

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

MODELO PARA O ENSINO DE ALGORITMOS DE RECONSTRUÇÃO DE SINAL EM ENGENHARIA

Tese de Doutoramento

MÓNICA ISABEL TEIXEIRA DA COSTA



Vila Real, 2015

**Mónica Isabel Teixeira da
Costa**

**MODELO PARA O ENSINO DE ALGORITMOS DE
RECONSTRUÇÃO DE SINAL EM ENGENHARIA**

Tese apresentada à Universidade de Trás-os-Montes e Alto Douro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Informática, realizada sobre a orientação científica do Professor Doutor João Manuel Pereira Barroso, Professor Associado com Agregação do Departamento de Engenharias da Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro, e do Professor Doutor Salviano Filipe Silva Pinto Soares, Professor Auxiliar do Departamento de Engenharias da Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro.

Júri

Presidente

Prof. Doutor Vitor Manuel de Jesus Filipe

Professor Associado c/ Agregação da Universidade de Trás-os-Montes e Alto Douro

Prof. Doutor Paulo Jorge dos Santos Gonçalves Ferreira

Professor Catedrático da Universidade de Aveiro

Prof. Doutor Vítor Manuel Pereira Duarte dos Santos

Professor Auxiliar Convidado do ISEGI da Universidade Nova de Lisboa

Prof. Doutor Manuel José Cabral dos Santos Reis

Professor Associado c/ Agregação da Universidade de Trás-os-Montes e Alto Douro

Prof. Doutor Tiago Oliveira Machado de Figueiredo Cardoso

Professor Auxiliar do Departamento de Engenharia Eletrotécnica da Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa

Prof. Doutor Fernando Reinaldo Da Silva Garcia Ribeiro

Professor Adjunto da Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco

Agradecimentos

Começo por expressar os meus sinceros agradecimentos aos meus orientadores Professor Doutor João Manuel Pereira Barroso e Professor Doutor Salviano Filipe Silva Pinto Soares pela orientação científica, o incentivo, as sugestões e discussões sempre oportunas, a confiança e o apoio constantes ao longo destes anos.

Ao Instituto Politécnico de Castelo Branco (IPCB) e à Unidade Técnico Científica de Informática da Escola Superior de Tecnologia de Castelo Branco, agradeço o apoio que me foi concedido no âmbito do Contrato-Programa de Qualificação do corpo Docente do IPCB.

À minha família agradeço o carinho e o apoio constante durante este trabalho.

A todos aqueles que, de forma direta ou indireta, contribuíram para a realização deste trabalho, o meu sincero obrigado.

Resumo

Nesta tese apresenta-se um modelo de ensino de algoritmos de reconstrução de sinal e respetivos condicionalismos inerentes a este problema de engenharia. Para tal desenvolveu-se um protótipo aplicacional, o qual foi denominado de *Signal Processing Interpolation Educational Workbench* (SPIEW), tendo por finalidade melhorar e avaliar a aprendizagem dos alunos em reconstrução de sinal, recorrendo a um modelo que aborda de forma gráfica e interativa os conceitos de áreas afins mas distintas, desde a Álgebra, até ao Processamento de Sinal passando por exemplo, pela aplicação de Códigos de Correção de Erros.

Com o SPIEW é possível estabelecer conjeturas sobre limitação em frequência de sinais, geometrias de padrão de erros, classes de algoritmos de reconstrução de sinal de dimensão do espaço (máxima) ou da ordem do número de amostras perdidas (de dimensão mínima), o uso de métodos de cálculo: diretos, iterativos ou semi-iterativos para a resolução de sistemas de equações lineares ou medir o desempenho computacional dos métodos ótimos. É possível ainda analisar-se a estabilidade verificando se um problema é bem/mal condicionado, bem como avaliar as condições de convergência dos algoritmos em vários cenários de problemas de interpolação lineares. Os dados utilizados neste trabalho provenientes de sinais: 1D-sinais áudio mono e de sensores de biotelemetria; 2D-imagens entre as quais a área de biomecânica; 3D-mapas de profundidade em codificação MDC (*Multiple Description Coding*) de vídeo 3D; testemunham a aplicabilidade dos métodos propostos em vários contextos permitindo e motivando a discussão de vários cenários reais no ambiente de aprendizagem proposto.

O modelo proposto para o estudo de algoritmos de reconstrução de sinal tem por missão permitir aos seus utilizadores melhor compreender ou aprofundar os conceitos inerentes a esta área de estudo. Abordando os conceitos e operações de forma gráfica e interativa, o utilizador tem sempre a oportunidade de visualizar o resultado das operações efetuadas, passo a passo, facilitando a compreensão dos conceitos do processamento de sinal mais abstratos, mesmo para utilizadores com conhecimentos limitados na área.

O protótipo aplicativo desenvolvido fornece aos seus utilizadores uma experiência concreta, de um problema real e atual da engenharia. Este permite a simulação de cenários reais de problemas de reconstrução, dando aos alunos a possibilidade de testar diferentes métodos de reconstrução e aferir sobre a eficácia dos mesmos relativamente a tempo de execução e avaliação dos resultados obtidos, podendo garantir-se assim *a priori* a realização de sistemas de engenharia para determinados (projetos) requisitos iniciais.

Comprovou-se através da realização de testes *Mean Opinion Score (MOS)* que a possibilidade de interação com o utilizador poderá ser efetivamente uma mais-valia para a aprendizagem.

Abstract

This thesis presents a teaching model for signal reconstruction algorithms and the constraints that are inherent to this engineering problem. For this purpose an application prototype was developed, it was named *Signal Processing Interpolation Educational Workbench* (SPIEW). The main goal is to improve the students' learning processes in the signal reconstruction area using a model that approaches the various versed concepts in a graphical and interactive way, from Algebra to Signal Processing, for instance through the application of Error Correction Codes.

With SPIEW it is possible to establish conjectures about the signals' band limiting, error pattern geometries, reconstruction algorithm classes for space dimension signals (maximum) or of the order of the number of lost samples (minimum size), the use of calculation methods: direct, iterative or semi iterative to solve linear equation systems or to measure the optimum computational method's performance. It is further possible to analyse the stability by verifying if a problem is well/badly conditioned, as well as evaluate the algorithms' convergence in several scenarios of linear interpolation problems. The data used in this work originating from signals: 1D-mono audio signals and biotelemetry sensors; 2D-images including the biomechanic's area; 3D- depth maps using MDC (Multiple Description Coding) 3D video coding; witness the applicability of the proposed methods in various contexts allowing and encouraging the discussion of several real scenarios in the proposed learning environment.

The aim of the proposed model, to study signal reconstruction algorithms, is to allow its users to better understand and increase the concepts inherent to this area of study. By addressing the concepts and operations in a graphical and interactive manner, the user always has the opportunity to view the operations' results as these are executed step by step, and as such it facilitates the understanding of the more abstract signal processing concepts, even when the user has little knowledge in this area.

The developed prototype provides its users with a concrete engineering experience, of a real and actual engineering problem. It allows for the simulation of realistic scenarios of reconstruction problems, giving students the opportunity to test different reconstruction methods and to validate

their effectiveness regarding the execution time, and to evaluate the results obtained. In doing so, the engineering systems (projects) for the specific initial requirements can be guaranteed a priori.

It was proved, with the realization of Mean Opinion Score (MOS) tests, that the interaction with the users demonstrated it as being an asset in the learning process.

Índice

Júri	i
Agradecimentos	iii
Resumo.....	v
Abstract	vii
Índice de Figuras	xiii
Índice de Tabelas.....	xv
Acrónimos	xvii
Capítulo 1 – Introdução.....	1
1.1 Enquadramento e Motivação.....	1
1.2 Objetivos	3
1.3 Contribuições	4
1.4 Metodologia de Investigação	5
1.5 Organização do Trabalho	8
1.6 Publicações do Trabalho	8
Capítulo 2 – Resolução de Sistemas de Equações Lineares	11
2.1 Álgebra de Matrizes: Conceitos	13
2.2 Propriedades das Matrizes.....	14
2.3 Sistema de Equações Bem e Mal Condicionados.....	18
2.4 Métodos Diretos.....	20
2.4.1 Método de Gauss	21
2.4.2 Fatorização LU	22
2.4.3 Cálculo da Matriz Inversa	24
2.4.4 Método de Cholesky	25
2.5 Métodos Iterativos	26
2.5.1 Método de Jacobi.....	28

2.5.2 Método de Gauss-Seidel.....	29
2.5.3 SOR - Successive Over-Relaxation	30
2.6 Métodos Semi-Iterativos	32
2.6.1 Steepest Descent	32
2.6.2 Gradientes Conjugados	33
2.7 Pré-Condicionamento de Matrizes.....	34
Capítulo 3 – Interpolação Linear de Sinais	39
3.1 Papoulis-Gerchberg – Um Algoritmo de Dimensão Máxima.....	39
3.1.1 Notação e Terminologia	40
3.1.2 Condições de Aplicabilidade.....	40
3.1.3 Sinais de Banda Limitada ou de Energia Finita	41
3.1.4 Papoulis-Gerchberg – Interpretação Gráfica.....	43
3.1.5 Convergência do Algoritmo de Papoulis-Gerchberg	46
3.2 Dimensão Mínima.....	47
Capítulo 4 – Um Modelo para o Ensino de Algoritmos de Reconstrução de Sinal.....	51
4.1 Ciclo de Aprendizagem de Kolb	52
4.2 Ciclo de Aprendizagem Implementado	54
4.2.1 Primeira Etapa: Extração de Sinais e Condicionamento do Problema	55
4.2.2 Segunda Etapa: Avaliação da Convergência e Reconstrução dos Sinais	62
Capítulo 5 – Protótipo Aplicacional	67
5.1 Metodologia de Desenvolvimento	67
5.2 Seleção e Extração dos Sinais	69
5.2 Limitação em Frequência	71
5.4 Decimação dos Sinais	73
5.5 Análise do Condicionamento do problema	76
5.6 Reconstrução dos Sinais	78
5.6 Avaliação Objetiva dos Resultados da Reconstrução	81

5.8 Validação da Robustez do Protótipo Aplicacional - SPIEW	82
5.8.1 Resultados da Reconstrução – Decimação Contígua (Rajada)	85
5.8.2 Resultados da Reconstrução – Decimação Entrelaçada	87
5.8.3 Resultados da Reconstrução – Decimação Aleatória	90
Capítulo 6 – Avaliação subjetiva do modelo de aprendizagem: testes MOS.....	95
6.1 Metodologia Utilizada na Realização dos Testes MOS	95
6.2 Resultados obtidos com a primeira versão do SPIEW.....	96
6.3 Resultados obtidos com a segunda versão do SPIEW.....	97
Capítulo 7 – Conclusões e Trabalho Futuro	103
Referências.....	105
Anexos.....	111
Anexo A – Tutorial do SPIEW.....	113
Anexo B – Inquéritos SPIEW.....	125

Índice de Figuras

Figura 1.1 – Metodologia <i>Design Science</i> (adaptado de [Vaishnavi04])	6
Figura 3.1 – Sinal não limitado na frequência (a) e respetiva FFT (b)	42
Figura 3.2 – Limitação de banda de um sinal utilizando o operador B	43
Figura 3.3 – (a) Sinal $x(t)$ contínuo limitado em frequência; (b) Sinal $x(t)$ discretizado	44
Figura 3.4 – (a) Trem de impulsos para amostrar o sinal $x[n]$; (b) Sinal $x[n]$ corrompido	44
Figura 3.5 – Limitação de banda de um sinal	46
Figura 3.6 – Substituição das amostras conhecidas no sinal de banda limitada	46
Figura 4.1 – Ciclo de aprendizagem de <i>Kolb</i>	52
Figura 4.2 – Fluxo de informação etapa de extração e condicionamento de sinal	56
Figura 4.3 – Processo de extração dos sinais de uma sequência	57
Figura 4.4 – Processo para guardar todos os <i>frames</i> da sequência numa única matriz	57
Figura 4.5 – Processo para guardar todas as linhas da imagem nas colunas de uma matriz	58
Figura 4.6 – Decimação entrelaçada. Sinal observado (a), Entrelaçamento=2 (b), Entrelaçamento=4 (c), Entrelaçamento=8 (d).	60
Figura 4.7 – Decimação aleatória, total de amostras perdidas é 5	60
Figura 4.8 – Decimação contínua, total de amostras perdidas é 4	61
Figura 4.9 – Decimação contínua, total de amostras perdidas é 5	61
Figura 4.10 – Fluxo de informação etapa de análise de convergência e reconstrução de sinal	63
Figura 4.11 – Métodos Iterativos, Diretos e Semi-iterativos Disponíveis no Protótipo Aplicacional	64
Figura 5.1 – Modelo de prototipagem (adaptado de [Miguel03])	69
Figura 5.2 – Protótipo aplicacional desenvolvido. Interface de extração e condicionamento dos sinais	70
Figura 5.3 – Informação sobre os <i>frames</i> ou imagens.	70
Figura 5.4 – Processo de extração dos sinais.	71
Figura 5.5 – Informação sobre os sinais extraídos.	71

Figura 5.6 – Operação de limitação em frequência. Escolha do fator de sobre-amostragem.....	72
Figura 5.7 – Limitação em frequência: janela de progresso (a) e largura de banda do filtro passa baixo utilizado (b).....	73
Figura 5.8 – Processo de Decimação - geometrias de padrão de erro disponíveis.....	73
Figura 5.9 – Decimação entrelaçada, janela para escolher o valor de entrelaçamento.	73
Figura 5.10 – Decimação aleatória, 200 amostras perdidas, processo concluído.	74
Figura 5.11 – Imagem após da decimação aleatória, 200 amostras perdidas por linha.	75
Figura 5.12 – Versões: original, observado e decimado do sinal correspondente linha 500 extraída de uma imagem.....	75
Figura 5.13 – SPIEW, Janela de Reconstrução.....	76
Figura 5.14 – Análise do condicionamento do problema.....	77
Figura 5.15 – Análise da condição de convergência (a) e respetiva legenda (b).....	78
Figura 5.16 – Métodos de reconstrução disponíveis.	78
Figura 5.17 – Resultados da reconstrução da Linha 200, método dimensão mínima - iterativo.....	79
Figura 5.18 – Resultados da reconstrução da Linha 200, método direto.	80
Figura 5.19 – Resultados da reconstrução da Linha 200, método Gradientes Conjugados.....	81
Figura 5.20 – Informação sobre o TU de referência a ser utilizado na reconstrução.	82
Figura 5.21 – Parâmetros para avaliar a qualidade e velocidade de reconstrução.	82
Figura 6.1 – SPIEW - Resultados Obtidos nas questões de Verdadeiro e Falso. Teste com Docentes..	97
Figura 6.2 – SPIEW - Resultados nas 6 primeiras questões. Alunos UTAD.....	98
Figura 6.3 – SPIEW - Resultados nas 6 primeiras questões. Alunos EST-IPCB.....	99
Figura 6.4 – SPIEW – Resultados obtidos no inquérito, excepto na questão 4 e 7. Alunos de doutoramento UTAD.	100
Figura 6.5 – SPIEW – Resultados obtidos no inquérito, excepto na questão 7. Alunos de Engenharia Informática IPCB.	101

Índice de Tabelas

Tabela 5.1 – Resultados da reconstrução, linha 500, decimação contígua, 20 amostras perdidas.....	85
Tabela 5.2 – Resultados da reconstrução, linha 500, decimação contígua, 50 amostras perdidas.....	86
Tabela 5.3 – Resultados da reconstrução, linha 500, decimação contígua, 100 amostras perdidas.....	87
Tabela 5.4 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump =2), 512 amostras perdidas.....	88
Tabela 5.5 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump =4), 768 amostras perdidas.....	89
Tabela 5.6 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump = 8), 896 amostras perdidas.....	90
Tabela 5.7 – Resultados da reconstrução, linha 500, decimação aleatória, 250 amostras perdidas.	91
Tabela 5.8 – Resultados da reconstrução, linha 500, decimação aleatória, 850 amostras perdidas.	92
Tabela 5.9 – Resultados da reconstrução, linha 500, decimação aleatória, 900 amostras perdidas.	93

Acrónimos

DS – *Design Science*

FFT – *Fast Fourier Transform*

FPB – *Filtro Passa Baixo*

IFFT – *Inverse Fast Fourier Transform*

LB – *Largura de Banda*

MDC - *Multiple Description Coding*

MOS - *Mean Opinion Score*

PDS – *Processamento Digital de Sinal*

PSNR – *Peak Signal-to-Noise Ratio*

RMSE - *Root Mean-Squared Error*

SOR - *Successive Over-Relaxation*

SPIEW – *Signal Processing Interpolation Educational Workbench*

TU – *Time Unit*

Capítulo 1 – Introdução

1.1 ENQUADRAMENTO E MOTIVAÇÃO

A engenharia é uma área de conhecimento que se foi desenvolvendo pela necessidade que a humanidade teve de aumentar a capacidade de produção e deste modo colmatar as suas necessidades de conforto e bem-estar [Carvalho01]. Hoje em dia, com o crescente desenvolvimento de novas tecnologias e com o acesso facilitado a todo o tipo de informação, podemos afirmar que terá mais sucesso aquele indivíduo que conseguir desenvolver ou aprender de forma mais célere os saberes. O homem é por si só um ser de distinta curiosidade, não se conforma com o que desenvolve e está na maior parte das vezes à procura de novos conhecimentos.

A engenharia vive uma época de grandes transformações [Vieira13] [Belhot01], as técnicas e tecnologias estão em constante mudança. Tendo esta prerrogativa em mente, as escolas de engenharia precisam de ensinar aos seus alunos mais do que conceitos e fórmulas. Os futuros engenheiros necessitam de adquirir competências que lhes permitam enfrentar adversidades sem surpresas [Vieira13] [Belhot01]. A formação de um engenheiro não pode estar unicamente baseada em fórmulas e conceitos. Um engenheiro vai ter que tomar decisões, saber procurar informação e saber aplicá-la. Resumindo, o aluno necessita de aprender a aprender.

A aprendizagem pode ser entendida como um processo que envolve duas dimensões: a perceção da informação e o processamento da informação. As teorias de aprendizagem são as referências base quando se deseja melhorar o processo de ensino/aprendizagem [Belhot01]. É através delas que se torna mais fácil compreender porque é que alguns alunos aprendem e outros não; porque obtêm alguns professores mais sucesso no ensino; porque é que algumas matérias são de mais fácil compreensão do que outras, entre outros. Algumas dessas teorias estão descritas em [Carvalho01] e [Kolb84].

O conhecimento de técnicas pedagógicas adequadas facilita o desenvolvimento do aluno e o planeamento do professor, de modo a atingir eficazmente os objetivos inicialmente propostos [Belhot01]. O ciclo da aprendizagem é uma ferramenta importante que permite ao professor entender

de que forma a aprendizagem está a ser efetuada e como o aluno poderá ser acompanhado adequadamente [Belhot01].

No desenvolvimento do trabalho apresentado nesta tese foi utilizado o ciclo de aprendizagem de Kolb [Kolb84], este será descrito com mais detalhe no capítulo 4 desta tese. Existem outros ciclos de aprendizagem descritos em [Carvalho01], no entanto o público-alvo são, neste caso concreto, alunos de engenharia, cuja grande percentagem é classificada como assimiladores e convergentes [Pierre99]. A metodologia aqui proposta visa complementar e melhorar o processo de aprendizagem através de uma experimentação ativa. O objetivo é complementar o ensino típico de conceitos com experiências concretas onde os alunos possam comprovar e testar os conceitos teóricos préviamente ministrados. No ciclo de aprendizagem de Kolb um indivíduo é submetido a uma experiência, denominada experiência concreta. Primeiramente o indivíduo analisa o fenómeno e faz uma reflexão sobre o mesmo, passa depois para a etapa seguinte denominada de observação reflexiva. Para obter uma consolidação de conceitos, que lhe permitirá encarar situações semelhantes com maior eficácia no futuro, o indivíduo abstrai o conceito passando à generalização de ideias que é denominada de conceituação abstrata. Esses conceitos, em forma de ideias, serão transformados em ações, que serão posteriormente testadas em situações reais. Esta fase é também conhecida como experimentação ativa, e nela o indivíduo analisa o grau significativo que o facto generalizado teve, e se for significativo, passará a fazer parte do seu conhecimento, finalizando a aprendizagem.

Hoje em dia devemos valorizar a complexidade e a magnitude dos problemas reais, dando oportunidade aos estudantes de manipular esse tipo de problemas, dentro de condições de aprendizagem controlada. O problema da reconstrução de sinal, um problema atual em engenharia, é um exemplo disso. A capacidade de análise desta classe de problemas está fortemente relacionada com competências adquiridas em várias áreas do saber da Engenharia, criam-se pois as condições ideais para o desenvolvimento de ferramentas tecnológicas auxiliares cujo objetivo é potenciar o processo de aprendizagem. Muitos dos conceitos abordados requerem diversos conhecimentos em matemática, e sendo esta disciplina, desde algum tempo, uma disciplina problemática, é frequentemente um fator que dissuade os alunos de concorrer a cursos de engenharia. Depois, regra geral, os alunos não vêem qualquer aplicação prática para os conceitos ou teorias lecionados em diversas disciplinas. É possível explicar aos alunos dos vários graus de ensino qual a utilidade dos conceitos lecionados no mundo real. Recorrendo a exemplos e situações reais, eles poderão ver que os conceitos e fórmulas que lhes foram ensinados realmente funcionam e podem igualmente ver onde estes poderão ser aplicados.

Uma estratégia educacional que está sendo usada com muito sucesso é a simulação combinada com o ciclo de ensino - aprendizagem, a qual vai além da simulação convencional. Apesar do uso da simulação no ensino não ser uma ideia nova, usá-la como uma ferramenta isolada tende a simplificar o ambiente, restringir o intuito do problema e a limitar o envolvimento das pessoas. Parece evidente que, para dotar os estudantes de engenharia de conhecimento, experiência e compreensão da realidade, estes atributos poderão ser enriquecidos com o suporte de estratégias educacionais adequadas. A importância do uso da simulação e de laboratórios de experiências no ensino da engenharia tem sido um tema bastante abordado em diversos artigos [Belhot01; Abdulwahed09; Konak13].

Tendo toda esta problemática em mente, decidiu-se investigar de que forma o ensino da reconstrução de sinal e seus respectivos condicionalismos através do uso de um auxiliar tecnológico permitiria a alunos com e sem conhecimento na área, consolidar ou obter conhecimentos sobre o tema.

No processamento digital de sinal abordam-se tópicos como a análise, amostragem, filtragem, corrupção, interpolação, entre outros. Aos alunos são apresentadas as técnicas e teorias que possibilitam a sua implementação. Mas o que significa, na prática, ter um sinal subamostrado ou sobre-amostrado? Uma operação de limitação em frequência (ou filtragem) terá como resultado um sinal pouco ou muito diferente do sinal original? Porque se limitam os sinais em frequência? Por exemplo, num canal de transmissão de voz, se devido a problemas no canal de transmissão perdermos várias amostras, será mais ou menos difícil reconstruir esse sinal no receptor? São questões como estas que um aluno deverá responder quando completado o ciclo de ensino-aprendizagem na área. Pretende-se que, no fim, os alunos possam compreender, por exemplo, de que modo o raio espectral de uma matriz irá afetar a convergência de um sistema em engenharia, de como a representação matemática do problema pode tornar o sistema estável ou não, e concluir quais os parâmetros críticos para a estabilidade de um sistema em engenharia.

1.2 OBJETIVOS

Existem diversos algoritmos (formulações) que permitem reconstruir sinais que foram corrompidos por falhas em canais de transmissão, problemas de aquisição, erros no armazenamento, entre outros. Todos eles possuem especificidades e só podem ser utilizados quando estão verificados alguns condicionalismos.

O principal objetivo deste trabalho é a apresentação de um novo modelo de ensino de algoritmos de reconstrução de sinal e respetivos condicionalismos inerentes a este problema de engenharia. Para tal, desenvolveu-se um protótipo aplicacional, o qual foi denominado de *Signal Processing Interpolation Educational Workbench* (SPIEW), tendo por finalidade melhorar a aprendizagem dos alunos na área da

reconstrução de sinal, recorrendo a um modelo que aborda de forma gráfica e interativa os diversos conceitos versados.

O modelo proposto para o estudo de algoritmos de reconstrução de sinal tem por missão permitir aos seus utilizadores melhor compreender ou aprofundar os conceitos inerentes a esta área de estudo, como por exemplo: limitação em frequência, decimação, condicionamento, interpolação, reconstrução, avaliação de qualidade objetiva/subjetiva, entre outros. Abordando os conceitos e operações de forma gráfica e interativa, o utilizador tem sempre a oportunidade de visualizar o resultado das operações efetuadas, passo a passo, facilitando a compreensão dos conceitos do processamento de sinal mais abstratos, mesmo para utilizadores com conhecimentos limitados na área.

Nos problemas de reconstrução de sinal, na formulação de sistemas de equações lineares, é mais eficiente a escolha do algoritmo para a obtenção de uma solução numérica, fazendo-se uma análise prévia ao condicionamento do problema em estudo [Kincaid02]. A alteração de parâmetros no condicionamento de determinados problemas de engenharia, pode auxiliar na escolha do algoritmo mais eficiente para a sua resolução ou então, caso não esteja garantida uma solução numérica ótima, conseguir alterar o condicionamento. Um objetivo fundamental deste trabalho é conseguir que os alunos ganhem competências nesta área que lhes permitam escolher, de forma sustentada, o algoritmo que melhor se adequa a cada problema em particular de acordo com a estratégia de aprendizagem proposta.

1.3 CONTRIBUIÇÕES

Nesta tese apresenta-se um modelo de ensino do processamento digital de sinal usando como caso de estudo o problema da reconstrução de sinal. Seguidamente apresentam-se os principais contributos deste trabalho:

- Modelo para facilitar e promover o estudo dos algoritmos de reconstrução de sinal.
- Protótipo aplicacional para experimentação ativa de conceitos e algoritmos previamente apresentados de forma expositiva.
- Aprendizagem por observação e *debug*/comportamento dos algoritmos.
- Simulação de cenários de realizações em engenharia.
- Contribuir para o estímulo dos alunos de engenharia para o estudo do processamento digital de sinal.
- Aumentar as competências dos alunos de engenharia nesta área.

- Medir os ganhos comprovados com testes *Mean Opinion Score* (MOS) de eficiência de aprendizagem.

Para corroborar o modelo proposto, implementou-se um protótipo aplicativo que permite uma experimentação interativa de conceitos e algoritmos, previamente apresentados de forma expositiva. Desta forma, os alunos poderão, em nossa opinião, entrelaçar os conceitos teóricos lecionados nas disciplinas do processamento e interpolação de sinal e aferir sobre a sua aplicação prática. Através dos resultados e *feedback* obtidos neste trabalho acreditamos que a possibilidade de dispor de um ambiente de interpretação onde poderão, de acordo com o contexto, serem simulados casos reais, permitirá promover o conhecimento na área.

1.4 METODOLOGIA DE INVESTIGAÇÃO

A experiência adquirida, ao longo de diversos anos letivos, na lecionação de disciplinas relacionadas com o processamento digital de sinal, a alunos de diversas áreas como por exemplo: informática, saúde e música, juntamente com as publicações na área do ensino deram à autora algum conhecimento e capacidade introspectiva que levaram a propor uma abordagem diferente no ensino do processamento digital de sinal, tendo por base o caso de estudo da reconstrução de sinal em engenharia. Tentando de certa forma melhorar a relação dos alunos com esta área de saber. O objetivo é então desenvolver uma metodologia de ensino, distinta da abordagem convencional, que permita aos alunos assimilar conceitos de forma mais suave, mas também dando a possibilidade de os mesmos testarem na prática os conceitos ministrados através de exposição teórica. Fornecendo assim uma aplicabilidade para os conceitos abordados.

Identificado o problema, definiu-se posteriormente uma metodologia de investigação para atingir os objetivos propostos. Foi realizado um estudo conceptual da área e de uma possível solução o que levou à definição de um modelo que irá solucionar o problema identificado. Posteriormente foi contruído um protótipo e mais tarde procedeu-se à validação do mesmo seguindo a metodologia de investigação Design Science (DS), Figura 1.1.

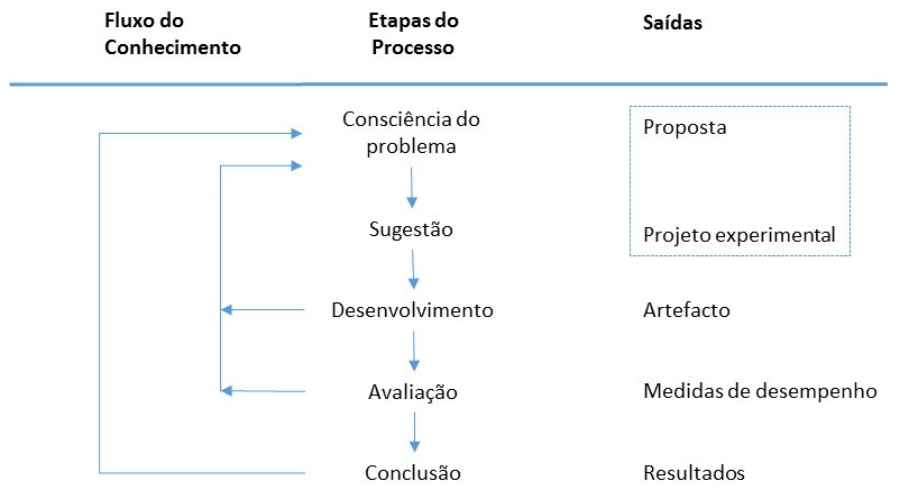


Figura 1.1 – Metodologia *Design Science* (adaptado de [Vaishnavi04])

A metodologia de investigação *Design Science* (DS) tem como principal objetivo desenvolver conhecimento que possa ser utilizado na resolução de problemas [Peffers08]. O termo DS é escolhido para destacar a orientação do novo conhecimento ao *design* e às ferramentas necessárias para ações adequadas. O processo de utilizar conhecimento para planear ou criar um artefacto, quando é cuidadoso, sistemático e rigorosamente analisado sobre a eficácia com que atinge o seu objetivo é denominado de *design research* [Peffers08]. Para desenvolver este trabalho, foi escolhida a abordagem DS, por se tratar do desenvolvimento de um modelo de ensino, que incluiu a implementação de um protótipo aplicacional que serve de suporte ao modelo de ensino proposto.

O método *Design Science* é composto por cinco fases principais: sensibilização para o problema, sugestão, desenvolvimento, avaliação e conclusão [Hevner04]. A primeira fase do processo consiste na sensibilização do problema onde está presente a deficiência de conhecimento, e tem por objetivo aumentar a familiarização sobre o problema. O resultado desta fase é a descrição do tópico alvo de investigação. O passo seguinte refere-se à sugestão da solução. Depois de aprofundados os conhecimentos na área, o resultado obtido após finalizada a revisão da literatura permite ao investigador ultrapassar as lacunas de conhecimento existentes na fase inicial, e servirão de suporte teórico para a investigação. A próxima fase é o desenvolvimento da solução, e é aqui que está centrada a maior parte da investigação, tendo como resultados final o artefacto corretamente definido. A fase de avaliação refere-se à validação do artefacto. Finalmente, a conclusão, e última fase do método, é onde termina o ciclo de conceção do artefacto, avaliando a qualidade e impacto do artefacto produzido.

Esta abordagem foi realizada focando as orientações estabelecidas por [Hevner04]: o artefacto, relevância do problema, rigor na investigação, *design* como um processo de investigação, avaliação do *design*, contribuições da investigação e comunicação da investigação.

Artefacto – Desenvolvimento de um novo modelo para o ensino de algoritmos de reconstrução de sinal, não apenas com exposição teórica de conceitos, mas também criar um protótipo de experimentação ativa, que permita testar situações reais em engenharia. Para implementar o novo modelo realizou-se um protótipo de uma aplicação, o SPIEW (*Signal Processing Interpolation Educational Workbench*), que fornece aos seus utilizadores uma ferramenta de fácil uso e que permite a aquisição ou consolidação de conceitos de processamento digital de sinal. Um dos pontos fortes do SPIEW é que este permite a visualização gráfica (associada a uma série de recursos desenvolvidos para acompanhar as várias fases do processo conhecimento sobre o problema) do resultado de diversas operações típicas do processamento digital de sinal.

Relevância do Problema – No caso específico dos algoritmos de reconstrução de sinal, o modelo de ensino utilizado é o típico, exposição teórica de conceitos e resolução de exercícios. O objetivo é complementar o ensino típico de conceitos com experiências concretas onde os alunos possam comprovar e testar os conceitos teóricos previamente ministrados.

Rigor na Investigação – Todos os resultados, métodos e protótipo desenvolvido são analisados e revistos diversas vezes, de modo a aumentar credibilidade e valor nas conclusões retiradas. Todos os desenvolvimentos são documentados.

Processo de pesquisa e avaliação do design, contribuições e comunicação da investigação – Criação de um protótipo aplicativo que serve de suporte ao novo modelo de ensino proposto. Para além da análise objetiva com métricas apropriadas e comuns dos algoritmos utilizados, a validação do protótipo foi feita através de testes subjetivos (MOS) e os resultados apresentados no capítulo 7 desta tese permitem igualmente atestar a utilidade e eficácia da ferramenta desenvolvida. Foram efetuados diversos testes ao SPIEW que implementa o ciclo de aprendizagem descrito em [Stice87; Abdulwahed09; Konak13]. O SPIEW foi testado em grupos de alunos com conhecimento distinto na área do processamento digital de sinal, e provenientes de diferentes níveis de ensino: licenciatura, mestrado e doutoramento. A ferramenta foi igualmente testada por professores. Os resultados obtidos nestes testes foram publicados em conferência e em revistas, e podem ser consultados em [Costa09; Costa13a; Costa13b].

1.5 ORGANIZAÇÃO DO TRABALHO

Neste primeiro capítulo, foi feito o enquadramento do tema de trabalho, indicando qual a motivação, objetivos, contribuições e metodologia utilizada do trabalho desenvolvido.

O segundo capítulo é constituído por definições e tópicos de Álgebra Linear e de Análise Numérica usados ao longo de todo o trabalho. É igualmente feita uma revisão de métodos diretos, iterativos e semi-iterativos, sem e com pré-condicionamento, para a resolução de sistemas de equações lineares.

No terceiro capítulo são apresentados os algoritmos de reconstrução de sinal que serviram de teste para o modelo de ensino proposto.

No quarto capítulo é apresentado o novo modelo para o ensino de algoritmos de reconstrução de sinal, especificando a metodologia de desenvolvimento e o ciclo de aprendizagem implementado.

No quinto capítulo cinco é descrito o protótipo aplicativo desenvolvido que materializa a experimentação interativa do modelo de ensino proposto no capítulo quatro.

No sexto capítulo são apresentados os resultados da avaliação subjetiva do modelo de ensino proposto através da realização de testes *Mean Opinion Score* (MOS).

Finalmente, no sétimo e último capítulo, apresentam-se as conclusões do trabalho realizado, apontando algumas direções possíveis para futuros desenvolvimentos e extensão do modelo aqui apresentado.

1.6 PUBLICAÇÕES DO TRABALHO

O trabalho descrito neste documento resultou de uma investigação cujos resultados foram alvo de várias publicações em revistas e conferências internacionais científicas. Enumeram-se nos parágrafos seguintes essas publicações bem como a principal contribuição para a tese proposta.

- Mónica Costa, Salviano Soares, João Barroso, (2010): ***“Reconstruction of High Frame Rate Image Sequences in Biomechanical Related Areas”***, 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, August 31 - September 4, 2010. Pp: 9 – 12, ISBN: 978-1-4244-4124-2/10.

Numa fase inicial do trabalho foram identificadas diversas áreas onde são utilizadas imagens de elevada resolução temporal para analisar e compreender determinados fenómenos, como é o caso da engenharia biomédica. Este artigo propõe uma metodologia para otimizar o espaço

necessário para armazenar sequências de imagens de elevada resolução temporal no caso específico da área da biomecânica. O algoritmo garante a recuperação da informação em conformidade com a QoS pretendida.

- M. Costa, J. Barroso, S. Soares, (2012): *“Signal Processing Interpolation Educational Workbench”*, Comput Application in Engineering Education 20: 356-363, 2012; DOI 10.1002/cae.20402. (Article first published online: 21 December 2009)

Este artigo descreve a primeira versão de uma ferramenta educacional para ser usada em disciplinas que abordam os temas do processamento e da interpolação de sinal. O objetivo foi contribuir para uma melhor consolidação dos conhecimentos teóricos adquiridos, permitindo assim que os estudantes testassem algoritmos de reconstrução de sinal e visualizassem os seus resultados. Pretendeu-se igualmente que os alunos pudessem aferir de que forma os diversos parâmetros afetam a convergência e desempenho desses algoritmos. Começou-se por disponibilizar no início apenas um algoritmo e uma geometria de padrão de erros para efetuar a corrupção dos sinais utilizados. Daqui surgiu a ideia de criar uma nova metodologia para o ensino da reconstrução de sinal, abordando o tema de forma mais iterativa.

- Mónica Isabel T. da Costa, Simão Cardeal, Salviano Soares, João Barroso, (2013): *“Problem Conditioning Interpolation Educational Tool”*, IEEE Global Engineering Education Conference (EDUCON), Berlin, Germany, 2013. ISBN: 978-1-4673-6109-5 /13. pp:996-1003.

Tendo por base o ciclo de aprendizagem escolhido neste trabalho, este artigo descreve a nova versão da ferramenta educacional para o ensino da reconstrução de sinal em engenharia. A esta nova versão, e tendo em conta o *feedback* obtido nos testes da versão anterior, foram adicionadas novas funcionalidades e novos algoritmos de reconstrução. Com esta nova versão, deu-se também aos utilizadores a possibilidade de analisar o condicionamento do problema para posteriormente optar pelo algoritmo de reconstrução mais favorável.

- Mónica Isabel T. da Costa, Salviano Soares, João Barroso, (2013): *“Signal Processing Interpolation and Problem Conditioning Educational Workbench”*, Modern Applied Science; Vol. 7, No. 12; 2013, Online Published: November 18, 2013 doi:10.5539/mas.v7n12p56.

Neste trabalho apresentaram-se também, para além da metodologia, resultados preliminares da avaliação à robustez da ferramenta proposta.

Capítulo 2 – Resolução de Sistemas de Equações Lineares

A análise numérica envolve o estudo, desenvolvimento e análise de algoritmos para obter soluções numéricas para diversos problemas matemáticos. Esta área de conhecimento é genericamente denominada de matemática para computação científica [Kincaid02].

Os algoritmos estudados na análise numérica são destinados a ser utilizados em computadores de alto desempenho, no entanto, antes mesmo de ser obtida uma solução para o problema, é necessário escrever o código ou programa do respectivo algoritmo para que o computador o possa executar e assim obter a solução desejada.

Frequentemente, o desenvolvimento de um algoritmo é estimulado por uma prova matemática construtiva [Kincaid02]. Na análise clássica, os métodos não-construtivos são frequentemente utilizados mas, geralmente, não resultam em algoritmos. No entanto, nem todas as provas construtivas resultarão num algoritmo de sucesso. Pode ocorrer que a solução analítica de um determinado problema possa estar a vários passos de distância de uma solução numérica. Ou então esta pode mesmo ser impraticável devido a uma convergência lenta ou à necessidade de um tempo de computação muito elevado.

Em cálculos numéricos, especialmente em computadores de alto desempenho, acontece frequentemente que a resposta a um problema não é obtida de uma só vez. É frequente obter-se uma sequência de respostas aproximadas, que geralmente exibem uma precisão progressivamente mais elevada [Kincaid02].

Neste capítulo será abordado o tópico para a resolução de problemas na formulação $Ax = b$. O principal objetivo deste capítulo é discutir os aspetos numéricos da resolução de sistemas de equações lineares na forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases} \quad (2.1)$$

O sistema apresentado previamente é um sistema de n equações e n incógnitas: x_1, x_2, \dots, x_n . Os elementos a_{ij} são os coeficientes e b_j são os termos independentes do sistema de equações.

As matrizes são elementos bastante úteis na representação de sistemas de equações. Sendo assim, o sistema de equações lineares descrito anteriormente (Eq 2.1) pode ser escrito da seguinte forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}. \quad (2.2)$$

Considerando \mathbf{A} a matriz dos coeficientes de dimensão $n \times n$, \mathbf{x} o vetor das incógnitas e \mathbf{b} o vetor dos termos independentes, podemos reescrever a equação anterior (Eq. 2.2) na forma $Ax = b$.

A resolução numérica de um sistema de equações lineares é obtida geralmente por duas metodologias: métodos diretos e métodos iterativos. Os métodos diretos determinam as soluções de um sistema linear com um número finito de operações, sendo que estes são baseados no processo de escalonamento, que consiste em resolver sistemas de pequena dimensão. Os métodos iterativos conduzem à solução aproximada de um sistema de equações lineares a partir de uma estimativa inicial da solução através de um procedimento iterativo. A cada iteração utiliza-se a aproximação anterior da solução para o cálculo de uma nova aproximação, a qual deve convergir para a solução do sistema, sendo estes métodos indicados para solucionar sistemas grandes e esparsos.

Os métodos diretos são aqueles que, supondo não haver erros de arredondamento ou quaisquer outros, nos permitem obter a solução exata do problema num número finito de operações aritméticas. Os métodos iterativos são sobretudo usados para matrizes esparsas de grandes dimensões. Neste tipo de problemas os métodos iterativos são competitivos face aos métodos diretos. Para matrizes densas ou de pequena dimensão os métodos diretos são mais vantajosos [Araújo13].

Existe ainda uma outra classe de métodos, os semi-iterativos, que são utilizados na resolução de sistemas lineares quando as matrizes são esparsas não estruturadas e de tamanho médio [Kincaid02] [Golub96]. Na secção 2.6, são descritos alguns exemplos destes métodos.

Nas secções seguintes deste capítulo serão descritos métodos diretos, iterativos e semi-iterativos que permitem a resolução de sistemas de equações lineares na forma $Ax = b$.

2.1 ÁLGEBRA DE MATRIZES: CONCEITOS

Nesta secção serão revistos conceitos básicos da teoria de matrizes. Outros conceitos serão introduzidos posteriormente e sempre que for necessário. Uma matriz é um conjunto de números dispostos em linhas e colunas, um escalar é um caso especial e contém apenas uma linha e uma coluna. A matriz poderá conter várias linhas ou colunas, mas existe também a possibilidade de a mesma conter apenas elementos numa linha ou numa coluna, como por exemplo:

$$X = \begin{bmatrix} 1 & 2 & 5 \\ 0.2 & 2.5 & 10 \\ 11 & -3.0 & 4 \end{bmatrix} \quad Y = [3 \quad 6 \quad 11/7 \quad -18] \quad Z = \begin{bmatrix} -12.4 \\ 5 \end{bmatrix}.$$

A dimensão das matrizes apresentadas anteriormente é respetivamente: 3×3 , 1×4 e finalmente 2×1 . Quando definimos a dimensão de uma matriz, em primeiro lugar, é indicado o número de **linhas** (linhas horizontais) e em segundo lugar, é apresentado o número de **colunas** (linhas verticais). No caso específico de uma matriz com dimensão $1 \times n$, esta é igualmente denominada de **vetor linha**. Já uma matriz $m \times 1$ é simplesmente designada de **vetor coluna** ou simplesmente **vetor** [Kincaid02] [Pina10].

Se A é uma matriz, a notação a_{ij} , $(A)_{ij}$, ou $A(i, j)$ é utilizada para identificar o elemento que está situado na intersecção da linha i com a coluna j . Considerando a primeira matriz, matriz X , $X(2,1) = 0.2$

A **transposta** de uma matriz é denominada A^T , e é definida por $(A^T)_{ij} = a_{ji}$ [Kincaid02]. Considerando novamente a matriz X a sua transposta é

$$X^T = \begin{bmatrix} 1 & 0.2 & 11 \\ 2 & 2.5 & -3.0 \\ 5 & 10 & 4 \end{bmatrix}.$$

Se a transposta de uma matriz for igual à própria matriz, ou seja, se $A^T = A$, então diz-se que a matriz A é **simétrica**. Por outro lado, se a matriz A for igual à sua transposta conjugada, ou seja, $A = \bar{A}^T$, então estamos perante uma matriz **Hermítica** ou **Hermitiana** [Kincaid02] [Pina10].

Uma vez que estamos a lidar com sistemas de equações lineares, o conceito de equivalência é igualmente importante. Considerando assim, dois sistemas de n equações e n incógnitas com as seguintes equações:

$$Ax = b$$

$$Bx = d$$

Se ambos os sistemas tiverem exatamente as mesmas soluções, são denominados de sistemas **equivalentes**. Assim, para resolver um qualquer sistema de equações, podemos resolver um dos seus

equivalentes, não sendo omitido ou acrescentado qualquer valor à solução. Esta ideia simples é o cerne dos procedimentos numéricos. Dado um sistema de equações do qual se pretende obter a solução, podemos transformá-lo, através de determinadas operações elementares, num sistema equivalente mais simples [Kincaid02]. As **operações elementares** são dos três tipos apresentados a seguir. (Onde ε_i representa a equação i do sistema)

1. Permutar duas equações do sistema: $\varepsilon_i \leftrightarrow \varepsilon_j$
2. Multiplicar a equação por um valor não nulo: $\lambda\varepsilon_i \rightarrow \varepsilon_i$
3. Adicionar a uma equação o múltiplo de uma outra equação: $\varepsilon_i + \lambda\varepsilon_j \rightarrow \varepsilon_i$

Teorema dos Sistemas Equivalentes – Se um sistema de equações é obtido através de outro após uma série de operações elementares, estão diz-se que os dois sistemas são equivalentes. [Kincaid02]

2.2 PROPRIEDADES DAS MATRIZES

A seguinte matriz de dimensão $n \times n$ é denominada **matriz identidade**.

$$I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Esta matriz possui a seguinte propriedade $IA = A = AI$, isto para qualquer matriz A de dimensão $n \times n$.

Se A e B são duas matrizes tais que, $AB = I$, então diz-se que B é a **inversa direita** de A e A é a **inversa esquerda** de B .

Teorema da Inversa Direita – Uma matriz quadrada pode possuir no máximo uma inversa direita. [Kincaid02]

Teorema da Matriz Inversa – Se A e B são matrizes quadradas tais que $AB = I$, então $BA = I$. [Kincaid02]

Através dos dois teoremas anteriores conclui-se que, se uma matriz quadrada A possui como inversa direita a matriz B , então B é única e $BA = AB = I$. Chamamos a B **inversa** de A , o que indica que A é **invertível** ou **não-singular**. Claramente, B é invertível e A é a sua inversa. Podemos expressar estas relações da forma $B = A^{-1}$ e $A = B^{-1}$. Um exemplo é apresentado a seguir.

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} -1.5 & 2.5 & 0.5 \\ 2 & -2 & -1 \\ -0.5 & 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} -1.5 & 2.5 & 0.5 \\ 2 & -2 & -1 \\ -0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 0 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Se A é invertível então, o sistema de equações $Ax = b$ possui a solução $x = A^{-1}b$. Se A^{-1} já está disponível, esta equação fornece um bom método para o cálculo de x . Caso A^{-1} não esteja disponível, então o seu cálculo não deve ser efetuado apenas com o propósito de obter x . Outros métodos serão apresentados nas próximas secções deste capítulo.

As operações elementares descritas na secção anterior poderão ser implementadas através da multiplicação de matrizes. Uma **matriz elementar** é definida como uma matriz de dimensão $n \times n$ que advém do resultado de uma operação elementar com a matriz identidade de dimensão $n \times n$. As operações elementares, expressas em termos de linhas da matriz A , são [Kincaid02]:

1. Permutar duas linhas da matriz A : $A_s \leftrightarrow A_t$
2. Multiplicar uma linha por um elemento não nulo: $\lambda A_s \rightarrow A_s$
3. Adicionar a uma linha o múltiplo de outra: $A_s + \lambda A_t \rightarrow A_s$

Anteriormente, foram identificadas as linhas de A pelas variáveis A_s e A_t . Cada operação elementar sobre as linhas de A é obtida através da multiplicação à esquerda de A por uma matriz elementar.

Se quisermos aplicar uma sequência de operações elementares sobre as linhas de A , introduzimos as matrizes elementares E_1, E_2, \dots, E_m e escrevemos a matriz de transformação como $E_m E_{m-1} \dots E_2 E_1 A$.

Se uma matriz é invertível, essa sequência de operações elementares sobre as linhas pode ser aplicada a A , transformando-a em I , e assim temos $E_m E_{m-1} \dots E_2 E_1 A = I$.

Podemos assim afirmar que $A^{-1} = E_m E_{m-1} \dots E_2 E_1$. Consequentemente, A^{-1} pode ser obtida submetendo I à mesma sequência de operações elementares sobre as linhas. De seguida é apresentado um exemplo do cálculo de uma inversa.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

$$E_1 A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 2 & 4 & 7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = E_1 I$$

$$E_1 E_2 A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_1 E_2 I$$

$$E_1 E_2 E_3 A = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & -2 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_1 E_2 E_3 I$$

$$E_1 E_2 E_3 E_4 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_1 E_2 E_3 E_4 I$$

Neste caso as matrizes elementares são:

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$E_3 = \begin{bmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad E_4 = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Teorema das Propriedades de uma Matriz Não-Singular – Para uma matriz A de dimensão $n \times n$ as seguintes propriedades são equivalentes. [Kincaid02]

1. A inversa de A existe, então A é não-singular.
2. O determinante de A é não-zero.
3. As linhas de A formam uma base para \mathbb{R}^n .
4. As colunas de A formam uma base para \mathbb{R}^n .
5. Como mapa de \mathbb{R}^n para \mathbb{R}^n , A é injetiva (um para um).
6. Como mapa de \mathbb{R}^n para \mathbb{R}^n , A é sobrejetiva.
7. A equação $Ax = 0$ implica $x = 0$.
8. Para cada $b \in \mathbb{R}^n$, existe apenas um $x \in \mathbb{R}^n$ tal que $Ax = b$.
9. A é o produto de matrizes elementares.
10. 0 não é um valor próprio de A .

Um conceito importante em álgebra é a noção de matriz positiva definida. Uma matriz A diz-se **positiva definida** se a forma quadrática associada é positiva, ou seja, $x^T Ax > 0$, para qualquer vetor x não-nulo [kincaid02] [Rodrigues03]. Por exemplo a matriz:

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

é positiva definida desde que

$$x^T Ax = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = (x_1 + x_2)^2 + x_1^2 + x_2^2 > 0$$

para todo x_1 e x_2 exceto $x_1 = x_2 = 0$. Determinar se uma matriz é positiva definida, através da definição, é normalmente uma tarefa complicada uma vez que envolve um $x \neq 0$ arbitrário. Se A é positiva definida e simétrica, então os seus valores próprios são reais e positivos ($\lambda > 0$). [Kincaid02]

Norma Vetorial – Num espaço vetorial V , uma norma é uma função $\| \cdot \|$ de V para um conjunto de reais não negativos que satisfaz a:

$$\circ \quad \|x\| \geq 0 \text{ se } x \neq 0, x \in V, \tag{2.3}$$

$$\circ \quad \| \alpha x \| = |\alpha| \| x \| \text{ se } \alpha \in \mathbb{R}, x \in V, \quad (2.4)$$

$$\circ \quad \| x + y \| \leq \| x \| + \| y \| \text{ se } x, y \in V. \quad (2.5)$$

Podemos entender a função $\|x\|$ como sendo o tamanho ou comprimento do vetor x . Num espaço vetorial a norma de um vetor $\| \cdot \|$ generaliza o conceito de valor absoluto $| \cdot |$. Assim é possível definir “distância” entre dois elementos de um espaço normado $d(a, b) = \|a - b\|$, ou seja, $d(\cdot)$ é a distância induzida pela norma Euclideana, a que mais se aproxima ao conceito intuitivo de comprimento de um vetor,

$$\|x\|_e = \sqrt{\sum_{i=1}^N x_i^2}, x = [x_1, x_2, \dots, x_N]^T.$$

Para vetores existem normas alternativas designadas por normas l_p

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^N |x_i|^p}.$$

Entre estas, as mais utilizadas são

$$\|x\|_\infty = \max_{1 \leq i \leq N} |x_i|$$

e

$$\|x\|_1 = \sum_{i=1}^N |x_i|, l_\infty \text{ e } l_1 \text{ respectivamente. De notar que para vetores } \|x\|_2 \text{ e } \|x\|_e.$$

Norma Matricial – Podem estabelecer-se funções norma matricial a partir dos mesmos pressupostos que os indicados para a norma de vetores (2.3) a (2.5). No entanto, tem neste caso tem mais utilidade definir uma norma matricial relacionada com a norma vetorial, o que está mais de acordo com a interpretação da matriz como operador linear.

Norma matricial induzida por uma vetorial – Definição [Kincaid02]:

Admita-se uma matriz A de dimensão $N \times N$, e uma função $\| \cdot \|$ norma em \mathbb{R}^N . A equação

$$\|A\| = \sup_{\|u\|=1} \{\|Au\| : u \in \mathbb{R}^N\}$$

define a norma no espaço linear de todas as matrizes $N \times N$.

Desta definição surge uma relação importante,

$$\|Ax\| \leq \|A\| \|x\|, x \in \mathbb{R}^N. \quad (2.6)$$

Esta expressão é verdadeira para $x = 0$. Se $x \neq 0$, o vetor $v = x / \|x\|$ tem norma unitária. Assim, temos

$$\|A\| \geq \|Av\| = \|Ax\| / \|x\|.$$

Para ilustrar este conceito, vamos calcular a norma matricial induzida pela vetorial l_∞ .

$$\begin{aligned} \|A\|_\infty &= \sup_{\|u\|_\infty=1} \|Au\|_\infty = \sup_{\|u\|_\infty=1} \{ \max_{1 \leq i \leq N} |(Au)_i| \} = \max_{1 \leq i \leq N} \{ \sup_{\|u\|_\infty=1} |(Au)_i| \} \\ &= \max_{1 \leq i \leq N} \{ \sup_{\|u\|_\infty=1} |\sum_{j=1}^N a_{ij}u_j| \} = \max_{1 \leq i \leq N} \sum_{j=1}^N |a_{ij}|. \end{aligned}$$

Considerou-se que o supremo de $|\sum_{j=1}^N a_{ij}u_j|$ para um i fixo e $\|u\|_\infty = 1$, é obtido colocando $u_j = +1$, se $a_{ij} \geq 0$ e $u_j = -1$, se $a_{ij} < 0$ [Kincaid02].

Uma função norma matricial induzida por uma vetorial tem propriedades adicionais áquelas defindas por (2.3) a (2.5). Assim a partir da definição e de 2.6 podemos escrever

$$\|AB\| \leq \|A\| \|B\|.$$

2.3 SISTEMA DE EQUAÇÕES BEM E MAL CONDICIONADOS

Um sistema de equações é considerado **bem condicionado** quando uma pequena alteração nos coeficientes da matriz do sistema ou na matriz dos coeficientes independentes resulta numa pequena variação no vetor solução [Kincaid02] [Rodrigues03].

Um sistema de equações diz-se **mal condicionado** quando uma pequena alteração nos coeficientes da matriz do sistema ou na matriz dos coeficientes independentes resulta numa grande variação no vetor solução [Kincaid02] [Rodrigues03].

Toda a matriz invertível possui um **número de condição**. Analisando esse número simultaneamente com a **resolução da máquina**, podemos quantificar o número de algarismos significativos que podemos confiar na solução obtida.

Para calcular o número de condição de uma matriz invertível é normalmente necessário calcular a **norma** da matriz [Kincaid02] [Rodrigues03]. A norma de uma matriz é um número escalar único. Este valor é sempre positivo e é definido para todas as matrizes: quadradas ou retangulares, e matrizes quadradas invertíveis ou não-invertíveis. Uma definição bastante popular da norma é a norma da soma das linhas, também denominada de norma uniforme de uma matriz, que foi definida na secção 2.2 deste documento.

Considerando o sistema $Ax = b$, com $\det(A) \neq 0$, cuja solução é x' . Supondo que conhecemos os coeficientes da matriz A mas, em vez de se utilizar o vetor b , vamos utilizar o vetor perturbado

$$\tilde{b} = b + \delta b \quad (2.7)$$

para resolver o sistema, ou seja:

$$Ax = \tilde{b}. \quad (2.8)$$

Representando agora por $x' + \delta x$ a solução do sistema $Ax = \tilde{b}$. O objetivo é medir a diferença δx em função de δb , ou seja, pretende-se estabelecer uma majoração para o erro δx , usando δb , da forma:

$$\|\delta x\| \leq C \|\delta b\| \quad (2.9)$$

onde C é uma constante positiva.

Assim, temos $A(x' + \delta x) = b + \delta b$ e $Ax' = b$, de onde $A\delta x = \delta b$. Como $\det(A) \neq 0$, podemos escrever que $\delta x = A^{-1}\delta b$. Aplicando normas a ambos os lados desta igualdade obtém-se:

$$\|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\|. \quad (2.10)$$

Daqui concluí-se que é o valor de $\|A^{-1}\|$ que determina a influência dos efeitos de $\|\delta b\|$ na solução do sistema [Rorigues03].

Uma vez que, $\frac{\|\delta x\|}{\|x\|}$ designa o erro relativo da solução e, $\frac{\|\delta b\|}{\|b\|}$ designa a perturbação relativa do vetor b [Rorigues03], vamos agora estabelecer uma relação entre eles. Na equação (2.10) vamos multiplicar ambos os membros por $\frac{1}{\|x\|}$, obtendo assim:

$$\frac{\|\delta x\|}{\|x\|} \leq \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|x\|}. \quad (2.11)$$

Por outro lado, de $Ax = b$, concluí-se que $\|b\| = \|Ax\| \leq \|A\| \cdot \|x\|$, logo $\|x\| \geq \frac{\|b\|}{\|A\|}$. Assim,

$$\frac{\|\delta x\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\| \frac{\|\delta b\|}{\|b\|}. \quad (2.12)$$

O número $\|A\|\|A^{-1}\|$ é o **número de condição** da matriz A , ou $k(A)$. O **número de condição** de um problema é uma medida que indica se o problema está bem condicionado para ser tratado numericamente. Um problema com um número de condição pequeno é designado de **bem condicionado**. Os problemas que possuem um número de condição elevado são denominados **mal condicionados**. Na realidade se $k(A) \approx 10^d$ então cerca de d dígitos da solução poderão ser questionados [Kincaid02].

O número de condição associado a um sistema linear do tipo $Ax = b$ é um número que estabelece uma estimativa da precisão que se pode obter para uma solução aproximada de x . De referir que este resultado surge antes dos efeitos dos erros de arredondamento serem tomados em consideração. O

condicionamento é uma propriedade da matriz, não do algoritmo ou da precisão em ponto flutuante do computador usado para resolver o sistema correspondente.

Propriedades do número $k(A)$ [Rodrigues03]:

1. $k(A) \geq 1$;
2. $k(A) \geq \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}$, onde $\lambda_{\max}(A)$ é o valor próprio da matriz A com o módulo máximo, $\lambda_{\min}(A)$ é o valor próprio da matriz A com o módulo mínimo;
3. $k(AB) \leq k(A) \cdot k(B)$.

Se u for o vetor próprio da matriz A correspondente ao valor próprio máximo:

$$Au = \lambda_{\max}(A)u,$$

então

$$\|Au\| = |\lambda_{\max}(A)| \cdot \|u\|.$$

Atendendo a que $\|Au\| \leq \|A\| \cdot \|u\|$, podemos concluir que $\lambda_{\max}(A) \leq \|A\|$.

Considerando agora a matriz A^{-1} , sabe-se que

$$\lambda_{\max}(A^{-1}) = \frac{1}{\lambda_{\min}(A)},$$

pelo que se pode concluir que

$$\lambda_{\max}(A^{-1}) = \frac{1}{\lambda_{\min}(A)} \leq \|A^{-1}\|.$$

Consequentemente,

$$k(A) = \|A\| \cdot \|A^{-1}\| \geq \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}. \quad (2.13)$$

2.4 MÉTODOS DIRETOS

Considerando um sistema linear de n equações e n incógnitas descrito pela equação (Eq. 2.2). As matrizes desta equação são definidas como A , x e b . E o sistema pode ser escrito da forma:

$$Ax = b. \quad (2.14)$$

Supondo que $\det(A) \neq 0$, uma solução do sistema (Eq. 2.14) é o vetor $x^* \in \mathbb{R}^n$, que é o único para o qual se verifica a igualdade $Ax^* = b$.

Observe-se que se a matriz for triangular existem formas mais expeditas para obter a solução do problema. Vamos então supor que o sistema de equações lineares tem a seguinte forma especial:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{nn}x_n = b_n \end{cases}$$

onde $a_{ii} \neq 0, i = 1, 2, \dots, n$. Um sistema deste tipo diz-se **triangular superior**. A resolução deste sistema é simples e pode efetuar-se utilizando o seguinte método:

$$x_n = \frac{b_n}{a_{nn}}$$

$$x_i = (b_i - \sum_{j=i+1}^n a_{ij}x_j) / a_{ij} \quad i = n-1, n-2, \dots, 1 \quad (2.15)$$

Este método é denominado das substituições ascendentes ou regressivas [Rodrigues03]. O número de operações de multiplicação e divisão necessárias neste método é dado por:

$$\sum_{i=1}^n (n-i) = \frac{n(n-1)}{2} \approx O(n^2).$$

Diz-se que a função definida por $g(h)$ é da ordem de $O(h^p)$ quando $h \rightarrow a$, se existe uma constante M tal que:

$$\lim_{h \rightarrow a} \frac{g(h)}{h^p} < M.$$

O método clássico para resolver os sistemas de equações lineares na forma geral é o método de eliminação de Gauss.

2.4.1 Método de Gauss

Para um sistema genérico de dimensão $n \times n$, o método de eliminação de Gauss tem os mesmos passos. Considerando agora o sistema descrito na equação (2.1), o primeiro passo do método consiste na eliminação de x_1 da segunda à n -ésima equação, obtendo assim o sistema:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases} \quad (2.16)$$

onde

$$a_{ij}^{(1)} = a_{ij} - a_{1j} \frac{a_{i1}}{a_{11}}, \quad b_i^{(1)} = b_i - b_1 \frac{a_{i1}}{a_{11}}, \quad i, j = 2, \dots, n \quad (2.17)$$

Estas operações denominam-se transformações elementares da matriz e ao coeficiente a_{11} chama-se *pivot* das transformações. De seguida repete-se o processo com o subsistema de ordem $(n-1) \times (n-1)$ para eliminar a incógnita x_2 das últimas $(n-3)$ equações.

$$\begin{cases} a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases}$$

No final de $(n - 1)$ passos destes obtemos:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \end{cases}$$

O índice superior nos coeficientes indica o número de vezes que os coeficientes foram alterados. Nestas condições termina-se o processo de triangulação. A solução do sistema pode ser agora obtida pelo método das substituições ascendentes.

No que se refere ao número de operações necessário para aplicar este método, o cálculo da matriz na forma triangular superior requer $\frac{(n^2-1)n}{3}$ ou $O\left(\frac{n^3}{3}\right)$ de multiplicações e divisões no total, assim como o processo das substituições ascendentes requer $O(n^2)$ multiplicações. Sendo assim, o número total de operações do método de Gauss é

$$O\left(\frac{n^3}{3}\right) + O(n^2) = O\left(\frac{n^3}{3}\right),$$

para um n suficientemente grande [Rodrigues03].

2.4.2 Fatorização LU

Vamos supor que a matriz A pode ser fatorizada no produto de uma matriz triangular inferior L e uma matriz triangular superior U , ou seja

$$A = LU. \tag{2.18}$$

Para resolver o sistema de equações $Ax = b$, podemos dividir a sua resolução em duas etapas:

$$Ly = b \quad \text{para obter } y.$$

$$Ux = y \quad \text{para obter } x.$$

Resolver os dois sistemas triangulares anteriores é regra geral simples [Kincaid02]. De seguida vamos ver como se pode obter a fatorização $A = LU$, desde que divisões por zero em determinados passos não sejam encontradas. Nem todas as matrizes podem ser fatorizadas desta forma, casos mais específicos e soluções alternativas podem ser consultadas em [Rodrigues03], [Kincaid02] e [Pina10].

Vamos supor então uma matriz A de dimensão $n \times n$ da qual se pretende obter as matrizes L e U tais que $A = LU$.

$$L = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}$$

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Quando esta operação é possível diz-se que A possui uma **decomposição LU**. Para efetuar esta decomposição existem diferentes abordagens ao método. O **algoritmo de Doolittle** quando a matriz L é triangular inferior com um na diagonal principal ($l_{ii} = 1$ para $1 \leq i \leq n$), e o **algoritmo de Crout** quando U é triangular superior com um na diagonal principal ($u_{ii} = 1$ para $1 \leq i \leq n$). Quando $U = L^T$ e $l_{ii} = u_{ii}$ para $1 \leq i \leq n$, o método é denominado **algoritmo de Cholesky**. Este método será abordado numa secção à parte, uma vez que requer que a matriz A possua determinadas propriedades: seja real, simétrica e a sua forma quadrática associada seja positiva (positiva definida) [Kincaid02].

Como se pode constatar, a fatorização da matriz A é independente do vetor b . Do ponto de vista da resolução de sistemas de equações lineares esta fatorização permite escrever:

$$Ax = b \Leftrightarrow (LU)x = b \Leftrightarrow L(Ux) = b.$$

Assim, introduzindo o vetor auxiliar $y \in \mathbb{R}^n$, o sistema $Ax = b$ é equivalente aos sistemas triangulares inferior e superior $Ly = b$ e $Ux = y$, que se resolvem por métodos de substituições descendentes e ascendentes, respectivamente.

Nestas condições, a resolução do sistema $Ax = b$ pelo método de Gauss consistem em:

1. Fatorizar a matriz A (para obter as matrizes L e U);
2. Resolver o sistema $Ly = b$;
3. Resolver o sistema $Ux = y$.

Deste modo, se tivermos alguns sistemas $Ax = b^k$, $k = 1, \dots, M$, com a mesma matriz A e os vetores b^k diferentes, esta modificação do método de Gauss, conhecida por método de Doolittle, torna-se muito eficiente, pois é possível fazer apenas uma única vez a fatorização da matriz $A = LU$ e seguidamente resolver apenas os sistemas triangulares $Ly = b$ e $Ux = y$.

De seguida apresenta-se em pseudo-código do método de Doolittle retirado de [Rodrigues03].

Método de Doolittle

- ◆ Factorização da matriz $A = (a_{ij})_{i,j=1,\dots,n}$

- ◆ Ciclo das linhas: $i = 1$ a n

Ciclo das colunas: $j = 1$ a $i - 1$

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) / u_{jj}$$

Fim de ciclo j

$$l_{ii} = 1$$

$$u_{ii} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{ki}$$

Ciclo das colunas: $i = j + 1$ a n

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

Fim de ciclo j

Fim de ciclo i

- ◆ Resolução do sistema $Ax = b$

Ciclo das linhas: $i = 1$ a n

$$y_i = \left(b_i - \sum_{k=1}^{i-1} l_{ik} y_k \right)$$

Fim de ciclo ii

Ciclo das linhas: $i = n$ a 1

$$x_i = \left(y_i - \sum_{k=i+1}^n u_{ki} x_k \right) / u_{ii}$$

Fim de ciclo i

2.4.3 Cálculo da Matriz Inversa

O método de eliminação de Gauss é igualmente vantajoso quando pretendemos calcular a inversa da matriz A , ou seja A^{-1} . Considerando $e_i = (0, \dots, 1, \dots, 0)$ um vetor que tem a coordenada i igual a um e as restantes nulas, e_i por ser interpretado como a i -ésima coluna da matriz identidade. Consequentemente a solução do sistema $Ax = e_k$, será a k -ésima coluna da matriz A^{-1} , designada por v_k , atendendo que $AA^{-1} = I$.

Desta forma, a matriz A^{-1} pode ser obtida, coluna a coluna, através da resolução dos n sistemas $Ax = e_k, k = 1, \dots, n$.

Este problema pode ser resolvido efetivamente pela seguinte sucessão das operações:

1. Fatorização $A = LU$;
2. Resolução do sistema $Ly_k = b_k, k = 1, \dots, n$;
3. Resolução do sistema $Ux_k = y_k, k = 1, \dots, n$.

A primeira operação é feita uma vez. As outras são repetidas m vezes. Então, o número das operações para resolver m sistemas é $O(n^3) + O(m \cdot n^2)$. [Rodrigues03] [Pina10]

2.4.4 Método de Cholesky

Conforme referido anteriormente, um dos métodos de fatorização, que é útil em alguns casos, é o método de Cholesky. Seja A uma matriz real *simétrica e positiva definida*, então possui uma fatorização única $A = LL^T$ onde L é triangular inferior com uma diagonal positiva [Kincaid02].

O algoritmo de Cholesky é um caso especial do algoritmo geral da fatorização da LU . Logo, se $A = LU$ e $A = LL^T$ então $U = L^T$. No passo j do algoritmo geral, a entrada da diagonal principal é calculada por

$$l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{1/2}. \quad (2.19)$$

O facto de A ser simétrica e positiva definida garante que $l_{jj} > 0$. A equação (2.19) fornece o seguinte limite para $k \leq j$:

$$a_{jj} = \sum_{s=1}^j l_{js}^2 \geq l_{jk}^2$$

do qual podemos concluir que

$$|l_{jk}| \leq \sqrt{a_{jj}}, (1 \leq k \leq j).$$

Qualquer elemento de L está limitado pela raiz quadrada do correspondente elemento da diagonal de A . Isto implica que os elementos de L não se tornem maiores relativamente aos de A [Kincaid02]. O pseudo-código deste algoritmo é apresentado a seguir [Rodrigues03].

Método de Cholesky

- ◆ Factorização da matriz $A = (a_{ij})_{i,j=1,\dots,n}$
- ◆ Ciclo nas colunas: $j = 1$ a n

$$l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{1/2}$$

Ciclo das linhas: $i = j + 1$ a n

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk} \right) / l_{jj}$$

Fim de ciclo i

Fim de ciclo j

- ◆ Resolução do sistema $Ax = b$
- Ciclo das linhas: $i = 1$ a n

$$y_i = \left(b_i - \sum_{k=1}^{i-1} l_{ik}y_k \right) / l_{ii}$$

Fim de ciclo i

Ciclo das linhas: $i = n$ a 1

$$x_i = \left(y_i - \sum_{k=i+1}^n l_{ki}y_k \right) / l_{ii}$$

Fim de ciclo i

O número das multiplicações necessárias para calcular a factorização de Cholesky é $O\left(\frac{n^3}{6}\right)$. Este número é duas vezes menor do que o método de Gauss. Este facto deve-se à simétrica da matriz A [Rodrigues03].

2.5 MÉTODOS ITERATIVOS

Nesta secção serão estudados métodos iterativos para obter a solução de sistemas do tipo

$$Ax = b \text{ com } \det(A) \neq 0, x \in \mathbb{R}^n.$$

Os métodos iterativos produzem uma sequência de vetores por passos que idealmente convergem para a solução. O processo é terminado no final de um número pré-estabelecido de iterações ou quando a precisão desejada é atingida.

Teoricamente, a solução exata destes sistemas pode ser obtida por métodos iterativos só após um número infinito de passos [Rodrigues03]. Na prática, os métodos iterativos podem produzir boas aproximações com um número reduzido de iterações, isto desde que sejam satisfeitas algumas condições.

Ao contrário do que acontece nos métodos diretos, a matriz do sistema A nunca é alterada após aplicação de um método iterativo. Sendo assim, estes métodos são adequados na resolução de sistemas de grandes dimensões com *matrizes esparsas* (com poucos coeficientes não-nulos). Uma outra vantagem destes métodos é o facto de eles terem a possibilidade de corrigir erros de arredondamento durante o processo de cálculo [Rodrigues03] [Pina10].

Relativamente aos métodos iterativos, há que ter em atenção os seguintes aspetos:

1. Condições da aplicação do método;
2. Escolha da estimativa inicial;
3. Rapidez de convergência.

Vamos então considerar dois exemplos simples de métodos iterativos para resolver os sistema $Ax = b$ com $\det(A) \neq 0$, $x \in \mathbb{R}^n$.

Considerando a decomposição $A = B - C$, onde B é uma matriz qualquer com $\det(B) \neq 0$, de preferência facilmente invertível. Sendo assim a equação do sistema virá:

$$(B - C)x = b$$

$$Bx = Cx + b$$

$$x = B^{-1}Cx + B^{-1}b$$

$$x = Gx + g$$

onde $G = B^{-1}C$ e $g = B^{-1}b$.

Deste modo, podemos definir um método iterativo através da fórmula recorrente

$$x^{(k+1)} = Gx^{(k)} + g. \tag{2.20}$$

Como critério de paragem, podemos utilizar a condição:

$$\|x^{(k+1)} - x^{(k)}\| < \varepsilon \tag{2.21}$$

onde ε é a precisão da aproximação da solução, previamente escolhida. A distância

$$\|x^{(k+1)} - x^{(k)}\|$$

pode ser calculada usando uma norma definida em \mathbb{R}^n , por exemplo, a norma do máximo

$$\|x^{(k+1)} - x^{(k)}\|_{\infty} = \max_{i=1, \dots, n} |x_i^{(k+1)} - x_i^{(k)}|. \quad (2.22)$$

A convergência do método da equação (2.20) é assegurada, independentemente da aproximação inicial, se e só se $|\lambda_i(G)| < 1, i = 1, \dots, n$ com $\sigma(G) = \{\lambda_1(G), \dots, \lambda_n(G)\}$, que é o conjunto dos valores próprios de G , também denominado de *espectro* de G . O raio espectral de G é definido por

$$\rho(G) = \max_{\lambda \in \sigma(G)} |\lambda|. \quad (2.23)$$

A condição necessária e suficiente para a convergência do método definido na equação (2.20) é $\rho(G) < 1$. Este método converge também se $\|G\| < 1$ (condição suficiente) uma vez que $\rho(G) < \|G\|$ para todas as normas de matrizes [Rodrigues03] [Kincaid02] [Golub96].

2.5.1 Método de Jacobi

Genericamente, o **método de Jacobi** consiste em construir matrizes: triangular inferior, triangular superior e diagonal, respetivamente.

$$A_1 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{n1} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

As matrizes apresentadas anteriormente são formadas por elementos da matriz A , o que corresponde à seguinte decomposição $A = D - (-A_1 - A_2)$. Obtemos assim a seguinte equação

$$x = -D^{-1}A_1x - D^{-1}A_2x + D^{-1}b.$$

Podemos escrever o método de Jacobi da seguinte forma:

$$x^{(k+1)} = -D^{-1}A_1x^{(k)} - D^{-1}A_2x^{(k)} + D^{-1}b \quad (2.24)$$

ou

$$x^{(k+1)} = -D^{-1}(A_1 + A_2)x^{(k)} + D^{-1}b. \quad (2.25)$$

De seguida apresenta-se o pseudo-código do método de Jacobi retirado de [Kincaid02].

Método de Jacobi

input $n, (a_{ij}), (b_i), (x_i), M$

for $k = 1$ to M do

for $i = 1$ to n do

$$u_i \leftarrow \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \right) / a_{ii}$$

end do

for $i = 1$ to n do

$x_i \leftarrow u_i$

end do

output $k, (x_i)$

end do

2.5.2 Método de Gauss-Seidel

Podemos observar que, no método de Jacobi, para o cálculo de $x^{(k+1)}$ usamos as informações da iteração anterior. Este método pode ser melhorado se para o cálculo de $x^{(k+1)}$ for utilizada a melhor informação disponível sobre cada uma das suas coordenadas [Rodrigues03]. Ou seja, efetua-se a seguinte alteração à equação (2.24):

$$x^{(k+1)} = -D^{-1}A_1x^{(k+1)} - D^{-1}A_2x^{(k)} + D^{-1}b. \quad (2.26)$$

Ao método descrito na equação (2.26) chama-se **método de Gauss-Seidel**. Esta equação pode ainda ser escrita na seguinte forma:

$$x^{(k+1)} = -(D + A_1)^{-1}A_2x^{(k)} + (D + A_1)^{-1}b \quad (2.27)$$

ou ainda

$$x^{(k+1)} = Gx^{(k)} + g.$$

Neste caso, $G = -(D + A_1)^{-1}A_2$ e $g = (D + A_1)^{-1}b$.

Os métodos de Jacobi e Gauss-Seidel são igualmente conhecidos como métodos das substituições simultâneas e sucessivas, respetivamente.

De seguida apresenta-se em pseudo-código o método de Gauss-Seidel retirado de [Kincaid02].

Método de Gauss-Seidel

```
input  $n, (a_{ij}), (b_i), (x_i), M$ 
for  $k = 1$  to  $M$  do
    for  $i = 1$  to  $n$  do
        
$$x_i \leftarrow \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \right) / a_{ii}$$

    end do
    output  $k, (x_i)$ 
end do
```

É de referir que no algoritmo de Gauss-Seidel, os valores atualizados de x_i substituem imediatamente os anteriores, contrariamente ao que acontece no algoritmo de Jacobi, em que os novos componentes do vetor x são calculados antes de ser efetuada a substituição. No algoritmo de Jacobi, os novos componentes de x (representados por u_i no pseudo-código da secção 2.5.1) podem ser calculados simultaneamente. No algoritmo de Gauss-Seidel, estes novos componentes de x devem ser calculados em série uma vez que, o cálculo do novo x_i , requer todos os novos valores de x_1, x_2, \dots, x_{i-1} .

2.5.3 SOR - Successive Over-Relaxation

O algoritmo de SOR é uma variação do algoritmo de Gauss-Seidel que resulta numa convergência mais rápida [Kincaid02] [Rodrigues03].

Designa-se por forma canónica do método iterativo de um passo, a seguinte representação

$$B^{(k+1)} \frac{x^{(k+1)} - x^{(k)}}{\tau^{(k+1)}} + Ax^{(k)} = b \quad (2.28)$$

onde $B^{(k+1)}$ é uma matriz e $\tau^{(k+1)}$ é o parâmetro iterativo específico para cada método.

Quando $B^{(k+1)} = D$ e $\tau^{(k+1)} = 1$, obtêm-se o método de Jacobi. No caso de $B^{(k+1)} = D + \omega A_1$ e $\tau^{(k+1)} = \omega$ um número real positivo, que irá funcionar como parâmetro de aceleração de convergência, obtém-se

$$D + \omega A_1 \frac{x^{(k+1)} - x^{(k)}}{\tau^{(k+1)}} + Ax^{(k)} = b. \quad (2.29)$$

Quando colocamos $\omega = 1$, obtém-se o método de Gauss-Seidel.

$$(D + A_1)(x^{(k+1)} - x^{(k)}) + Ax^{(k)} = b \quad (2.30)$$

Se $\omega > 0$, $\omega \neq 1$, obtemos uma generalização do método de Gauss-Seidel, que se designa por método de relaxação superior ou SOR (*Successive Over-Relaxation*). Se a matriz A for simétrica positiva definida, o método de SOR converge com $0 \leq \omega \leq 2$ [Rodrigues03] [Kincaid02] [Golub96].

Reescrevendo a equação (2.30) na forma matricial obtém-se:

$$(I + \omega D^{-1}A_1)x^{(k+1)} = \left(\frac{(1 - \omega)}{-\omega D^{-1}A_2} \right) x^{(k)} + \omega D^{-1}b. \quad (2.31)$$

O valor ω que assegura a convergência ótima do método de SOR relaciona-se com $\rho(G)$, da seguinte forma:

$$\omega_{op} = 2 / (1 + \sqrt{1 - \rho(G)}), \quad (2.32)$$

com $G = -(D + A_1)^{-1}A_2$.

Se a matriz A for de diagonal estritamente dominante por linhas, o método de SOR converge se $0 < \omega \leq 1$. Seguidamente apresenta-se o pseudo-código do método de SOR retirado de [Kincaid02].

Método de SOR

```

input  $\omega, n, (a_{ij}), (b_i), (x_i), M$ 
for  $k = 1$  to  $M$  do
    for  $i = 1$  to  $n$  do
         $u_i \leftarrow \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \right) / a_{ii}$ 
    end do
     $x^{(k)} \leftarrow \omega u_i + (1 - \omega)x^{(k-1)}$ 
    output  $k, (x_i)$ 
end do

```

Para uma escolha ótima de ω , o método de SOR pode convergir uma ordem de grandeza mais rápido do que o de Gauss-Seidel [Barret94].

Nos métodos iterativos de Jacobi e Gauss-Seidel, constata-se que o raio espectral das correspondentes matrizes G , está relacionado com a norma da distância entre duas iterações sucessivas, tendo-se:

$$\rho(G) \approx \frac{\|x^* - x^{(k+1)}\|}{\|x^* - x^{(k)}\|} \approx \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)} - x^{(k-1)}\|},$$

para um k suficientemente grande [Rodrigues03].

2.6 MÉTODOS SEMI-ITERATIVOS

Este tipo de métodos é muito usado na resolução de sistemas lineares quando as matrizes são esparsas não estruturadas e de tamanho médio. São métodos com precisão infinita (exata) têm um erro zero no máximo de n passos. Normalmente, nestes métodos, o número de iterações $M \ll n$ [Soares03].

Quando estamos perante um matriz densa, o tempo gasto na sua fatorização é comparável ao tempo necessário para resolver (semi-) iterativamente o problema. Neste caso, a vantagem está na fatorização da matriz já que a possibilidade de calcular múltiplos valores para o vetor b é evidente. Se compararmos uma matriz densa com uma equivalente esparsa em relação à quantidade de memória ocupada, verifica-se que uma fatorização triangular (quando esta existe) de uma matriz esparsa contém mais elementos diferentes de zero do que a própria matriz [Kincaid02].

Um conceito importante para a compreensão dos métodos descritos nesta secção é a noção de forma quadrática de uma matriz. Nas secções anteriores deste documento definiram-se algumas propriedades das matrizes simétricas e positivas definidas. Agora podemos acrescentar que, para matrizes deste tipo, a forma quadrática $f(x) = x^T Ax$ é minimizada pela solução do sistema de equações lineares $Ax = b$.

Apesar de ser possível avaliar a solução de sistemas pela norma de $\|Ax - b\|$, demonstra-se em [Kincaid02] que se uma matriz A for simétrica e positiva definida, o problema de resolver o sistema de equações $Ax = b$ é equivalente ao problema de minimizar a seguinte forma quadrática

$$f(x) = \langle x, Ax \rangle - 2\langle x, b \rangle.$$

2.6.1 Steepest Descent

Utilizar este método para resolver o sistema de equações lineares do tipo $Ax = b$, onde a matriz A de dimensão $n \times n$ é real, simétrica e positiva definida, consiste na produção de uma sequência de vetores a partir de um vetor inicial x_0 que resulta da pesquisa do ponto mínimo para a superfície induzida pela forma quadrática associada à matriz A . A direção que se escolhe em cada passo relaciona-se com aquela segundo a qual $f(x)$ mais decresce, ou seja, a direção do residual r ($r = b - A\tilde{x}$, sendo \tilde{x} a solução obtida), que em [Kincaid02] está identificado como $v^{(k)}$. Uma descrição formal deste método, também denominado *Steepest Descent* é apresentada a seguir.

Método Steepest Descent

input $x^{(0)}, A, b, M$

output $0, x^{(0)}$

for $k = 0$ to $M - 1$ **do**

$$v^{(k)} \leftarrow b - Ax^{(k)}$$

$$t_k \leftarrow \langle v^{(k)}, v^{(k)} \rangle / \langle v^{(k)}, Av^{(k)} \rangle$$

$$x^{(k+1)} \leftarrow x^{(k)} + t_k v^{(k)}$$

output $k + 1, x^{(k+1)}$

end do

A convergência deste método em função do número de condição da matriz (secção 2.3) é analisada em [Shewchuk, 94] onde se verifica que para $k(A) > 40$, a taxa de decréscimo no erro de iteração para iteração é pouco significativa. Logo, quando existe a convergência, esta é lenta.

Uma família de métodos, denominada de direções conjugadas, partilha esta estratégia de minimizar a forma ao longo de sucessivas iterações, e conforme poderemos verificar nas secções seguintes, estas são mais eficientes.

2.6.2 Gradientes Conjugados

O método dos gradientes conjugados descrito em [Hestenes52] é um caso particular dos métodos das direções conjugadas, e difere do *Steepest Descent* apenas no processo de encontrar a direção que melhor aproxima a solução (minimiza a forma quadrática). Neste caso, as direções são construídas de modo a que cada uma seja tomada apenas uma vez. O pseudo código deste método retirado de [Kincaid02] apresenta-se a seguir.

Método dos Gradientes Conjugados

input $x^{(0)}, A, b, M, \varepsilon$

$r^{(0)} \leftarrow b - Ax^{(0)}$

$v^{(0)} \leftarrow r^{(0)}$

output $0, x^{(0)}, r^{(0)}$

for $k = 0$ **to** $M - 1$ **do**

if $v^{(k)} = 0$ **then** *exit loop*

$t_k \leftarrow \langle r^{(k)}, r^{(k)} \rangle / \langle v^{(k)}, Av^{(k)} \rangle$

$x^{(k+1)} \leftarrow x^{(k)} + t_k v^{(k)}$

$r^{(k+1)} \leftarrow r^{(k)} - t_k Av^{(k)}$

if $\|r^{(k+1)}\|_2^2 < \varepsilon$ **then** *exit loop*

$s_k \leftarrow \langle r^{(k+1)}, r^{(k+1)} \rangle / \langle r^{(k)}, r^{(k)} \rangle$

$v^{(k+1)} \leftarrow r^{(k+1)} + s_k v^{(k)}$

output $k + 1, x^{(k+1)}, r^{(k+1)}$

end do

No algoritmo descrito anteriormente, se $r^{(k)} = 0$ temos que $x^{(k)}$ é a solução teórica do sistema de equações lineares. A execução deste algoritmo necessita de armazenar quatro vetores $x^{(k)}, r^{(k)}, v^{(k)}$ e $Av^{(k)}$. Conforme referido em [Soares, 2003] este método não é recomendado para matrizes densas ou de banda.

Pode verificar-se em [Shewchuk94] que a convergência deste método é superior à do *Steepest Descent*. Conclui-se também que quanto pior for o condicionamento da matriz, $k(A) \rightarrow \infty$, pior é a convergência do método.

2.7 PRÉ-CONDICIONAMENTO DE MATRIZES

O pré-condicionamento de matrizes tem por ideia subjacente *melhorar* ou tornar a estrutura da matriz mais *favorável*. Algumas ações de pré-condicionamento tais como, equilíbrio de linha e equilíbrio e coluna [Kincaid02] [Golub96], ocorrem na fatorização quando se usam métodos diretos de resolução de equações lineares.

A taxa de convergência dos métodos iterativos depende das propriedades espectrais da matriz de iteração. Se for possível transformar o sistema de equações lineares de um dado sistema num outro equivalente (com a mesma solução), mas com uma matriz associada de estrutura mais favorável,

estamos perante o pré-condicionamento de uma matriz. À matriz que realiza essa transformação chama-se matriz de pré-condicionamento.

Supondo que M é uma matriz simétrica, positiva definida que aproxima A mas é mais fácil de inverter, podemos resolver $Ax = b$ indiretamente a partir do sistema $M^{-1}Ax = M^{-1}b$.

Se a condição, $k(M^{-1}A) \ll k(A)$, se verificar então, pode-se resolver o sistema equivalente de forma mais eficiente.

Muitos métodos iterativos dependem em grande parte do pré-condicionamento da matriz do sistema. Existem mesmo situações em que só com a aplicação da matriz de pré-condicionamento é possível existir convergência. No entanto, um compromisso deve ser tomado entre o custo de encontrar e aplicar a matriz de pré-condicionamento e o ganho que se obtém na velocidade de convergência do sistema [Kincaid02].

Em alguns casos é possível melhorar ou tornar mais favorável a estrutura da matriz do sistema. Algumas ações de pré-condicionamento tais como equilíbrio de linha e equilíbrio da coluna [Kincaid02] ocorrem na fatorização quando se usam métodos diretos de resolução de equações lineares.

A taxa de convergência dos métodos iterativos depende das propriedades espectrais da matriz de iteração. Se for possível transformar o sistema de equações lineares de um determinado problema noutro equivalente (com a mesma solução) mas com uma matriz de sistema de estrutura mais favorável, estamos perante uma operação de pré-condicionamento de uma matriz. À matriz que efetua essa transformação chamamos matriz de pré-condicionamento.

Supondo que M é uma matriz simétrica, positiva definida que aproxima A mas é mais simples de inverter, pode obter-se a solução de $Ax = b$ indiretamente a partir do sistema

$$M^{-1}Ax = M^{-1}b.$$

Se $k(M^{-1}A) \ll k(A)$ ou o espectro da matriz $M^{-1}A$ estiver mais concentrado, é possível resolver o sistema equivalente de forma mais eficiente.

Muitos métodos iterativos dependem em grande parte do pré-condicionamento da matriz do problema. Podem ocorrer situações para as quais, com a aplicação da matriz de pré-condicionamento, é possível atingir a convergência. No entanto, deve ter-se em conta o tempo de obtenção e aplicação da matriz de condicionamento e o ganho que se obtém na velocidade de convergência do sistema [Soares03].

Relativamente ao método dos gradientes conjugados com pré-condicionamento, é um método para resolver sistemas do tipo $Ax = b$, onde a matriz do sistema é simétrica e positiva definida.

Considerando S uma matriz não singular, o novo sistema pode escrever-se

$$\tilde{A}\tilde{x} = \tilde{b}$$

onde,

$$\begin{cases} \tilde{A} = S^T A S \\ \tilde{x} = S^{-1} x, \\ \tilde{b} = S^T b \end{cases}$$

e a relação entre os número de condição deve obdecer a $k(\tilde{A}) \ll k(A)$. O que se pretende é que o método iterativo utilizado para resolver o sistema pré-condicionado seja mais rápido a convergir que o original.

Vamos admitir a matriz Q simétrica e positiva definida que pode ser fatorizada a partir de S da seguinte forma:

$$Q^{-1} = S S^T.$$

Partindo do pseudo-código apresentado na secção 2.6.2 apresenta-se a seguir o método do gradientes conjugados com pré-condicionamento retirado de [Kincaid02].

Método dos Gradientes Conjugados (com pré-condicionamento)

input $x^{(0)}, A, b, M, Q$

$r^{(0)} \leftarrow b - Ax^{(0)}$

solve $Q\tilde{r}^{(0)} = r^{(0)}$ for $\tilde{r}^{(0)}$

$v^{(0)} \leftarrow r^{(0)}$

output $0, x^{(0)}$

for $k = 0$ **to** $M - 1$ **do**

if $v^{(k)} = 0$ **then** *exit loop*

$\hat{t}_k \leftarrow \langle \tilde{r}^{(k)}, r^{(k)} \rangle / \langle v^{(k)}, Av^{(k)} \rangle$

$x^{(k+1)} \leftarrow x^{(k)} + \hat{t}_k v^{(k)}$

$r^{(k+1)} \leftarrow r^{(k)} - \hat{t}_k Av^{(k)}$

 solve $Q\tilde{r}^{(k+1)} = r^{(k+1)}$ for $\tilde{r}^{(k+1)}$

if $\langle \tilde{r}^{(k+1)}, r^{(k+1)} \rangle < \varepsilon$ **then**

if $\langle r^{(k+1)}, r^{(k+1)} \rangle < \varepsilon$ **then** *exit loop*

end if

$\hat{s}_k \leftarrow \langle \tilde{r}^{(k+1)}, r^{(k+1)} \rangle / \langle \tilde{r}^{(k)}, r^{(k)} \rangle$

$v^{(k+1)} \leftarrow \tilde{r}^{(k+1)} + \hat{s}_k v^{(k)}$

output $k + 1, x^{(k+1)}, r^{(k+1)}$

end do

Este método reduz-se ao dos gradientes conjugados quando $Q^{-1} = I$, isto porque nesta situação $\tilde{r}^{(k)} = r^{(k)}$, $\hat{t}^{(k)} = t^{(k)}$ e $\hat{s}^{(k)} = s^{(k)}$.

Se $Q = A$ e $S = A^{-\frac{1}{2}}$, o sistema pré-condicionador reduz-se a $\tilde{x}^{(k)} = \tilde{b}$ e é de resolução simples [Soares03]. No entanto, esta condição ideal, $k(\tilde{A}) = 1$, não é interessante em termos computacionais, pois determinar o valor de $\tilde{b} = S^T b$ é tão difícil como resolver o sistema original.

Como em cada iteração é necessário resolver o sistema $Qx = y$, a matriz Q deve ser escolhida de modo a que o sistema seja fácil de resolver. Se Q for uma matriz diagonal esta condição está garantida, mas se se pretende velocidade na convergência teremos de escolher matrizes de pré-condicionamento mais complexas [Soares03].

Neste capítulo foi efetuada a apresentação de diversos métodos de resolução de sistemas de equações lineares bem como alguns condicionalismos inerentes à utilização dos mesmos.

Capítulo 3 – Interpolação Linear de Sinais

Neste capítulo são apresentados os algoritmos de reconstrução que serão utilizados para efetuar a reconstrução dos sinais extraídos de uma sequência de *frames* ou de uma imagem. Um dos algoritmos de reconstrução de sinal, utilizado para efetuar a reconstrução dos sinais é a versão discreta do algoritmo Papoulis [Papoulis75], o outro é um algoritmo de dimensão mínima [Ferreira94b]. Este último algoritmo está disponível para ser utilizado no modo de cálculo direto ou iterativo. O objetivo é dar ao utilizador do protótipo desenvolvido uma forma de comparação de desempenho entre os algoritmos de dimensão máxima e mínima.

Nesta secção, é feita uma introdução à notação e terminologia utilizadas e seguidamente são descritos os algoritmos e indicadas quais as restrições de utilização dos referidos algoritmos. No final do capítulo, são analisadas as condições de convergência dos algoritmos.

3.1 PAPOULIS-GERCHBERG – UM ALGORITMO DE DIMENSÃO MÁXIMA

O algoritmo Papoulis-Gerchberg pode ser utilizado para recuperar amostras perdidas em sinais de banda limitada e de tamanho finito [Ferreira94a]. Neste caso, não é feita nenhuma suposição relativamente à distribuição das amostras perdidas permitindo geometrias de padrões de erros distintas (entrelaçada, aleatória e contígua). É possível considerar o sinal observado\corrompido como sendo uma versão amostrada do sinal original e interpretar o resultado da reconstrução estudada como o resultado de uma amostragem.

Demonstra-se que o algoritmo de Papoulis-Gerchberg converge, se a densidade do conjunto de amostras excede um certo valor mínimo: este valor aumenta com a largura de banda dos dados [Ferreira94a]. É também salientado que o padrão de erros pode afetar a convergência do algoritmo de reconstrução. Por exemplo, para sinais de baixa frequência (*low-pass signals*), as melhores taxas de convergência são obtidas quando a distância entre as amostras perdidas é um múltiplo de um inteiro. As piores taxas de convergência ocorrem geralmente quando as amostras perdidas são contíguas [Ferreira94a].

3.1.1 Notação e Terminologia

Considerando o espaço complexo de dimensão n , \mathbb{C}^n , com o produto interno e norma usuais, definimos *senal* ou *sequência* de tamanho n , como sendo qualquer vetor complexo x de dimensão n , com componentes, ou amostras, x_0, x_1, \dots, x_{n-1} . A matriz de *Fourier* F é a matriz unitária de dimensão $n \times n$ com componentes F_{mk} dados por

$$F_{mk} = \frac{1}{\sqrt{n}} e^{j\frac{2\pi}{n}mk} \quad (3.1)$$

onde j representa a unidade imaginária. A transformada discreta de Fourier (DFT- *Discrete Fourier Transform*) de x , representada por \hat{x} , é por definição a sequência $\hat{x} = Fx$.

As duas operações, definidas em \mathbb{C}^n , nas quais estamos interessados, serão denominadas por *amostragem* e por *limitação de banda*.

Por definição, as *operações de amostragem* fazem o mapeamento da sequência numa outra, colocando a zero um subconjunto de amostras. Na forma matricial, isto corresponde à multiplicação de uma matriz diagonal D , contendo apenas zeros e uns. A diagonal de D será denominada de conjunto de amostras associada com a operação de amostragem D , e a própria matriz D será denominada de *matriz de amostragem*. A *densidade* de um conjunto de amostragem é por definição, $\frac{s}{n}$, sendo s o número de entradas não-nulas no conjunto de amostragem. Assume-se que $0 < s < n$, $D \neq I$, $D \neq 0$ [Ferreira94a].

Pode definir-se operador limitador de banda pela matriz B da forma $B = F^{-1}\Gamma F$, onde Γ é uma matriz de amostragem na frequência. Assim, de acordo com esta definição, a operação limitação em banda em \mathbb{C}^n é semelhante à operação de amostragem efetuada no domínio do tempo. Definiu-se largura de banda da sequência $y = Bx$ como sendo q/n , onde q é o número de entradas não-nulas na matriz Γ . Para sequências reais de baixa frequência, uma largura de banda igual a b significa que a maior frequência normalizada na sequência é $b/2$. As entradas não-nulas da diagonal de Γ [Ferreira94a] constituem a *banda de passagem* da matriz B .

3.1.2 Condições de Aplicabilidade

O problema da reconstrução de sinais parcialmente conhecidos e limitados em banda (senal observado), que implicará o conhecimento de um dado subconjunto de amostras da DFT dos dados, os harmônicos nulos, é equivalente a modelo discreto-discreto [Sanz83]. Os sinais são parcialmente conhecidos uma vez que apenas um subconjunto das suas amostras se encontra disponível. A tarefa

consiste em, a partir das amostras disponíveis, obter as amostras desconhecidas. Este problema apresenta semelhanças com o problema de amostragem, que basicamente consiste na reconstrução de um sinal dado um subconjunto das suas amostras [Ferreira96; Ferreira99; Vieira97; Reis00].

Quando o padrão de erro é contíguo estamos perante um *problema de extrapolação*. Para uma determinada largura de banda podem ser recuperadas um número máximo de amostras desconhecidas. Este número aumenta com o decréscimo da largura de banda dos sinais. Isto pode ser interpretado como resultado da amostragem, garantindo a possibilidade de reconstrução sem erros, de um sinal de banda-limitada, se um subconjunto suficientemente denso das suas amostras é conhecido.

O algoritmo de reconstrução utilizado é um método iterativo que se reduz à versão discreta de dimensão finita do algoritmo de extrapolação de Papoulis-Gerchberg, se as amostras desconhecidas são contíguas, e se a constante de relaxação $\mu = 1$ [Ferreira94a].

É muitas vezes conveniente conhecer os limites superiores e inferiores para a taxa de convergência de um algoritmo iterativo.

Iremos igualmente referir qual a natureza dos conjuntos de amostragem para os quais a taxa de convergência do algoritmo é a melhor ou a pior possível num determinado sentido. Verifica-se que, para o mesmo número de amostras desconhecidas, o pior padrão de erro para sinais passa-baixo será um conjunto contíguo de amostras perdidas [Ferreira94a]. Por outro lado, os conjuntos de amostragem ótimos são aqueles que surgem quando a distância entre as amostras perdidas é múltiplo de um inteiro. Para sinais que não são passa-baixo, isto não é necessariamente verdade. Os sinais passa-banda fornecem um exemplo de uma classe importante de sinais para os quais os piores conjuntos de amostragem podem não ser contíguos.

Demonstra-se em [Ferreira94a] que, para um conjunto de amostragem óptimo, o raio espectral da matriz de iteração, que é um importante indicador da taxa de convergência, é independente do número de amostras desconhecidas. De facto, um sinal com diversas amostras desconhecidas pode ser tão facilmente reconstruído como um sinal com apenas uma amostra desconhecida, isto no sentido de convergência assintótica, se as posições das amostras desconhecidas forem bem escolhidas. Utilizando esta propriedade no problema da reconstrução garante-se a estabilidade do problema e o bom condicionamento da matriz de iteração.

3.1.3 Sinais de Banda Limitada ou de Energia Finita

Uma das condições para que possa ser aplicado o algoritmo de reconstrução aos sinais, é que estes sejam de banda-limitada. Para que esta condição seja garantida, será necessário neste caso filtrar

todos os sinais, utilizando um Filtro Passa-Baixo (FPB), eliminando, desta forma, as altas-frequências (ruído), e obtendo sinais de banda-limitada. Os sinais de banda-limitada serão a partir deste momento denominados **sinais observados**.

A limitação de banda dos sinais é obtida através de $y = Bx$ onde $B = F^{-1}\Gamma F$, onde y representa o sinal observado, e x o sinal original. Na Figura 3.1 estão apresentados graficamente os resultados das operações especificadas pelo operador B . Em primeiro lugar, é efectuado o cálculo da DFT de x , tendo sido utilizado, no nosso caso, o algoritmo da *Fast Fourier Transform* (FFT) [Cooley65]. Depois de obtidas as componentes de x no domínio da frequência (Figura 3.1(b)), multiplica-se o sinal x pela matriz Γ , obtendo-se, deste modo, um sinal limitado na frequência (Figura 3.2(a)). Esta filtragem, no domínio da frequência, é equivalente a colocar a zero um determinado número de amostras conhecidas. As amostras que são colocadas a zero são as componentes centrais do espectro de frequência, uma vez que estas representam as componentes de frequência mais alta. Para obter o sinal observado no domínio do tempo calcula-se a *Inverse Fast Fourier Transform* (IFFT), do sinal limitado em frequência (Figura 3.2(b)).

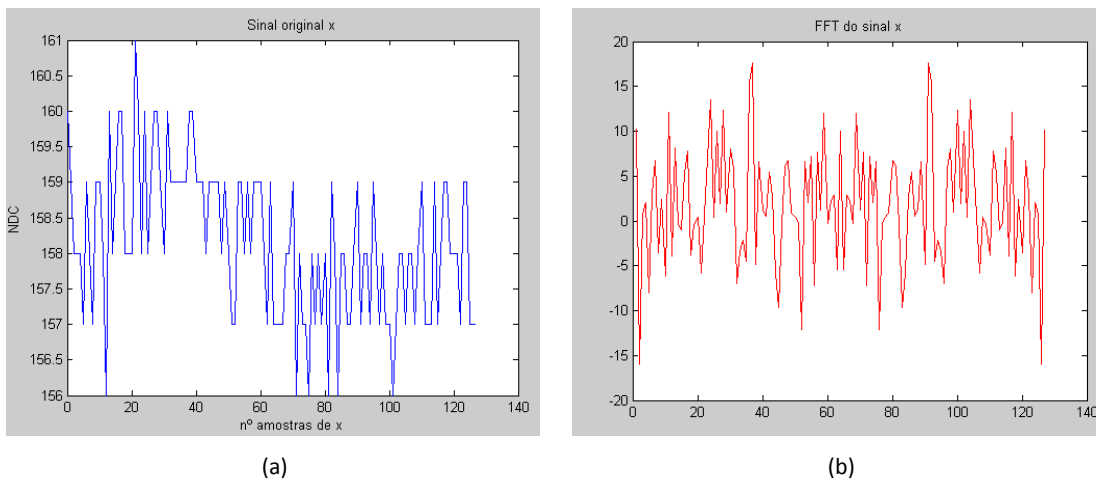
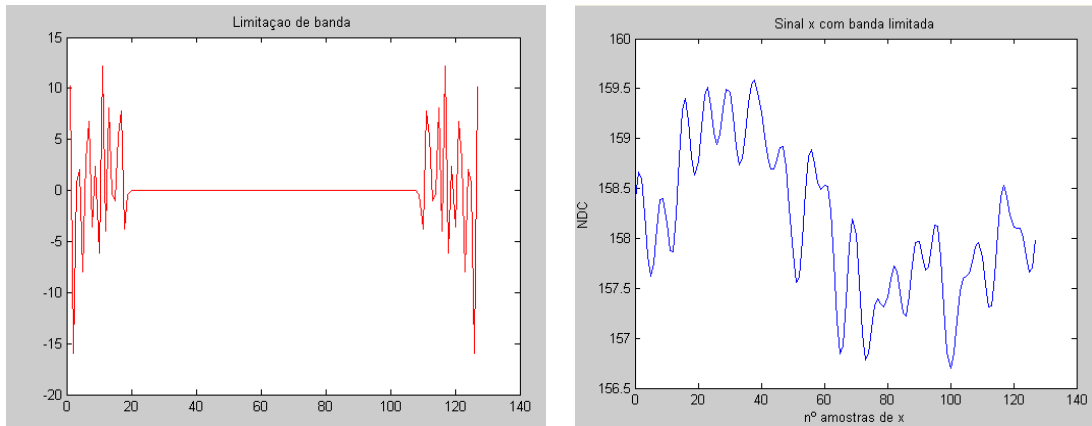


Figura 3.1 – Sinal não limitado na frequência (a) e respetiva FFT (b)



(a)

(b)

Figura 3.2 – Limitação de banda de um sinal utilizando o operador B

3.1.4 Papoulis-Gerchberg – Interpretação Gráfica

O algoritmo apresentado nesta secção é um algoritmo linear iterativo de 1ª ordem, com matrizes de iteração não-Hermiticas [Ferreira94a]. Neste algoritmo em particular, utilizam-se as operações de amostragem e limitação de banda definidas na secção 3.1.1, e mencionadas como distorção e restrição em [Ferreira94a]. O sinal original satisfaz a $x = Bx$ e o sinal distorcido é $y = Dx$, para as matrizes B e D adequadas. Isto sugere a seguinte equação

$$x = Bx + \mu(y - Dx) \quad (3.2)$$

$$x = \mu y + (I - \mu D)Bx \quad (3.3)$$

onde μ é uma constante fixa qualquer, que no nosso caso tomará o valor de 1, e I é a matriz identidade. A sequência das aproximações sucessivas é representada por

$$u_{n+1} = \mu y + (I - \mu D)Bu_n = T_1 u_n \quad (3.4)$$

onde

$$T_1(\cdot) = \mu y + (I - \mu D)B(\cdot) \quad (3.5)$$

O algoritmo de Papoulis-Gerchberg é baseado na filtragem seguida de amostragem. A dimensão do problema é máxima, pois utiliza o número total de amostras do problema, ao contrário de outros métodos em que a dimensão do problema é do tamanho das amostras perdidas: neste caso, diz-se que a dimensão é mínima [Ferreira96]. O esforço computacional envolvido está diretamente relacionado com o número total de amostras do sinal e não com o número de amostras perdidas. Em [Ferreira94b; Soares03] pode comprovar-se este facto.

Se tivermos um sinal limitado em frequência, por exemplo, o sinal da Figura 3.3, podemos, sob certas condições, perder um determinado conjunto de amostras, sendo ainda possível a sua recuperação.

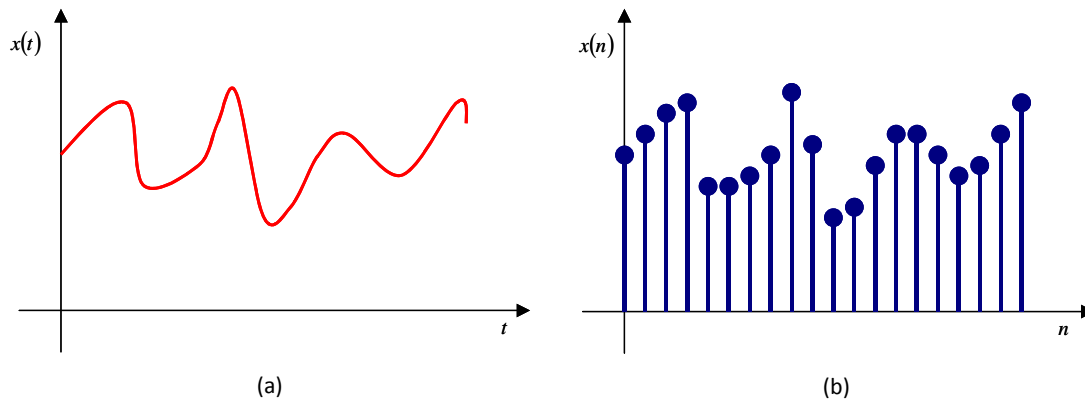


Figura 3.3 – (a) Sinal $x(t)$ contínuo limitado em frequência; (b) Sinal $x(t)$ discretizado

Admitindo um padrão de erro modelado pela função $r[n]$, obtém-se $x'[n]$ através da multiplicação de $x[n]$ por $r[n]$. As componentes nulas do sinal de $x'[n]$ (Figura 3.4(b)) representam as posições onde foram perdidas as amostras.

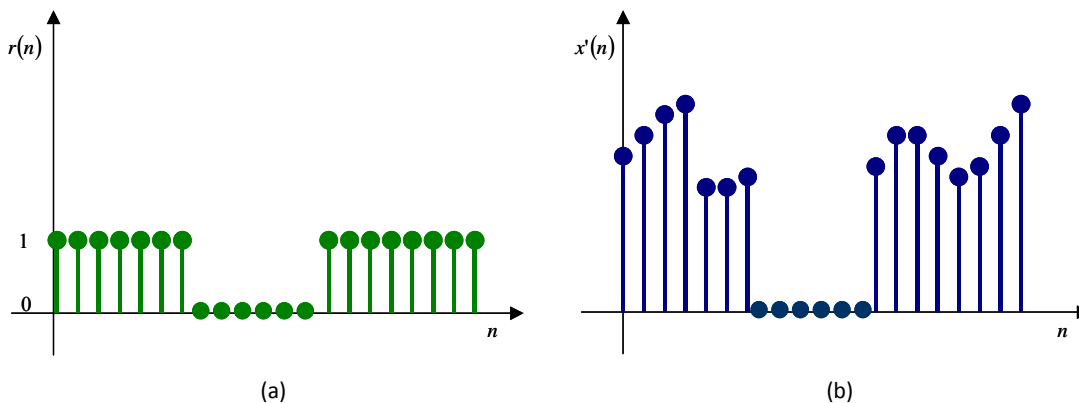


Figura 3.4 – (a) Trem de impulsos para amostrar o sinal $x[n]$; (b) Sinal $x[n]$ corrompido

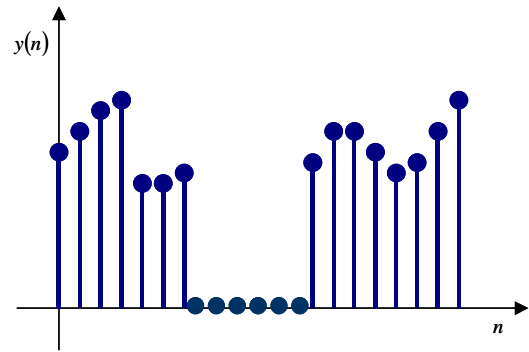
O processo iterativo é descrito nas Figuras 3.5 e 3.6. A equação utilizada para implementar este algoritmo é a equação (3.3).

A primeira etapa consiste em calcular $x = Bx$, como já foi descrito na secção 3.1.3. A segunda etapa do processo iterativo consiste em multiplicar $(I - D)$ pelo resultado de Bx , onde D é a matriz de amostragem (neste caso o sinal da Figura 3.4(a)) e I é a matriz de identidade. Com esta operação consegue-se colocar a zero as amostras conhecidas do sinal, deixando as amostras desconhecidas com um valor diferente de zero (resultado da operação Bx). O resultado desta operação pode ser visualizado na Figura 3.6(a). Ao sinal resultante da operação anteriormente descrita é adicionado o sinal com as amostras desconhecidas $x = y + (I - D)Bx$. O resultado pode ser visualizado na Figura 4.6(b).

Este processo repete-se até que o valor das amostras desconhecidas se aproxime do seu valor observado. Ao fim de k iterações estaremos afastados da solução do problema de um dado erro residual \mathcal{E} , equação (2.21). Em cada iteração do algoritmo será medida a diferença entre os sinais

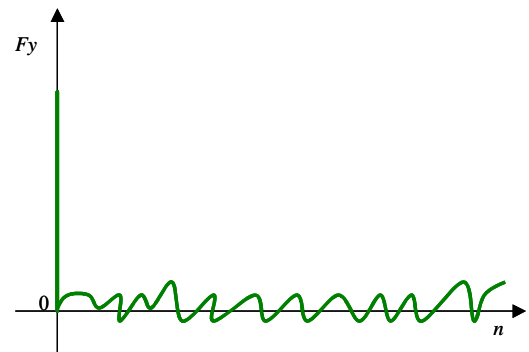
obtidos na iteração k e na iteração $k - 1$. Quando esta diferença for inferior a um determinado valor, que será previamente estabelecido, o processo terminará. Essa diferença pode ser expressa matematicamente, por uma distância entre o sinal da iteração k e o da iteração $k - 1$. Quando esta diferença for inferior a um determinado erro \mathcal{E} o processo iterativo termina, uma vez que temos a garantia que o erro residual é inferior ou igual a \mathcal{E} .

Sinal observado com amostras desconhecidas
 $y = Dx$



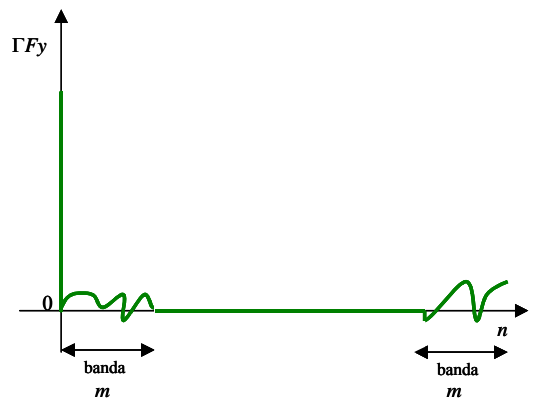
(a)

Transformada de Fourier de $y(n)$
 Fy



(b)

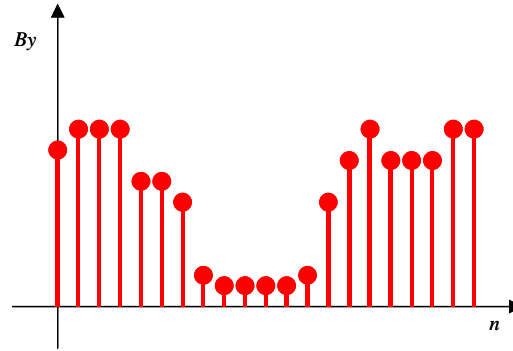
Limitação em frequência do sinal
 ΓFy



(c)

Transformada Inversa de Fourier

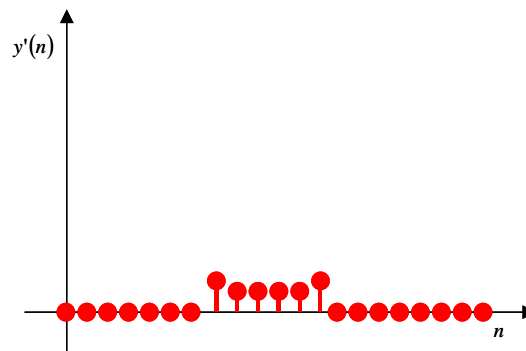
$$F^{-1}\Gamma Fy = By$$



(d)

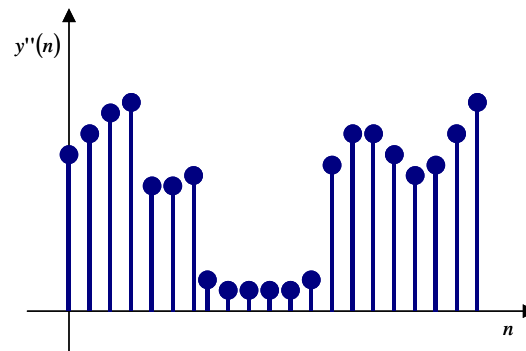
Figura 3.5 – Limitação de banda de um sinal

$$y' = (I - D)By$$



(a)

$$y'' = y + (I - D)By$$



(b)

Figura 3.6 – Substituição das amostras conhecidas no sinal de banda limitada

3.1.5 Convergência do Algoritmo de Papoulis-Gerchberg

Demonstra-se em [Ferreira94a] que o algoritmo converge nas seguintes condições: $0 < \mu < 2$ e $d \geq \omega$, onde $d = (n - q)$ e $\omega = (n - s)$. Lembra-se que n é o número de amostras do sinal, s é o número de entradas não-nulas da matriz D , e q é o número de entradas não-nulas da matriz Γ . A segunda condição:

$$(n - q) \geq (n - s) \tag{3.6}$$

equivale a,

$$q \leq s \quad (3.7)$$

Como já foi referido anteriormente q é o número de entradas não-nulas da matriz Γ , ou seja, o número de amostras da banda de passagem do FPB, logo $q = 2m + 1$, onde m é a banda de amostras do FPB. Assim a equação (3.7) pode ser rescrita da seguinte forma:

$$s \geq 2m + 1 \quad (3.8)$$

Para o caso de $\mu = 1$, é fornecido em [Ferreira94a] o limite inferior e superior para o erro de reconstrução em qualquer iteração. Os limites são os melhores possíveis e são fornecidos os vetores iniciais para os quais estes são alcançados.

3.2 DIMENSÃO MÍNIMA

Um algoritmo de dimensão mínima é caracterizado por um sistema de apenas l equações correspondentes às l amostras desconhecidas. Nesta secção será descrito um algoritmo de dimensão mínima que requer, à semelhança do algoritmo descrito na secção 3.1, sinais de banda limitada e de dimensão finita. Ou seja, a equação $x = Bx$ tem que ser válida. Mais detalhe sobre esta operação pode ser consultada na secção 3.1.3 deste documento.

Para estabelecer os conceitos básicos deste algoritmo, utilizou-se o caso específico de um sinal original $x[n]$ de comprimento $n = 5$, isto é $x = \{x_1, x_2, x_3, x_4, x_5\}$. Para este sinal em particular a equação $x = Bx$ fica

$$\begin{aligned} x_1 &= B_{11}x_1 + B_{12}x_2 + B_{13}x_3 + B_{14}x_4 + B_{15}x_5 \\ &\quad \vdots \\ x_5 &= B_{51}x_1 + B_{52}x_2 + B_{53}x_3 + B_{54}x_4 + B_{55}x_5 \end{aligned} \quad (3.9)$$

Para efeitos de reconstrução, vamos supor que a segunda e quarta amostra de $x[n]$ são perdidas. Então o conjunto de equações (3.9) será limitado àquelas que contêm as amostras perdidas. Em cada uma destas equações, estamos interessados em separar o termo direito da equação, que contém as amostras perdidas (x_2, x_4) daqueles que contêm as amostras conhecidas. Isto resulta em

$$\begin{aligned} x_2 &= B_{21}x_1 + B_{22}x_2 + B_{23}x_3 + B_{24}x_4 + B_{25}x_5 \\ x_4 &= B_{41}x_1 + B_{42}x_2 + B_{43}x_3 + B_{44}x_4 + B_{45}x_5 \end{aligned} \quad (3.10)$$

que é equivalente a

$$\begin{bmatrix} x_2 & x_4 \end{bmatrix} = \begin{bmatrix} B_{22} & B_{24} \\ B_{42} & B_{44} \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} + \begin{bmatrix} B_{21} & B_{23} & B_{25} \\ B_{41} & B_{43} & B_{45} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix}. \quad (3.11)$$

Vamos definir u como sendo o subconjunto do sinal original $x[n]$ que contém as amostras desconhecidas. Neste caso $u = \{x_2, x_4\}$, este conjunto tem cardinalidade $k = 2$. Definiu-se ainda $U = \{i_1, i_2, \dots, i_k\}$ como sendo o conjunto de índices das k amostras desconhecidas em $x[n]$. Neste caso $U = \{2, 4\}$. Sendo assim, a equação (3.11) pode ser escrita da seguinte forma

$$x_i = \sum_{j \in U} B_{ij}x_j + \sum_{j \notin U} B_{ij}x_j, \quad i \in U \quad (3.12)$$

ou na forma matricial

$$u = Su + h \quad (3.13)$$

onde S é a principal submatriz $k \times k$ de B , conforme definido na equação (3.11), e h é o vetor de dimensão $(n - k)$ no segundo somatório da equação (3.12), que corresponde a uma combinação linear das amostras conhecidas de $x[n]$. As condições sob as quais estas equações fornecem a solução para u podem ser consultadas em [Ferreira94b]. No caso em que se pretende obter a solução diretamente, sem recorrer ao método iterativo, a equação (3.13) é equivalente a

$$\begin{aligned} u - Su &= h \\ (I - S)u &= h \\ u &= (I - S)^{-1}h. \end{aligned} \quad (3.14)$$

Este resultado é válido desde que $(I - S)^{-1}$ exista (ver secção 2.4). Assim, teoricamente, a equação (3.13) possui uma única solução, independentemente do número e do tipo de distribuição das amostras perdidas. Se a equação (3.13) for resolvida através de um processo iterativo, então, no caso de se utilizar um método de não-relaxação, sugere-se a seguinte formulação

$$u^{(i+1)} = Su^{(i)} + h. \quad (3.15)$$

Quando $u(k)$ é obtido na iteração k e a solução é dada pelo limite

$$u = \lim_{i \rightarrow \infty} u^{(i)}, \quad (3.16)$$

isto, independentemente do valor de $u(0)$. A condição $\rho(S) < 1$ garante que o limite anterior existe, onde S é a matriz do sistema [Ferreira94b] e $\rho(S)$ é o seu raio espectral, calculado através da equação (2.15).

Conforme apresentado no capítulo anterior a solução da equação (3.13) pode ser obtida a partir de diferentes abordagens ao problema de interpolação. Podem ser utilizadas duas formulações para resolver a equação (3.13): o cálculo direto e o método iterativo. O cálculo direto de u , através da equação (3.14), tem a vantagem de ser efetuada através de um só passo, isto desde que $(I - S)^{-1}$

exista. Na prática, podem ser vários os fatores que podem tornar difícil o cálculo da inversa de $(I - S)$. Por exemplo, se um dos valores próprios de S for suficientemente próximo da unidade, então o cálculo de $(I - S)^{-1}$ pode tornar-se difícil ou mesmo impossível, o que se traduz num problema mal condicionado. Nestes casos, um método iterativo poderá ser usado para evitar esta adversidade e assim encontrar uma aproximação precisa para a solução u . Na eventualidade de estarmos perante um problema mal condicionado, utilizando o cálculo direto, este problema é impossível de resolver, no entanto, recorrendo aos métodos iterativos, é sempre possível obter uma aproximação, embora a sua precisão possa ser diminuta.

Os valores próprios da matriz do sistema S dependem da distribuição das amostras perdidas. Concretamente, é mais provável que o seu raio espectral seja **um** em distribuições do tipo rajada do que em distribuições equidistantes de amostras perdidas [Ferreira94c]. No caso da reconstrução de sinal é importante referir que, se a distribuição de amostras perdidas $U = \{i_0, i_1, \dots, i_{n-1}\}$ for equidistante segundo um inteiro fixo $m \geq 1$, ou seja, $U = \{i_0 m, i_1 m, \dots, i_{n-1} m\}$, então os valores próprios de S , λ_i , são dados por $\lfloor rm \rfloor / m$ isto é

$$\frac{\lfloor rm \rfloor}{m} \leq \lambda_i(S) \leq \frac{\lceil rm \rceil}{m} \leq 1$$

onde $\lfloor rm \rfloor$ representa o maior inteiro menor ou igual a rm e $\lceil rm \rceil$ representa o menor inteiro maior ou igual a rm . No caso em particular de $r = \lfloor rm \rfloor / m$, os valores próprios de S são todos iguais $\lambda_i(S) = r, \forall i$.

Dada a análise anterior, é possível tornar o problema bem condicionado ao criar um intervalo apropriado entre as amostras perdidas. Deste modo, é possível colocar $\lambda_i(S)$ próximo de r ou dos seus múltiplos, independentemente do número de amostras perdidas. Ao escolhermos um valor apropriado para os fatores de sobre-amostragem (r) e de entrelaçamento (m) (i.e. tais que $m \times r$ seja um inteiro), é possível colocar $\lambda_i(S)$ menor que a unidade de modo a controlar a precisão da reconstrução e a velocidade de processamento [Neves08].

Para que se possa utilizar o algoritmo de dimensão mínima, descrito na secção anterior, é necessário que os sinais sejam de banda limitada. Esta é uma condição necessária em ambos os algoritmos descritos neste capítulo.

Para além disso, utilizar o algoritmo de dimensão mínima nas suas formulações: direta, equação (3.13), e iterativa, equação (3.14), é necessário que determinadas condições de convergência estejam garantidas. Sendo assim, e de acordo com o descrito na secção 2.5, para os métodos iterativos a condição necessária e suficiente para a convergência é que o raio espectral da matriz de iteração seja

menor do que a unidade. No caso específico da equação (3.14) é S , logo a convergência dos métodos iterativos é garantida sempre que $\rho(S) < 1$.

Atendendo agora ao caso da formulação direta, equação (3.13), o condicionamento a matriz $(I - S)^{-1}$ deverá ser analisado através do cálculo do número de condição da mesma, secção 2.3. O número de condição de uma matriz, neste caso $(I - S)^{-1}$, é calculado através da equação (2.13). Podemos então aferir que um número de condição elevado implica a não utilização dos métodos diretos para a obtenção da solução numérica do problema em estudo. No nosso caso, um valor alto é maior ou igual a 1×10^{15} , referente à precisão *double* nos PCs.

Será pois através da análise destes parâmetros, $\rho(S)$ e $k((I - S)^{-1})$, e da condição de convergência da equação (3.8), que será efetuada a análise do condicionamento e convergência dos algoritmos descritos nas secções 3.1 e 3.2. O valor destes parâmetros irá permitir ao utilizador saber previamente se o algoritmo de reconstrução irá ou não convergir para a solução desejada.

Neste capítulo foram descritos dois algoritmos de interpolação linear de sinais, o algoritmo de Papoulis-Gerchberg e um algoritmo de Dimensão Mínima, e foram ainda apresentados alguns condicionalismos à sua utilização e quais as condições de convergências dos mesmos.

Capítulo 4 – Um Modelo para o Ensino de Algoritmos de Reconstrução de Sinal

Durante as últimas décadas, com o disseminar das tecnologias da informação, surgiram algumas ferramentas ou simuladores cujo principal objetivo é garantir e consolidar os conceitos teóricos ministrados durante as aulas [Marín05; Caulfield11; Velte13]. O principal objetivo destas ferramentas é dar aos alunos a oportunidade de experimentar conceitos lecionados, bem como testar esses conceitos em diferentes cenários. Com este tipo de ferramentas, os alunos podem praticar e aprender sozinhos, melhorando o seu conhecimentos e competências na área do saber.

O processamento digital de sinal (PDS) pode ser aplicado em diversas áreas da engenharia. No entanto, são bastantes os alunos que vêem esta área como uma espécie de matemática abstrata. O que acontece é que só ocasionalmente é feita a ponte entre a matemática e a aplicação prática de conceitos, especialmente no secundário e no ensino supeior. Para alguns alunos, esta conexão é crítica, uma vez que os motiva para a aprendizagem dos conceitos teóricos, mas também porque os vai ajudar a reter ou consolidar os conhecimentos adquiridos [Pierre99].

Existem diversos trabalhos que abordam o ensino da engenharia [Stice87], [Belhot01] e [Carvalho01], sendo neles mencionados diferentes tipos de aprendizagem e o ato de ensinar e recorrendo a diferentes metodologias.

Nesta tese propõe-se um novo modelo para o ensino de conceitos de processamento digital de sinal, utilizando, como caso de estudo, o problema da reconstrução de sinal. O modelo consiste numa prévia apresentação oral de conceitos teóricos, nomeadamente, dos algoritmos de reconstrução apresentados préviamente no capítulo 3, e uma posterior comprovação dos conceitos lecionados através de uma experimentação interativa. Para cumprir este desígnio foi desenhado um protótipo aplicativo que permite aos alunos testar diferentes algoritmos de reconstrução, obtidos a partir de geometrias de padrão de erros distintas, que será apresentado delhadamente no capítulo 5 desta tese. Este protótipo reveste-se de extrema importância para dar cumprimentos aos objetivos do modelo de ensino proposto dado que serve de meio para que os alunos possam observar o funcionamento dos algoritmos passo a passo e, ao mesmo tempo, fazerem a ligação dos conceitos teóricos com a aplicação

na prática, recorrendo não apenas à exposição teórica de conceitos, mas também tirando partido do uso de novas tecnologias de simulação e experimentação.

O novo modelo tem como principal objetivo melhorar a aprendizagem dos alunos na área do processamento e reconstrução de sinal.

Para a sua implementação optou-se por utilizar o ciclo de aprendizagem de Kolb [Kolb84], à semelhança de outros trabalhos na área da engenharia [Abdulwahed09] [Konak13], ou seja, criou-se uma experiência para desmistificar o problema da reconstrução de sinal, que permite aos alunos finalizar o ciclo de aprendizagem através da observação refletiva de uma experiência concreta e consequente conceptualização dos conceitos abordados, estimulando assim uma experimentação ativa. Através do modelo proposto promove-se uma experimentação ativa, tendo por base um problema real de engenharia, o que dá aos alunos a oportunidade de testar diferentes cenários, de modo a prever soluções estáveis para o problema. Nas próximas secções, descreve-se o ciclo de aprendizagem e o processo de conceptualização do modelo proposto.

4.1 CICLO DE APRENDIZAGEM DE KOLB

O ciclo de aprendizagem de Kolb possui as seguintes fases: 1- Experiência Concreta (sentir, experimentar); 2 – Observação Refletiva (observar); 3 – Conceptualização Abstrata (pensar), e finalmente 4- Experimentação Ativa (fazer) [Kolb84].

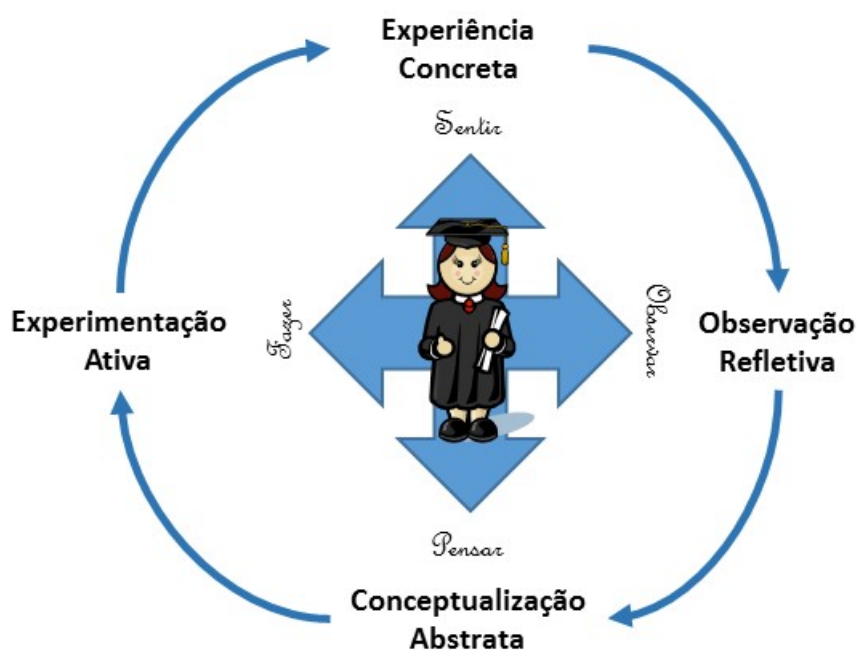


Figura 4.1 – Ciclo de aprendizagem de Kolb

Segundo Kolb, os alunos classificam-se como: *convergentes*, *assimiladores*, *divergentes* ou *acomodados* [Kolb84]. Os alunos ditos *convergentes* gostam da aplicação prática de conceitos e aprendem melhor com a experimentação ativa e conceptualização abstrata. Os alunos *assimiladores* estão normalmente mais interessados na lógica base de uma ideia do que na sua aplicação prática. Estes aprendem através da conceptualização abstrata e da observação refletiva. Já os alunos *divergentes* tendem a ser criativos e a gerar, criar ou enumerar alternativas. Estes aprendem mais facilmente através da experiência concreta e observação refletiva. Os *acomodados* são orientados a objetivos, decisivos, agem predominantemente pela emoção do que pela lógica. O seu estilo de aprendizagem preferido é a experiência concreta e a experimentação ativa [Pierre99] [Tokdemir07]. Conclui-se que, recorrendo a diferentes metodologias de ensino, os docentes poderão ajudar um maior número de alunos com diferentes estilos de aprendizagem, e ao mesmo tempo melhorar as competências mais fracas dos alunos. Verifica-se em [Stice87], que uma variedade de atividades de aprendizagem melhora, a longo prazo, a retenção de conceitos.

Os alunos de engenharia podem ser classificados em qualquer uma das quatro categorias referidas anteriormente no entanto, a maior percentagem dos engenheiros é *convergente* e em segundo lugar aparecem os *assimiladores*. Segundo [Pierre99], as metodologias de ensino usadas na engenharia tendem a assentar melhor no perfil dos *assimiladores*, isto porque dão ênfase ao conhecimento teórico. No que se refere aos *convergentes*, estes não irão aprender ou trabalhar de forma eficaz se não virem uma aplicação prática para os conceitos lecionados. Os laboratórios e ferramentas de simulação são um excelente método de fornecer aos alunos a etapa da experimentação prática, mas se os exercícios não forem bem escritos, a parte aplicacional pode ainda estar escondida do aluno. Recorrendo ao protótipo aplicacional desenvolvido, os alunos terão oportunidade de verificar e testar diversos conceitos teóricos ligados ao processamento de sinal, mais concretamente, o caso específico da reconstrução de sinal.

Existem bons livros que são atualmente usados nas disciplinas de processamento digital de sinal, no entanto, a existência de um laboratório ou ferramentas adicionais complementam os currículos das disciplinas e melhoram o ciclo de aprendizagem dos alunos [Ambardar97; Proakis99; Proakis12].

O que se apresenta nesta tese é um modelo de ensino do processamento digital de sinal, recorrendo ao caso de estudo da reconstrução de sinal. É igualmente apresentado o protótipo aplicacional de interpolação de sinal desenvolvido. Este modelo poderá ser utilizado em disciplinas de processamento digital de sinal, e o mesmo promove a aprendizagem utilizando uma experimentação ativa e uma experiência concreta, para a reconstrução de sinal. Ao recorrer a sinais provenientes de imagens ou sequências de vídeo, dá-se aos alunos a oportunidade de aplicar técnicas base de PDS visualizando o

seu resultado graficamente. A escolha do *software* MATLAB¹ no desenvolvimento da aplicação, prende-se com o facto de o mesmo possuir um ambiente excelente para o processamento de sinal.

4.2 CICLO DE APRENDIZAGEM IMPLEMENTADO

Os potenciais utilizadores deste modelo são, na sua maioria, alunos das áreas da engenharia, e estes são maioritariamente classificados como tendo um perfil convergente. Os alunos convergentes logo dão uma grande importância à aplicação prática de conceitos e aprendem melhor com a experimentação ativa. Sendo assim, optou-se pelo desenvolvimento de um protótipo que fornecesse aos seus utilizadores a oportunidade de experimentar e testar conceitos teóricos anteriormente ministrados. A ferramenta foi desenvolvida tendo em mente o problema de reconstrução de sinal e as operações inerentes a este problema, permitindo assim adquirir ou consolidar conhecimentos numa área específica do processamento digital de sinal.

Partiu-se então da ideia que os sinais a utilizar no modelo e consequentemente na ferramenta seriam a uma ou duas dimensões: imagens ou áudio. Razão pela qual se utilizaram, na primeira versão, sinais áudio e, na segunda versão, imagens ou *frames* provenientes de uma sequência de vídeo. A ideia é apresentar aos alunos diversos tipos de sinais, provenientes de problemas reais, fornecendo assim os dados a uma experiência concreta.

O ciclo de aprendizagem proposto no modelo foi dividido em duas partes, uma onde o aluno poderá efetuar o condicionamento do problema em estudo e uma outra onde poderá testar diversos algoritmos de reconstrução de sinal e aferir sobre a sua eficácia e desempenho. Resumindo, ambas as partes permitem completar o ciclo de aprendizagem relativamente ao condicionamento de sinais e à reconstrução dos mesmos utilizando diferentes algoritmos ou métodos. Através das experiências disponíveis, o aluno poderá testar diferentes condicionamentos e mais tarde concluir sobre qual o método mais favorável à reconstrução desses sinais, fazendo de forma mais natural a ligação entre os conceitos ministrados anteriormente a sua aplicabilidade num problema real de engenharia. Numa apreciação global, o modelo proposto e o protótipo aplicativo desenvolvido permitem a realização de experiências controladas, e dão ao aluno a possibilidade de criar diferentes cenários de teste, alterando as regras ou parâmetros para obter cenários mais ou menos favoráveis. Através da observação dos resultados obtidos, o que irá permitir a conceptualização de conhecimentos, o modelo seguido induz o aluno a testar novas possibilidades e obter conclusões sobre a experiência realizada, fechando assim o ciclo de aprendizagem.

¹ Matlab 7.9 é um produto da *MathWorks Inc.* – URL: <http://www.mathworks.com/>

Mais detalhes sobre as diferentes operações realizadas no protótipo aplicativo desenvolvido serão descritos no capítulo 5 desta tese.

4.2.1 Primeira Etapa: Extração de Sinais e Condicionamento do Problema

Na primeira fase pretende-se que o aluno/utilizador efetue a extração e o condicionamento dos sinais. A experiência consiste em extrair os sinais de uma situação ou problema real, e efetuar o processo de corrupção dos mesmos para posteriormente testar os algoritmos existentes no protótipo. Com esta primeira etapa pretende-se que o aluno/utilizador possa, através da criação de diferentes cenários de corrupção, conceptualizar sobre como melhorar ou piorar (caso seja o pretendido) o condicionamento do seu problema. Para tal, conceitos como limitação em frequência de sinais, geometrias de padrão de erro e decimação são abordados.

O fluxo de informação, nesta primeira etapa, de condicionamento segue o fluxo indicado na Figura 4.2. As operações têm que ser efetuadas pela ordem correta, para tal no protótipo aplicativo desenvolvido os botões são ativados sempre que a operação precedente já tenha sido realizada. Esta funcionalidade foi implementada para que o aluno/utilizador execute as operações pela ordem necessária.

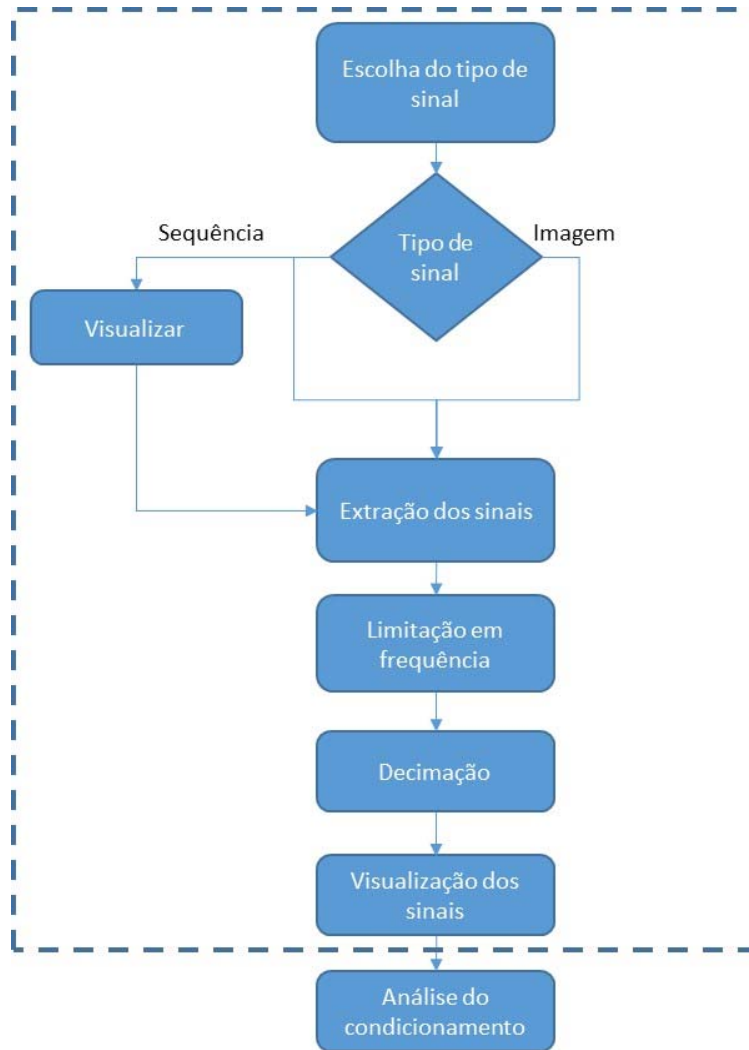


Figura 4.2 – Fluxo de informação etapa de extração e condicionamento de sinal

Depois de abordados os conceitos de forma expositiva, neste modelo o objetivo é dar ao aluno a possibilidade de testar os conceitos com exemplos práticos concretos. O primeiro passo consiste na extração dos sinais. Os sinais extraídos poderão ser imagens ou sequências de *frames* com origem em diversas áreas científicas, como por exemplo: imagens médicas, imagens com mapas de profundidade para criação de cenários 3D [Norkin06; Karim09; Liu12], entre outras. A pré-condição é que o número total de *frames* da sequência seja 1024, ou no caso de imagens que estas tenham 1024 colunas, não existindo nenhuma pré-condição para o número de linhas. Isto permite ao aluno associar sempre os sinais ao um determinado problema real, ou seja, saber qual o tipo de informação os sinais representam.

No caso genérico, numa sequência com n *frames*, cuja dimensão dos *frames* é $M \times N$ *pixels*, sendo M e N o número de colunas e linhas respetivamente, cada sinal é composto por n amostras e o número de sinais que serão retirados da sequência para análise é $M \times N$. Na Figura 4.3 está descrito o processo de extração dos sinais de uma sequência.

O processo consiste no seguinte: depois de lida a imagem para a matriz A , são guardadas todas as linhas de A , numa linha da matriz X . As dimensões de X são as seguintes, $M \times N$ colunas, sendo M e N a dimensão de A (imagem), e n linhas, onde n é como indicado anteriormente, o número de quadros da sequência. A Figura 4.4 exemplifica com maior detalhe este processo. Cada um dos sinais extraídos representa o valor em NDC (Níveis de Cinzento) de cada um dos *pixels* dos *frames* ao longo dos n *frames* da sequência.

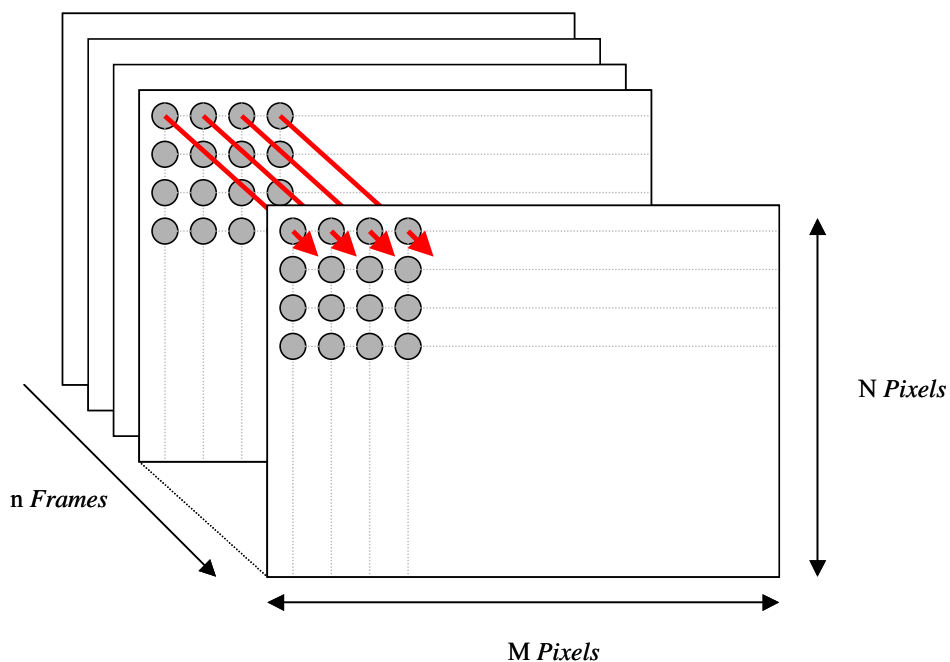


Figura 4.3 – Processo de extração dos sinais de uma sequência

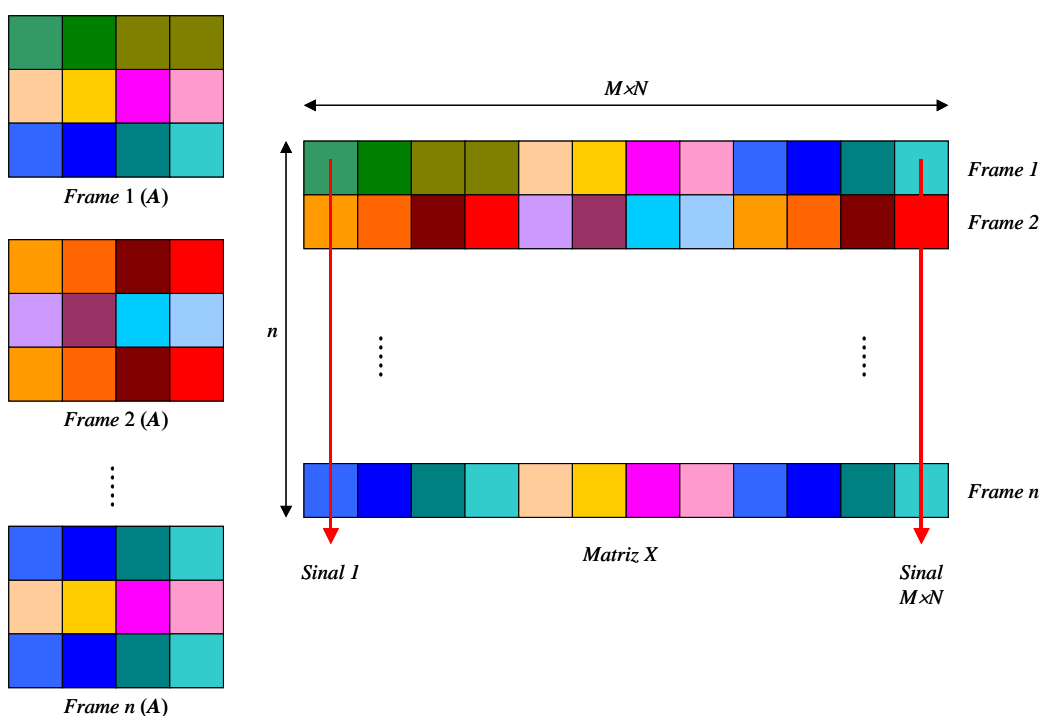


Figura 4.4 – Processo para guardar todos os *frames* da sequência numa única matriz

No caso de o utilizador escolher uma imagem, a dimensão genérica de uma imagem é $M \times N$ onde M representa o número de colunas e N é o número de linhas dessa imagem. A aplicação permite trabalhar com imagens a cor mas, para efeitos de cálculo, essas imagens são convertidas em imagens em NDC.

O processo consiste no seguinte: depois de lida a imagem para a matriz A , são guardadas todas as linhas de A , nas colunas da matriz X . As dimensões de X são $N \times M$, correspondendo a N colunas e M linhas. A Figura 4.5 exemplifica com maior detalhe este processo.

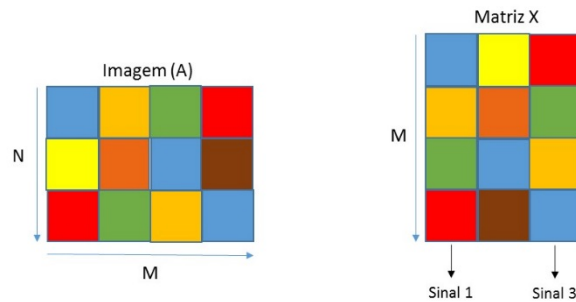


Figura 4.5 – Processo para guardar todas as linhas da imagem nas colunas de uma matriz

O processo de extração será aplicado aos sinais todos, da sequência ou da imagem, o mesmo se passa com as restantes operações disponíveis na aplicação, com a exceção da reconstrução que será apenas aplicada ao *pixel*/linha seleccionados. A generalidade das operações serão aplicadas a todos os sinais extraídos, mas a visualização será apenas do *pixel*/linha seleccionados.

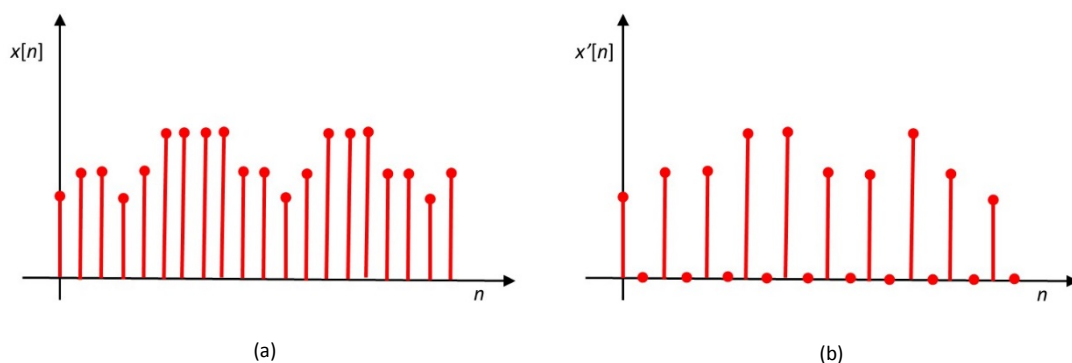
Posteriormente deve ser efetuada a operação de limitação em frequência dos sinais. Pretende-se que, com a utilização do protótipo aplicacional desenvolvido, os alunos/utilizadores possam compreender o conceito de limitação em frequência de sinais, em que consiste e quais as consequências desta operação. Por outras palavras, pretende-se que após utilizar o protótipo aplicacional o aluno possa responder às seguintes questões: O que é um sinal limitado em frequência? Como é que a limitação em frequência vai afetar os sinais? Qual a relação entre o fator de amostragem a quantidade de amostras que se poderá perder posteriormente no processo de decimação/corrupção? Qual o melhor algoritmo de reconstrução e em que situações poderão ou não utilizá-lo? Questões como estas poderão ser respondidas através de uma experimentação ativa promovida neste modelo. Através do uso do protótipo aplicacional fornece-se, em primeiro lugar, ao aluno uma experiência concreta que lhe permitirá fazer uma observação refetiva e uma conceptualização abstrata dos resultados visualizados. O querer saber o que acontece quando se utilizam os diferentes algoritmos de reconstrução, e o querer descobrir qual o melhor, leva-os a testar o protótipo disponível neste modelo em diferentes cenários, promovendo assim a experimentação ativa. Mais tarde, isso leva a que os mesmos testem os algoritmos em situações de condicionamento diferentes (mais ou menos favoráveis).

Os sinais originais representativos da sequência ou imagem em análise, são filtrados utilizando um FPB (Filtro Passa Baixo), para garantir que todos os sinais sejam de banda limitada, e para que possam ser utilizados os dois algoritmos de reconstrução disponíveis na aplicação: o algoritmo discreto de Papoulis-Gerchberg e um algoritmo de Dimensão Mínima, anteriormente descritos no capítulo 3. Os sinais limitados em frequência, conforme referido no capítulo 3, secção 3.1.3, são denominados sinais observados.

Recorrendo à experimentação ativa disponível no modelo, e à oportunidade de escolha de diversos valores para o fator de sobre amostragem (r), os alunos poderão ver na prática de que forma este parâmetro condiciona o número de amostras perdidas. Através do teste e utilização de diferentes valores de r na operação de limitação em banda, mais uma vez recorrendo a uma experimentação ativa, o aluno conseguirá aferir qual o melhor valor do fator de sobre amostragem quando o objetivo é perder uma maior (ou menor) quantidade de amostras nos sinais no processo de decimação. Este será o próximo passo a efetuar.

Neste modelo são abordadas três geometrias de padrão de erros distintas: entrelaçada, aleatória e contígua (*rajada*). Pretende-se que os alunos entendam que a geometria de padrão de erros será um fator determinante no condicionamento do problema. Para tal, os alunos deverão utilizar e testar nos sinais extraídos as diferentes geometrias de padrão de erro no processo de decimação/corrupção. No fim o aluno será capaz de identificar qual o caso mais desfavorável à reconstrução, tendo em conta a geometria de padrão de erros escolhida e o número de amostras perdidas. Um outro objetivo é que o aluno possa visualizar em que consiste cada uma destas geometrias de padrão de erros. De seguida explica-se, com maior detalhe, cada um dos tipos de corrupção disponíveis.

Relativamente à corrupção entrelaçada, esta permite que o aluno escolha de três valores distintos de entrelaçamento: 2, 4 ou 8. Na prática, e para o caso de 1024 amostras, isto resulta em sinais decimados com 512, 256 e 128, respetivamente. Na Figura 4.6 é demonstrada a metodologia utilizada na corrupção dos sinais utilizando este método de decimação.



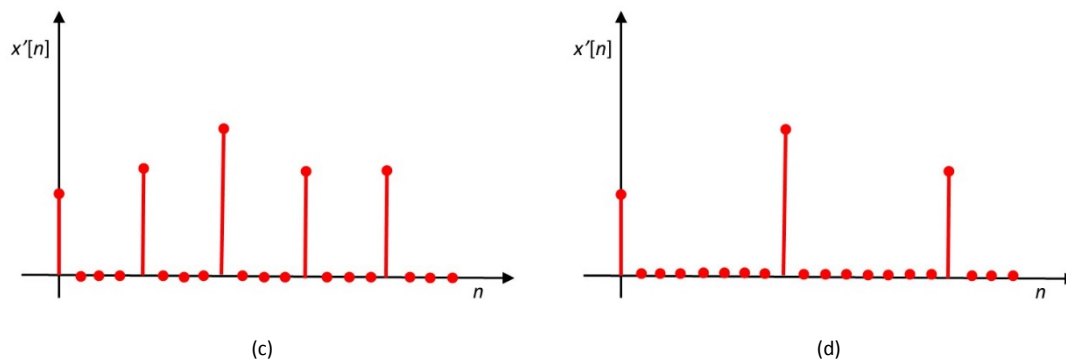


Figura 4.6 – Decimação entrelaçada. Sinal observado (a), Entrelaçamento=2 (b), Entrelaçamento=4 (c), Entrelaçamento=8 (d).

No caso do entrelaçamento de valor 2, e considerando uma amostra em $n=0$, são eliminadas as amostras (1,3,5,...), Figura 4.6(b). Se o valor de entrelaçamento for 4, são eliminadas as amostras (1,2,3; 5,6,7;...), Figura 4.6(c). Finalmente se o valor do entrelaçamento for 8, são eliminadas as amostras (1,2,3,4,5,6,7; 9,10,11,12,13,14,15; 17,18...), Figura 4.6(d). Ou seja, o valor do espaçamento entre amostras é aumentado à medida que vamos aumentando o valor do entrelaçamento.

No caso da decimação aleatória, é gerado um vetor de corrupção aleatório. O aluno terá apenas que especificar o número de amostras que pretende perder. Imaginemos que queremos perder 5 amostras do sinal da Figura 4.6(a). O número de amostras total deste sinal é 20 (da amostra 0 à 19), logo aleatoriamente são escolhidas 5 dessas 20 amostras possíveis, sem possibilidade de repetir qualquer uma das amostras. É então gerado um vetor de 5 elementos com a posição das amostras que serão perdidas. Por exemplo, o vetor {3,9,12,15,19} significa que iremos eliminar as amostras destas respetivas posições no sinal $x[n]$, obtendo-se assim o sinal da Figura 4.7. Este processo será aplicado a todos os sinais extraídos e a posição das amostras perdidas é idêntica para todos os sinais.

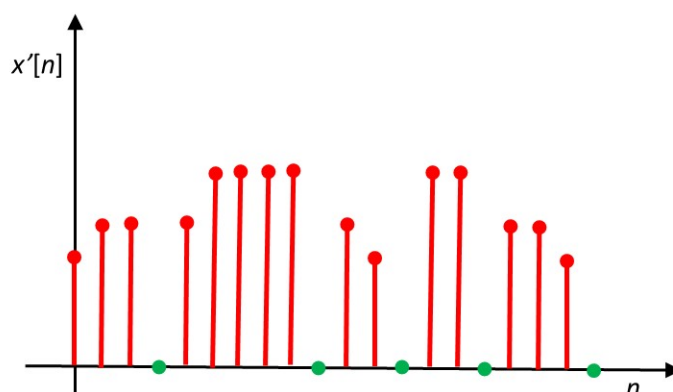


Figura 4.7 – Decimação aleatória, total de amostras perdidas é 5

No caso da decimação em rajada ou contínua, optou-se por perder as amostras de forma contínua, a partir do meio dos sinais. Ou seja, determina-se o meio dos sinais, e perder-se-ão as amostras a partir

daí para ambos os lados. O objetivo é abrir um buraco no centro dos sinais. A nível de implementação, apenas se determina se o número de amostras a perder é par ou ímpar, e será isso que vai determinar o número de amostras que se perdem para ambos os lados. Aplicando esta metodologia ao sinal da Figura 4.6(a) ou seja um sinal $x[n]$ com 20 amostras, a localização do centro será na amostra $n=10$, e considerando que queremos perder um número de amostras par, neste caso em concreto 4, o resultado desta corrupção está apresentado na Figura 4.8. Na eventualidade de o número de amostras a perder ser ímpar (neste caso perder 5 amostras), o resultado será o apresentado na Figura 4.9.

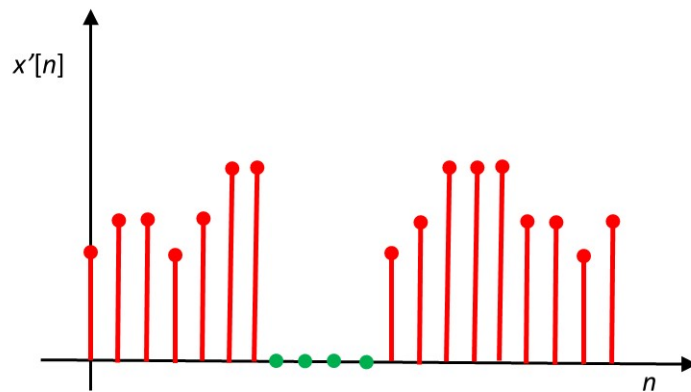


Figura 4.8 – Decimação contínua, total de amostras perdidas é 4

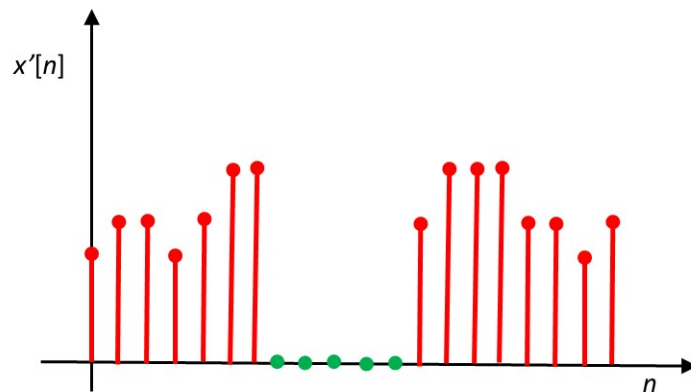


Figura 4.9 – Decimação contínua, total de amostras perdidas é 5

Após executada a operação de decimação está terminado o condicionamento do problema. É através da análise das diferentes versões dos sinais em estudo que os alunos/utilizadores poderão concluir sobre quais as implicações da utilização de um fator de sobre amostragem baixo ou alto, e que atenuação é produzida nos sinais na operação de limitação em frequência. Neste modelo ainda lhes é permitido visualizar quais as amostras perdidas no processo de corrupção utilizando as diferentes geometrias de padrão de erros.

Finalizada esta etapa, a análise do condicionamento previamente efetuado poderá ser analisado na etapa seguinte do ciclo de aprendizagem.

4.2.2 Segunda Etapa: Avaliação da Convergência e Reconstrução dos Sinais

Como já foi referido anteriormente, nos capítulos 2 e 3, os parâmetros analisados para avaliar o condicionamento do problema são o raio espectral da matriz S (Eq. 3.13) e o número de condição da matriz $(I-S)$ (Eq. 3.14). O principal objetivo do modelo é que, através da análise dos valores destes parâmetros, fique facilitada a escolha do algoritmo ou método mais favorável à obtenção de uma solução para o problema em estudo. Isto permite ao aluno/utilizador saber antecipadamente qual a classe de métodos que deverá escolher para reconstruir o sinal ou então optar por alterar o condicionamento anteriormente efetuado. No entanto, mesmo que não estejam reunidas as condições para a obtenção de uma solução numérica com determinada classe de métodos, o aluno/utilizador poderá testar sempre todos os métodos presentes no protótipo aplicacional. O objetivo é que os mesmos tenham uma perceção prática que o valor destes parâmetros, de facto, condiciona a escolha dos algoritmos de reconstrução e que os valores dos mesmos se alteraram com diferentes cenários de corrupção.

Na Figura 4.10 está apresentado o fluxo de dados da etapa da análise de convergência e reconstrução dos sinais, implementado neste modelo de ensino.

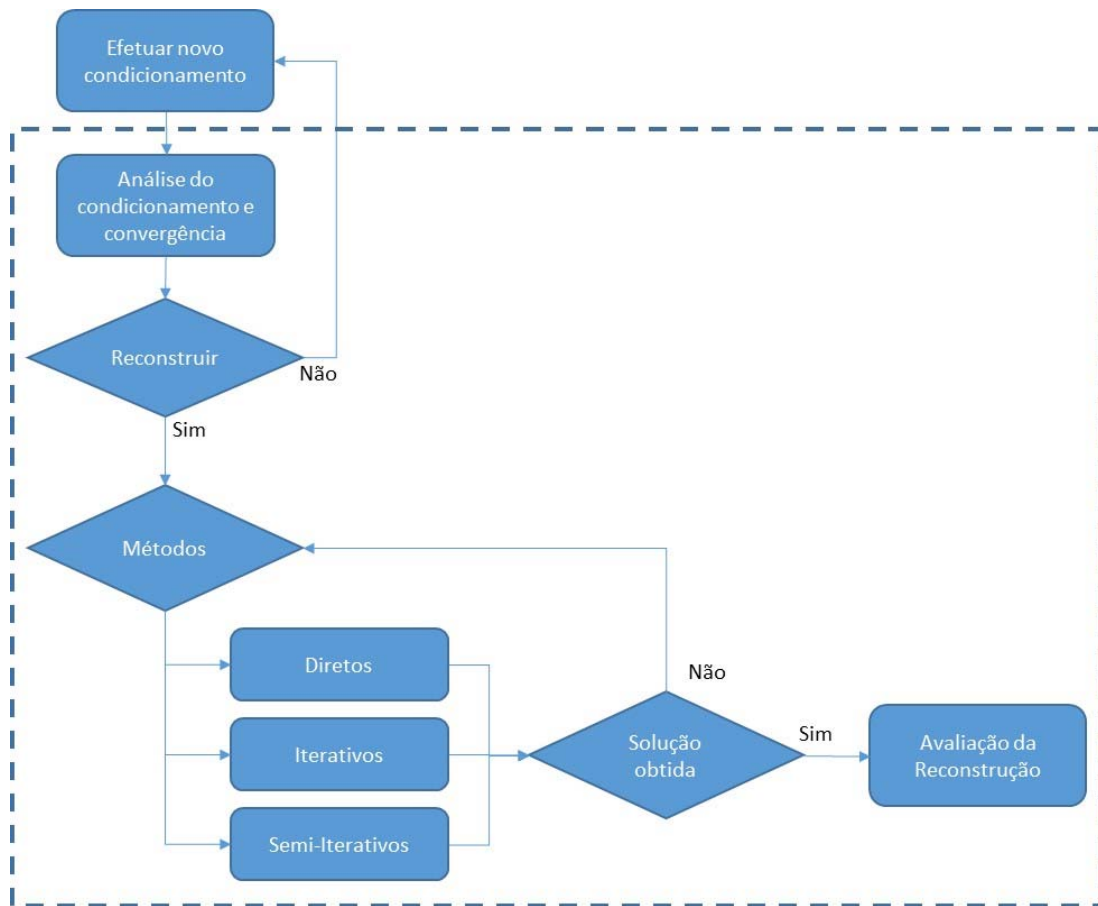


Figura 4.10 – Fluxo de informação etapa de análise de convergência e reconstrução de sinal

A ligação entre as duas fases do modelo de ensino permite ao aluno avaliar todo o processo realizado anteriormente (primeira etapa), e poderá então optar por testar a reconstrução nas condições atuais ou então tentar obter uma situação mais favorável alterando o fator de sobre amostragem e/ou o tipo de decimação ou número de amostras perdidas. É feita aqui a ligação entre o condicionamento e a possibilidade de obter uma solução para o problema. Sem entrar na reconstrução dos sinais propriamente dita, e através do protótipo desenvolvido, é possível recriar uma experiência de corrupção de sinais e, através de uma observação refletiva e conceptualização abstrata, aferir sobre as condições mais ou menos favoráveis ao processo de reconstrução e à obtenção numérica do problema em estudo. A possibilidade de uma experimentação ativa dá ao utilizador a oportunidade de alterar os dados do problema inicial de modo a melhorar (se assim o desejar) o condicionamento previamente efetuado.

Um dos outros objetivos deste modelo de ensino é dar ao aluno/utilizador a possibilidade de testar algoritmos e métodos de resolução distintos. Sendo assim, optou-se por abordar e implementar dois algoritmos distintos: o algoritmo de Papoulis-Gerchberg e um algoritmo de Dimensão Mínima, descritos no capítulo 3. A escolha destes algoritmos prende-se com o facto de o primeiro ser um algoritmo de dimensão máxima e o segundo de dimensão mínima. Uma outra característica é que o

Papoulis-Gerchberg é um algoritmo iterativo e o algoritmo de Dimensão Mínima permite a sua resolução utilizando a formulação iterativa e a direta. Estas escolhas permitem aos alunos/utilizadores a comparação direta entre algoritmos de dimensão máxima e mínima, e identificar, qual a vantagem entre optar por uma resolução direta ou iterativa.

Para além dos dois algoritmos mencionados anteriormente, foram igualmente adicionados novos métodos de resolução do problema, todos eles anteriormente descritos no capítulo 2, secções: 2.4, 2.5 e 2.6. Estes métodos serão utilizados para obter a solução do nosso problema quando o mesmo estiver na formulação direta do algoritmo de dimensão mínima, equação (3.14). Na Figura 4.11 podemos observar quais os métodos abordados neste modelo e que poderão ser utilizados na reconstrução dos sinais préviamente corrompidos. Na zona vermelha estão apresentados os métodos iterativos, na zona castanha os métodos diretos e a amarelo o método semi-iterativo disponível no protótipo aplicacional.

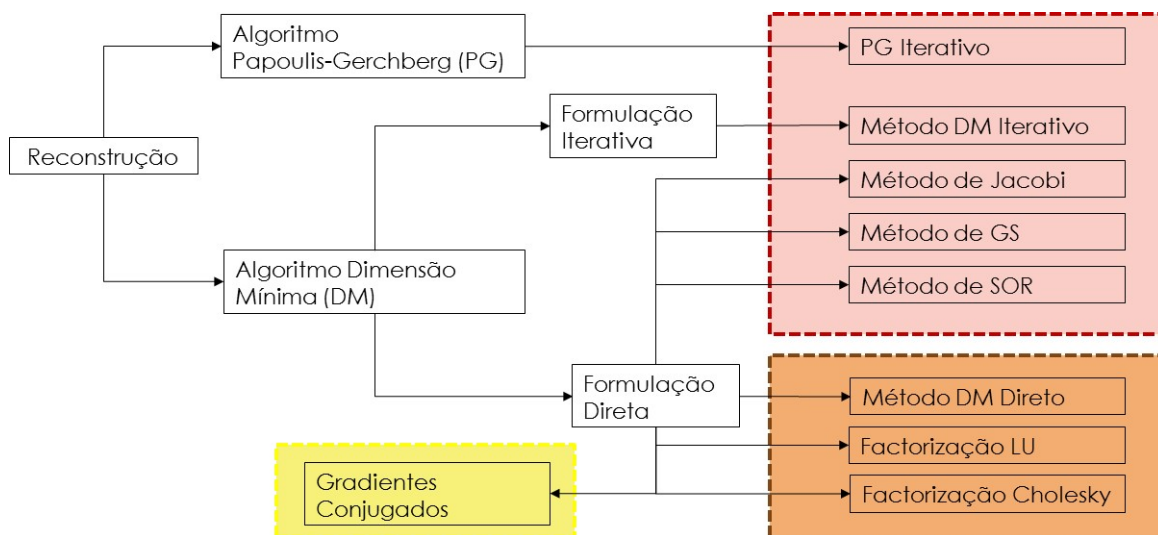


Figura 4.11 – Métodos Iterativos, Diretos e Semi-iterativos Disponíveis no Protótipo Aplicacional

Observando o fluxo de informação apresentado na Figura 4.10, podemos observar que, depois de escolhido o algoritmo ou método de reconstrução, podemos ou não obter uma solução ótima para o problema em estudo. Isto porque se pretende que o aluno/utilizador possa efetivamente testar os diversos algoritmos e métodos de resolução, mesmo em situações de não convergência. Optou-se por esta funcionalidade porque é mais fácil os alunos assimilarem os conceitos através da experimentação ativa. Dizer que o sistema não converge, porque as condições de convergência não estão verificadas, ou porque o raio espectral é um, ou porque o número de condição é elevado, é geralmente difícil de entender sem observar, de facto, o que acontece quando estas situações ocorrem na prática. Esta é uma grande vantagem deste modelo, permitindo fazer a ligação entre conceitos ministrados e observar na prática o que acontece quando determinadas condições são ou não verificadas. Permite

assim associar um valor a uma situação específica. Com a utilização deste modelo os alunos começarão a fazer a ligação entre conceitos sem grande esforço, uma vez que as operações são sequenciais e frequentemente implica que eles mesmos criem cenários distintos só para ver o que acontece depois. Esse tipo de curiosidade é despertada com o uso de simuladores, como é o caso do protótipo desenvolvido, ou seja, introduz a oportunidade de ver na prática o porquê de terem surgido diferentes tipos de algoritmos de reconstrução, com abordagens diferentes, bem como o porquê de tantos métodos de resolução de equações lineares e quais as vantagens de usar uns ou outros e em que situações. São conceitos que vão sendo assimilados de forma natural recorrendo a este modelo de ensino. Uma outra vantagem é o poder observar os algoritmos a sua aplicação prática e o seu funcionamento, estes deixam de ser apenas equações apresentadas numa sala de aula, que, na maior parte dos casos, os alunos não conseguem entender nem visualizar a sua aplicação em casos práticos.

Uma vez que o objetivo primordial da reconstrução de sinal é obter uma solução ótima para um problema, é igualmente abordado no modelo a avaliação do desempenho dos diversos algoritmos e métodos resolução. A avaliação é feita através de uma medida objetiva da qualidade dos sinais reconstruídos e do tempo de reconstrução. Para os métodos iterativos é igualmente possível comparar o número de iterações efetuadas.

Sendo assim, para medir a qualidade dos sinais após reconstrução, será calculado o *Peak Signal-to-Noise Ratio* (PSNR) e o *Root Mean-Squared Error* (RMSE), equações (4.1) e (4.2) respetivamente. Refira-se que na equação (4.2) N é o número total de amostras dos sinais, f é o sinal original e \tilde{f} é o sinal reconstruído. Para comparar os tempos de reconstrução, utiliza-se a *Time Unit* (TU) em detrimento de segundos. Uma TU é o tempo que determinada máquina demora a calcular uma FFT (*Fast Fourier Transform*). Esta característica mantém a avaliação de desempenho dos algoritmos (em termos de cálculo computacional), independente da velocidade da máquina utilizada.

$$PSNR = 20 \log_{10} \frac{255}{RMSE} \text{ (dB)} \quad (4.1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N [f(i) - \tilde{f}(i)]^2} \quad (4.2)$$

Capítulo 5 – Protótipo Aplicacional

Neste capítulo é descrito com detalhe o protótipo aplicacional desenvolvido, denominado de SPIEW (*Signal Processing Interpolation Educational Workbench*), e que materializa a experimentação interativa do modelo de ensino proposto no capítulo anterior. Neste protótipo, o utilizador poderá testar várias operações típicas na área de processamento digital de sinal, como por exemplo: filtragem, decimação, condicionamento, interpolação, entre outras. Um aspeto positivo do protótipo desenvolvido é que os seus utilizadores podem, em diversas fases, visualizar os resultados graficamente, o que facilita a aquisição e consolidação de conhecimentos que o protótipo aborda.

Pretende-se igualmente que o protótipo aplicacional desenvolvido sirva como ferramenta de teste e simulação para problemas de engenharia na área da reconstrução de sinal. Através de uma análise prévia do condicionamento do problema de reconstrução em estudo, o utilizador saberá se pode obter, ou não, uma solução numérica para o problema, mesmo antes de testar os algoritmos de reconstrução disponíveis no protótipo. Esta característica dá ao utilizador a possibilidade de alterar o condicionamento do problema para situações mais ou menos favoráveis. Na parte da reconstrução propriamente dita e depois de identificados quais os métodos mais favoráveis à obtenção de uma solução numérica para o problema em estudo, o protótipo calcula parâmetros que permitam avaliar a qualidade e o tempo da reconstrução. Isto fornece dados ao utilizador sobre qual o método de reconstrução a escolher quando estamos limitados no tempo e/ou quando nos é imposta uma qualidade mais elevada.

Na secção seguinte é descrita a metodologia de desenvolvimento do protótipo aplicacional.

5.1 METODOLOGIA DE DESENVOLVIMENTO

Desenvolver *software* é um processo de aprendizagem iterativo e o resultado é uma corporização de conhecimento recolhido, destilado e organizado à medida que o processo é conduzido [Miguel03]. Neste caso em questão, foi identificado uma oportunidade de melhorar o processo de aprendizagem dos alunos de engenharia, no caso específico dos problemas de reconstrução de sinal. A ideia original consistia no desenvolvimento de uma ferramenta que, recorrendo a uma abordagem visual e

interativa de determinadas operações de processamento digital de sinal, permitisse aos alunos consolidar os seus conhecimentos, completando o ciclo de aprendizagem inicialmente estipulado.

Na engenharia de *software* existem diversos modelos de desenvolvimento que estão descritos em [Miguel03] e [Sommerville09]. No desenvolvimento da ferramenta optou-se pelo modelo de desenvolvimento em Prototipagem, isto porque, embora tenha sido identificado o problema a abordar, existiam diversas incertezas sobre a forma como abordar ou apresentar diversas operações e respetivos resultados. A ideia foi fazer o desenvolvimento do protótipo da aplicação centrado no utilizador. Para isso foi realizada, em fases intermédias, a recolha de *feedback* dos futuros (ou possíveis) utilizadores, avaliando não só a interação com a ferramenta mas também os conhecimentos adquiridos. Foi através desta interação que foi possível corrigir determinadas falhas e adicionar novas funcionalidades que permitissem melhorar o processo de aprendizagem. O grupo de utilizadores que permitiu fazer este desenvolvimento está caracterizado no capítulo 6.

O modelo de prototipagem (Figura 5.1) inicializa-se com uma *especificação* de requisitos, ou seja, são definidos os objetivos globais para o sistema. Identificam-se os requisitos que são conhecidos no momento e definem-se aspetos a abordar ou implementar no futuro (ou nas próximas versões). Segue-se o desenho do protótipo do sistema que se concentra na representação dos aspetos do *software* visíveis para o utilizador. Este protótipo é avaliado pelo utilizador e depois usado para refinar os requisitos do *software* a ser desenvolvido. Ocorre aqui a iteração apresentada na Figura 5.1, em que o protótipo é refinado para satisfazer as necessidades do cliente/ utilizador e, ao mesmo tempo, usado pelo engenheiro para obter uma compreensão mais profunda dos requisitos do sistema. A prototipagem é uma experimentação ativa com o propósito de obter informação para o processo de desenvolvimento com o objectivo de melhorar o desempenho do produto final.

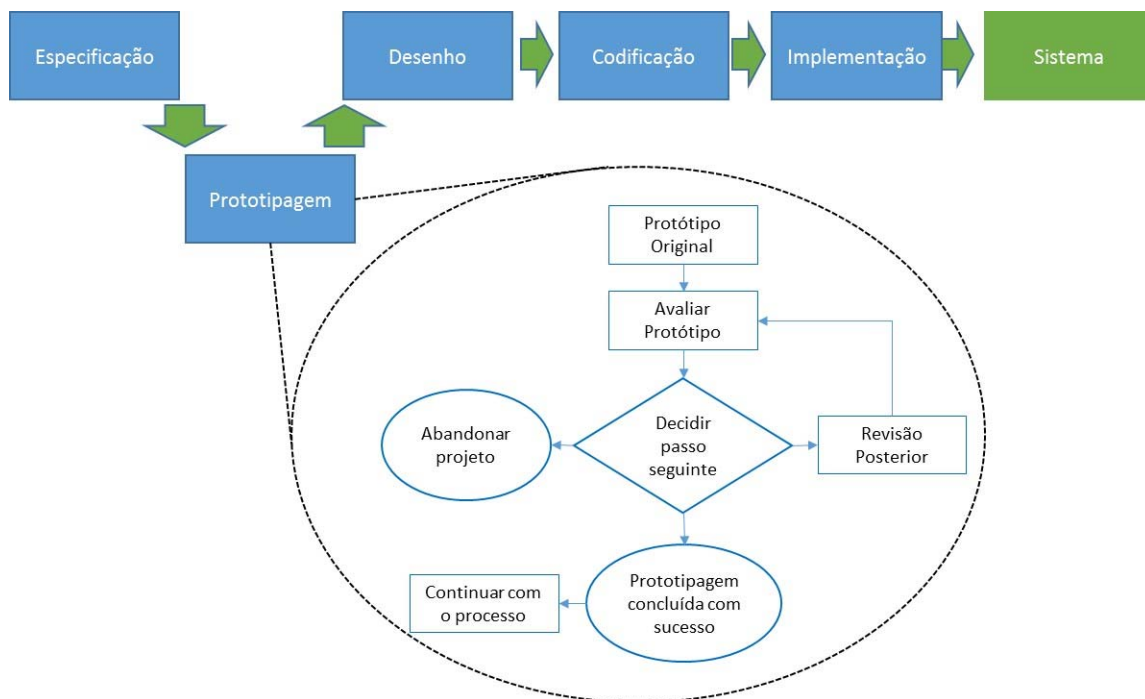


Figura 5.1 – Modelo de prototipagem (adaptado de [Miguel03])

Nas secções seguintes apresentam-se as funcionalidades do protótipo desenvolvido e que implementam as etapas descritas na secção 4.2 do capítulo anterior. De salientar que as operações terão que ser efetuadas numa determinada ordem, razão pela qual as operações vão ficando ativas sequencialmente, impedindo assim que o utilizador tente efetuar operações, sem que outras que as precedem tenham sido completadas.

5.2 SELECÇÃO E EXTRAÇÃO DOS SINAIS

Na Figura 5.2 apresenta-se uma vista da interface do protótipo (SPIEW) que permite implementar a primeira etapa do ciclo de aprendizagem descrito na secção 4.2 do capítulo 4. O SPIEW permite escolher qual o tipo de sinal com que se pretende trabalhar, podendo a escolha ser realizada entre dois tipos de dados disponíveis e identificados na interface por “*Signal Type*”, que são “*Sequence*” ou “*Still Image*”. O processo inicia-se com a operação de extração de sinais, “*Extract Signals*”, implementado o modelo de extração de sinais descrito na secção 4.2.1 (capítulo 4) desta tese. No caso do tipo de sinal *Sequence* (sequência de *frames*) é disponibilizada ainda a funcionalidade “*Play*” que permite visualizar do início até ao fim toda a sequência. Esta funcionalidade não é abilitada caso o tipo de sinal escolhido seja uma imagem.

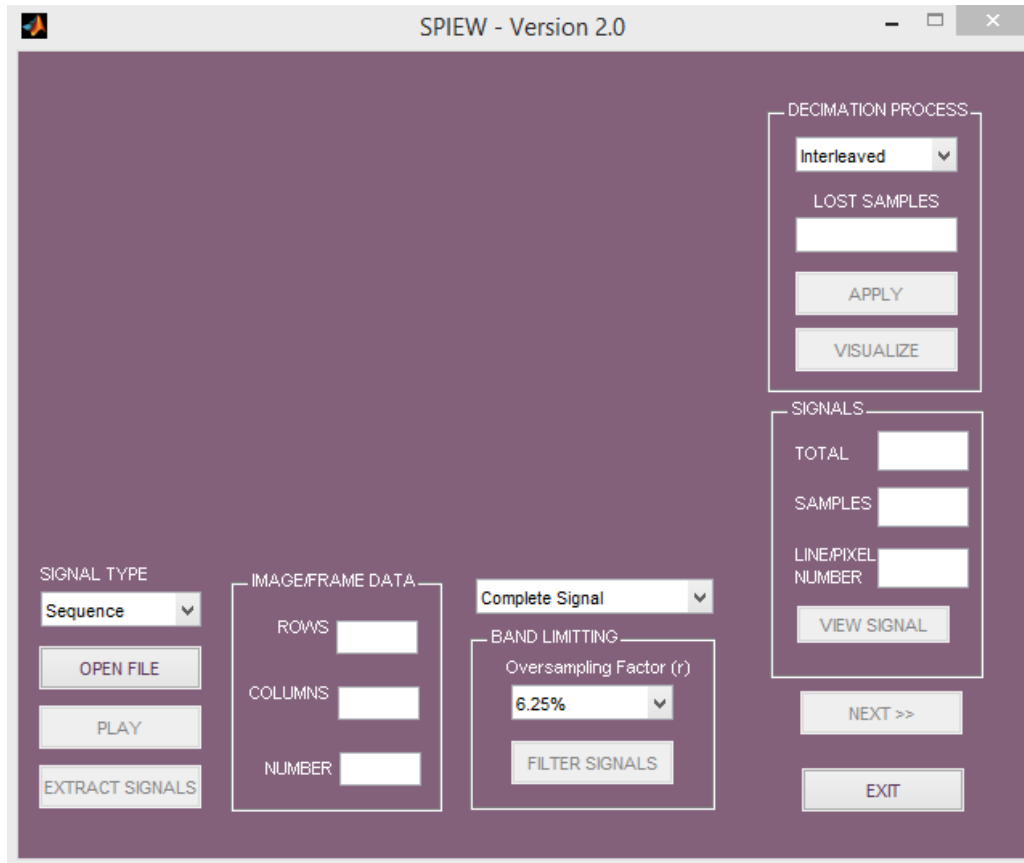


Figura 5.2 – Protótipo aplicativo desenvolvido. Interface de extração e condicionamento dos sinais.

Automaticamente é apresentada a informação relativa ao número de colunas e de linhas do *frame* ou imagem, Figura 5.3. “*Number*” representa o *frame* de uma sequência. No caso do “*Play*” este número é auto-incrementado e indica qual o número do *frame* em *display* na janela de visualização. No caso do sinal em análise ser uma imagem este valor é apresentado como N/A (*Not Available*).

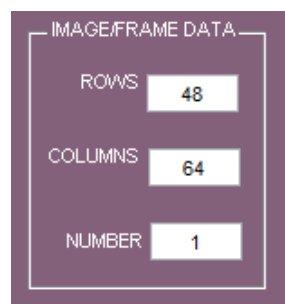


Figura 5.3 – Informação sobre os *frames* ou imagens.

A execução do processo de extração de sinais está identificada na interface por “*Extract Signals*” e é sinalizado por uma barra que acompanha o progresso desta operação, Figura 5.4.

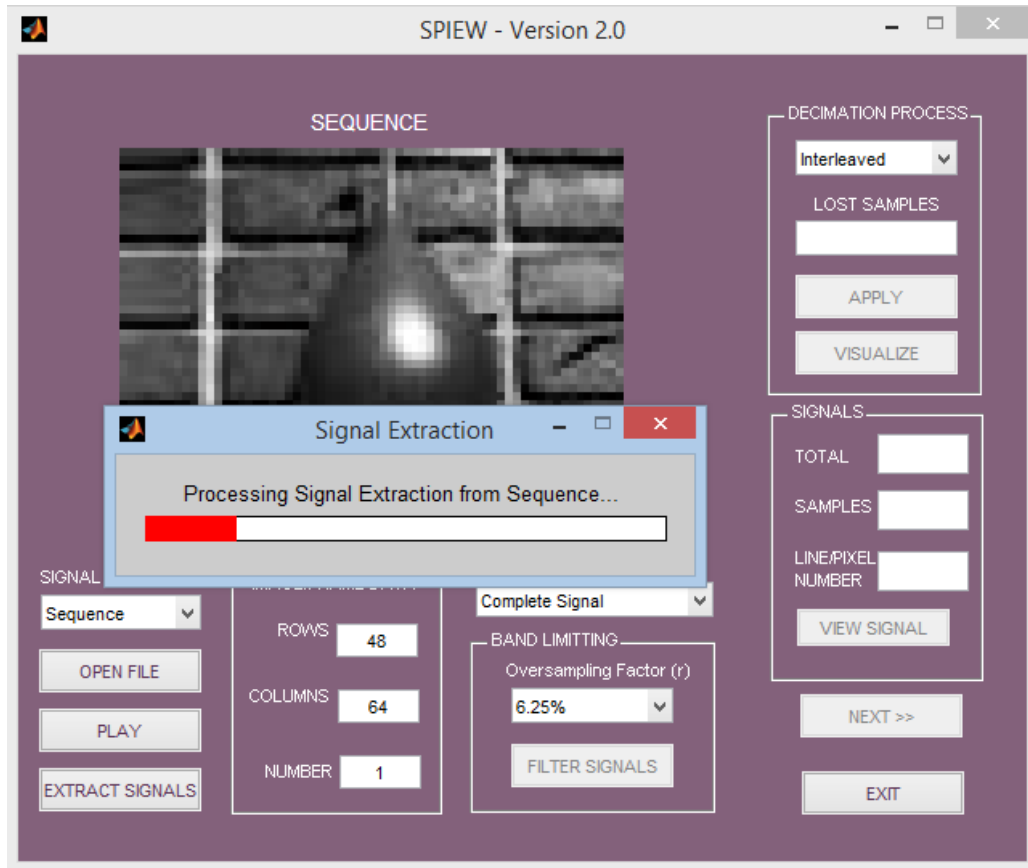


Figura 5.4 – Processo de extração dos sinais.

O número total de sinais extraídos e o número total de amostras desses sinais são apresentados automaticamente, Figura 5.5.

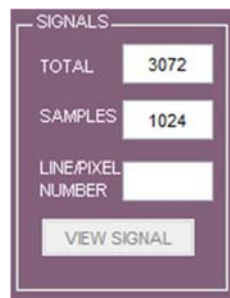


Figura 5.5 – Informação sobre os sinais extraídos.

5.2 LIMITAÇÃO EM FREQUÊNCIA

O processo de limitação de banda dos sinais descrito na secção 4.2.1 está implementado no SPIEW, sendo apresentada no bloco “*Band Limiting*” da interface. O utilizador deve interagir com o SPIEW e efetuar a referida operação de acordo com os objetivos estabelecidos, ajustando o valor para o fator de sobre-amostragem (r), que corresponderá a uma largura de banda (LB) específica no FPB (Figura 5.6). A operação de limitação em frequência dos sinais é executada recorrendo ao uso do FPB. A

apresentação das três versões do sinal selecionado: original, observado e corrompido, permite que o utilizador tenha uma percepção visual das alterações introduzidas no sinal e do seu comportamento.

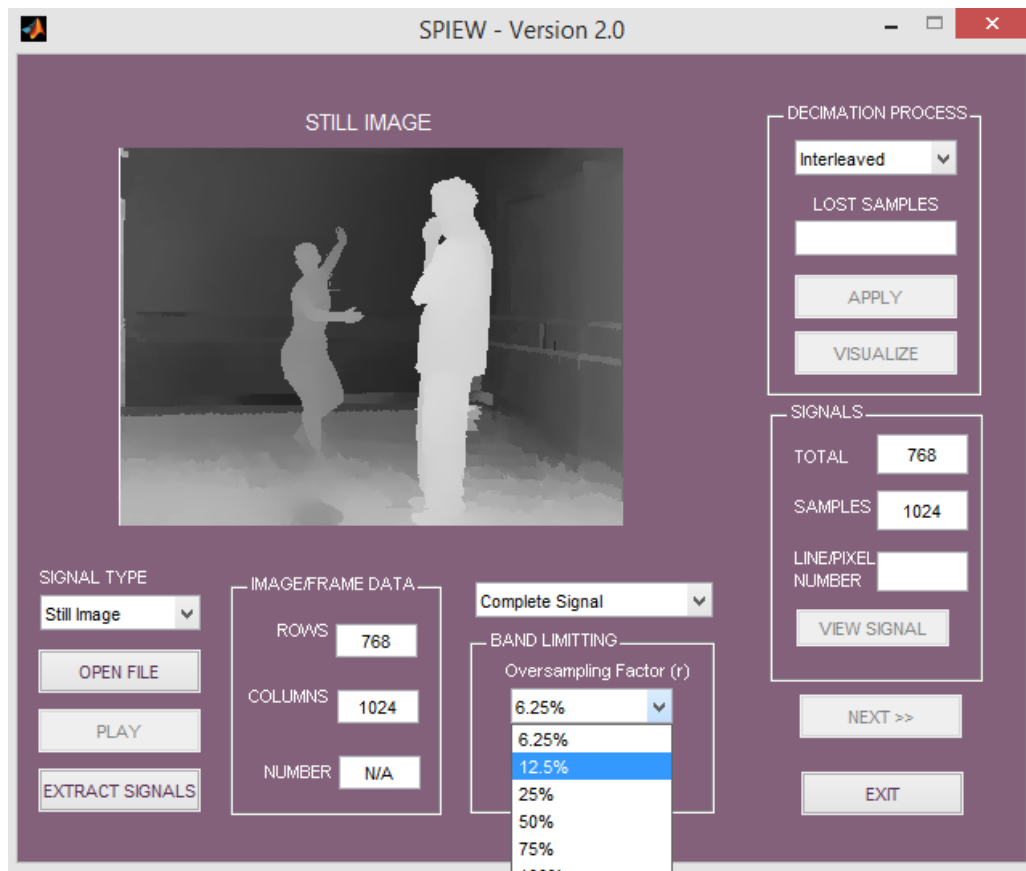


Figura 5.6 – Operação de limitação em frequência. Escolha do fator de sobre-amostragem.

O processo de limitação em frequência dos sinais extraídos é acompanhado por uma caixa de progresso que indica a evolução desta operação (Figura 5.7(a)). O SPIEW passa a apresentar uma caixa de mensagem onde é exibido o valor da LB do FPB utilizado, que corresponde ao fator de sobre-amostragem r escolhido anteriormente (Figura 5.7(b)). É importante referir que o valor de r escolhido determina se os sinais serão mais ou menos sobre-amostrados. Um valor de r alto implica que são necessárias mais amostras para representar o sinal e, sendo assim, o utilizador poderá perder menos amostras no processo de decimação. Já com um valor baixo de r os sinais obtidos serão sobre-amostrados e no processo de decimação e o número de amostras que o sinal perderá será mais elevado.

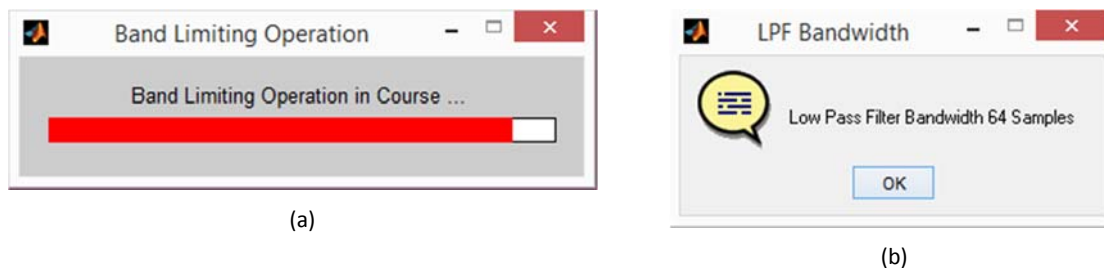


Figura 5.7 – Limitação em frequência: janela de progresso (a) e largura de banda do filtro passa baixo utilizado (b).

5.4 DECIMAÇÃO DOS SINAIS

Após concluída a fase de limitação em frequência dos sinais será realizada a operação de decimação, de acordo com o exposto no capítulo anterior (secção 4.2.1), onde estão descritos com mais pormenor os métodos utilizados para implementar as geometrias de erro disponíveis. Lembra-se, no entanto, que o utilizador poderá escolher uma das três geometrias de padrão de erro disponíveis no SPIEW: entrelaçada, aleatória ou contígua (Figura 5.8).



Figura 5.8 – Processo de Decimação - geometrias de padrão de erro disponíveis.

No caso da geometria de erro “*Interleaved*” (entrelaçada), o SPIEW apresentará uma janela onde deverá ser indicado o valor do entrelaçamento ou salto. Conforme indicado na secção 4.2.1, apenas os valores 2, 4 ou 8 são permitidos (Figura 5.9).

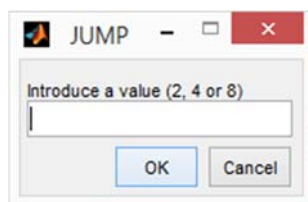


Figura 5.9 – Decimação entrelaçada, janela para escolher o valor de entrelaçamento.

Caso o utilizador escolha uma das outras opções disponíveis, “*Random*” (aleatória) ou “*Contiguous*” (contínua ou rajada), deverá também indicar, nestes casos, qual o número de amostras que pretende perder (ou eliminar). Para inicializar o processo de decimação, deve clicar no botão “*Apply*”. Quando o processo estiver terminado irá aparecer uma mensagem a indicar a sua conclusão, conforme se pode visualizar na Figura 5.10. De referir que o número de amostras perdidas no processo de decimação irá

afetar diretamente a convergência dos métodos de reconstrução disponíveis na aplicação, conforme será demonstrado na secção de validação do protótipo aplicacional, secção 5.8.

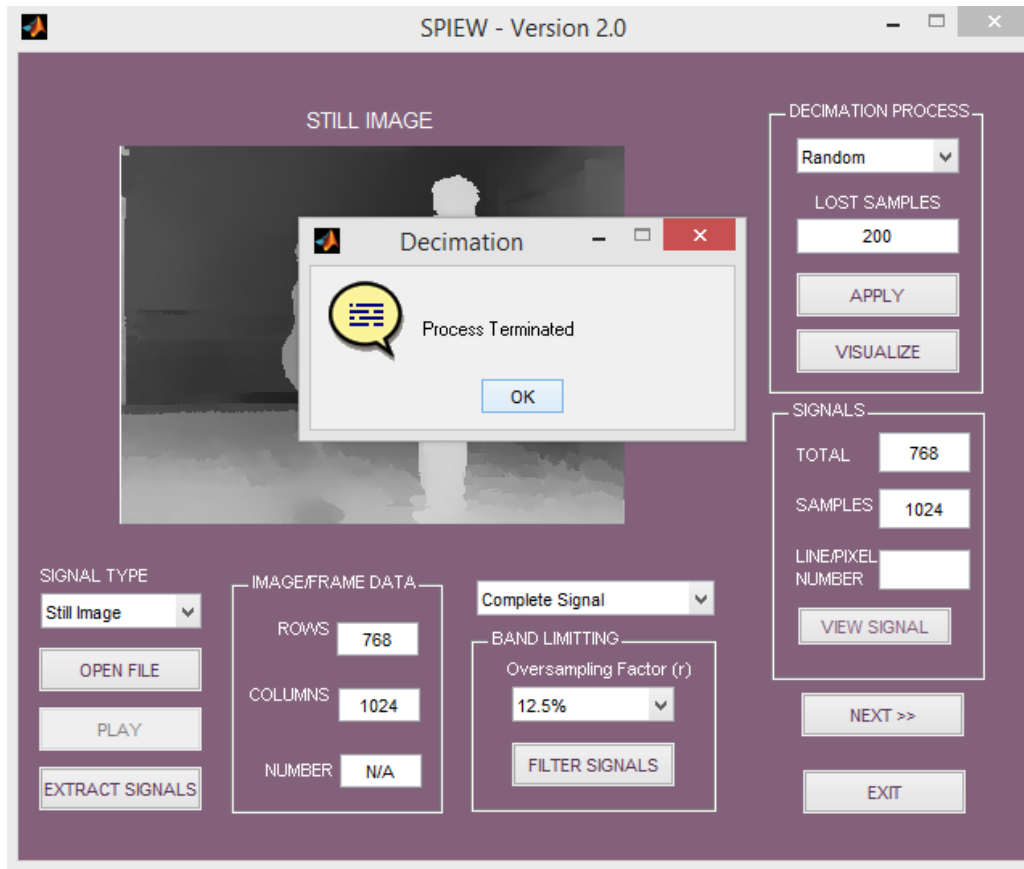


Figura 5.10 – Decimação aleatória, 200 amostras perdidas, processo concluído.

Durante os testes e recolha de *feedback* dos utilizadores foi possível detetar que existia dificuldade em compreenderem de que forma cada geometria de padrão de erros afetava os sinais. Para fazer face a esta dificuldade o SPIEW reproduz gráficamente os sinais obtidos após a operação de decimação (sinal decimado), no primeiro *frame* da sequência de imagens ou na imagem selecionada. Na Figura 5.11 pode observar-se o resultado da corrupção efetuada na imagem da Figura 5.6, utilizando a geometria de padrão de erros aleatória.

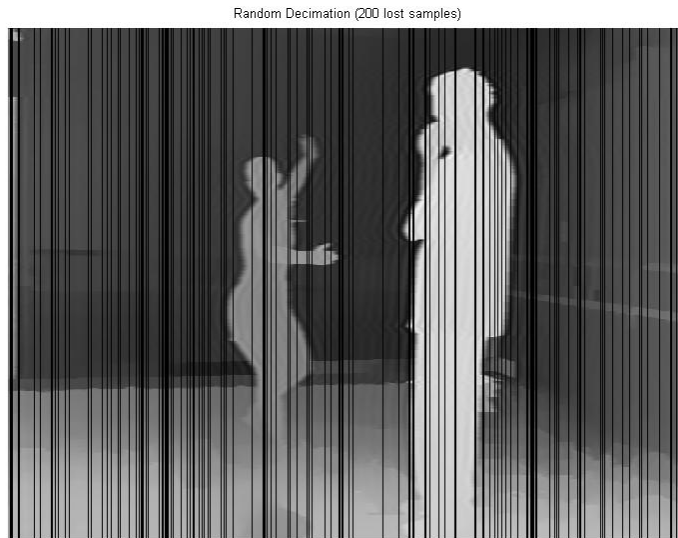


Figura 5.11 – Imagem após da decimação aleatória, 200 amostras perdidas por linha.

Depois de concluído o processo de decimação ficam criadas as três versões dos sinais extraídos: original, observado e decimado. Através da interface do SPIEW é possível, a partir de agora, observar um dos sinais extraídos, e respectivas versões, sendo necessário indicar qual o número correspondente ao *pixel*, no caso de sequência, ou à linha no caso de uma imagem. O sinal escolhido e as suas três versões: original, observado e decimado serão mostrados numa janela, como se pode ver na Figura 5.12.

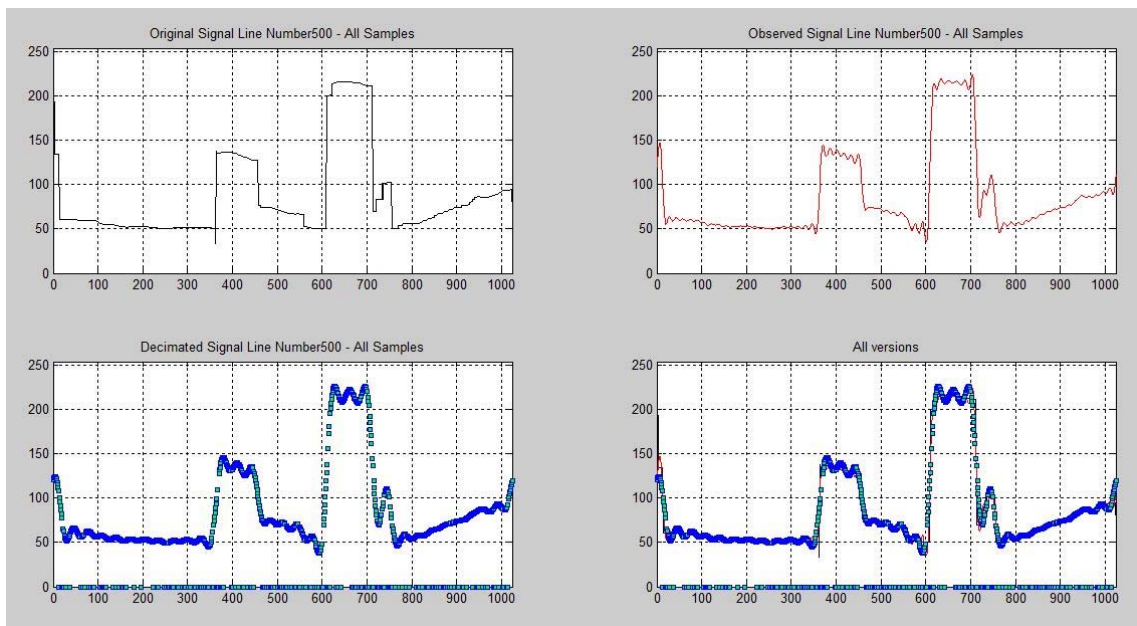


Figura 5.12 – Versões: original, observado e decimado do sinal correspondente linha 500 extraída de uma imagem.

5.5 ANÁLISE DO CONDICIONAMENTO DO PROBLEMA

Terminada a fase de condicionamento do problema, é possível agora testar os algoritmos disponíveis no SPIEW. Como já foi referido capítulo anterior, é objetivo deste trabalho testar métodos de resolução iterativos, diretos e semi-iterativos, tendo sido, para o efeito, implementados os seus algoritmos no SPIEW. Esta interface implementa a segunda etapa do ciclo de aprendizagem descrito na secção 4.2.2 do capítulo 4.

O SPIEW foi desenhado para interagir com o utilizador e permitir analisar e observar as seguintes características:

- Analisar o condicionamento do problema previamente efetuado;
- Analisar convergência dos diversos métodos;
- Utilizar os diversos métodos para reconstruir os sinais;
- Avaliar a qualidade de reconstrução dos sinais,
- Avaliar a velocidade de reconstrução dos sinais.

Na Figura 5.13 apresenta-se a interface do SPIEW para a fase de reconstrução de sinais

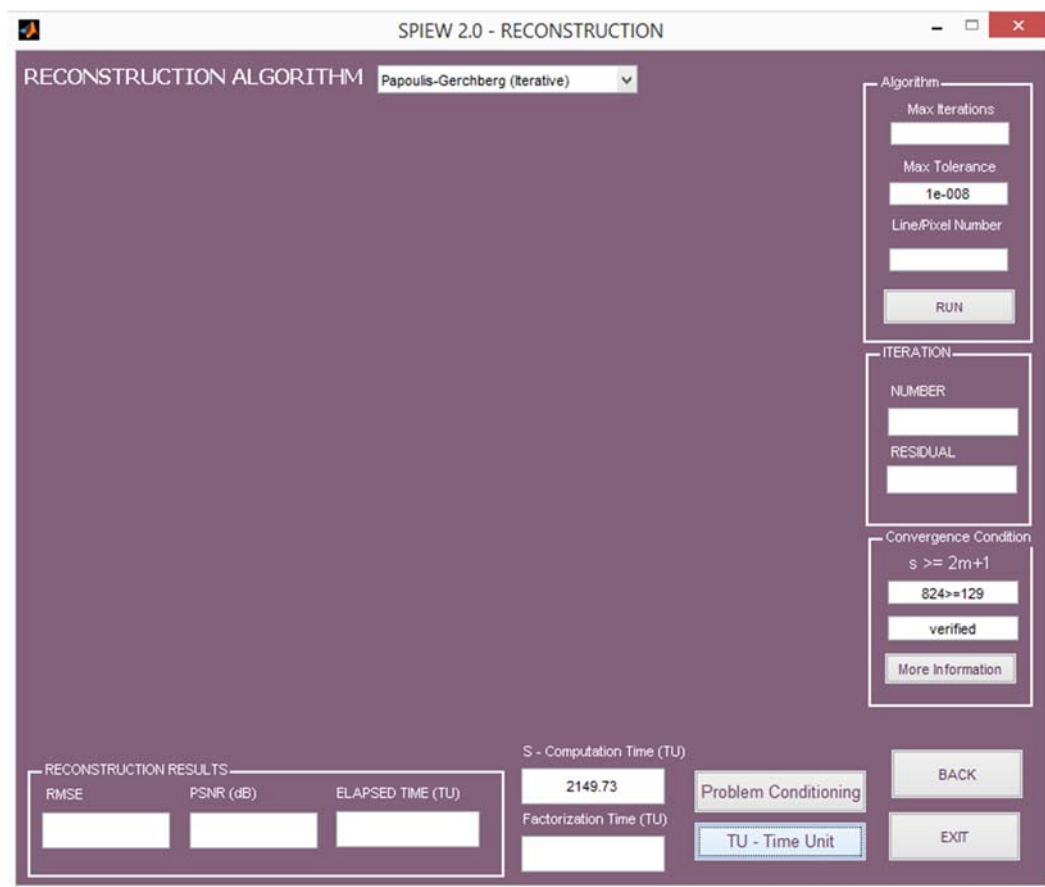


Figura 5.13 – SPIEW, Janela de Reconstrução.

Para avaliar o condicionamento do problema, que irá permitir saber antecipadamente se será possível, ou não, obter uma solução numérica para o problema, será utilizada a opção “*Problem Conditioning*”, visível na Figura 5.13. Aparece então uma nova janela onde se poderão ver calculados o raio espectral da matriz S (Eq. 3.13) e o número de condição da matriz $(I - S)$ (Eq. 3.14), que se apresentam na Figura 5.14. Através da análise do valor destes dois parâmetros será tomada a decisão de quais os métodos a escolher para reconstruir os sinais.

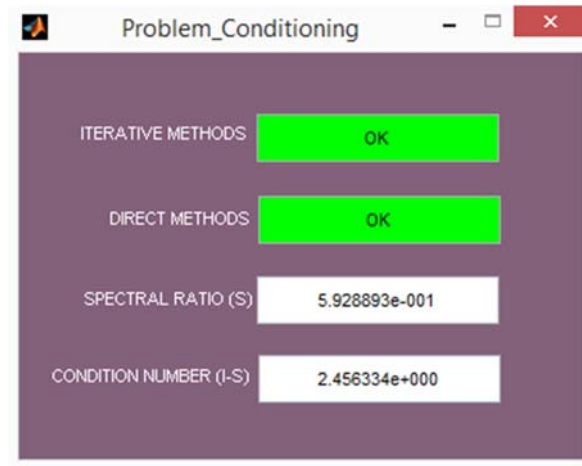


Figura 5.14 – Análise do condicionamento do problema.

Para os métodos iterativos deve analisar-se o valor do raio espectral de S , este valor deve ser menor do que um. Conforme o indicado na secção 2.5 desta tese, a condição necessária e suficiente para a convergência de um método iterativo, é o raio espectral da matriz de iteração ser <1 . Se esse valor for menor do que 1, o SPIEW indica que é possível utilizar os métodos iterativos para tentar obter a solução numérica do problema. Se o valor do raio espectral for 1, o SPIEW indica que os métodos iterativos não devem ser utilizados, uma vez que a convergência dos algoritmos não é garantida.

Relativamente aos métodos diretos, deverá ser feita a análise do número de condição da matriz $(I - S)$. Conforme o referido na secção 2.3 desta tese, o número de condição de um problema é uma medida que indica se o problema está bem ou mal condicionado para ser tratado numericamente. Um número de condição pequeno indica um problema bem condicionado. Um problema com um número de condição elevado é mal condicionado. Neste caso, um valor na ordem de 10^{15} (secção 3.2) será um valor elevado, e significa que o problema está mal condicionado e, como tal, a obtenção de uma solução numérica através dos métodos diretos não é garantida. Este valor está diretamente relacionado com a precisão da máquina. Estes dados poderão ser comprovados na secção 5.8 onde são apresentados os resultados da validação do protótipo SPIEW.

No entanto, como o objetivo é que o utilizador possa testar todas as possíveis soluções, embora seja alertado que não deve utilizar determinado conjunto de métodos, o SPIEW permite optar por testar

os algoritmos/métodos mesmo em condições adversas à sua convergência. Isto permite ao utilizador verificar que os sinais não são reconstruídos quando as condições de convergência não estão garantidas.

A condição de convergência dos algoritmos, apresentada na equação (3.8), secção 3.1.5, é verificada no SPIEW, indicando se é ou não verificada (Figura 5.15(a)). É ainda possível obter mais informação sobre todos os parâmetros desta equação (Figura 5.15(b)) através da opção “*More Information*”.



Figura 5.15 – Análise da condição de convergência (a) e respetiva legenda (b).

5.6 RECONSTRUÇÃO DOS SINAIS

Conforme referido na secção anterior deste capítulo, deverá ser feita a análise do condicionamento do problema antes de escolher qualquer um dos métodos de reconstrução disponíveis no protótipo aplicacional (Figura 5.16). No caso de ambos os parâmetros, raio espectral e número de condição, possuírem valores dentro da gama de convergência, o utilizador poderá escolher qualquer um dos métodos de reconstrução disponíveis.

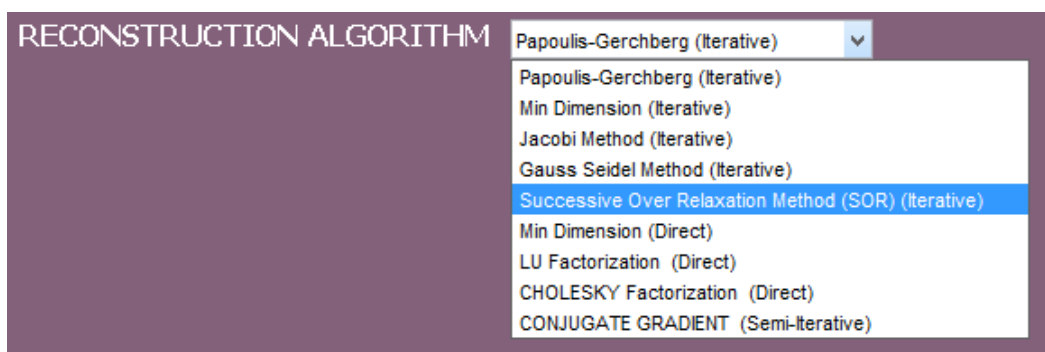


Figura 5.16 – Métodos de reconstrução disponíveis.

Conforme indicado na secção 4.2.2 desta tese, existem três tipos de métodos de reconstrução disponíveis na aplicação: os diretos, os iterativos, e ainda um método semi-iterativo. Todos eles estão descritos com maior detalhe nos capítulos 2 e 3.

Nas figuras seguintes, apresentam-se os resultados da reconstrução com: um método iterativo (Figura 5.17), um método direto (Figura 5.18) e um método semi-iterativo (5.19). Foi realizada, a título exemplificativo, uma experiência de reconstrução utilizando os 3 métodos de reconstrução com o mesmo sinal. No caso dos métodos iterativos, foi necessário indicar o número máximo de iterações que poderá ser executado, que funciona como um dos critérios de paragem dos algoritmos iterativos, o outro é a tolerância máxima entre iterações consecutivas, que por norma é 10^{-8} .

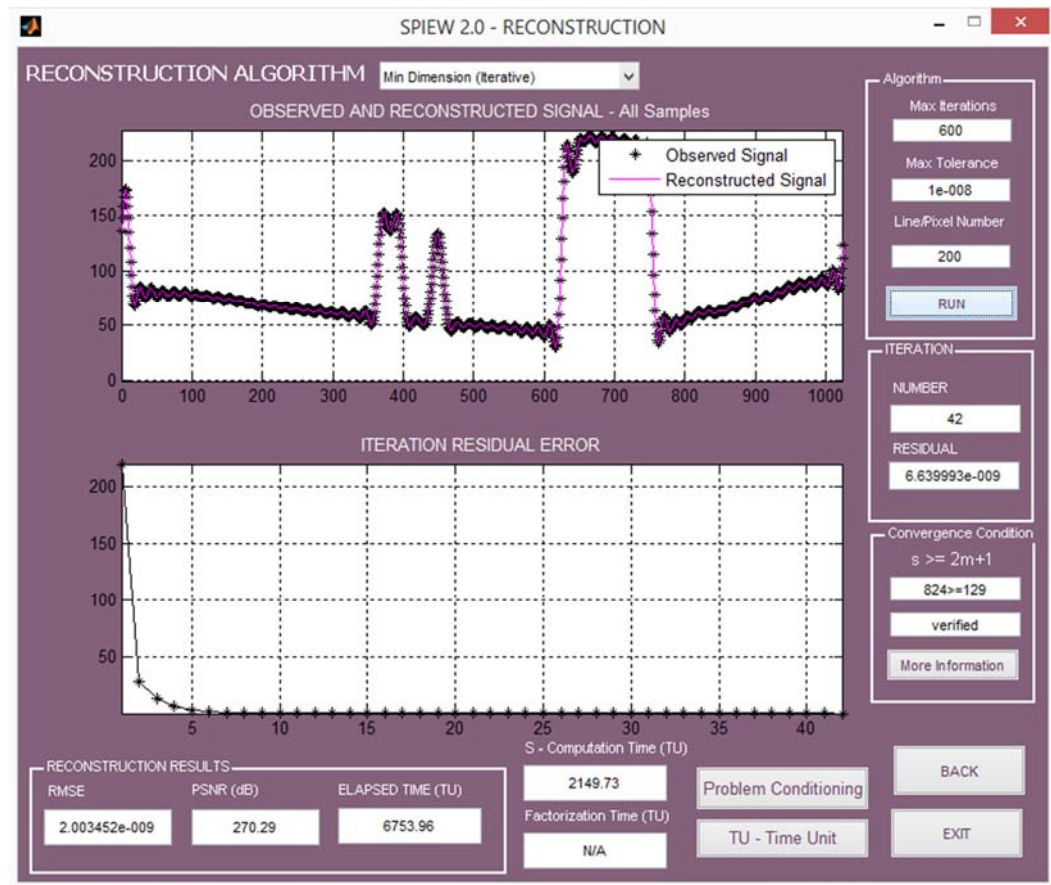


Figura 5.17 – Resultados da reconstrução da Linha 200, método dimensão mínima - iterativo.

Na figura anterior (Figura 5.17) pode observar-se o resultado da reconstrução da linha 200 de uma imagem, através de um método iterativo. Nela pode verificar-se que a reconstrução foi bem-sucedida ao fim de 42 iterações, o residual é próximo de zero a partir da quinta iteração e que a qualidade do sinal reconstruído é superior a 270 dB.

Um outro aspeto a ter em atenção na reconstrução dos sinais são os tempos medidos em TU: o tempo de reconstrução e o tempo necessário para o cálculo das diversas matrizes necessárias para a utilização dos diferentes algoritmos/métodos.

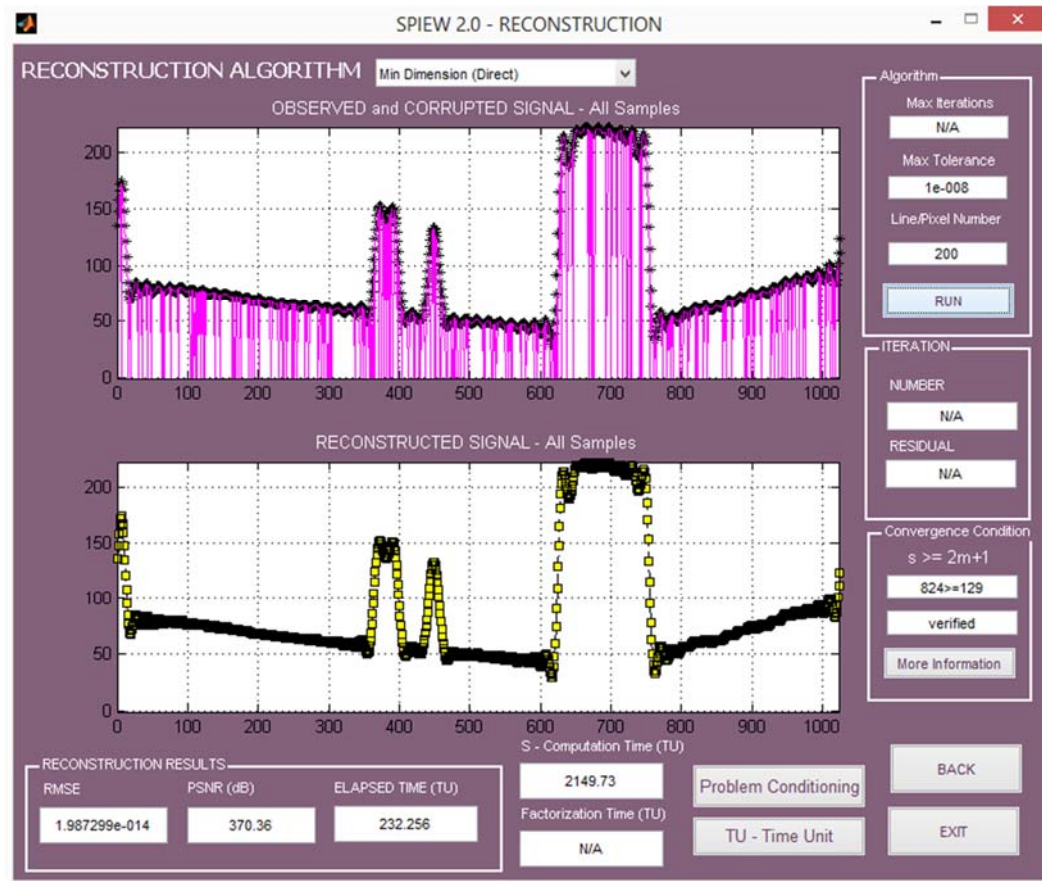


Figura 5.18 – Resultados da reconstrução da Linha 200, método direto.

Na Figura 5.18, está o mesmo sinal referido anteriormente, mas agora reconstruído através de um método direto. Nela podem ser visualizadas as versões do sinal corrompido antes e após a reconstrução. Verifica-se igualmente que o sinal reconstruído possui um PSNR superior a 350 dB. Relativamente a tempos, o tempo de cálculo de S é igual à reconstrução da Figura 5.17, mas o tempo de reconstrução é substancialmente mais baixo, conforme seria de esperar, uma vez que foi utilizado um método de resolução direto.

Na figura seguinte (Figura 5.19) apresentam-se os resultados da reconstrução do mesmo sinal, mas agora com a utilização do método semi-iterativo disponível no SPIEW. Podem observar-se as duas versões do sinal: corrompido e reconstruído, bem como o número de iterações necessárias para atingir a solução (neste caso 12 iterações) e o erro residual do método no final da reconstrução. O valor do PSNR do sinal reconstruído é superior a 230 dB, sendo que o tempo de reconstrução é bastante inferior quando comparado com o do método iterativo (Figura 5.17).

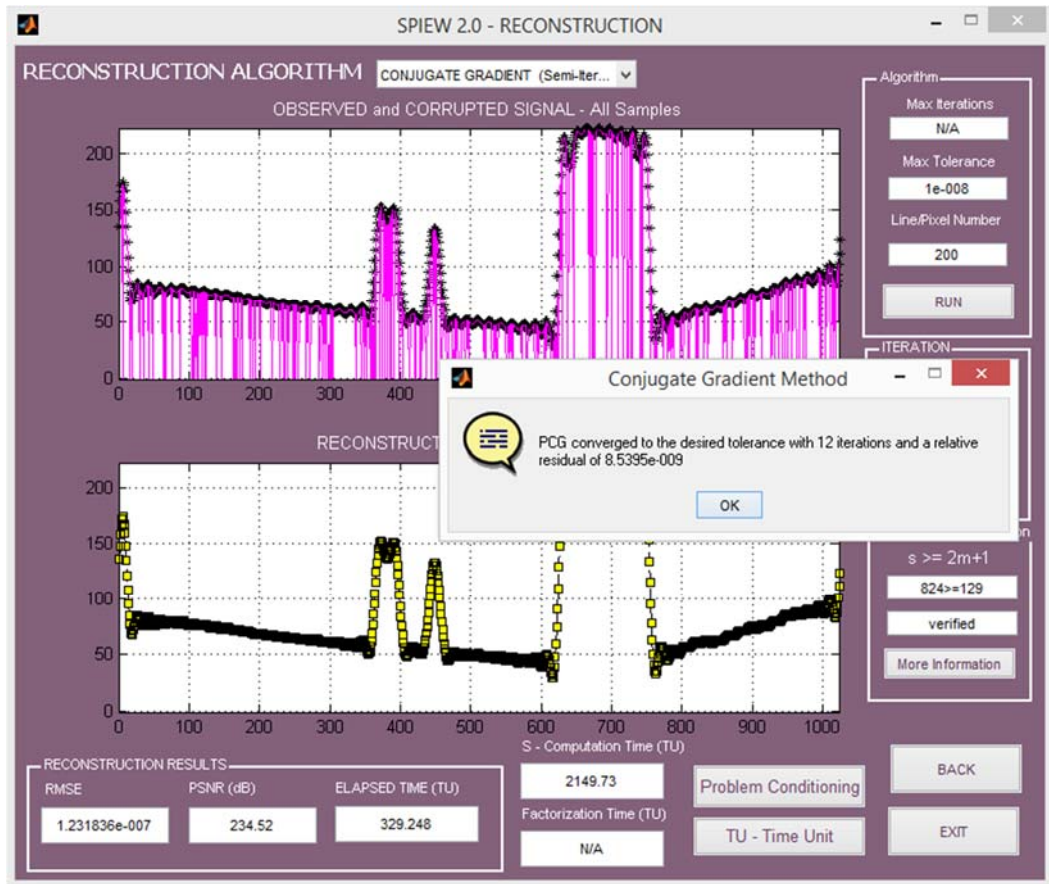


Figura 5.19 – Resultados da reconstrução da Linha 200, método Gradientes Conjugados.

5.6 AVALIAÇÃO OBJETIVA DOS RESULTADOS DA RECONSTRUÇÃO

Conforme indicado na secção 4.2.2 do capítulo anterior, para avaliar a reconstrução do sinal, o SPIEW utiliza os parâmetros: *Root Mean Square Error* (RMSE) e *Peak Signal to Noise Ratio* (PSNR), calculados através das equações 4.2 e 4.1 respetivamente, e calcula ainda o tempo de reconstrução expresso em *Time Units* (TU).

O parâmetro TU foi utilizado em detrimento do tempo em segundos, para tornar a execução independente da velocidade da máquina. Assim, o tempo de referência é o tempo que a máquina demora a executar uma FFT (*Fast Fourier Transform*), isto porque esta operação é crucial para os algoritmos escolhidos.

A opção “TU – Time Unit” permite obter informação mais detalhada sobre este parâmetro, como por exemplo o TU de referência para cada reconstrução (Figura 5.20). Este valor depende do processador da máquina e dos sinais utilizados.

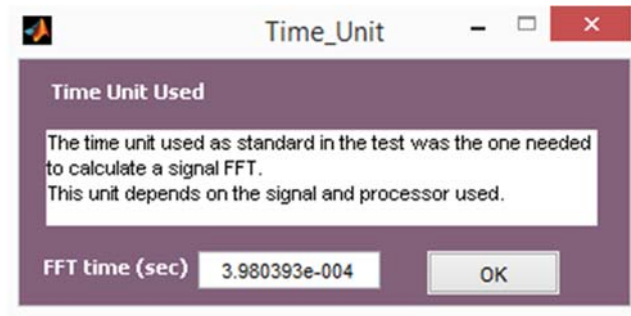


Figura 5.20 – Informação sobre o TU de referência a ser utilizado na reconstrução.

Já os outros parâmetros, RMSE e PSNR avaliam a qualidade do sinal reconstruído. Quanto mais elevado for o valor de PSNR, mais qualidade terá o sinal reconstruído. Os valores de cada um dos parâmetros referido anteriormente são visíveis na Figura 5.21.

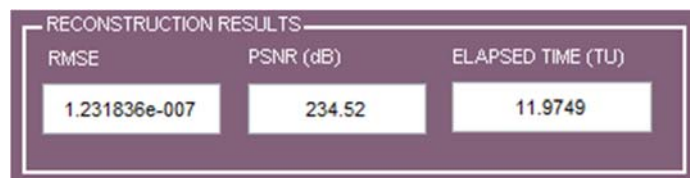


Figura 5.21 – Parâmetros para avaliar a qualidade e velocidade de reconstrução.

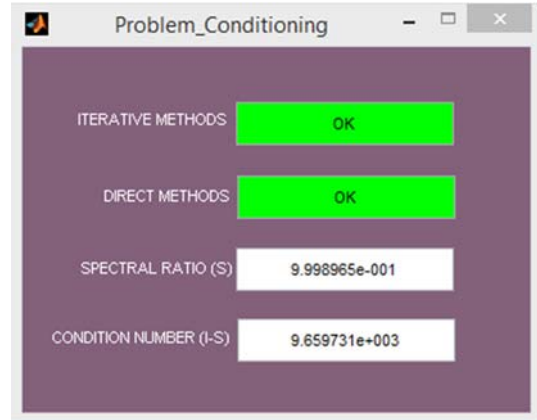
Para além do valor do tempo de reconstrução expresso em TU, o SPIEW avalia ainda o tempo de cálculo da matriz S do algoritmo de dimensão mínima, que é necessário para todos os métodos, exceto para o algoritmo de Papoulis-Gerchberg. No caso de métodos que necessitem de tempo para cálculo de outras matrizes, como por exemplo os métodos de fatorização, a aplicação apresentará igualmente o tempo de cálculo das referidas matrizes. Importa referir que a unidade de tempo utilizada para medir os tempos referidos anteriormente são as Time Units (TU).

Na secção seguinte são apresentados alguns testes realizados na SPIEW para que sejam comprovados as condições de aplicabilidade dos diferentes métodos.

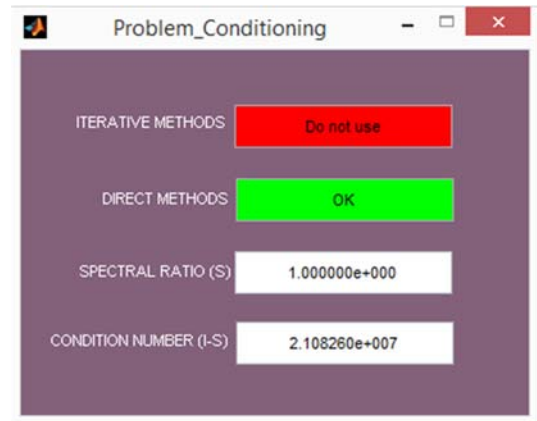
5.8 VALIDAÇÃO DA ROBUSTEZ DO PROTÓTIPO APLICACIONAL - SPIEW

Para testar a preponderância da operação de banda na decimação dos sinais optou-se por realizar os testes utilizando a geometria de padrão de erros menos favorável em problemas de engenharia: a contígua. O valor de r utilizado foi 12,5%. Um valor mais elevado limitará o número de amostras que se poderiam perder. Por exemplo, na Figura 5.22 são apresentadas as análises do condicionamento perdendo respetivamente 30, 50 ou 100 de forma contígua, com $r = 12,5\%$.

(a) 30 Amostras perdidas



(b) 50 Amostras perdidas



(c) 100 Amostras perdidas

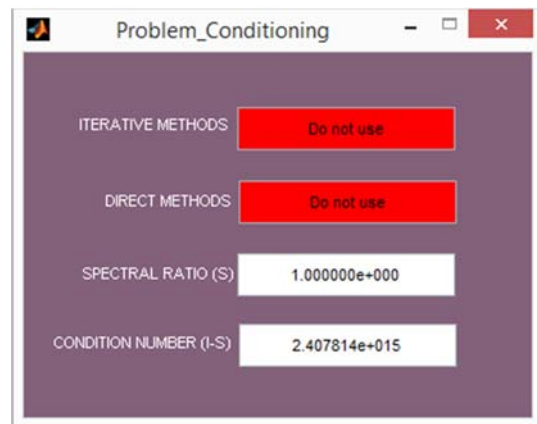
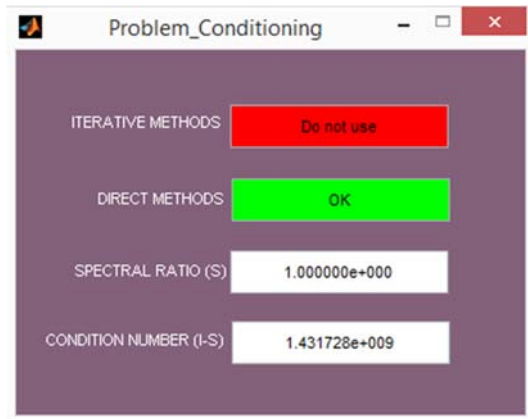


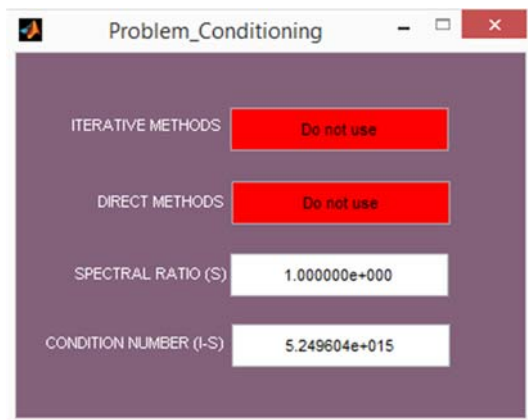
Figura 5.22 – Avaliação do condicionamento do problema perdendo 30 (a) 50 (b) ou 100 (c) amostras na decimação contígua, $r = 12,5\%$.

Alterando agora o valor de r para 25%, os resultados para o mesmo número de amostras perdidas (30, 50 e 100) é o apresentado na Figura 5.23.

(a) 30 Amostras perdidas



(b) 50 Amostras perdidas



(c) 100 Amostras perdidas

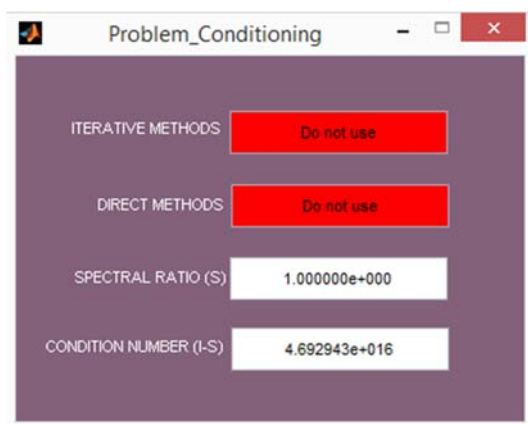


Figura 5.23 – Avaliação do condicionamento do problema perdendo 30 (a) 50 (b) ou 100 (c) amostras na decimação contígua, $r = 25\%$.

Observando as Figuras 5.22 e 5.23 o utilizador pode comprovar que o aumento do valor de r condiciona o número de amostras que poderemos perder na operação de decimação. Conforme seria de esperar, com um r mais elevado, o número de amostras que podemos perder é menor, o que condiciona o uso de alguns dos métodos de reconstrução disponíveis no SPIEW e conseqüentemente, a obtenção de uma solução numérica para o problema de reconstrução.

Depois e comprovada a influência do fator de sobre amostragem no número de amostras perdidas, pode agora verificar-se o desempenho dos diversos algoritmos em situações de condicionamento distintas.

5.8.1 Resultados da Reconstrução – Decimação Contígua (Rajada)

Nas tabelas seguintes, apresentam-se os resultados da reconstrução do sinal correspondentes à linha 500 do ficheiro indicado. Foram testados todos os métodos existentes no SPIEW. As condições iniciais e de condicionamento foram as seguintes:

- Ficheiro: *depth-cam0-f000.bmp*, Linha 500;
- $r=12,5\%$, Largura do FPB 64 amostras;
- Número máximo de Iterações = 600; Tolerância máxima = $10e-8$;
- Decimação contígua – 20, 50 e 100 amostras perdidas;
- Condição de convergência verificada em todos os casos;
- $TU = 1,050961e-003$.

Na Tabela 5.1 apresentam-se os resultados da reconstrução do sinal com 20 amostras perdidas. O valor do raio espectral da matriz S é $9,957257e-001$ e o número de condição da matriz $(I - S)$ é $2,339560e+002$. Através da análise destes dois parâmetros, verifica-se que poderão ser usados todos os métodos disponíveis no SPIEW para obter a solução numérica para o problema em estudo.

Tabela 5.1 – Resultados da reconstrução, linha 500, decimação contígua, 20 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	4,02844E-01	104,23	3,89E+04	7,21E+02	N/A	600	2,91333E-02
Min Dimension Iterativo	2,78050E+00	87,45	3,52E+04	7,21E+02	N/A	600	3,02207E-02
Jacobi Method	1,92623E+00	90,64	3,41E+04	7,21E+02	3,00E-02	600	2,39683E-02
Gauss Seidel Method	1,00879E-01	116,25	3,43E+04	7,21E+02	6,92E+01	600	2,52708E-03
SOR ($\omega = 1,8202$)	5,89029E-08	240,93	1,14E+04	7,21E+02	4,78E+01	200	9,07815E-09
Min Dimension Direct	2,44915E-12	328,55	9,20E+01	7,21E+02	N/A	N/A	N/A
LU Factorization	1,72706E-12	331,58	2,44E-02	7,21E+02	4,00E-02	N/A	N/A
CHOLESKY Factorization	2,43139E-12	328,61	8,23E-02	7,21E+02	3,00E-02	N/A	N/A
CONJUGATE GRADIENT	2,12364E-12	329,79	1,28E+02	7,21E+02	N/A	7	1,05850E-12

Na Tabela 5.2 apresentam-se os resultados da reconstrução do sinal com 50 amostras perdidas. O valor do raio espectral da matriz S é 1 e o número de condição da matriz $(I - S)$ é $2,108260e+007$.

Verifica-se que os métodos iterativos não deveriam ser utilizados. No entanto, verifica-se que o parâmetro que condiciona a utilização dos métodos diretos (número de condição de $I-S$) indica que os mesmos poderão ser utilizados.

Tabela 5.2 – Resultados da reconstrução, linha 500, decimação contígua, 50 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	8,66006E+01	57,58	4,10E+04	5,96E+02	N/A	600	9,12019E-03
Min Dimension Iterativo	8,60020E+01	57,64	3,44E+04	5,96E+02	N/A	600	7,03421E-03
Jacobi Method	8,57427E+01	57,67	3,56E+04	5,96E+02	3,00E-02	600	7,79132E-03
Gauss Seidel Method	8,35163E+01	57,89	3,47E+04	5,96E+02	3,00E-01	600	1,25840E-02
SOR ($\omega = 1,9993$)	4,12019E-01	104,03	3,39E+04	5,96E+02	9,50E-01	600	3,50265E+01
Min Dimension Direct	1,55091E-07	232,52	1,91E-01	5,96E+02	N/A	N/A	N/A
LU Factorization	2,39651E-07	228,74	4,43E-02	5,96E+02	7,00E-02	N/A	N/A
CHOLESKY Factorization	1,56798E-07	232,42	8,30E-02	5,96E+02	5,00E-02	N/A	N/A
CONJUGATE GRADIENT	2,77153E-07	227,48	4,73E+00	5,96E+02	N/A	19	2,82270E-09

Na Tabela 5.3 apresentam-se os resultados da reconstrução do sinal com 100 amostras perdidas. O valor do raio espectral da matriz S é 1 e o número de condição da matriz $(I - S)$ é $2,407814e+015$. Verifica-se que nenhum dos métodos garante a obtenção de uma solução numérica para o problema. Mesmo o método semi-iterativo existente no SPIEW, método dos gradientes conjugados, não convergiu com o máximo de iterações.

Tabela 5.3 – Resultados da reconstrução, linha 500, decimação contígua, 100 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	1,94681E+02	50,54	3,78E+04	7,08E+02	N/A	600	2,04673E-02
Min Dimension Iterativo	1,96041E+02	50,48	3,87E+04	7,08E+02	N/A	600	1,11882E-02
Jacobi Method	1,95637E+02	50,50	3,67E+04	7,08E+02	4,00E-02	600	1,22127E-02
Gauss Seidel Method	1,92585E+02	50,64	3,70E+04	7,08E+02	1,13E+00	600	2,15478E-02
SOR ($\omega = 2$)	6,32753E+00	80,30	3,99E+04	7,08E+02	2,77E+00	600	6,96665E+01
Min Dimension Direct	4,44097E+01	63,38	5,35E-01	7,08E+02	N/A	N/A	N/A
LU Factorization	5,25205E+01	61,92	1,23E-01	7,08E+02	2,90E-01	N/A	N/A
CHOLESKY Factorization	4,49070E+01	63,28	1,04E-01	7,08E+02	1,60E-01	N/A	N/A
CONJUGATE GRADIENT	1,57610E+02	52,38	2,92E+00	7,08E+02	N/A	N/C	N/C

N/C – não convergiu com o máximo de iterações

5.8.2 Resultados da Reconstrução – Decimação Entrelaçada

Nas tabelas seguintes, apresentam-se os resultados da reconstrução do sinal correspondente à linha 500 do ficheiro indicado. Foram testados todos os métodos existentes no SPIEW. As condições iniciais e de condicionamento foram as seguintes:

- Ficheiro: *depth-cam0-f000.bmp*, Linha 500;
- $r=12,5\%$, Largura do FPB 64 amostras;
- Número máximo de Iterações = 600; Tolerância máxima = $10e-8$;
- Decimação entrelaçada – 512, 768, 896 amostras perdidas;
- Condição de convergência verificada exceto para 896 amostras perdidas;
- $TU = 1,055687e-003$.

Na Tabela 5.4 apresentam-se os resultados da reconstrução do sinal com 512 amostras perdidas. O valor do raio espectral da matriz S é 0.5 e o número de condição da matriz $(I - S)$ é 2. Através da análise destes dois parâmetros, verifica-se que poderão ser usados todos os métodos disponíveis no SPIEW para obter a solução numérica para o problema em estudo.

Tabela 5.4 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump =2), 512 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	4,02037E-08	244,24	8,02E+03	5,01E+03	N/A	35	6,51104E-09
Min Dimension Iterativo	4,02041E-08	244,24	5,69E+03	5,01E+03	N/A	34	6,50624E-09
Jacobi Method	2,81979E-08	247,33	6,49E+03	5,01E+03	7,26E+00	29	6,10032E-09
Gauss Seidel Method	1,34910E-10	293,73	4,89E+03	5,01E+03	1,55E+03	35	7,64842E-09
SOR ($\omega = 2$)	4,45497E-11	303,35	8,25E+03	5,01E+03	1,40E+03	36	8,65369E-09
Min Dimension Direct	3,14193E-13	346,39	3,38E+02	5,01E+03	N/A	N/A	N/A
LU Factorization	3,57048E-07	345,27	3,71E+00	5,01E+03	9,48E+01	N/A	N/A
CHOLESKY Factorization	3,26184E-13	346,06	1,72E+01	5,01E+03	4,89E+01	N/A	N/A
CONJUGATE GRADIENT	2,95541E-13	346,92	2,26E+02	5,01E+03	N/A	1	1,19240E-15

Na Tabela 5.5 apresentam-se os resultados da reconstrução do sinal com 768 amostras perdidas. O valor do raio espectral da matriz S é 0.75 e o número de condição da matriz $(I - S)$ é 4. Através da análise destes dois parâmetros, verifica-se que, mais uma vez, poderão ser usados todos os métodos disponíveis no SPIEW para obter a solução numérica para o problema em estudo.

Tabela 5.5 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump =4), 768 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	2,09560E-07	229,90	1,90E+04	6,89E+03	N/A	80	7,54187E-09
Min Dimension Iterativo	2,09561E-07	229,90	1,27E+04	6,89E+03	N/A	79	7,53624E-09
Jacobi Method	1,66027E-07	231,93	1,28E+04	6,89E+03	1,48E+01	69	7,17577E-09
Gauss Seidel Method	3,49628E-10	285,46	1,56E+04	6,89E+03	1,50E+03	99	8,66217E-09
SOR ($\omega = 2$)	4,75478E-10	282,79	1,89E+04	6,89E+03	2,93E+03	121	9,08966E-09
Min Dimension Direct	3,37508E-13	345,76	1,57E+02	6,89E+03	N/A	N/A	N/A
LU Factorization	2,62901E-13	347,93	6,03E+00	6,89E+03	2,87E+02	N/A	N/A
CHOLESKY Factorization	3,68150E-13	345,01	7,73E+00	6,89E+03	1,51E+02	N/A	N/A
CONJUGATE GRADIENT	4,61187E-13	343,05	2,23E+01	6,89E+03	N/A	1	3,31130E-15

Na Tabela 5.6 apresentam-se os resultados da reconstrução do sinal com 896 amostras perdidas. O valor do raio espectral da matriz S é 1 e o número de condição da matriz $(I - S)$ é $4,084583e+015$. Através da análise destes dois parâmetros, verifica-se que a utilização dos ambos os métodos, iterativos e diretos, não é recomendada, isto porque a convergências dos métodos não está garantida.

Tabela 5.6 – Resultados da reconstrução, linha 500, decimação entrelaçada (jump = 8), 896 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	7,45890E-07	218,88	2,51E+04	8,34E+03	N/A	164	9,86103E-09
Min Dimension Iterativo	7,45890E-07	218,88	2,77E+04	8,34E+03	N/A	163	9,86125E-09
Jacobi Method	6,28635E-07	220,36	3,03E+04	8,34E+03	2,28E+01	143	9,70886E-09
Gauss Seidel Method	3,93438E-10	284,43	4,08E+04	8,34E+03	2,02E+03	175	9,24038E-09
SOR ($\omega = 2$)	1,93680E+02	50,59	3,09E+04	8,34E+03	1,12E+04	200	2,69563E+02
Min Dimension Direct	1,50990E-13	352,75	7,70E+02	8,34E+03	N/A	N/A	N/A
LU Factorization	3,33067E-13	345,88	3,05E+01	8,34E+03	4,15E+02	N/A	N/A
CHOLESKY Factorization	*	*	*	8,34E+03	1,83E+02	N/A	N/A
CONJUGATE GRADIENT	8,96616E-13	337,28	1,31E+01	9,50E+03	N/A	2	7,10740E-15

* Matriz $(I-S)$ não é positiva definida.

5.8.3 Resultados da Reconstrução – Decimação Aleatória

Nesta secção são apresentados os resultados da reconstrução do sinal correspondente à linha 500 da imagem selecionada. Foram igualmente testados todos os métodos existentes no SPIEW. As condições iniciais e de condicionamento foram as seguintes:

- Ficheiro: *depth-cam0-f000.bmp*, Linha 500;
- $r=12,5\%$, Largura do FPB 64 amostras;
- Número máximo de Iterações = 600; Tolerância máxima = $10e-8$;
- Decimação aleatória – 250, 850 e 900 amostras perdidas;
- Condição de convergência verificada exceto para 900 amostras perdidas;
- $TU = 8,878851e-004$.

Na Tabela 5.7 apresentam-se os resultados da reconstrução do sinal com 250 amostras perdidas. O valor do raio espectral da matriz S é $6,777263e-001$ e o número de condição da matriz $(I - S)$ é 3,102953. Através da análise destes dois parâmetros, verifica-se que poderão ser usados todos os métodos disponíveis no SPIEW para obter a solução numérica para o problema em estudo.

Tabela 5.7 – Resultados da reconstrução, linha 500, decimação aleatória, 250 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	4,76495E-02	122,77	4,69E+03	7,03E+02	N/A	57	7,08641E-09
Min Dimension Iterativo	4,59461E-09	263,08	4,29E+03	7,03E+02	N/A	56	9,94535E-09
Jacobi Method	3,18334E-09	266,27	3,78E+03	7,03E+02	1,60E-01	49	8,46389E-09
Gauss Seidel Method	8,04065E-10	278,22	2,04E+03	7,03E+02	7,47E+01	27	7,24147E-09
SOR ($\omega = 1,1264$)	3,78790E-11	304,76	1,61E+03	7,03E+02	3,93E+01	21	2,89255E-09
Min Dimension Direct	6,66134E-14	359,86	6,06E+01	7,03E+02	N/A	N/A	N/A
LU Factorization	6,66134E-14	359,86	1,65E-01	7,03E+02	1,49E+00	N/A	N/A
CHOLESKY Factorization	6,61693E-14	359,92	6,52E-01	7,03E+02	7,70E-01	N/A	N/A
CONJUGATE GRADIENT	2,69366E-07	227,72	1,07E+02	7,03E+02	N/A	13	2,69260E-09

Na Tabela 5.8 apresentam-se os resultados da reconstrução do sinal com 850 amostras perdidas. O valor do raio espectral da matriz S é 1 e o número de condição da matriz $(I - S)$ é $1,204401e+009$. Através da análise destes dois parâmetros, verifica-se que o uso dos métodos iterativos não é recomendável, uma vez que o condicionamento do problema não garante a sua convergência. Já os métodos diretos podem ser utilizados para obter a solução numérica do problema em estudo.

Tabela 5.8 – Resultados da reconstrução, linha 500, decimação aleatória, 850 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	3,22968E+02	46,15	4,50E+04	2,29E+03	N/A	600	6,14539E-02
Min Dimension Iterativo	3,24208E+02	46,11	4,87E+04	2,29E+03	N/A	600	7,72983E-02
Jacobi Method	3,19201E+02	46,25	4,54E+04	2,29E+03	5,41E+00	600	7,19024E-02
Gauss Seidel Method	2,85066E+02	47,23	4,54E+04	2,29E+03	2,76E+02	600	5,97342E-02
SOR ($\omega = 1,9999$)	2,75950E+02	47,51	4,67E+04	2,29E+03	7,04E+02	600	3,17427E+02
Min Dimension Direct	2,78294E-06	207,44	3,15E+01	2,29E+03	N/A	N/A	N/A
LU Factorization	1,17905E-06	214,9	4,80E+00	2,29E+03	3,38E+01	N/A	N/A
CHOLESKY Factorization	2,83773E-06	207,27	1,94E+00	2,29E+03	2,15E+01	N/A	N/A
CONJUGATE GRADIENT	3,20332E+02	46,22	2,25E+01	2,29E+03	N/A	N/A	N/A

Finalmente, na Tabela 5.9 apresentam-se os resultados da reconstrução do sinal com 900 amostras perdidas. O valor do raio espectral da matriz S é 1 e o número de condição da matriz $(I - S)$ é $5,673685e+016$. Importa também referir que a condição de convergência da equação (3.8) não se verifica. Através da análise destes parâmetros, verifica-se que a utilização de todos os métodos disponíveis não é aconselhável. Sendo assim, e nestas condições, não é garantida a obtenção de uma solução numérica aceitável para o problema em estudo.

Tabela 5.9 – Resultados da reconstrução, linha 500, decimação aleatória, 900 amostras perdidas.

Algoritmo	Resultados Reconstrução			Tempo Calculo Matrizes		Processo Iterativo	
	RMSE	PSNR (dB)	Tempo (TU)	S (TU)	Outras (TU)	Num Iterações	Erro Residual
Papoulis-Gerchberg	5,69244E+02	41,22	4,89E+04	2,60E+03	N/A	600	9,10121E-02
Min Dimension Iterativo	6,20565E+02	40,47	4,81E+04	2,60E+03	N/A	600	1,07621E-01
Jacobi Method	6,11744E+02	40,6	4,58E+04	2,60E+03	6,89E+00	600	1,15563E-01
Gauss Seidel Method	5,40677E+02	41,67	4,66E+04	2,60E+03	3,42E+02	600	1,41852E-01
SOR ($\omega = 2$) oscila	2,82007E+02	47,32	5,07E+04	2,60E+03	2,26E+03	600	2,95442E+02
Min Dimension Direct	3,36411E+02	45,79	7,76E+01	2,60E+03	N/A	N/A	N/A
LU Factorization	3,38458E+02	45,74	8,34E+00	2,60E+03	3,99E+01	N/A	N/A
CHOLESKY Factorization	*	*	*	2,60E+03	*	N/A	N/A
CONJUGATE GRADIENT	6,10952E+02**	40,61**	1,78E+01**	2,60E+03	N/A	N/A	N/A

* Matriz (I-S) não é positiva definida. ** Atingiu o máximo de iterações mas não convergiu.

Capítulo 6 – Avaliação subjetiva do modelo de aprendizagem: testes MOS

Neste capítulo são apresentados os testes *Mean Opinion Score* (MOS) efetuados para comprovar a utilidade do protótipo aplicativo desenvolvido [Minium93] [Cohen00]. Com os resultados obtidos pretendemos certificar a utilidade do SPIEW como ferramenta educacional de interpolação e processamento de sinal.

Como já foi referido anteriormente, o protótipo aplicativo desenvolvido tem por base o problema de reconstrução de sinal, ainda hoje um problema atual em engenharia, e alguns dos tópicos inerentes a este conceito. O objetivo é que os seus utilizadores possam aplicar diferentes tipos de corrupção de sinal, com sinais provenientes de uma imagem ou de uma sequência de *frames*, e após análise do condicionamento efetuado, possam escolher o método de reconstrução mais adequado à reconstrução dos sinais em estudo.

A utilização do SPIEW fornece uma experiência concreta e promove uma observação refletiva e uma conceptualização abstrata, isto tudo através de experimentação ativa.

6.1 METODOLOGIA UTILIZADA NA REALIZAÇÃO DOS TESTES MOS

A metodologia utilizada para realizar os testes foi a seguinte: realização de uma pequena introdução teórica ao tema; apresentação do protótipo aplicativo desenvolvido; realização de testes com exemplos fornecidos no tutorial; realização de um pequeno teste para avaliar os conhecimentos adquiridos com a utilização do SPIEW.

O tutorial (Anexo A) foi desenvolvido de modo a que os alunos possam executar pela ordem correta todas as operações disponíveis no SPIEW. Este tutorial serve de apoio principalmente, quando lhes é pedido que testem o SPIEW individualmente, quer seja em sala de aula ou mesmo em casa. Embora a grande maioria dos testes tenham sido efetuados em sala de aula, foi sempre fornecido aos alunos uma cópia executável do SPIEW e o respetivo tutorial, o que lhes permitiu testar o protótipo em casa sem problemas. Dúvidas pontuais, maioritariamente relativas à parte de condicionamento dos sinais,

e que surgiram quando os alunos testaram o protótipo em casa foram esclarecidas facilmente por correio eletrónico.

Depois de introduzido o tema da reconstrução de sinal, através da exposição teórica, faz-se a apresentação do SPIEW. Aqui os alunos já têm o protótipo a funcionar nos seus computadores. A instalação é efetuada antes da aula. O objetivo é que os alunos sigam *in loco* todas as operações realizadas. Neste primeiro contacto com o SPIEW são apresentados casos específicos, para que os alunos possam ver o que acontece em diferentes situações de convergência. Ao longo dos tutorial é igualmente explicado o que acontece quando determinadas condições não estão garantidas e o que fazer nestas situações. São igualmente sugeridos alguns testes com valores de parâmetros diferentes, para que os utilizadores possam compreender melhor qual a influência desses parâmetros num problema de reconstrução de sinal. Seguidamente, e durante aproximadamente trinta minutos, os alunos testam livremente o SPIEW. Previamente, é-lhes fornecido um inquérito ao qual devem responder até ao final da aula. É através desse inquérito que são avaliados os conhecimentos adquiridos, e identificadas as áreas onde os alunos demonstram mais dificuldade.

Os primeiros testes efetuados ao SPIEW, como ferramenta educacional, foram realizados com docentes do ensino superior, na área da engenharia provenientes de duas instituições de ensino superior distintas, o Instituto Politécnico de Castelo Branco (IPCB) e o Instituto Politécnico de Leiria (IPL). O *feedback* obtido permitiu efetuar algumas alterações antes de a aplicação ter sido testada com alunos.

Posteriormente o SPIEW foi testado com alunos de diversos graus de ensino: licenciatura, mestrado e doutoramento, oriundos da Universidade de Trás-os-Montes e Alto Douro (UTAD) e do IPCB.

6.2 RESULTADOS OBTIDOS COM A PRIMEIRA VERSÃO DO SPIEW

Nesta secção apresentam-se os resultados obtidos nos diversos grupos de testes. Como já foi referido anteriormente, o primeiro teste foi realizado com docentes do ensino superior. O inquérito efetuado pode ser consultado no Anexo B (primeira versão).

Foram inquiridos um total de 11 docentes. O questionário possui 18 questões: 16 questões de resposta, Verdadeiro e Falso; uma questão para avaliação qualitativa do SPIEW; 1 pergunta de resposta livre, para avaliar se o inquirido considera o SPIEW uma mais-valia no processo de aprendizagem.

No gráfico seguinte apresentam-se os resultados obtidos no grupo das questões com resposta Verdadeiro e Falso, Figura 6.1.

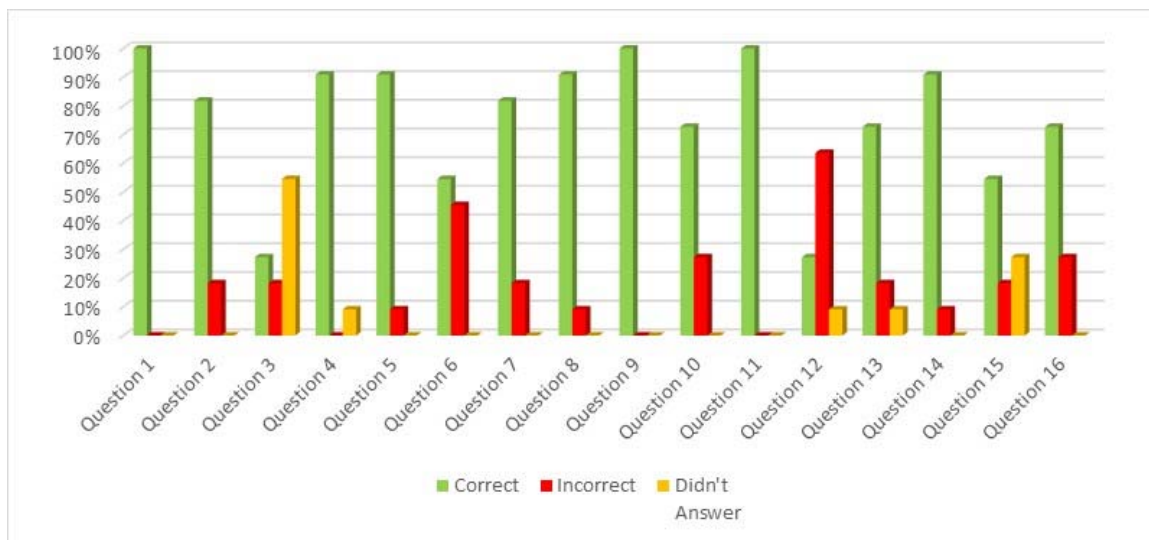


Figura 6.1 – SPIEW - Resultados Obtidos nas questões de Verdadeiro e Falso. Teste com Docentes.

Relativamente à questão 17, onde era pedido para caracterizar a aplicação, 81.82% dos inquiridos classifica o SPIEW como “Boa” e 18.18% classificou a aplicação como “Muito Boa”, considerando a escala: Má, Razoável, Boa e Muito Boa.

Quando questionados se consideravam o SPIEW uma mais-valia no processo de aprendizagem, 90.91% dos inquiridos responderam afirmativamente. No entanto, apontaram alguns aspetos de melhoria, tais como, melhorar a interface gráfica para ser mais intuitiva e *user friendly* e implementar o *Help* em diversos tópicos. Seguindo a abordagem prédefinida para o desenvolvimento do SPIEW, centrada no *feedback* do utilizador, optou-se por reformular a aplicação e desenvolver uma segunda versão do SPIEW, anteriormente apresentado no capítulo 5.

6.3 RESULTADOS OBTIDOS COM A SEGUNDA VERSÃO DO SPIEW

Nesta secção apresentam-se os resultados obtidos com a nova versão do protótipo SPIEW.

Os primeiros testes foram efetuados com um grupo de alunos da UTAD. A metodologia utilizada foi a descrita na secção 6.1. O inquérito consiste em 6 questões de resposta livre para avaliar os conhecimentos adquiridos, e uma última questão para que os inquiridos possam avaliar o protótipo, SPIEW. O inquérito realizado pode ser consultado no Anexo B (segunda versão). Os alunos inquiridos eram alunos de doutoramento, foram realizados um total de 11 inquéritos.

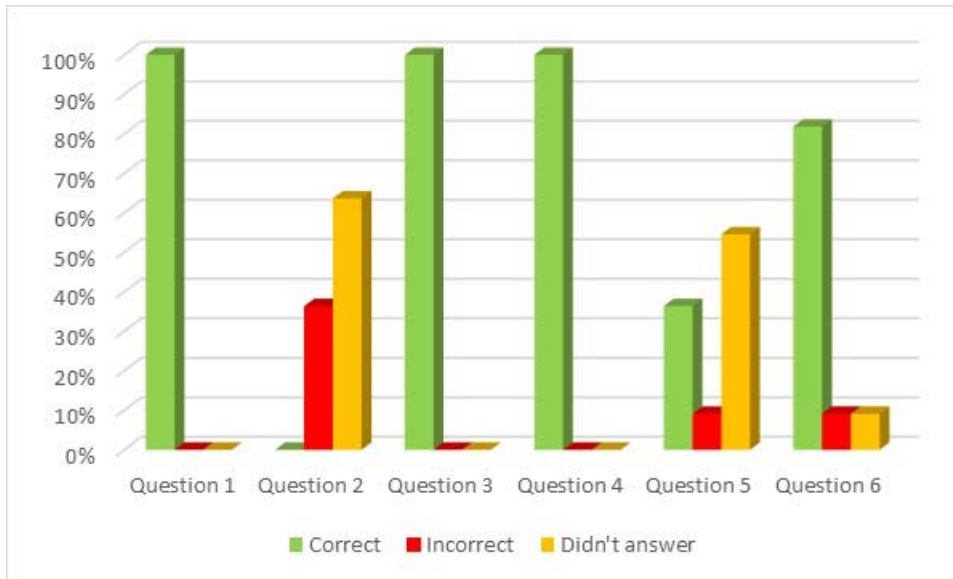


Figura 6.2 – SPIEW - Resultados nas 6 primeiras questões. Alunos UTAD.

Relativamente à questão 7, onde se questionava sobre a utilidade do SPIEW no processo de aprendizagem, 100% dos inquiridos respondeu afirmativamente. Maioritariamente, o aspeto ao qual deram mais ênfase foi a oportunidade de visualizar graficamente o resultado de diversas operações de PDS abordadas no SPIEW. Este fator permitiu uma melhor compreensão de conceitos como filtragem, amostragem, reconstrução, entre outros. Estes resultados foram extremamente encorajadores, uma vez que foram totalmente de encontro aos objetivos previamente estabelecidos para esta tese.

Após obtidos os resultados destes testes, foram acrescentadas novas funcionalidades e novos algoritmos ao protótipo. Na fase seguinte, o utilizador passou a poder analisar o condicionamento do problema, antes de testar os algoritmos de reconstrução. Isto permite aos utilizadores alterar as condições do seu problema em estudo para garantir uma solução para o seu problema.

Os testes que se apresentam a seguir foram realizados por um grupo de 5 alunos de licenciatura da Escola Superior de Tecnologia do IPCB. Importa referir que os conhecimentos destes alunos na área de PDS é limitado.

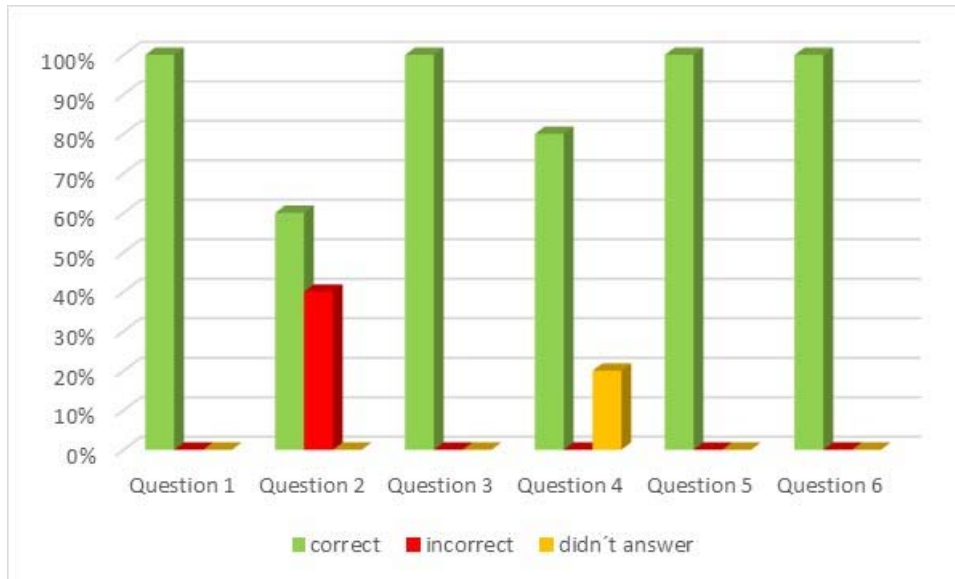


Figura 6.3 – SPIEW - Resultados nas 6 primeiras questões. Alunos EST-IPCB.

Podemos observar que, apesar de não possuírem grandes conhecimentos na área, os alunos responderam corretamente a uma grande percentagem de questões, sendo que na questão 2 mostraram mais dificuldades. A pergunta em questão é: *“Um fator de sobreamostragem baixo corresponde a uma Largura de Banda do Filtro Passa Baixo alta ou baixa?”*

Relativamente à avaliação qualitativa do SPIEW, 100% dos inquiridos respondeu que conseguiram melhorar os seus conhecimentos na área. A aplicação tem os passos todos bem definidos, o que lhes permitiu seguir corretamente a ordem das operações e testar a aplicação sem problemas. Os alunos em causa referiram ainda, que através do uso de imagens, a visualização dos resultados de operações como a decimação ou filtragem lhes permitiu assimilar alguns destes conceitos. O que mais gostaram foi a visualização da reconstrução dos sinais. Demonstraram, no entanto, algumas dificuldades na parte da análise do condicionamento.

Foram ainda realizados dois testes com alunos de doutoramento na UTAD, onde participaram 4 alunos, dois grupos de dois alunos. Estes testes foram realizados no módulo *“Reconstrução de Imagem e Sinal”* lecionado nos cursos de doutoramento em engenharia eletrotécnica e em informática. O objetivo destes testes com utilizadores, com conhecimentos mais elevados da área de PDS, foi obter as suas opiniões relativamente à utilização do SPIEW como ferramenta de apoio à experimentação e simulação de diversos problemas de engenharia. Foi assim dada total liberdade no uso da plataforma e pedida uma resposta ao inquérito que se apresenta no Anexo B (última versão).

Foi com agrado que se obteve a resposta de dois grupos, e que os mesmos executaram e registaram uma série de testes com os diversos algoritmos, o que lhes permitiu aferir sobre o desempenho de

cada um dos algoritmos disponíveis no SPIEW. Os resultados obtidos nestes dois grupos estão apresentados na Figura 6.4.

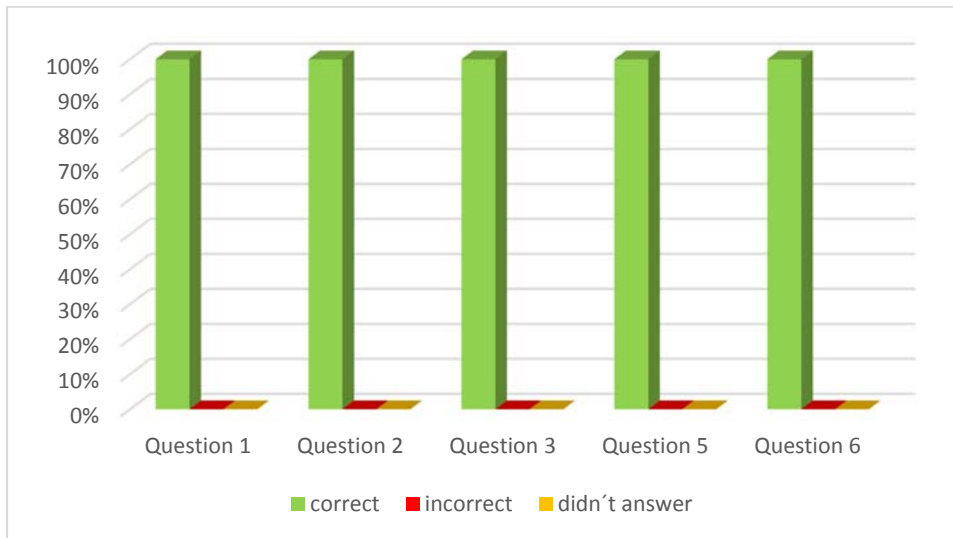


Figura 6.4 – SPIEW – Resultados obtidos no inquérito, excepto na questão 4 e 7. Alunos de doutoramento UTAD.

Na figura anterior podemos observar que os alunos responderam corretamente às questões 1,2,3,5 e 6. Relativamente à questão 7, “*Considera que o uso deste protótipo ajudou-o a adquirir ou a melhorar os seus conhecimentos na área?*”, todos eles responderam afirmativamente salientando o facto de poderem observar o resultado diversas operações gráficamente.

Na questão 4, “*No caso da análise do condicionamento do problema ser favorável em ambos os casos, iterativo e direto, qual o algoritmo de reconstrução que escolheria?*”, o objetivo era que os alunos testassem os diversos algoritmos e analisassem o seu desempenho a nível de: qualidade do resultado e tempo de reconstrução.

Os alunos testaram todos os algoritmos disponíveis no SPIEW utilizando dois tipos de geometria de padrão de erros, a aleatória (200 amostras perdidas) e a entrelaçada (com um *jump* de 4). Chegaram a conclusão que os métodos diretos são mais rápidos a nível de tempo de execução, mas indicam que o resultado final tem um PSNR mais baixo, o que na prática não se verifica pela análise dos valores obtidos. De todos os métodos diretos, escolheram o Método de Cholesky, porque afirmam ter um tempo de reconstrução mais baixo quando o número de amostras perdidas é maior.

No caso dos métodos iterativos, optaram pelo Método de Jacobi, esta escolha foi igualmente efetuada pela análise do tempo de reconstrução. No entanto, pela análise dos resultados obtidos, o número de iterações necessárias, para obter a solução, com os Métodos de Gauss Seidel e SOR é bastante inferior, fator que não foi tido em conta. Os alunos concluíram que a escolha do algoritmo depende do caso onde é aplicado, podendo escolher algoritmos distintos quando se pretende velocidade de execução

ou qualidade de reconstrução mais elevada. Destas respostas concluí-se que os alunos assimilaram corretamente a maioria dos conceitos abordados, no entanto, demonstram algumas dificuldades na análise de todos os parâmetros, dando especial importância ao tempo de execução, mas analisando erradamente os parâmetros da qualidade (PSNR e RMSE).

Finalmente, foi realizado um último teste com um grupo de 5 alunos do 1º ano do curso de Engenharia Informática do IPCB. Importa referir que os seus conhecimentos na área do PDS são limitados. No entanto, a nível de álgebra e métodos numéricos os mesmos demonstraram conhecimentos razoáveis. A metodologia seguida foi a descrita na secção 6.1, e à semelhança do que aconteceu no teste anterior foi dada total liberdade no uso do protótipo. Foi igualmente pedida a resposta ao inquérito que se apresenta no Anexo B (última versão). Os resultados obtidos neste grupo estão apresentados na Figura 6.5.

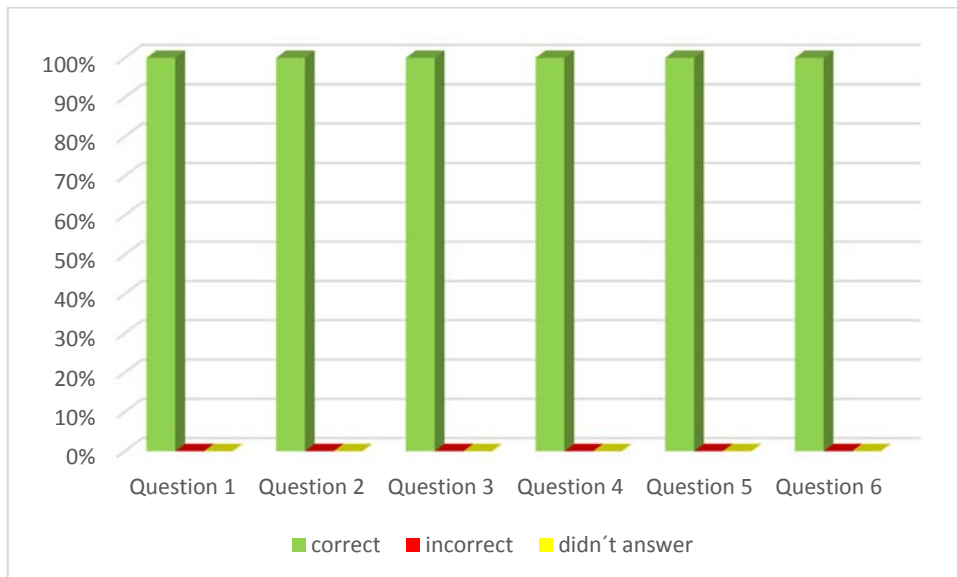


Figura 6.5 – SPIEW – Resultados obtidos no inquérito, excepto na questão 7. Alunos de Engenharia Informática IPCB.

Os alunos responderam acertadamente a todas as questões, no entanto, algumas justificações são incoerentes com as respostas dadas. Mais concretamente as justificações nas questões 1 e 2. Nas respostas obtidas à questão 4, todos os inquiridos optaram por escolher os métodos de resolução direta (o método de Cholesky foi o mais escolhido). A justificação escolhida foi o facto de serem mais rápidos que os iterativos. No entanto, a qualidade dos sinais reconstruídos foi também um fator tido em conta, uma vez que referem ter escolhido o método com o qual obtiveram um PSNR mais elevado.

Relativamente à questão 7, “Considera que o uso deste protótipo ajudou-o a adquirir ou a melhorar os seus conhecimentos na área?”, o *feedback* dos alunos inquiridos foi o seguinte:

- “Sim. Visto não ter nenhum conhecimento sobre esta área antes destas experiências aprendi muito em relação à reconstrução de imagens.”
- “Bastante, porque usamos métodos matemáticos lecionados nos cursos científicos num problema de engenharia.”
- “Sim, e acho que me dá uma melhor ideia dos conteúdos aparentemente inúteis que se aprendem nas escolas, neste caso, as matrizes.”
- “Sim, permite relacionar métodos matemáticos lecionados nos cursos científicos num problema de engenharia.”
- “Sim consegui ver na prática e utilidade de alguns métodos matemáticos.”

Em relação ao protótipo este grupo de alunos referiu que gostaria de ver as imagens com os pixels todos reconstruídos e não apenas um dos sinais escolhidos.

Capítulo 7 – Conclusões e Trabalho Futuro

O modelo de ensino de algoritmos de reconstrução de sinal apresentado nesta tese revelou ser uma mais-valia uma vez que contribui para o melhoramento da aprendizagem dos alunos. Para além disso, acreditamos ter contribuído para desmistificar, em parte, alguns tópicos da área do processamento de sinal, que normalmente despertam nos alunos pouco interesse. Foi muito interessante e enriquecedor sentir o entusiasmo de alguns participantes na observação e acompanhamento da execução dos referidos algoritmos.

Através da utilização do SPIEW, protótipo aplicacional desenvolvido para ser utilizado neste novo modelo de ensino, acrescentamos uma componente de simulação que, através da visualização gráfica de resultados, permite aos seus utilizadores consolidar ou adquirir conhecimentos na área de algoritmos de reconstrução de sinal. O protótipo aplicacional desenvolvido fornece aos seus utilizadores uma experiência concreta de um problema real e atual da engenharia. Este permite a simulação de cenários reais de problemas de reconstrução, dando aos alunos/utilizadores a possibilidade de testar diferentes métodos de reconstrução e aferir sobre a eficácia dos mesmos relativamente a tempo de execução e avaliação dos resultados obtidos. A possibilidade de interação com o utilizador demonstrou efetivamente ser uma mais-valia na aprendizagem. O SPIEW permite testar diversos algoritmos aplicando-os a sinais reais, recorrendo à visualização gráfica de resultados, tendo sempre em conta os condicionalismos que são normalmente abordados de forma teórica e raramente têm oportunidade de testar o que acontece quando determinados parâmetros não possuem um valor óptimo. Estas funcionalidades permitem assim aos utilizadores do SPIEW observar, conceptualizar e experimentar novamente, aumentando a facilidade de aprendizagem dos conceitos.

Realça-se a particularidade do protótipo permitir aos utilizadores alterar o contexto do problema, potenciando uma aprendizagem assistida sobre os condicionalismos da convergência dos diversos métodos e algoritmos disponíveis no SPIEW, que são conceitos de natureza matemática e, normalmente, é onde os alunos apresentam grande resistência na sua aprendizagem.

Resumindo, os alunos aprendem de uma forma mais célere e intuitiva a concretização prática dos conceitos teóricos lecionados.

Podemos concluir que foram cumpridos os principais objetivos e prestados os contributos estabelecidos para este trabalho. As conclusões que se apresentam a seguir são suportadas pelos resultados obtidos com os testes MOS realizados:

- Foi desenhado um modelo para facilitar e promover o estudo dos algoritmos de reconstrução de sinal.
- Foi implementado o protótipo aplicacional (SPIEW) para experimentação ativa de conceitos e algoritmos, previamente apresentados de forma expositiva, que se demonstrou ser uma plataforma muito útil para apoio à compreensão dos algoritmos implementados.
- A aprendizagem por observação e *debug*/comportamento dos algoritmos demonstrou-se também muito eficaz na ajuda à compreensão dos algoritmos e respetivos condicionalismos.
- A simulação de cenários de realizações em engenharia foi igualmente testada, utilizando cenários reais, tendo-se demonstrado muito útil com utilizadores mais avançados, como foi o caso dos alunos de doutoramento.
- Foi claramente conseguida uma contribuição para a estimulação os alunos de engenharia, para o estudo do processamento digital de sinal, nomeadamente, no caso da reconstrução de sinais. Utilizando a interface gráfica de visualização da execução destes algoritmos, foi possível tornar mais clara a sua aplicabilidade e os resultados produzidos.
- Com o modelo proposto e com o recurso ao SPIEW ficou demonstrado que é possível aumentar as competências dos alunos de engenharia nesta área, captando a sua atenção e o seu interesse de uma forma natural.

Futuramente, pretende-se realizar novos testes com alunos de áreas de conhecimento distintas, de modo a obter mais contributos que permitam melhorar o modelo de ensino proposto assim como, o protótipo aplicacional desenvolvido. Os testes serão realizados com dois grupos, um com acesso ao SPIEW e outro sem acesso ao SPIEW de modo a melhor comprovar a eficácia do mesmo. Relativamente a este último, pretendemos que no futuro o SPIEW possa trabalhar não apenas com ficheiros de imagem ou sequências de imagens, mas também com ficheiros áudio. A integração de novos algoritmos no protótipo é igualmente um objetivo. A parte que permite analisar e classificar o desempenho dos diferentes algoritmos abordados será igualmente melhorada, de modo a permitir comparações quer a nível de qualidade do resultado final, quer a nível de velocidade de desempenho dos mesmos.

Conforme sugerido nos testes MOS, será igualmente dada a possibilidade de os alunos visualizarem as imagens após a reconstrução.

Referências

- [Abdulwahed09] Mahmoud Abdulwahed, Zoltan K. Nagy: “*Applying Kolb’s Experiential Learning Cycle for Laboratory Education*”, *Journal of Engineering Education*, Volume 98, Issue 3, Article first published online: 2 JAN 2013
- [Ambardar97] Ambardar, A., Borghesani, G., “*Mastering Dsp Concepts Using Matlab*”, Prentice Hall, 1997. ISBN: 978-0135349762.
- [Araújo13] Adérito Luís Martins Araújo, “*Matemática Computacional*”, Notas de apoio às aulas de Matemática Computacional do Mestrado Integrado em Engenharia Electrotécnica e de Computadores, no ano lectivo de 2013/2014. FCTUC, 2013. URL: <http://www.mat.uc.pt/~alma/publicat/coursenotes/MatematicaComputacional.pdf>
- [Barret94] Barret, R., M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine e H. Van der Vost. “*Templates for the Solutions of Linear Systems: Building Blocks for Iterative Methods.*” SIAM. Philadelphia, 1994. (<http://www.netlib.org/templates/Templates.html>).
- [Belhot01] Renato V. Belhot, Reginaldo S. Figueiredo, Cesar O. Malavé: “*O Uso da Simulação no Ensino da Engenharia*”, 2001.
- [Carvalho01] Anna Carvalho, Arthur Porto, Renato Belhot. “*Aprendizagem Significativa no Ensino de Engenharia*”, *Revista PRODUÇÃO*, v.11 n.1, Novembro de 2001
- [Caulfield11] Caulfield, C., Xia, J., Veal, D., & Maj, S. P.: “*A Systematic Survey of Games Used for Software Engineering Education*”. *Modern Applied Science*, 5(6), 28-43. <http://dx.doi.org/10.5539/mas.v5n6p28>

- [Cohen00] Cohen, L, Manion, L, & Morrison, K. "Research Methods in Education". Londres: RoutledgeFalmer, 2000.
- [Cooley65] Cooley, J. W., John W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Computation, 19: 297-301, 1965.
- [Costa09] M. I. COSTA, J. BARROSO, S. SOARES: "Signal Processing Interpolation Educational Workbench", 2009 Wiley Periodicals, Inc. Comput Appl Eng Educ 20: 356-363, 2012; DOI 10.1002/cae.20402
- [Costa13a] Mónica Isabel T. da Costa, Simão Cardeal, Salviano Soares, João Barroso: "Problem Conditioning Interpolation Educational Tool", IEEE Global Engineering Education Conference (EDUCON), Berlin, Germany, 2013. ISBN: 978-1-4673-6109-5/13. pp:996-1003.
- [Costa13b] Mónica Isabel T. da Costa, Salviano Soares, João Barroso: "Signal Processing Interpolation and Problem Conditioning Educational Workbench", Modern Applied Science; Vol. 7, No. 12; 2013, Online Published: November 18, 2013 doi:10.5539/mas.v7n12p56
- [Ferreira94a] Ferreira, P. J. S. G., "Interpolation and the Discrete Papoulis-Gerchberg Algorithm", IEEE Transactions on Signal Processing, 42(10):2596-2606, 1994. (DOI: 10.1109/78.324726).
- [Ferreira94b] Ferreira, P. J. S. G., "Non Iterative and Fast Iterative Methods for Interpolation and Extrapolation", IEEE Transactions on Signal Processing, 42(11): 3278-3282, 1994.
- [Ferreira94c] Ferreira, P. J. S. G., "The Stability of a Procedure for the Recovery of Lost Samples in Band-Limited Signals", Signal Processing, 40(3): 195-205, 1994.
- [Ferreira96] Ferreira, P. J. S. G., "Interpolation in the Time and Frequency Domains." IEEE Signal Processing Letters, 3(6):176-178, 1996.

- [Ferreira99] Ferreira, P. J. S. G., *"Mathematics for multimedia signal processing II - discrete finite frames and signal reconstruction."* Signal Processing for Multimedia, pages 35-54. IOS Press, J. S. Byrnes, Ed. 1999.
- [Golub96] Golub, G., & Van Loan, C., *"Matrix computations"* (3rd ed.), 1996. The Johns Hopkins Univ Press.
- [Hestenes52] Hestenes, M. e E. Stiefel. *"Methods of Conjugate Gradients for Solving Linear Systems."* Journal of Research of the National Bureau of Standards, 49(6):409-436. 1952.
- [Hevner04] A. R. Hevner, S. T. March, J. Park, and S. R., *"Design science in information systems research"*, MIS Quarterly, 75–105, 2004.
- [Karim09] H. Abdul Karim, A. Sali, S. Worrall, Abdul H. Sadka, A. M. Kondo: *"Multiple description video coding for stereoscopic 3D"*, IEEE Transactions on Consumer Electronics, Vol. 55, No. 4, pp. 2048–2056, November 2009.
- [Kincaid02] Kincaid, D. e W. Cheney. *"Numerical Analysis: Mathematics of Scientific Computing"*. The Brooks/Cole Series in Advanced Mathematics, Paul J. Sally, Jr., Editor, SIAM, USA. 2002.
- [Kolb84] *"Experiential Learning: Experience as the source of learning and development"*, Englewood Cliffs, NJ: Prentice-Hall.
- [Konak13] Abdullah Konak, Tricia K. Clark, Mahdi Nasereddin: *"Using Kolb's Experiential Learning Cycle to Improve student learning in virtual computer laboratories"*, Journal of Engineering Education, Volume 98, Issue 3, pages 283–294, July 2009
- [Liu12] Zhi Liu, G. Cheung, J. Chakareski, and Yusheng Ji: *"Multiple description coding of free viewpoint video for multi-path network streaming"*, in Global Communications Conference (GLOBECOM), 2012 IEEE, Dec 2012, pp. 2150–2155.

- [Marín05] Sergio L. Toral Marín, Federico J. Barrero García, Rocío Martínez Torres, Sergio Gallardo Vázquez, and Antonio Javier Lillo Moreno: *“Implementation of a Web-Based Educational Tool for Digital Signal Processing Teaching Using the Technological Acceptance Mode”*, IEEE TRANSACTIONS ON EDUCATION, VOL. 48, NO. 4, NOVEMBER 2005, pp: 632-641.
- [Miguel03] António Miguel, *“Gestão de Projectos de Software”*, FCA – Editora de Informática, 2003. ISBN: 972-722-352-4
- [Minium93] Minium, E. W., King, B. M., & Bear, G. *“Statistical Reasoning in Psychology and Education”*. New York: John Wiley & Sons, Inc., 1993.
- [Neves08] Neves, F., Soares, S., Reis, M. C., Tavares, F., Assunção, P., *“VoIP reconstruction under a minimum interpolation algorithm”*, IEEE International Symposium on Consumer Electronics, ISCE 2008. (DOI:10.1109/ISCE.2008.4559434).
- [Norkin06] A. Norkin, A. Gotchev, K. Egiazarian, J. Astola: *“Two stage multiple description image coders: Analysis and comparative study”*, Signal Processing: Image Communications, vol. 11, pp. 609–625, 2006.
- [Papoulis75] Papoulis, A., *“A New Algorithm in Spectral Analysis and Band-Limited Extrapolation”*, IEEE Transactions on Circuits and Systems, 22(9):735-742, 1975. (DOI:10.1109/TCS.1975.1084118)
- [Peffer08] Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S., *“A design science research methodology for information systems research.”* Journal of management information systems, 24(3), 45-77, 2008.
- [Pierre99] J.W. Pierre, R.F. Kubichek and J.C. Hamann, *“Reinforcing the Understanding of Signal Processing Concepts Using Audio Exercises”*, Acoustics, Speech, and Signal Processing, 1999. Proceedings, 1999 IEEE International Conference on (Volume:6), pp: 3577 – 3580. (DOI: 10.1109/ICASSP.1999.757616)

- [Pina10] Heitor Pina, "Métodos Numéricos", Escolar Editora, 2010. ISBN: 9789725922842.
- [Proakis99] Proakis, J. G. e Ingle, V.K., "Digital Signal Processing Using MATLAB", 1999. CL-Engineering. ISBN: 978-0534371746.
- [Proakis12] John G. Proakis, Masoud Salehi, Gerhard Bauch, "Contemporary Communication Systems Using MATLAB", 2012. Cengage Learning. ISBN: 978-0495082514.
- [Reis00] Reis, M. J. C. S. "*Representação multi-resolução de sinais baseados em Splines e sobre-amostragem.*", Tese de Doutoramento, Universidade de Aveiro, 2000.
- [Rodrigues03] José Alberto Rodrigues, "Métodos Numéricos- Introdução, Aplicação e Programação", Edições Sílabo, 2003. ISBN: 972-618-311-1
- [Sanz83] Sanz, Jorge L. C., Thomas S. Huang, "Some Aspects of Band-Limited Signal Extrapolation: Models, Discrete Approximations, and Noise.". IEEE Transactions on Acoustic Speech and Signal Processing, 31(6): 1492-1501, 1983.
- [Shewchuk94] J. R. Shewchuk, "*An Introduction to the Conjugate Gradient Method without the Agonizing Pain*". School of Computer Science, Carnegie University. 1994.
- [Soares03] Salviano Filipe Silva Pinto Soares, "*Algoritmos de Reconstrução de Sinal e Correção de Erros*", Tese de Doutoramento, Universidade de Aveiro, 2003.
- [Sommerville09] Ian Sommerville, "Software Engineering", 9th Edition. Pearson Education, 2009. ISBN: 978-0-13-703515-1.
- [Stice87] J.E. Stice, "Using Kolb's learning cycle to improve student learning", Engineering Education, vol. 77, no. 15, Fev. 1987, pp: 291-296.
- [Tokdemir07] Tokdemir, G., Cagiltay, N.E., "*Using Learning Style Theory in Remote Laboratory Applications*", IEEE 18th International Symposium on Personal, Indoor and Mobile

Radio Communications, 2007. pp: 1-3. ISBN: 978-1-4244-1144-3. DOI: 10.1109/PIMRC.2007.4394815.

[Vaishnavi04] Vaishnavi, V., & Kuechler, B. "Design Science Research in Information Systems". Association for Information Systems, 2004.

[Velte13] C.M. Velte, P. H Andersson, K. E Meyer: "Coupling of theory and practice through inductive learning in experimental fluid mechanics education - A practical study". 41st SEFI Conference, 16-20 September 2013, Leuven, Belgium.

[Vieira97] Vieira, J. M. N. e P. J. S. G. Ferreira. "*Interpolation, spectrum analysis, error-control coding, and fault-tolerant computing.*" Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP 97, (III):1831-1834. 1997.

[Vieira13] José Vieira, "*Bolonha e os Novos Paradigmas para a Formação e Emprego dos Engenheiros*", Ingenium, II série, Número 138, Novembro e Dezembro de 2013.

Anexos

Anexo A – Tutorial do SPIEW

Anexo B – Inquérito SPIEW – Três versões

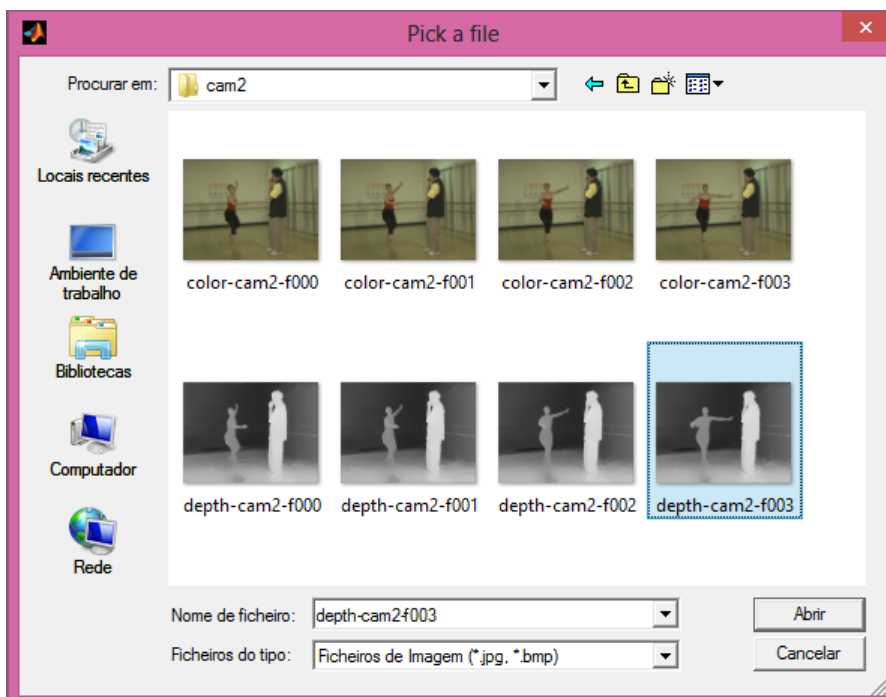
ANEXO A – TUTORIAL DO SPIEW

Tutorial da aplicação SPIEW2.0

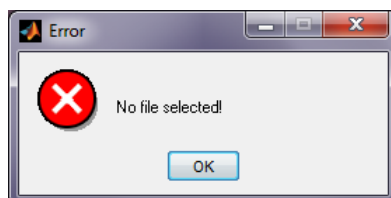
Esta é a janela principal da aplicação.



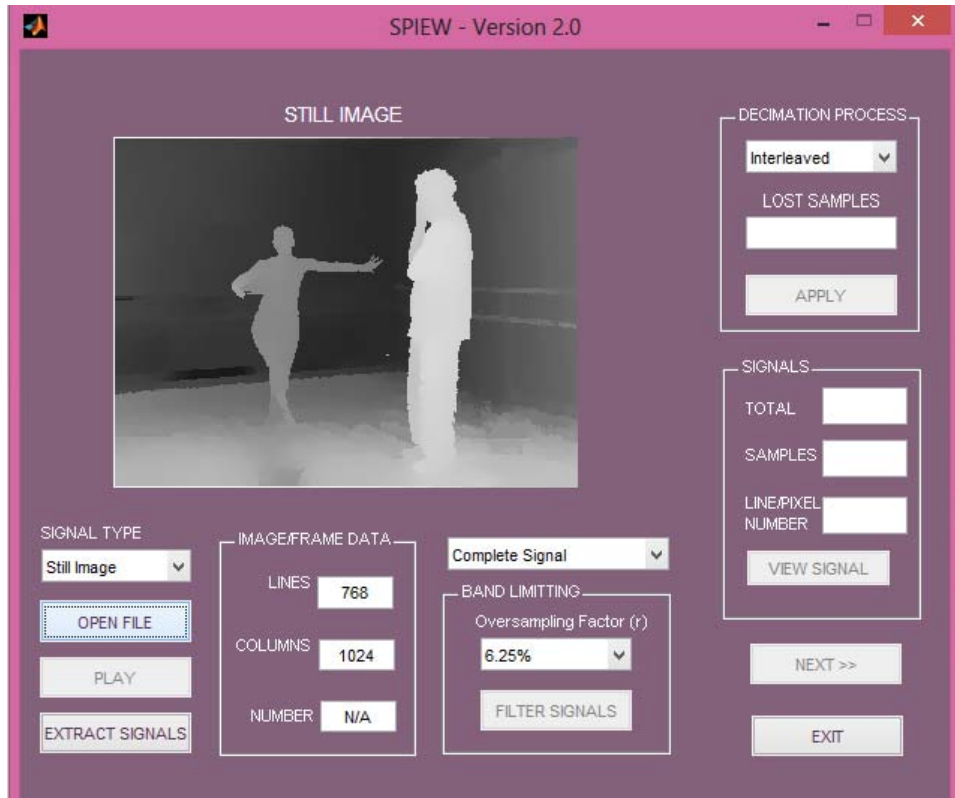
1. No primeiro passo o utilizador tem que escolher o **SIGNAL TYPE**, que poderá ser uma imagem ou sequência de imagens, para posterior extração dos sinais. Os sinais correspondem a cada linha da imagem, no caso do utilizador selecionar uma imagem, ou ao valor de um pixel da imagem ao longo da sequência.
2. Posteriormente o utilizador deve pressionar o botão **OPEN FILE**. Para que o utilizador siga os passos corretos, todos os outros botões estão inativos. Os mesmos serão ativados de acordo com a sequência de utilização correta.



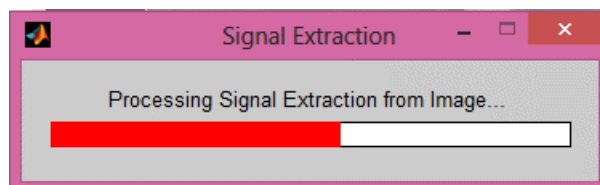
3. No caso de uma sequência de imagens o utilizador deve seleccionar a primeira imagem da sequência. No caso de seleccionar uma imagem deve apenas seleccionar o seu ficheiro. Caso não selecione nenhuma imagem a seguinte caixa de aviso é apresentada.



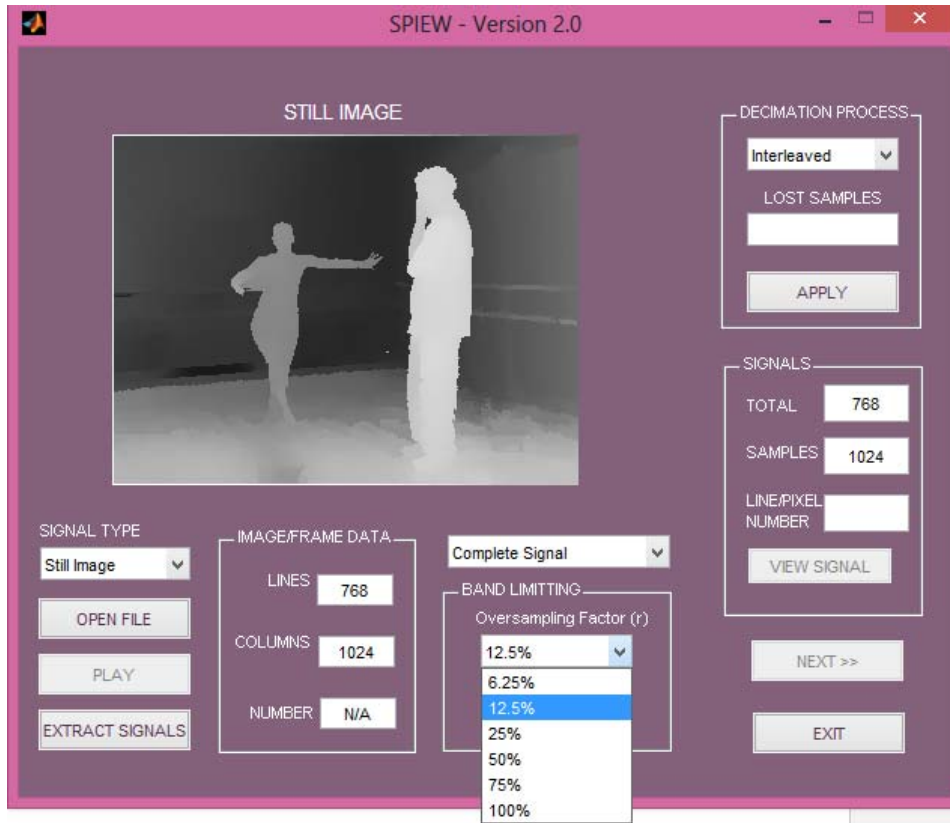
4. Para abrir corretamente a sequência o utilizador deve repetir o processo, e seleccionar a primeira imagem da sequência de frames ou uma imagem.
5. Automaticamente são apresentados o número de linhas e colunas da imagem ou dos frames da sequência escolhida.



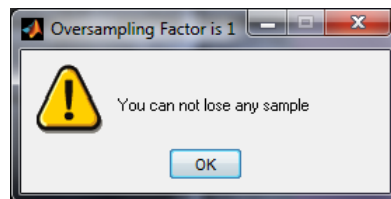
6. No caso de ter sido escolhida uma sequência de frames, a aplicação permite visualizar a sequência, para tal o utilizador têm que pressionar o botão **PLAY SEQUENCE**. Caso tenha sido selecionada uma imagem normal, o botão fica inativo.
7. Um outro botão que fica ativado é o botão **EXTRACT SIGNALS**. O próximo passo consiste em extrair os sinais da imagem/sequência para uma matriz. Para tal o utilizador deve pressionar o botão **EXTRACT SIGNALS**. A barra de progresso desta operação aparece até a mesma ser completada. Na zona com informação relativa aos sinais extraídos já aparece o número total de sinais extraídos e número total de amostras de cada sinal.



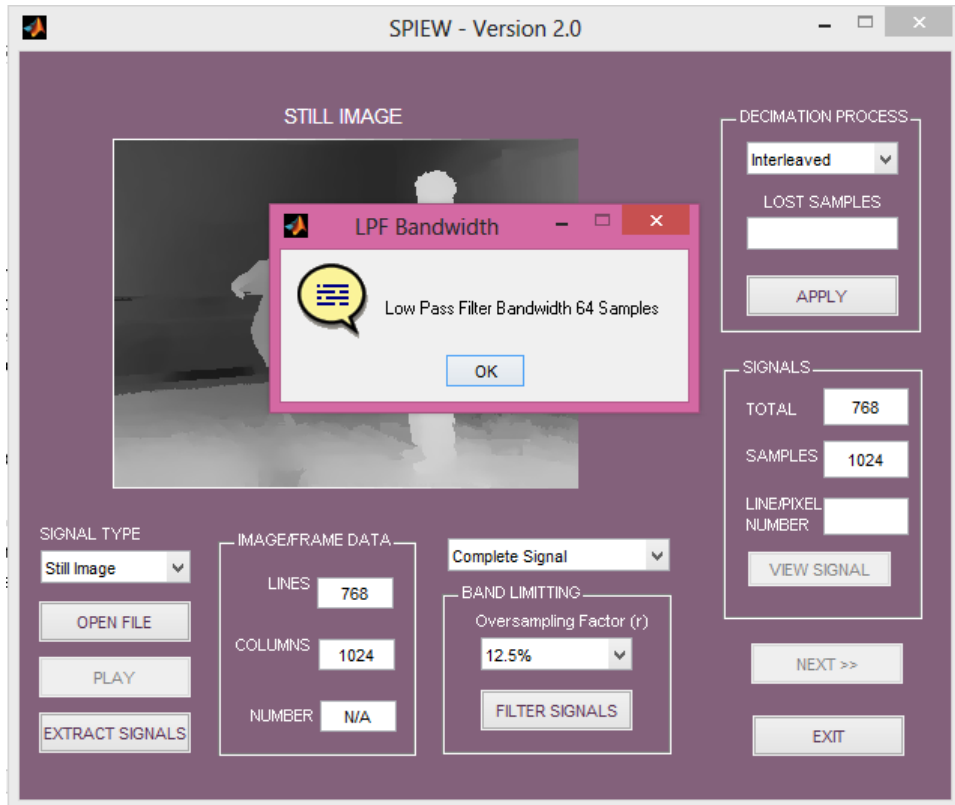
8. Após a extração dos sinais, fica ativado o botão **FILTER SIGNALS**, que será o passo seguinte na aplicação. Estes consiste na limitação em frequência dos sinais extraídos. Nesta caso específico utiliza-se um filtro passa baixo para efetuar esta operação. Para tal o utilizador têm que especificar qual a percentagem de amostras as colocar a zero, ou por outras palavras, estipular a redundância dos sinais, através do fator de subamostragem escolhido. Este valor é apresentado em percentagem. Depois de escolhido este valor o utilizador deve pressionar o botão **FILTER SIGNALS**.



É importante referir que no caso de o utilizador escolher a percentagens de 100%, aparecerá a seguinte mensagens de aviso. Indicando que neste caso não poderá perder nenhuma amostra dos sinais.



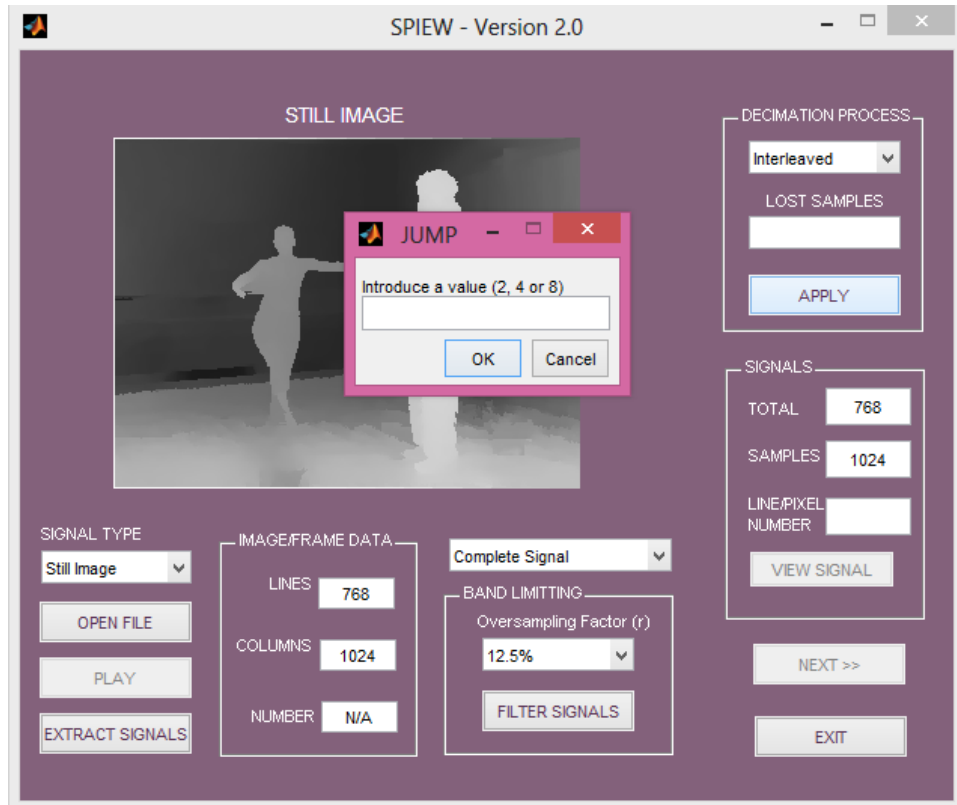
9. Caso o utilizador escolha uma das outras percentagens disponíveis, será efetuada a limitação em frequência com o valor escolhido. Aparece igualmente uma barra a indicar a progressão da filtragem e no final e também a titulo elucidativo é indicada a largura de banda do FPB (Filtro Passa Baixo) correspondente à percentagem de redundância escolhida.



10. Após esta operação fica ativo o botão **APPLY** na zona **DECIMATION PROCESS**. Depois de limitados em frequência o passo seguinte é então o processo de decimação/corrupção, que permitirá corromper os sinais com o padrão de erros escolhido. A aplicação permite três tipos de geometria de padrão de erros. São eles o interleaved (igualmente espaçados), random (aleatória) e o contiguos (rajada).

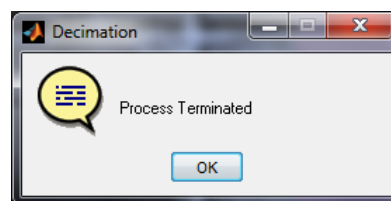


11. Aqui o utilizador deve escolher qual a geometria de padrão de erro que quer aplicar. Se optar pela opção **“Interleaved”**, não é necessário indicar o número de amostras a perder. Pressione o botão **APPLY** e aparecerá uma janela onde o utilizador deve indicar qual o espaçamento a utilizar neste tipo de decimação. Existem 3 opções disponíveis 2, 4 ou 8, nenhum outro valor é permitido.

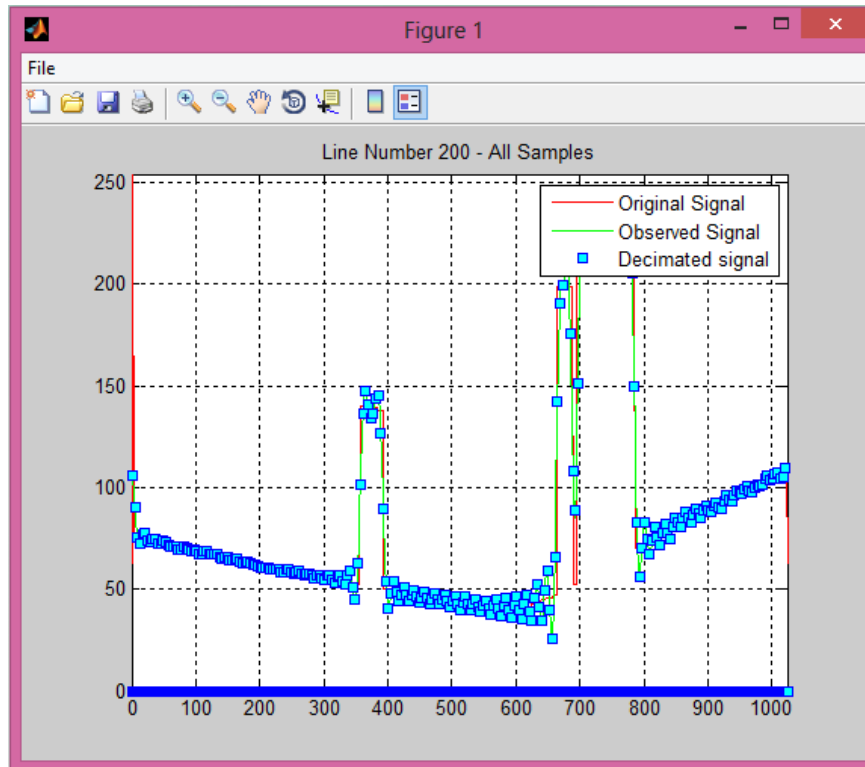


No final aparece uma janela a informar que o processo foi concluído. E a janela com o número de amostras perdidas é preenchida de acordo com o valor de JUMP escolhido.

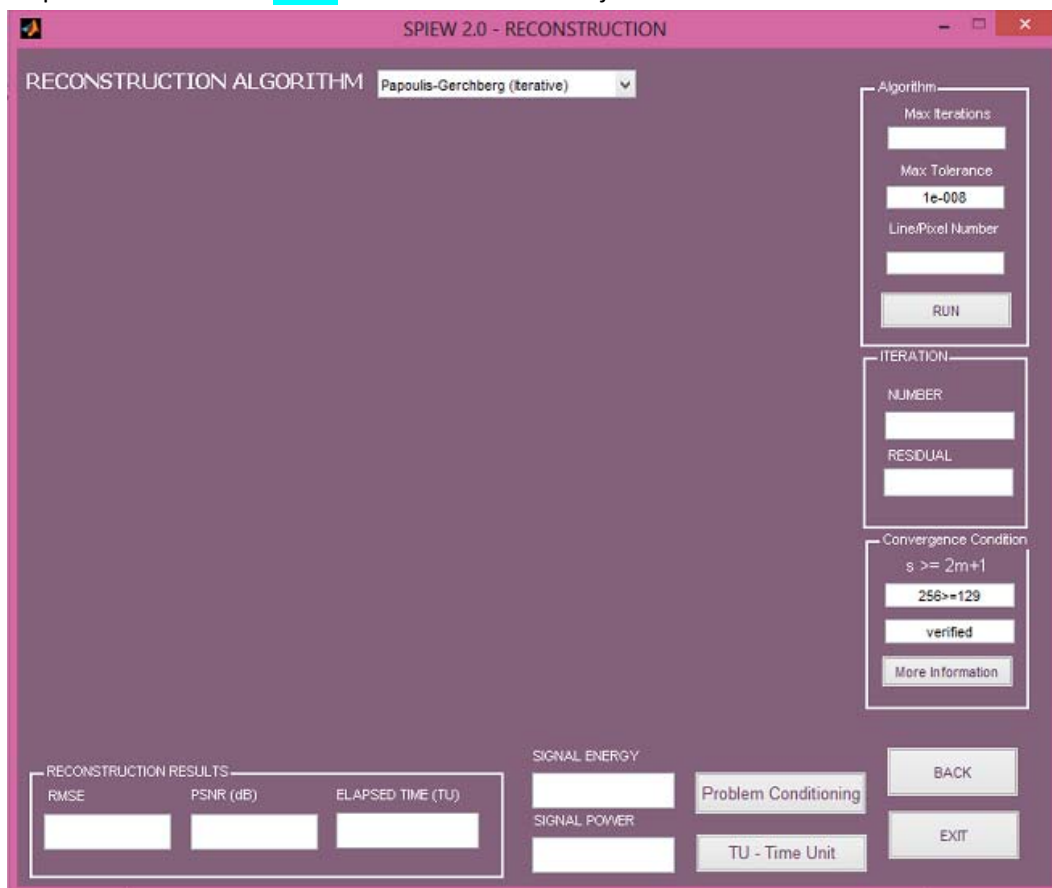
Aqui sugere-se o teste com os 3 padrões de geometria de erros, ou métodos de decimação, e ver como é condicionada a convergência dos algoritmos de reconstrução.



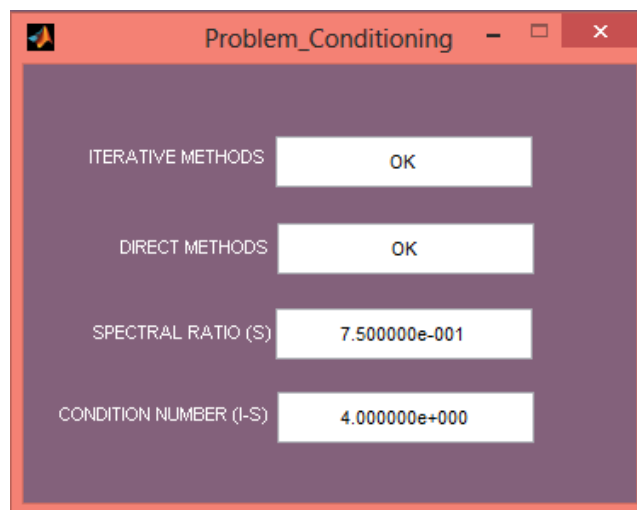
Depois de terminada a decimação dos sinais, ficam ativos o botão **NEXT** e **VIEW SIGNAL**. O primeiro abre diretamente a janela de para a reconstrução dos sinais. O segundo permite que o utilizador visualize as três versões de cada um dos sinais extraídos da sequência, sinal original, observado e decimado. Para tal tem que preencher o **PIXEL/LINE NUMBER**, com o número do sinal escolhido pressionar o botão **VIEW SIGNAL**.



12. Para passar ao então ao processo de reconstrução e testar os algoritmos escolhidos o utilizador deve pressionar o botão **NEXT**. Abrir-se-á uma nova janela.

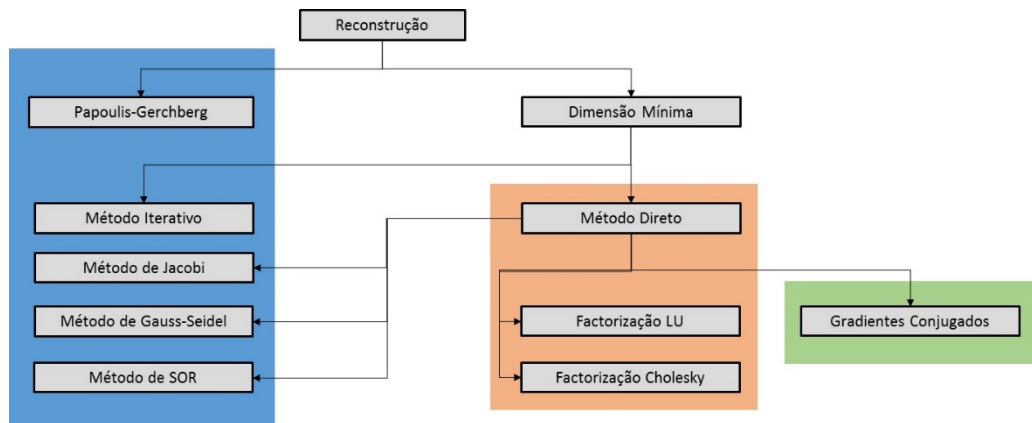


13. Aqui o utilizador poderá fazer a análise da convergência do problema em estudo, antes de escolher o algoritmo de reconstrução. Para tal tem ao seu dispor:
- Condição de convergência.
 - E os parâmetros de análise do condicionamento do sinal.
 - Clicando no botão **PROBLEM CONDITIONING** aparecem os valores do raio espectral da matriz S e o número de condição da matriz (I-S).
 - São estes os valores utilizados para determinar se o problema tem ou não solução
 - Para os métodos iterativos o utilizador deve analisar o raio espectral da matriz S. Um valor $\ll 1$ significa que o método iterativo convergirá para a solução. Já um valor muito próximo de 1 ou igual a 1, significa que utilizando os métodos iterativos, o problema não irá convergir para a solução.
 - Para o método direto o valor de referência é o do número de condição. Um valor inferior à 10^{15} significa que o problema tem solução. Mas um valor próximo ou igual a esta ordem de grandeza significa que o problema não tem solução.

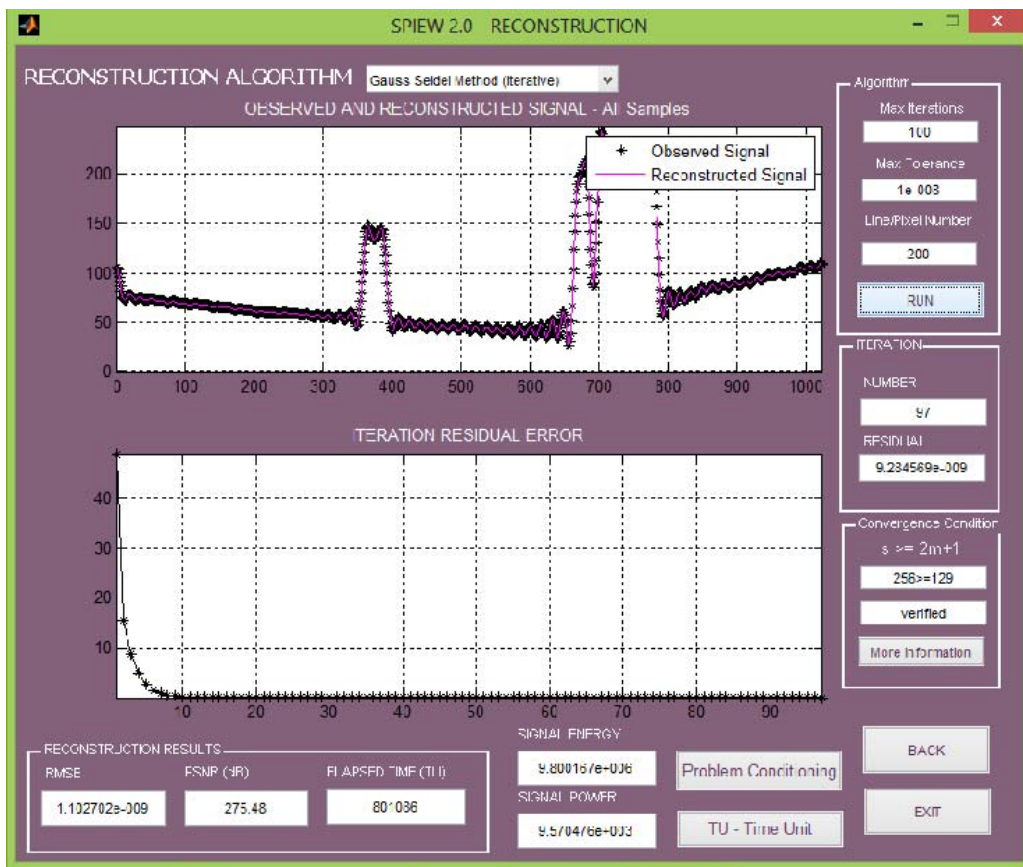


O objetivo desta aplicação é dar ao utilizador a possibilidade de testar algoritmos e métodos de resolução distintos. Sendo assim optou-se por implementar dois algoritmos distintos, o algoritmo de Papoulis-Gerchberg e um algoritmo de Dimensão Mínima.

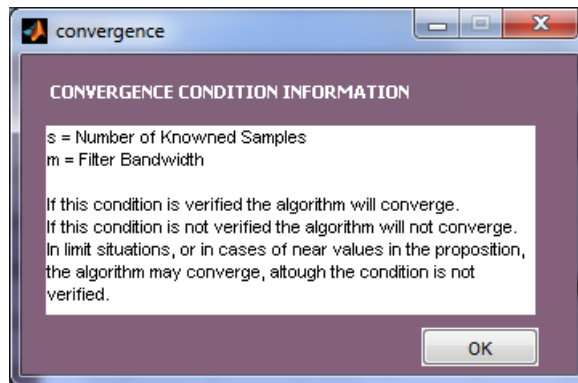
Para além dos dois algoritmos mencionados anteriormente, foram igualmente adicionados à aplicação novos métodos de resolução do problema, quando o mesmo está na formulação direta do algoritmo de dimensão mínima. Na figura seguinte podemos observar quais os métodos que poderão ser utilizados na reconstrução dos sinais previamente corrompidos.



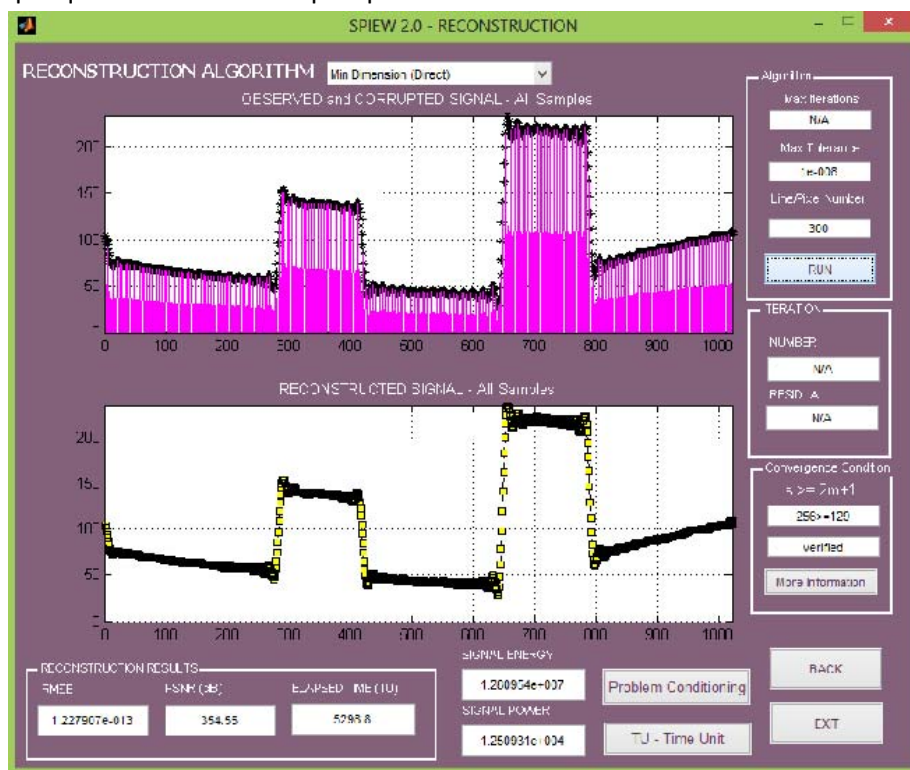
14. No caso de escolher um dos métodos iterativos, o utilizador deve indicar o número do sinal que pretende reconstruir e o número máximo de iterações.
15. Em todos os casos é indicado qual o número e qual o residual da iteração. No final o utilizador poderá avaliar o desempenho do algoritmo escolhidos através do tempo de reconstrução e do RMSE e PSNR do sinal reconstruído.



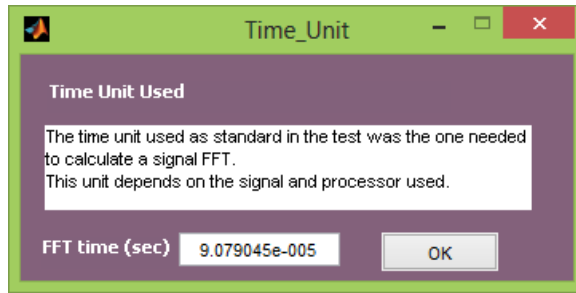
16. A convergência dos algoritmos é verificada através da condição indicada na aplicação. Para obter mais informação sobre a mesma e sobre as suas variáveis o utilizador deve pressionar o botão **MORE INFORMATION**.



17. No caso de escolher um dos Métodos Diretos, o utilizador apenas têm que indicar o número do sinal que quer reconstruir e depois pressionar o botão RUN.



18. Aqui a informação relativa às iterações não está disponível uma vez que o método é direto. Em todos os casos é possível visualizar o sinal observado e o sinal a ser reconstruído (durante as iterações) o após a sua reconstrução (nos métodos diretos).
19. É igualmente importante referir que o tempo de Reconstrução é medido em TU (Time Units). Para mais informação sobre este parâmetro, pressionar o botão **Time Unit**.



ANEXO B – INQUÉRITOS SPIEW

Inquérito SPIEW- Primeira versão

Questões	Verdadeiro	Falso
A um sinal decimado mais vezes corresponde um sinal com mais amostras não nulas.		
A versão reconstruída do sinal decimado 3x é mais perceptível do que o sinal decimado 1x antes de ser reconstruído.		
A uma filtragem passa-baixo com mais harmónicos não nulos corresponde um fator de sobre-amostragem, r , de menor valor.		
Quanto mais vezes o sinal for decimado, maior a garantia de reconstrução.		
A um filtro com maior largura de banda corresponde um sinal mais perceptível.		
O valor do fator de sobre-amostragem, r , depende do número de amostras nulas.		
O sinal reconstruído é mais perceptível do que o sinal original.		
O valor do fator de sobre-amostragem, r , depende do número de harmónicos não nulos.		
Se a condição de convergência for garantida para uma decimação de 3x, então está garantida a convergência para o mesmo sinal decimado 1x.		
O sinal observado é mais perceptível do que o sinal original.		
A garantia de convergência para um sinal decimado 1x não é garantia de convergência para um sinal decimado 2x.		
O valor do fator de sobre-amostragem, r , depende do número de amostras nulas do sinal.		
O tempo de reconstrução depende do valor do fator de sobre-amostragem, r .		
O sinal decimado 1x é mais perceptível do que o sinal observado.		
Uma filtragem com 1024 harmónicos não nulos dá origem a um sinal mais perceptível do que uma filtragem com 2048 harmónicos não nulos (conservando todas as restantes grandezas inalteradas).		
A uma filtragem com 4096 harmónicos não nulos corresponde um tempo de reconstrução superior ao de uma filtragem com 2048 harmónicos não nulos.		

Como caracteriza esta aplicação?

Muito Boa Boa Razoável Má

Considera a SPIEW uma mais-valia no seu processo de aprendizagem?

Inquérito SPIEW - Segunda versão

1. Qual o valor do factor de sobre amostragem (oversampling factor) que escolheria para perder o maior número de amostras possível?
2. Um factor de sobre amostragem baixo corresponde a uma Largura de Banda de Filtro Passa Baixo alta ou baixa?
3. Quando se pretende reconstruir os sinais, qual é a pior caso de geometria de padrão de erros? Porquê?
4. No caso da análise do condicionamento do problema ser favorável em ambos os casos, iterativo e directo, qual o algoritmo de reconstrução que escolheria? Porquê?
5. Se o raio espectral da matriz S é inferior a 1 qual o método de reconstrução pode utilizar na reconstrução dos sinais? Porquê?
6. Se o número de condição da matriz $(I-S)$ tiver um valor na ordem dos 10^{15} , a reconstrução do sinal é possível utilizando um dos 3 métodos disponíveis? Porquê?
7. Considera que o uso deste protótipo ajudou-o a adquirir ou a melhorar os seus conhecimentos na área?

Interpolação linear de Sinais

(SPIEW2.0, teste MeanOpinionScore)

1. Qual o valor do fator de sobre amostragem (*oversampling factor*) que escolheria para perder o maior número de amostras possível?

- 6.25%
- 12.5%
- 25.0%
- 50.0%
- 75.0%
- 100%

Explique porquê.

2. Um fator de sobre amostragem baixo corresponde a uma Largura de Banda de Filtro Passa Baixo alta ou baixa? (verifique com a aplicação)

3. Quando se pretende reconstruir os sinais, qual é a situação mais desfavorável no que diz respeito à geometria de padrão de erros?

- Entrelaçada
- Aleatória
- Contígua

Explique porquê.

4. No caso da análise do condicionamento do problema ser favorável em ambos os casos, iterativo e direto, qual o algoritmo de reconstrução que escolheria?

- Papoulis-Gershberg (iterativo)
- Dimensão Mínima Iterativo
- Jacobi (iterativo)
- Gauss Seidel (iterativo)
- SOR (iterativo)
- Dimensão Mínima Direto
- Factorização LU (direto)
- Cholesky (direto)
- Gradientes Conjugados (semi-iterativo)

Explique porquê.

5. Se o raio espectral da matriz S é inferior a 1 qual o método de reconstrução pode utilizar na reconstrução dos sinais? Porquê?

6. Se o número de condição da matriz $(I-S)$ tiver um valor na ordem dos 10^{15} , a reconstrução do sinal é possível utilizando todos métodos disponíveis? Porquê?

7. Considera que o uso deste protótipo ajudou-o a adquirir ou a melhorar os seus conhecimentos na área?