Mr,Mi,abs(Hz),'FaceColor','blue','EdgeColor','none')
%camlight left; lighting phong

% garante superposicao de curvas
%
hold on

% desenha a curva sobre a superficie
%
plot3(Re_z_line,Im_z_line,abs(Hz_line),'r')

% desenha a curva projetada
%
plot(Re_z_line,Im_z_line,'k')

% coloca labels
%
if (Sel_type == 'LP')
    tit_Hz = {'Sistema de Média Móvel';
             'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)$ ';
             'Curva: Resposta em Freqüência = DTFT  $\{ h[n] \} = H(e^{j \Omega})$ '};
else
    tit_Hz = {'Uso do Sistema de Média Móvel como subsistema';
             'Superfície: Função de Transferência =  $Z \{ h[n] \} = H(z)$ ';
             'Curva: Resposta em Freqüência = DTFT  $\{ h[n] \} = H(e^{j \Omega})$ '};

```

```

endif
title(tit_Hz)
xlabel('Re\{z\}')
ylabel('Im\{z\}')
zlabel('|H(z)|')

% controla area visivel
%
v = axis;
v(5) = 0;
%v(6) = 1.2;
v(6) = 2.0;
axis(v)

% define ponto de vista
%
%%%view(15,10)
%%%view(-45,30)
%view(-45,65)
view(22,15)

%%%%%%%%%%

% desenha:
%   a curva

% cria canvas
FigNbr = FigNbr + 1;
figure(FigNbr)

% desenha a curva
%
subplot(2,1,1)
plot(Omega/pi,abs(Hz_line))

% coloca labels
%
if (Sel_type == 'LP')
    tit_Hjw = {'Sistema de Média Móvel';
              'Curva: Resposta em Frequência = DTFT \{ h[n] \} = H(e^{j \Omega})'};
else
    tit_Hjw = {'Uso do Sistema de Média Móvel como subsistema';
              'Curva: Resposta em Frequência = DTFT \{ h[n] \} = H(e^{j \Omega})'};
endif
title(tit_Hjw)
ylabel('| H(e^{j \Omega}) |')

% desenha a curva
%
```


Capítulo 10

Códigos para *kit* Arduino

10.1 Introdução

A seguir, são apresentados códigos para *kit* Arduino, que implementam um sistema de média móvel para alguma aplicação específica.

10.2 Sensor LDR

Abaixo, é apresentado um código para *kit* Arduino, que implementa um sistema de média móvel para a leitura de um sensor LDR (*Light Dependent Resistor*).

```
//  
// =====  
//  
// Titulo: Implementacao de Sistema de Media Movel  
//         para leitura de sensor LDR  
//         (Light Dependent Resistor).  
//  
// Plataforma: Kit Arduino.  
//  
// Versao: 2018_10_08.  
//  
// Autor : Thiago Chequer Coelho (PET-Tele/UFF).  
// =====  
//  
  
#define N 10  
  
#define SENSOR A0  
  
int foto_res = 0;  
  
int loop_read[N];
```

```
int media;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int read_res = analogRead(SENSOR);

    for (int i = N-1; i > 0; i--)
    {
        loop_read[i] = loop_read[i-1];
    }

    loop_read[0] = read_res;

    long soma = 0;

    for (int i = 0; i < N; i++)
    {
        soma = soma + loop_read[i];
    }

    media = soma/N;

    Serial.print(read_res);

    Serial.print("    ");
```

```

Serial.println(media);

}

//
// EOF
//

```

10.3 Medidor de vazão

Abaixo, é apresentado um código para *kit* Arduino, que implementa um sistema de média móvel para a leitura de um medidor de vazão.

```

//
// =====
//
// Titulo: Implementacao de Filtro de Media Movel
//         para leitura de medidor de vazao.
//
// Plataforma: Kit Arduino.
//
// Versao: 2018_10_08.
//
// Autor : Thiago Chequer Coelho (PET-Tele/UFF).
//         Lucas Pontes Siqueira (PET-Tele/UFF).
//
// =====
//

#include "PinChangeInt.h"
#define PinItrpt11 11
#define N 10

float vazao_entrada;           //Variavel para armazenar o valor em L/min
float vazao_intermediaria;
float vazao_saida;           //Variavel para armazenar o valor em L/min

float read_entrada[N];
float read_intermediaria[N];
float read_saida[N];

int contaPulso_entrada;       //Variavel para a quantidade de pulsos
int contaPulso_intermediaria;
int contaPulso_saida;
int i=0; //Variavel para contagem

void setup(){
  Serial.begin(9600); //Inicia a serial com uma baud rate de 9600

```

```

pinMode(2, INPUT);
attachInterrupt(0, incpulso_entrada, RISING);

pinMode(3, INPUT);
attachInterrupt(1, incpulso_saida, RISING);

pinMode(PinItrpt11, INPUT);
PCintPort::attachInterrupt(PinItrpt11, incpulso_intermediaria, RISING);

Serial.println("\nINICIO\n"); //Imprime Inicio na serial
Serial.println("ENTRADA -- INTERMEDIARIO -- SAIDA");
}

void loop (){

    contaPulso_entrada = 0; //Zera a Variavel para contar os giros por segundos
    contaPulso_intermediaria = 0;
    contaPulso_saida = 0;

    sei(); //Habilita interrupcao
    delay (1000); //Aguarda 1 segundo
    cli(); //Desabilita interrupcao

    vazao_entrada = contaPulso_entrada/7.5; //Converte para L/min
    vazao_intermediaria = contaPulso_intermediaria/7.5;
    vazao_saida = contaPulso_saida/7.5; //Converte para L/min
    i++;

    float SMM_entrada = SMM(read_entrada, vazao_entrada);
    float SMM_intermediaria = SMM(read_intermediaria, vazao_intermediaria);
    float SMM_saida = SMM(read_saida, vazao_saida);

    Serial.print(" ");
    Serial.print(SMM_entrada); //Imprime na serial o valor da vazao de entrada
    Serial.print(" ");
    Serial.print(SMM_intermediaria); //Imprime na serial o valor da vazao de saída
    Serial.print(" ");
    Serial.print(SMM_saida); //Imprime na serial o valor da vazao de saída
    Serial.print(" ");
    Serial.print(" L/min"); //Imprime L/min
    Serial.print(" ");
    Serial.print(i); //Imprime a contagem i (segundos)
    Serial.println("s"); //Imprime s indicando que está em segundos
}

void incpulso_entrada (){
    contaPulso_entrada++; //Incrementa a Variavel de contagem dos pulsos
}

```

```
}

void incpulso_intermediaria (){
    contaPulso_intermediaria++; //Incrementa a Variavel de contagem dos pulsos
}

void incpulso_saida (){
    contaPulso_saida++; //Incrementa a Variavel de contagem dos pulsos
}

float SMM(float* loop_read, float in_read){
    for (int i = N-1; i > 0; i--){
        loop_read[i] = loop_read[i-1];
    }
    loop_read[0] = in_read;
    long soma = 0;
    for (int i = 0; i < N; i++){
        soma = soma + loop_read[i];
    }
    float media = soma/N;
    return media;
}

//
// EOF
//
```


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 L^AT_EX and Electronic Publishing*. Addison-Wesley, Harlow, England, 4th edition, 2004.
- [MG04] F. Mittelbach and M. Goossens. *The L^AT_EX 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. Schaffer. *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.

