

Ramon Vieira Canales

**PROCESSAMENTO DE SINAIS PARA
INSPEÇÃO DE CORROSÃO EM DUTOS
POR PIG ULTRASSÔNICO USANDO
FILTRO CASADO**

São Paulo
Março de 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Ramon Vieira Canales

**PROCESSAMENTO DE SINAIS PARA
INSPEÇÃO DE CORROSÃO EM DUTOS
POR PIG ULTRASSÔNICO USANDO
FILTRO CASADO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia Mecânica.

São Paulo
Março de 2010

Ramon Vieira Canales

**PROCESSAMENTO DE SINAIS PARA
INSPEÇÃO DE CORROSÃO EM DUTOS
POR PIG ULTRASSÔNICO USANDO
FILTRO CASADO**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Mecânica.

Área de concentração:
Engenharia de Controle e Automação Mecânica

Orientador:
Prof. Dr. Celso Massatoshi Furukawa

São Paulo
Março de 2010

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 24 de maio de 2010.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Canales, Ramon Vieira

Processamento de sinais para inspeção de corrosão em dutos por PIG ultrassônico usando filtro casado / R.V. Canales. -- ed.rev. -- São Paulo, 2010.

144 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Ensaaios não destrutivos 2. Processamento de sinais acústicos 3. Corrosão 4. Dutos 5. Tempo-real I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II. t.

*Dedico este trabalho à todos aqueles que lutam pela preservação do planeta e da vida
(pelo menos a vida que merece ser preservada)*

Agradecimentos

Não posso deixar de começar estes agradecimentos pela pessoa responsável pela minha criação, por me dar a possibilidade de trilhar os caminhos que eu quis escolher, por sempre ter me apoiado, e por ter me ensinado tudo que foi preciso para caminhar sozinho, minha mãe. Também agradeço à todos os meus familiares, ao menos os que sabem que fiz mestrado.

Agradeço à minha companheira (espero que para vida inteira), Marcela Fanti, que me dá forças quando preciso, que me atura e que compartilha os bons e maus momentos da vida.

Seguindo, agradeço ao meu orientador, Celso Furukawa, que me guiou desde o começo da faculdade, compartilhando seu conhecimento e experiência, e que possibilitou que hoje eu me tornasse um mestre em Engenharia.

Agradeço às instituições que possibilitaram este trabalho: ao CNPQ (pela bolsa de estudo, sem a qual eu não teria sobrevivido estes anos), ao CENPES/Petrobras e Pipeway (pelo apoio técnico) e à FINEP (pelo apoio financeiro, convênio *n*º 01.05.0823.00).

Agradeço a todos os meus amigos, que de uma forma ou de outra estiveram presentes durante este trabalho.

E lógico, agradeço à você, que investiu tempo lendo estas palavras, e que provavelmente investirá muito mais lendo o restante deste trabalho!

Por último, muito obrigado a Deus, Poseidon, Thor, Baco, Amon-RÁ, Tupã, O Segredo, ou seja lá como quiserem chamar!

Resumo

Este trabalho descreve o desenvolvimento de um algoritmo para avaliar o grau de corrosão em paredes de oleodutos, através de ultrassom. O algoritmo desenvolvido será implementado em um circuito digital, fazendo parte de um sistema que poderá ser embarcado em pigs ultrassônicos para inspeção de dutos à procura de corrosão. O modelo matemático conhecido como filtro casado é usado como base do algoritmo, além de um método de detecção de picos para localização dos sinais de eco de ultrassom, provenientes das reflexões do sinal nas paredes dos dutos. Um dos problemas é a definição do sinal (e do filtro) que será utilizado, mas por meio de sinais sintetizados (*signal design*) uma grande gama de alternativas foi analisada e apenas algumas foram selecionadas para testes em experimentos. Por último os melhores sinais foram utilizados na inspeção de um corpo de prova feito à partir de um duto corroído, e os resultados são analisados. A detecção da primeira parede do oleoduto apresenta resultados promissores em comparação com a medição tridimensional das mesmas paredes. O sistema foi capaz de reconstruir o perfil da parede interna do duto com elevada verossimilhança, mesmo em regiões severamente corroídas. A detecção da segunda parede apresenta resultados satisfatórios para baixos níveis de corrosão, porém alguns problemas foram encontrados para níveis mais altos de corrosão, prejudicando a medição direta da espessura da parede nestas condições.

Palavras-chave: *Pig ultrassônico, Inspeção não destrutiva por ultrassom, DSP em tempo-real, Filtro casado, Códigos de Barker, Signal Design*

Abstract

Signal processing for corrosion assessment in pipelines with ultrasound PIG using matched filter

This work describes the development of an algorithm that uses ultrasound to evaluate the degree of corrosion in pipeline walls. The developed algorithm was implemented in a dedicated digital circuit that can be embedded in ultrasonic pigs for pipeline corrosion inspection. The algorithm is based on the mathematic model known as matched filter. It also incorporates a peak detection method for the localization of ultrasound echoes, originated from the signal reflections in the pipeline walls. One of the problems is the determination of the signal (and filter) that should be used, but using a signal design approach, a large number of possibilities could be analyzed so that only a few had to be tested in experiments. Finally, the best signals were used to inspect a sample made of a corroded pipeline section, and the results were analyzed. The detection of the first wall showed promising results when compared to the results of a tridimensional inspection of the same walls. It was possible to reconstruct the profile of the internal wall with good likelihood, even in badly corroded areas. The detection of the second wall showed some satisfactory results on low corrosion levels, but showed some problems with high corrosion levels, compromising the direct measurement of the pipeline wall thickness.

Keywords: *Ultrasonic pig, Ultrasonic non-destructive testing, Real-time DSP, Matched Filter, Barker codes, Signal Design*

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

1	Introdução	18
1.1	Contextualização e Motivação	18
1.2	Objetivos	19
1.3	Descrição do Trabalho	20
2	Revisão Bibliográfica	22
2.1	Pigs	22
2.2	Tratamento estatístico	23
2.3	Processamento Digital	24
2.4	Filtro Casado e Código de Barker	26
3	Aspectos Teóricos	29
3.1	Ultrassom	29
3.1.1	Os sinais de Ultrassom	29
3.1.2	A reflexão das ondas	30
3.1.3	O TOA (<i>Time of Arrival</i>) e a inspeção por ultrassom	31
3.1.4	Transdutores de Ultrassom	33

3.2	Processamento digital de sinais	37
3.2.1	Sistema Embarcado - FPGA	38
3.2.2	Processamento Digital	39
3.3	Algoritmos matemáticos	39
3.3.1	Estimação do TOA	39
3.3.2	Filtro Casado	41
3.3.3	Códigos de Barker	45
3.3.4	Comparação do erro entre duas medidas	49
3.4	Os problemas da inspeção por ultrassom	50
3.4.1	Critério	52
4	Metodologia	53
4.1	Cálculo do Tempo de Inspeção	53
4.2	Excitação dos Transdutores	55
4.2.1	Potência enviada	58
4.2.2	Códigos de Barker e número de ciclos	60
4.2.3	Detecção de Picos	67
4.3	Sistema Proposto	69
4.3.1	Placa Pulsadora	75
5	Testes reais	77
5.1	Primeira etapa	77
5.1.1	Corpo de Prova 1	78
5.1.2	Resultados	78
5.2	Segunda Etapa	81
5.2.1	Corpo de Prova 2	82

5.2.2	Escaneamento 3D	83
5.2.3	Inspeção ultrassônica	84
6	Resultados	86
6.1	Método de comparação	86
6.2	Medição da Primeira Parede	87
6.2.1	Principais problemas da inspeção	90
6.3	Medição da Segunda parede	94
7	Conclusão	102
7.1	Trabalhos Futuros	105
	Referências	107
	Anexo A – Código MATLAB do Detector de picos locais	111
	Anexo B – Esquemáticos do Sistema Digital	113
B.1	TOP	114
B.2	MEMORIA	115
B.3	MEMORIA PICOS	116
B.4	M8	117
B.5	IRQBUF	118
B.6	IOBUF8	119
B.7	FILTRO	120
B.8	DETECTOR QUADRATURA	121
B.9	DETECTA TRIGGER	122
B.10	CLOCK IN	123
	Anexo C – Códigos em VHDL dos blocos do esquemático	124

C.1	Contador Encoder	125
C.2	Contador Sinal	126
C.3	Controle Acq	127
C.4	Controle Detec	128
C.5	Delay Acq	129
C.6	Detector Pico	131
C.7	Endereçador	132
C.8	Filtro Casado	133
C.9	OBUS MUX	136
C.10	Prog Register	137
C.11	Salva Sinal	140
C.12	Shifter	141

Lista de Figuras

2.1	Em (a) imagem de um pig ultrassônico, e em (b) uma figura com seus principais componentes	23
3.1	a) Propagação da onda longitudinal e b) Propagação da onda transversal .	30
3.2	Reflexão e Transmissão de uma onda acústica	30
3.3	Envio e recebimento dos ecos do sinal de ultrassom, e seus respectivos TOAs	32
3.4	Sinais de eco obtidos em água, (a) gerado por uma placa de alumínio plana e polida; e (b) gerado em região corroída da placa	32
3.5	Modelo simplificado de um transdutor de ultrassom	34
3.6	Em (a) um transdutor não-focalizado, e em (b) um transdutor focalizado .	36
3.7	Em (a) o transdutor não focal está dentro do campo de ondas refletidas por uma superfície irregular, o que não acontece em (b) com o transdutor focalizado	37
3.8	Estrutura interna simplificada de uma FPGA. E/S - blocos de entrada e saída, B.L. - blocos lógicos e M - matrizes de interconexão	38
3.9	Em (a) o sinal conhecido, e em (b) com adição de ruído branco, dificultando a determinação do TOA	40
3.10	Exemplo da saída de um sistema linear	43
3.11	Em a) tem-se o sinal original, em b) sua inversão no tempo e escalado por um fator k , e em c) aplicando um atraso A tem-se a função de transferência do filtro causal	44
3.12	Em (a) o sinal em meio a ruído e com um atraso desconhecido (o tracejado mostra o sinal original) e em (b), a saída do filtro casado, com o pico mostrando qual o valor de A que maximiza a função, e determinando assim a localização do sinal original no sinal recebido.	45

3.13	Exemplo de modulação BPSK. A onda portadora é defasada conforme o código binário (0° para código 1 ou 180° para código -1)	46
3.14	Uma função de correlação ideal	47
3.15	Código de Barker com (a) 3 bits; e (b) 7 bits	48
3.16	Auto correlação dos códigos de Barker com (a) 3 bits; e (b) 7 bits	48
3.17	Sinal enviado sofre interferências do meio em a), e depois é correlacionado consigo mesmo em b), resultando em um pico distinto mostrando a posição do sinal, com um atraso fixo	49
3.18	Exemplo de onda de excitação com 3 bits, 2 ciclos por bit	49
3.19	Em (a), a dispersão que pode ser causada pela corrosão, e em (b) um caso onde um eco lateral chega no transdutor antes do eco do ponto abaixo do mesmo	51
4.1	Representação do pig durante uma inspeção dentro do oleoduto. As flechas claras indicam o sinal sendo enviado e recebido pelo transdutor	54
4.2	Função de Transferência do Transdutor ALPHA: (a) no domínio do tempo e; (b) densidade espectral de potência	56
4.3	Comparação entre sinal real e sinal sintetizado por computador, no caso de 2 bits e 8 ciclos por bit. Sinal no tempo: (a) Real e (b) Sintetizado. Densidade Espectral de Potência: (c) Real e (d) Sintetizado.	57
4.4	Sinal real codificado que excita o transdutor (7 bits, 1 ciclo por bit). (a) Sinal real e (b) sinal utilizado nas simulações	57
4.5	Transmissão de potência relativa ao maior valor, em dB, para: (a) todos os sinais e, (b) apenas os sinais dentro do limite estabelecido	59
4.6	A onda de excitação é convoluída com o sinal de eco padrão, e o resultado montado em um sinal maior com distância d_{ecos} . Por fim adiciona-se ruído branco gaussiano.	61
4.7	Sinal obtido em uma superfície plana de alumínio com seus múltiplos ecos gerados por reflexões internas nas paredes	62

4.8	Resultado das simulações para o caso: (a) 2 bits e 4 ciclos e (b) 2 bits e 7 ciclos. Os erros são expressos em percentagem em relação ao valor de espessura pré-determinado na síntese dos sinais	64
4.9	Saídas do filtro casado. 2 bits 4 ciclos com (a) F8B e (b) F1B, e 2 bits 7 ciclos com (c) F8B e (d) F1B	64
4.10	Erros em percentagem na simulação dos códigos. Em (a) todos os resultados e em (b) limitando nos erros menores que 2%	66
4.11	Em (a) a sobreposição do sinal causou muita ambiguidade na saída do filtro, e o detector de picos detectou o pico errado ("×"), ao invés do correto, representado por "o". Em (b), mesmo com a sobreposição, a saída do filtro é muito clara, resultando numa detecção correta	67
4.12	Parâmetros configuráveis na detecção de picos	68
4.13	Exemplo do detector de picos locais, resultando na envoltória do sinal. São detectadas as envoltórias positiva e negativa, pois no caso do segundo eco procura-se o pico negativo	69
4.14	Diagrama de blocos simplificado do sistema implementando na FPGA	70
4.15	Exemplo de uma configuração de um filtro qualquer, com somadores em série	73
4.16	Em a) o filtro de 2 bits, e suas respectivas amostras, e em b) o diagrama representando a implementação do mesmo	74
4.17	Diagrama de blocos da placa pulsadora/receptora	76
5.1	Corpo de prova de alumínio, usinado como uma escada com diversas espessuras de degraus. (a) Imagem real e (b) suas dimensões	78
5.2	(a) Sinal adquirido na inspeção do corpo de prova 1, para o caso de 7 bits, 2 ciclos por bit, no degrau de 10 mm; e (b) Saída do filtro, com picos bem definidos, resultando num bom resultado na inspeção	80
5.3	Saída do filtro casado para o caso de: (a) 7 bits, 1 ciclos por bit e (b) 7 bits, 2 ciclos por bit, ambos na espessura de 2 mm	80
5.4	Saída do filtro casado para o caso de 3 bits, 3 ciclos por bit na espessura de 2 mm	81

5.5	Diagrama representando o sistema de inspeção utilizado na segunda etapa de testes	82
5.6	Circuito utilizado para o separador de sinais, com $R1 = 100K\Omega$ e diodos rápidos 1N4148	82
5.7	Imagens da placa utilizada como corpo de prova nos experimentos	83
5.8	Foto da máquina BN710 da Mitutoyo	83
5.9	Linhas escaneadas pela máquina 3D, reproduzidas em MATLAB	84
5.10	(a) Foto do sistema mecânico; (b) Diagrama mostrando o sistema de inspeção e o posicionamento da peça a ser escaneada	85
5.11	Diagrama exemplificando o alinhamento visual que foi feito durante a inspeção	85
6.1	(a) Resultados das inspeções soltos no espaço, e (b) resultados com primeiro ponto igualado	86
6.2	(a) Resultados com diferença angular, e (b) resultados alinhados prontos para inspeção	87
6.3	Comparação entre os resultados da primeira parede para a linha D, com o código 2 Bits, 3 ciclos	88
6.4	Comparação entre os resultados da primeira parede para (a) linha A, com o código 3 Bits, 3 ciclos, e (b) linha B, 7 Bits, 1 ciclo	89
6.5	Onda de ultrassom refletindo nas paredes laterais de um pite, antes de chegar no fundo do mesmo	90
6.6	Dois exemplos onde o pite não foi corretamente detectado, aparecendo como um buraco menor que o buraco real	91
6.7	Sinal de eco do ponto 60 <i>mm</i> do comprimento, da linha A, para o código de 3 bits, 3 ciclos. E ponto detectado pertence a um ciclo anterior ao ciclo do TOA real do eco.	92
6.8	Sinal com seu primeiro eco corrompido	93

6.9	Comparação entre os resultados para a primeira e segunda parede: (a) Linha A, 2 Bits, 3 Ciclos; (b) Linha B, 3 Bits, 3 Ciclos; (c) Linha D, 3 Bits, 3 Ciclos	94
6.10	(a) Resultado da inspeção e (b) sinal para o ponto 23 <i>mm</i> do comprimento	95
6.11	Sinal para o ponto 24 <i>mm</i> do comprimento	96
6.12	(a) Resultado da inspeção e (b) sinal para o ponto 23 <i>mm</i> do comprimento	96
6.13	(a) Resultado da inspeção e (b) sinal para o ponto 51 <i>mm</i> do comprimento	97
6.14	(a) Resultado da inspeção e (b) sinal para o ponto 60 <i>mm</i> do comprimento	97
6.15	(a) Resultado da inspeção e (b) sinal para o ponto 30 <i>mm</i> do comprimento	98
6.16	Sinal para o ponto 45 <i>mm</i> do comprimento	98
6.17	(a) Resultado da inspeção e (b) sinal para o ponto 37 <i>mm</i> do comprimento	99
6.18	Sinal para o ponto 69 <i>mm</i> do comprimento	99
6.19	Simulação de detecção da segunda parede com uma janela variável de tempo, com duração menor	100
6.20	Inspeção da segunda parede da linha A para 3 combinações de códigos e ciclos	100

Lista de Tabelas

3.1	Códigos de Barker	47
4.1	Códigos que serão simulados	63
4.2	Códigos para os testes reais	65
5.1	Resultados (em <i>mm</i>) dos testes com o corpo de prova de alumínio.	79

Lista de Abreviaturas

ASIC *Application-Specific Integrated Circuit*

BPSK *Binary Phase Shift Keying*

DSP *Digital signal Processing*

FIR *Finite Impulse Response*

FPGA *Field-Programmable Gate Array*

GPS *Global Positioning System*

LIT *Linear e Invariante no Tempo*

MFL *Magnetic Flux Leakage*

ML *Maximum Likelihood*

MPS *Minimum Peak Sidelobe*

PDF *Probability Density Function*

PSK *Phase Shift Keying*

Pigs *Pipeline Inspection Gauge*

RFEC *Remote Field Eddy Current*

S/N *Sinal/Ruído*

TOA *Time of Arrival*

VR *Variable Reluctance*

WGN *White Gaussian Noise*

1 Introdução

O petróleo está longe de ser um combustível obsoleto, já que mesmo com algumas reservas perto de seu fim, novas são constantemente encontradas, e a previsão para o esgotamento do óleo fica cada vez mais distante. Com isso, continua o crescimento desta indústria, e conseqüentemente a necessidade de inovações tecnológicas para essa área.

1.1 Contextualização e Motivação

Dutos são o principal meio de transporte utilizado na indústria do petróleo, desde 1859, com o primeiro poço comercial na Pensilvânia (PIPELINE101, 2008). Com o passar dos anos, muita coisa mudou, porém os dutos continuam sendo o modo mais barato e seguro de se transportar o óleo, sendo utilizados em larga escala em todo o mundo. Porém, apesar de ser o mais seguro, os dutos estão sujeitos a sofrer corrosão, e com isso causar sérios acidentes.

As grandes causas desta corrosão são o fato do duto trabalhar em ambiente altamente corrosivo (como no fundo do mar, ou até mesmo ao ar livre) e transportar fluidos contendo contaminantes (como água salgada e areia), que tendem a se acumular na parte inferior do mesmo, deteriorando essa região.

A corrosão pode levar ao vazamento de fluidos tóxicos, contaminando o ambiente e podendo causar sérios acidentes, envolvendo inclusive a vida de seres humanos, seja direta (por exemplo explosões) ou indiretamente (contaminando um manancial). Esses acidentes acarretam o gasto de muito dinheiro, seja no conserto de dutos, ou reparando os danos ambientais. Entre 1990-99, apenas nos Estados Unidos, foram registrados cerca de 3900 vazamentos de óleo e gás, com 201 mortes, e mais de \$780 milhões gastos apenas com danos em propriedades (INSLEE, 2008). Já a extinta União Soviética obteve uma taxa de 46% de problemas em seus oleodutos, sendo que 39,9% eram por problemas de corrosão

(STAROSTINV, 1990). Por isso, é de grande interesse das indústrias do petróleo, e da sociedade, que estes acidentes possam ser prevenidos através de inspeções dos oleodutos à procura de regiões danificadas.

Porém, estas inspeções podem ser complexas e dispendiosas. Os diâmetros estreitos das tubulações tornam inviável para uma pessoa fazer uma inspeção interna das mesmas, e além disso, é desejável que o funcionamento do duto seja minimamente afetado, evitando parar o fluxo e causar prejuízos na produção. Por estes motivos é necessária a utilização de equipamentos que possam ser inseridos na tubulação. Equipamentos desse tipo são conhecidos como *Pigs (Pipeline Inspection Gauge)* instrumentados.

Os pigs podem carregar diversos tipos de sensores (como magnéticos, táteis e ultrassônicos), e através da análise dos sinais produzidos e colhidos por estes sensores geram-se conclusões sobre o estado dos dutos. Este tipo de equipamento já é utilizado nos dias de hoje, porém o resultado das inspeções ainda não é totalmente confiável, devido a diversos problemas que podem aparecer durante as mesmas. Falando especificamente de inspeções ultrassônicas, encontram-se problemas devido a atenuação e dispersão dos sinais de ultrassom, sobreposição de ecos e presença de ecos laterais. Além disso poderão existir ruídos nos sinais da inspeção. Estes problemas serão explicados mais adiante.

Devido a estas dificuldades, se torna muito importante o desenvolvimento de novas tecnologias, que diminuam ao máximo o uso de recursos para gerar resultados confiáveis nas inspeções, beneficiando não somente à indústria (com a diminuição de gastos, e a possibilidade de aumentar a quantidade de inspeções), mas também à sociedade, que não sofrerá com grandes catástrofes.

1.2 **Objetivos**

Neste trabalho será desenvolvido um algoritmo para detecção dos tempos de chegada (*TOAs - Time of Arrival*) dos ecos de um sinal de ultrassom em uma inspeção não destrutiva. O objetivo deste algoritmo é melhorar o resultado de inspeções de dutos, aumentando a sua confiabilidade, e diminuindo os efeitos causados pelos problemas citados anteriormente. O algoritmo será implementado em um circuito digital através de um dispositivo lógico programável (*FPGA - Field-Programmable Gate Array*), testado em inspeções controladas simulando situações reais, com o intuito de ser embarcado em pigs e testado em inspeções reais. O algoritmo consiste no envio de sinais de ultrassom

codificados com código de Barker, de modo que a detecção de seus ecos em meio à ruído seja otimizada, através de um filtro casado implementado como um filtro *FIR - Finite Impulse Response*.

Ao final do trabalho o algoritmo desenvolvido estará implementado em um circuito digital, pronto para ser embarcado em um pig e testado em inspeções reais.

1.3 Descrição do Trabalho

Este trabalho começa com uma revisão bibliográfica sobre instrumentos utilizados (capítulo 2) na inspeção de oleodutos, assim como métodos e sensores utilizados nesses equipamentos. Após isso é feita uma revisão sobre processamento digital de sinais, e como isso influencia este presente trabalho. Por último, a revisão se concentra em algoritmos matemáticos que podem se relacionar com o escopo geral, e como alguns destes algoritmos já são utilizados para estes fins.

Após isso é feita uma apresentação dos aspectos teóricos relacionados com este trabalho (capítulo 3), e que são de suma importância para compreensão do algoritmo que foi elaborado, assim como a compreensão de sua implementação e funcionamento. São apresentados aspectos teóricos sobre ultrassom (incluindo inspeção e transdutores de ultrassom), processamento digital, e implementação de circuitos digitais, e matemática relativa ao algoritmo (filtro casado e códigos de Barker). Neste momento também serão explicados os problemas encontrados nas inspeções ultrassônicas, e como o algoritmo proposto pode contornar alguns deles.

Seguindo, é apresentada a metodologia utilizada para desenvolvimento de todo o sistema (capítulo 4). Na metodologia estão inclusos todos os cálculos e simulações para determinação dos parâmetros que são utilizados no algoritmo.

A decisão de que sinal de excitação deve ser utilizado foi tomada em duas fases. Inicialmente foram feitos estudos de *signal design* para decidir um conjunto de sinais (codificados com código de Barker) que seriam testados em experimentos reais. Esta primeira fase foi executada através de simulações no computador, com base em sinais reais adquiridos em inspeções, de modo a conseguir resultados os mais fiéis possíveis. Com isso foi possível ter resultados de quais sinais de Barker seriam mais eficazes levando em conta uma relação de compromisso entre potência enviada, precisão do resultado e maior taxa S/R (*Sinal / Ruído*). Esta abordagem se mostrou muito eficaz, pois reduziu a necessidade de inúmer

ros testes em laboratório, que teriam que ser repetidos incessantemente e tomariam um tempo indeterminado. Com o *signal design*, a quantidade de possibilidades testada foi muito superior e com isso conseguiu-se determinar os melhores candidatos para sinal de excitação dos transdutores, podendo-se testar apenas os mesmos na parte experimental.

Além disso são apresentadas características do sistema, assim como algoritmos secundários que foram utilizados.

Após isso, os melhores resultados foram testados em inspeções controladas, para saber a eficiência real do algoritmo, além de validar a implementação do algoritmo digitalmente.

Com o algoritmo praticamente definido são apresentados os corpos de prova que foram utilizados para realizar os primeiros testes de inspeção reais. Dois corpos de prova foram utilizados. O primeiro é uma peça de alumínio usinada em degraus. Com este corpo de prova o algoritmo foi testado já implementado no circuito digital, e os resultados se mostraram bastante promissores (capítulo 5).

O segundo corpo de prova consiste de um pedaço de oleoduto corroído aplainado, formando uma placa. As inspeções com esse corpo mostraram como o sistema se comportaria em uma inspeção real. Os resultados para a primeira parede da peça foram excelentes, mostrando grande robustez no método utilizado. Os resultados para a segunda parede mostraram que ainda existem certos problemas que precisam ser resolvidos quando se inspeciona uma região muito corroída, porém a segunda parede passa a ser um subproduto quando lidamos com regiões deterioradas, pois a partir da primeira parede (do duto) já vemos que a peça está comprometida e necessita de reposição.

Todos estes resultados e problemas são analisados e explicados (capítulo 6), e assim, são apresentadas as conclusões de todo o trabalho realizado. Por último são listados os próximos passos que podem ser tomados para a continuidade dessa pesquisa.

2 Revisão Bibliográfica

2.1 Pigs

Os *pigs* são dispositivos que servem para levar sensores e outros instrumentos pelo interior de tubulações, e são usados já há cerca de 40 anos como ferramentas de detecção de falhas e perdas de metal em oleodutos (e outras funções, como limpeza de dutos). De uma geração tecnológica para outra, mais e mais refinamentos vão sendo introduzidos, seja para simplesmente melhorar o resultado das análises ou para acompanhar novas normas de inspeção (REBER et al., 2002).

Os pigs se movimentam nas tubulações utilizando o próprio fluxo de fluido existente, sendo impulsionados através de discos colocados em seus corpos, e podem levar diversos tipos de sensores para detectar a perda de metal, seja por contato (como nos pigs táteis), ou por meios sem contato (como nos magnéticos e ultrassônicos). Os pigs ultrassônicos vêm se destacando em comparação aos demais, devido a possibilidade de se medir diretamente a espessura da parede, o que não é possível com os outros pigs. A figura 2.1 mostra um típico pig ultrassônico dentro de uma tubulação, e seus principais componentes.

O ultrassom oferece melhor precisão do que as medidas por campo magnético, nas quais se detectam perdas de material pela variação da relutância da parede do duto. Trabalhos como o de Katragadda et al. (1996) utilizam o vazamento de fluxo magnético *MFL* (*Magnetic Flux Leakage*) para recuperar informações sobre as condições do duto. Porém, essas informações são mais incompletas que as fornecidas por uma inspeção por ultrassom. Por isso é proposto explorar processos físicos adicionais que são gerados pelo veículo de inspeção, como o efeito *RFEC* (*Remote Field Eddy Current*) induzido por movimento, o efeito de corrente nas bordas de defeitos induzidas por movimento, e o efeito *VR* (*Variable Reluctance*). Mesmo com esses processos adicionais, os resultados alcançados ainda não são satisfatórios. Já Tao, Peiwen e Zhengsu (2004) demonstram o

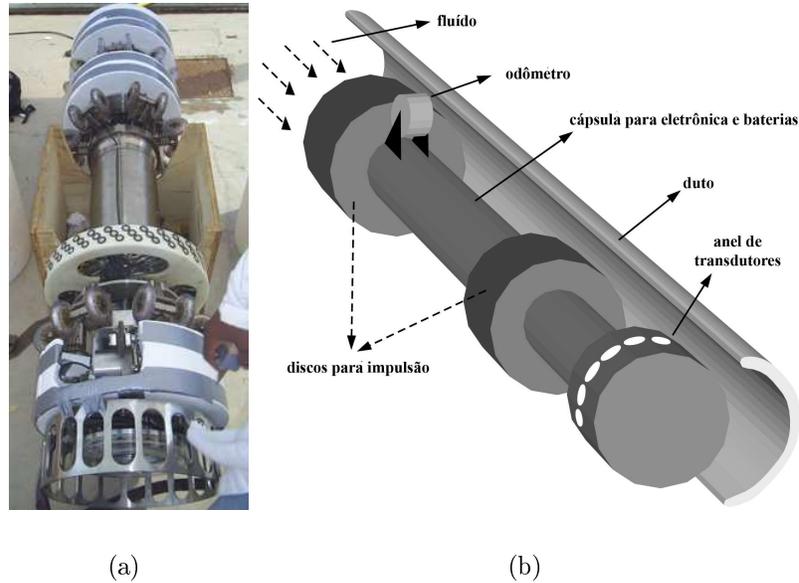


Figura 2.1: Em (a) imagem de um pig ultrassônico, e em (b) uma figura com seus principais componentes

desenvolvimento de um robô para inspeção de dutos baseado também no MFL, utilizando sensores magnéticos Hall. É mostrado que a presença de defeitos (como trincas, mossas, etc.), e variações de material afetam diretamente os resultados do robô, e o mesmo precisa ser equipado com diversos outros tipo de sensores para corrigir estes problemas.

Já os pigs táteis utilizam espécies de palitos para ir "tateando" a parede interna do duto. Estes palitos são ligados a sensores que em geral medem o movimento angular dos mesmos para medir o relevo interno da parede (PIPEWAY, 2008). Com essa abordagem consegue-se apenas medir o relevo interno do duto, e com uma resolução limitada.

Esses são alguns dos motivos que levam ao uso de pigs ultrassônicos nas inspeções de dutos. Trabalhos como o de Okamoto et al. (1999) mostram resultados com esse tipo de dispositivo, utilizando o envio de sinais acústicos, e detectando seus TOAs (a inspeção ultrassônica será explicada na seção 3.1.3).

2.2 Tratamento estatístico

Alguns trabalhos utilizam abordagens estatísticas para tentativa de melhorar a detecção dos TOAs. É o caso de Chan, Yau e Ching (2003), que apresentam um estimador linear de máxima verossimilhança para ser utilizado na indústria de telefonia, para localizar

telefones celulares. Já Swiercz (2006) propõe o uso do código de Barker de 13 bits para detectar um sinal em meio a ruído com dois estágios: aproximação caótica dinâmica e tomada de decisão através de redes neurais. Porém, métodos estatísticos não podem ser usados em processamentos em tempo-real, já que é necessário o acúmulo de informações para o pós-processamento dos dados, e o tempo e espaço para armazenamento de informações em aplicações desse tipo são restritos. Logo, métodos como estes não podem ser implementados em uma inspeção automatizada por pigs.

2.3 Processamento Digital

O uso do processamento digital de sinais se tornou indispensável na indústria atual, já que as possibilidades vão muito além das analógicas, tornando praticamente qualquer tratamento de sinal possível. Um exemplo disso é a evolução na detecção de *TOAs* comparando-se a abordagem utilizada por Furukawa, Adamowski e Camerini (1998), que era predominantemente analógica no tratamento do sinal: o momento de chegada do eco era detectado por um circuito de detecção de limiar (o *TOA* era definido no momento que o sinal ultrapassava uma certa amplitude), e a abordagem de Canales et al. (2007), apresenta um algoritmo de detecção de picos dos ecos de ultrassom, implementado digitalmente, resultando em estimativas muito mais precisas para os *TOAs* na inspeção de oleodutos, em comparação ao tratamento analógico.

Por outro lado, avanços no sentido de melhorar a detecção do sinal são apresentados por Reber et al. (2002), que apresenta uma nova geração de ferramentas de inspeção ultrassônica de alta resolução para inspeção de oleodutos. Seu trabalho sugere o uso de recursos de redução de dados e comenta o uso de ferramentas de software usadas para elaborar o controle do sistema. Identifica a realização de ferramentas de inspeção por ultrassom como sendo as mais desafiadoras em termos de processamento de sinal e sugere, pela menção ao uso de FPGAs no tratamento do sinal recuperado, uma abordagem digital do problema de determinação do momento de chegada do eco. O uso de sistemas ultrassônicos pode, inclusive, ser feito externamente à tubulação. Park, Jeong e Lim (2002), por exemplo, apresentam dois tipos de robôs autônomos para a inspeção de espessura de paredes de tubos de caldeiras e tubulações industriais. Estes robôs usam um sistema que mede a propagação do som em seções das paredes do tubo realizando uma inspeção externa do mesmo. Obviamente, esta abordagem somente permitiria a inspeção de trechos descobertos de oleodutos, tornando-a específica para poucos casos.

O trabalho de Balzer e Stripf (2002), por sua vez, apresenta um pig de alta resolução e grande alcance (menciona distâncias de até 250 km), capaz de pesquisar defeitos em toda a circunferência do tubo com uma resolução longitudinal de 3 mm à velocidade de 1 m/s. Este pig gera uma grande quantidade de dados que é submetida a um sistema de compressão on-line com grandes taxas de compactação, de forma a tornar possível o armazenamento de grandes volumes de dados em memória de estado sólido (cita-se que o sistema gerou 700 Tbytes, que foram comprimidos por um fator de 7000 para ser armazenados). Para tanto, o sistema faz uso de um conjunto DSP-FPGA, para pré-tratamento e compactação dos dados. O sistema utiliza um algoritmo para detecção dos picos no sinal digitalizado para fazer um passo da compactação do sinal. Porém, neste sistema, a classificação de defeitos é feita totalmente *off-line*, ou seja, o resultado não é gerado em tempo real, e nem mesmo pelo próprio pig.

Dai et al. (2007) propõe um sistema com diversos transdutores para inspeções automáticas de dutos, também baseado na medição das suas paredes. O algoritmo proposto utiliza a estimação do espectro de potência do sinal recebido, e através da determinação de suas frequências principais é possível determinar os intervalos de repetição dos ecos gerados nas reflexões internas das paredes. O problema é que tal algoritmo (como a maioria dos utilizados para estes fins) é muito suscetível a presença de ruído no sinal, que distorce totalmente a frequência de repetição dos sinais. Um algoritmo semelhante é usado por Martin et al. (1992), que mostra uma técnica para medir camadas (por exemplo em artérias humanas) baseada em transformada inversa de Fourier da derivada logarítmica corrigida do espectro de energia do sinal. Esse algoritmo já se mostra menos suscetível ao ruído, porém não totalmente livre, e além disso é um algoritmo que necessita de muito esforço computacional, não podendo ser implementado para inspeções *on-line*.

Trabalhos como os de Suh, Kim e Chung (1999) também sugerem que uma abordagem digital para o problema da recepção dos ecos pode trazer mais vantagens. Neste caso, é introduzida uma maneira para inspeção de parafusos em plantas críticas de usinas sem a necessidade de remoção, usando um método de Deconvolução Preditiva Dinâmica, visando melhorar a detecção de rachaduras dos testes convencionais. Este processo é aplicado para altas frequências (10 MHz), e é sugerido o desenvolvimento de um circuito integrado *ASIC* (*Application-Specific Integrated Circuit*) para aumentar sua velocidade de processamento. A inspeção de parafusos por ultrassom, assim como a inspeção de paredes de oleodutos, é baseada na determinação do tempo de retorno do eco. O uso da deconvolução preditiva elimina os sinais repetidos originados nas reflexões do sinal nas diferentes partes do corpo

do parafuso. Da mesma forma, haverá reflexões e ruídos que precisarão ser eliminados para a correta detecção do momento de chegada do pulso do eco durante a medição de espessura da parede de um oleoduto. Lin, Lee e Saul (2004) apresentam um algoritmo semelhante, utilizando deconvolução não-negativa para estimar a diferença de tempo de chegada entre sinais de som para localização de uma fonte emissora. O algoritmo é comparado com auto-correlação, alinhamento de fase e deconvolução simples, se mostrando superior aos três tanto na simulação, quanto em testes em campo. Estes dois métodos também ajudam a diminuir a influência de ruído, porém como o trabalho citado anteriormente, são altamente custosos computacionalmente, o que impossibilita suas implementações *on-line* com a tecnologia disponível atualmente.

O trabalho de Higuti (1994) apresenta diversos métodos para inspeção de corrosão em tubulações, focando principalmente nos métodos de detecção da espessura das mesmas. São exibidos métodos simples (como detecção de limiar e de picos, que não resolvem os problemas de ruídos em uma inspeção) e métodos mais complexos (como correlação cruzada, filtro de Wiener, filtro de conformação e múltiplas reflexões, que resolvem boa parte dos problemas). O problema é que os métodos mais complexos apresentados exigem alta carga computacional, e não podem ser implementados em *real-time*, que é a proposta deste trabalho. Além disso, Higuti faz um estudo sobre a influência da rugosidade da superfície inspecionada no resultado da inspeção.

Simon (1993) também mostra técnicas de processamento digital de sinais para aplicação em ensaios não destrutivos de ultrassom, muito semelhantes às apresentadas por Higuti, que caem no mesmo problema de não serem passíveis de implementação em tempo real. Simon também não considerou os problemas de dispersão e atenuação em seus modelos para simulações, e seus resultados experimentais são focados na influência da adição de ruído branco para cada um dos métodos.

2.4 Filtro Casado e Código de Barker

Pelos trabalhos citados, percebe-se que a influência do ruído no sinal é um ponto muito importante para a melhoria dos resultados das inspeções. Existem diversos métodos matemáticos para melhorar essa relação S/R, entre eles o filtro-casado.

Um trabalho que aponta para esta direção, juntamente com processamento digital, é o de Lima e Furukawa (2002), que propõe o uso de um estimador de máxima veros-

semelhança (o filtro casado) para detectar sinais conhecidos com adição de ruído branco gaussiano, com utilização de pulsos codificados com código Barker (códigos conhecidos por suas propriedades de autocorrelação, que serão melhor explicados na seção 3.3.3). A implementação do sistema também é feita digitalmente em uma FPGA. O sistema é utilizado para navegação de veículos submarinos, através de transponders ultrassônicos, e mostra resultados promissores para a estimação dos TOAs.

No trabalho de Akopian e Aghaian (2004) é mostrado um algoritmo de um filtro casado rápido no domínio do tempo, para uso em *GPS (Global Positioning System)*. O algoritmo desenvolvido se mostra mais rápido que abordagens com FFT. Já Jo et al. (1999) apresenta um filtro casado digital compacto, que consegue atingir um tempo de processamento menor que um filtro convencional, gerando assim resultados mais rápidos. Esse tipo de algoritmo se mostra muito importante num caso de inspeção em tempo-real.

O trabalho de Yedlin e Majer (1993) mostra as vantagens de uma modulação por fase (no caso uma *PSK (Phase Shift Keying)* com código de Barker) em relação a uma modulação de frequência, reforçando os benefícios de uso destes códigos.

Zhao, Mo e Gao (2007) utiliza o código de Barker para geração de imagens coloridas de fluxos por ultrassom. Neste trabalho é demonstrado como a seleção de um código de Barker deve ser feita baseando-se em diversos parâmetros do sistema, como banda do transdutor, frequência central, duração do sinal, entre outras. Em testes, conseguiu-se uma taxa de ruído de apenas 0.5 dB em relação à potência do sinal, utilizando um código de Barker de 5 bits e 4 ciclos, com fluxos rápidos de até 5 m/s.

O uso de filtros casados com códigos de Barker também foi explorado por Hu et al. (2006). Em seu trabalho é demonstrado o uso de filtros casados e "des-casados" (*mismatched*), com códigos de Barker e *MPS (Minimum Peak sidelobe)* (com lóbulos laterais mínimos). Ótimos resultados são alcançados na relação S/R, e o uso do código de Barker produz uma performance superior que o uso de códigos muito longos (como os MPS). Isso é de extrema importância para aplicações nas quais existe um limite para o comprimento do sinal que será enviado, que é o caso deste trabalho como será visto na seção 4.1. Já Levanon (2005) faz a otimização de filtros "des-casados", com 3 vezes o comprimento inicial do sinal, para minimizar a integral ou o pico dos lóbulos laterais, utilizando sequências binárias longas.

Em (HIGUTI, 1994) já era chamada atenção para os problemas de paredes finas (em relação ao comprimento de onda) e a sobreposição dos ecos, propondo o uso de técnicas

de processamento digital de sinais para solucionar estes casos. Um dos métodos propostos (método da correlação cruzada) se assemelha ao método do filtro casado que será utilizado neste trabalho. Porém, Higuti (1994) manteve sua análise experimental limitada a placas com superfícies paralelas, não estudando casos com superfícies inclinadas (que é o caso de locais com alta corrosão).

Simon (1993) mostrou que materias com grãos em sua estrutura também causam efeitos de dispersão dos ecos. Esse efeito é tão maior quanto maior for a frequência do sinal (caso semelhante ao da rugosidade superficial estudado por Higuti (1994)). Também é proposta uma técnica de correlação cruzada, com geração de pulsos para excitação de transdutores, que não foi testada por falta de equipamento, mas que também se assemelha com a técnica que será utilizada neste trabalho.

Levando em conta a direção destes trabalhos, concluiu-se que apostar em inspeção não-destrutiva por ultrassom, com processamento digital de sinais utilizando filtro casado e códigos de Barker é o caminho mais interessante para tentar solucionar os problemas de ruídos e dúvidas/ambiguidades sobre o estado real de um duto.

3 Aspectos Teóricos

Neste capítulo serão abordados todos os aspectos teóricos envolvidos neste trabalho, de modo a explicar ao leitor alguns conceitos, além de demonstrar o porque do uso de certos algoritmos, através de suas formulações matemáticas.

3.1 Ultrassom

Nesta seção será apresentada a teoria envolvendo os sinais de ultrassom, de modo a introduzir ao leitor alguns termos e conceitos que serão utilizados ao longo deste trabalho, e explicar como o ultrassom é utilizado em uma inspeção não destrutiva.

3.1.1 Os sinais de Ultrassom

Sinais de ultrassom se comportam como qualquer outro sinal acústico, e por isso podem ter este comportamento muito bem previsto, já que a teoria de propagação acústica é bem consolidada. Em primeiro lugar, as ondas acústicas podem ser divididas em longitudinais (onde a movimentação de uma partícula é na direção de propagação da onda) e transversais (onde a movimentação das partículas é perpendicular à direção de propagação da onda) (KINO, 1987). A figura 3.1 exemplifica estes dois tipos de propagação. No geral, ondas acústicas acabam sendo uma combinação tanto de ondas transversais quanto longitudinais.

Em aplicações específicas, tais como transdutores, a construção do emissor de ondas pode ser feita de maneira que este emita apenas um dos tipos (transversal ou longitudinal). Transdutores para inspeções não destrutivas são feitos de modo a emitir apenas ondas longitudinais, porém, durante as inspeções podem ocorrer conversão de modos, e assim o aparecimento de ondas transversais.

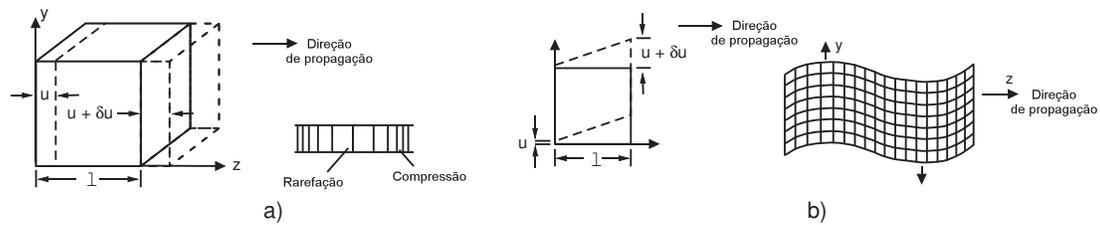


Figura 3.1: a) Propagação da onda longitudinal e b) Propagação da onda transversal

3.1.2 A reflexão das ondas

As ondas acústicas se propagam normalmente em um meio até que encontrem uma fronteira, isto é, uma divisão deste meio com outro. A divisão de meios do ponto de vista das ondas ocorre quando há uma mudança na impedância acústica. Assim como tem-se o conceito de impedância elétrica - a propriedade de um componente impedir mais ou menos a passagem de uma corrente elétrica, em acústica também define-se a impedância de um material como sendo a propriedade que ele possui em transmitir uma vibração acústica com maior ou menor facilidade. A impedância acústica de um material é definida como o produto da densidade do material pela velocidade das ondas no mesmo, $Z = \rho.v$.

A onda, ao encontrar uma diferença de impedâncias, será parcialmente refletida e parcialmente transmitida ao outro meio. A figura 3.2 exemplifica esta situação. As frações da onda que serão refletida e transmitida dependem das impedâncias acústicas dos meios (Z_1 e Z_2). O índice de amplitude refletida R e o de amplitude transmitida T são expressos por (KINSLER; FREY, 1950):

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad e \quad T = \frac{2Z_2}{Z_2 + Z_1} \quad (3.1)$$

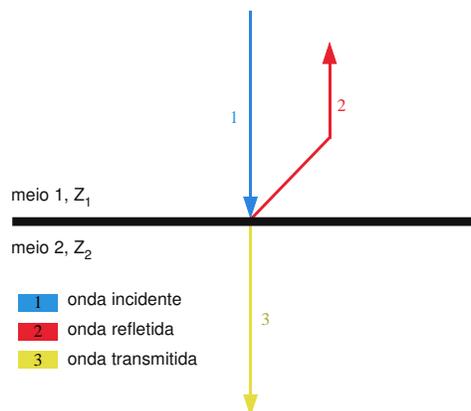


Figura 3.2: Reflexão e Transmissão de uma onda acústica

Pode-se perceber que se Z_1 e Z_2 forem muito próximos, praticamente toda a energia será transmitida (como se não houvesse mudança de meio). Além disso, mesmo que não haja uma troca de meios, as ondas vão perdendo intensidade durante sua propagação devido à expansão da frente de onda, que faz com que a intensidade decaia com o inverso da distância percorrida, e também devido à viscosidade do material. Essas propriedades são muito importantes para que se possa entender as dificuldades em uma inspeção não destrutiva por ultrassom.

Dependendo do meio que o transdutor for trabalhar, a diferença de impedâncias pode ser muito grande, e desta maneira pouca energia conseguirá ser transmitida, refletida, e recuperada para análise. Com isso, os ecos podem voltar com baixas amplitudes, e comprometidos por ruído. Na seção 3.1.3 isso será melhor explicado.

Por último, é importante dizer que quando a onda encontra uma divisão de meios, indo de um meio com impedância acústica maior para um meio com impedância menor, ocorre uma inversão de fase da mesma.

3.1.3 O TOA (*Time of Arrival*) e a inspeção por ultrassom

Uma inspeção não destrutiva por ultrassom consiste basicamente em enviar e receber sinais de ultrassom através de uma peça, e analisando-se o sinal recebido, tirar conclusões sobre o estado da mesma. No caso de inspeções de paredes de oleodutos, utilizam-se pulsos acústicos gerados por transdutores de ultrassom, que se refletem nas paredes internas do duto e são recebidos como sinais de eco, pelo mesmo ou por outros transdutores. Conhecendo-se a velocidade de propagação da onda no meio, e medindo-se o tempo entre o envio da onda e o recebimento do seu eco (conhecido como TOA) é possível determinar a distância entre o transdutor e a parede. Determinando-se estas distâncias seguidamente ao longo de um corpo pode-se reconstruir o relevo do mesmo. Além disso, é possível medir a espessura da parede, pela diferença de tempo entre o primeiro e o segundo eco.

O TOA é estimado analisando-se o sinal de eco recebido. Por exemplo, pode-se adotar o instante de maior amplitude do sinal como estimativa do TOA. A figura 3.3 mostra um exemplo de envio e recebimento dos ecos gerados pelas paredes, e seus respectivos TOAs (adotando-se um critério de maior amplitude como sendo o TOA). A propagação de ondas de ultrassom pode ser prevista com precisão em líquidos homogêneos e bem conhecidos como a água. Nesses meios, a reflexão das ondas em superfícies metálicas planas e polidas

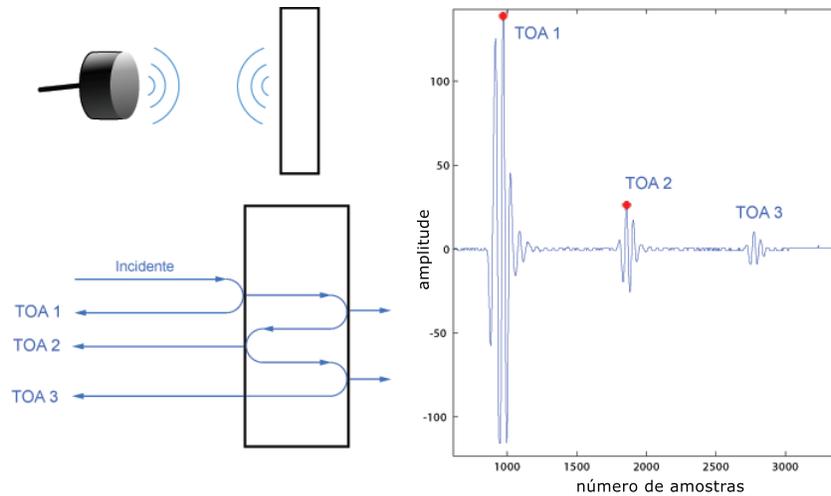


Figura 3.3: Envio e recebimento dos ecos do sinal de ultrassom, e seus respectivos TOAs

é praticamente especular, gerando sinais de eco intensos e bem comportados. Porém, quando tenta-se receber sinais de eco gerados por superfícies corroídas e rugosas, em um meio com alta atenuação, a relação S/R do eco recebido é fortemente prejudicada. A figura 3.4 mostra claramente a diferença de ecos obtidos em uma superfície plana em meio limpo, e os ecos misturados a ruído em superfícies corroídas, e meios altamente atenuantes (como o petróleo). Estes ecos com baixa taxa S/R (figura 3.4.(b)) se tornam difíceis de ser detectados corretamente.

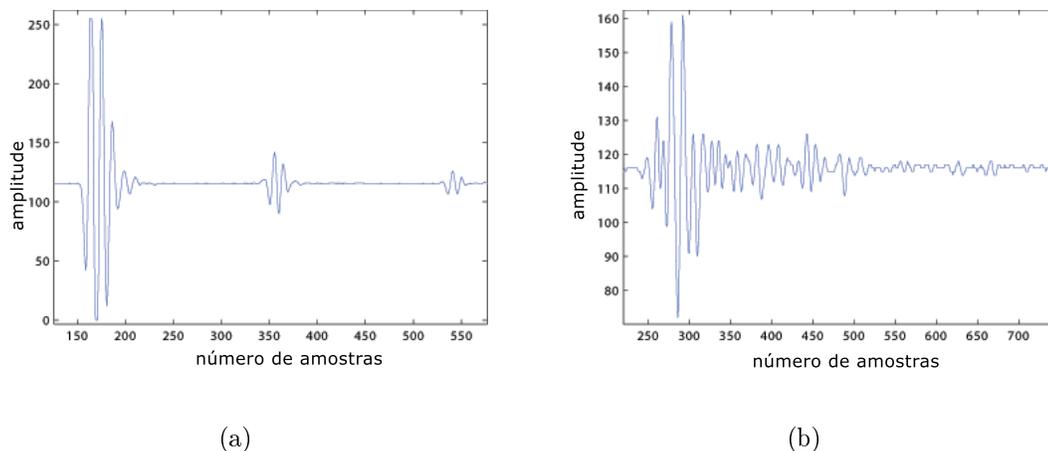


Figura 3.4: Sinais de eco obtidos em água, (a) gerado por uma placa de alumínio plana e polida; e (b) gerado em região corroída da placa

Além disso, quando uma onda ultrassônica encontra uma diferença de meios, e o ângulo de incidência não é normal à superfície, ocorre uma conversão de modos, isto é, uma

onda longitudinal se converte em uma longitudinal, mais uma componente transversal. Estas ondas transversais acabam também voltando ao transdutor, gerando mais incertezas na detecção dos ecos. Essas ondas são encaradas como ruído adicional aos que já foram citados.

Verifica-se portanto a necessidade do desenvolvimento de novas tecnologias, que melhorem essa relação S/R nos ecos recebidos, para que a estimativa do TOA possa ser feita com maior precisão, e desta maneira o resultado da inspeção tornar-se mais confiável.

Para aplicações em tubulações, pode-se calcular qual será a atenuação esperada do primeiro e do segundo eco, utilizando os índices de reflexão e transmissão definidos na equação 3.1. Para a amplitude do primeiro eco, aplica-se apenas a equação de reflexão. Já para o segundo eco, aplica-se a de transmissão (do meio em que se encontra o transdutor para a primeira parede), a de reflexão (na segunda parede), e novamente a de transmissão (da primeira parede de volta para o meio).

Para a impedância acústica do óleo pode-se utilizar uma aproximação com o mesmo valor da impedância acústica da água, que é de 1.48 MRayl , e para a parede do oleoduto usa-se a do aço que é de 40 MRayl (KINSLER; FREY, 1950).

Dessa forma, a amplitude do primeiro eco será 93% da amplitude inicial emitida, e a do segundo eco será 13%.

Por último, é importante ressaltar que devido à inversão de fase que ocorre no segundo eco, deve se ter em mente que se procura-se um pico positivo no primeiro eco, deve-se procurar um pico negativo no segundo.

3.1.4 Transdutores de Ultrassom

Para melhor entender todo o conceito da inspeção por ultrassom, é interessante compreender como funciona o transdutor ultrassônico, que é a peça responsável pelo envio e recebimento dos sinais de ultrassom.

Os transdutores são feitos de material piezoelétrico (material que converte tensão mecânica em potencial elétrico, ou vice-versa) e podem ter diferentes formatos, tamanhos e frequências. A figura 3.5 mostra como seria a configuração mais simples possível para um transdutor. Ele é composto basicamente do material piezoelétrico e de dois eletrodos como terminais elétricos.

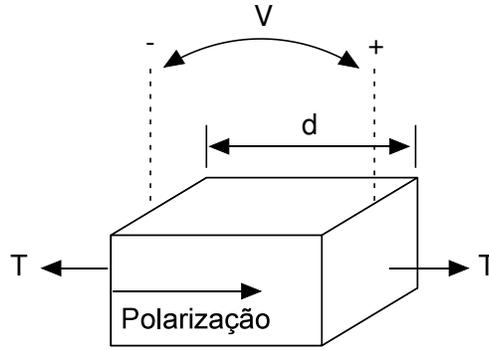


Figura 3.5: Modelo simplificado de um transdutor de ultrassom

Utilizando as equações constitutivas piezoelétricas (SZABO, 2004), chega-se a conclusão de que quando uma tensão elétrica é aplicada nos terminais do transdutor, uma força aparece no corpo, e seu valor é,

$$F(t) = (hC_0V/2)[-δ(t) + δ(t - d/c)], \quad (3.2)$$

onde h é a constante piezoelétrica, C_0 é a capacitância, V é a tensão aplicada, d é a distância entre os terminais, e c a velocidade do som entre os eletrodos (dada por $c = \sqrt{C^D/\rho}$, onde C^D é constante de rigidez e ρ é a densidade do material. Tomando a transformada de Fourier da equação 3.2 temos,

$$F(f) = -i(hC_0V)e^{-i\pi fd/c} \sin[\pi(2n+1)f/2f_0]. \quad (3.3)$$

A equação 3.3 mostra que a força é máxima nas harmônicas ímpares ($n = 1, 2, 3, \dots$) da frequência fundamental (ou de ressonância) $f_0 = c/2d$. Daí vem a grande importância de excitar os transdutores em sua frequência de ressonância (e suas harmônicas), de modo a transmitir a maior energia possível. Fora destas frequências, pouquíssima energia será transmitida, impossibilitando a geração e envio de sinais de ultrassom.

No entanto, sinais com banda base de baixa frequência também podem ser usados como excitação utilizando-se uma onda portadora na frequência de ressonância. Esta técnica é utilizada neste trabalho, onde uma onda quadrada portadora na frequência de ressonância do transdutor é modulada em fase com um código binário.

3.1.4.1 Frequência do transdutor

Para escolher a frequência base do transdutor, deve-se levar em consideração dois fatores principais:

- Espessura a ser medida - Quanto menor a espessura que se espera medir, menor deve ser o comprimento de onda do sinal ultrassônico (e conseqüentemente, maior sua frequência), de modo que os ecos provenientes da inspeção não se sobreponham. Isto é conhecido como a resolução axial do transdutor, ou seja, qual a medida mínima de espessura que pode-se distinguir utilizando o mesmo.
- Atenuação do sinal - A diferença de impedâncias acústicas dos meios, somada a atenuações por ruído, dispersão, rugosidade e corrosão tornam a detecção dos ecos difíceis por camuflar os ecos muito atenuados. De modo geral, quanto maior a frequência, maior a atenuação.

Nas aplicações de inspeções de dutos, espera-se medir espessuras de até 5 *mm*. A norma SAE AMS 2632 recomenda que para espessuras menores de 13 *mm* utilize-se frequências entre 2 e 10 *MHz* (SAE AMS 2632, 1974).

Em relação ao item atenuação, sabe-se que para maiores frequências, as ondas ultrassônicas sofrem maior atenuação quanto maior o espaço percorrido. Além disso, o estudo da rugosidade superficial feito por Higuti (1994) concluiu que a rugosidade da peça que está sendo inspecionada também age de forma diferente para diferentes comprimentos de onda (ou seja, para diferentes frequências). A rugosidade age principalmente diminuindo a amplitude do sinal recebido (espalhando a energia), e essa atenuação é maior quanto maior for a frequência do sinal. Como também já foi citado (SIMON, 1993), o mesmo acontece para materiais com granulação em sua estrutura.

Desta maneira tem-se uma relação de compromisso para escolher uma frequência alta (com um comprimento de onda baixo), que resulta em uma resolução axial maior (para detecção de paredes menos espessas), porém que tem uma atenuação maior devido à rugosidade, corrosão e percurso da onda.

Por estes motivos, decidiu-se utilizar a frequência base de 5 *MHz*, tomando uma média destes fatores.

3.1.4.2 Transdutores focais

Também existe a possibilidade de utilizar transdutores focalizados. A figura 3.6 mostra exemplos de um transdutor normal e um focalizado.

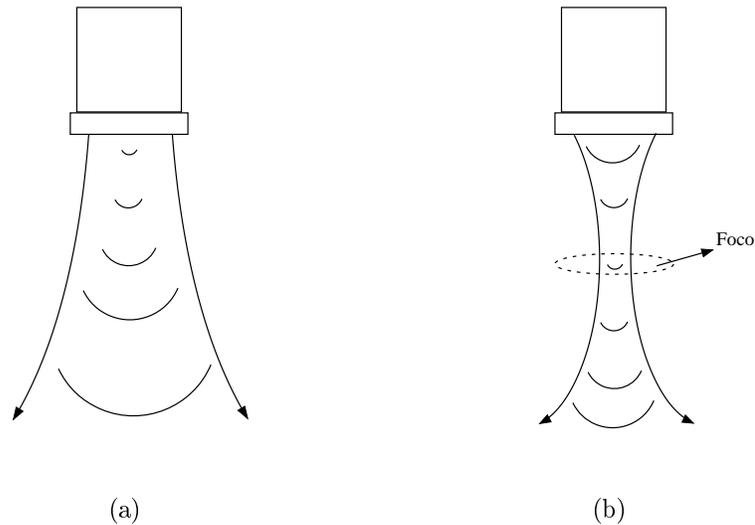


Figura 3.6: Em (a) um transdutor não-focalizado, e em (b) um transdutor focalizado

O transdutor focalizado concentra o feixe acústico emitido, o que poderia beneficiar a localização exata de pontos corroídos. Porém, como as superfícies corroídas apresentam perfis irregulares e ligeiramente inclinados, pode ocorrer o espalhamento na reflexão do sinal. Desta maneira não é indicado o uso de transdutores focalizados, pois esse espalhamento pode refletir o eco para uma região não coberta pelo transdutor. Mesmo perdendo um pouco de resolução lateral com o feixe não diretivo, a chance de receber energia refletida se torna maior com um transdutor não focalizado. A figura 3.7 demonstra como o espalhamento do sinal é prejudicial no caso do transdutor focal. Por este motivo foi escolhido utilizar transdutores não-focalizados.

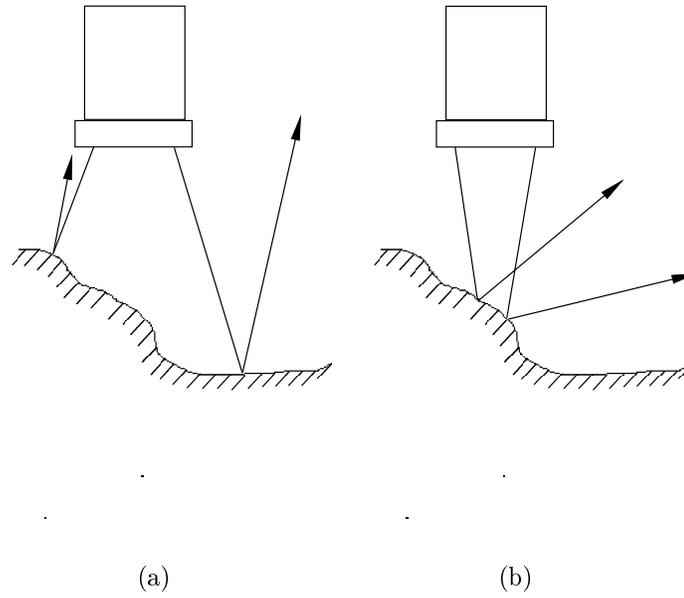


Figura 3.7: Em (a) o transdutor não focal está dentro do campo de ondas refletidas por uma superfície irregular, o que não acontece em (b) com o transdutor focalizado

3.2 Processamento digital de sinais

Nesta seção serão descritas as razões do uso do processamento digital de sinais. O processamento de sinais consiste na análise e/ou modificação de sinais de forma a extrair informações dos mesmos e/ou torná-los mais apropriados para alguma aplicação específica. Este processamento pode ser feito tanto analogicamente como digitalmente. A grande vantagem de um processamento digital de sinais (*DSP - Digital Signal Processing*) é que se pode empregar algoritmos muito mais complexos e eficientes que em um processamento analógico. O DSP começou em meados dos anos 70, quando os computadores começaram a ficar disponíveis, porém eram extremamente limitados a aplicações específicas devido aos altos custos envolvidos. A partir destes primórdios, avanços enormes foram conseguidos, tornando o DSP uma ferramenta indispensável em todas as áreas que envolvem qualquer tipo de aquisição de sinais (SMITH, 1997).

Basicamente qualquer algoritmo pode ser implementado digitalmente, já que são inúmeras as possibilidades de desenvolvimento de um circuito digital. Ele pode ser implementado de diversas maneiras: via software com processadores dedicados, com processadores normais, ou até com circuitos digitais eletrônicos totalmente dedicados. Isso explica sua atual importância, já que para qualquer necessidade e aplicação, é possível criar um sistema que as atenda.

3.2.1 Sistema Embarcado - FPGA

No caso das inspeções com pigs, deseja-se que todo o sistema desenvolvido esteja embarcado no mesmo (para que ele possa correr os dutos sem necessidade de estar ligado a uma base fixa). Por isso é necessário reduzir o consumo de energia e a quantidade de dados a ser armazenada. Assim o desenvolvimento de um sistema totalmente dedicado e que realize o processamento em tempo real é necessário.

Nesta fase de estudo e pesquisa, desenvolver diversos circuitos dedicados consumiria muitos recursos financeiros, além de muito tempo. Uma maneira de resolver estes problemas, e atender as exigências do sistema é a criação de um circuito digital em uma FPGA. A primeira FPGA foi desenvolvida em 1985 pela Xilinx. Estas são dispositivos em silício formados por blocos lógicos (que implementam qualquer função digital, através de memórias), matrizes de interconexão (que roteiam as ligações entre os blocos lógicos), e blocos de entrada e saída para conexão com o ambiente externo (XILINX, 2008). A figura 3.8 mostra um exemplo simplificado desta estrutura.

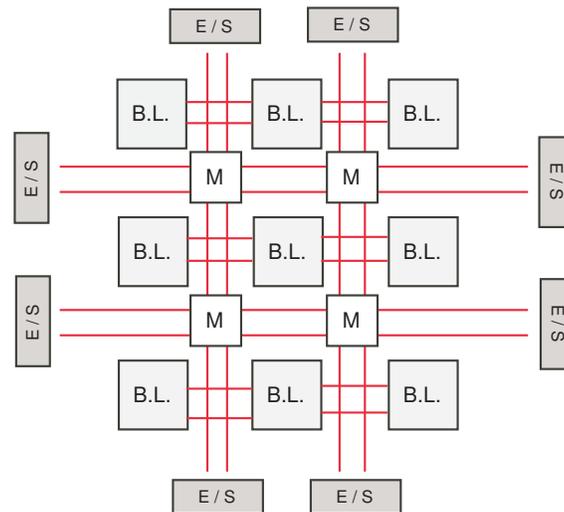


Figura 3.8: Estrutura interna simplificada de uma FPGA. E/S - blocos de entrada e saída, B.L. - blocos lógicos e M - matrizes de interconexão

Em uma FPGA pode ser implementado qualquer circuito lógico, por meio da programação e re-programação das memórias, dos blocos lógicos e das matrizes de interconexão, o que traz grandes benefícios no desenvolvimento de um projeto como esse, onde os algoritmos implementados têm que ser mudados constantemente. Assim consegue-se economizar muito tempo, já que o protótipo pode ser mudado com facilidade e rapidez.

3.2.2 Processamento Digital

Para se realizar um DSP, primeiramente deve-se converter o sinal adquirido para uma versão digital (já que qualquer sinal na natureza é analógico, existindo em tempo contínuo). No caso de inspeção por ultrassom, como a frequência do sinal é relativamente alta (cerca de 5 MHz) é necessário uma conversão analógica/digital muito rápida. Por exemplo, neste trabalho será utilizada uma taxa de aquisição de cerca de 50 MS/s, para proporcionar uma resolução adequada.

Além disso, como os percursos de inspeção são longos, a quantidade de dados que deveriam ser armazenados é muito grande, tornando inviável salvar todos os dados adquiridos. Para resolver este problema é proposto realizar o DSP com um hardware dedicado, isto é, um circuito que trabalhe apenas fazendo o processamento, e em tempo real, tornando necessário salvar não os sinais completos, mas somente a informação desejada (no caso o TOA do primeiro eco e a espessura da parede em cada ponto).

3.3 Algoritmos matemáticos

Nesta seção será apresentada toda a teoria envolvida para o desenvolvimento do algoritmo que será implementado, desde os conceitos básicos de representação matemática do sinal, até a descrição do algoritmo em si.

3.3.1 Estimação do TOA

Como foi visto na seção 3.1.3, o problema básico na inspeção ultrassônica é a estimação do tempo de chegada dos ecos do pulso ultrassônico. O modelo matemático simplificado do sinal recebido pode ser descrito como:

$$r(t) = s(t - A) + w(t), \text{ com } 0 \leq t \leq T \quad (3.4)$$

onde,

- $r(t)$ - sinal observado
- $s(t)$ - sinal conhecido (já com a atenuação devido à primeira reflexão, calculada na seção 3.1.2)

- A - instante de chegada (TOA)
- $w(t)$ - ruído branco gaussiano
- T - tempo de observação do sinal

Pela equação 3.4 pode-se perceber que o sinal recebido é basicamente composto de um sinal conhecido (que é o sinal de interesse para cálculo do TOA) presente a partir de um certo instante (no caso, o TOA que procuramos), somado de um ruído branco *WGN* (*White Gaussian Noise*). O WGN é uma boa aproximação de situações do mundo real, pois é um sinal aleatório, cujo espectro de potência é plano, isto é, tem igual intensidade para qualquer frequência, e suas amostras em qualquer instante do tempo tem distribuição gaussiana com média nula.

A figura 3.9 demonstra como a adição de ruído consegue prejudicar a estimativa do TOA, ou seja, a detecção do eco dentro do sinal recebido.

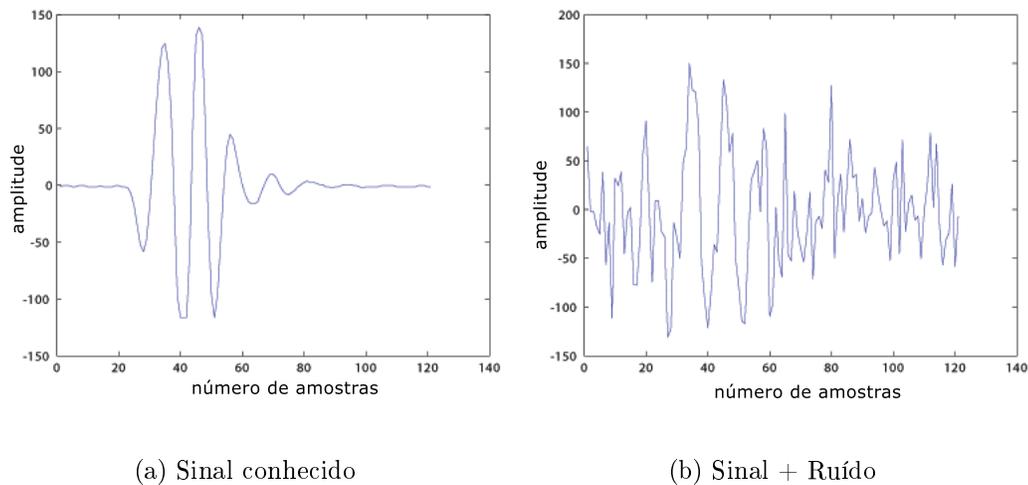


Figura 3.9: Em (a) o sinal conhecido, e em (b) com adição de ruído branco, dificultando a determinação do TOA

O problema se torna estimar o valor do instante de chegada A em meio ao sinal observado $r(t)$ a partir de um modelo $s(t)$ do sinal conhecido com o menor erro possível. No caso deste trabalho temos limitação de tempo de processamento para o algoritmo, já que como foi explicado na seção 3.2.1, o processamento deverá ser feito em tempo real com a inspeção, gerando os resultados ao longo do percurso dos pigs. Por isso foi escolhido o uso de um algoritmo que utiliza o filtro casado, e códigos de Barker, de maneira semelhante

ao desenvolvido por Lima e Furukawa (2002), pois este filtro pode ser implementado como um filtro *FIR*, gerando respostas com um atraso fixo.

3.3.2 Filtro Casado

A abordagem geralmente utilizada para se estimar um parâmetro determinístico (no caso o TOA) através de observações aleatórias é conhecido como estimação de máxima verossimilhança (*ML - Maximum Likelihood*). Supõe-se que se queira estimar uma variável X , a partir de uma grandeza u que foi observada. Matematicamente isso corresponderia a encontrar um valor de X , para um u dado, que maximizasse a função densidade de probabilidade (*PDF - Probability Density Function*) condicional $p_{u|X}(u|X)$ (LIMA, 2002).

O sinal $r(t)$ pode ser descrito como uma aproximação de K coeficientes utilizando um conjunto de funções arbitrárias ortonormais $\phi_i(t)$, tal que,

$$r(t) = \sum_{i=1}^K r_i \phi_i(t), \quad 0 \leq t \leq T, \quad (3.5)$$

$$r_i = \int_0^T r(t) \phi_i(t) dt, \quad i = 1, 2, \dots, K, \quad (3.6)$$

Aplicando a decomposição da equação 3.5 na formulação geral do sinal, mostrada na equação 3.4, chega-se que suas componentes serão

$$r_i = \int_0^T s(t - A) \phi_i(t) dt + \int_0^T w(t) \phi_i(t) dt = s_i + w_i, \quad (3.7)$$

Pode-se demonstrar que as componentes w_i do ruído são variáveis aleatórias gaussianas com média nula e variância igual ao valor quadrático médio do ruído. Além disso, essas componentes são variáveis aleatórias estatisticamente independentes entre si (devido ao fato do ruído ser branco) (TREES, 1968). Assim, para um dado valor de A , a PDF de cada componente r_i será

$$p_{r_i|A}(R_i|A) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{1}{2} \frac{[R_i - s_i(A)]^2}{\sigma_n^2}\right), \quad (3.8)$$

onde σ_n^2 é a variância do ruído e $s_i(A)$ é a componente de $s(t-A)$ para um dado valor de A , que corresponde ao valor médio da variável r_i .

Como as componentes r_i são estatisticamente independentes, a PDF do sinal pode ser escrita como a produtória das PDFs das componentes

$$p_{r|A}(R|A) = \prod_{i=1}^A \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{1}{2} \frac{[R_i - s_i(A)]^2}{\sigma_n^2}\right), \quad (3.9)$$

A estimação ML é obtida calculando-se a equação 3.9 para um dado sinal $r(t)$ recebido, encontrando-se o valor de A que maximiza a mesma.

Por se tratar de um problema de maximização, a equação pode ser simplificada com os seguintes passos:

1. Encontrar A que maximiza a expressão 3.9 é o mesmo que encontrar A que maximiza o logaritmo da expressão. Aplicando o logaritmo e executando a produtória tem-se

$$K \cdot \log\left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right) - \frac{1}{2\sigma_n^2} \sum_{i=1}^K (R_i^2 - 2R_i s_i(A) + s_i^2(A)) \quad (3.10)$$

2. Como σ_n^2 não depende de A , pode-se simplesmente procurar um A que **minimize** a expressão

$$\sum_{i=1}^K R_i^2 - \sum_{i=1}^K 2R_i s_i(A) + \sum_{i=1}^K s_i^2(A) \quad (3.11)$$

3. Mas

$$\sum_{i=1}^K R_i s_i(A) = \sum_{i=1}^K R_i \int_0^T s(t-A) \phi_i(t) dt = \int_0^T (s(t-A) \sum_{i=1}^K R_i \phi_i(t)) dt = \int_0^T r(t) s(t-A) dt \quad (3.12)$$

e de maneira semelhante,

$$\sum_{i=1}^K R_i^2 = \int_0^T r^2(t) dt, \quad (3.13)$$

$$\sum_{i=1}^K s_i^2 = \int_0^T s^2(t-A) dt. \quad (3.14)$$

4. Assim, a expressão do item 2 é equivalente a

$$\int_0^T r^2(t)dt - 2 \int_0^T r(t)s(t-A)dt + \int_0^T s^2(t-A)dt. \quad (3.15)$$

5. Porém, para um dado $r(t)$, temos que a primeira e a última integral da equação 3.15 correspondem respectivamente às energias dos sinais $r(t)$ e $s(t)$, e não dependem do valor de A no modelo adotado. Logo, a estimativa de ML é o valor de A que **maximiza** a expressão

$$\Lambda(r(t), A) = \int_0^T r(u)s(u-A)du \quad (3.16)$$

Assim, o problema de maximização de $p_{r|A}(r|A)$ foi reduzido a maximizar a função $\Lambda(r(t), A)$. Esta equação é conhecida como função de verossimilhança.

A equação 3.16 poder ser vista como uma integral de convolução. Para que isso seja possível, basta enxergá-la como sendo a saída de um sistema linear com entrada $r(t)$ e resposta impulsiva $s(-t)$ (que nada mais é que o sinal de interesse revertido no tempo). A figura 3.10 exemplifica a saída de um sistema linear.

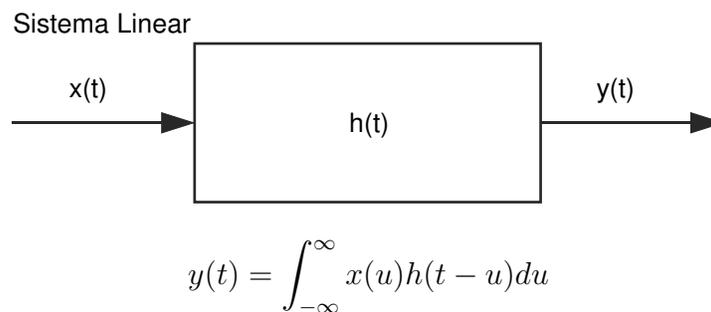


Figura 3.10: Exemplo da saída de um sistema linear

Em suma, um receptor capaz de fornecer estimativas do TOA de um sinal pode ser um sistema linear, cuja resposta impulsiva é o sinal desejado revertido no tempo. Um receptor deste tipo é conhecido como **filtro casado**. A estimativa é obtida detectando-se o pico de saída do filtro, o que é equivalente a calcular a função de verossimilhança para vários valores de A e tomar o máximo. Uma observação pertinente no momento é que numa aplicação real não é possível implementar um filtro com resposta $s(-t)$ (pois seria um filtro não-causal). Portanto, o que se faz é utilizar uma resposta do tipo $s(-t + t_f)$, onde t_f é a duração do sinal. Desta maneira as estimativas do TOA são feitas sempre

com um atraso fixo e conhecido.

Por definição, um filtro casado a um sinal $s(t)$ qualquer é aquele cuja função de transferência (ou resposta impulsiva) é

$$h(t) = ks(t_f - t), \quad (3.17)$$

onde k é uma constante arbitrária, e t_f como dito é a duração (o comprimento, no caso de tempo discreto) do sinal. Ou seja, o filtro é basicamente o sinal inicial invertido no tempo, atrasado, e escalado. A figura 3.11 mostra um exemplo de um sinal, e um possível filtro casado ao mesmo.

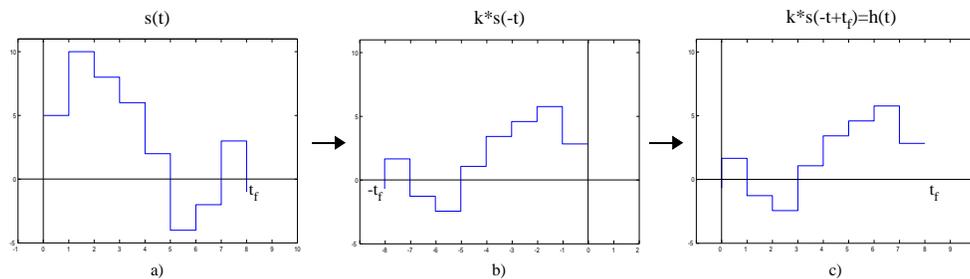


Figura 3.11: Em a) tem-se o sinal original, em b) sua inversão no tempo e escalado por um fator k , e em c) aplicando um atraso A tem-se a função de transferência do filtro causal

Utilizando o mesmo sinal da figura 3.11 e seu correspondente filtro casado, é mostrado na figura 3.12 este sinal atrasado e envolvido em ruído, e a saída depois de uma filtragem. Pode-se perceber o pico da saída, que é a melhor estimativa para o atraso A mostrado na figura 3.12.

Na figura 3.12 pode-se perceber que o pico ocorre com um atraso igual ao comprimento do sinal utilizado para gerar o filtro (t_f). Como explicado anteriormente, isso acontece pelo fato do filtro ter que ser causal (para que possa ser implementado fisicamente). Porém este atraso sempre será fixo e conhecido.

Para determinar-se corretamente este atraso então, espera-se que a saída do filtro seja um pulso o mais estreito possível centrado no atraso A (TURIN, 1960). Porém, dependendo dos sinais utilizados, muitas ambiguidades podem aparecer, por exemplo devido a onda portadora (que é necessária na excitação de transdutores, como foi descrito na seção 3.1.4). Uma grandeza utilizada para medir qualitativamente essa ambiguidade é a função de ambiguidade, que no caso da estimação do TOA em um sinal $s(t)$ é expressa por sua

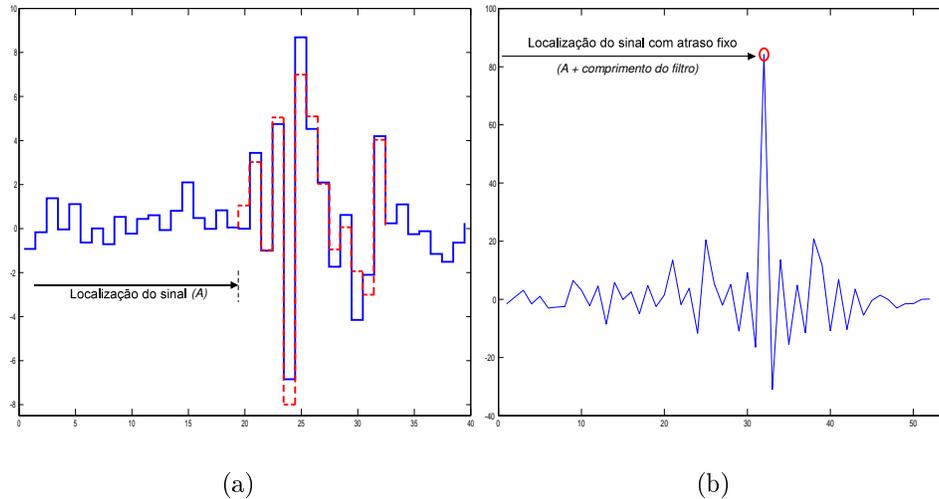


Figura 3.12: Em (a) o sinal em meio a ruído e com um atraso desconhecido (o tracejado mostra o sinal original) e em (b), a saída do filtro casado, com o pico mostrando qual o valor de A que maximiza a função, e determinando assim a localização do sinal original no sinal recebido.

autocorrelação (LIMA, 2002),

$$\Phi(\tau) = \int_{-\infty}^{\infty} s(t - \frac{\tau}{2})s(t + \frac{\tau}{2})dt \quad (3.18)$$

Portanto, é desejável trabalhar com sinais que tenham picos acentuados em sua autocorrelação.

3.3.3 Códigos de Barker

Um sinal de ultrassom não modulado não tem uma autocorrelação como a desejável. Para melhorar a autocorrelação utiliza-se a modulação, que consiste basicamente em variar a amplitude, frequência, intensidade, comprimento e/ou fase de uma onda portadora. Uma das alternativas para isso, é a modulação *BPSK* (*Binary Phase Shift Keying*), ou modulação binária por chaveamento de fase. Neste tipo de modulação, a fase de uma onda portadora é chaveada (em 0° ou 180°) de acordo com uma sequência pré-definida. A figura 3.13 exemplifica como essa modulação funciona, utilizando-se um sinal senoidal ideal.

Como dito na seção 3.1.4, os transdutores devem ser excitados em sua frequência de ressonância, por isso a onda portadora será uma onda com a mesma frequência que a do

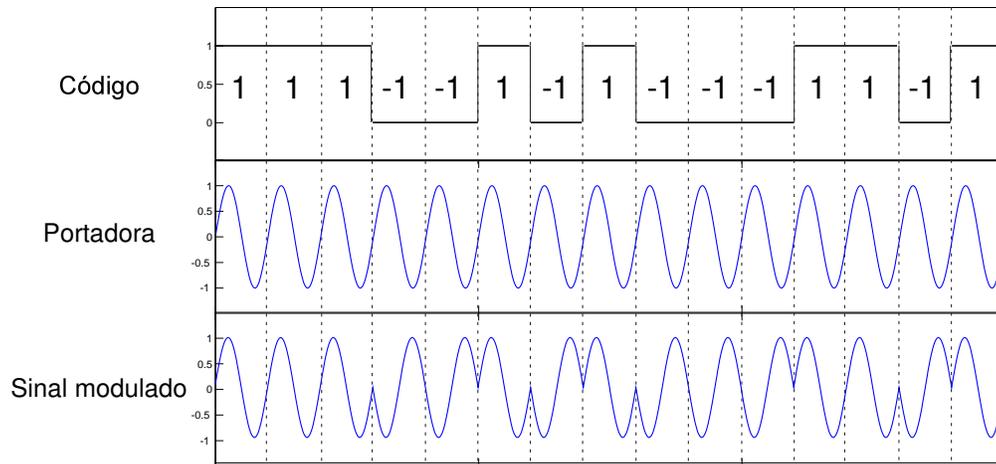


Figura 3.13: Exemplo de modulação BPSK. A onda portadora é defasada conforme o código binário (0° para código 1 ou 180° para código -1)

transdutor escolhido.

Na seção 3.3.2 foi explicado que para o filtro casado é desejável o uso de uma função (no caso, o sinal expresso matematicamente) que tenha uma função de autocorrelação com a menor ambiguidade possível. No caso de sequências binárias, temos as seguintes condições para isso (TREES, 1971):

- O pico central deve ser estreito em relação ao eixo do tempo. A amplitude do pico é diretamente relacionada ao comprimento do sinal.
- O pico central deve ser estreito em relação ao eixo de frequências.
- Fora da região do pico central, a função de autocorrelação deve ser relativamente plana.
- O volume da função (integral da mesma no espaço tridimensional) deve ser igual a 1.

A figura 3.14 mostra o gráfico simplificado do qual a função de autocorrelação deve se aproximar.

Barker (1953) desenvolveu sequências que satisfizeram estas condições para alguns comprimentos $N \leq 13$. Os códigos de Barker de N bits são sequências de N valores a_j tais que:

- $a_j = \pm 1$, para $j = 1, \dots, N$

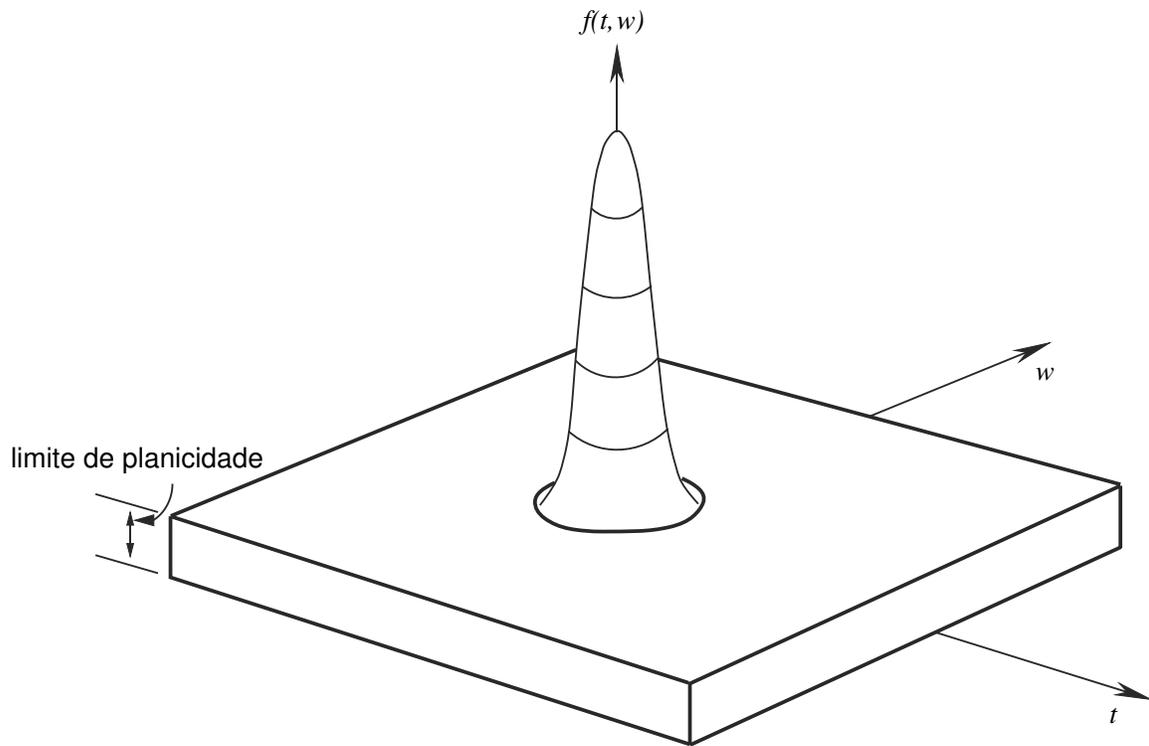


Figura 3.14: Uma função de correlação ideal

- $\left| \sum_{j=1}^{N-v} a_j a_{j+v} \right| \leq 1$, para todo $1 \leq v \leq N$

Atualmente são conhecidos 7 códigos de Barker, que são listados na tabela 3.1, e dois exemplos podem ser vistos na figura 3.15. Infelizmente não foram achados códigos de Barker com $N \geq 13$. Pode ser provado que não existem sequências ímpares maiores que 13, e nenhuma sequência par foi achada com $4 \leq N \leq 6084$ (TURYN, 1963; LUENBERGER, 1963).

Tabela 3.1: Códigos de Barker

Número de Bits	Código
2	+1 -1
3	+1 +1 -1
4	+1 -1 +1 +1
5	+1 +1 +1 -1 +1
7	+1 +1 +1 -1 -1 +1 -1
11	+1 +1 +1 -1 -1 -1 +1 -1 -1 +1 -1
13	+1 +1 +1 +1 +1 -1 -1 +1 +1 -1 +1 -1 +1

O resultado da autocorrelação de um sinal desse tipo é um sinal com um pico centralizado de amplitude N , e lóbulos laterais todos com amplitude ≤ 1 . A figura 3.16 mostra

o resultado da autocorrelação dos mesmos sinais mostrados na figura 3.15.

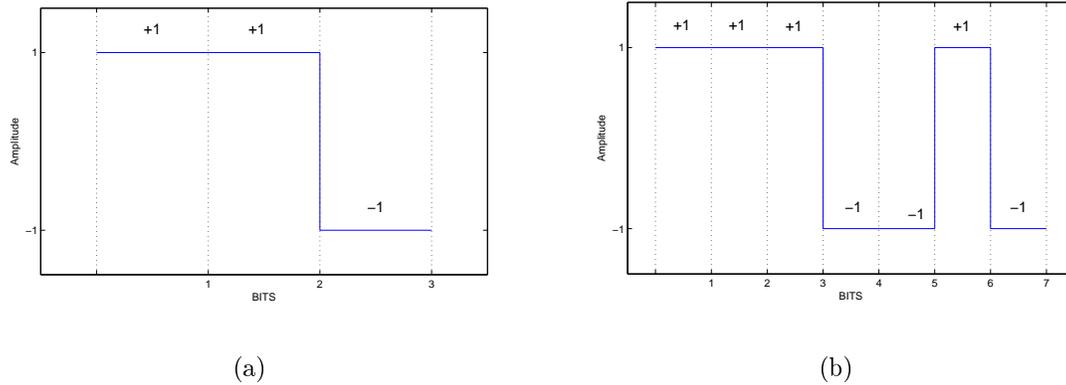


Figura 3.15: Código de Barker com (a) 3 bits; e (b) 7 bits

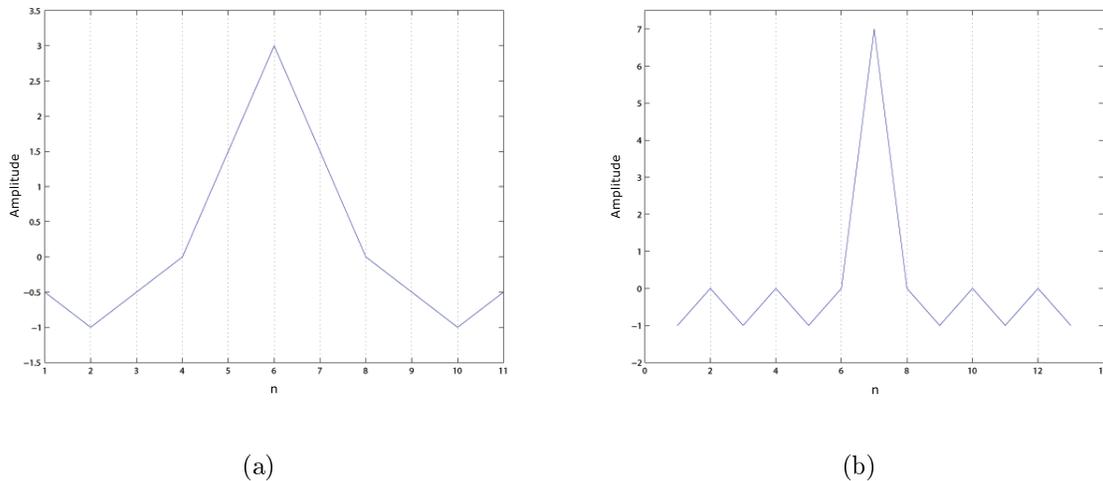


Figura 3.16: Auto correlação dos códigos de Barker com (a) 3 bits; e (b) 7 bits

Quando se transmite um sinal utilizando estes códigos, e o mesmo é alterado devido a ruídos no meio, pode-se tentar localizá-lo em meio ao ruído convoluindo-o com sua versão original (o próprio sinal transmitido, e no caso do filtro casado, invertido no tempo). A figura 3.17 exemplifica esta situação.

Para melhorar ainda mais a relação S/R deve-se transmitir o máximo de energia possível. Quanto maior a duração da onda transmitida, maior a energia transmitida, e consequentemente maior será a energia recebida dos ecos. Para isso pode-se transmitir os códigos de Barker com mais de um ciclo a cada bit, ou seja, cada bit é representado por

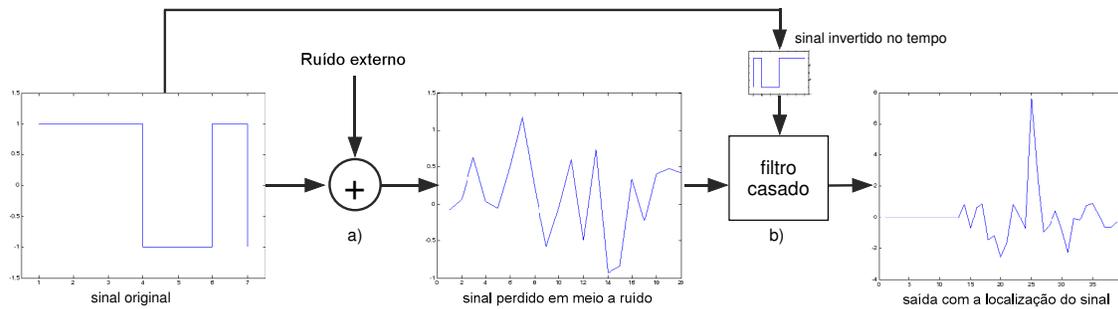


Figura 3.17: Sinal enviado sofre interferências do meio em a), e depois é correlacionado consigo mesmo em b), resultando em um pico distinto mostrando a posição do sinal, com um atraso fixo

mais de um ciclo da onda portadora. A figura 3.18 mostra um exemplo de sinal contendo uma onda portadora de 5 MHz, modulada em BPSK com um código de 3 bits (no caso +1, -1, +1). Neste exemplo, cada bit corresponde a 2 ciclos da onda portadora. Na seção 4.2.2 serão testadas as melhores combinações de códigos e números de bits para serem utilizados neste trabalho.

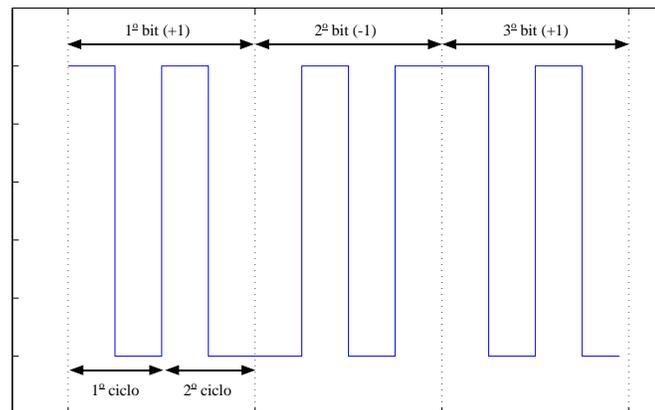


Figura 3.18: Exemplo de onda de excitação com 3 bits, 2 ciclos por bit

3.3.4 Comparação do erro entre duas medidas

No capítulo 6 serão apresentadas comparações entre os resultados de uma inspeção utilizando o algoritmo criado neste trabalho, e uma inspeção feita por uma máquina de medição 3D.

Para comparar a proximidade destes resultados será utilizado um método matemático conhecido como covariância.

3.3.4.1 Covariância

A covariância mede a dependência linear entre duas variáveis aleatórias. No caso do resultado das inspeções, lidar-se-a com duas linhas num espaço bidimensional, e a covariância estará dizendo o quanto essas linhas são "semelhantes".

A covariância entre duas variáveis aleatórias reais X e Y , com valores esperados $E(X) = \mu_X$ e $E(Y) = \mu_Y$ é definida como uma medida de como duas variáveis variam conjuntamente:

$$\text{cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y)),$$

onde E é o operador do valor esperado. Isto equivale à seguinte fórmula, a qual é geralmente usada para fazer os cálculos:

$$\text{cov}(X, Y) = E(XY) - E(X)E(Y)$$

Considerando o perfil das paredes do duto como uma variável aleatória no espaço, pode-se utilizar a covariância para medir o nível de "igualdade" entre os resultados. No caso deste trabalho, quanto maior for o coeficiente, melhor é o resultado da inspeção.

3.4 Os problemas da inspeção por ultrassom

Agora que já foram descritos os métodos que serão utilizados, pode-se entender mais facilmente quais são os problemas nas inspeções, e como eles afetam a mesma. O objetivo se resume em estimar corretamente o TOA. Os principais problemas na estimação ocorrem devido a ruídos presentes no sinal. Estes ruídos podem ser:

- Externos - Ruídos do sistema eletrônico, ruídos causados por impurezas nos meios, ou até ruídos mecânicos.
- Internos - Ecos de ondas longitudinais e transversais que aparecem nas reflexões em pontos adjacentes ao ponto inspecionado.

Além destes ruídos, existe a atenuação que o sinal vai sofrendo em suas reflexões, como já explicado anteriormente. Somado a isso, no caso de superfícies corroídas, pode ocorrer dispersão da onda incidente, e presença de ecos laterais, como pode ser visto nas figuras 3.19.a e 3.19.b, respectivamente. Essa dispersão ocorre pela corrosão, rugosidade e granulação, como já citado anteriormente.

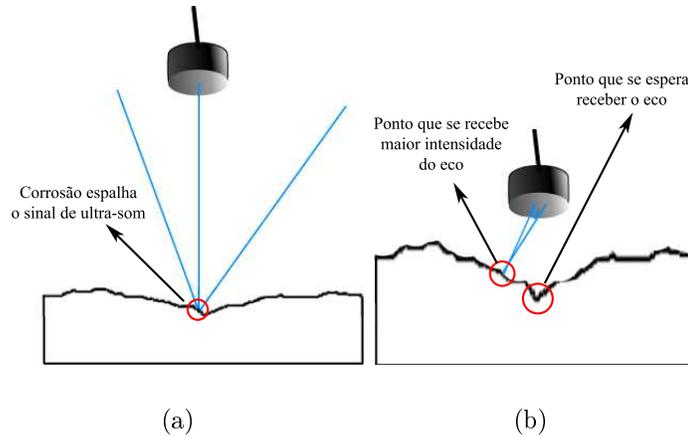


Figura 3.19: Em (a), a dispersão que pode ser causada pela corrosão, e em (b) um caso onde um eco lateral chega no transdutor antes do eco do ponto abaixo do mesmo

O eco lateral ocorre pelo fato da onda de ultrassom se propagar para todos os lados. Desta maneira, a distância do transdutor até o pico lateral pode ser menor que até o ponto exatamente em baixo do mesmo (que é o ponto que se espera medir em cada insonificação), fazendo com que o eco proveniente do pico lateral volte antes, dando um resultado de perfil maior que o verdadeiro para o ponto em questão.

Finalmente, existe o problema de sobreposição dos ecos, no caso de espessuras muito pequenas, isto é, antes do primeiro eco acabar, o segundo eco também chega, se sobrepondo ao primeiro.

O algoritmo proposto neste trabalho visa principalmente resolver os problemas dos ruídos. Não se conhece um algoritmo que resolva todos os problemas de inspeções, e estes devem ser estudados e abordados de maneiras diferentes. Os próprios estudos citados até este momento mostraram que não existe essa solução única (como em Higuti (1994) e Simon (1993))

Ainda assim, a abordagem proposta consegue resolver em partes os problemas de dispersão, e de sobreposição. O de dispersão é resolvido enviando uma onda ultrassônica de grande abertura (transdutor não focalizado), de modo que algum eco, por menor que seja, volte ao transdutor, e possa ser encontrado com o filtro casado. O filtro também consegue distinguir sinais sobrepostos até aproximadamente 50%, como será visto no capítulo 5.

3.4.1 Critério

Nesta abordagem, trabalha-se com o critério de que a detecção do primeiro eco deve ser a mais perfeita possível. Se esta falhar, muito provavelmente a do segundo eco também falhará. Porém a detecção correta do primeiro eco já produz resultados importantes para a inspeção, já podendo-se estimar o nível de corrosão do local, mesmo com uma detecção falha do segundo. Com os testes realizados, que serão descritos no capítulo 5, percebeu-se que a detecção do primeiro eco consegue atingir níveis altos de acerto, e a partir deste já consegue-se deduzir se um duto está altamente corroído ou não.

Para todo o desenvolvimento do trabalho, certos métodos serão escolhidos, e certas escolhas serão feitas, caso contrário, seria inviável chegar a algum resultado. Cada decisão tomada aqui pode abrir um novo leque de estudos a serem feitos, e a abrangência é muito grande para que se possa cobrir todas as possibilidades. Portanto este trabalho também serve como um guia para que outros trabalhos possam ser feitos (por exemplo, um estudo sobre qual o transdutor ideal para ser utilizado neste, ou em outros casos).

Finalmente, vale citar que os resultados encontrados neste trabalho já se mostram melhores que os alcançados apenas com a detecção de picos implementada em Canales et al. (2007).

4 Metodologia

Neste capítulo serão apresentados os métodos utilizados para elaborar o sistema, desde as simulações do algoritmo para definir quais os melhores códigos, até a descrição do sistema eletrônico.

4.1 Cálculo do Tempo de Inspeção

Para o desenvolvimento do sistema, é essencial o conhecimento do tempo disponível para que o mesmo realize cada inspeção, que consiste da insonificação, recebimento do eco e processamento dos dados, para cada ponto do oleoduto. Este tempo foi calculado levando-se em conta a velocidade do *pig*, a espessura do duto corroído, a distância entre sua parede e o transdutor, a velocidade do ultrassom no meio (no caso petróleo) e no metal do duto, e a resolução que se deseja na inspeção (distância entre linhas consecutivas inspecionadas do duto).

Os cálculos devem ser feitos com base no pior caso possível de condições de trabalho para o *pig*, ou seja, quando ele terá o menor tempo disponível para efetuar a inspeção. Isso ocorrerá quando o *pig* estiver na maior velocidade possível, com a menor resolução de inspeção. Além disso devem ser levados em conta a distância entre o transdutor e a parede do oleoduto (consequentemente menor tempo entre emissão e recepção do sinal) e considerar uma espessura baixa no local devido à corrosão, para saber o tempo mínimo entre duas reflexões de sinal.

A velocidade máxima de deslocamento (v_m) que se espera que o *pig* trabalhe dentro dos oleodutos é de 1 m/s , e a distância mínima do transdutor até a parede (d_{tp}) é de 30 mm (OKAMOTO et al., 1999). A espessura de uma área corroída (e) é em média de 5 mm . A resolução máxima (r) de inspeção desejada (espaço entre pontos consecutivos que serão inspecionados ao longo do comprimento da tubulação) é de 1 mm , que já é

suficientemente grande para detectar problemas de corrosão. A figura 4.1 mostra um esquema com as grandezas envolvidas.

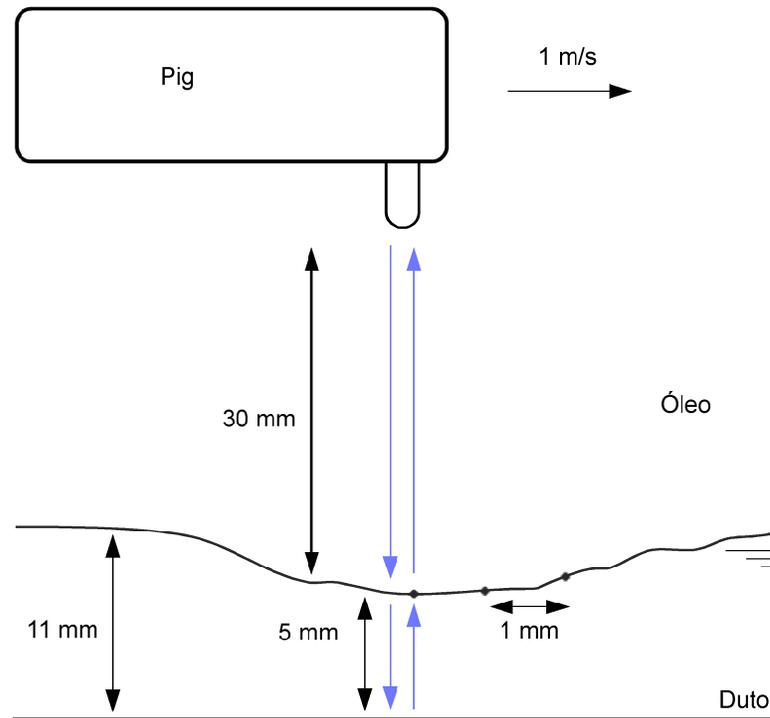


Figura 4.1: Representação do pig durante uma inspeção dentro do oleoduto. As flechas claras indicam o sinal sendo enviado e recebido pelo transdutor

A velocidade do ultrassom no petróleo (v_p) é de 1600 m/s , e no metal (v_{mt}) das paredes do duto é de 6200 m/s . O tempo total (t_t) só depende da velocidade do pig e da resolução da inspeção. Este é o tempo máximo que o sistema deve levar para realizar a insonificação, receber o eco e processar o sinal. Além deste tempo, temos dois tempos de viagem do sinal (do transdutor até a primeira parede do duto e sua volta (t_{v1}), e do transdutor até a segunda parede e sua volta (t_{v2})). Estes tempos são importante para que se possa calcular qual a duração máxima que o sinal codificado enviado pode ter, de modo que o recebimento de seus ecos não se sobreponha ao próprio envio, gerando assim ambiguidades na medida do TOA.

As equações 4.1, 4.2 e 4.3 mostram como calcular respectivamente os valores de t_t , t_{v1}

e t_{v2} :

$$t_t = \frac{r}{v_m} \quad (4.1)$$

$$t_{v1} = \frac{2 * d_{tp}}{v_p} \quad (4.2)$$

$$t_{v2} = \frac{2 * d_{tp}}{v_p} + \frac{2 * e}{v_{mt}} \quad (4.3)$$

Utilizando os valores definidos anteriormente, chega-se em $t_t = 1000 \mu s$, $t_{v1} = 37,5 \mu s$ e $t_{v2} = 39,11 \mu s$. Com estes tempos pode-se calcular também o tempo efetivo que o sistema terá para processar o sinal, que é o tempo total menos o tempo para o retorno do segundo eco aproximadamente. Esse tempo de processamento t_p é de $960,9 \mu s$.

4.2 Excitação dos Transdutores

Como já mencionado, os transdutores devem ser excitados em sua frequência de ressonância. Neste trabalho utilizar-se-á transdutores Aerotech ALPHA de 5 MHz, com diâmetro de 0.25" (GE, EUA). Portanto, deve-se utilizar uma onda portadora de frequência 5 MHz, modulada com BPSK e códigos de Barker.

É possível prever o comportamento do transdutor quando excitado por uma determinada onda. Para isso deve-se em primeiro lugar considerá-lo como um sistema *LIT* (*Linear e Invariante no Tempo*), o que é uma suposição válida para transdutores quando excitados em sua frequência de ressonância. A saída de um sistema LIT é a convolução da entrada (que no caso é a onda de excitação do transdutor) com sua função de transferência. Desta maneira é necessário determinar a função de transferência do transdutor operando em modo pulso/eco. Idealmente isso poderia ser feito modelando-se matematicamente o transdutor e estimando-se os parâmetros deste modelo. No entanto, esta abordagem apresenta algumas limitações: os modelos em geral representam apenas o comportamento do transdutor em uma faixa de frequência, o processo é trabalhoso e sujeito a erros.

Neste trabalho, a função de transferência foi obtida experimentalmente excitando-se o transdutor com um pulso estreito de alta tensão e adquirindo-se o sinal de eco recebido de uma superfície metálica e plana por este mesmo transdutor. Este método permitiu obter a função de transferência do transdutor em operação pulso/eco de forma aproximada, que apesar de também estar sujeito a erros mostrou um resultado suficientemente bom para

todas as simulações que serão feitas.

Este procedimento foi feito com um equipamento pulsador/receptor da Panametrics, modelo 5072PR (Olympus Panametrics, EUA), repetido 100 vezes, e dos resultados foi tirada a média, de modo a eliminar ao máximo ruídos eletrônicos. Da média resultante foi separado o primeiro eco, eliminando valores abaixo de 1% do valor da faixa dinâmica, que podia ser considerado como ruído. A figura 4.2 mostra o resultado obtido.

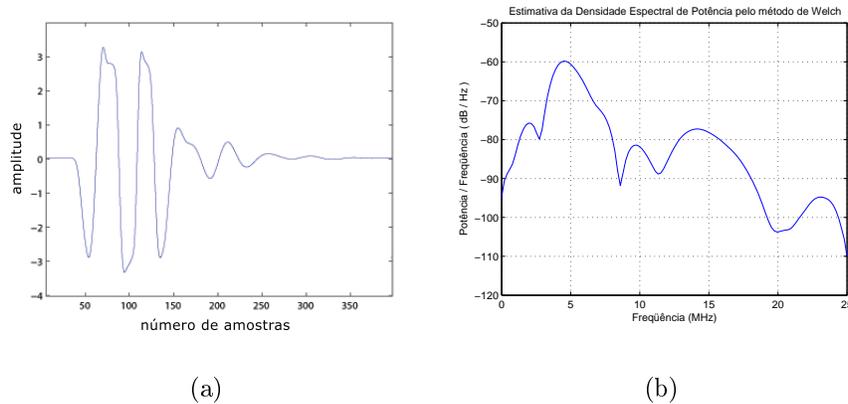


Figura 4.2: Função de Transferência do Transdutor ALPHA: (a) no domínio do tempo e; (b) densidade espectral de potência

A densidade espectral de potência foi estimada através do método de Welch (COOLEY; LEWIS; WELCH, 1967). Tendo a função de transferência, o passo seguinte é investigar qual seria a forma de onda de excitação mais apropriada. Por computador podem-se gerar os sinais esperados produzidos pelo transdutor, convoluindo-se a função de transferência com a forma de onda escolhida. Esta abordagem é conhecida como *signal design*, onde simula-se a criação de sinais para decidir qual o mais adequado. A figura 4.3 mostra a comparação entre um sinal produzido excitando-se um transdutor com um sinal modulado com código de Barker de 2 bits, 8 ciclos por bit, e um sinal sintetizado com este mesmo código de Barker convoluído com a função de transferência mostrada na figura 4.2.

Percebe-se que os dois sinais são muito semelhantes e a representação da densidade espectral de potência mostra que as componentes espectrais estão muito próximas. As pequenas diferenças ocorrem principalmente pelo fato da onda de excitação real não ser um sinal perfeitamente quadrado como o usado na geração dos sinais sintetizados. Como o sinal quadrado ideal possui componentes espectrais de alta frequência, o espectro do sinal sintetizado apresenta também componentes de alta frequência que não estão presentes no sinal real com a mesma intensidade. A figura 4.4 mostra o exemplo de um sinal real de

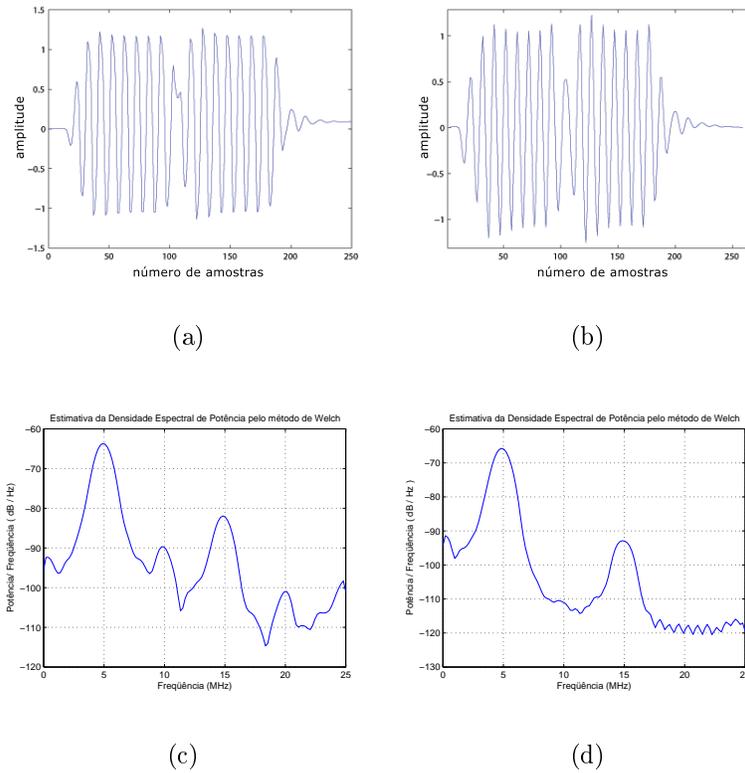


Figura 4.3: Comparação entre sinal real e sinal sintetizado por computador, no caso de 2 bits e 8 ciclos por bit. Sinal no tempo: (a) Real e (b) Sintetizado. Densidade Espectral de Potência: (c) Real e (d) Sintetizado.

excitação, relativamente diferente do sinal ideal.

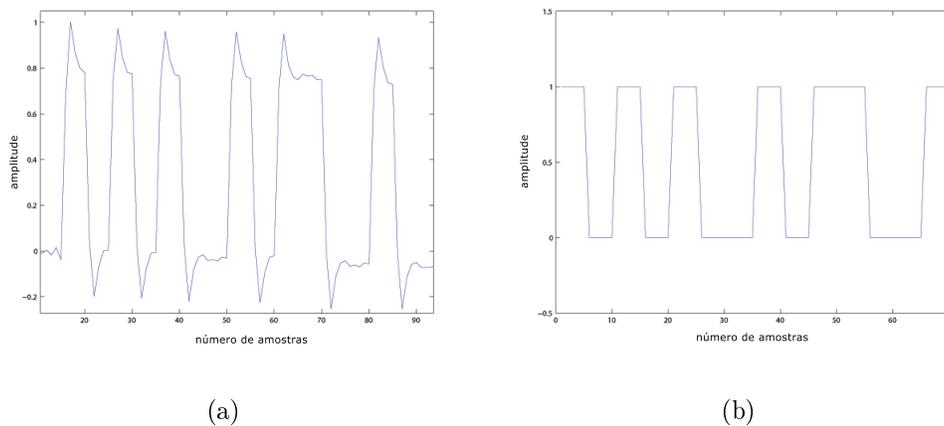


Figura 4.4: Sinal real codificado que excita o transdutor (7 bits, 1 ciclo por bit). (a) Sinal real e (b) sinal utilizado nas simulações

Na seção 4.1 definiu-se que a duração da onda emitida não deve ultrapassar $37,5 \mu s$.

Numa frequência de 5 MHz, isso corresponde a 187,5 ciclos. Logo a onda de ultrassom não pode ultrapassar 187 ciclos para evitar sobreposição do sinal enviado com o recebido.

Porém, é desejável também evitar a sobreposição do primeiro eco com o segundo eco. Para isso a duração da onda não deve ultrapassar a diferença entre t_{p1} e t_{p2} , que é de $2,05 \mu s$. Isso corresponde a aproximadamente 10 ciclos, que ainda é uma quantidade razoável de energia a ser enviada. A duração de um sinal emitido depende da duração da resposta impulsiva, e da duração do sinal de excitação (matematicamente isso é explicado pela equação 4.4, que é a equação de convolução, onde a duração do sinal resultante é igual a soma das durações dos sinais menos 1 amostra).

$$w(k) = \sum_j u(j)v(k+1-j) \quad (4.4)$$

No domínio do tempo isso quer dizer que a soma das durações dos sinais não deve ultrapassar $2,25 \mu s$. Na seção 4.2.2 será testada a influência da sobreposição no resultado.

4.2.1 Potência enviada

O primeiro passo para decidir qual o sinal codificado será usado é avaliar se o transdutor é capaz de responder eficientemente a essa excitação, uma vez que a banda do transdutor é limitada. Para obter uma boa relação S/R é necessário em primeiro lugar que seja possível enviar uma quantidade razoável de potência.

Para isso, foram sintetizados sinais com diversas combinações de número de bits e ciclos, e foi calculada a potência média para cada um destes sinais, através da equação 4.5.

$$P_{sinal} = \frac{\sum_{n=1}^N s(n)^2}{n} \quad (4.5)$$

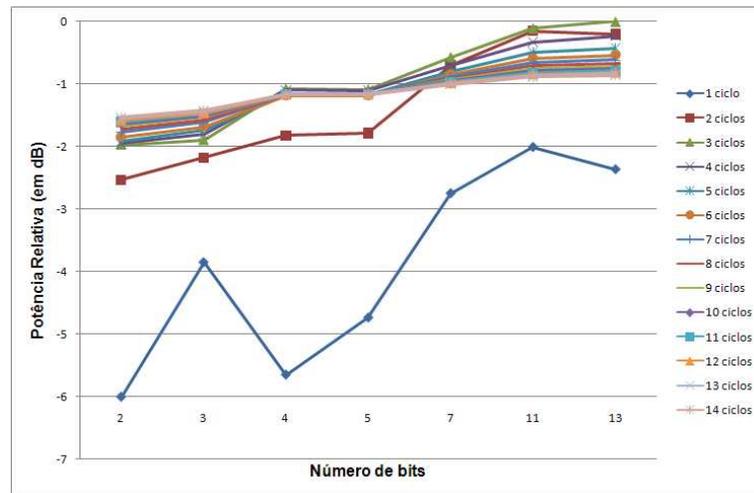
,onde n é o número de amostras do sinal.

O sinal de maior potência encontrado foi utilizado como referência (0 dB), e foram calculadas as atenuações dos outros sinais em relação a este (também em dB).

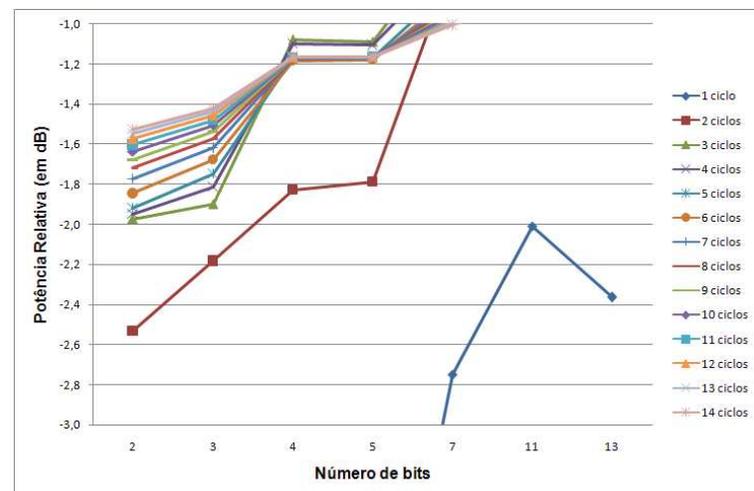
Foram simulados sinais de 2 a 13 bits, com 1 a 14 ciclos por bit, já que o maior sinal possível (13 bits, 14 ciclos) corresponde quase ao limite de ciclos calculado na seção 4.1, que era de 187 ciclos. A figura 4.5 mostra o resultado para os sinais sintetizados.

A escolha dos sinais a serem utilizados é uma combinação de diversas variáveis que podem ser consideradas, ou não. Como o sistema será embarcado, uma grande potência se torna um problema para as baterias do mesmo, mas quanto maior a potência, melhor os resultados da inspeção. Dessa maneira, tem-se uma relação de compromisso para se decidir qual é o melhor sinal a ser utilizado.

Definiu-se uma região entre -1 dB e -3 dB como faixa de sinais candidatos para a utilização no algoritmo, considerando-se estas potências como uma boa relação de com-



(a)



(b)

Figura 4.5: Transmissão de potência relativa ao maior valor, em dB, para: (a) todos os sinais e, (b) apenas os sinais dentro do limite estabelecido

promisso (eliminando os de baixa potência, e os de comprimento muito grande).

Com estes resultados percebeu-se que o que mais influência na potência dos sinais são as inversões de fase dos códigos, que causam mudanças bruscas no sinal, e a quantidade de ciclos acaba não colaborando com o envio de uma potência muito maior.

Tendo estes sinais candidatos, considerou-se mais um critério para seleção dos sinais. Como dito, espera-se medir espessuras de até 5 mm, que correspondem a 10 ciclos de sinal. Contando com isso serão escolhidos os códigos que ocupem até 20 ciclos (o que já ocasiona uma sobreposição de 50% do sinal). Estes sinais serão simulados em testes de detecção para definir suas respectivas eficácias.

4.2.2 Códigos de Barker e número de ciclos

Nesta seção são apresentadas simulações para determinar qual a combinação de códigos e número de ciclos será usada para excitar o transdutor.

Os testes foram feitos utilizando sinais sintetizados em computador, com uma distância pré-definida entre os ecos (d_{ecos}). Estes sinais são sintetizados seguindo as seguintes etapas:

- Convolui-se um eco padrão adquirido com a onda de excitação (portadora modulada pelo código de Barker), gerando assim um eco sintetizado.
- Gera-se um sinal contendo o primeiro e o segundo eco, utilizando-se duas vezes o eco sintetizado, separados pela distância d_{ecos} . A amplitude do segundo eco é multiplicada por um fator k , simulando assim a atenuação do sinal, e também inverte-se a fase deste para simular a reflexão na segunda parede.
- Este novo sinal é considerado o sinal resultante de uma insonificação, sem ruído.
- Adiciona-se um nível pré-determinado de ruído gaussiano branco (em relação à potência média do eco sintetizado).

A figura 4.6 exemplifica estas etapas com um código de Barker de 4 bits, 2 ciclos por bit, distância $d_{ecos} = 20 \text{ mm}$ (aproximadamente $3,5 \mu\text{s}$), e 10 dB de S/R.

O fator k foi determinado a partir de um sinal colhido de uma reflexão em superfície limpa. A figura 4.7 mostra este sinal, com os múltiplos ecos gerados nas paredes de uma

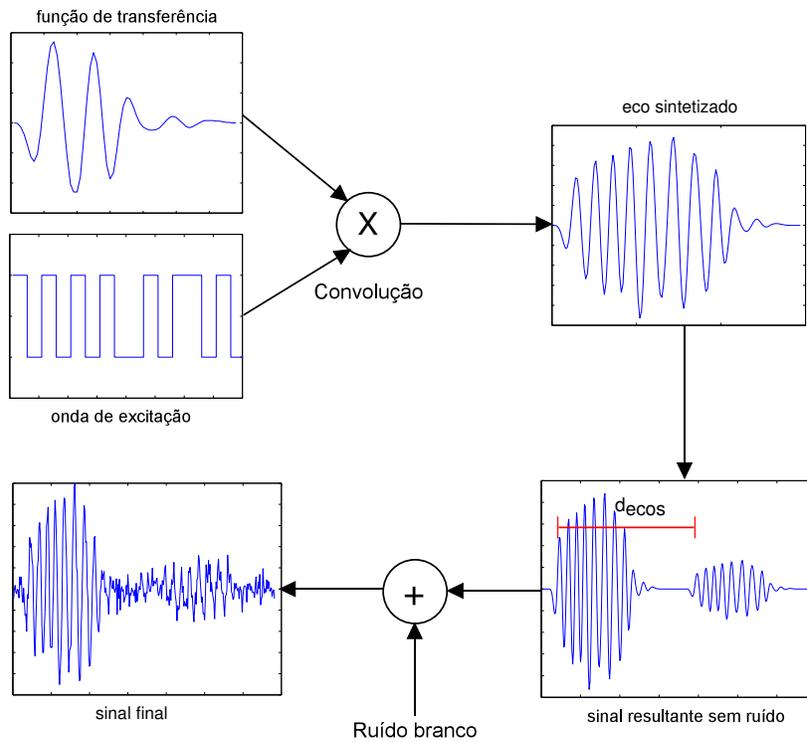


Figura 4.6: A onda de excitação é convoluída com o sinal de eco padrão, e o resultado montado em um sinal maior com distância d_{ecos} . Por fim adiciona-se ruído branco gaussiano.

placa de alumínio imersa em água. Para calcular a atenuação, o primeiro e o segundo eco deste sinal foram separados, e a potência média de cada um calculada. Com estas potências calculou-se a atenuação de energia na primeira reflexão. Tendo o valor desta atenuação (que no caso foi de 10 dB), podemos fazer o processo inverso para chegar a ao fator que multiplicará a amplitude gerando o segundo eco nos sinais sintetizados. Este caminho inverso é mostrado abaixo na equação 4.6, resultando em um $k = 0.32$.

$$\begin{aligned}
 10 * \log_{10} \left(\frac{P_{eco2}}{P_{eco1}} \right) &= Atenuacao(dB) & (4.6) \\
 10 * \log_{10} (k^2) &= -10 \\
 k &= 0.3162
 \end{aligned}$$

Para uma relação S/R desejada (equação 4.7) adiciona-se ao sinal resultante um ruído gaussiano branco, cuja a potência é uma fração (em porcentagem) da potência média do eco sintetizado, calculada de acordo com a equação 4.5.

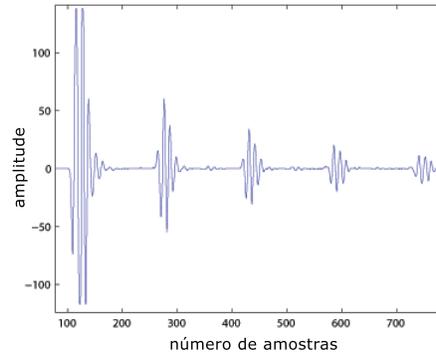


Figura 4.7: Sinal obtido em uma superfície plana de alumínio com seus múltiplos ecos gerados por reflexões internas nas paredes

$$S/R(dB) = 10 * \log_{10} \left(\frac{P_{sinal}}{P_{ruído}} \right) \quad (4.7)$$

Tem-se assim o sinal sintetizado final.

Com os sinais sintetizados, foram feitos testes de detecção dos ecos com o filtro casado (convoluindo-se com o sinal original invertido no tempo e detectando-se os picos na saída). A saída ideal do filtro casado seria um sinal com dois picos muito distintos, cuja localização forneceria o TOA. Porém, com a adição de ruído, estes picos vão ficando cada vez menos destacados do resto do sinal.

A detecção dos picos também será feita pelo sistema embarcado, e desta maneira, sua simulação também será feita em computador, de modo que os resultados gerados sejam os mais fiéis aos que serão encontrados no sistema real. O algoritmo para detecção dos picos será explicado na seção 4.2.3.

Como foi definido, o eco resultante deve ter menos que 20 ciclos. A resposta impulsiva do transdutor tem aproximadamente 4 ciclos, logo o código deve ter no máximo 16 ciclos. Utilizando este critério, mais o critério de potência definido na seção 4.2.1 chega-se nos seguintes códigos (tabela 4.1) para serem utilizados nas simulações.

Nas simulações serão utilizados dois tipos de filtro. O primeiro contendo o sinal com resolução máxima do sistema (8 bits por coeficiente - valores de -128 até 127) codificado com Barker (ou seja, a convolução do eco padrão com o código), denominado F8B, e o segundo apenas um sinal com resolução de apenas 1 bit por coeficiente (0 ou +1), que nada mais é que a própria onda de excitação (código de Barker com os ciclos por bit), denominado F1B. Isso é feito para testar se este segundo filtro pode ter um funcionamento

Tabela 4.1: Códigos que serão simulados

Número de Bits	Número de Ciclos
2	2, 3, 4, 5, 6, 7 e 8
3	2, 3, 4 e 5
4	2, 3 e 4
5	2 e 3
7	1 e 2

satisfatório em relação ao filtro de maior resolução, pois sua implementação é inúmeras vezes mais simples e rápida, já que se trata de um filtro apenas com somas e subtrações, ao contrário do primeiro, que necessita também de multiplicações.

No caso do filtro de 1 bit, ao invés de se usar os valores 0 e 1 (filtro com somas), serão usados os valores +1 e -1 (filtro com somas e subtrações), que gera um resultado com amplitudes maiores e custo computacional igual.

Foram injetados ruídos para se obterem sinais com S/R de 30, 20, 10 e 7 dB. Isso corresponde respectivamente a potências de ruído de 0,1%, 1%, 10% e 20% em relação a potência do primeiro eco.

A distância entre os dois ecos nas simulações foi fixada em 4,5 mm (situação de espessura 10% menor que a que se espera encontrar nas inspeções). Com isso ocorrerá sobreposição em alguns casos (para os sinais com mais de 9 ciclos no total) e algumas conclusões poderão ser tiradas da influência desta sobreposição no resultado.

Todas as simulações serão realizadas 100 vezes, e a média do resultado será considerado como resultado padrão para análise de cada combinação.

4.2.2.1 Resultados das simulações

O primeiro resultado importante das simulações foi a verificação de eficiência do F1B comparado ao F8B. Surpreendentemente, nem sempre o F8B se mostrou melhor que o F1B. Em alguns casos os erros do filtro de 8 bits foram maiores. Isso mostra que o uso do filtro simples é totalmente viável, e pode gerar resultados confiáveis na inspeção. A figura 4.8 mostra um exemplo onde o F8B foi melhor e um exemplo onde ele foi pior, e a figura 4.9 mostra as saídas dos filtros para estes exemplos.

Observando as saídas dos filtros, percebe-se que são muito parecidas, ressaltando o fato do F1B ser quase tão poderoso quanto o F8B. Os erros nestes casos se dão principalmente

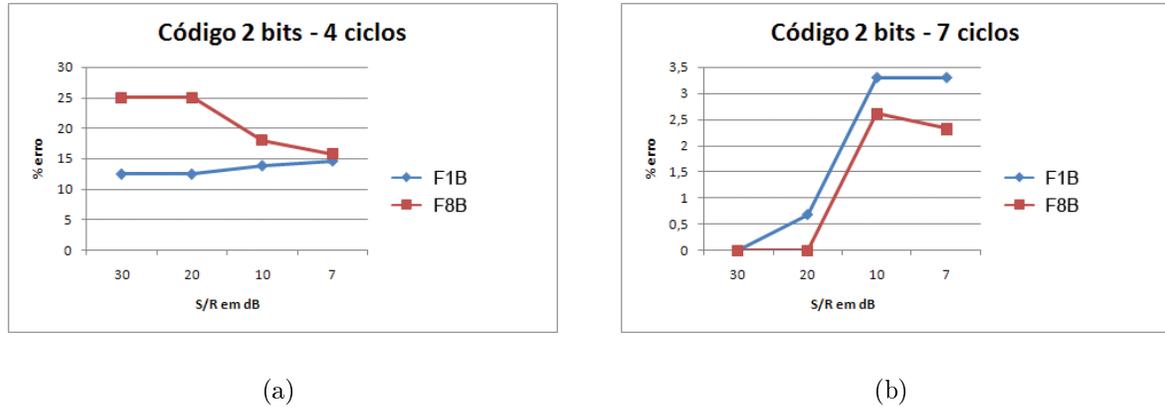


Figura 4.8: Resultado das simulações para o caso: (a) 2 bits e 4 ciclos e (b) 2 bits e 7 ciclos. Os erros são expressos em porcentagem em relação ao valor de espessura pré-determinado na síntese dos sinais

pelos picos de saída terem valores muito próximos (devido a atenuação do segundo eco), e desta maneira o detector de picos falha na detecção do pico correto. Com isso fica

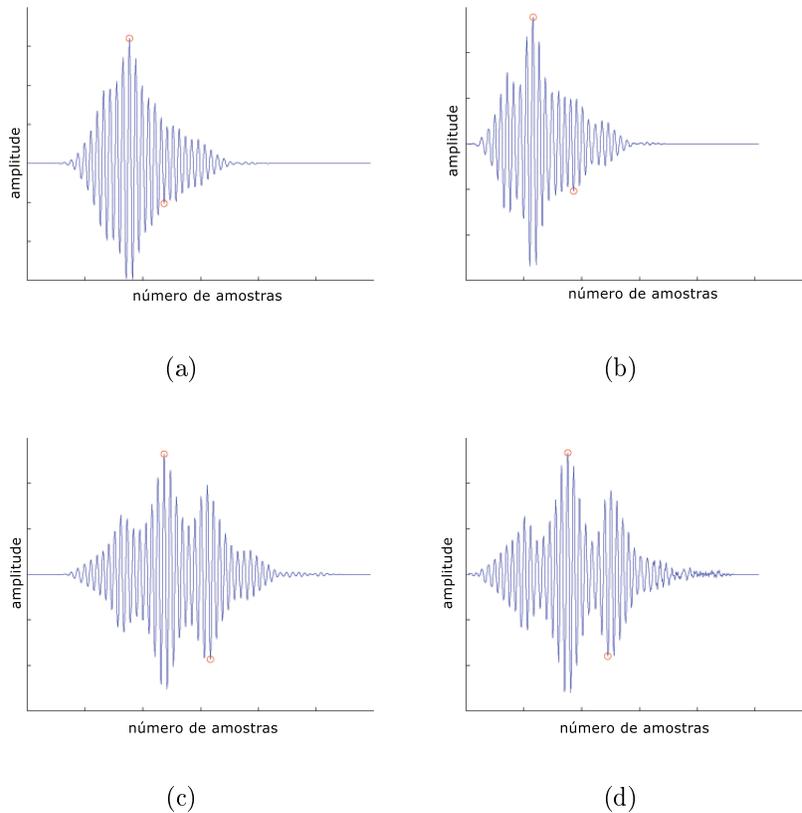


Figura 4.9: Saídas do filtro casado. 2 bits 4 ciclos com (a) F8B e (b) F1B, e 2 bits 7 ciclos com (c) F8B e (d) F1B

determinado que o F1B será utilizado na implementação do sistema.

A primeira variável a ser considerada agora para determinação do melhor sinal é a detecção do primeiro eco. No caso do filtro completo, a detecção deste eco teve erro 0 em todas as simulações, podendo ser considerada. Com o F1B, o maior erro encontrado foi de 0,1% com o maior nível de ruído (7 dB), o que é considerado extremamente baixo para esta aplicação.

Em seguida, deve-se analisar dentre os sinais simulados quais tiveram uma melhor eficiência na detecção do segundo eco (e desta maneira, na determinação da espessura). A figura 4.10 mostra os erros para todos os sinais de acordo com a quantidade de ruído injetada.

Limitando os erros em 2% é possível separar alguns códigos que servirão de candidatos para os testes reais. Os códigos escolhidos são:

Tabela 4.2: Códigos para os testes reais

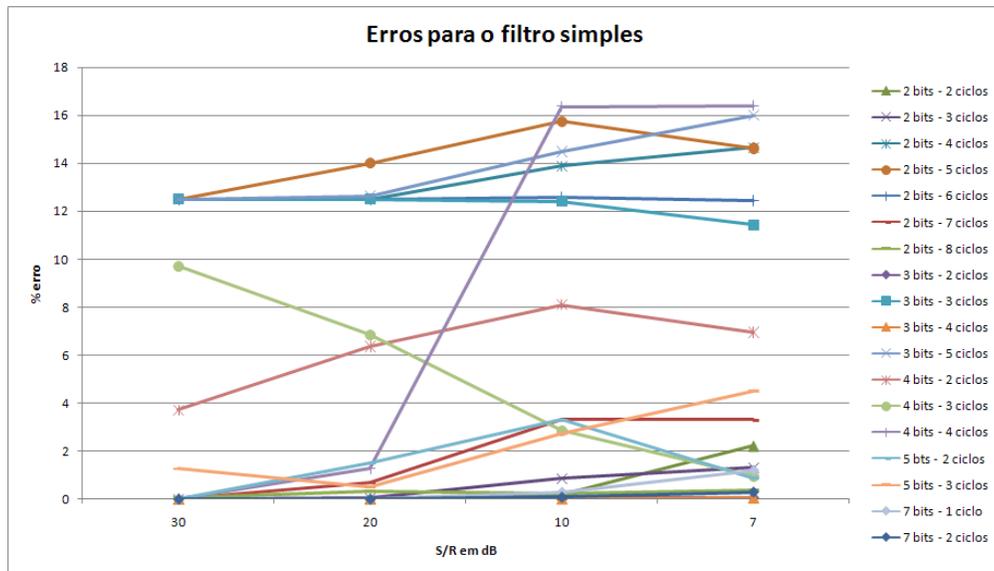
Número de Bits	Número de Ciclos
2	3 e 8
3	3 e 4
7	1 e 2

Além disso, pode-se perceber que o algoritmo de detecção de picos não funcionou perfeitamente, detectando um segundo pico ainda pertencente ao primeiro sinal. A figura 4.11 mostra um exemplo onde isso aconteceu, e outro exemplo onde mesmo com a sobreposição foi possível distinguir os sinais.

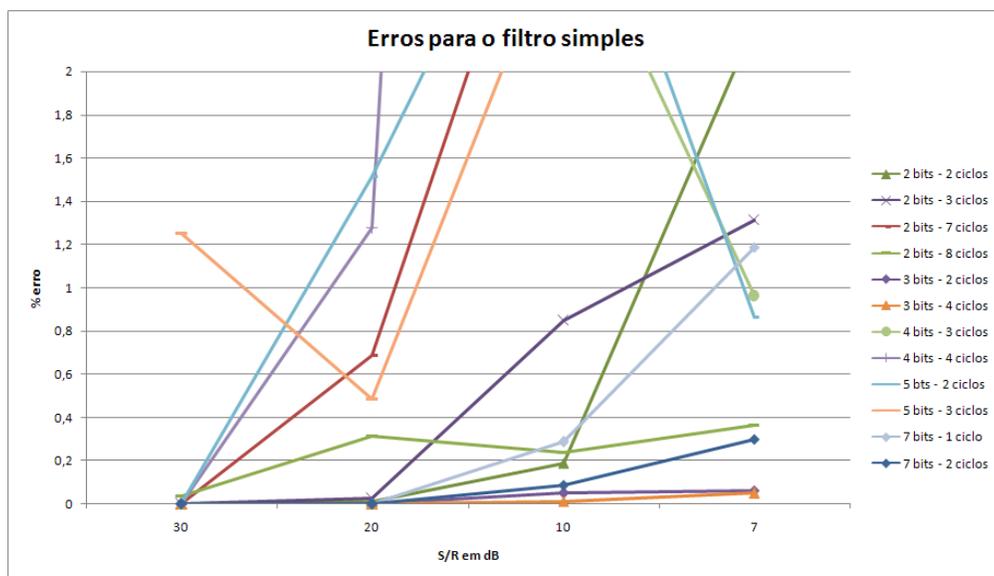
Pode-se perceber pelos resultados que o algoritmo funciona muito bem mesmo com a influência de ruído e em alguns casos sobreposição. Mesmo com níveis de S/R altos, os resultados se mostraram fiéis ao que era esperado, portanto o ruído não deverá ser um fator limitante para o sistema.

Um adendo que pode ser feito neste momento é o da distância mínima que o transdutor deve estar da parede do oleoduto para que a volta do eco não ocorra ainda durante o envio do próprio sinal.

O maior sinal dos testes reais é o de 2 bits, com 8 ciclos por bit, totalizando uma duração de $3,2 \mu s$. Portanto, fazendo o cálculo inverso ao que foi feito na seção 4.1 chega-se que a distância mínima do transdutor é de 10 mm, ainda com sobra para os 30 mm citados anteriormente.



(a)



(b)

Figura 4.10: Erros em porcentagem na simulação dos códigos. Em (a) todos os resultados e em (b) limitando nos erros menores que 2%

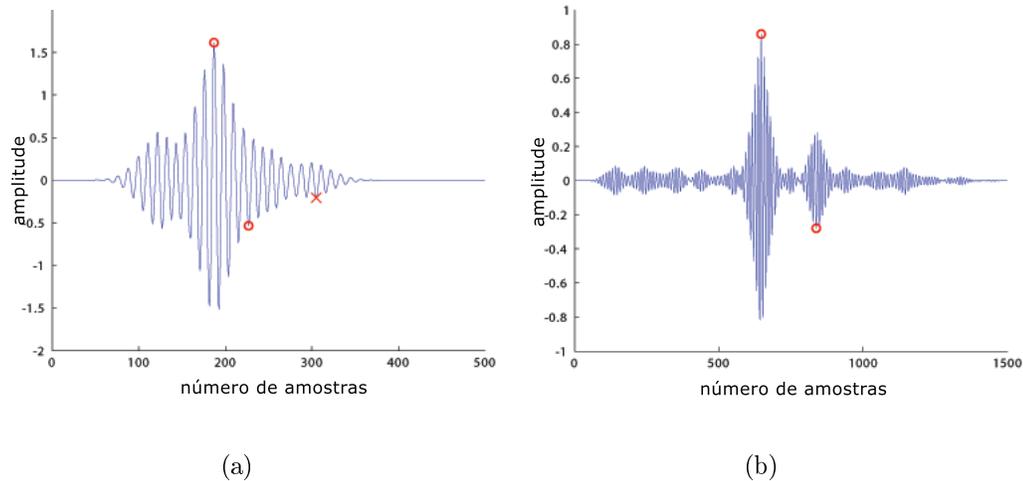


Figura 4.11: Em (a) a sobreposição do sinal causou muita ambiguidade na saída do filtro, e o detector de picos detectou o pico errado ("×"), ao invés do correto, representado por "o". Em (b), mesmo com a sobreposição, a saída do filtro é muito clara, resultando numa detecção correta

4.2.3 Detecção de Picos

O detector de picos é um algoritmo relativamente simples, mas necessário para detectar de maneira correta os picos que realmente interessam (evitando picos falsos, como será descrito a seguir). O sinal que sai do filtro casado contém inúmeros picos locais, devido a onda portadora. A detecção do primeiro eco é relativamente mais simples, pois se trata do maior pico na saída do filtro. Porém, nas simulações foi observado que o segundo eco não é o segundo maior pico na maioria das vezes, já que o primeiro eco ainda causa outros picos na saída. Desta maneira é necessário um critério que evite a detecção de um pico referente ainda ao primeiro eco como sendo o segundo eco.

Para isso foi desenvolvido um algoritmo que conta com "tempos mortos", que são intervalos de tempo onde o detector de picos é desligado. A figura 4.12 mostra um exemplo de um sinal e os seguintes parâmetros configuráveis:

- DELAY - Este parâmetro não faz parte diretamente do detector, mas sim da própria aquisição do sinal. Este valor é uma janela no tempo no qual as amostras do sinal são ignoradas.
- SIZE - Este parâmetro também está relacionado à aquisição do sinal, e representa o número de amostras (comprimento) do sinal.

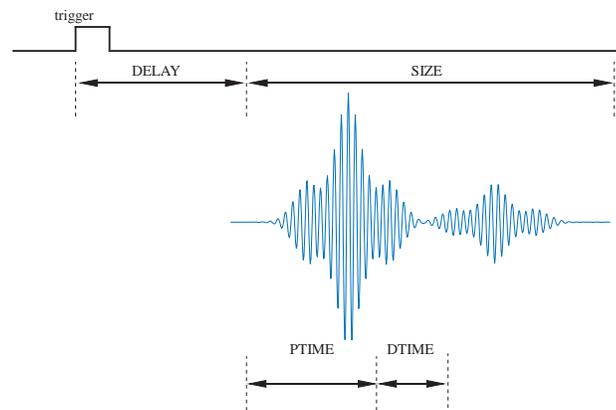


Figura 4.12: Parâmetros configuráveis na detecção de picos

- PTIME - Este parâmetro representa a janela no tempo em que o detector de picos irá procurar o primeiro pico. Isto serve para o processamento ocorrer em tempo real, e não seja preciso procurar no sinal inteiro o maior pico, e só depois procurar o segundo.
- DTIME - Este parâmetro é a janela de tempo a partir do final da procura pelo primeiro pico (final de PTIME), que o detector de picos para de funcionar, de modo a não dar como resultado do segundo eco um pico ainda pertencente ao primeiro eco. Este tempo pode ser determinado tomando como base a espessura mínima esperada da parede, e pode ser atualizado a cada nova insonificação, tomando como base o valor da última espessura determinada.

Com a correta determinação destes parâmetros, a detecção dos picos se torna muito mais precisa, e os resultados mais confiáveis.

Para melhorar a eficiência do algoritmo, o sinal passa primeiro por um detector de picos locais, que procura pontos maiores que seus 4 adjacentes, isto é, os dois pontos anteriores e posteriores a ele (isso é feito visando procurar picos destacados no sinal, o que gera uma envoltória mais “suave” do mesmo). Com isso eliminam-se os picos presentes devido aos ciclos do sinal, e passa-se a observar apenas os picos que mostram a presença do sinal na saída do filtro casado. Essa envoltória passa então pelo detector de picos explicado anteriormente. Dessa maneira, a procura do segundo pico é dependente tanto do tempo morto quanto da presença de picos na envoltória do sinal. A figura 4.13 mostra um exemplo de um sinal antes e depois de passar por esse detector de picos locais.

A detecção de picos locais não foi implementada na FPGA, que contou apenas com

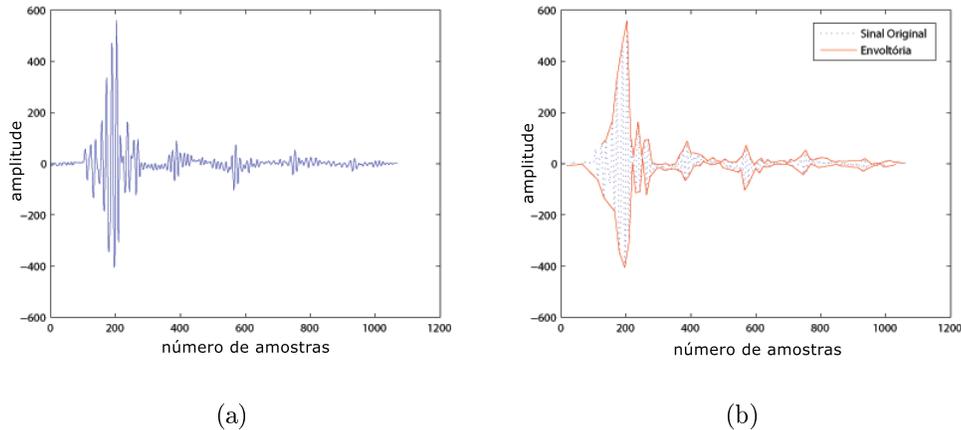


Figura 4.13: Exemplo do detector de picos locais, resultando na envoltória do sinal. São detectadas as envoltórias positiva e negativa, pois no caso do segundo eco procura-se o pico negativo

o detector de picos, que procura o maior (ou menor) valor dentro das janelas de tempo. Porém, a detecção de picos locais foi codificada em MATLAB (Mathworks, EUA) e funcionou como planejada, ficando faltando apenas a transcrição do código em matlab para VHDL para que seja implementada no sistema. O código em Matlab pode ser encontrado no anexo A.

4.3 Sistema Proposto

Com os parâmetros do filtro definidos através das simulações, sua implementação digital pode ser feita na FPGA. A FPGA abrigará não apenas o filtro, mas todo o circuito responsável pela aquisição e processamento de sinais. A figura 4.14 mostra um esquema de blocos do circuito.

Os blocos apresentados na figura 4.14 são responsáveis pelas seguintes funções:

- MUX - Não faz parte do circuito da FPGA, porém é controlado por ela. É responsável por selecionar qual sinal será analisado, para casos em que haja mais de um transdutor trabalhando.
- Condicionador de sinal - Também não faz parte da FPGA, mas é essencial para o trabalho da mesma. É neste bloco que o o sinal será filtrado (por um filtro analógico), amplificado e amostrado por um conversor A/D. Este bloco está em

- Controlador de Parâmetros - Neste bloco são armazenados todos os parâmetros configuráveis dos outros blocos do sistema. Ele é responsável por receber os parâmetros do usuário e os repassar para cada bloco.
- Controlador de I/O - Este bloco é quem faz a interface de qualquer dado que queira entrar ou sair da FPGA. Como o barramento de comunicação com o computador tem um número inferior de bits ao necessário para controlar todos os blocos da FPGA, este bloco faz a multiplexagem dos dados, a partir de endereços também fornecidos a partir do computador.
- Computador - Este é o bloco que inclui o microprocessador responsável pelo controle do sistema. No caso deste trabalho ele é um computador padrão PC-104..

Como dito anteriormente, este sistema deve funcionar em tempo real, e também foi decidido que o mesmo deve funcionar em *sampling-time*. Deste modo, a cada período de amostragem do sinal de entrada, o filtro também gera uma amostra de saída, isto é, o sinal é processado na medida em que é amostrado, não sendo necessário armazená-lo que seja processado durante a aquisição de um novo sinal. Para esse desenvolvimento em *sampling time* foi necessário estudar uma maneira de fazer o processamento em alta frequência, com alto nível de paralelismo, e essa abordagem não foi trivial. A seguir será explicado o desenvolvimento dos principais blocos do sistema. No C podem ser vistos os códigos em VHDL dos blocos, e no anexo B podem se vistos os esquemáticos do sistema, sendo o esquemático da página 114 o diagrama de blocos principal do sistema. Neste esquemático temos os seguintes principais blocos:

4.3.0.1 Controle de Aquisição

Em primeiro lugar, foi necessário desenvolver os blocos que controlassem o funcionamento do sistema. Um conjunto de três blocos descritos a seguir é responsável pelos sinais de controle do sistema, que guiarão os outros blocos a exercerem suas respectivas funções. São eles:

- Delay Acq - Bloco que controla o tempo de delay entre o recebimento do trigger, e o início da aquisição. É basicamente um contador sensibilizado pelo trigger, e no final da contagem, libera um sinal indicando o início de aquisição.

- Controle Acq - Este bloco é controlado pelo sinal de início de aquisição, e pelo parâmetro SIZE (tamanho do sinal que deve ser adquirido). Também é basicamente um contador, e indica para outros blocos por quanto “tempo” (amostras) o sistema deve adquirir e filtrar o sinal.
- Controle Detec - Bloco que através dos parâmetros de detecção de picos gera os sinais que controlarão os intervalos de tempo de funcionamento do detector de picos.

4.3.0.2 Trigger

Todo o funcionamento do sistema (aquisição, filtragem, armazenamento dos sinais) é disparado a partir de um sinal de trigger, e no caso do pig, esse trigger virá de um odômetro. Porém, como o sistema foi desenvolvido visando diversos tipos de testes, também foram adicionadas as opções de trigger interno (gerado via software), ou um trigger externo de uma fonte qualquer. O trigger que será utilizado é selecionado por um MUX, e por um sinal de controle definido como parâmetro de inspeção. Além do MUX, foi desenvolvido um bloco para detectar casos de re-trigger (bloco “Detecta trigger”), isto é, um novo trigger chega durante no meio da aquisição do sinal. Neste caso o sistema despreza esse novo trigger, mas atualiza o próximo endereço de aquisição atual (em suma, o sistema despreza o ponto de aquisição que gerou o novo trigger). O bloco libera a entrada de um novo trigger apenas quando o sistema acaba a aquisição e processamento dos dados, deixando de enviar o sinal que indica este estado (esse sinal é gerado pelo bloco “Controle acq”).

Como último detalhe, é válido citar que o sistema também conta com um detector de quadratura e um contador, que servem para tratar os sinais vindos de um odômetro, ou de um encoder, e transformar estes sinais em trigger, a partir de uma resolução definida também como parâmetro de inspeção.

4.3.0.3 Filtro casado

Como dito anteriormente, o filtro casado deverá funcionar em *sampling-time*, e com a abordagem do filtro de coeficientes de 1 bit, a filtragem se resume na utilização de somadores e subtratores de amostras (para que possamos somar e subtrair as amostras, conforme o código definido para o filtro). Quando o coeficiente do filtro for igual a 1, basta passar a amostra do sinal pelo somador, e se for igual a -1, a amostra do sinal deve ter seu sinal trocado antes de entrar no somador. Além dos somadores, é necessário uma

cadeia de atrasadores (que consiste basicamente em diversos registradores em série) para trabalhar com diversas amostras do sinal ao mesmo tempo (a quantidade de amostras necessárias é igual ao número de coeficientes do filtro).

A idéia seria colocar a cadeia de atrasadores, e inserir todas as amostras em um grande somador, conforme mostra o exemplo da figura 4.15 (no exemplo da figura é implementado um filtro hipotético com coeficientes iguais a $-1,1,-1,1,1$). Essa abordagem implementaria o filtro como desejado, e iria gerar resultados em *sampling-time*.

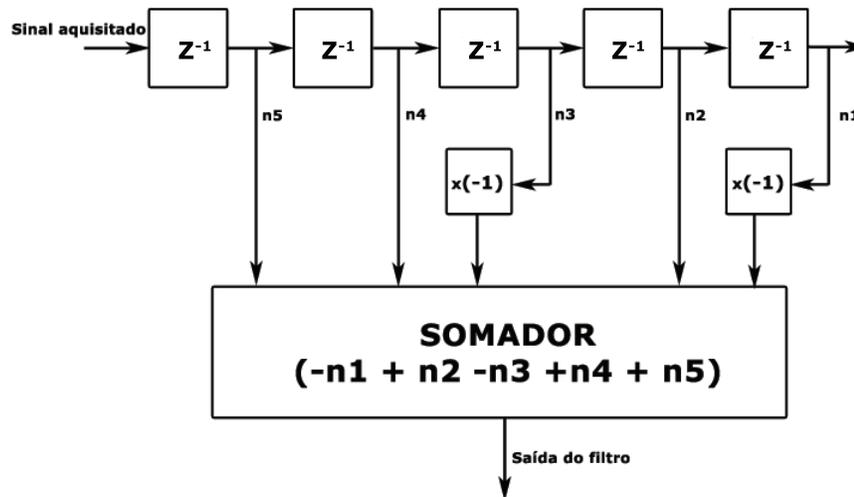
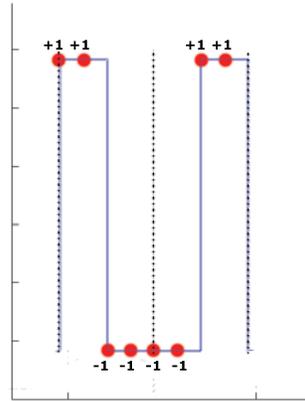


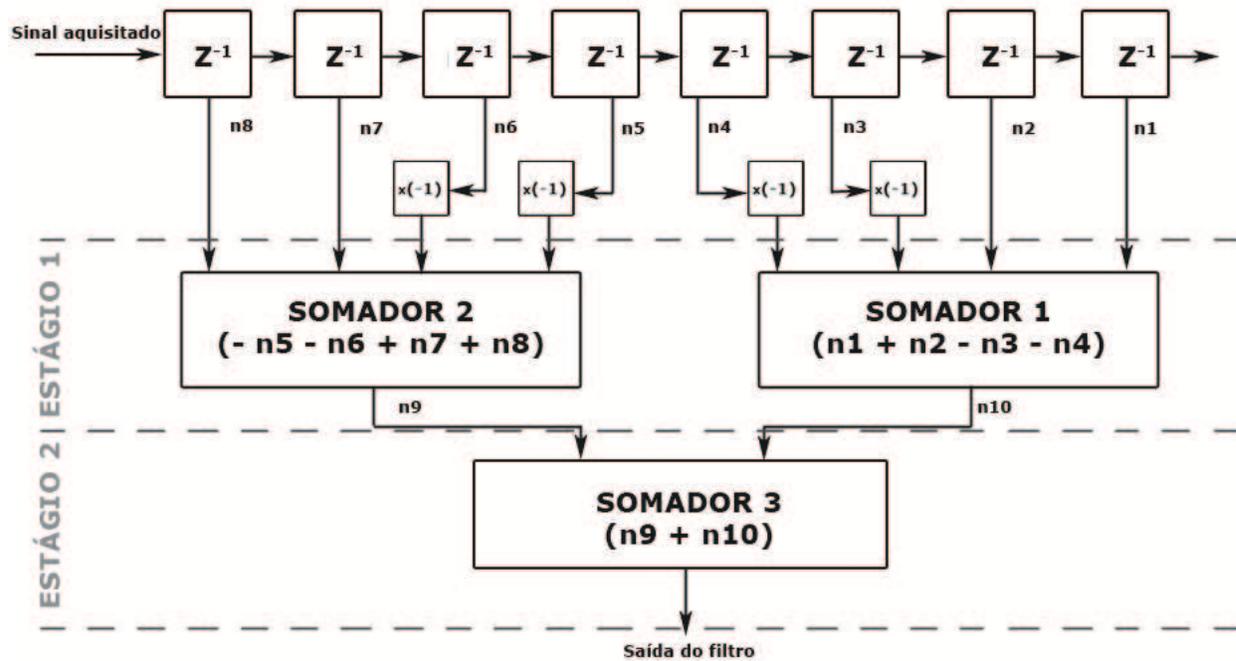
Figura 4.15: Exemplo de uma configuração de um filtro qualquer, com somadores em série

O problema é que quando o filtro foi implementado desta maneira em VHDL, o mesmo não conseguiu funcionar na frequência de amostragem do sistema (50 MHz). Para resolver esse problema decidiu-se construir um filtro em estágios, funcionando com somadores cascadeados. A figura 4.16 mostra o sinal e o diagrama de blocos de um exemplo de filtro com código de Barker de 2 bits, um ciclo por bit, com uma taxa de amostragem de 4 amostras por ciclo.

Neste caso, o primeiro estágio do filtro divide o sinal amostrado em dois blocos, realiza as operações com cada metade do sinal, e depois soma o resultado das metades, gerando o resultado final. Neste caso temos que no primeiro estágio o filtro soma as duas primeiras amostras e subtrai a terceira e quarta, além de subtrair a quinta e sexta e somar as duas últimas separadamente. Após isso, no segundo estágio, o sistema apenas soma os resultados. Essa abordagem gera resultados com atrasos iguais a quantidade de coeficientes do filtro, isto é, se o filtro tem 10 coeficientes, ele irá gerar os resultados com 10 períodos de amostragem de atraso em relação ao sinal de entrada.



(a)



(b)

Figura 4.16: Em a) o filtro de 2 bits, e suas respectivas amostras, e em b) o diagrama representando a implementação do mesmo

4.3.0.4 Detector de picos

Como explicado na seção 4.2.3, o detector de picos foi implementado no sistema, e os sinais que controlam seu funcionamento vem do bloco *Controle Detec*. Ele recebe um sinal indicando em que momento deve procurar pelo pico positivo (maior valor dentro da janela do tempo), que representa o TOA da primeira parede, e outro sinal indicando quando deve

procurar o pico negativo (menor valor dentro da janela de tempo), que representa o TOA da segunda parede. A procura do pico negativo ocorre devido a inversão de fase do sinal na reflexão da segunda parede, explicada na seção 3.1.3.

4.3.0.5 Demais blocos

Além dos citados acima, o sistema conta com outros blocos que completam sua funcionalidade. São blocos de controle da comunicação com o PC-104, blocos de registro dos parâmetros definidos através do PC, blocos que geram os endereços de escrita e leitura de memória, blocos para leitura de encoder e blocos de memória para salvar os dados da inspeção. O sistema também permite que o usuário selecione se quer salvar o sinal original, ou o sinal filtrado (visando principalmente testes com processamento *off-line*).

4.3.1 Placa Pulsadora

Para o funcionamento de todo o sistema, também é necessário um módulo capaz de excitar os transdutores com o código de Barker escolhido. Como essa excitação envolve altas tensões (em torno de 150 V) e alta frequência (de 5 MHz) o circuito não tem um desenvolvimento simples.

Para resolver este problema está em desenvolvimento também na Escola Politécnica, uma placa pulsadora/receptora. Essa placa será responsável por ser o condicionador de sinal descrito na seção 4.3. Ela produzirá o sinal de excitação do transdutor, assim como receberá o eco para depois enviá-lo para processamento. O eco do sinal transmitido volta pelo mesmo canal de envio e é então amplificado em 40dB, para então ser transmitido ao módulo de processamento. A placa tem dimensões padrão PC-104 e barramento padrão ISA para comunicação com a placa que contém o circuito de processamento, e o próprio PC-104. A figura 4.17 mostra um diagrama de blocos dessa placa.

As características deste módulo são:

- 4 Canais para conexão de transdutores, multiplexados em uma saída.
- Conectores Coaxiais.
- Transistores de alta velocidade e alta isolamento Supertex modelo MD1210.
- Gate Driver Supertex modelo TC6320, para acionamento dos transistores.

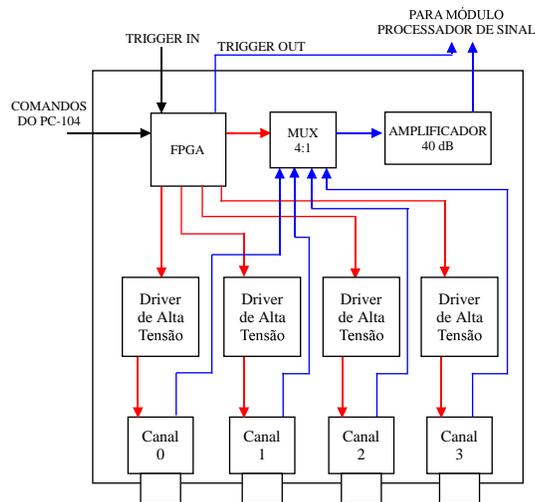


Figura 4.17: Diagrama de blocos da placa pulsadora/receptora

- Alimentação de alta-tensão externa.
- Trigger IN/OUT comum a cada quatro canais, para debug.
- Ganho de amplificação do sinal de RF configurável.
- FPGA que controla a excitação.

O funcionamento da placa é o seguinte: suponha que se queira disparar o transdutor do canal 3; a sequência de comandos que deve ser dada para os disparos deve partir do PC-104, que deve programar os registradores internos da FPGA do módulo Pulsador-Receptor com o código a ser pulsado, qual dos canais será pulsado e executar o comando de start, para início das transmissões.

A partir deste instante, o Microcomputador PC-104 só entra em ação quando quiser parar as transmissões, modificar o formato da onda ou ainda trocar de canal. Quem controla os disparos e também a transmissão do sinal de retorno para a placa é a FPGA deste módulo.

O circuito carregado na FPGA controla os disparos de alta-tensão através dos sinais de controle dos gate drivers.

Ao iniciar os disparos, a FPGA também deve direcionar o canal correto do MUX, para que o eco do sinal transmitido possa ser recebido e amplificado.

Para sincronismo entre os módulos, poderão ser utilizados recursos de software ou sinais de TRIGGER IN e TRIGGER OUT, muito comuns em aplicações deste tipo.

5 Testes reais

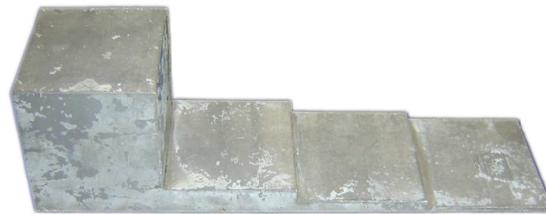
Os testes reais foram divididos em duas etapas. Na primeira etapa foi utilizado um corpo de prova mais simples (denominado corpo de prova 1) para se obter resultados preliminares sobre o algoritmo. Na segunda etapa foi utilizado um corpo de prova mais complexo (denominado corpo de prova 2) que nada mais é que um pedaço aplainado de oleoduto real, corroído. Neste momento é importante ressaltar que todos os testes foram feitos com água, e não petróleo. Ondas acústicas tem o comportamento muito semelhante nestes dois líquidos. A diferença que pode ser encontrada no petróleo é a adição de ruído nos sinais devido a particulados, mas como vimos o algoritmo reage bem mesmo com altos níveis de ruído. A escolha da água visou a grande facilidade na hora de realizar os testes, principalmente na manipulação dos corpos de prova, e também o fato que não seria fácil ter acesso ao óleo.

5.1 Primeira etapa

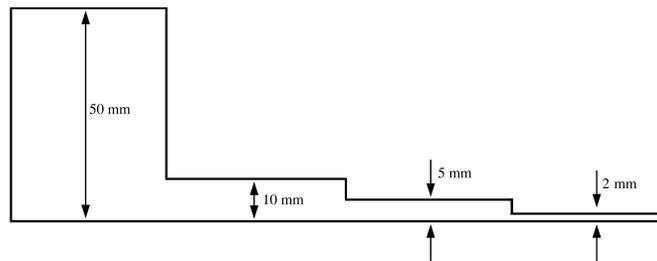
Na primeira etapa, o sistema utilizado para gerar os pulsos é composto por um amplificador de potência (Amplifier Research, EUA) modelo 150A100A para excitar os transdutores (enquanto a placa pulsadora não está disponível), e um gerador de funções (Tektronix, EUA) modelo AFG 3102 para gerar os códigos de excitação que devem ser enviados. Os ecos então são adquiridos e digitalizados por um Osciloscópio digital (Agilent, EUA) modelo DSO 6032A, para posteriormente serem analisados *off-line* em MATLAB (Mathworks, EUA). Com isso foi possível analisar o algoritmo sem a necessidade do sistema estar completo.

5.1.1 Corpo de Prova 1

O corpo de prova para os testes reais é uma peça de alumínio usinada em degraus. A figura 5.1 mostra essa peça. Ela será utilizada para a primeira etapa pois como suas dimensões são conhecidas pode-se calcular a eficiência do algoritmo em funcionamento. O seu formato em degraus nos permite testar várias situações, inclusive com sobreposição dos sinais para os degraus de menor espessura.



(a)



(b)

Figura 5.1: Corpo de prova de alumínio, usinado como uma escada com diversas espessuras de degraus. (a) Imagem real e (b) suas dimensões

5.1.2 Resultados

Foram feitos testes de detecção utilizando os códigos definidos na tabela 4.2. Os sinais digitalizados pelo osciloscópio foram transferidos para um computador. Em MATLAB os sinais foram filtrados por filtros casados cujas funções de transferência eram os referidos códigos.

A tabela 5.1 mostra as espessuras que foram medidas com cada código para cada degrau do corpo de prova. O sinal resultante do filtro casado foi analisado visualmente

para detecção dos picos corretos de cada eco, e conseqüentemente, o instante em que os mesmos ocorreram. Neste caso, não foi utilizado o algoritmo de detecção de picos, para que o resultado fosse apenas influenciado pelo código escolhido. Como já foi dito, a detecção de picos ainda pode ser melhor desenvolvida de modo a detectar com maior certeza os picos corretos.

Tabela 5.1: Resultados (em *mm*) dos testes com o corpo de prova de alumínio.

Código	Degrau 1 (50 mm)	Degrau 2 (10 mm)	Degrau 3 (5 mm)	Degrau 4 (2.0 mm)	Erro médio
2x3	50,00	9,95	5,07	2,63	10,8 %
2x8	50,00	10,59	5,13	5,78	65,8 %
3x3	50,00	9,95	4,49	4,43	36,9 %
3x4	50,00	9,95	5,13	5,00	50,7 %
7x1	50,00	9,95	5,07	2,05	1,1 %
7x2	50,00	9,95	5,07	2,05	1,1 %

O erro médio presente na tabela 5.1.2 foi calculado como a média dos erros percentuais para os degraus de 10, 5 e 2 mm

Para os 2 primeiros degraus, as medições foram praticamente perfeitas, com todos os códigos chegando no mesmo resultado (exceção do código de 2 bits, 8 ciclos por bit, que teve diferença de apenas uma amostra no sinal digitalizado). O primeiro degrau foi utilizado como base para definir a velocidade da onda no alumínio para então calcular as outras distâncias.

Já no terceiro degrau (5 mm), as diferenças foram um pouco maiores, principalmente para o código de 2 bits e 8 ciclos por bit, que chegou a 15% de diferença para o esperado. Ainda assim, são resultados muito bons para uma inspeção, que poderiam fornecer informações muito precisas sobre o estado da tubulação. A figura 5.2 mostra um exemplo do sinal resultante da inspeção (7 bits, 2 ciclos, no degrau de 10 mm), e a saída do filtro casado para este caso.

Percebe-se pela figura 5.2 que os ecos ficam muito claros na saída do filtro, e que para encontrar os TOAS não seria uma tarefa difícil.

Já para o último degrau, os resultados divergiram bastante, tornando-se bem inconclusivos. Isso ocorreu devido a grande sobreposição dos ecos, impossibilitando que até mesmo visualmente estes fossem distinguidos. Apenas para os casos de 7 bits o resultado foi coerente, mostrando que apesar da sobreposição a inversão de bits ajudou na detecção. Ainda assim a distinção dos ecos não foi tão clara. A figura 5.3 mostra essas saídas para

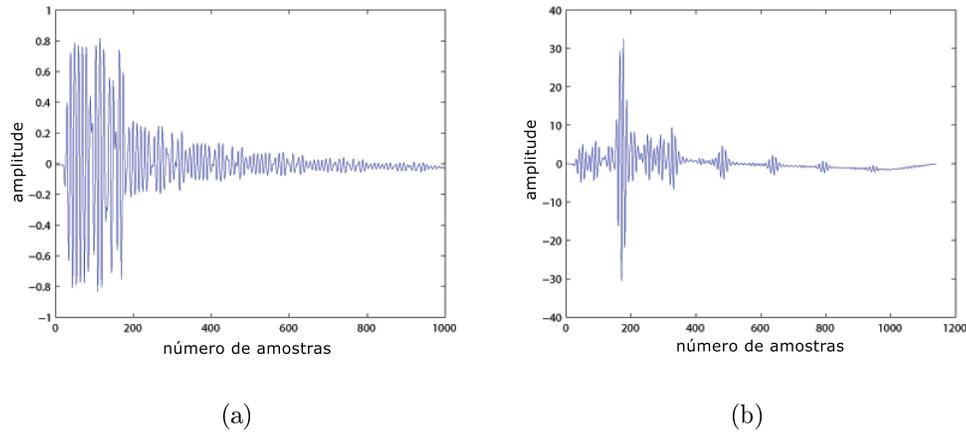


Figura 5.2: (a) Sinal adquirido na inspeção do corpo de prova 1, para o caso de 7 bits, 2 ciclos por bit, no degrau de 10 mm; e (b) Saída do filtro, com picos bem definidos, resultando num bom resultado na inspeção

7 bits e a localização do primeiro e segundo eco (pontos demarcados com "o"). Os picos laterais à localização do eco podem ser facilmente confundidos com o eco verdadeiro.

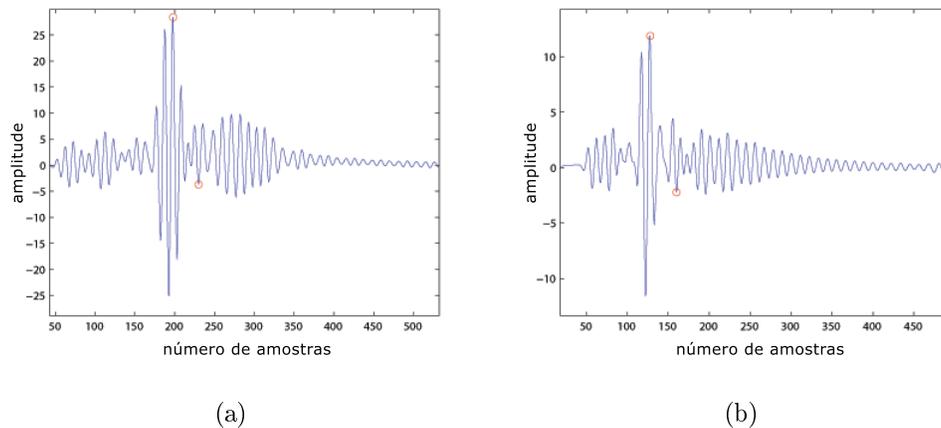


Figura 5.3: Saída do filtro casado para o caso de: (a) 7 bits, 1 ciclos por bit e (b) 7 bits, 2 ciclos por bit, ambos na espessura de 2 mm

Para os outros códigos a situação se torna ainda pior. A figura 5.4 mostra a saída no caso de 3 bits e 3 ciclos.

O primeiro e segundo eco são demarcados com "o". A localização do segundo eco não é nada trivial. Mesmo visualmente fica difícil determinar essa localização. As saídas para os outros códigos seguem este mesmo padrão.

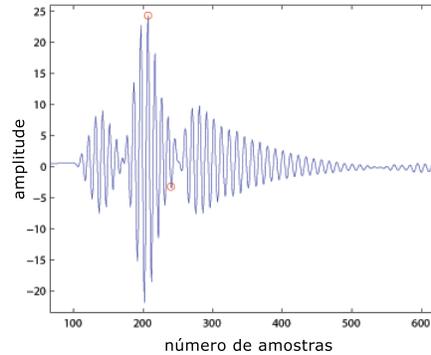


Figura 5.4: Saída do filtro casado para o caso de 3 bits, 3 ciclos por bit na espessura de 2 mm

Foi percebido que sinais com comprimento maior tendem a apresentar resultados piores que os de comprimento menor. Como neste trabalho deseja-se medir espessuras até 5 mm, os resultados foram promissores, porém para a próxima etapa serão eliminados os códigos de maior comprimento para cada número de Bits. Assim serão utilizados apenas os códigos:

- 2 Bits, 3 Ciclos
- 3 Bits, 3 Ciclos
- 7 Bits, 1 Ciclo

5.2 Segunda Etapa

Para a segunda etapa de testes já foi utilizado o sistema de processamento digital de sinais em tempo real, para além de testar o algoritmo, testar o sistema de inspeção em si. A placa pulsadora ainda não ficou pronta, e por isso o sistema de excitação dos transdutores ainda é composto pelo amplificador e pelo gerador de funções. Neste novo arranjo, o sistema de inspeção é controlado pelo PC-104, através de um software desenvolvido em linguagem C++. O arranjo de inspeção pode ser visualizado no diagrama da figura 5.5.

A utilização do amplificador de sinais neste arranjo, além de cabos longos ligando todos os equipamentos, colaborou para a adição de ruídos eletrônicos nas inspeções, o que atrapalha na eficiência dos testes. Além disso, utilizou-se um separador de sinais, que é apenas um circuito para separar os sinais de alta tensão dos ecos que serão recebidos

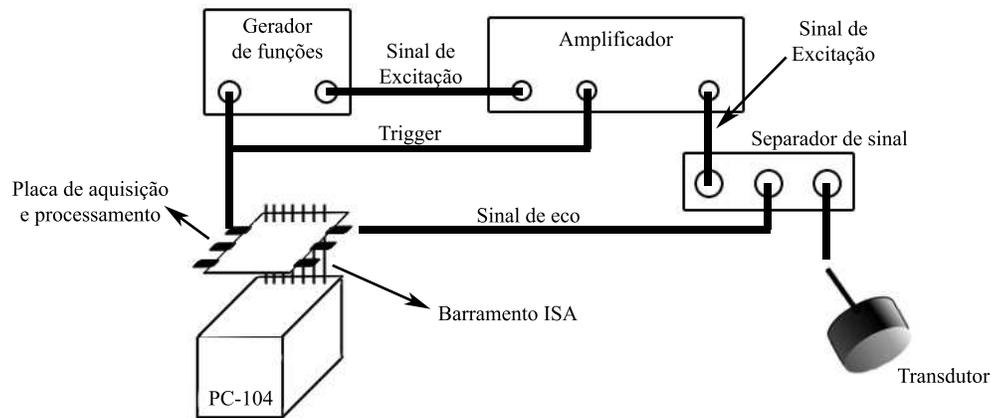


Figura 5.5: Diagrama representando o sistema de inspeção utilizado na segunda etapa de testes

(de baixa voltagem) e proteger os equipamentos. O esquemático deste circuito pode ser observado na figura 5.6. Vale lembrar que quando a placa pulsadora estiver pronta, boa parte destes ruídos poderão ser fortemente reduzidos.

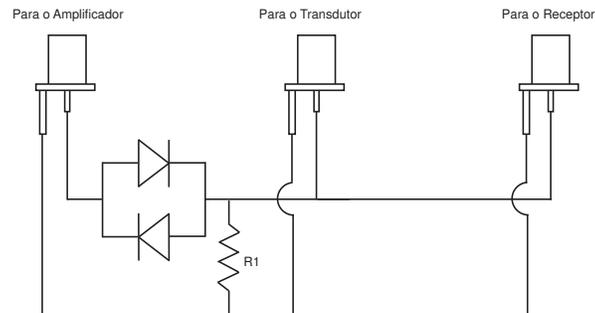


Figura 5.6: Circuito utilizado para o separador de sinais, com $R1 = 100K\Omega$ e diodos rápidos 1N4148

5.2.1 Corpo de Prova 2

Como dito, o segundo corpo de prova consiste de um pedaço real de oleoduto corroído, aplainado em forma de uma placa para facilitar sua inspeção em laboratório. A figura 5.7 mostra a placa. Com este corpo de prova é possível se aproximar de uma situação real de trabalho de um FIG.

O corpo de prova será inspecionado em apenas algumas das linhas definidas na figura 5.7. Porém, para que se possa realmente saber a eficiência do algoritmo nesse caso, é necessário conhecer as dimensões reais da placa corroída, para que se possa comparar os

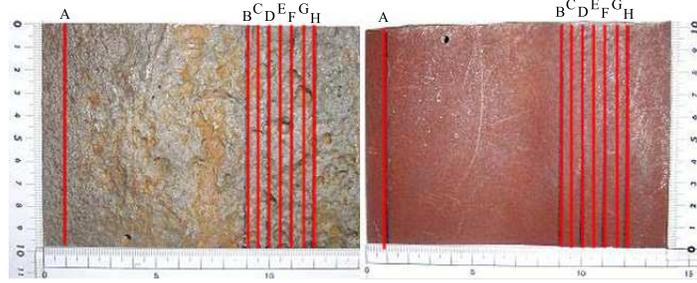


Figura 5.7: Imagens da placa utilizada como corpo de prova nos experimentos

resultados obtidos na inspeção por ultrassom. Para isso a placa foi escaneada em uma máquina de medição 3D, nas linhas definidas na figura 5.7.

5.2.2 Escaneamento 3D

A máquina utilizada para o escaneamento 3D é uma Mitutoyo modelo BN710 (Mitutoyo, Japão) disponível na Escola Politécnica. Esta é uma máquina de medição de coordenadas que conta com uma sonda de rubi (*probe*) suportada por pórticos que se movimentam através de um sistema pneumático. Com isso a sonda de rubi pode se movimentar nos 3 eixos, e a inspeção é feita através do contato da mesma com a peça. No momento de toque da sonda na peça, a máquina salva a coordenada, e realizando essa operação em diversos pontos da peça, é feito o escaneamento. A máquina tem uma precisão de $10\ \mu\text{m}$, e com esta precisão é válido tomar o resultado do escaneamento como parâmetro para comparação e eficiência da inspeção ultrassônica. A figura 5.8 mostra a máquina.



Figura 5.8: Foto da máquina BN710 da Mitutoyo

A figura 5.9 mostra o resultado do escaneamento.

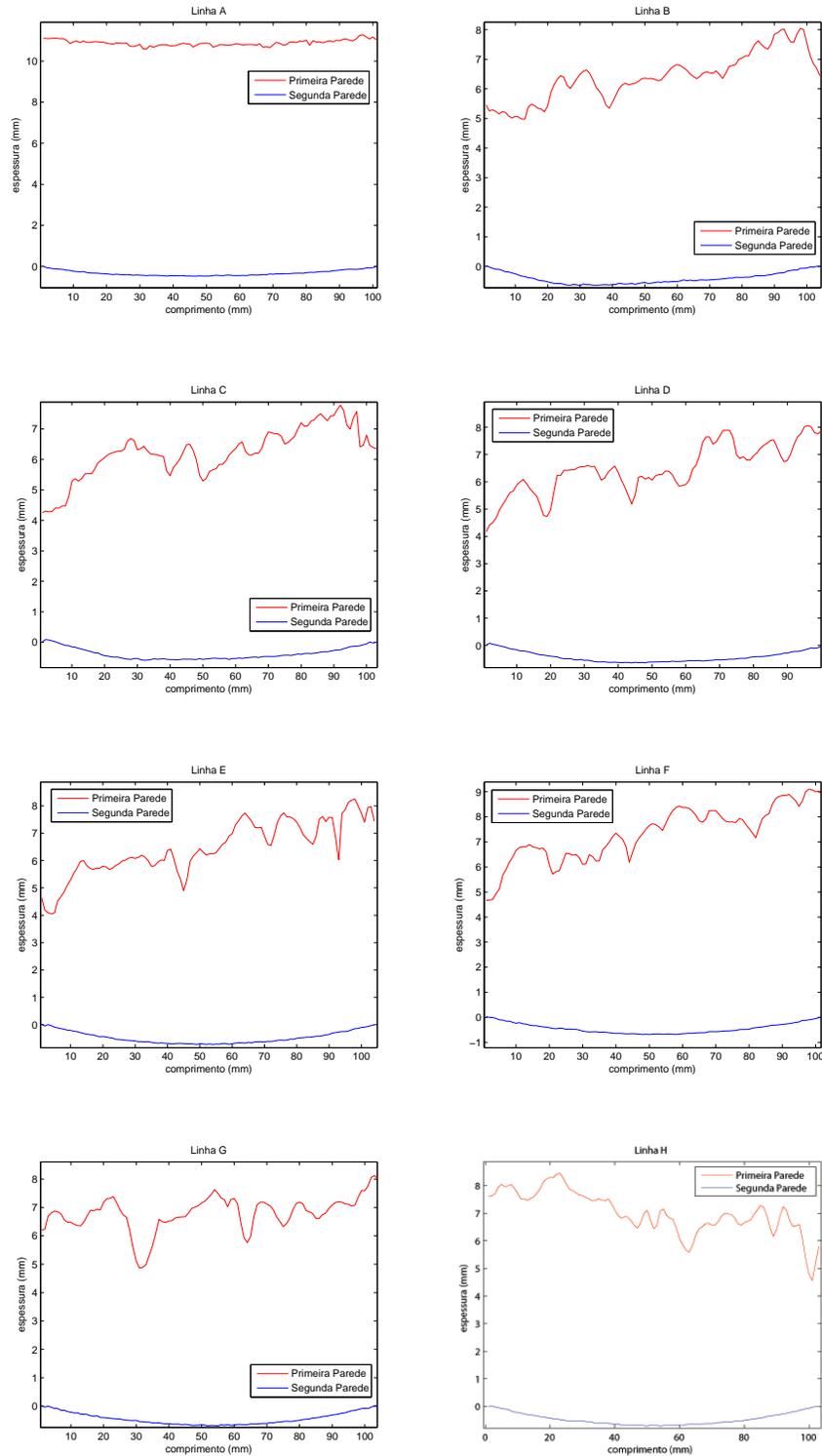


Figura 5.9: Linhas escaneadas pela máquina 3D, reproduzidas em MATLAB

5.2.3 Inspeção ultrassônica

O corpo de prova foi inspecionado utilizando os 3 códigos definidos na seção 5.1.2, nas linhas A, B, D e E. Na inspeção utilizou-se um sistema mecânico para realizar a varredura

da peça fazendo a inspeção com o transdutor ultrassônico. A figura 5.10 mostra uma foto e um diagrama deste sistema com a peça a ser escaneada.

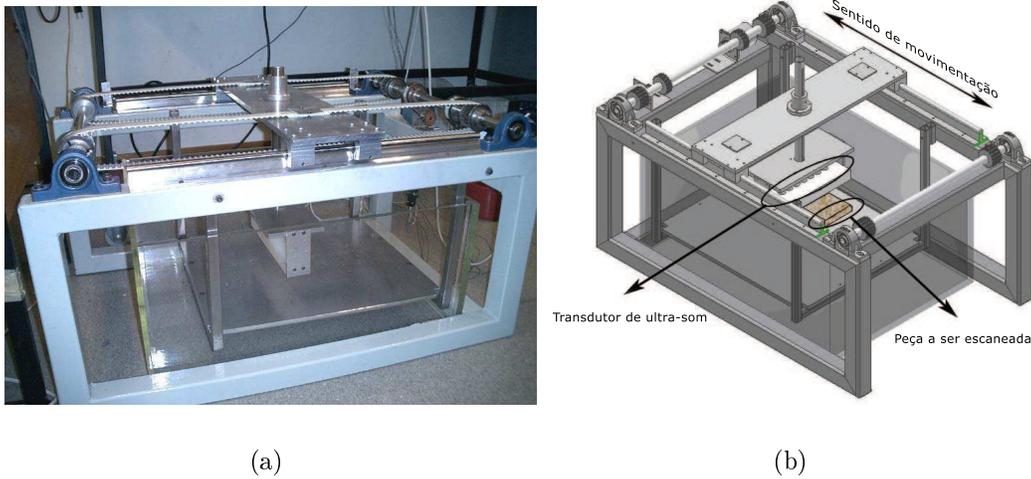


Figura 5.10: (a) Foto do sistema mecânico; (b) Diagrama mostrando o sistema de inspeção e o posicionamento da peça a ser escaneada

Um grande problema desse sistema é o alinhamento da peça com o trajeto do transdutor. O posicionamento da peça é feito manualmente para cada linha, e o alinhamento da linha com o trajeto é feito visualmente. A figura 5.11 exemplifica esta situação.

Essa abordagem “manual” e “visual” pode gerar erros de qual a região específica que se está realmente inspecionando.

Os resultados desta inspeção serão discutidos no capítulo a seguir.

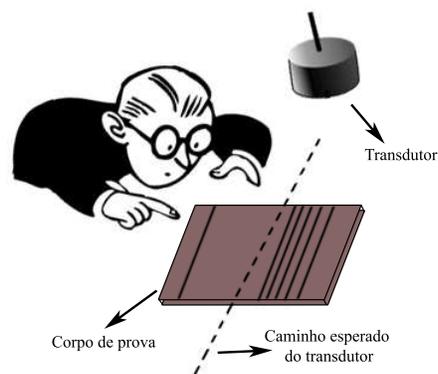


Figura 5.11: Diagrama exemplificando o alinhamento visual que foi feito durante a inspeção

6 Resultados

Este capítulo apresentará os resultados das inspeções com o sistema digital de inspeção, e as comparações com a inspeção 3D. Os resultados serão divididos em primeira e segunda parede, e serão analisados os problemas encontrados na inspeção.

6.1 Método de comparação

A inspeção 3D gerou curvas em um espaço tridimensional, e para compará-las com a inspeção ultrassônica, foi necessário tratar os dados. Em primeiro lugar, as curvas no espaço 3D foram reduzidas ao espaço 2D (espessura e comprimento da linha inspecionada). Porém, ainda assim a comparação não era possível, pois a inclinação da peça durante a inspeção 3D poderia não ser a mesma da inspeção por ultrassom. Para resolver este problema, definiu-se:

- A peça tem uma parede reta, que foi considerada como o ponto inicial das linhas escaneadas, para ambas inspeções. A figura 6.1 mostra essa etapa.

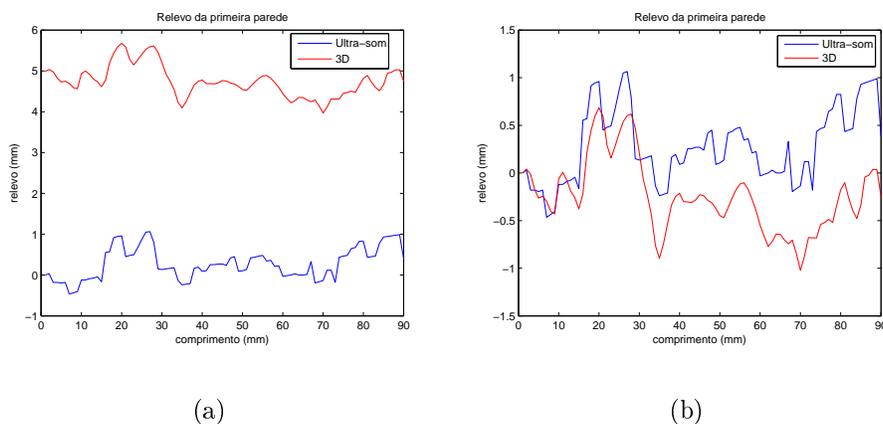


Figura 6.1: (a) Resultados das inspeções soltos no espaço, e (b) resultados com primeiro ponto igualado

- Agora é necessário rotacionar um dos resultados, de modo a colocá-los finalmente num mesmo sistema de referência. Para realizar essa etapa definiu-se visualmente mais um ponto comum entre os dois resultados, que melhor parecesse aproximar as linhas. Tendo estes dois pontos de referência pode-se verificar qual o ângulo que era necessário rotacionar a linha. A figura 6.2 exemplifica essa situação.

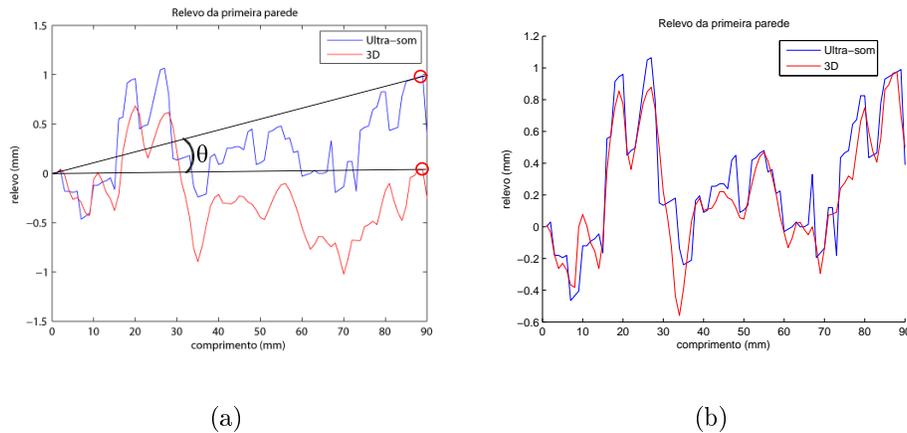


Figura 6.2: (a) Resultados com diferença angular, e (b) resultados alinhados prontos para inspeção

Com isto definido, aplicou-se este algoritmo para todas as linhas inspecionadas. A linha base (fixa) é a linha da inspeção por ultrassom (já que a mesma é gerada em 2 dimensões), e a linha que é rotacionada é a linha 3D.

Dessa maneira tornou-se possível comparar os dois resultados.

6.2 Medição da Primeira Parede

Para uma boa análise e entendimento dos resultados (tanto da primeira parede quanto da segunda) é importante lembrar que os métodos utilizados para a inspeção por ultrassom (principalmente de alinhamento manual e visual da peça) eram passíveis de gerar alguns erros, inclusive o método utilizado para fazer a comparação dos resultados ultrassônicos e da medição 3D. Levando em conta estes fatos já é de se esperar que os resultados não sejam perfeitos.

Para medir o quanto os resultados foram próximos calculou-se a covariância entre as duas paredes. A tabela 6.1 mostra este coeficiente para as linhas inspecionadas, para cada código.

Tabela 6.1

Linha	2 bits, 3 ciclos	3 bits, 3 ciclos	7 bits, 1 ciclo
A	0.0078	0.0180	0.0184
B	0.1298	0.1280	0.1281
D	0.4885	0.4710	0.4163
E	0.2581	0.2559	0.3026
Média	0.2210	0.2182	0.2163

A linha A mostrou valores relativamente baixos para a covariância. Observando-se a figura que mostra a comparação entre as duas inspeções para essa linha percebe-se que os resultados não ficam muito parecidos. Isso acontece porque nessa região a peça demonstrava pouquíssima corrosão, com variações de espessura muito pequenas porém ocorrendo em muita quantidade, tanto no comprimento da linha quanto perpendicularmente à mesma. Aliando isso ao método de inspeção e alinhamento utilizado é difícil ter certeza que se está comparando exatamente a mesma linha da peça. Para as outras regiões mais corroídas, as variações laterais são menores e ocorrem mais suavemente, apesar da espessura variar muito mais, e por isso aumenta a probabilidade de se estar comparando a mesma região.

Pela covariância percebe-se que o código que apresentou o melhor resultado na média foi o de 2 Bits, 3 Ciclos. Mais uma vez percebe-se que para os sinais menores os resultados são melhores, principalmente por causa da sobreposição dos sinais.

A seguir são mostrados alguns gráficos comparativos da primeira parede resultante da inspeção ultrassônica e da inspeção feita com a máquina 3D.

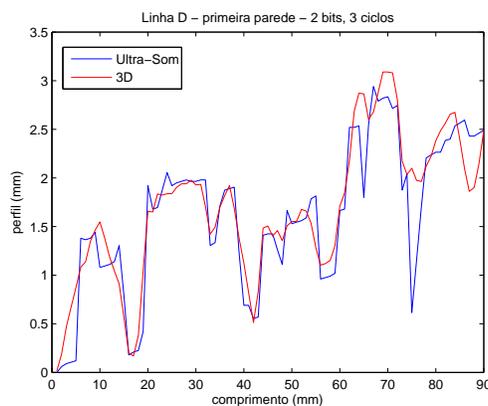


Figura 6.3: Comparação entre os resultados da primeira parede para a linha D, com o código 2 Bits, 3 ciclos

Na figura 6.3 o resultado da inspeção por ultrassom ficou extremamente próximo do 3D. As diferenças são menores de 1 *mm* na maioria dos pontos. Dois pontos aparecem como buracos na inspeção por ultrassom (aos arredores dos pontos 65 e 75 *mm* do comprimento). No primeiro caso tivemos um problema de detecção do ciclo errado, e no segundo um problema de primeiro eco corrompido. Próximo do ponto 87 *mm* a inspeção não conseguiu detectar corretamente um buraco no perfil. Este erro ocorre devido a uma limitação do sistema para detectar *pites*. Todos estes erros serão explicados no item 6.2.1.

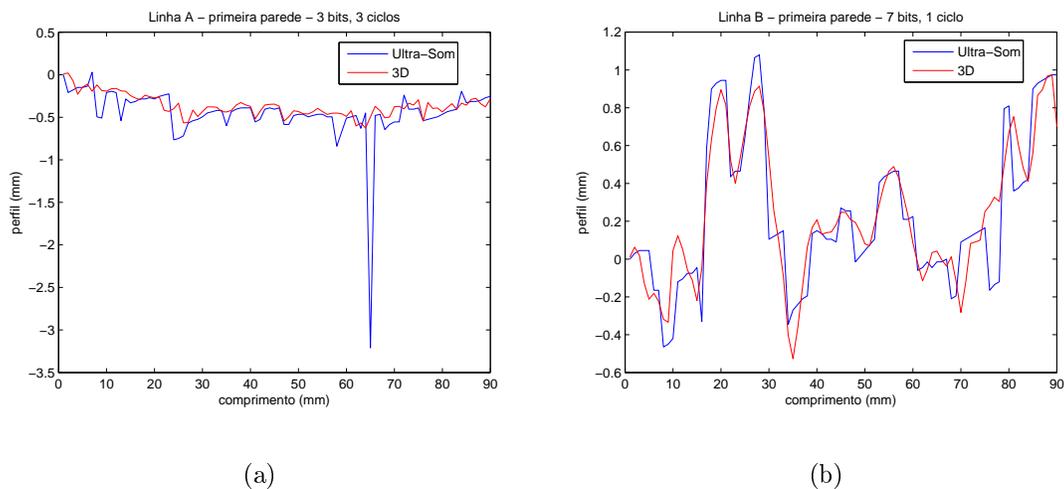


Figura 6.4: Comparação entre os resultados da primeira parede para (a) linha A, com o código 3 Bits, 3 ciclos, e (b) linha B, 7 Bits, 1 ciclo

Na figura 6.4.a os resultados também são muito próximos, porém, como foi explicado, a inspeção da linha A é mais complicada. Ao redor do ponto 66 *mm* do comprimento, a inspeção ultrassônica falhou bastante. Este erro é semelhante ao erro do ponto 87 *mm* do comprimento da figura 6.3. A figura 6.4.b também mostra resultados muito bons, e erros semelhantes aos já descritos.

Considerando os aspectos mencionados para análise dos resultados, e analisando-se as figuras, pode-se perceber que a detecção da primeira parede apresentou ótimos resultados na maioria dos pontos. Esse já é um ponto muito importante que valida a implementação do algoritmo, pois tendo-se certeza do estado da primeira parede já se consegue ter uma boa noção do nível de corrosão do duto, e para regiões muito corroídas, nem é preciso ter um resultado preciso da segunda parede, pois já fica claro que é necessário alguma ação de correção com aquela parte do duto.

6.2.1 Principais problemas da inspeção

Apesar dos resultados terem sido considerados promissores, deve-se chamar atenção para os principais erros que ocorreram na detecção da primeira parede.

- Corrosão por pites
- Detecção do ciclo errado
- Sistema de detecção:
 - Amplitude de ecos laterais
 - Sistema e Algoritmo

Estes erros serão explicados a seguir:

6.2.1.1 Corrosão por pites

A corrosão por pites ocorre quando temos uma corrosão concentrada em um ponto da peça, resultando em um “buraco”. Como o sinal de ultrassom não é um feixe estreito de som, mas sim uma onda que se espalha para todas as direções, ao encontrar um pite é possível que esta onda seja refletida com muita intensidade antes que parte da mesma chegar até o fim do buraco. Com isso, ao se receber o eco acredita-se que este seja da parte inferior do buraco, quando na verdade é resultado de reflexões nas paredes do mesmo. A figura 6.5 mostra essa situação.

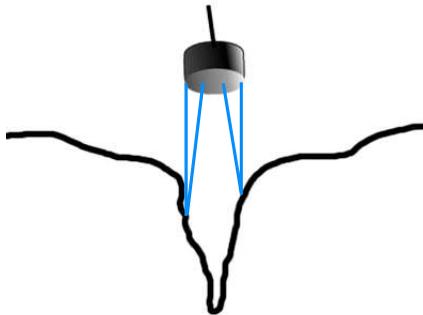


Figura 6.5: Onda de ultrassom refletindo nas paredes laterais de um pite, antes de chegar no fundo do mesmo

A figura 6.6 mostra dois casos onde isso ocorreu na inspeção da corpo de prova 2.

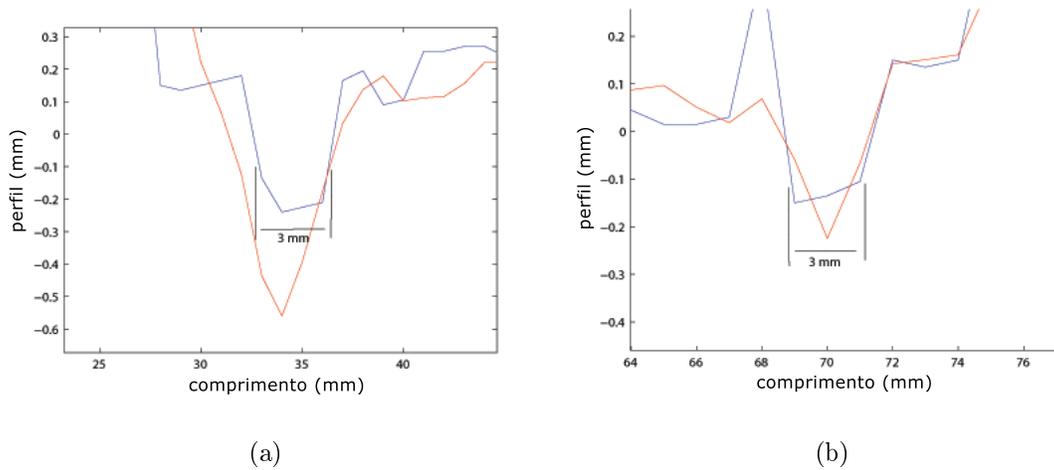


Figura 6.6: Dois exemplos onde o pite não foi corretamente detectado, aparecendo como um buraco menor que o buraco real

O pite geralmente aparece na inspeção como um buraco de menor profundidade, com um patamar em seu fundo. Analisando-se todos os resultados da inspeção, percebeu-se que com este tipo de transdutor consegue-se medir pites de até no máximo 3 *mm* de diâmetro.

6.2.1.2 Detecção do ciclo errado

Como já explicado, o sinal de ultrassom é um sinal com uma determinada frequência, e também contém diversos ciclos. Esses ciclos ocorrem também na saída do filtro casado. Por diversos motivos, inclusive pela amostragem do sinal, um ciclo pode ter uma amplitude um pouco maior/menor que o ciclo anterior/posterior, e algumas vezes um pico que foi detectado como o *TOA* do sinal pode na verdade ser o pico anterior ou posterior ao pico do verdadeiro *TOA*. Essa diferença de apenas um ciclo no sinal irá representar um determinado espaço percorrido pela onda, dependendo da velocidade da mesma no material. No caso de inspeções de duto, considerando a velocidade do som no metal como 6200 *m/s* e o transdutor de 5 *MHz*, um ciclo irá representar uma diferença de 1,24 *mm*. Essa diferença já é consideravelmente grande para casos em que se está medindo espessuras de aproximadamente 5 *mm*. Na figura 6.7 pode-se observar uma situação na qual o ciclo anterior ao *TOA* real ficou mais proeminente, causando a detecção errada da primeira parede.

Nessa figura percebe-se outra situação na inspeção, que é o caso de sinais saturados.

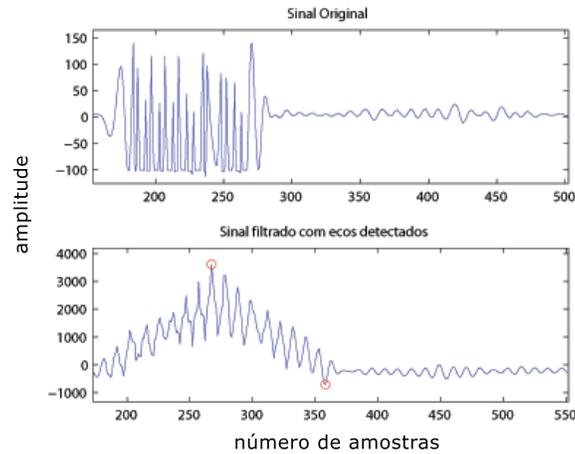


Figura 6.7: Sinal de eco do ponto 60 mm do comprimento, da linha A, para o código de 3 bits, 3 ciclos. E ponto detectado pertence a um ciclo anterior ao ciclo do TOA real do eco.

Sem o filtro casado ficaria muito difícil detectar o TOA do sinal saturado, porém a saída do filtro, apesar de apresentar um formato estranho, deixa clara a localização do sinal.

Alguns erros são gerados pelas características do sistema e tipo de inspeção:

6.2.1.3 Amplitude de ecos laterais

Devido a grande corrosão em alguns pontos da peça, a onda de ultrassom pode sofrer um grande espalhamento, e o eco recebido pelo transdutor conter pouca energia. Isso irá gerar ecos com baixa amplitude, e conseqüentemente a saída do filtro casado ter baixa amplitude. Isso não seria problema se o transdutor recebesse apenas o eco daquele ponto, porém o transdutor está recebendo ecos de diversos pontos da peça, e algum destes pontos pode estar produzindo um eco com maior amplitude que o do ponto que se está teoricamente inspecionando. Isso irá gerar pontos errados durante a inspeção, porém é um erro inevitável. Uma possível maneira de diminuir seus efeitos é colocar o transdutor mais perto do objeto a ser inspecionado (respeitando uma distância mínima para não haver grande sobreposição, como definido na seção 4.2.2.1), diminuindo a presença de ecos laterais, e utilizar transdutores focais, cujo ponto focal esteja no objeto (lembrando que como mostrado na seção 3.1.4.2, o uso de transdutores focais tem seus pontos negativos).

6.2.1.4 Sistema e Algoritmo

Além de todos os erros apresentados, o próprio algoritmo desenvolvido é passível de outros erros. O sistema ainda não é o sistema final que será utilizado em *pigs*, e está sujeito a erros eletrônicos e mecânicos. Na figura 6.8 vemos que o primeiro eco está corrompido. Quando o sinal é passado pelo filtro, o primeiro eco fica com uma amplitude menor que o segundo eco, e por isso o segundo é detectado como se fosse o primeiro, gerando um resultado errado em relação às paredes do duto. Este erro é o que pode ser visto na figura 6.4.a no ponto 65 *mm* do comprimento.

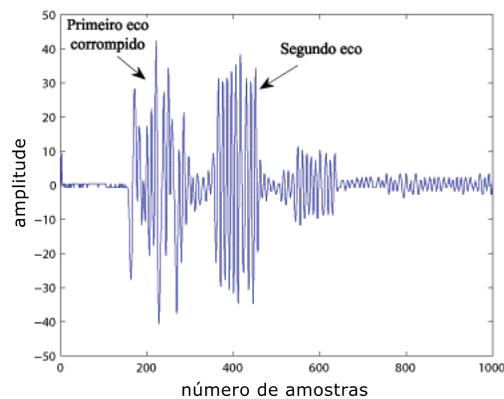


Figura 6.8: Sinal com seu primeiro eco corrompido

Este tipo de erro aconteceu algumas vezes durante as inspeções (cerca de 2% das mesmas). Como as inspeções foram feitas manualmente, com alinhamento visual e outras dificuldades já comentadas, era inviável repetir a inspeção para procurar o mesmo erro e estudar o motivo do mesmo. Este tipo de falha não é reprodutível, porém como sua ocorrência é baixa, e trata-se de um protótipo decidiu-se não estudar este erro. Acredita-se que seja um erro mecânico durante a movimentação do transdutor, que recebe o primeiro eco durante algum tipo de vibração, e corrompe a recepção.

O algoritmo também tem suas limitações. A detecção de picos, que tira primeiro a envoltória do sinal visando eliminar alguns picos que sabe-se não serem os picos do TOA, pode eventualmente eliminar algum pico verdadeiro, tanto do primeiro, quanto do segundo eco. Outra limitação é a eficiência do filtro casado com sinais muito sobrepostos. Como foi visto, quando os sinais tem grande sobreposição (o que acontece principalmente devido aos ecos laterais, como explicado anteriormente), o filtro fornece resultados inconclusivos sobre a real localização do sinal.

Devido a estes principais problemas, a detecção da primeira parede não foi perfeita, porém, foi considerada como suficiente para detecção da corrosão em uma situação real.

6.3 Medição da Segunda parede

A inspeção da segunda parede encontrou um número maior de problemas que da primeira. A figura 6.9 mostra os resultados para 3 linhas.

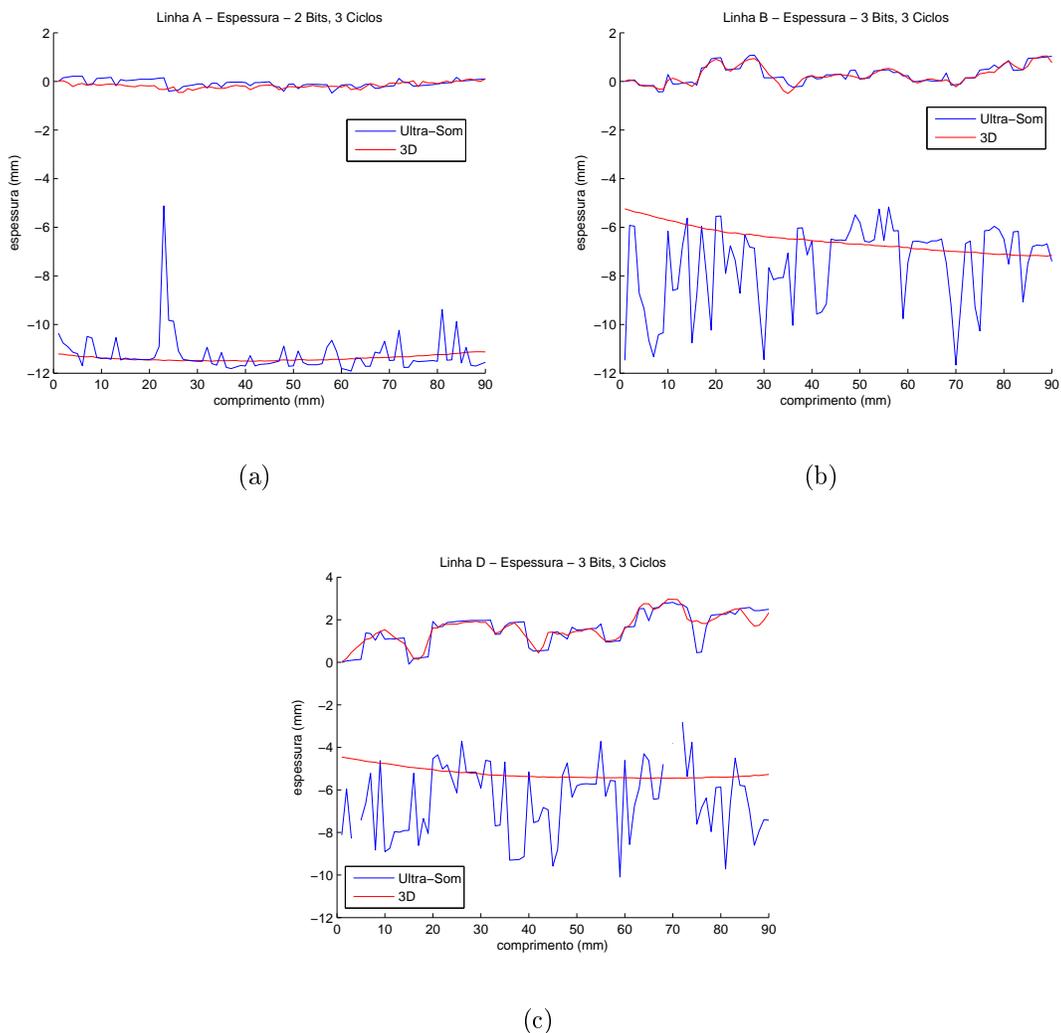


Figura 6.9: Comparação entre os resultados para a primeira e segunda parede: (a) Linha A, 2 Bits, 3 Ciclos; (b) Linha B, 3 Bits, 3 Ciclos; (c) Linha D, 3 Bits, 3 Ciclos

Como pode-se perceber, a detecção da segunda parede varia muito, gerando resultados

pouco confiáveis. Todos os problemas encontrados na detecção da primeira parede se repetem na detecção da segunda, porém novos problemas também são encontrados.

A seguir serão mostradas figuras com resultados da inspeção (primeira e segunda parede), e o sinal correspondente a um ponto inspecionado, com seus respectivos TOAs detectados pelo algoritmo.

No caso da figura 6.9.c, existem alguns pontos da segunda parede em que o algoritmo não encontrou o segundo eco, e por isso na imagem existe uma interrupção desta parede. Essa perda ocorre quando por algum motivo a envoltória não conteve picos dentro da janela na qual o segundo pico era procurado. Estes pontos são descartados no resultado da inspeção.

A figura 6.10 mostra um caso onde houve uma grande sobreposição de diversos ecos, e um segundo eco é detectado ainda dentro do primeiro, se sobrepondo ao mesmo. Na saída do filtro não é possível identificar o segundo TOA real pois a amplitude de queda da envoltória do primeiro eco (ou seja, as amplitudes dos picos dos ciclos do primeiro eco) é ainda muito maior que a amplitude do segundo.

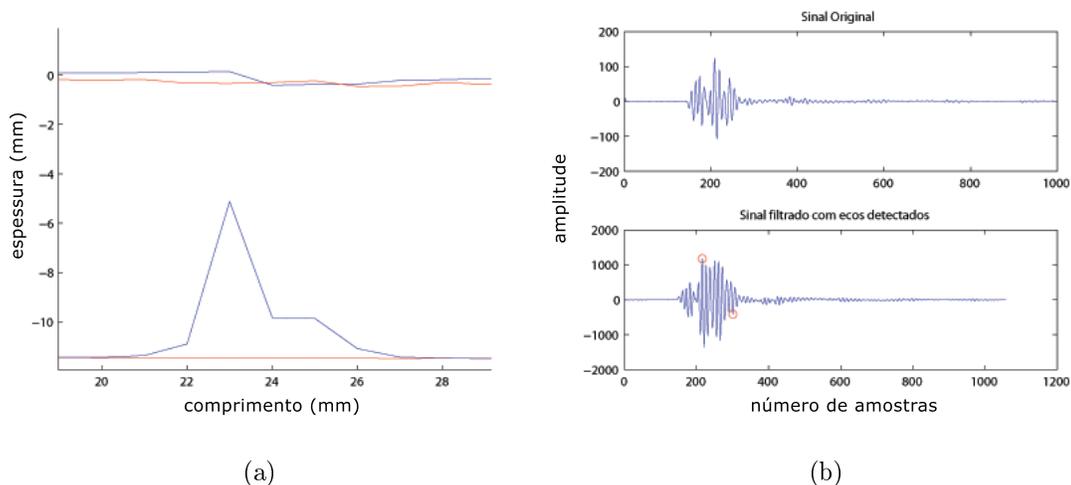


Figura 6.10: (a) Resultado da inspeção e (b) sinal para o ponto 23 mm do comprimento

A figura 6.11 mostra um caso onde o erro ocorreu porque na saída do filtro o ciclo

anterior ao ciclo real do TOA ficou mais proeminente, causando um erro de alguns milímetros na espessura.

Já a figura 6.12 mostra um caso no qual a amplitude de saída do segundo eco no filtro foi muito baixa (o sinal foi atenuado demais devido ao espalhamento causado pela corrosão). Devido a essa baixíssima amplitude, um outro ponto acabou sendo detectado mais adiante.

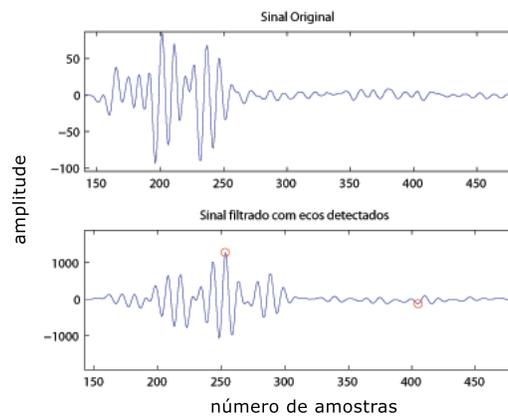


Figura 6.11: Sinal para o ponto 24 *mm* do comprimento

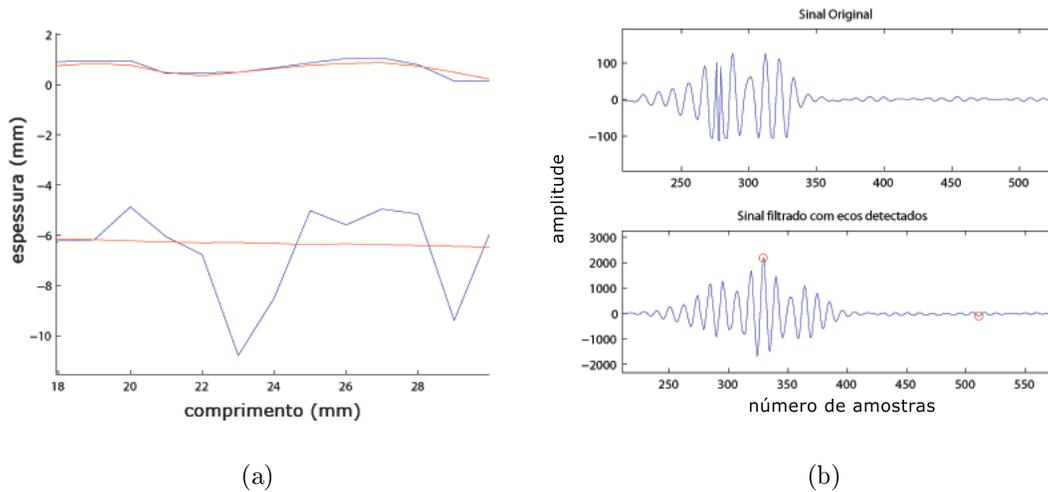


Figura 6.12: (a) Resultado da inspeção e (b) sinal para o ponto 23 *mm* do comprimento

No caso da figura 6.13 houve o mesmo problema onde o primeiro eco foi corrompido.

Dessa maneira o sistema capta o segundo eco como primeiro, e o terceiro como segundo, fazendo com que as paredes apareçam distorcidas, porém a espessura é detectada corretamente.

A figura 6.14 mostra um caso onde o sinal original foi saturado, gerando uma saída diferente no filtro. Mesmo assim o filtro funciona corretamente, porém o eco saturado gerou um pico espúrio na saída que foi erroneamente detectada como segundo TOA.

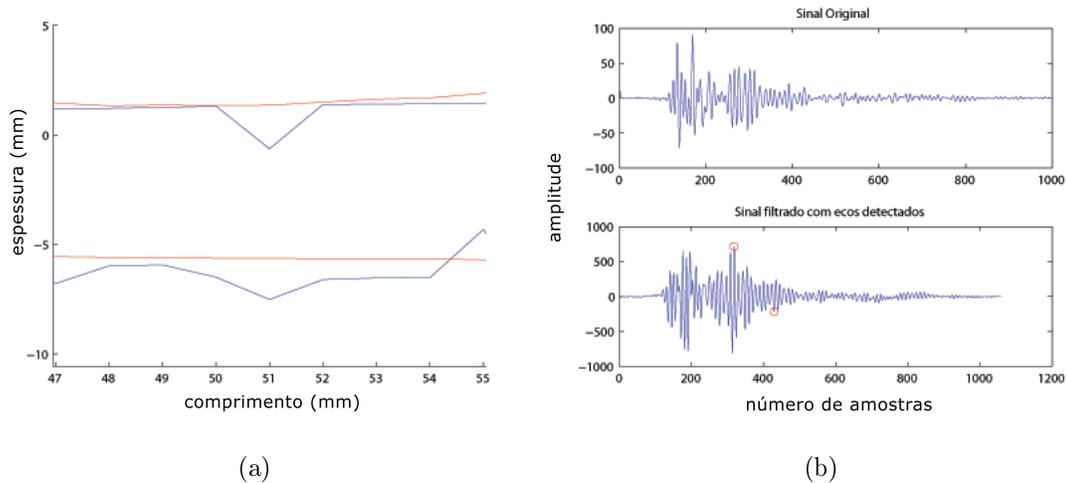


Figura 6.13: (a) Resultado da inspeção e (b) sinal para o ponto 51 *mm* do comprimento

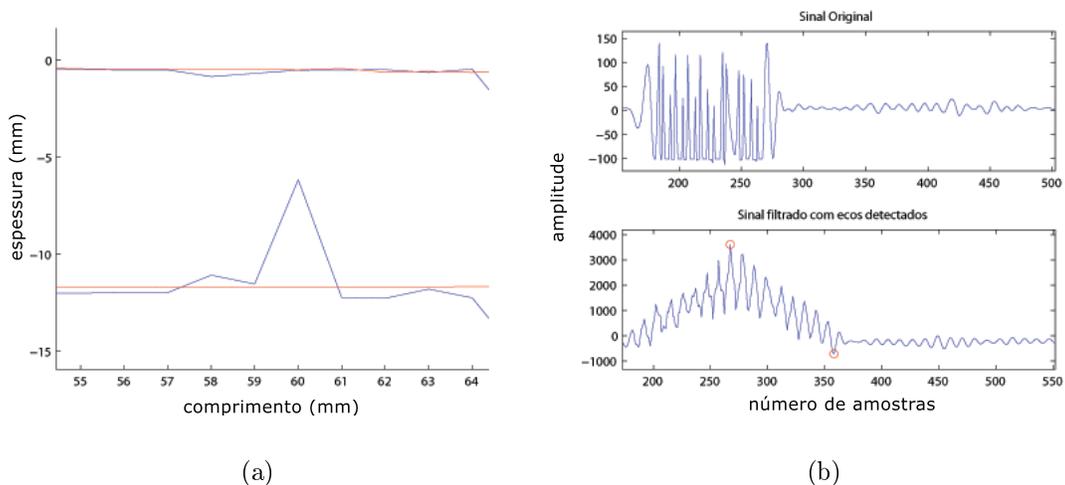


Figura 6.14: (a) Resultado da inspeção e (b) sinal para o ponto 60 *mm* do comprimento

Na figura 6.15 o segundo eco também se sobrepõe ao primeiro, e o detector acabou

pegando um ponto falso mais adiante no sinal.

A figura 6.16 mostra um caso onde o algoritmo e o filtro funcionaram muito bem, e a detecção dos dois TOAs foi feita corretamente.

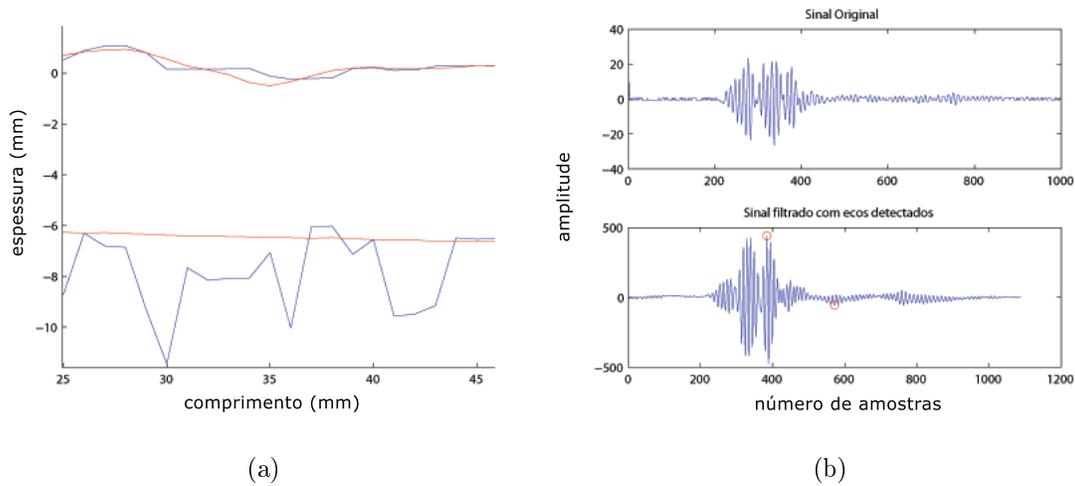


Figura 6.15: (a) Resultado da inspeção e (b) sinal para o ponto 30 *mm* do comprimento

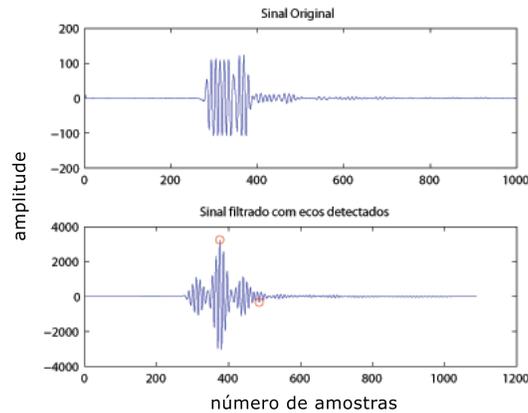


Figura 6.16: Sinal para o ponto 45 *mm* do comprimento

No caso da figura 6.17 o segundo eco é praticamente inexistente, e dessa maneira fica muito difícil mesmo visualmente de detectá-lo. Nestes casos onde a atenuação do sinal é muito grande fica impossível fazer a detecção corretamente.

Por último tem-se a figura 6.18 que mostra um caso no qual devido os parâmetros do detector de picos de detectar primeiro as envoltórias, o algoritmo acaba não encontrando nenhum segundo pico. Por isso o mesmo recebe o valor zero.

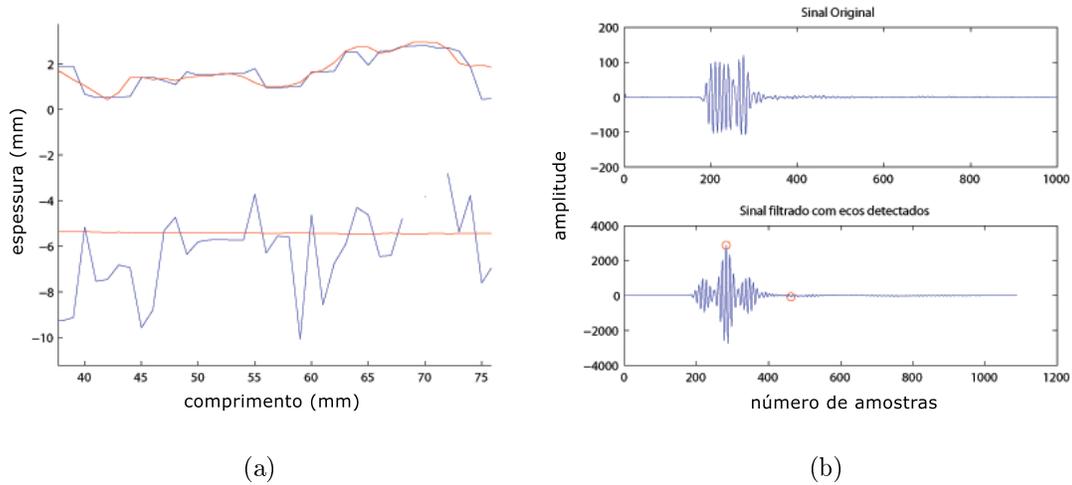


Figura 6.17: (a) Resultado da inspeção e (b) sinal para o ponto 37 *mm* do comprimento

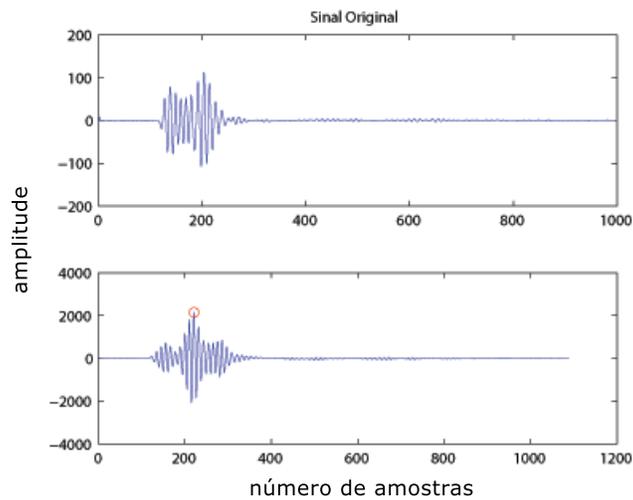


Figura 6.18: Sinal para o ponto 69 *mm* do comprimento

Para tentar melhorar os resultados, é possível criar um algoritmo que trabalhe procurando o segundo eco dentro de uma janela variável. Começando a inspeção em uma região com pouca corrosão, consegue-se saber aonde se encontra a segunda parede com maior precisão. Tendo-se este conhecimento, ao longo da inspeção pode-se ir variando a janela de procura do segundo eco. Uma simulação deste caso foi feita e a figura 6.19 mostra o resultado. Pode-se perceber que perde-se um número maior de pontos já que a janela de procura é menor, porém consegue-se ter uma idéia bem razoável de onde se encontra a segunda parede.

Também é possível definir um limiar o qual pontos com amplitude menor seriam

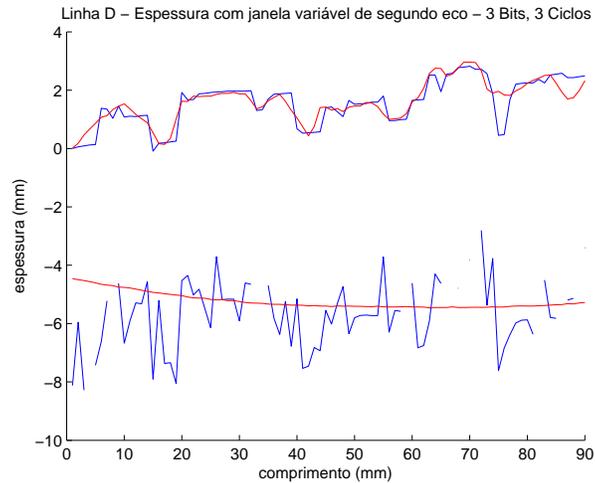


Figura 6.19: Simulação de detecção da segunda parede com uma janela variável de tempo, com duração menor

dispensados. Isso aumentaria o número de pontos perdidos, porém ajudaria a eliminar resultados errados para a segunda parede. Utilizando esta abordagem de descartar pontos, seria possível aumentar o resultado da covariância entre as duas paredes. Porém como o objetivo principal é a análise da corrosão, e não maximizar a covariância, os coeficientes não foram calculados para estes casos.

Para mostrar que estes erros não são provenientes de algum defeito da peça, a figura 6.20 mostra o resultado da inspeção da segunda parede da linha A para três códigos diferentes, e pode-se ver que as variações de perfil ocorrem em localizações diferentes.

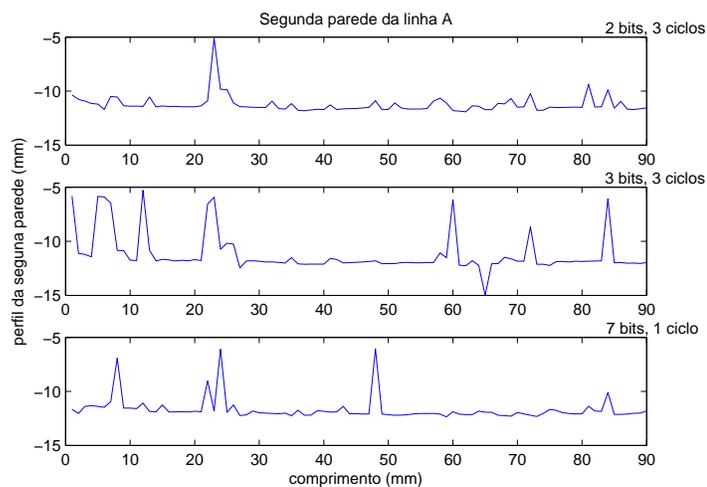


Figura 6.20: Inspeção da segunda parede da linha A para 3 combinações de códigos e ciclos

Esses resultados mostram que a detecção da segunda parede ainda precisa de novos estudos para poder fornecer resultados confiáveis. Porém os resultados gerais são animadores, já que não existe mais o problema de perda de pontos da primeira parede (que ocorriam com métodos antigos de inspeção), e a reconstrução da mesma apresentou resultados melhores que os encontrados antigamente. Além disso, a partir da primeira parede já é possível ter uma ótima idéia da condição da tubulação, já que para regiões muito corroídas sugere-se a troca da mesma, e para regiões com baixo nível de corrosão consegue-se reconstruir também a segunda parede.

7 Conclusão

Este trabalho teve como principal objetivo o estudo e o desenvolvimento de um algoritmo que melhorasse e tornasse mais confiável o resultado de inspeções ultrassônicas de dutos de petróleo, visando a redução de custos e problemas envolvidos com a corrosão desses dutos. Como objetivo secundário, o algoritmo desenvolvido deveria ser implementado em um circuito eletrônico, de modo que pudesse ser embarcado em um *Pig* Ultrassônico. O estudo concentrou-se inicialmente em algoritmos utilizados nos dias de hoje (e anteriormente estudados e desenvolvidos pelo autor (CANALES et al., 2007)), e como tais algoritmos poderiam evoluir para uma inspeção mais completa. Foram levantados os principais problemas enfrentados em uma inspeção ultrassônica, que são a sobreposição de sinais, a dispersão e atenuação dos ecos, as reflexões laterais, e os ruídos.

A elaboração deste trabalho gerou contribuições em três áreas. A primeira é a de circuitos eletrônicos digitais, e sua programação em linguagem VHDL. Esse tipo de abordagem gera economia de recursos muito grande (físicos, financeiros e de tempo), fazendo com que o desenvolvimento de novos circuitos, e todos os testes envolvidos, sejam feitos de maneira rápida e eficiente. Em pouco tempo é possível redesenhar um circuito, simulá-lo e testá-lo com as modificações feitas. Essa expertise possibilita o estudo e desenvolvimento de novos trabalhos em inúmeros campos. O circuito foi desenvolvido, simulado e testado, totalmente em software, e em seguida sua implementação em hardware seguiu exatamente o que era esperado.

A segunda é no método de *signal design*. Essa abordagem se mostrou muito eficiente, pois eliminou a necessidade de elaborar diversos experimentos e repetí-los incessantemente. Simulando os sinais no computador foi possível “testar” cerca de 180 sinais (com diferentes combinações de códigos e ciclos) em diversas situações com muita rapidez. Quando foram feitos os testes reais percebeu-se que as simulações tinham atingido um alto nível de proximidade com a situação real.

A terceira colaboração é no campo de inspeções ultrassônicas. Foi mostrada que a abordagem com filtro-casado e códigos de Barker é robusta e pode gerar resultados muito melhores que os alcançados anteriormente. Os resultados para a inspeção da primeira parede dos dutos apresentou excelente nível de fidelidade, sendo possível ter uma idéia bem acertiva sobre o nível de corrosão do duto. Os resultados para a segunda parede foram satisfatórios para baixos níveis de corrosão, mas encontram diversos problemas para níveis maiores. Porém isso não tirou a validade do trabalho e do algoritmo, já que como comentado, reconstruindo-se a primeira parede com maior confiabilidade, é possível verificar se o duto já se encontra comprometido.

A união destas duas últimas colaborações abre portas para uma gama de estudos com outros algoritmos matemáticos que podem ser implementados digitalmente com grande facilidade, não só para inspeções ultrassônicas, mas para qualquer aplicação que envolva processamento de sinais.

Os resultados das simulações mostraram que os estudos feitos no trabalho comprovaram que a base teórica estava correta. As simulações provaram que o filtro casado pode melhorar a taxa S/R de um sinal, aumentando a probabilidade de identificar a presença de um sinal mesmo em condições desfavoráveis (com ruído e sobreposição). Outro ponto importante das simulações foi mostrar que o uso de um filtro mais simples, e de mais fácil implementação, seria tão poderoso quanto um filtro mais complexo em determinadas situações. Nesta fase percebeu-se que os problema de ruídos, sobreposição e atenuação poderiam ser minimizados com a utilização do filtro casado.

Os testes reais com o primeiro corpo de prova mostraram que as simulações estavam corretas, principalmente em relação à sobreposição de sinais. Nesta etapa foi possível verificar que o uso de sinais de menor comprimento trariam melhores resultados. Além disso, esta fase mostrou que a implementação do algoritmo no circuito digital, em tempo real, era realmente possível.

O desenvolvimento do circuito digital passou por diversas fases de grande complexidade devido à problemas relacionados ao software envolvido (ISE - Xilinx), mas que comprovaram que a abordagem de programação por VHDL é uma poderosa ferramenta. No final os problemas foram solucionados e o hardware pôde ser implementado.

A última etapa de testes reais, feita com o segundo corpo de prova demonstraram que o algoritmo implementado de fato gera resultados com alta fidelidade à realidade. Um adendo neste momento é o fato que o filtro consegue gerar bons resultados mesmo quando

a intensidade do sinal recebido chega a saturar a saída do circuito de recepção. Isto possibilita aumentar a potência dos sinais enviados para os casos de grande atenuação.

Acredita-se que os problemas encontrados (principalmente na detecção da segunda parede) ocorreram também devido às condições e equipamentos utilizados para realização dos testes. Utilizando-se um equipamento mais sofisticado poderia se comparar as inspeções com maior certeza, e assim chegar a resultados mais conclusivos. O amplificador de voltagem utilizado colaborou com a adição de ruídos ao sinal, assim como o separador de sinais.

Nos trabalhos citados no capítulo 2, não são descritos testes com corpos de prova reais, e portanto fica difícil comparar os resultados encontrados com a eficiência do algoritmo desenvolvido neste trabalho. Porém alguns avanços podem ser percebidos. No trabalho de Balzer e Stripf (2002) comentam-se algoritmos para reduzir a quantidade de dados que serão armazenados para análise, o que não é necessário no algoritmo desenvolvido aqui, já que as únicas informações que precisam ser armazenadas são a localização das duas paredes do duto, e não o sinal. Dai et al. (2007) cita algoritmos que realizam parte do processamento *off-line*, o que não é necessário neste trabalho.

Os experimentos também mostraram que a detecção de pites (corrosão localizada) é altamente influenciada pelo diâmetro dos mesmos, e existe um grande compromisso entre a frequência e foco dos transdutores e a detecção da profundidade correta dos pites. Ainda assim os resultados mostraram grande avanço em relação ao trabalho de Reber et al. (2002), que detectava pites de no mínimo 10 *mm* de diâmetro, ou 20 *mm* quando se desejava medir a profundidade, e também ao trabalho de Okamoto et al. (1999), que detectava pites de 10 *mm* com 1 *mm* de profundidade, contra os pites de 3 *mm* de diâmetro detectados neste trabalho.

Por último, vale citar que os resultados alcançados pelo autor em estudos anteriores (CANALES et al., 2007) utilizando a detecção por limiar, ou até mesmo a detecção por picos, demonstraram erros maiores na detecção da primeira parede, onde nem visualmente enxergava-se boa correlação entre os resultados da inspeção ultrassônica e da medição tridimensional.

Conclui-se que os resultados do trabalho representam um avanço no campo de inspeções ultrassônicas de oleodutos, levando a uma redução de custos com correção de desastres que podem ocorrer devido ao desgaste e rompimento dos dutos. Os benefícios alcançados são importantes para a indústria e acima de tudo para a sociedade. O algo-

ritmo e o sistema podem ser aplicados diretamente em *Pigs* utilizados nos dias de hoje, apenas definindo-se os parâmetros de inspeção. Porém novos estudos podem ainda melhorar a eficiência destas inspeções, principalmente visando atacar os problemas de dispersão e ecos laterais, que não conseguiram ser solucionados neste trabalho.

7.1 Trabalhos Futuros

Como o desempenho do algoritmo não foi totalmente satisfatório, mantém-se a possibilidade de aprofundamento nos estudos envolvidos neste trabalho.

No campo de desenvolvimento do circuito digital, sugere-se criar algoritmos que atualizem os parâmetros de inspeção (tempo morto, tempo de atuação dos detectores de pico, delay, etc.) dinamicamente. Desta maneira, a procura pelos *TOAs* pode se tornar mais precisa, diminuindo as janelas de tempo, e procurando os sinais dentro de um intervalo com maior probabilidade de presença dos mesmos. Sabendo-se que a espessura tende a não ter variações maiores que uma determinada porcentagem em relação à última espessura medida, é possível programar estas janelas para procurar o sinal em um intervalo mais específico.

Além da atualização dinâmica dos parâmetros, o circuito pode realizar uma correção automática dos resultados encontrados, para casos que a variação de espessura se demonstrar fora dos padrões esperados (variações grandes e bruscas da derivada da linha resultante). Nesta mesma abordagem, o algoritmo pode fazer correções baseado em tendências de evolução da espessura, isto é, ao encontrar variações bruscas das linhas da parede, o algoritmo corrigiria a linha, mantendo-a seguindo a tendência dos últimos pontos. Estas idéias são passíveis de serem implementadas em tempo real dentro do circuito digital, utilizando-se a programação por linguagem VHDL.

Mudando o foco, os testes com inspeções reais podem ser feitos novamente, utilizando-se instrumentação mais apropriada. A placa pulsadora de sinais, ainda em desenvolvimento na Escola Politécnica pode trazer grande redução de ruído nos testes. Além disso, um sistema de inspeção mais preciso, e/ou um corpo de prova complexo usinado com dimensões conhecidas, irão trazer uma comparação mais fiel entre dos resultados alcançados.

Por último, sugere-se que o estudo com diferentes transdutores de ultrassom deve ser feito para se chegar a uma conclusão certa sobre qual a melhor combinação (frequência,

e transdutor focal ou não focal) deve ser utilizado.

Esses estudos irão definir uma combinação de sistema, algoritmo e transdutor que conseguirão realizar inspeções de dutos com maior acertividade.

Referências

- AKOPIAN, D.; AGAIAN, S. A fast matched filter in time domain. *Position Location and Navigation Symposium, 2004. PLANS 2004*, p. 455–460, 26-29 April 2004.
- BALZER, M.; STRIPF, H. Online data reduction with a dsp-fpga multiprocessor system. *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, v. 2, p. 819–822 vol.2, 2002.
- BARKER, R. H. Group synchronizing of binare digital systems. *Communication Theory*, p. 273–287, 1953.
- CANALES, R. V.; FURUKAWA, C. M.; MARUYAMA, N.; TAKARABE, E. W. Digital ultrasonic system for internal corrosion assessment on oil pipelines. In: *Proceedings of COBEM 2007*. [S.l.: s.n.], 2007.
- CHAN, Y.; YAU, C.; CHING, P. Linear and approximate maximum likelihood localization from toa measurements. *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, v. 2, p. 295–298 vol.2, 1-4 July 2003.
- COOLEY, J.; LEWIS, P.; WELCH, P. Application of the fast fourier transform to computation of fourier integrals, fourier series, and convolution integrals. *Audio and Electroacoustics, IEEE Transactions on*, v. 15, n. 2, p. 79–84, Jun 1967. ISSN 0018-9278.
- DAI, B.; SHENG, S.; TIAN, X.; YANG, Z.; XIE, Z. Research on multi-probe ultrasonic automated in-line inspection system of pipeline corrosion. *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, p. 3105–3110, 5-8 Aug 2007.
- FURUKAWA, C. M.; ADAMOWSKI, J. C.; CAMERINI, C. Desenvolvimento de um pig instrumentado para inspeção de oleodutos por ultra-som. *III Induscon IEEE*, set. 1998.
- HIGUTI, R. T. *Ensaio não-destrutivo por ultra-som para Inspeção de Corrosão em tubulações*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 1994.
- HU, C.-H.; LIU, R.; ZHOU, Q.; YEN, J.; SHUNG, K. K. Coded excitation using biphas-coded pulse with mismatched filters for high-frequency ultrasound imaging. *Ultrasonics*, v. 44, n. 3, p. 330–336, jul. 2006. Disponível em: <<http://www.sciencedirect.com/science/article/B6TW2-4JW10WF-1/2/05e38ec7e7ff36c50b00d5fb7bd6ca91>>.
- INSLEE, J. *Pipeline Issues*. 2008. Website. Disponível em: <<http://www.house.gov/inslee/index.shtml>>.

- JO, G.-D.; CHANG, S.-H.; KWEON, H.-Y.; KIM, S.-Y. A compact matched filter for rapid code acquisition. *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, v. 2, p. 713–717 vol.2, 1999.
- KATRAGADDA, G.; LORD, W.; SUN, Y.; UDPA, S.; UDPA, L. Alternative magnetic flux leakage modalities for pipeline inspection. *Magnetics, IEEE Transactions on*, v. 32, n. 3, p. 1581–1584, May 1996. ISSN 0018-9464.
- KINO, G. S. *Acoustic Wave: devices, imaging, and analog signal processing*. [S.l.]: Prentice-Hall, 1987.
- KINSLER, L. E.; FREY, A. R. *Fundamentals of Acoustics*. [S.l.]: John Wiley & Sons, 1950.
- LEVANON, N. Cross-correlation of long binary signals with longer mismatched filters. *Radar, Sonar and Navigation, IEE Proceedings*, v. 152, n. 6, p. 377–382, 9 Dec 2005. ISSN 1350-2395.
- LIMA, F.; FURUKAWA, C. Development and testing of an acoustic positioning system - description and signal processing. *Ultrasonics Symposium, 2002. Proceedings. 2002 IEEE*, v. 1, p. 849–852 vol.1, 8-11 Oct 2002. ISSN 1051-0117.
- LIMA, F. V. F. de. *Desenvolvimento de um sistema acústico de posicionamento submarino*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2002.
- LIN, Y.; LEE, D.; SAUL, L. Nonnegative deconvolution for time of arrival estimation. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, v. 2, p. ii–377–80 vol.2, 17-21 May 2004. ISSN 1520-6149.
- LUENBERGER, D. On barker codes of even length. *Proceedings of the IEEE*, v. 51, n. 1, p. 230–231, Jan 1963. ISSN 0018-9219.
- MARTIN, C.; MEISTER, J.-J.; ARDITI, M.; FARINE, P.-A. A novel homomorphic processing of ultrasonic echoes for layer thickness measurement. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, v. 40, n. 7, p. 1819–1825, Jul 1992. ISSN 1053-587X.
- OKAMOTO, J.; ADAMOWSKI, J. C.; TSUZUKI, M. S. G.; BUIOCHI, F.; CAMERINI, C. S. Autonomous system for oil pipelines inspection. *Mechatronics*, v. 9, n. 7, p. 731–743, out. 1999.
- PARK, S.; JEONG, H. D.; LIM, Z. S. Development of mobile robot systems for automatic diagnosis of boiler tubes in fossil power plants and large size pipelines. *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, v. 2, p. 1880–1885 vol.2, 2002.
- PIPELINE101. *History of Pipelines*. 2008. Website. Disponível em: <<http://www.pipeline101.com/History/index.html>>.

- PIPEWAY. *Soluções Inteligentes em Inspeção de Dutos*. 2008. Website. Disponível em: <<http://www.pipeway.com>>.
- REBER, K.; BELLER, M.; WILLEMS, H.; BARBIAN, O. A new generation of ultrasonic in-line inspection tools for detecting, sizing and locating metal loss and cracks in transmission pipelines. *Ultrasonics Symposium, 2002. Proceedings. 2002 IEEE*, v. 1, p. 665–671 vol.1, 8-11 Oct 2002. ISSN 1051-0117.
- SAE AMS 2632. *Ultrasonic Inspection of Thin Materials 0.5 Inch (12.7 mm) and Thinner*. [S.l.], 1974.
- SIMON, C. *Técnicas de processamento digital de sinais aplicadas a ensaios não destrutivos por ultra-som*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 1993.
- SMITH, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. [S.l.]: California Technical Publishing, 1997.
- STAROSTINV. Pipeline disaster in the ussr, it had to happen, yet it could have been averted. *pipes pipelines Int*, v. 35, p. 7–8, 1990.
- SUH, D.-M.; KIM, W.-W.; CHUNG, J.-G. Ultrasonic inspection of studs (bolts) using dynamic predictive deconvolution and wave shaping. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, v. 46, n. 2, p. 457–463, Mar 1999. ISSN 0885-3010.
- SWIERCZ, E. A new method of detection of coded signals in additive chaos on the example of barker code. *Signal Processing*, v. 86, n. 1, p. 153–170, jan. 2006. Disponível em: <<http://www.sciencedirect.com/science/article/B6V18-4GDSCCK-1/2/3df06b5c895a612392f9b387464c00e0>>.
- SZABO, T. L. *DIAGNOSTIC ULTRASOUND IMAGING: INSIDE OUT*. [S.l.]: Elsevier Academic Press, 2004.
- TAO, J.; PEIWEN, Q.; ZHENGSU, T. Development of magnetic flux leakage pipe inspection robot using hall sensors. *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004. Proceedings of the 2004 International Symposium on*, p. 325–329, 31 Oct.-3 Nov. 2004.
- TREES, H. L. van. *Detection, Estimation, and Modulation Theory - Part I*. [S.l.]: New York, 1968.
- TREES, H. L. van. *Detection, Estimation, and Modulation Theory - Part III*. [S.l.]: New York, 1971.
- TURIN, G. An introduction to matched filters. *Information Theory, IEEE Transactions on*, v. 6, n. 3, p. 311–329, Jun 1960. ISSN 0018-9448.
- TURYN, R. On barker codes of even length. *Proceedings of the IEEE*, v. 51, n. 9, p. 1256–1256, Sept 1963. ISSN 0018-9219.

XILINX. *Getting Started with FPGA's*. [S.l.], 2008. Disponível em:
<<http://www.xilinx.com/company/gettingstarted/index.htm>>.

YEDLIN, M.; MAJER, E. Phase modulation of acoustic transducers. *Electrical and Computer Engineering, 1993. Canadian Conference on*, p. 270–273 vol.1, 14-17 Sep 1993.

ZHAO, H.; MO, L. Y. L.; GAO, S. Barker-coded ultrasound color flow imaging: theoretical and practical design considerations. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, v. 54, n. 2, p. 319–331, February 2007. ISSN 0885-3010.

Anexo A - Código MATLAB do Detector de picos locais

```
function [picos1 , picos2] = envoltoria(sinal)

z=1; %indice para vetor de picos locais positivos
y=1; %indice para veotr de picos locais negativos

% Calcula primeira envoltoria (picos maiores que os 2 adjacentes para cada
lado)

for i=3:length(sinal) - 2

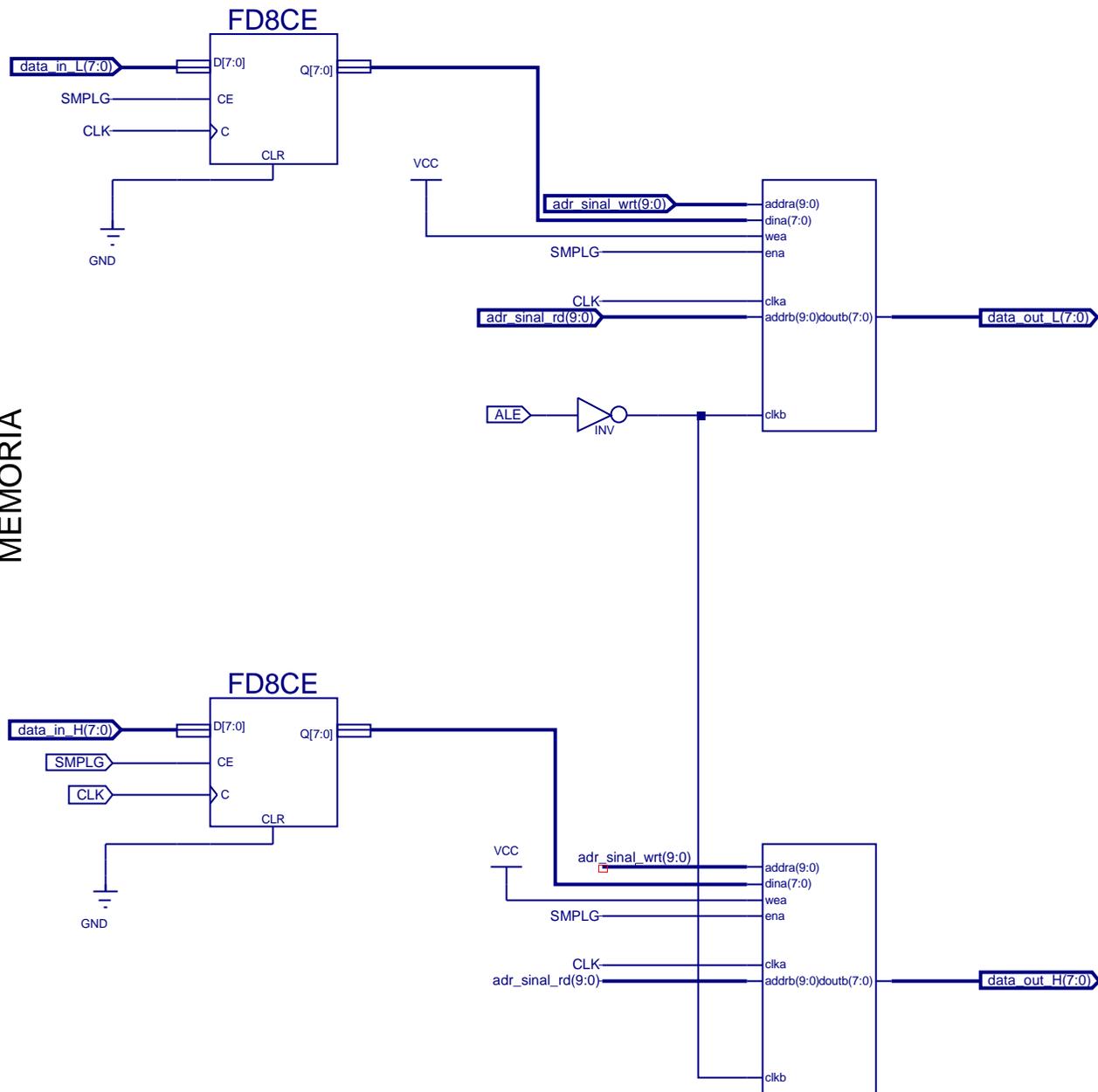
    if (sinal(i)>sinal(i-1)&&senal(i)>sinal(i-2)&&senal(i)>sinal(i+1)&&senal(i)>sina
l(i+2))
        picos1(z,1)= i;
        picos1(z,2)= sinal(i);
        z=z+1;
    end

    if (sinal(i)<sinal(i-1)&&senal(i)<sinal(i-2)&&senal(i)<sinal(i+1)&&senal(i)<sina
l(i+2))
        picos2(y,1)= i;
        picos2(y,2)= sinal(i);
        y=y+1;
    end
end
```

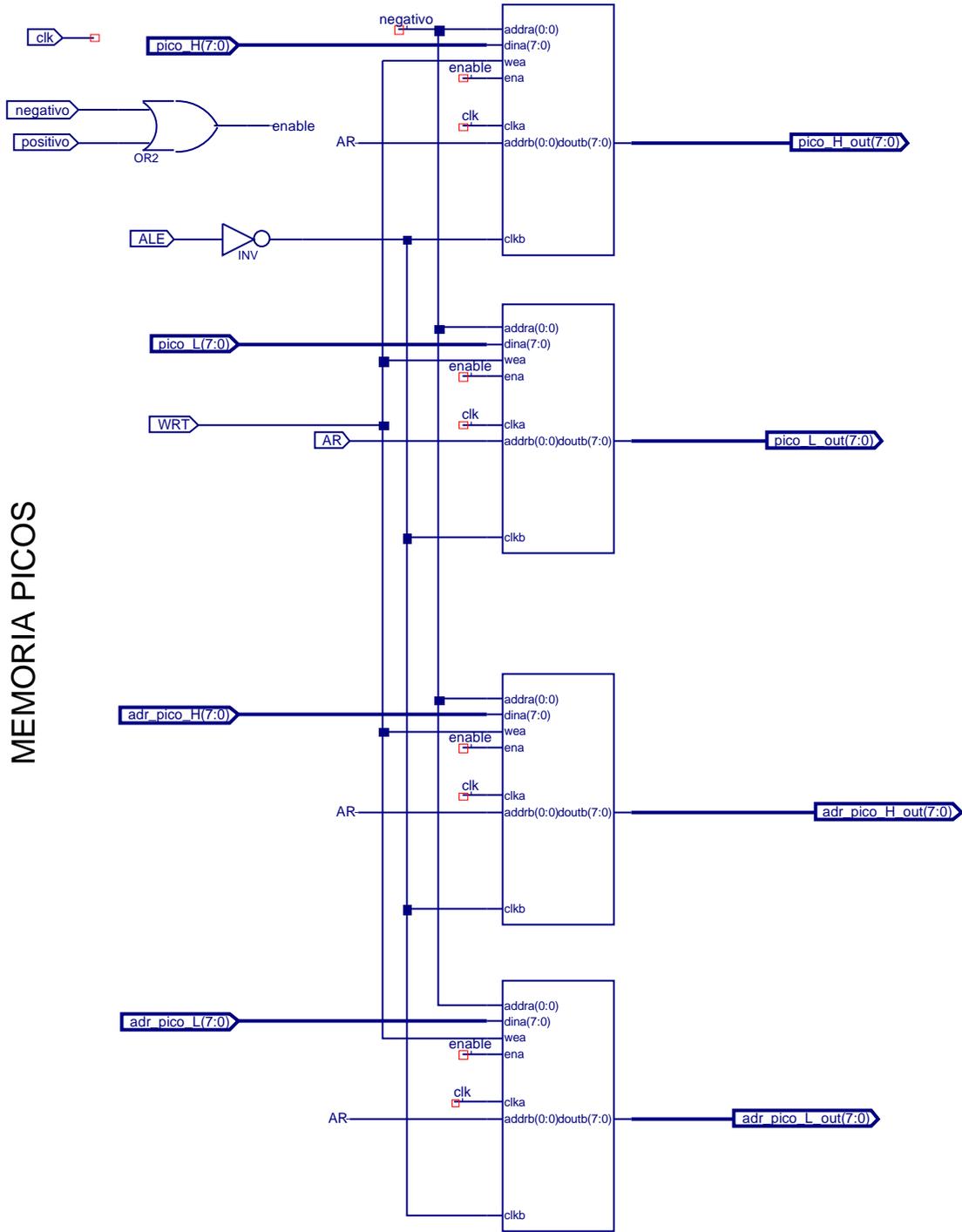
Anexo B - Esquemáticos do Sistema Digital

Todos os esquemáticos aqui apresentados encontram-se em arquivos no DVD que acompanha a tese, dentro do projeto *AD104_2*. Este projeto pode ser aberto dentro do programa ISE, da Xilinx.

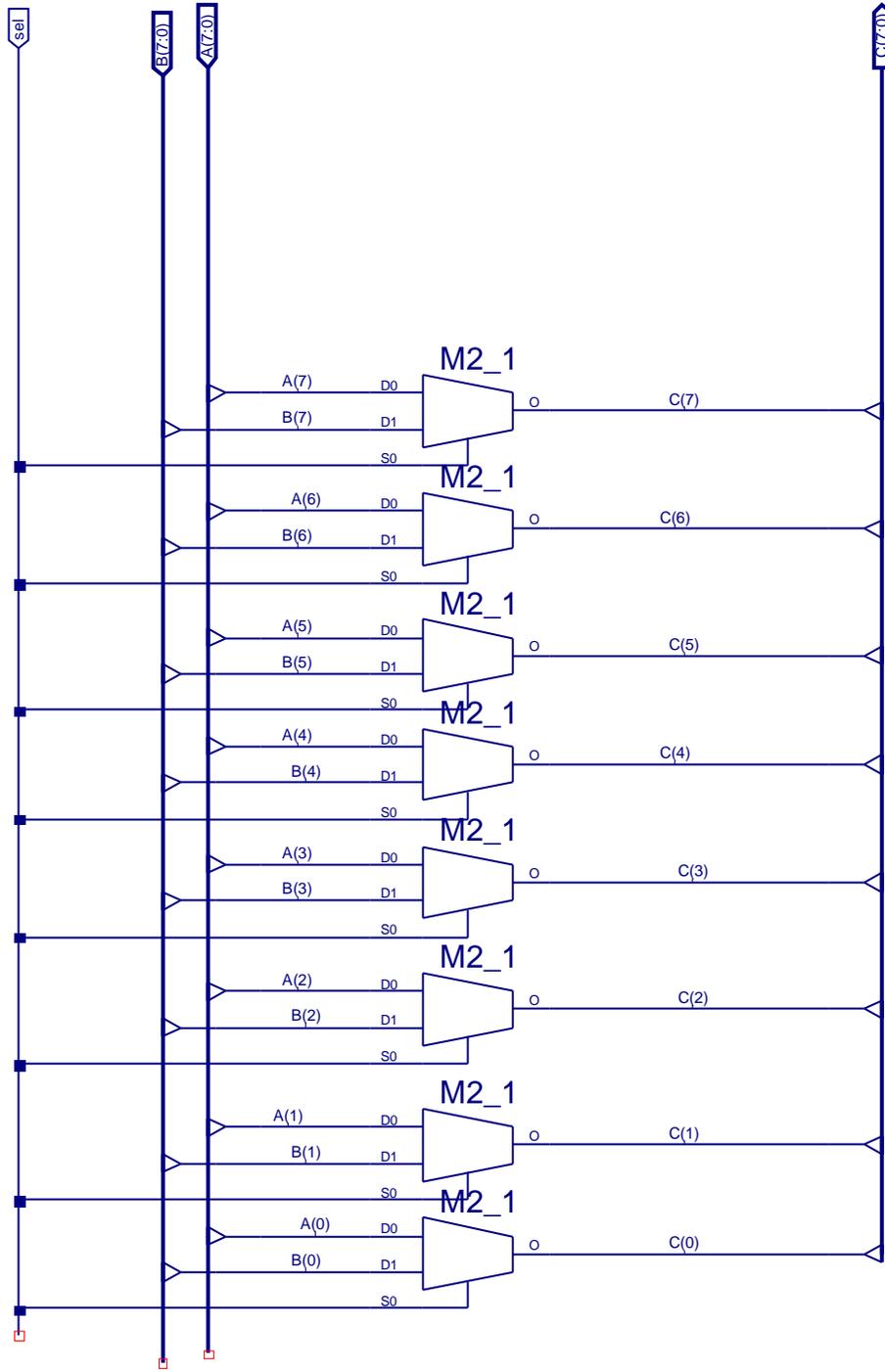
MEMORIA



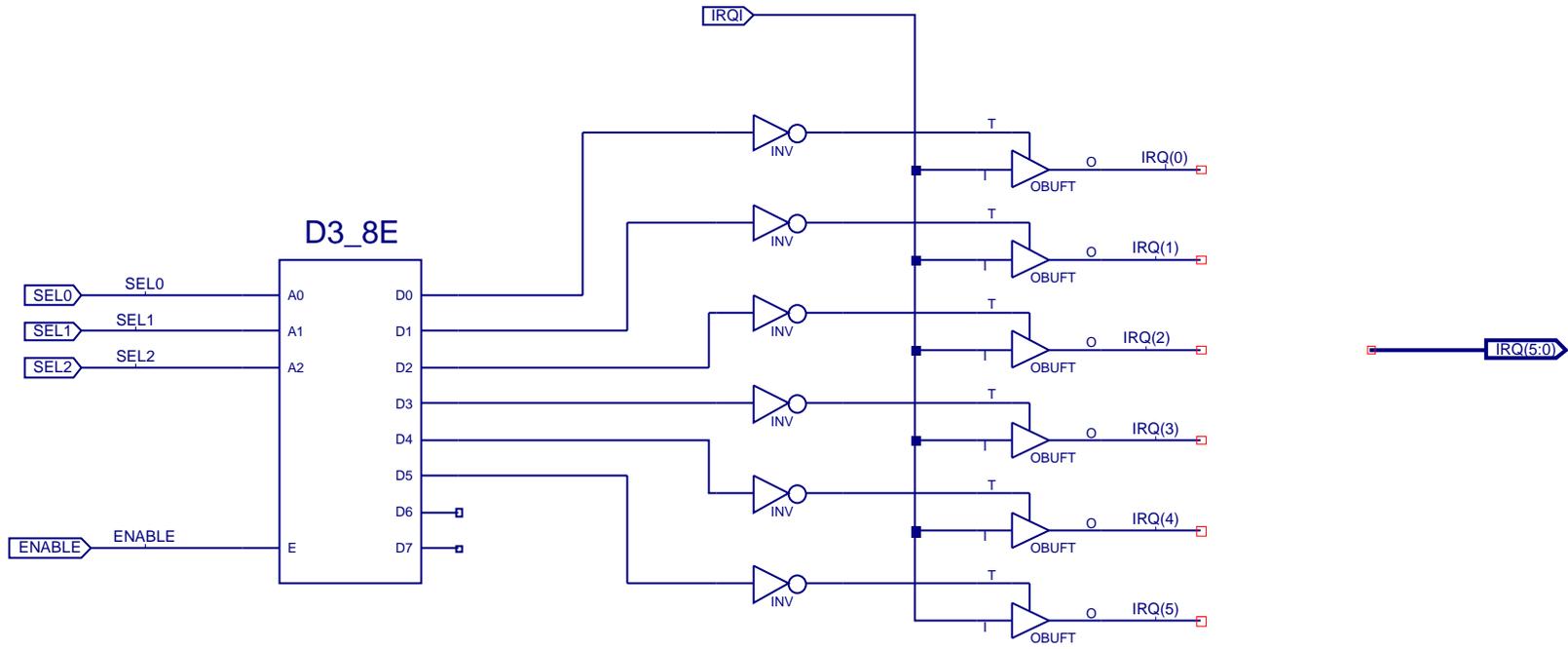
MEMORIA PICOS



M8



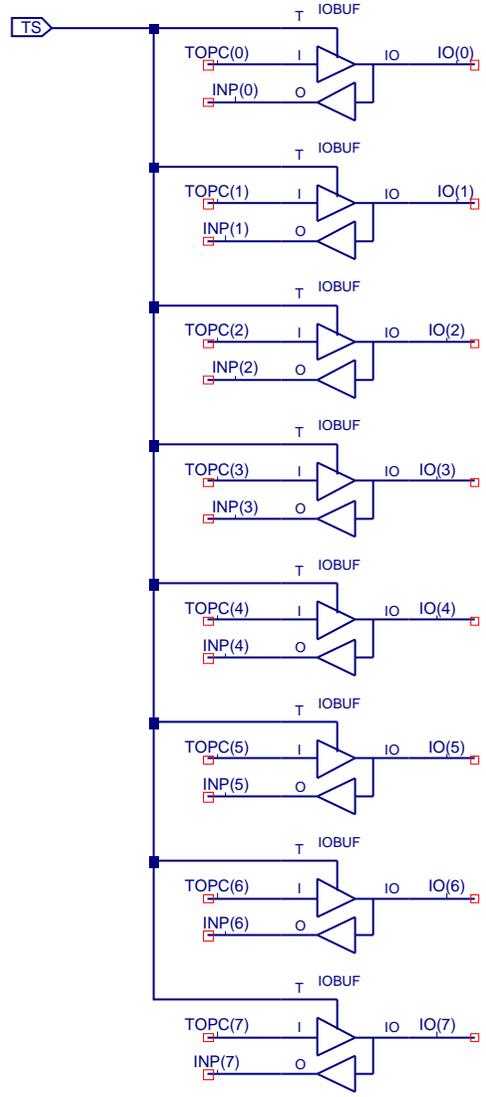
IRQBUF



IOBUF8

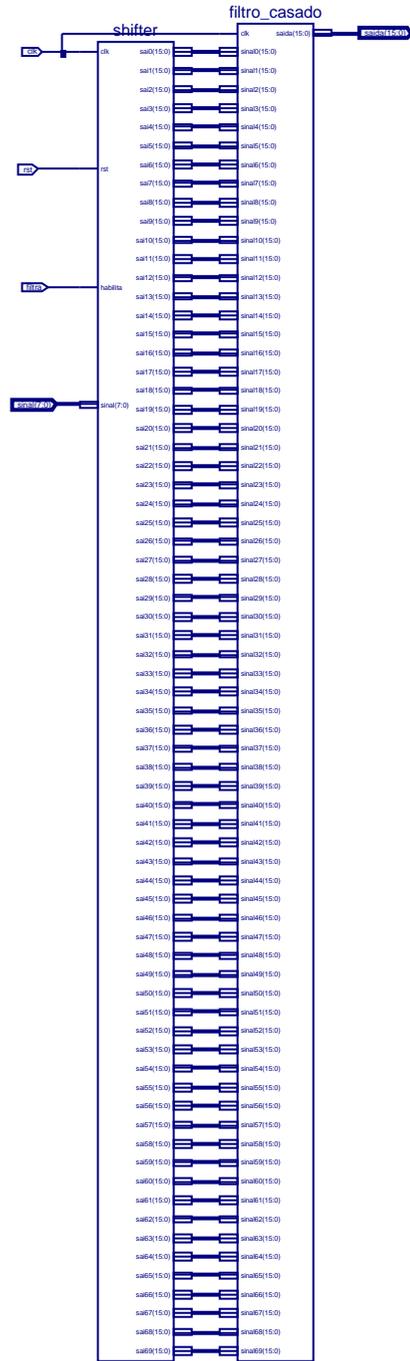
TOPC(7:0)

INP(7:0)

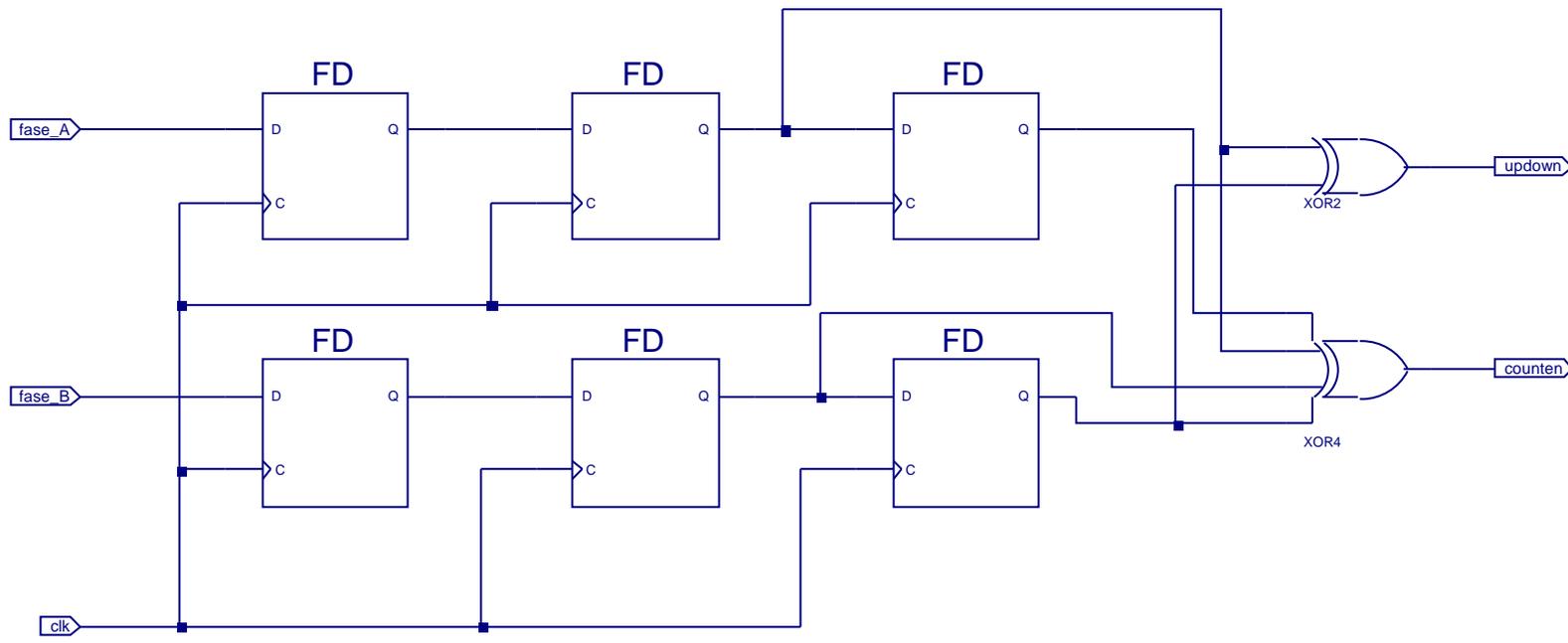


IO(7:0)

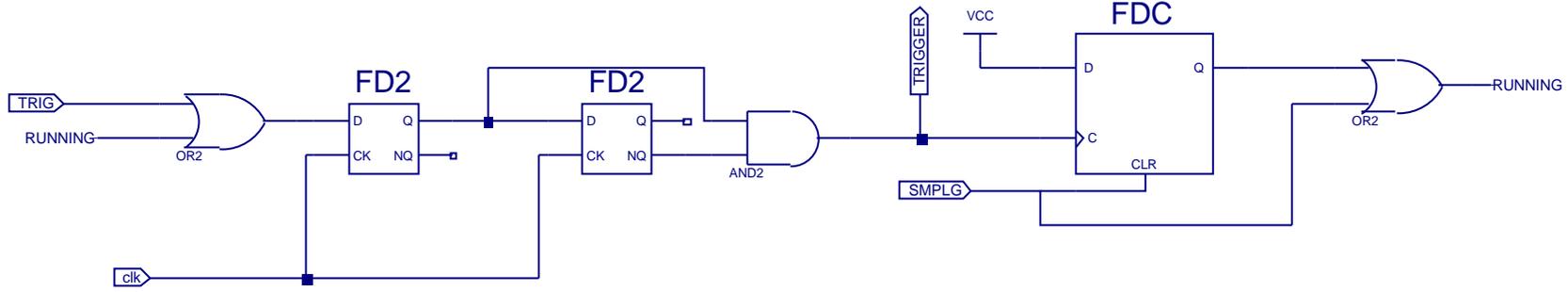
FILTRO



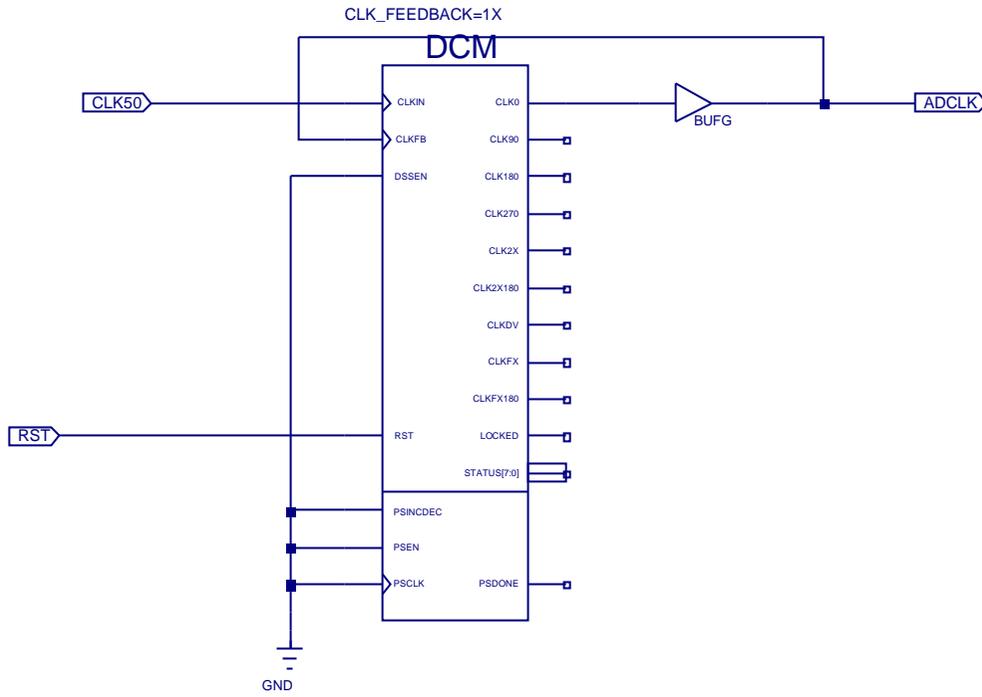
DETECTOR DE QUADRATURA



DETECTA TRIGGER



CLOCKIN



Anexo C – Códigos em VHDL dos blocos do esquemático

Todos os códigos aqui apresentados encontram-se em arquivos no DVD que acompanha a tese, dentro do projeto *AD104_2*. Este projeto pode ser aberto dentro do programa ISE, da Xilinx.

C.1 Contador Encoder

Este bloco recebe os sinais do detector de quadratura (“updown”, que indica a direção de movimento, e “contar”, pulsos que indicam a contagem), e realiza a contagem de pulsos do Encoder/Odômetro. Além da contagem, o bloco é programado para emitir um sinal de trigger sempre que contar uma quantidade determinada de pulsos. Essa quantidade é configurável através do sinal “resolução”.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity contador_encoder is
port    (  resolucao: in  std_logic_vector (7  downto 0);
          clk: in  std_logic;
          rst: in  std_logic;
          updown: in  std_logic;
          contar: in  std_logic;
          contagem_out: out  std_logic_vector (15  downto 0);
          trigger: out  std_logic
        );

end contador_encoder;

architecture Behavioral of contador_encoder is

component saida_contador is
Port ( trigger_in: in  std_logic;
      contar: in  std_logic;
      resolucao: in  std_logic_vector (7  downto 0);
      trigger_out: out  std_logic
    );
end component;

signal contador: std_logic_vector (15  downto 0);
signal trigger_in : std_logic;
signal contagem: std_logic_vector (15  downto 0);

begin

contagem_out <= contagem;

c0: saida_contador port map (trigger_in , contar , resolucao , trigger);

process(contar , rst , clk)
begin
    if rst='1' then
        contador <=x"0000";
        contagem <= x"0000";
    end if;
end process;

```

```

        trigger_in <= '0';
    elsif (clk'event and clk = '1' and contar = '1') then
        if contador = resolucao - '1' then
            trigger_in <= '1';
            contador <= x"0000";
            if updown = '0' then
                contagem <= contagem + '1';
            else
                contagem <= contagem - '1';
            end if;
        else
            if updown = '0' then
                contador <= contador + '1';
                trigger_in <= '0';
                contagem <= contagem + '1';
            else
                contador <= contador - '1';
                trigger_in <= '0';
                contagem <= contagem - '1';
            end if;
        end if;
    end if;
end process;
end Behavioral;

```

C.2 Contador Sinal

Este bloco atualiza os endereços de memória para cada amostra digitalizada.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity contador_sinal is
    Port ( clk : in STD_LOGIC;
          aquisitando : in STD_LOGIC;
          endereco : out STD_LOGIC_VECTOR (9 downto 0));
end contador_sinal;

architecture Behavioral of contador_sinal is

    signal endereco_in: std_logic_vector (9 downto 0);

begin

    endereco <= endereco_in;

    process(clk , aquisitando)
    begin

```

```

if aquisitando = '0' then
    endereco_in <= "0000000000";
else
    if clk'event and clk = '1' then
        endereco_in <= endereco_in + '1';
    end if;
end if;
end process;
end Behavioral;

```

C.3 Controle Acq

Este bloco controla o sinal “SMPGL”, responsável por dizer ao resto do sistema o tempo que o mesmo deve adquirir o sinal. Ele é controlado pelo parâmetro “SIZE”.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity controle_acq is
    Port ( clk : in std_logic;
          rst : in std_logic;
          trigger : in std_logic;
          SIZE: in std_logic_vector (9 downto 0);
          SMPGL: out std_logic
        );
end controle_acq;

architecture Behavioral of controle_acq is

    signal contador: std_logic_vector (9 downto 0);
    signal SMPGL_in: std_logic;

    begin

    SMPGL <= SMPGL_in;

    process(clk, trigger, rst)

    begin
    — Deteccao do trigger
    if rst = '1' then
        SMPGL_in <= '0';
        contador <= "0000000000";
    else
        if trigger='1' then
            contador <= "0000000000";

```

```

        SMPLG_in <= '1';

    -- Estado funcionando
    else
        if (clk='1' and clk'event and SMPLG_in = '1') then
            if contador = SIZE - 1 then
                SMPLG_in <= '0';
            elsif contador < SIZE - 1 then
                contador <= contador + '1';
            end if;
        end if;

    end if;

end if;
end process;
end Behavioral;

```

C.4 Controle Detec

Este bloco controla os tempos de procura pelos picos do eco de ultrassom. O sinal “positivo” habilita a procura pelo primeiro TOA, e o sinal “negativo” habilita a procura pelo segundo. Ele é controlado pelos parâmetros “PTIME”, “DTIME” e “SIZE”.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity controle_detec is
    Port ( clk : in std_logic;
          rst : in std_logic;
          trigger : in std_logic;
          contador: in std_logic_vector (9 downto 0);
          SIZE: in std_logic_vector (9 downto 0);
          PTIME: in std_logic_vector (9 downto 0);
          DTIME: in std_logic_vector (9 downto 0);
          positivo: out std_logic;
          negativo: out std_logic
        );
end controle_detec;

architecture Behavioral of controle_detec is

    signal SMPLG: std_logic;

begin

    process (clk, trigger, rst)

```

```

begin
  -- Deteccao do trigger
  if rst = '1' then
    SMPLG <= '0';
    positivo <= '0';
    negativo <= '0';

  else
    if trigger='1' the
      SMPLG <= '1';
      positivo <= '1';
      negativo <= '0';

    -- Estado funcionando
    else
      if (clk='1' and clk'event and SMPLG = '1') then
        if contador = SIZE - 1 then
          SMPLG <= '0';
          negativo <= '0';
        elsif contador >= PTIME+DTIME -1 then
          negativo <= '1';
        end if;

        if contador = PTIME -1 then
          positivo <= '0';
        end if;

      end if;
    end if;
  end if;
end process;
end Behavioral;

```

C.5 Delay Acq

Este bloco gera o início da aquisição baseado no tempo de delay determinado pelo usuário através do parâmetro “DELAY”.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity delay_acq is
  Port ( clk : in std_logic;
         rst : in std_logic;
         trigger : in std_logic;
         DELAY : in STD_LOGIC_VECTOR (15 downto 0));

```

```

                                TRIG_OUT : out STD_LOGIC
                                );
end delay_acq;

architecture Behavioral of delay_acq is

    signal counter_delay: std_logic_vector (15 downto 0);
    signal foi: std_logic;

begin

    process(clk , trigger , rst )

    begin
        -- Deteccao do trigger
        if rst = '1' then

            trig_out <= '0';
            foi <= '0';
            counter_delay <= x"0000";

        else

            if trigger = '1' then

                foi <= '1';
                counter_delay <= x"0000";
                if delay = x"0000" then
                    trig_out <= '1';
                end if;

                -- Estado funcionando
                elsif foi = '1' then
                    if delay = x"0000" then
                        trig_out <= '0';
                        foi <= '0';

                        elsif(clk='1' and clk'event) then
                            if counter_delay = delay then
                                trig_out <= '1';
                                counter_delay <= counter_delay + '1';
                            elsif counter_delay < delay then
                                counter_delay <= counter_delay + '1';
                            else
                                trig_out <= '0';
                                foi <= '0';
                            end if;
                        end if;
                    end if;
                end if;
            end process;
        end Behavioral;

```

C.6 Detector Pico

Detector de pico máximo e mínimo. Quando o sinal “positivo” for 1, ele procura pelo maior pico do sinal, e quando “negativo” for 1, procura pelo menor. O endereço de saída dos picos é contado a partir do momento que “habilita” for 1. O bloco mantém na saída (enquanto “habilita” for 1) o pico encontrado desejado e o sinal “*picoOk*” é 1 quando no clock atual o pico for o maior (ou menor).

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;

entity detector_pico is
Port (  clk: in std_logic;
        rst: in std_logic;
        positivo: in std_logic;
        negativo: in std_logic;
        sinal: in std_logic_vector (15 downto 0);
        valor: out std_logic_vector (15 downto 0);
        pico_ok: out std_logic
    );
end detector_pico;

architecture Behavioral of detector_pico is

signal atual_max, atual_min: std_logic_vector (15 downto 0);
begin

process(clk, rst)
begin
    if rst = '1' then
        atual_max <= x"0000";
        atual_min <= x"FFFF";
        pico_ok <= '0';

    elsif clk'event and clk = '1' then
        if positivo = '1' then
            if sinal > atual_max then
                atual_max <= sinal;
                valor <= sinal;
                pico_ok <= '1';
            else
                pico_ok <= '0';
            end if;
        elsif negativo = '1' then
            if sinal < atual_min then
                atual_min <= sinal;
                valor <= sinal;
                pico_ok <= '1';
            end if;
        end if;
    end process;

```

```

                else
                    pico_ok <= '0';
                end if;
            end if;
        end if; — geral
end process;
end Behavioral;

```

C.7 Endereçador

Este bloco indica a posição que está sendo inspecionada. Quando recebe um trigger ainda em meio a uma aquisição, ele pula uma posição. Através da análise das posições sabe-se que posição foi perdida na aquisição.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity enderecador is
    Port (
        rst : in std_logic;
        clk : in std_logic;
        acq_st : in std_logic;
        endereco : out STD_LOGIC_VECTOR (15 downto 0)
    );
end enderecador;

architecture Behavioral of enderecador is

    signal endereco_in: std_logic_vector (15 downto 0);
    signal acq_in: std_logic;

begin

    endereco <= endereco_in;

    process(clk , rst)

    begin

        if rst='1' then
            endereco_in <= x"0000";
            acq_in <= '0';
        elsif clk'event and clk = '1' then
            if acq_st = '1' then
                if acq_in='0' then
                    acq_in <= '1';
                    endereco_in <= endereco_in + 1;
                end if;
            end if;
        end if;
    end process;
end Behavioral;

```

```

        else
            acq_in <= '0';
        end if;
    end if;
end process;
end Behavioral;

```

C.8 Filtro Casado

Bloco que implementa o filtro casado como soma e subtração de amostras, divididas em etapas. O código abaixo implementa o filtro de 7 bits e 1 ciclo. Para outras configurações é necessário atualizar a programação do mesmo, modificando os “signals” e a ordem/quantidade de somas e subtrações, de acordo com o código escolhido. Cada estágio do filtro deve separar as somas/subtrações em grupos de 4 amostras, pois com mais do que isso o programa ISE-XILINX travará.

— *Description: Este componente implementa o filtro casado por etapas*

— *Additional Comments:*

— *Para utilizar outras configuracoes do filtro casado, visualizar o arquivo filtros.txt (em baixo de AD104_2)*

— *e copiar o filtro desejado substituindo o atual (quebrar os filtros em etapas de soma/subtracao de*

— *4 sinais (limitacao do ISE que trava com mais disso). Alem disso atualizar os signals*

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
entity filtro_casado is
    Port(
        clk : in std_logic;
        sinal0 : in std_logic_vector (15 downto 0);
        sinal1 : in std_logic_vector (15 downto 0);
        sinal2 : in std_logic_vector (15 downto 0);
        sinal3 : in std_logic_vector (15 downto 0);
        sinal4 : in std_logic_vector (15 downto 0);
        sinal5 : in std_logic_vector (15 downto 0);
        sinal6 : in std_logic_vector (15 downto 0);
        sinal7 : in std_logic_vector (15 downto 0);
        sinal8 : in std_logic_vector (15 downto 0);
        sinal9 : in std_logic_vector (15 downto 0);
        sinal10 : in std_logic_vector (15 downto 0);
        sinal11 : in std_logic_vector (15 downto 0);
        sinal12 : in std_logic_vector (15 downto 0);
        sinal13 : in std_logic_vector (15 downto 0);
        sinal14 : in std_logic_vector (15 downto 0);

```



```

        sinal67: in std_logic_vector (15 downto 0);
        sinal68: in std_logic_vector (15 downto 0);
        sinal69: in std_logic_vector (15 downto 0);
        saida: out std_logic_vector (15 downto 0)
    );
end filtro_casado;

architecture Behavioral of filtro_casado is

    signal saida1, saida2, saida3, saida4, saida5, saida6, saida7, saida8, saida9,
    saida10, saida11, saida12, saida13, saida14, saida15, saida16, saida17, saida18,
    saida19, saida20, saida21, saida22, saida23, saida24, saida25: std_logic_vector
    (15 downto 0);
begin

    saida <= saida25 + saida24; -- ULTIMO ESTAGIO

    process(clk)
    begin
        if clk'event and clk='1' then -- FILTRO 7x1
            -- ESTAGIO 1
            saida1 <= + sinal0 + sinal1 + sinal2 + sinal3;
            saida2 <= + sinal4 - sinal5 - sinal6 - sinal7;
            saida3 <= - sinal8 - sinal9 + sinal10 + sinal11;
            saida4 <= + sinal12 + sinal13 + sinal14 - sinal15;
            saida5 <= - sinal16 - sinal17 - sinal18 - sinal19;
            saida6 <= + sinal20 + sinal21 + sinal22 + sinal23;
            saida7 <= + sinal24 - sinal25 - sinal26 - sinal27;
            saida8 <= - sinal28 - sinal29 - sinal30 - sinal31;
            saida9 <= - sinal32 - sinal33 - sinal34 + sinal35;
            saida10 <= + sinal36 + sinal37 + sinal38 + sinal39;
            saida11 <= - sinal40 - sinal41 - sinal42 - sinal43;
            saida12 <= - sinal44 + sinal45 + sinal46 + sinal47;
            saida13 <= + sinal48 + sinal49 + sinal50 + sinal51;
            saida14 <= + sinal52 + sinal53 + sinal54 - sinal55;
            saida15 <= - sinal56 - sinal57 - sinal58 - sinal59;
            saida16 <= - sinal60 - sinal61 - sinal62 - sinal63;
            saida17 <= - sinal64 + sinal65 + sinal66 + sinal67;
            saida18 <=          + sinal68 + sinal69;
            -- ESTAGIO 2
            saida19 <= saida1 + saida2 + saida3 + saida4;
            saida20 <= saida5 + saida6 + saida7 + saida8;
            saida21 <= saida9 + saida10 + saida11 + saida12;
            saida22 <= saida13 + saida14 + saida15 + saida16;
            saida23 <= saida17 + saida18;
            -- ESTAGIO 3
            saida24 <= saida19 + saida20 + saida21;
            saida25 <= saida22 + saida23;

        end if;
    end process;
end Behavioral;

```

C.9 OBUS MUX

Este bloco faz a multiplexagem dos dados de saída, a partir do endereço definido pelo sinal “addr”. Esse endereço é enviado pelo PC-104, que lê o dado de saída no barramento. A lista dos endereços e os respectivos dados de saída se encontra no esquemático “top.sch”

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity OBUS_MUX is
  Port ( obus : out  STD_LOGIC_VECTOR (7 downto 0);
        sinal_filtrado_L : in  STD_LOGIC_VECTOR (7 downto 0);
        sinal_filtrado_H : in  STD_LOGIC_VECTOR (7 downto 0);
        pico_H : in  STD_LOGIC_VECTOR (7 downto 0);
        pico_L : in  STD_LOGIC_VECTOR (7 downto 0);
        addr_pico_H : in  STD_LOGIC_VECTOR (7 downto 0);
        addr_pico_L : in  STD_LOGIC_VECTOR (7 downto 0);
        encoder_h : in  STD_LOGIC_VECTOR (7 downto 0);
                encoder_l : in  STD_LOGIC_VECTOR (7 downto 0);
                endereco_l : in  std_logic_vector (7 downto 0);
                endereco_h : in  std_logic_vector (7 downto 0);
        status : in  STD_LOGIC_VECTOR (7 downto 0);
        addr : in  STD_LOGIC_VECTOR (12 downto 0);
        aen : in  STD_LOGIC;
        memrdi : in  STD_LOGIC);
end OBUS_MUX;

architecture Behavioral of OBUS_MUX is

  signal nada : std_logic;

begin

  process (aen, memrdi)
  begin
    if (aen = '1') and (memrdi'event and memrdi = '0') then

      -- SINHAL
      if addr(11) = '0' and addr < 2047 then
        if addr(10) = '0' then
          obus <= sinal_filtrado_L;
        elsif addr(10) = '1' then
          obus <= sinal_filtrado_H;
        end if;
      end if;

      case addr is
        -- PICOS E ENDEREÇOS
        when "0100000000000" => -- 0800

```

```

        obus <= pico_L;
when "010000000100" => -- 0804
        obus <= pico_L;
when "010000000001" => -- 0801
        obus <= pico_H;
when "010000000101" => -- 0805
        obus <= pico_H;
when "010000000010" => -- 0802
        obus <= addr_pico_L;
when "010000000110" => -- 0806
        obus <= addr_pico_L;
when "010000000011" => -- 0803
        obus <= addr_pico_H;
when "010000000111" => -- 0807

        obus <= addr_pico_H;

-- ENCODER

when "010000001000" =>
        obus <= encoder_L; -- 0808
when "010000001001" =>
        obus <= encoder_H; -- 0809

-- STATUS

when "010000001010" =>
        obus <= status; -- 080A

-- ENDERECO DE AQUISICAO

when "010000001011" => -- 080B
        obus <= endereco_l;
when "010000001100" => -- 080C
        obus <= endereco_h;

when others =>
        nada <= '0';

        end case;
    end if;
end process;
end Behavioral;

```

C.10 Prog Register

Este bloco é um registrador de parâmetros de inspeção. O sinal “SEL” define qual parâmetro está sendo enviado pelo PC-104. A definição dos valores de SEL, e de qual parâmetro está sendo configurado se encontram na própria definição das constantes no código a seguir. Os endereços de escrita que deverão ser enviados pelo PC-104 estão no esquemático

“top.sch”.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity prog_register is
  Port ( IBUS : in  STD_LOGIC_VECTOR (7 downto 0);
        SEL  : in  STD_LOGIC_VECTOR (3 downto 0);
        SELEN : in  STD_LOGIC;
        RST  : in  STD_LOGIC;
        WRT  : in  STD_LOGIC;
        CFG  : out STD_LOGIC_VECTOR (15 downto 0);
        DELAY : out STD_LOGIC_VECTOR (15 downto 0);
        SIZE  : out STD_LOGIC_VECTOR (9  downto 0);
        DTIME : out STD_LOGIC_VECTOR (9  downto 0);
        PTIME : out STD_LOGIC_VECTOR (9  downto 0);
        RESOLUCAO: out std_logic_vector (7  downto 0);
        trigger_interno: out std_logic;
        CLR_FF: out std_logic
    );
end prog_register;

architecture Behavioral of prog_register is

  constant cfgL: std_logic_vector := x"0";
  constant cfgH: std_logic_vector := x"1";
  constant delayL: std_logic_vector := x"2";
  constant delayH: std_logic_vector := x"3";
  constant sizeL: std_logic_vector := x"4";
  constant sizeH: std_logic_vector := x"5";
  constant dtimeL: std_logic_vector := x"6";
  constant dtimeH: std_logic_vector := x"7";
  constant ptimeL: std_logic_vector := x"8";
  constant ptimeH: std_logic_vector := x"9";
  constant resol: std_logic_vector := x"A";
  constant clearFF: std_logic_vector := x"E"; -- Reset para todos os blocos ao
  final da aquisicao
  constant trigger: std_logic_vector := x"F";

begin

  process(wrt , rst)
  begin
    if rst = '1' then -- Reset Assincrono
      CFG <= x"0000";
      DELAY <= x"0000";
      SIZE <= "0000000000";
      DTIME <= "0000000000";
      PTIME <= "0000000000";
      RESOLUCAO <= x"00";
      CLR_FF <= '1';
    end if;
  end process;
end Behavioral;

```

```
trigger_interno <= '0';

else
  if wrt'event and wrt = '1' and SELEN = '1' then -- Evento de escrita no
  Registrador
    case SEL is
      when cfgL =>
        CFG(7 downto 0) <= IBUS;
        CLR_FF <= '0';

      when cfgH =>
        CFG(15 downto 8) <= IBUS;
        CLR_FF <= '0';

      when delayL =>
        DELAY (7 downto 0) <= IBUS;
        CLR_FF <= '0';

      when delayH =>
        DELAY (15 downto 8) <= IBUS;
        CLR_FF <= '0';

      when sizeL =>
        SIZE (7 downto 0) <= IBUS;
        CLR_FF <= '0';

      when sizeH =>
        SIZE (9 downto 8) <= IBUS(1 downto 0);
        CLR_FF <= '0';

      when dtimeL =>
        DTIME (7 downto 0) <= IBUS;
        CLR_FF <= '0';

      when dtimeH =>
        DTIME (9 downto 8) <= IBUS(1 downto 0);
        CLR_FF <= '0';

      when ptimeL =>
        PTIME (7 downto 0) <= IBUS;
        CLR_FF <= '0';

      when ptimeH =>
        PTIME (9 downto 8) <= IBUS(1 downto 0);
        CLR_FF <= '0';

      when resol =>
        RESOLUCAO <= IBUS;
        CLR_FF <= '0';

      when clearFF =>
        CLR_FF <= IBUS(0);
```

```

        when trigger =>
            trigger_interno <= IBUS(0);
            CLR_FF <= '0';

        when others =>
            null;

    end case;
end if;
end if;
end process;
end Behavioral;

```

C.11 Salva Sinal

Este bloco serve para centrar o sinal adquirido em zero (o sinal entra com valores entre 0 e 255, e sai com valores entre -128 a 127).

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity salva_sinal is
    Port ( clk: in std_logic;
          rst: in std_logic;
          sinal : in STD_LOGIC_VECTOR (7 downto 0);
          saida: out std_logic_vector (7 downto 0)
        );
end salva_sinal;

architecture Behavioral of salva_sinal is

    signal sinal_in: std_logic_vector (7 downto 0);

begin

    saida <= sinal_in - 116;

    process(clk , rst)

    begin

        if rst = '1' then
            sinal_in <= x"74";
        elsif clk'event and clk = '1' then
            sinal_in <=sinal;
        end if;
    end process;
end architecture;

```

```
end process;  
end Behavioral;
```

C.12 Shifter

Este bloco é o shift register necessário para o funcionamento do filtro casado.

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_SIGNED.ALL;  
  
entity shifter is  
    Port ( clk : in std_logic;  
          rst : in std_logic;  
          habilita : in std_logic;  
          sinal : in STD_LOGIC_VECTOR (7 downto 0);  
          sai0 : out std_logic_vector (15 downto 0);  
          sai1 : out std_logic_vector (15 downto 0);  
          sai2 : out std_logic_vector (15 downto 0);  
          sai3 : out std_logic_vector (15 downto 0);  
          sai4 : out std_logic_vector (15 downto 0);  
          sai5 : out std_logic_vector (15 downto 0);  
          sai6 : out std_logic_vector (15 downto 0);  
          sai7 : out std_logic_vector (15 downto 0);  
          sai8 : out std_logic_vector (15 downto 0);  
          sai9 : out std_logic_vector (15 downto 0);  
          sai10 : out std_logic_vector (15 downto 0);  
          sai11 : out std_logic_vector (15 downto 0);  
          sai12 : out std_logic_vector (15 downto 0);  
          sai13 : out std_logic_vector (15 downto 0);  
          sai14 : out std_logic_vector (15 downto 0);  
          sai15 : out std_logic_vector (15 downto 0);  
          sai16 : out std_logic_vector (15 downto 0);  
          sai17 : out std_logic_vector (15 downto 0);  
          sai18 : out std_logic_vector (15 downto 0);  
          sai19 : out std_logic_vector (15 downto 0);  
          sai20 : out std_logic_vector (15 downto 0);  
          sai21 : out std_logic_vector (15 downto 0);  
          sai22 : out std_logic_vector (15 downto 0);  
          sai23 : out std_logic_vector (15 downto 0);  
          sai24 : out std_logic_vector (15 downto 0);  
          sai25 : out std_logic_vector (15 downto 0);  
          sai26 : out std_logic_vector (15 downto 0);  
          sai27 : out std_logic_vector (15 downto 0);  
          sai28 : out std_logic_vector (15 downto 0);  
          sai29 : out std_logic_vector (15 downto 0);  
          sai30 : out std_logic_vector (15 downto 0);  
          sai31 : out std_logic_vector (15 downto 0);  
          sai32 : out std_logic_vector (15 downto 0);
```

```
sai33 : out std_logic_vector(15 downto 0);
sai34 : out std_logic_vector(15 downto 0);
sai35 : out std_logic_vector(15 downto 0);
sai36 : out std_logic_vector(15 downto 0);
sai37 : out std_logic_vector(15 downto 0);
sai38 : out std_logic_vector(15 downto 0);
sai39 : out std_logic_vector(15 downto 0);
sai40 : out std_logic_vector(15 downto 0);
sai41 : out std_logic_vector(15 downto 0);
sai42 : out std_logic_vector(15 downto 0);
sai43 : out std_logic_vector(15 downto 0);
sai44 : out std_logic_vector(15 downto 0);
sai45 : out std_logic_vector(15 downto 0);
sai46 : out std_logic_vector(15 downto 0);
sai47 : out std_logic_vector(15 downto 0);
sai48 : out std_logic_vector(15 downto 0);
sai49 : out std_logic_vector(15 downto 0);
sai50 : out std_logic_vector(15 downto 0);
sai51 : out std_logic_vector(15 downto 0);
sai52 : out std_logic_vector(15 downto 0);
sai53 : out std_logic_vector(15 downto 0);
sai54 : out std_logic_vector(15 downto 0);
sai55 : out std_logic_vector(15 downto 0);
sai56 : out std_logic_vector(15 downto 0);
sai57 : out std_logic_vector(15 downto 0);
sai58 : out std_logic_vector(15 downto 0);
sai59 : out std_logic_vector(15 downto 0);
sai60 : out std_logic_vector(15 downto 0);
sai61 : out std_logic_vector(15 downto 0);
sai62 : out std_logic_vector(15 downto 0);
sai63 : out std_logic_vector(15 downto 0);
sai64 : out std_logic_vector(15 downto 0);
sai65 : out std_logic_vector(15 downto 0);
sai66 : out std_logic_vector(15 downto 0);
sai67 : out std_logic_vector(15 downto 0);
sai68 : out std_logic_vector(15 downto 0);
sai69 : out std_logic_vector(15 downto 0)
);

end shifter;

architecture Behavioral of shifter is

type vector is array(69 downto 0) of std_logic_vector(15 downto 0);
signal interno: vector;

begin

sai0 <= interno(0);
sai1 <= interno(1);
sai2 <= interno(2);
sai3 <= interno(3);
sai4 <= interno(4);
```

```
sai5 <= interno(5);
sai6 <= interno(6);
sai7 <= interno(7);
sai8 <= interno(8);
sai9 <= interno(9);
sai10 <= interno(10);
sai11 <= interno(11);
sai12 <= interno(12);
sai13 <= interno(13);
sai14 <= interno(14);
sai15 <= interno(15);
sai16 <= interno(16);
sai17 <= interno(17);
sai18 <= interno(18);
sai19 <= interno(19);
sai20 <= interno(20);
sai21 <= interno(21);
sai22 <= interno(22);
sai23 <= interno(23);
sai24 <= interno(24);
sai25 <= interno(25);
sai26 <= interno(26);
sai27 <= interno(27);
sai28 <= interno(28);
sai29 <= interno(29);
sai30 <= interno(30);
sai31 <= interno(31);
sai32 <= interno(32);
sai33 <= interno(33);
sai34 <= interno(34);
sai35 <= interno(35);
sai36 <= interno(36);
sai37 <= interno(37);
sai38 <= interno(38);
sai39 <= interno(39);
sai40 <= interno(40);
sai41 <= interno(41);
sai42 <= interno(42);
sai43 <= interno(43);
sai44 <= interno(44);
sai45 <= interno(45);
sai46 <= interno(46);
sai47 <= interno(47);
sai48 <= interno(48);
sai49 <= interno(49);
sai50 <= interno(50);
sai51 <= interno(51);
sai52 <= interno(52);
sai53 <= interno(53);
sai54 <= interno(54);
sai55 <= interno(55);
sai56 <= interno(56);
```

```
sai57 <= interno(57);
sai58 <= interno(58);
sai59 <= interno(59);
sai60 <= interno(60);
sai61 <= interno(61);
sai62 <= interno(62);
sai63 <= interno(63);
sai64 <= interno(64);
sai65 <= interno(65);
sai66 <= interno(66);
sai67 <= interno(67);
sai68 <= interno(68);
sai69 <= interno(69);

process(rst , clk)
begin
    if rst = '1' then
        for i in 0 to 69 loop
            interno(i) <= x"0000";
        end loop;
    elsif clk'event and clk = '1' and habilita = '1' then
        for i in 0 to 68 loop
            interno(i) <= interno(i+1);
        end loop;
        if sinal < x"00" then
            interno(69) <= x"FF"&sinal;
        else
            interno(69) <= x"00"&sinal;
        end if;
    end if;
end process;
end Behavioral;
```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)