

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Processador Integrado para a Navegação de AUVs

Helder Campos

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: José Carlos dos Santos Alves

Co-orientador: Aníbal Castilho Coimbra de Matos

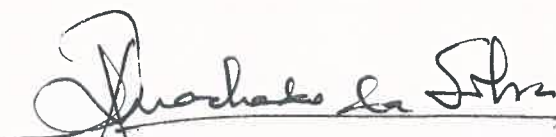
Junho de 2011

A Dissertação intitulada

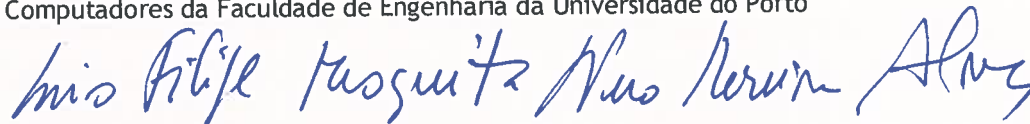
“Processador Integrado para Navegação de Veículos Submarinos Autónomos
(AUVs)”

foi aprovada em provas realizadas em 12-07-2011

o júri



Presidente Professor Doutor José Alberto Peixoto Machado da Silva
Professor Associado do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Luís Filipe Mesquita Nero Moreira Alves
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática
da Universidade de Aveiro



Professor Doutor José Carlos dos Santos Alves
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Aníbal Castilho Coimbra de Matos
Professor Auxiliar Departamento de Engenharia Eletrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Hélder Filipe Santana Campos

Faculdade de Engenharia da Universidade do Porto

Resumo

Este documento descreve um processador integrado, realizado com tecnologias baseadas em [FPGA](#), para a navegação de submarinos autónomos. O trabalho teve em vista a sua integração no [MARES](#), um submarino autónomo produzido pelo grupo *OceanSys* na Faculdade de Engenharia da Universidade do Porto. O objectivo do processador é detectar e enviar sinais acústicos para que o submarino possa navegar em meio aquático, calculando a sua posição através dos tempos de propagação destes sinais. Se forem obtidos tempos de propagação em relação a pelo menos dois referenciais, a posição do submarino pode ser obtida.

Este sistema já existia mas era implementado de forma analógica, o que torna esta dissertação mais uma prova da revolução digital que se está a verificar. Com o digital surgem vantagens como menor sensibilidade a factores externos como a temperatura, a possibilidade de usar melhores métodos de navegação e até mesmo a possibilidade de realizar comunicações subaquáticas com elevada fiabilidade que não seriam tão facilmente conseguidas com meios analógicos.

Abstract

This document describes an [FPGA](#) based integrated processor for the navigation of autonomous submarines. The work was seeking its integration on [MARES](#), an autonomous submarine produced by the *OceanSys* group at Faculdade de Engenharia da Universidade do Porto. The processor objective is to detect and send acoustic signals to make the submarine able to navigate, calculating its position using the signal propagation delay. If the propagation delay relative to two or more references are obtained, the submarine position may be known.

This system already exists but it was implemented with analog electronics, what makes this thesis one more proof of the digital revolution we are facing. With digital technologies rises several advantages like less sensitivity to external factors like temperature, the possibility to use better ways to navigate and even the possibility to use high fidelity underwater communications which would not be easily made using analog systems.

Agradecimentos

Antes de mais, gostava de agradecer ao professor José Carlos Alves, pela ajuda, motivação e apoio que me deu ainda antes do começo deste trabalho.

Gostava de agradecer também ao Eng. Eduardo Sousa pela disponibilidade e ajuda que deu.

Agradeço ao professor Aníbal Matos pela disponibilidade que teve para o teste do trabalho, mesmo em dias importantes para o envio do TriMARES para o Brasil.

Por último, agradeço à Patrícia pela compreensão que teve enquanto estive ocupado no desenvolvimento do projecto.

Helder Campos

Conteúdo

1	Introdução	1
1.1	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	Autonomous Underwater Vehicles (AUVs)	3
2.2	Ondas Acústicas	3
2.3	Navegação Subaquática	5
2.3.1	Long Baseline (LBL)	5
2.3.2	Variantes de LBL	6
2.3.3	Short Baseline (SBL)	7
2.3.4	Ultra Short Baseline (USBL)	7
2.3.5	Dead Reckoning (DR)	7
2.4	Modular Autonomous Robot for Environment Sampling (MARES)	7
2.4.1	Sistema de Navegação	8
2.4.2	Hardware da Acústica	8
2.5	Teoria da Amostragem	9
2.6	Filtros Digitais	10
2.6.1	Finite Impulse Response (FIR)	10
2.6.2	Infinite Impulse Response (IIR)	12
3	Processador integrado para navegação de AUVs	15
3.1	Esquema de Detecção	15
3.1.1	Filtros Passa-Banda	16
3.1.2	Ruído de Banda Estreita	19
3.1.3	Níveis de Decisão	19
3.1.4	Alcance	21
3.1.5	Precisão	22
3.1.6	Implementação Digital	23
3.2	Plataforma de Desenvolvimento	24
3.3	Interface Acústica	26
3.4	Field Programmable Gate Array (FPGA)	27
3.4.1	Sistema Embebido	27
3.4.2	Hardware Dedicado	28
3.5	Software	35
4	Implementação e Validação Experimental	39
4.1	Implementação	39
4.2	Validação Experimental	40

4.2.1	Curto Alcance	40
4.2.2	Médio Alcance	42
4.2.3	Integração com o MARES	42
5	Conclusões e Trabalho Futuro	45
5.1	Satisfação dos Objectivos	45
5.2	Trabalho Futuro	45
	Referências	47

Lista de Figuras

2.1	LBL	6
2.2	Espectro de um sinal amostrado	10
2.3	Resposta impulsional ideal	11
2.4	Resposta impulsional FIR	11
2.5	Mapeamento bilinear	13
3.1	Resposta em frequência dos filtros passa-banda.	16
3.2	Resposta adaptada de um filtro passa-banda.	18
3.3	Resposta desadaptada de um filtro passa-banda.	18
3.4	Distribuição de probabilidade de ruído de banda estreita.	20
3.5	Dimensionamento do nível de detecção.	20
3.6	Suzaku-SZ130	25
3.7	Interface com o MARES	27
3.8	Arquitectura geral da FPGA	28
3.9	Arquitectura do hardware dedicado.	28
3.10	Implementação digital dos filtros.	30
3.11	Explicação pictórica do detector de envolvente.	31
3.12	Valor absoluto de uma sinusóide.	32
4.1	Amostras de um sinal enviado nos 21kHz.	41
4.2	Amostras do detector de envolvente de 21kHz.	41
4.3	Amostras do detector de envolvente de 22kHz.	42

Lista de Tabelas

4.1	Sumario da utilização da Spartan3E-1200.	39
4.2	Sumario da utilização da Spartan3E-500.	40
4.3	Repetibilidade do Sistema.	43

Abreviaturas e Símbolos

AUV	Autonomous Underwater Vehicle
A/D	Analog/Digital
BRAM	Block Random Access Memory
BSS	Block Started by Symbol
DCM	Digital Clock Manager
DR	Dead Reckoning
DSP	Digital Signal Processor
DVL	Doppler Velocity Log
FIFO	First In First Out
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
GNU	GNU's Not Unix
GPS	Global Positioning System
IIR	Infinite Impulse Response
I2C	Inter-Integrated Circuits
JTAG	Joint Test Action Group
LBL	Long Baseline
LED	Light Emitting Diode
MAC	Multiply And Accumulate
MARES	Modular Autonomous Robot for Environment Sampling
MB	Mega Byte
MSPS	Mega Samples Per Second
OPB	On-chip Peripheral Bus
PCB	Printed Circuit Board

RAM Random Access Memory

SBL Short Baseline

SDRAM Synchronous Dynamic Random Access Memory

SPI Serial Peripheral Interface

UPS Underwater Positioning System

USBL Ultra Short Baseline

Capítulo 1

Introdução

O fundo dos oceanos ainda permanece, em grande parte, um mistério. Mais de 70% da crosta terrestre está coberta de água e, só recentemente, é que se começaram a explorar os oceanos usando técnicas de mapeamento por sonar [1]. Este reconhecimento pode ser feito com navios ou pequenas embarcações mas é difícil ou mesmo impossível recolher dados de certas zonas. Estes processos levam em geral muito tempo. Surge então necessidade em criar submersíveis autónomos que possam chegar a zonas de difícil acesso.

Um dos maiores desafios associados à operação de submarinos autónomos está na sua navegação [2]. A atenuação de sinais electromagnéticos em ambiente aquático é muito elevada, rondando os $45\sqrt{f}$ dB/Km, em que f é a frequência em Hz. Em contra partida, a atenuação de sinais acústicos na água é três ordens de grandeza inferior para as frequências de interesse [3]. Esta é uma das principais razões pelo que se usam sinais sonoros no meio aquático, tanto para transmissão de dados como para a navegação.

Actualmente a lei de Moore ainda se verifica e a demanda aos produtos digitais nunca foi tão elevada. Esta realidade traduz-se num aumento na capacidade de processamento dos circuitos digitais. Por causa disso, existem problemas que apenas podiam ser resolvidos com circuitos analógicos que podem agora ser resolvidos digitalmente. A navegação de submarinos inclui-se nestes problemas.

Esta dissertação pretende mostrar que a detecção de sinais acústicos em ambiente subaquático pode ser realizada digitalmente e com maior qualidade, face a métodos analógicos, abrindo janelas para novos e melhores métodos de navegação. Isto foi conseguido com a implementação de um processador integrado para a navegação de Autonomous Underwater Vehicles (AUVs) que vai ser descrito neste documento.

1.1 Estrutura da Dissertação

No capítulo 2 é feita uma explicação de algumas das tecnologias e conceitos usados ao longo da tese. É feita uma pequena apresentação aos AUV assim como algum estado de arte e são explicados conceitos imprescindíveis para este trabalho sobre ondas acústicas e sobre navegação subaquática.

O capítulo 3 explica o trabalho realizado, assim como alguns problemas obtidos durante a realização da tese e é seguido pelo capítulo 4 que explica que procedimentos foram tomados para testar o projecto mostrando alguns resultados pertinentes.

O último capítulo pretende esclarecer o leitor sobre o grau de satisfação do trabalho realizado e elucidá-lo sobre possíveis caminhos que este trabalho pode vir a seguir.

Capítulo 2

Revisão Bibliográfica

2.1 Autonomous Underwater Vehicles (AUVs)

Um **AUV** é um veículo submersível, normalmente de dimensão reduzida, que opera de forma autónoma. Entre muitos dos propósitos dos **AUVs** está o mapeamento do fundo dos oceanos. Neste modo o **AUV** segue um trajecto programado mapeando e recolhendo dados do oceano como a salinidade, temperatura, turvação ou fluorescência. Recentemente foram acrescentadas câmaras aos **AUVs** que permitem que o fundo do oceano seja filmado ou fotografado. Os dados são retidos em memória até que a missão acabe e sejam recuperados.

Existem **AUVs** que podem servir para inspeccionar estruturas como barragens, cascos de navios ou oleodutos substituindo vários mergulhadores humanos. Para tal tarefa, o **AUV** tem de seguir a estrutura a ser inspeccionada de muito perto. Estas estruturas possuem muitas vezes uma configuração elaborada e o submarino tem de se adaptar a ela, não podendo apenas limitar-se a seguir um trajecto. Neste caso são usadas técnicas, como por exemplo Doppler Velocity Log (**DVL**) para se saber a distância e a velocidade relativa à estrutura.

Outra aplicação usa o **AUV** para seguir um determinado padrão medido pelos seus sensores. O **AUV** deve adaptar melhor a sua trajectória para que o padrão a ser medido melhore. São usados **AUVs** neste modo para seguir embarcações, fugas de químicos, entre outros [4].

2.2 Ondas Acústicas

As ondas acústicas são o resultado da vibração das partículas da água provocada por qualquer fonte de energia com suficiente adaptação acústica para que haja transferência de energia. Esta energia pode ser aproveitada, por exemplo, para localização, mapeamento de oceanos, comunicação ou detecção.

Atenuação

A energia das ondas acústicas diminui com a distância percorrida. Esta diminuição é modelada essencialmente por dois grandes parâmetros. O primeiro é a dispersão radial da onda, a

energia distribui-se por uma esfera com raio cada vez maior fazendo com que a energia recebida diminua com o quadrado da distância. A atenuação neste caso é de $N_{spr} = 20 \cdot \log(r/ro)$ dB/m em que r é a distância percorrida e ro é uma distância de referência [5]. Existem casos em que a distancia percorrida pela onda é muito maior do que a profundidade do meio e a dispersão da onda é melhor modelada por uma distribuição cilíndrica. Neste caso a atenuação é modelada por $N_{spr} = 10 \cdot \log(r/ro)$ dB/m. O segundo parâmetro modela a resistência produzida pela água à propagação da onda. A atenuação neste caso é linear com a distância e é dada por $N_{att} = \alpha r$. Em que α é:

$$\alpha = \left[\frac{0.11 f^2}{1 + f^2} + \frac{0.70 f_t f^2 (S/35)}{f_t^2 + f^2} + \frac{0.03 f^2}{f_t} \right] (1 - 65 \cdot 10^{-6} d) \quad \text{dB/km} \quad (2.1)$$

[5]

onde:

f = frequência (kHz)

f_t = frequência de relaxamento(kHz) = $21.9 \cdot 10^6 - 1.520/(T+273)$

T = temperatura(°C)

S = salinidade(partes por milhar)

d = profundidade(m)

A atenuação total em dB é dada pela soma dos dois parâmetros.

$$N = N_{spr} + N_{att}$$

Velocidade

A velocidade do som é um parâmetro indispensável para a medição de distâncias com base nos tempos de ida e volta de sinais acústicos. Ao contrário da velocidade do som no ar, em meio aquático e em especial no mar, a velocidade do som varia significativamente com parâmetros como a profundidade, pressão, temperatura, e até salinidade. Erros no seu valor podem levar a erros graves na medida de distâncias pelo que é necessário obter valores precisos com base nas condições da água. Com base em [6] a velocidade do som na água pode ser calculada com base na seguinte formula empírica aqui simplificada [5]:

$$c = 1449.3 + 4.572T - 0.0445T^2 + 0.0165d + 1.398(S - 35) \quad \text{m/s} \quad (2.2)$$

Esta equação mantém-se aceitável para a seguinte gama de valores:

Temperatura $T = -3$ a 30°C

Profundidade $d = 0$ a $10,000$ m

Salinidade $S = 33$ a 37 partes por milhar

À superfície a precisão da equação 2.2 é inferior a 0.2%. A velocidade do som normalmente usada, corresponde à velocidade obtida com $T \simeq 13^{\circ}\text{C}$ e é de 1500 m/s [5].

Propagação

As ondas acústicas no oceano tendem a seguir vários caminhos devido a reflexões na superfície e no fundo do oceano. Perturbações na temperatura da água, na salinidade ou noutros factores provocam desvios na direcção da onda. Isto dificulta a recepção de sinais acústicos porque estes fenómenos levam ao aparecimento de múltiplos caminhos na transmissão do sinal, o que faz com que se possa receber replicas dos sinais ou, até mesmo, uma adição entre o sinal propagado directamente e uma reflexão.

2.3 Navegação Subaquática

Entende-se por navegação subaquática o acto de tomar decisões e seguir rotas com base na localização corrente em ambiente submarino. Para se poder tomar tais decisões correctamente, é necessário um conhecimento preciso da localização num determinado referencial relativo ou absoluto. Actualmente, são usadas frequentemente estruturas de apoio à navegação dos **AUV** equipadas com transdutores acústicos que neste documento serão chamadas de faróis. Nesta secção serão expostos vários métodos de navegação, assim como algumas vantagens e desvantagens entre elas.

2.3.1 Long Baseline (LBL)

LBL é um sistema de localização usado em **AUVs** que usa quatro ou mais faróis com transdutores acústicos dispostos ao longo da área de operação. Como a profundidade do **AUV** pode ser obtida através de sensores de profundidade, o método pode ser usado com apenas três faróis. O **AUV**, quando quer saber a sua localização, envia um sinal acústico diferente a cada farol. Cada farol apenas responde ao sinal que lhe é destinado. A resposta é recebida pelo **AUV**, o tempo de ida e volta é registado e a distância a um farol específico é calculada com base na velocidade do som na água. Esta velocidade pode estar pré programada no **AUV** ou pode ser calculada com base na salinidade, pressão e temperatura da água. Com base nas distâncias do **AUV** aos faróis, o submarino calcula a sua posição relativa. Antes da missão começar a localização absoluta de cada farol é registada no **AUV** para que se possa calcular a sua posição absoluta. Como cada farol

tem de ser inquirido individualmente, o tempo entre aquisições pode ser de vários segundos. Este tempo aumenta ainda mais se existirem vários AUVs em operação [4].

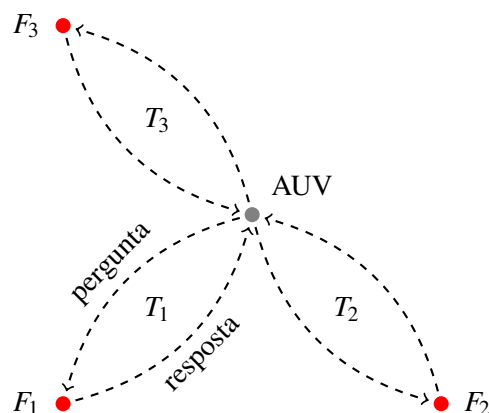


Figura 2.1: LBL

A figura 2.1 mostra um exemplo de uma disposição horizontal dos faróis em LBL. Os faróis podem estar afixados no leito do meio de operação ou à superfície. Com os tempos de ida e volta dos sinais acústicos a cada farol, T_n , e com a velocidade de propagação do som no meio c consegue-se obter as distâncias a cada farol, D_n , da seguinte maneira:

$$D_n = \frac{T_n - T_r}{2} \cdot c \quad (2.3)$$

onde T_r é o tempo levado na detecção do sinal (assumindo que este tempo é igual tanto nos faróis como no submarino). A posição do submarino é então obtida, por triangulação, usando as distâncias do submarino aos faróis e o valor do sensor de profundidade.

2.3.2 Variantes de LBL

Existe uma variante de LBL em que os faróis e o AUV possuem relógios sincronizados. O sincronismo é feito à superfície, antes da missão usando por exemplo rádio. Depois do começo da missão naturalmente que os relógios se irão desviar uns dos outros e por isso esse desvio terá de se manter em valores que não alterem significativamente a precisão do sistema. O sincronismo entre relógios permite que todos os faróis enviem sinais periódicos ao mesmo tempo possibilitando assim ao AUV obter a sua posição sem gastar potência no envio. O AUV sabe o tempo de envio de cada sinal, regista o tempo de chegada de cada um e calcula a sua distância usando, mais uma vez, a velocidade do som no meio. Este método tem a vantagem de diminuir o tempo de aquisição mesmo usando vários AUVs na mesma área de operação [4].

Como o sincronismo entre os relógios dos faróis e o do AUV pode ser difícil porque o AUV só pode sincronizar no início da missão (os faróis podem sincronizar em qualquer altura, caso estejam à superfície) uma variante de LBL surgiu, onde só existe sincronismo de relógios nos faróis. O

envio de sinais por parte dos faróis ocorre, à semelhança do método anterior, de modo síncrono e periódico. O AUV calcula a sua posição com a diferença entre chegada de sinais e não através dos tempos absolutos de envio e chegada. Este método também foi referenciado como Underwater Positioning System (UPS)[7].

2.3.3 Short Baseline (SBL)

Semelhante ao LBL, difere apenas na localização dos faróis. Cada farol, em vez de estar posicionado ao longo da área de operação, posiciona-se ao longo de uma estrutura (navio, pequena embarcação, plataforma etc.). Como os faróis estão mais próximos, a diferença entre tempos de ida e volta de faróis diferentes é menor diminuindo a precisão face a LBL convencional [8].

2.3.4 Ultra Short Baseline (USBL)

Este método usa apenas um farol para a navegação idêntico aos usados em LBL. A diferença reside agora no sistema de recepção do AUV que é composto por uma matriz de transdutores. À semelhança de LBL convencional, o AUV interroga o farol. A resposta enviada pelo farol é registado pela matriz de transdutores no AUV e a diferença de fase entre recepções é usada para o cálculo da direcção do sinal. Com base na posição do farol, registada no início da missão, da direcção e tempo de ida e volta do sinal, a posição exacta do submarino pode ser calculada apenas com uma interrogação. A precisão deste sistema é menor do que em LBL e SBL.

2.3.5 Dead Reckoning (DR)

DR é um método de navegação usado quando não é possível obter posições absolutas, com os métodos citados em cima. São usados dados de giroscópios, acelerómetros, sensores DVL [4] ou outros. Estes sensores apenas dão vectores de aceleração, no caso dos acelerómetros e giroscópios ou vectores de velocidade no caso de sensores DVL. Uma posição relativa pode ser obtida integrando os dados destes sensores. Como os dados de qualquer sensor são afectados de erro, e como estes dados são integrados, a posição calculada por este método vai sofrer de erro cumulativo, provocando um desvio cada vez maior da posição real. Por este motivo, este método apenas é usado nos tempos de espera entre actualizações da posição absoluta, podendo ser combinado com qualquer um dos métodos absolutos expostos anteriormente.

2.4 Modular Autonomous Robot for Environment Sampling (MARES)

Todo o trabalho foi desenvolvido e testado tendo em mente o MARES, um AUV desenvolvido e construído pelo Ocean Systems Group na Faculdade de Engenharia da Universidade do Porto. Embora toda a especificação do trabalho seja definida para este AUV em concreto com algumas modificações podia ser adaptado a outros.

O **MARES** é um **AUV** com 1.5m que, à semelhança de outros **AUVs**, pode ser programado para seguir trajetórias predefinidas enquanto recolhe dados com os seus sensores internos. O submarino pode mergulhar a uma profundidade de até 100m e, ao contrário de outros semelhantes, possui motores verticais que torna possível o movimento vertical puro. A velocidade horizontal pode ser definida, independentemente, de 0 a 2 m/s. O veículo possui uma autonomia de cerca de 10 horas ou 40Km [9].

2.4.1 Sistema de Navegação

A principal função do sistema de navegação do **MARES** é determinar a sua posição horizontal, já que a posição vertical é determinada directamente pelo sensor de pressão.

O veículo transporta um transdutor acústico omnidireccional capaz de transmitir e receber sinais acústicos. O algoritmo de navegação é em tudo semelhante ao **LBL** descrito na secção 2.3.1. No tempo que existe entre aquisições da posição, o **MARES** calcula a sua posição através do método de **DR** descrito na secção 2.3.5. As posições absolutas obtidas através de **LBL** e de **DR** são fundidas usando um filtro de Kalman, tomando vantagem das características de cada tipo de dados [2].

O sistema de navegação usa duas bóias como farol, cada uma equipada com um transdutor multi-frequência ressonante nas frequências entre os 20 e os 30kHz, semelhante ao do veículo. Tanto o veículo como as bóias podem transmitir e receber em 8 frequências. Cada bóia pode ser programada com vários pares de frequências em que a recepção numa determinada frequência leva ao envio da frequência correspondente. Este mecanismo de pares de frequências pode ser usado para a navegação de vários **AUVs** com o mesmo conjunto de bóias, bastando apenas usar pares diferentes entre submarinos [2].

Cada bóia é equipada com um modem rádio e um emissor de Global Positioning System (**GPS**) e envia, via rádio, todas as detecções assim como o seu tempo exacto. Como o submarino interroga a próxima bóia quando recebe a resposta da anterior, os tempos enviados para terra podem ser usados para seguir o **AUV** em tempo real. Imagine-se que a bóia A é interrogada na frequência f_1 e a B na frequência f_2 . Quando o **AUV** interroga a bóia A, o tempo exacto da recepção na bóia A é enviado para terra. Logo após a recepção da resposta pelo submarino, a bóia B é interrogada na frequência f_2 . O tempo exacto da recepção da frequência f_2 pela bóia A é enviado também para terra. A diferença entre os tempos de recepção da frequência f_1 e f_2 pela bóia A ($T_{f_2} - T_{f_1}$) é igual ao tempo de ida e volta do submarino à bóia A. Para saber o tempo de ida e volta à bóia B subtrai-se o tempo de recepção na bóia A da frequência f_1 pelo próximo tempo da frequência f_2 [2]. As coordenadas de **GPS** de cada bóia também são enviadas para terra e são usadas para calcular as coordenadas absolutas do **MARES**.

2.4.2 Hardware da Acústica

O hardware existente no **MARES** e nas bóias consiste numa série de filtros passa-banda analógicos, construídos com elementos passivos (bobinas, condensadores e resistências) com elevado

factor de qualidade Q e sintonizados a diferentes frequências entre os 20kHz e os 27kHz. São 8 ao todo, o que torna possível a recepção em 8 canais. Podem ser acrescentados mais caso seja necessário. Os filtros são alimentados com uma versão amplificada do sinal proveniente do transdutor. A saída de cada filtro passa por detectores de envolvente e por circuitos de detecção, circuitos estes que detectam se o nível de sinal passa acima de um determinado limiar, que pode ser configurado através de vários potenciômetros digitais. Os circuitos de detecção são monitorizados por um micro-processador que mantém um relógio interno de 16bits com resolução de cerca de 250us. Sempre que seja detectado um sinal, o micro-processador regista o valor do relógio e envia-o, por porto série, assim como o canal em que foi recebido, para o processador principal do MARES.

O hardware de envio é ligado ao mesmo transdutor que o de recepção e envia ondas quadradas de 2ms a qualquer uma das 8 frequências, a pedido. Estas ondas quadradas são enviadas para o transdutor que, sendo ressonante dos 20kHz aos 30kHz, vai filtrar as componentes de alta e baixa frequência fazendo com que o sinal enviado para a água se aproxime bastante de um tom puro.

2.5 Teoria da Amostragem

Este trabalho teve uma forte componente de processamento de sinal, pelo que se mostra aqui um breve resumo das técnicas de processamento de sinal usadas.

Para que se possam processar sinais digitalmente, tem de existir uma amostragem prévia dos mesmos, já que digitalmente só se conseguem processar quantidades discretas. Este processo pode ser estudado considerando as amostras como impulsos de *dirac*. Se o sinal for amostrado com período T_s e frequência de amostragem $f_s = \frac{1}{T_s}$ obtém-se o seguinte:

$$d(t) = s(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.4)$$

A série de *Fourier* de $\sum_{n=-\infty}^{\infty} \delta(t - nT)$ é:

$$A_k = \frac{1}{T} \int_0^T \delta(t) \cdot e^{-\frac{2\pi k}{T}t} dt = f_s \quad (2.5)$$

e a transformada de *Fourier* correspondente é:

$$\mathcal{F} \left\{ \sum_{n=-\infty}^{\infty} \delta(t - nT) \right\} = 2\pi f_s \cdot \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi f_s \cdot k) \quad (2.6)$$

Atendendo à seguinte propriedade das transformadas de *Fourier*:

$$x(t) \cdot y(t) \xleftrightarrow{\mathcal{F}} \frac{1}{2\pi} X(\omega) * Y(\omega) \quad (2.7)$$

Ou seja, a multiplicação no domínio dos tempos leva à convolução no domínio das frequências. Pode-se dizer que a transformada de *Fourier* do sinal amostrado é a convolução do sinal original

com a equação 2.6:

$$\begin{aligned} D(\omega) &= \frac{1}{2\pi} S(\omega) * \left[2\pi f_s \cdot \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi f_s \cdot k) \right] \\ &= f_s \sum_{k=-\infty}^{\infty} S(\omega - 2\pi f_s \cdot k) \end{aligned} \quad (2.8)$$

Mostra-se então que a transformada de *fourier* do sinal amostrado é uma repetição periódica em frequência do sinal original, multiplicado por uma constante. O período desta repetição é simplesmente a frequência de amostragem f_s , como se mostra na figura 2.2. A frequência f_h presente na figura é a frequência máxima do sinal. Se esta frequência fosse maior do que $\frac{f_s}{2}$ as repetições do sinal misturavam-se e o sinal original não podia ser restaurado, este fenómeno tem o nome de *aliasing*. Para evitar que isto aconteça, o sinal original pode ser filtrado para que a sua frequência máxima seja sempre inferior a $\frac{f_s}{2}$ (filtro anti-aliasing) ou amostra-se o sinal a mais do dobro da frequência máxima. A esta frequência, que limita superiormente a frequência máxima do sinal, chama-se frequência de *nyquist* e é igual a metade da frequência de amostragem, $f_n = \frac{f_s}{2}$.

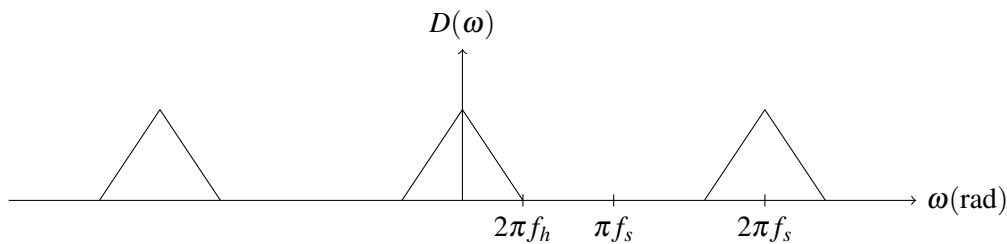


Figura 2.2: Espectro de um sinal amostrado

2.6 Filtros Digitais

O dimensionamento de filtros digitais passa por tentar aproximar a resposta impulsional de um filtro ideal, à resposta obtida pela de um filtro digital e, ao mesmo tempo, aproximar a resposta em frequência ao ideal. Existem vários métodos para se conseguir tal feito, uns deles tentando fazer um esforço por manter a proximidade da resposta impulsional e outros tentando manter a resposta em frequência.

2.6.1 Finite Impulse Response (FIR)

Este tipo de filtros tenta aproximar a sua resposta impulsional a uma de um filtro ideal. A saída do filtro é obtida realizando a convolução da resposta impulsional pela entrada. Obviamente muitas das respostas ideais são infinitas e a convolução de sinais não se pode realizar, pelo menos em tempo real. Por isso, a resposta impulsional de filtros FIR é limitada para que seja finita. Existem vários métodos para limitar a resposta impulsional destes filtros, o mais simples é o

método da janela, que multiplica a resposta impulsional ideal por uma janela limitadora. A janela mais simples que se pode usar é um pedestal como mostram os exemplos nas figuras 2.4 e 2.3. Como é obvio, a janela vai provocar desvios em relação à resposta em frequência ideal.

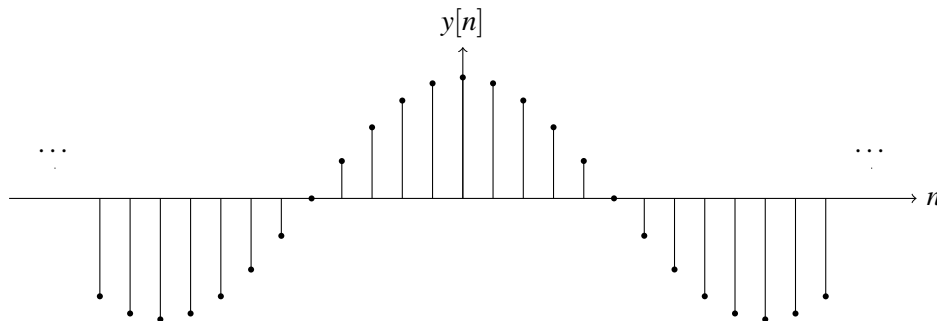


Figura 2.3: Resposta impulsional ideal

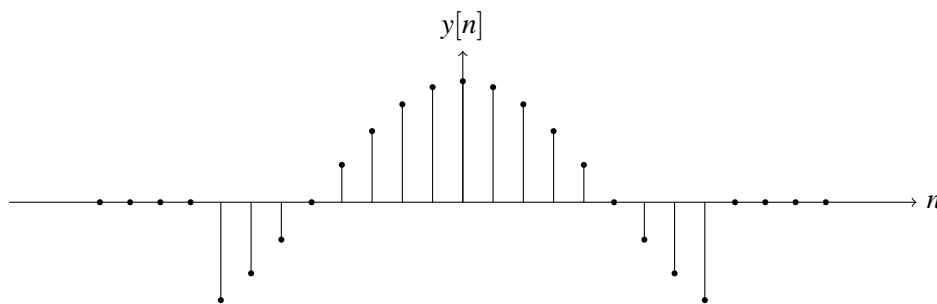


Figura 2.4: Resposta impulsional FIR

Um filtro FIR é implementado da seguinte maneira:

$$y[n] = \sum_{k=0}^N b_k \cdot x[n-k] \quad (2.9)$$

onde $y[n]$ é a saída do filtro no instante n , $x[n]$ é a entrada no mesmo instante e b_k é o coeficiente k do filtro.

A resposta impulsional $h[n]$ é obtida substituindo-se na equação 2.9, $x[n] = \delta[n]$, onde $\delta[n]$ é um impulso digital.

$$h[n] = \sum_{k=0}^N b_k \cdot \delta[n-k] = b_n \quad (2.10)$$

Como mostra a equação 2.10, os coeficientes b_k são obtidos directamente pela resposta impulsional. A ordem destes filtros é definida como o número de coeficientes N mais um.

Com estes filtros podem-se criar atrasos de fase linear, que são ideais em aplicações de processamento de som. Em contrapartida, são necessários muitos mais coeficientes para atingir propriedades em frequência semelhantes a outro tipo de filtros, nomeadamente filtros Infinite Impulse

Response (IIR). Os filtros FIR são naturalmente estáveis por terem respostas impulsivas finitas.

2.6.2 Infinite Impulse Response (IIR)

Como o nome indica, a resposta ao impulso deste tipo de filtros é infinita. Estes filtros, ao contrário dos FIR, usam realimentação ou seja, usam valores das saídas anteriores para calcular novos valores. Isto pode causar instabilidade. Filtros analógicos são recriados facilmente no domínio digital usando filtros IIR. Existem dois métodos principais para o seu dimensionamento que vão ser descritos em seguida.

Método da invariância ao impulso

Este método, como o nome indica, procura fazer com que a resposta impulsional do filtro seja exactamente igual à ideal amostrada. Para isso, calcula-se a transformada Z da resposta impulsional e, através das propriedades desta transformada, é descoberta uma equação às diferenças que é usada para implementar o filtro.

A título de exemplo, considere-se um filtro passa baixo analógico, com um único pólo (a) com a seguinte transformada de Laplace:

$$H(s) = \frac{a}{a+s} \quad (2.11)$$

a resposta impulsional deste filtro é, no domínio contínuo:

$$h(t) = a \cdot e^{-at} \cdot u(t). \quad (2.12)$$

A resposta impulsional digital é calculada amostrando a resposta analógica com frequência de amostragem $f_s = \frac{1}{T_s}$. O resultado é multiplicado pelo período de amostragem. A multiplicação é feita para que a resposta em frequência seja semelhante à do filtro analógico, já que, como foi provado na equação 2.8, o espectro do sinal amostrado é uma repetição periódica do original multiplicado pela frequência de amostragem. A resposta impulsional do exemplo é:

$$h[n] = \begin{cases} aT_s \cdot e^{-aT_s n} & , n \geq 0 \\ 0 & , n < 0 \end{cases} \quad (2.13)$$

Aplicando a transformada Z, obtém-se:

$$\begin{aligned} H(Z) &= \sum_{n=0}^{\infty} aT_s \cdot e^{-aT_s n} \cdot Z^{-n} \\ &= aT_s \cdot \sum_{n=0}^{\infty} (e^{-aT_s} \cdot Z^{-1})^n \end{aligned} \quad (2.14)$$

Pelas propriedades das séries geométricas.

$$H(Z) = aT_s \cdot \frac{(e^{-aT_s} \cdot Z^{-1})^\infty - 1}{(e^{-aT_s} \cdot Z^{-1}) - 1}$$

Que converge se $Z > e^{-aT_s}$.

$$H(Z) = \frac{aT_s}{1 - e^{-aT_s} \cdot Z^{-1}} \quad (2.15)$$

A equação 2.15 é descrita pela seguinte equação às diferenças:

$$y[n] = aT_s \cdot x[n] + e^{-aT_s} \cdot y[n-1] \quad (2.16)$$

A equação 2.16 pode ser usada para implementar o filtro digital e usa apenas um elemento de memória e dois coeficientes, sendo bastante menos exigente computacionalmente do que um filtro FIR com as mesmas características. A tentativa de manter a fidelidade da resposta impulsional, com este método, leva a que a resposta em frequência sofra *aliasing* como se explicou na secção 2.5.

Método da transformação bilinear

O problema do *aliasing*, provocado pelo método anterior, é corrigido ao usar uma transformação bilinear. Esta transformação procura manter a fidelidade da resposta em frequência na passagem para o digital. Para isto, toda a resposta em frequência do filtro contínuo é mapeada na gama de frequências de 0 a πf_s no domínio digital, como mostra a figura 2.5.

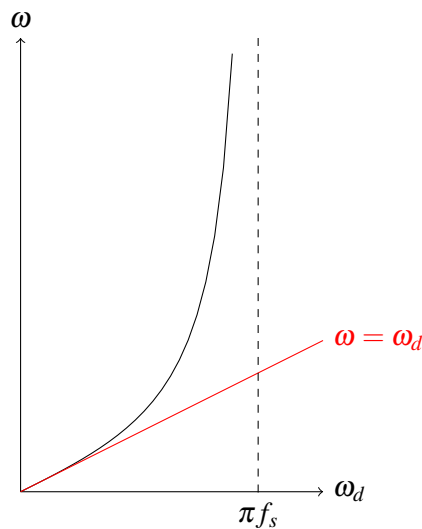


Figura 2.5: Mapeamento bilinear

A função,

$$\omega = 2f_s \cdot \tan\left(\frac{\omega_d}{2f_s}\right) \quad (2.17)$$

é usada para fazer o mapeamento, sendo ω a frequência analógica, ω_d a frequência digital e f_s a frequência de amostragem. A derivada desta função em relação a ω_d tende para a unidade na origem, o que faz com que a resposta em frequência digital seja mais parecida com a original para frequências muito menores do que a frequência de *Nyquist*. Com simples manipulação da equação 2.17, e realizando as substituições $s = j\omega$ e $Z = e^{j\omega_d f_s}$ chega-se ao seguinte:

$$s = 2f_s \cdot \frac{1 - Z^{-1}}{1 + Z^{-1}} \quad (2.18)$$

A relação anterior é usada para fazer uma mudança de variável de funções de transferência em *Laplace* para *Z*. Com a transformada *Z* do filtro, é trivial chegar à equação das diferenças. Mais uma vez, podem haver diferenças na resposta do filtro em relação ao original, se a frequência de amostragem não for suficientemente elevada, embora os filtros não produzam *aliasing*.

Capítulo 3

Processador integrado para navegação de AUVs

O hardware descrito na secção 2.4.2 possui muitos problemas. Os filtros, ao serem feitos com elementos passivos como condensadores e bobinas em especial, produzem fortes campos magnéticos e com eles muita interferência electromagnética nos sistemas próximos. Para atenuar tais interferências, todas as bobinas e transformadores estão blindados com folhas de alumínio. Para além da interferência electromagnética, os parâmetros dos elementos passivos variam com a temperatura, fazendo com que a frequência para qual os filtros estão sintonizados vá variando ao longo da missão, especialmente nas bóias, que podem aquecer bastante em dias de sol. A sintonização dos filtros também é um problema, porque é necessária a troca dos componentes passivos sempre que se queira mudar uma das frequências.

O objectivo deste processador integrado é provar que este problema em concreto pode ser resolvido digitalmente, com tecnologias actuais, abrindo portas para novos esquemas de detecção que podem melhorar muito a navegação de AUVs.

3.1 Esquema de Detecção

Sendo a escolha do esquema de detecção o primeiro passo para a mudança do sistema para digital, escolheu-se implementar um esquema que seja o mais semelhante possível ao que existe no MARES. Este é baseado em filtros passa-banda e detectores de envolvente como descrito em 2.4.2. Existem melhores esquemas de detecção que vão ser mencionados no capítulo 5.2 mas, para que a mudança fosse o menos evasiva possível em todo o *hardware* e *software* do MARES, optou-se por usar o mesmo esquema.

Como já foi dito, existem pelo menos oito filtros sintonizados a frequências diferentes, estando cada um ligado a um detector de envolvente. As saídas do detector de envolvente serão usadas para discriminar a existência de um sinal em cada filtro baseando-se para isso num nível de detecção.

Se o valor do sinal de um detector de envolvente for superior ao nível de detecção, existe sinal na frequência respectiva, caso contrário não existe.

3.1.1 Filtros Passa-Banda

A análise da resposta transitória dos filtros é necessária para que se possa analisar o sistema convenientemente. Sem esta análise não era possível saber o tempo de resposta dos filtros aos sinais recebidos sem simular o sistema, nem era possível calcular o valor dos limiares de detecção convenientemente. A análise que se segue foi tentada em Matlab, mas não se conseguiram expressões tratáveis, sendo posteriormente feita à mão.

Os filtros passa-banda, que já existiam no MARES, são filtros de segunda ordem com elevado factor de qualidade e exibem a seguinte função de transferência:

$$H(s) = \frac{K \cdot \frac{\omega_0}{Q} \cdot s}{s^2 + \frac{\omega_0}{Q} \cdot s + \omega_0^2} \quad (3.1)$$

onde:

K = ganho

Q = factor de qualidade

ω_0 = frequência central do filtro (rad)

$B = \frac{\omega_0}{Q}$ = Largura de Banda (rad)

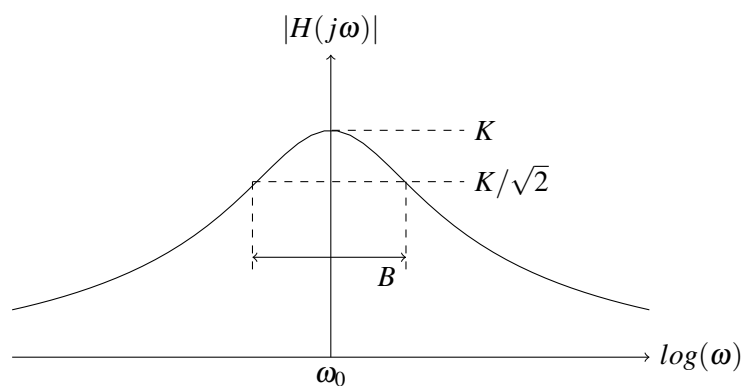


Figura 3.1: Resposta em frequência dos filtros passa-banda.

A figura 3.1 mostra a resposta em frequência dos filtros. Qualquer frequência que não seja a frequência central é atenuada, sendo mais atenuada quanto maior for o desvio de ω_0 . A largura de banda é medida quando a atenuação do filtro é igual a $K/\sqrt{2}$.

Analisando a expressão 3.1, conclui-se que se o factor de qualidade for $Q > 0.5$ o sistema tem dois pólos complexos em:

$$s = -\frac{1}{\tau} \pm j\omega' \quad (3.2)$$

onde, $\tau = 2/B$ e $\omega' = \sqrt{\omega_0^2 - (B/2)^2}$.

Os sinais a serem recebidos pelos filtros são, idealmente, sinusóides puras. A resposta a tal entrada tem a seguinte transformada de *Laplace*:

$$Y(s) = \frac{K \cdot \frac{\omega_0}{Q} \cdot s}{(s + \frac{1}{\tau} + j\omega') \cdot (s + \frac{1}{\tau} - j\omega')} \cdot \frac{s}{s^2 + \omega^2} \quad (3.3)$$

Aplicando a transformada inversa, a resposta transitória obtida é a seguinte:

$$y(t) = |K_a| \cdot \cos(\omega t + \angle K_a) \cdot u(t) - |K_b| \cdot e^{-t/\tau} \cdot \cos(\omega' t + \angle K_b) \cdot u(t) \quad (3.4)$$

K_a e K_b são números complexos em função de ω . K_a é exactamente igual à transformada de *Fourier* dos filtros $H(j\omega)$:

$$K_a = \frac{K \cdot B \cdot j\omega}{B \cdot j\omega + (\omega_0^2 - \omega^2)} \quad (3.5)$$

O complexo K_b é igual a:

$$K_b = \frac{2}{\tau\omega'} \cdot \frac{2\omega'\omega^2/\tau + j[2(\omega/\tau)^2 + \omega_0^2(\omega_0^2 - \omega^2)]}{(2\omega/\tau)^2 + (\omega_0^2 - \omega^2)^2} \quad (3.6)$$

Resposta Adaptada

As equações 3.4, 3.5 e 3.6 podem ser usadas para saber a resposta dos filtros a uma entrada frequência igual à frequência central de amplitude A . Substituindo ω , nas três equações, por ω_0 , tem-se o seguinte resultado:

$$y(t) = A \cdot K \cdot u(t) \left[\cos(\omega_0 t) - e^{-t/\tau} \frac{\omega_0}{\omega'} \cdot \cos \left(\omega' t + a \cos \left(\frac{\omega'}{\omega_0} \right) \right) \right] \quad (3.7)$$

O que mostra que a resposta a uma frequência adaptada é um sinal modelado, com frequência ω_0 e envolvente aproximadamente:

$$A \cdot K \cdot u(t) \left[1 - e^{-t/\tau} \right] \quad (3.8)$$

Esta envolvente apresenta as mesmas características que um passa-baixo de primeira ordem com constante de tempo $\tau = 2/B$. A figura 3.2 mostra a resposta directa do filtro a preto e a respectiva envolvente a tracejado.

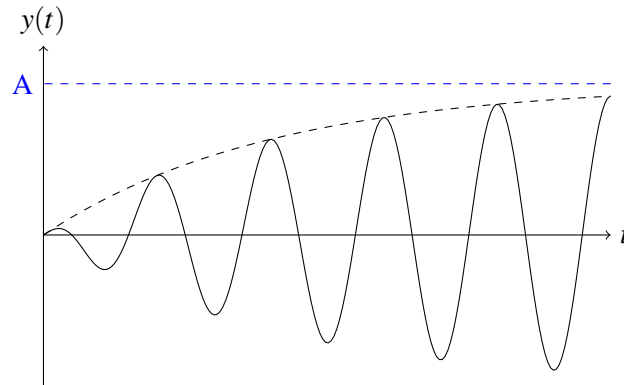


Figura 3.2: Resposta adaptada de um filtro passa-banda.

A partir da expressão 3.8, o tempo de resposta mínimo destes filtros (T_r) pode ser calculado simplesmente com o nível de detecção (N_d), a largura de banda (B) e a amplitude máxima do sinal de entrada (A) pela expressão:

$$T_r = \frac{2}{B} \cdot \ln \left(\frac{A}{A - N_d} \right) \quad (3.9)$$

Resposta Desadaptada

A resposta a sinais com frequência ω , desadaptada ao filtro pode ser vista como uma envolvente com a seguinte forma:

$$A \cdot K \cdot u(t) \left[|H(j\omega)| - |K_b| \cdot e^{-t/\tau} \cos \left((\omega' - \omega)t + \angle(K_b) - \angle(H(j\omega)) \right) \right] \quad (3.10)$$

que se assemelha à resposta de um filtro passa-baixo de segunda ordem. Como mostra a figura 3.3, a envolvente apresenta oscilação com frequência ω' e, mais importante ainda, apresenta *overshoot*. Quanto mais desadaptado for o sinal ao filtro, maior será o *overshoot* em relação ao valor final. Em contrapartida, o nível final do sinal diminui quanto maior for a diferença entre as duas frequências, como mostra a resposta em frequência dos filtros na figura 3.1.

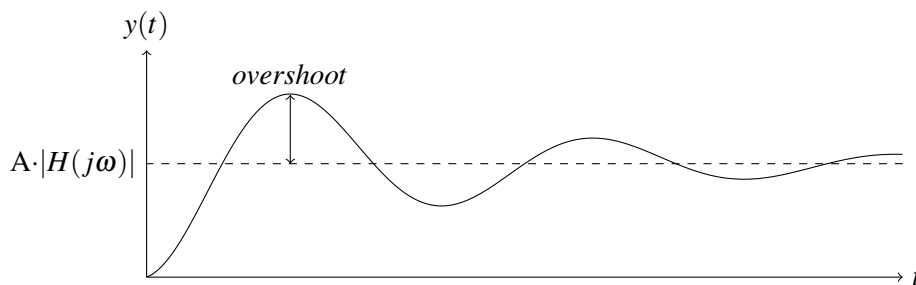


Figura 3.3: Resposta desadaptada de um filtro passa-banda.

Quando é enviado um sinal apenas um dos filtros estará adaptado a ele, todos os outros deverão ter uma resposta que seja inferior ao nível de detecção. Como o nível máximo de um canal desadaptado coincide com o *overshoot* de que se falou, é necessário obter uma expressão que permita calcular este nível máximo facilmente. Os mínimos e máximos da expressão 3.10 são obtidos com os zeros da sua derivada. É fácil de concluir que o máximo coincide com o segundo zero da derivada, por isso o instante em que ocorre *overshoot* é obtido com a seguinte formula:

$$t_{ov} = \frac{\angle H(j\omega) - \angle K_b - \text{atan}\left(\frac{1}{\tau(\omega' - \omega)}\right)}{\omega' - \omega} + \frac{\pi}{|\omega' - \omega|} \quad (3.11)$$

Ao realizar a substituição $t = t_{ov}$ na equação 3.10, obtém-se o valor máximo da resposta a uma entrada desadaptada. O valor máximo depende, obviamente, da amplitude do sinal de entrada A que, por sua vez, depende da distância percorrida pelo sinal ou, por outras palavras, da atenuação do sinal. Isto será levado em conta no dimensionamento dos níveis de decisão mais adiante.

3.1.2 Ruído de Banda Estreita

É possível demonstrar que ruído branco ($n(t)$) filtrado por um filtro passa banda pode ser modelado como uma sinusóide com frequência igual à frequência central do filtro (ω_0), envolvente ($r(t)$) e fase ($\Phi(t)$) aleatórias [10].

$$n(t) = r(t) \cdot \cos(\omega_0 \cdot t + \Phi(t)) \quad (3.12)$$

A variável aleatória que modela a envolvente do ruído, com potência média σ^2 tem uma função de distribuição de probabilidade (figura 3.4) igual a:

$$f_R(r) = \begin{cases} \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} & , r \geq 0 \\ 0 & , r < 0 \end{cases} \quad (3.13)$$

Com a distribuição de probabilidade da envolvente do ruído, é possível determinar probabilidades de passagem da envolvente pelos níveis de decisão, na presença de ruído. A envolvente de uma sinusóide modelada, somada ao ruído, não é a soma directa da envolvente do sinal com a envolvente do ruído, porque os dois nem sempre estão em fase. Isto faz com que a distribuição de probabilidade da diferença, entre a envolvente do sinal com e sem ruído, seja inferior à descrita pela equação 3.13, contudo, ela é usada como majorante.

3.1.3 Níveis de Decisão

Todo o sistema deve estar dimensionado para que, na existência de um sinal com frequência na gama de ressonância do transdutor, apenas uma detecção seja realizada. Isto pode ser conseguido

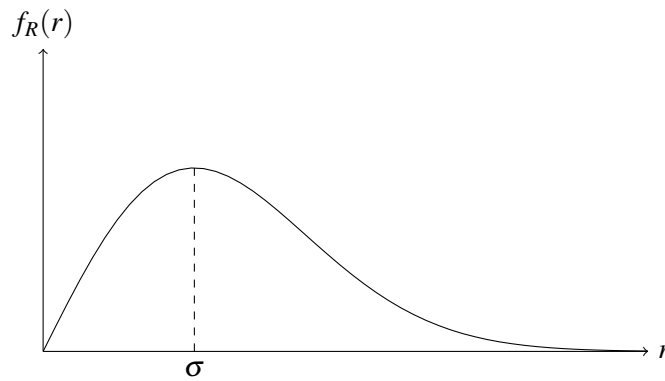


Figura 3.4: Distribuição de probabilidade de ruído de banda estreita.

jogando com a a largura de banda dos filtros e com os níveis de decisão. Escolhida a largura de banda, os níveis são dimensionados para que sejam maiores do que a amplitude máxima recebida, atenuada na frequência onde as respostas em frequência de dois canais adjacentes se interceptam. Na figura 3.5 são mostradas duas respostas em frequência de dois canais adjacentes. O nível de decisão relativo à amplitude máxima está marcado a tracejado. Se a frequência do sinal se desviar da frequência de intercepção, a amplitude resultante só vai ser maior do que o nível de decisão num canal.

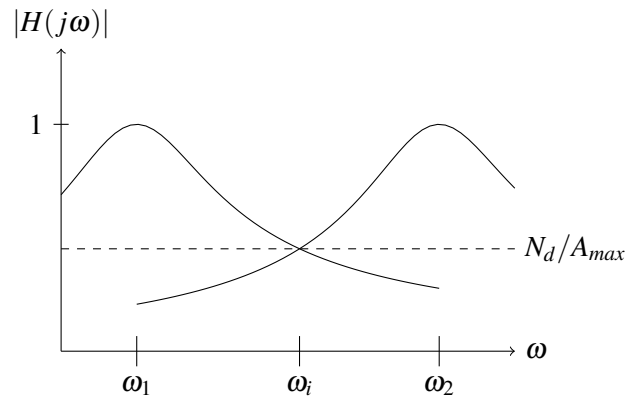


Figura 3.5: Dimensionamento do nível de detecção.

A intercepção das respostas em frequência de dois canais, com frequência central ω_1 e ω_2 , se a largura de banda for a mesma, acontece na frequência:

$$\omega_i = \sqrt{\frac{\omega_1^2 + \omega_2^2}{2}} \quad (3.14)$$

O nível de decisão deve ser maior do que o *overshoot* obtido à frequência de intercepção quando a entrada é uma sinusóide com amplitude máxima. Significa que quanto maior for o nível de sinal

máximo esperado, maiores vão ser os níveis de decisão. A amplitude máxima vai ser estudada na secção seguinte.

O ruído também é relevante para a escolha do nível de decisão, pelo que, somado ao nível obtido anteriormente, deve-se somar uma margem de erro para que a probabilidade de falsa detecção seja inferior a um certo limite. Quanto maior for a potência média do ruído, maior será esta margem. O cálculo exacto desta pode ser obtido usando a distribuição de probabilidade descrita em 3.1.2. A potência do ruído tem de ser medida antes da missão para que os níveis de decisão possam ser ajustados.

3.1.4 Alcance

O alcance do sistema tem que ver com o nível de decisão. O sistema perde totalmente a capacidade de detectar sinais, quando o nível do sinal que chega ao submarino for inferior ao nível de decisão. O sinal atenua, como visto na secção 2.2, com a distância percorrida e, a uma certa distância, quando o nível de sinal é inferior ao nível de decisão, diz-se que o sistema perde o alcance. Como visto na secção 3.1.3, os níveis de decisão são dimensionados com base na amplitude do sinal de entrada. Ora, à distância a que o sistema perde o alcance, como o nível de sinal é mais baixo, os níveis de decisão também pode ser mais baixos. Se houvesse um sistema que regulasse os níveis de decisão ou o ganho automaticamente com a distância, o alcance do sistema podia ser melhorado.

Actualmente no MARES, não existe nenhum mecanismo de regulação do ganho ou dos níveis de decisão com a distância, por isso a entrada dos filtros está amplificada de maneira a que atinjam o nível máximo para uma distância máxima. Para distâncias inferiores a essa, a sinusóide à entrada dos filtros satura, transformando-se numa onda quadrada. Esta amplificação permite que o nível do sinal seja sempre superior aos níveis de detecção até uma distância máxima definida, porém, por causa da saturação do sinal, o sistema perde em precisão. A solução ideal seria implementar um sistema automático de regulação do ganho com a distância.

A solução usada para resolver o problema do alcance implica que, no máximo, o sinal recebido será amplificado e saturado ao ponto de se tornar uma onda quadrada. É importante, por isso, saber o valor máximo à saída dos filtros passa-banda quando filtram uma onda quadrada para se dimensionar os níveis de decisão (secção 3.1.3). Aplicando a transformada de *Fourier* a uma onda quadrada com frequência ω_0 sem componente contínua e amplitude A (amplitude de saturação), é fácil de demonstrar que o espectro do sinal é:

$$X(j\omega) = \sum_{k=-\infty}^{\infty} \frac{4A}{1+2k} \delta(\omega - \omega_0(1+2k)) \quad (3.15)$$

ou seja, só existem componentes a frequências múltiplas ímpares de ω_0 . Como as frequências múltiplas ultrapassam em muito a largura de banda dos filtros¹, vão ser atenuadas quase na totalidade restando apenas as componentes $4A\delta(\omega - \omega_0)$ e $4A\delta(\omega + \omega_0)$. A transformada inversa

¹ A largura de banda dos filtros é, no máximo, de 200Hz e a menor frequência de um sinal é de 20kHz.

destas componentes resulta numa sinusóide com amplitude $4A/\pi$, que é a amplitude máxima que alguma vez vai sair de algum dos filtros se tiverem ganho unitário. Os níveis de decisão podem assim ser fixos e calculados em relação a este máximo.

3.1.5 Precisão

A precisão deste sistema está obviamente ligada ao tempo de resposta dos filtros. Para que o sistema seja o mais preciso possível este tempo deveria de ser constante, ou ser calculado com precisão. A variação neste tempo de resposta, se o nível de detecção for sempre constante, depende de muitos factores, entre eles:

Atenuação do sinal Se o nível de sinal máximo à saída dos filtros variar, as detecções vão acontecer mais cedo ou mais tarde do que o esperado, o que provoca erro na medida da distância, que depende da medição correcta do tempo. O nível de sinal, como já foi visto na secção 2.2, atenua com a distância percorrida. Esta atenuação provoca uma diminuição do nível máximo apresentado à saída dos detectores de envolvente. Contudo, a atenuação pode ser calculada, de forma aproximada, se houver uma estimativa da distância que o sinal tem de percorrer 2.2. Com este procedimento, seria possível calcular o novo tempo de resposta, atenuando assim o erro.

Desvio de frequência Desvios na frequência do sinal enviado em relação à frequência do filtro respectivo, também levam a uma diminuição do sinal à saída dos detectores de envolvente (ver secção 3.1.1). Os desvios de frequência podem ser provocados por erros no oscilador do circuito digital a enviar os sinais, ou por erros nos valores dos componentes passivos que criam a frequência central do filtro, no caso dos filtros serem digitais, são também erros na frequência do oscilador. Embora estes erros pudessem ser reduzidos calibrando as frequências do emissor com as do receptor, o desvio de frequência predominante provem do efeito de *Doppler*, que é muito maior do que qualquer erro que possa existir na adaptação de frequências. O MARES atinge uma velocidade máxima v de 2 m/s, e a velocidade do som na água c ronda os 1500 m/s. Se o AUV se estiver a deslocar na mesma direcção que a propagação do sinal, a frequência inicialmente transmitida pelo submarino f , será afectada pelo efeito de *Doppler* por $f_{Doppler} = f(\frac{c}{c \pm v})$. A frequência máxima transmitida pelo AUV é de 29kHz, pelo que o desvio máximo a essa frequência pelo efeito de *Doppler* é de, aproximadamente, $\pm 39\text{Hz}$. O desvio dos osciladores de cristal ronda os 100ppm, que à frequência de 29kHz representa um desvio de $\pm 3\text{Hz}$ que é desprezável face ao desvio pelo efeito de *Doppler*. Erros no tempo de resposta dos filtros devido a este problema são, à partida, impossíveis de contornar com este esquema de detecção, já que as bóias não sabem a velocidade do submarino. O aumento da largura de banda dos filtros, porém, atenua este erro.

Ruído O ruído apanhado pelo transdutor, é atenuado pelos filtros passa-banda mas não completamente. Por isso, as variações induzidas pelo ruído à saída do detector de envolvente

provocam erros no tempo de resposta do filtro. Sendo τ a constante de tempo do filtro e e uma variável aleatória que representa o ruído, ao fazer uma aproximação a uma recta, a saída do detector de envolvente seria $y(t) = t/\tau + e$. Admitindo um nível de detecção N_d , o tempo de resposta é $t = \tau(N_d - e)$. A variação ao tempo de resposta esperado torna-se $\Delta t = \tau e$. Por isto, quanto maior for a constante de tempo, maior seria de esperar a variação no tempo de resposta e, com isso, maior o erro na distância. Na realidade, para ter uma constante de tempo menor, é preciso ter uma largura de banda maior, que leva a mais ruído e mais erro. Existe, portanto, um compromisso na escolha da largura de banda.

Saturação do Sinal Por causa do problema do alcance 3.1.4, o sinal a ser filtrado é amplificado para detectar sinais enviados de uma distância máxima. Esta amplificação é de alguns milhares no MARES. Quando nenhum sinal está a ser enviado, obviamente que o ruído é amplificado também mas, na presença de sinal, até o ruído é saturado. Este fenómeno não linear, pode fazer com que a onda quadrada passe pelo zero mais tarde ou mais cedo por causa do ruído, alterando a frequência do sinal. Se isto acontecer, provoca erro. Embora esta teoria não tenha sido devidamente estudada, existem dados que a corroborem no capítulo 4.

A precisão do sistema tem muitos factores a considerar mas é possível obter um valor máximo. Admitindo que o nível de sinal ultrapassa o nível de detecção dentro do tempo de duração do sinal (2ms), o erro máximo do tempo de resposta é $e_t = 0.002 - Tr$ em segundos. O erro máximo equivale então a 2ms que convertido em distância resulta em 3m. Na realidade, actualmente no MARES, consegue-se uma precisão inferior a um metro. Se a variação do nível do sinal e a variação de frequência forem eliminados, ao manter o AUV parado, conseguem-se desvios entre medidas (repetibilidade) inferiores a alguns centímetros.

3.1.6 Implementação Digital

Qualquer um dos métodos descritos na secção 2.6 pode ser usados para implementar estes filtros passa-banda digitalmente, mas como se quer uma resposta o mais semelhante possível aos analógicos, os filtros FIR estão fora de questão. Para além de precisarem de uma grande capacidade computacional, as vantagens que eles exibem não se aplicam neste caso concreto. Tanto os dois métodos descritos em 2.6.2 serviam para aproximar a resposta dos filtros analógicos do MARES com filtros IIR mas, o método da transformação bilinear apresenta melhores propriedades neste caso. A atenuação à frequência central do filtro deve ser nula e, mais importante ainda, a atenuação nas frequências adjacentes deve ser a maior possível. O *aliasing* apresentado pelos filtros gerados com o método da invariância ao impulso não gera uma atenuação óptima nas frequências adjacentes à central. Os filtros gerados pela transformação bilinear não tem este tipo de problemas, por isso a escolha cai sobre eles.

Os coeficientes dos filtros podem ser obtidos facilmente usando ferramentas como *MatLab* mas escolheu-se obter as expressões de calculo com base na largura de banda, frequência central e

ganho para que, em fases posteriores, se pudessem mudar os parâmetros em tempo real, poupando tempo no uso da ferramenta.

Usando o método da transformação bilinear, usa-se a substituição 2.18 na equação 3.1 e obtém-se a transformada Z do filtro que é usada para obter a seguinte equação às diferenças:

$$y[n] = -a \cdot y[n-1] - b \cdot y[n-2] + c \cdot x[n] - c \cdot x[n-2] \quad (3.16)$$

Os parâmetros a , b e c podem ser calculados da seguinte forma:

$$a = \frac{2\omega_0(\Omega_0^2 - 4)}{\omega_0(\Omega_0^2 + 4) + 2\Omega_0 B} \quad (3.17)$$

$$b = \frac{\omega_0(\Omega_0^2 + 4) - 2\Omega_0 B}{\omega_0(\Omega_0^2 + 4) + 2\Omega_0 B} \quad (3.18)$$

$$c = \frac{2\Omega_0 B}{\omega_0(\Omega_0^2 + 4) + 2\Omega_0 B} \cdot G \quad (3.19)$$

Onde ω_0 é a frequência central do filtro, $\Omega_0 = \frac{\omega_0}{f_s}$ é a frequência normalizada do filtro e B é a largura de banda do filtro (angular), medida quando a atenuação do filtro é de 3dB. G é o ganho do filtro na frequência central. Como as frequências no domínio analógico são afectadas pela transformação bilinear, no domínio digital, a frequência central vai ser alterada. Para contrariar esta mudança, e para que a frequência central no domínio digital seja exactamente a esperada, deve-se afectar a frequência analógica pela transformação inversa que a digital irá sofrer. Portanto a variável ω_0 usadas nas equações anteriores não é exactamente a frequência central esperada mas sim a seguinte:

$$\omega_0 = 2f_s \cdot \tan\left(\frac{\omega}{2f_s}\right) \quad (3.20)$$

onde ω é a frequência central exacta.

3.2 Plataforma de Desenvolvimento

Suzaku SZ-130

A plataforma escolhida para a realização do projecto foi uma suzaku SZ-130 da AtmarkTechno (figura 3.6). É uma placa com, sensivelmente, o tamanho de um cartão de crédito, baseada numa Field Programmable Gate Array (FPGA) spartan3E-1200 [11]. Possui interface ethernet, rs-232, JTAG para programação, cerca de 86 pinos de entrada e saída para uso global, 16Mega Byte (MB) de SDRAM e 8MB de memória flash [12]. As configurações da FPGA, são guardadas em memória flash e, no arranque, são carregadas para a FPGA automaticamente por circuitos dedicados.

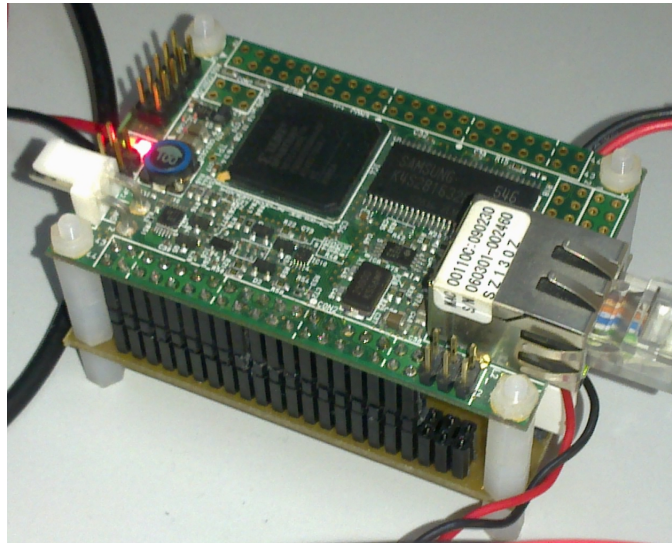


Figura 3.6: Suzaku-SZ130

Valores de Fábrica

A plataforma vem de fábrica com a **FPGA** programada com um Microblaze, um *soft-processor* proprietário da Xilinx [13] e hardware para controlar os periféricos existentes na placa (ethernet, rs-232, **SDRAM**, etc). O software que o Microblaze corre no arranque consiste num bootloader que lê a memória flash, copia o seu conteúdo para memória **SDRAM** e corre-o. Incluído no conteúdo da flash está outro bootloader e uma distribuição minimalista de linux chamada μ Linux. O segundo bootloader é necessário porque tem rotinas específicas necessárias para o arranque do kernel do linux.

Software de desenvolvimento

O software usado para o desenvolvimento do hardware existente na **FPGA** foi uma combinação da plataforma de desenvolvimento de sistemas embebidos da Xilinx, o EDK, com a plataforma de projecto de hardware para as **FPGAs** da Xilinx, o ISE. A arquitectura do sistema, incluindo os módulos proprietários eram definidos no EDK. O hardware criado à medida para o projecto foi desenvolvido no ISE, em verilog, onde apenas era criada uma rede de ligações (netlist) entre componentes da **FPGA** que era exportada para o EDK onde todo o projecto era mapeado em componentes físicos da **FPGA**. O projecto inicial da suzaku para esta plataforma foi modificado pelo professor José Carlos Alves e o Eng. Eduardo Sousa e foi gentilmente cedido para o desenvolvimento do trabalho. A modificação consistiu na criação de um novo periférico que faz a interface com os pinos externos da **FPGA** e implementa registos e uma memória que são partilhados com o Microblaze através de um barramento proprietário chamado On-chip Peripheral Bus (**OPB**). O software para a distribuição de linux foi compilado com ferramentas de compilação **GNU** para o Microblaze.

Programação

Para se programar um novo *bitstream* na **FPGA** pode-se usar software incluído na distribuição de linux que programa a memória flash. O *bitstream* obtido pelas ferramentas da Xilinx contém bytes iniciais, que são retirados usando um programa gentilmente cedido pelo professor José Carlos Alves, para que possa ser usado. Este método de programação só pode ser usado se o sistema operativo embebido estiver a funcionar. Caso aconteça alguma coisa ao linux, a **FPGA** e o sistema operativo podem ser reprogramados usando um dos bootloaders. O caso mais frequente, quando se programa um novo *bitstream* na **FPGA**, é o circuito digital presente na **FPGA** deixar de funcionar. Neste caso nenhum dos métodos de programação descritos funciona, sendo a única opção programar a memória flash directamente por interface série ou por **JTAG**. Como o software existente para programar a flash não existe em ambiente Linux², criei um programa para escrever a flash por porto paralelo, poupando tempo nas mudanças entre sistemas operativos.

3.3 Interface Acústica

Para o processamento digital dos sinais dos transdutores foi necessária a concepção de uma interface analógico digital. Foi desenhada e fabricada uma placa, com a ajuda do professor José Carlos Alves, que realiza a interface. A **PCB** possui um conversor Analog/Digital (**A/D**), alguns **LEDs**, e amplificadores operacionais para a amplificação do sinal e para lhe adicionar componente contínua. O conversor **A/D** usado é um AD7922 [14] com doze bits de resolução e interface série e dois canais. Este mesmo **A/D** permite uma taxa de amostragem de **1MSPS**, que é mais do que suficiente para a aplicação. Os amplificadores estão ligados na montagem inversora, com o terminal positivo ligado a um divisor de tensão, regulável através de um potenciómetro. A resistência realimentada é um potenciómetro digital controlado por um barramento Inter-Integrated Circuits (**I2C**). São amplificados dois canais que são ligados ao **A/D**. O ganho máximo destes amplificadores é de dez vezes. A placa está ligada à Suzaku através dos pinos de entrada e saída.

A figura 3.7 mostra a interface global com o **AUV** e o caminho dos dados. O sinal analógico captado é amplificado e amostrado pelo **A/D**, sendo transportados por interface série até à **FPGA**, onde são processados. O único meio de comunicação entre o sistema e o submarino é uma ligação RS-232, que vai ser usada para implementar um protocolo de comunicação, que já existia no **MARES**, e estava a ser usado pelo sistema acústico. Através das entradas e saídas de uso geral, o sistema controla o envio de sinais. A interface de envio consistem em apenas quatro sinais, três deles discriminam o canal a enviar e um serve para iniciar o envio. Esta interface era desnecessária já que o sistema é bem capaz de gerar os sinais eliminando a necessidade de outro sistema.

A figura mostra também a possibilidade de alteração do ganho dos amplificadores por **I2C**. Como foi dito na secção 3.1.4 seria óptimo regular o ganho com a distância do submarino às bóias, por isso optou-se por integrar na interface a possibilidade de alteração do ganho. Embora

²Este trabalho foi realizado totalmente em Linux.

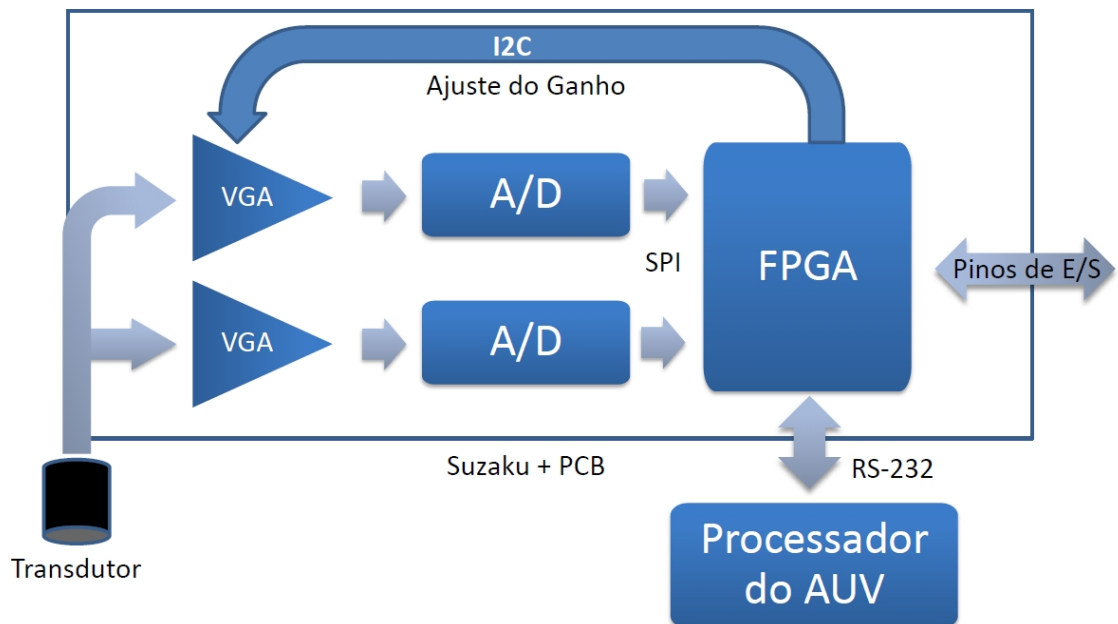


Figura 3.7: Interface com o [MARES](#).

todo o hardware e software estivesse pronto para fazer esta regulação automática, teriam de haver mudanças no hardware do [MARES](#) que eram de evitar pelo tempo disposto para a realização do trabalho pelo que, no final, se optou pela solução antiga do [MARES](#).

3.4 *FPGA*

3.4.1 Sistema Embebido

A estrutura interna do sistema digital implementado na *FPGA* está exposta na figura 3.8. O sistema usa um processador Microblaze a correr Linux e hardware dedicado para realizar a detecção dos sinais vindos do transdutor. Como se vê, é o processador que processa o protocolo de comunicação série com o *AUV*, controla todo o hardware dedicado através de registos e memórias partilhados mas não tem mais controlo algum com nenhum pino externo da *FPGA*. O processador é usado também para realizar alguns cálculos em virgula flutuante para ajudar no processo de *debug*.

Toda a arquitectura foi implementada usando o EDK. Não estão presentes na figura as interfaces com os periféricos que controlam a interface ethernet, a flash *SPI* e as memórias *SDRAM* porque estas foram apenas acessórios de desenvolvimento. É importante dizer que o relógio injetado na *FPGA* é multiplicado por um Digital Clock Manager (*DCM*) em quatorze vezes. O relógio externo tem uma frequência de 3.6864MHz que multiplicado atinge os 51.6MHz. Esta frequência de relógio não pode ser aumentada porque a versão do Microblaze usada não atinge velocidades superiores nesta *FPGA*.

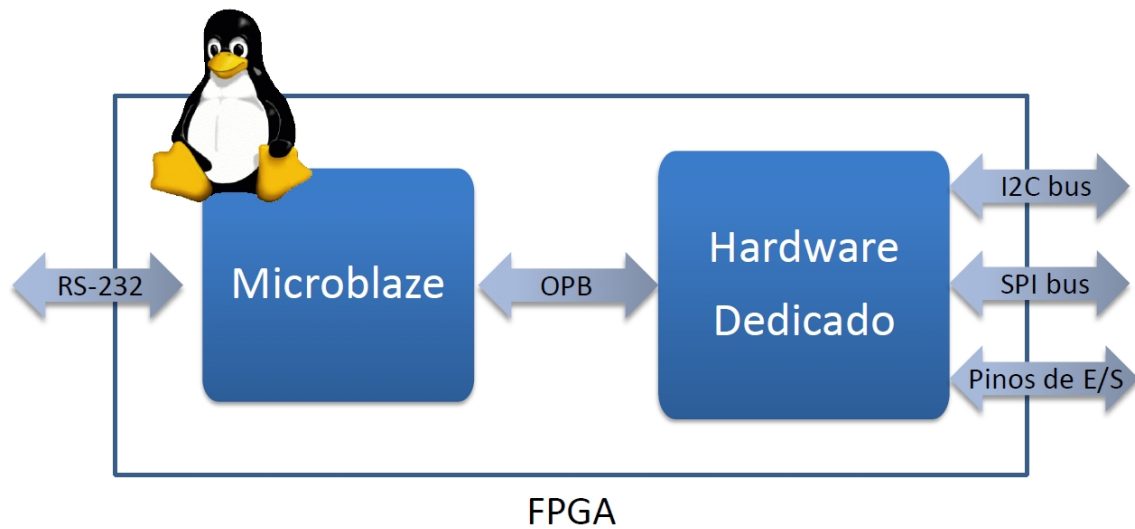


Figura 3.8: Arquitectura geral da [FPGA](#).

3.4.2 Hardware Dedicado

A arquitectura global do hardware digital implementado está representada na figura 3.9. O sistema trabalha em torno da cadeia de detecção dos sinais (filtros passa-banda, detectores de envolvente, níveis de detecção), representada como Digital Signal Processor ([DSP](#)) na figura e que é alimentada por dados provenientes do [A/D](#). O objectivo do sistema é detectar sinais acústicos nos vários canais e informar o Microblaze da sua presença e do tempo em que aconteceram.

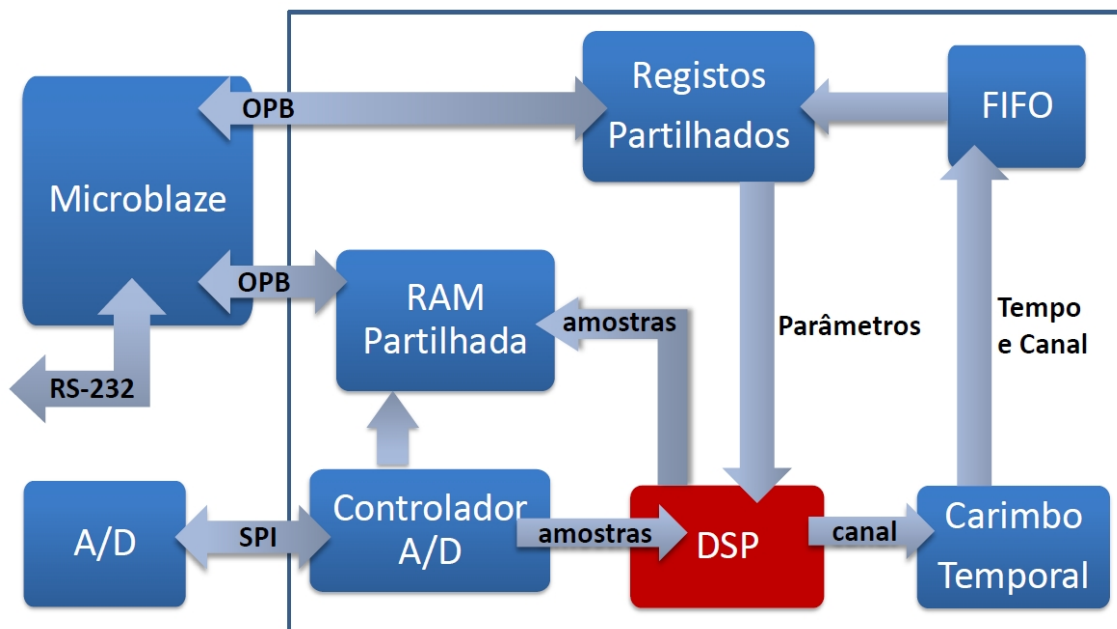


Figura 3.9: Arquitectura do hardware dedicado.

Controlador A/D

O controlador dos conversores A/D foi feito em *verilog* e está continuamente a pedir novos dados a uma frequência fixa de 92 ciclos de relógio por amostra. Isto significa que a frequência de amostragem real é de, aproximadamente, 561kHz. Foi visto na secção 2.6.2 que se a frequência de amostragem não for suficientemente elevada para além da frequência mínima de amostragem pelo teorema de *Nyquist*³, a resposta em frequência e ao impulso é diferente da resposta analógica. Esta diferença na resposta dos filtros invalidaria toda a análise feita até ao momento e, por causa disso, foram realizados testes em Matlab com filtros digitais iguais aos descritos em 3.1.6 e várias frequências de amostragem concluindo-se que, a frequências superiores a 250kHz, a análise analógica mantinha-se. A frequência usada (561kHz) é então suficiente para fazer a amostragem dos sinais.

Numa fase inicial, o módulo foi usado para preencher uma memória interna da *FPGA* (Block Random Access Memory (*BRAM*)) partilhada com o processador, que são pequenas mas funcionam à mesma velocidade que toda a lógica interna. Estas amostras eram depois lidas por um pequeno programa de teste que as guardava num ficheiro em memória, usando para isso o sistema operativo. Este ficheiro podia ser lido por ethernet ou RS-232 para assim se testar o funcionamento do módulo e do A/D. Depois de validado todo o seu funcionamento o módulo foi configurado para entregar amostras aos filtros digitais a uma frequência constante.

Interface I2C

Para controlar os potenciómetros digitais, foi criado um módulo que implementa o protocolo I2C. Já existiam módulos prontos a usar que o fizessem, mas eram incompatíveis com os potenciómetros usados. O módulo criado usa uma máquina de estados que envia os dados presentes num registo e mostra os dados recebidos noutra. Os dados a serem enviados são copiados para um *shift register* que envia os dados libertando o registo, possibilitando assim o envio contínuo dos dados que era a razão da incompatibilidade de outros módulos. Ambos os registos, assim como um registo de controlo, foram partilhados com o processador para que toda a operação seja controlada por ele. Como o envio de dados é feito a uma frequência de cerca de 100kHz, o processador pode facilmente acompanhar o envio e recepção dos dados do módulo. Todo o hardware e software criado à volta dos potenciómetros digitais tinha como propósito implementar algum mecanismo de ajuste automático de ganho embora não tenha sido usado como foi falado na secção 3.1.4. A única tarefa deste módulo agora é a programação do ganho dos amplificadores para um valor predefinido quando o sistema arranca.

Filtros Passa-Banda

A componente principal de todo o sistema é o módulo que implementa a cadeia de processamento de sinal digitalmente (*DSP* na figura 3.9). Incluídos na cadeia estão os filtros passa-banda que implementam a equação às diferenças descrita na equação 3.16 de modo sequencial. A

³A frequência de amostragem mínima para esta aplicação seria de 60kHz

frequência de amostragem é cerca de cem vezes menor do que a frequência de relógio dos circuitos digitais o que faz com que o sistema possa usar mais ou menos cem ciclos de relógio para o processamento de cada amostra. Este número de ciclos permite que se desenvolva uma arquitetura que minimize a área ocupada. A minimização da área tem em vista a redução do consumo do sistema, isto porque grande parte do consumo numa **FPGA** é estático levando a que **FPGAs** mais pequenas consumam muito menos. Embora não se possa diminuir o tamanho da **FPGA** da plataforma de desenvolvimento, uma solução futura poderá usar uma plataforma mais pequena e menos exigente ao nível do consumo.

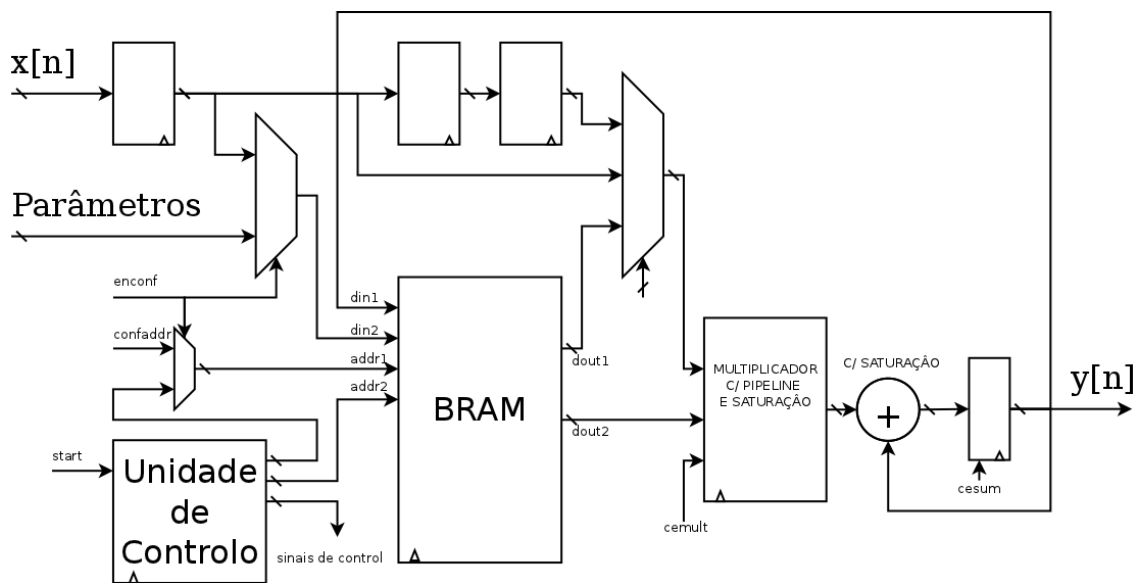


Figura 3.10: Implementação digital dos filtros.

A implementação, como se mostra na figura 3.10, é baseada numa das **BRAMs** e uma estrutura que muitas vezes chamada de Multiply And Accumulate (**MAC**), que consiste num multiplicador e um acumulador que implementa perfeitamente este tipo de filtros. A controlar tudo isto está uma máquina de estados. Todo o caminho de dados está partido numa *pipeline* para aumentar a frequência de relógio e aproveitar o processamento em série de muitos dados (embora não seja necessária porque o sistema tem um *bottleneck* no Microblaze). A **BRAM** serve para guardar todos os parâmetros dos filtros e versões atrasadas das respostas de cada um, que são necessárias para o cálculo da equação às diferenças. Todos os parâmetros podem ser configurados em tempo real usando um registo partilhado com o processador.

A representação binária dos dados é feita com virgula fixa porque, como foi provado em 3.1.4, os filtros atingem um limite bem definido se tiverem largura de banda suficientemente baixa. São usados 17 bits para representação fraccionaria e 16 para representação não fraccionaria. Todos os cálculos são realizados com teste de *overflow* e são saturados nesse caso. Claro que se os filtros atingirem um valor máximo nenhum dos cálculos irá saturar e, por causa disso, a saturação só é usada para efeitos de testes e *debug*.

O módulo foi validado com testes feitos em *verilog* e comparados com resultados obtidos em Matlab, não havendo nenhuma diferença entre eles. O circuito consegue processar cada filtro em quatro ciclos de relógio depois de enchida a *pipeline* que demora quatro ciclos a encher. Isto permite processar cerca de vinte filtros à corrente frequência de amostragem⁴, o que excede as expectativas já que apenas eram necessários oito.

Detectores de Envoltente

Existe um técnica simples para implementar detectores de envoltente com circuitos digitais que consiste na filtragem das frequências altas do valor absoluto de um sinal. O sinal não pode ter componente contínua para o método funcionar, que é o caso na saída dos filtros passa-banda. Segue-se uma pequena demonstração da veracidade deste algoritmo.

Assumindo um sinal $x(t)$ sempre positivo, modelado com frequência ω_0 :

$$y(t) = x(t)\cos(\omega_0 t) \quad , x(t) \geq 0 \quad (3.21)$$

a operação módulo do sinal modelado $|y(t)|$ pode ser vista como:

$$|y(t)| = y(t) \cdot (2sqr(\omega_0 t) - 1) \quad (3.22)$$

em que a função $sqr(t)$ é uma onda quadrada com período 2π e componente contínua de $1/2$. Como se vê na figura 3.11, a multiplicação do sinal modelado $y(t)$ com a onda quadrada da figura resulta no sinal da figura 3.12, que é igual ao valor absoluto do sinal original. Esta simplificação permite analisar o espectro do sinal em valor absoluto realizando a convolução do espectro da onda quadrada e a do sinal modelado.

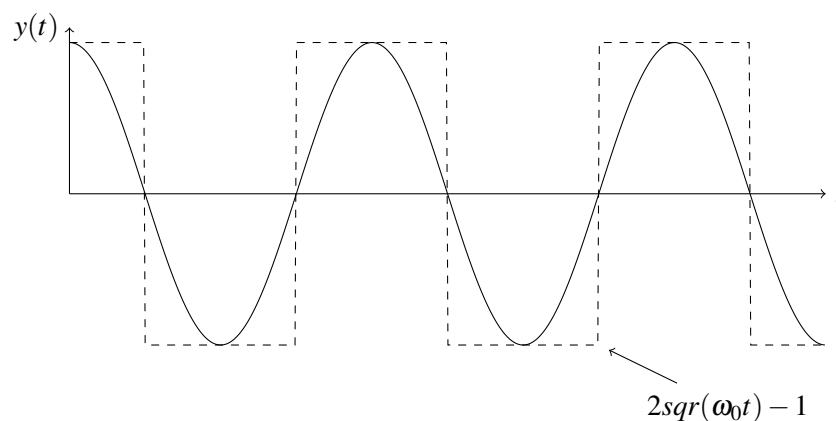


Figura 3.11: Explicação pictórica do detector de envoltente.

⁴561kHz

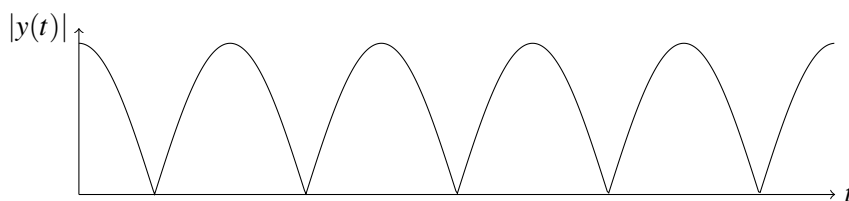


Figura 3.12: Valor absoluto de uma senoide.

A transformada de *Fourier* de uma onda quadrada com frequência ω_0 é:

$$SQR(j\omega) = \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0) \cdot \frac{2\sin(k\pi/2)}{k} \quad (3.23)$$

A transformada de *Fourier* do valor absoluto do sinal modelado $Y_{abs}(j\omega)$ é então:

$$Y_{abs}(j\omega) = 2X(j\omega) * SQR(j\omega) - X(j\omega) \quad (3.24)$$

que resulta em:

$$Y_{abs}(j\omega) = \frac{2}{\pi}X(j\omega) + \frac{1}{\pi} \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} X(j\omega - jk2\omega_0) \left[\frac{\sin(\frac{2k-1}{2}\pi)}{2k-1} + \frac{\sin(\frac{2k+1}{2}\pi)}{2k+1} \right] \quad (3.25)$$

Da análise da equação anterior conclui-se que o espectro do valor absoluto de um sinal modelado é uma repetição, com período $2\omega_0$, do espectro da envolvente do sinal, sendo cada repetição atenuada cada vez mais. A envolvente do sinal modelado pode ser recuperada se for aplicado um filtro com ganho $\pi/2$ que remova todas as frequências a partir de ω_0 . Como não existem filtros perfeitos, as repetições do espectro nunca vão ser totalmente removidas existindo sempre alguma oscilação à frequência $2\omega_0$ na saída do detector de envolvente, que provoca erro no sistema como se de ruído se tratasse. Na tentativa de eliminar a oscilação pode-se começar a filtrar abaixo de ω_0 , no entanto isto provoca um atraso no tempo de subida do sinal, resultando também em erro. Está-se, portanto, na presença de um compromisso entre oscilação no detector de envolvente e rapidez de resposta, e por isso existe um valor de frequência de corte do filtro para o qual é atingido um erro mínimo. Quanto maior for a ordem do filtro passa-baixo menor erro se obtém, mas como a ordem aumenta o número de coeficientes a usar no caso digital, o número de filtros processados para uma determinada frequência de amostragem será menor. Para evitar que a complexidade do sistema aumente demasiado, escolheu-se implementar um filtro *Butterworth* de segunda ordem que apresenta uma queda de -40dB/dec a partir da frequência de corte e apenas usa mais um coeficiente do que os filtros passa-banda. A função de transferência dos filtros passa-baixo é a seguinte:

$$H(s) = \frac{\pi/2}{(s/\omega_0)^2 + \sqrt{2}s/\omega_0 + 1} \quad (3.26)$$

Sem contar com a oscilação provocada pela atenuação incompleta das frequências acima de ω_0 , o erro transitório da saída do filtro passa-baixo em relação à envolvente real é máximo quando existe maior variação na entrada. A maior derivada da envolvente que se irá ter à saída de um filtro passa-banda é obtida derivando a equação 3.8:

$$d_{max} = \frac{K \cdot B}{2} \quad (3.27)$$

onde K , é a amplitude do sinal de entrada dos filtros passa-banda e B é a largura de banda dos mesmos. Sabendo a função de transferência dos filtros (equação 3.26) e a rampa máxima da envolvente, é possível calcular, usando o *teorema do valor final*, o erro a essa rampa, resultando em:

$$\frac{K \cdot B}{\sqrt{2}\omega_{pb}} \quad (3.28)$$

sendo ω_{pb} a frequência de corte do filtro passa-baixo.

A amplitude de oscilação do detector de envolvente alimentado com o sinal do filtro passa-banda com frequência ω_0 pode ser calculada levando apenas em conta os efeitos da frequência $2\omega_0$ depois de aplicado o módulo. Usando as equações 3.26 e 3.25 deduz-se que a amplitude de oscilação à saída dos filtros passa-baixo é:

$$A_{max} = \frac{2}{3} \frac{\omega_{pb}^2}{\sqrt{(2\omega_0)^4 + \omega_{pb}^2}} y_{max} \quad (3.29)$$

onde y_{max} é o valor máximo que a envolvente atinge com uma determinada largura de banda e duração do sinal.

A soma das equações 3.29 e 3.27 é igual ao erro máximo que a saída do detector de envolvente tem em relação à envolvente real. A subtracção dá o erro mínimo.

Minimizando o erro máximo resulta na seguinte expressão para a frequência de corte do filtro passa-baixo:

$$\omega_{pb} \simeq \sqrt[3]{\frac{3(2\omega_0)^2 B}{4\sqrt{2}} \cdot \frac{K}{y_{max}}} \quad (3.30)$$

Minimizando o erro mínimo, a frequência de corte pode ser calculada com a expressão:

$$\omega_{pb} \simeq \sqrt[3]{\frac{3(2\omega_0)^2 B}{\sqrt{8}} \cdot \frac{K}{y_{max}}} \quad (3.31)$$

Usando a expressão para minimizar o erro mínimo resulta em maior oscilação mas menor tempo de resposta. Com a minimização do erro máximo resulta o contrário. Podendo-se agora calcular a frequência de corte de cada detector de envolvente em função de y_{max} (depende da largura de banda dos filtros passa-banda), K (determinado em 3.1.4) e frequência central ω_0 podem-se implementar os detectores de envolvente.

A realização digital dos detectores usa uma arquitetura semelhante à dos filtros, só que para o cálculo do filtro de segunda ordem são precisos mais coeficientes. Além disso, o cálculo de todos os filtros não se baseia só numa única amostra, tendo agora de se guardar versões anteriores das entradas de cada detector na **BRAM** em vez de usar um *shift-register* como nos filtros passa-banda. Os filtros passa-baixo precisam de cinco ciclos de relógio para processar uma amostra e demoram quatro a encher a *pipeline*. O módulo que implementa os filtros passa-banda é atrasado para poder acompanhar os detectores de envolvente.

À semelhança dos filtros passa-banda, todos os parâmetros dos detectores podem ser modificados em tempo real.

Detector de Sinal

O último passo para se detectar o sinal é feito neste bloco, que consiste numa **BRAM** para guardar todos os níveis de detecção e um comparador. Os dados provenientes dos detectores de envolvente são comparados neste módulo e se algum passar o respectivo nível de detecção é sinalizada uma detecção e é guardado na **BRAM** que o canal passou o nível. Se o detector fosse só isto, sempre que o sinal estivesse acima do nível de detecção, eram sinalizadas detecções. Para prevenir tal caso, foi implementado um segundo nível que é usado para monitorizar se o sinal passou abaixo de um limite, só quando isto acontecer é que são sinalizadas novas detecções no canal. É preciso, no entanto garantir que, para uma determinada potência de ruído, o sinal caía abaixo do segundo nível na ausência de sinal.

Este módulo também pode ser configurado em tempo real assim como os filtros e os detectores de envolvente. Existe um descodificador de endereços que desvia os valores das configurações para os módulos certos, tornando possível toda a configuração do sistema com os mesmos registos partilhados.

Relógio Interno

O **MARES** necessita de uma base temporal precisa para a detecção e envio de sinais. Para isso, foi implementado um relógio interno à custa de um contador que conta à mesma frequência que a frequência de amostragem. Os tempos deste relógio interno são depois usados pelo **MARES** para calcular a posição do submarino que, como apresentam uma resolução limitada (561kHz), podem apresentar erros de no máximo $2\mu\text{s}$. Este erro, à velocidade do som no oceano ($\approx 2500\text{m/s}$), não representa mais do que 5mm, o que é desprezável face à precisão obtida por todo o esquema de detecção. O contador tem bits suficientes para conseguir contar mais de 16 segundos (mais de 23 bits), isto porque, segundo o pessoal que trabalha com o submarino, o tempo máximo entre eventos nunca ultrapassa os 16 segundos. Embora só fossem precisos 24 bits foram usados cerca de 32 porque, para além de não implicar nenhum problema na sua realização em **FPGA**, ajuda na fase de testes para que se mantenha uma base temporal fácil de observar. Associada ao contador, que por si só é de fácil implementação, foi criada uma fila First In First Out (**FIFO**) que guarda o canal e o valor do contador sempre que há uma detecção. Esta fila foi realizada à custa de uma **BRAM** e

dois registos que apontam para o início e o fim. Sempre que o processador lê do registo de leitura da fila, o registo que aponta para o início, avança uma posição. Sempre que existe uma detecção o registo que aponta para o fim da fila é que avança uma posição. Foi necessária a implementação desta fila para que o processador não perca recepções.

Memória de Amostras

O sistema tem a capacidade de armazenar 8192 amostras do [A/D](#), filtros passa-banda ou dos detectores de envolvente. Esta funcionalidade é controlada através de registos partilhados e uma Random Access Memory ([RAM](#)) partilhada. Com este sistema pode-se medir a potência do ruído existente nos filtros ou em banda base. Existe também a opção de parar a amostragem quando é recebida uma detecção, para que se veja a subida do sinal até atingir o nível de detecção. Estas técnicas foram muito úteis no desenvolvimento do projecto.

Emissão

Não representado na figura, o módulo de emissão serve para pedir um envio ao sistema de geração de sinais no [MARES](#). Este foi feito em hardware para que se possa medir o tempo de envio sem latência, como é feito com as recepções. O módulo é muito simples e apenas liga o valor do canal que tem num registo partilhado aos pinos de saída da [FPGA](#) e activa um bit quando quer enviar um sinal. Como o sistema de emissão do [MARES](#) funciona a uma frequência mais baixa do que o sistema a ser desenvolvido, realizou-se um contador para que o bit esteja activo tempo suficiente.

O processador aqui implementado também vai servir para substituir o sistema acústico das bóias que é igual ao do submarino. Para isso, é preciso que certas frequências sejam respondidas automaticamente com o mínimo de latência possível. Foi criado para isso um módulo, que pode ser configurado, onde são registadas os pares de frequências a ser respondidos e envia um sinal caso seja detectada um sinal no canal correspondente. Para isto apenas foi necessário uma [BRAM](#) e alguma lógica.

Também existe uma fila, no caso da emissão, que regista o valor do contador falado em [3.4.2](#) e o número do canal quando existe um envio criado automaticamente ou pelo processador.

3.5 Software

Foram implementados dois programas para que o sistema funcione. Ambos usam capacidades do sistema operativo para guardar configurações na flash servindo-se de um sistema de ficheiros em memória e de um programa, já pertencente à distribuição, que sincroniza os ficheiros em memória com os da flash. Um dos programas estará sempre a correr enquanto o sistema está a funcionar. A sua função é monitorizar as filas [FIFO](#) de envio e recepção enquanto implementa o protocolo de comunicação com o [MARES](#) por porta série.

A Suzaku não permitia, com as configurações originais, o uso exclusivo da porta série. Todas as mensagens do bootloader, kernel e consola eram enviadas automaticamente para lá, por isso para a reservar para uso exclusivo do programa, foram necessárias alterações tanto aos bootloaders como ao kernel. Os bootloaders tiveram de ser mudados para que, mediante um *jumper*, arrancassem o kernel com a consola desviada para "/dev/null", e não imprimissem nada por porta série. Infelizmente isto não resolveu o problema porque o kernel não reconhecia a linha de comandos. Investigou-se e descobriu-se que a zona de memória onde a linha de comandos era guardada era preenchida com zeros durante a execução do kernel e antes da leitura dos comandos, esta área de memória é chamada normalmente de Block Started by Symbol (**BSS**). Para resolver este *BUG*, foi mudado o ficheiro "arch/platform/suzaku/machine.c", para que, antes de inicializar a zona de memória **BSS**, a linha de comandos seja guardada temporariamente na stack e copiada de volta.

O protocolo implementado pelo programa de que se falava, funciona sobre RS-232, é simples, implementado ao byte e codificado para que tenha pouca latência. Existe um byte de sincronismo seguido por um byte de comando e argumentos variáveis. Nem todos os comandos do protocolo foram implementados porque não fazem sentido para o sistema, os que foram são os seguintes:

Actualizar o vector de respostas Este comando é enviado para o sistema para actualizar o par de frequências de resposta automática e guarda os dados na flash.

Actualizar resposta a um canal Realiza exactamente o mesmo que o anterior mas apenas para um par de frequências.

Ler vector de respostas Os dados das respostas automáticas são enviados para o processador central do **MARES**.

Ler resposta a um canal Mesmo do que o anterior mas apenas para um par de frequências.

Ordenar envio O sistema, como resposta a este comando, envia um sinal no canal pedido. Este comando é usado normalmente durante a navegação.

Alterar um nível de detecção Como o nome indica, o comando pede ao sistema para alterar o nível de detecção. Este comando é usado antes da missão quando é medido o nível de ruído e calibrados os níveis de detecção. Os níveis não são guardados em flash, apenas são mudados os parâmetros no hardware dedicado. O comando de leitura de um nível de detecção também está implementado.

Gravar nível de detecção Quando os níveis estão calibrados, este comando é usado para gravar um nível em flash.

Reset Este comando é enviado pelo sistema ao processador do **MARES** no arranque, logo que fique operacional.

Erro Enviado do sistema para o **MARES** para sinalizar um erro de comunicação.

Envio ou recepção Quando o software detecta algum envio ou detecção nas filas correspondentes envia um comando destes com o tempo exacto e o canal do evento.

Os comandos ligados com a medição do estado da bateria, envio de dados do [GPS](#), e controlo do rádio não foram implementados.

O segundo programa foi feito para servir como plataforma de configuração do sistema. Permite configurar os parâmetros do filtro em função da frequência central e largura de banda, o número de filtros a serem processados, níveis de decisão e respostas automáticas. Este é chamado no arranque para restaurar as configurações do hardware usando os dados guardados em flash.

Capítulo 4

Implementação e Validação Experimental

Neste capítulo são apresentados alguns resultados da implementação do sistema, assim como a sua validação. Para isso, foram realizados vários testes, que não seriam possíveis sem a ajuda do sistema operativo embebido. É feita aqui uma descrição de cada um deles assim como os resultados obtidos.

4.1 Implementação

A utilização da **FPGA** na plataforma de desenvolvimento está descrita na tabela 4.1. De notar que a tabela inclui *hardware* que não seria usado na implementação final do projecto. A frequência de relógio usada tanto para o Microblaze como para o *hardware* dedicado é de 51MHz e, ao todo, a plataforma consome cerca de 1.6W.

Utilização lógica	Usado	Disponível	Utilização
Número de Slices	5589	8672	64%
Número de BRAMs	28	28	100%
Número de Multiplicadores	11	28	39%

Tabela 4.1: Sumario da utilização da Spartan3E-1200.

O design, sem contar com o Microblaze e todos os periféricos de controlo de *hardware* não necessário para o projecto, foi sintetizado para uma **FPGA** Spartan3E-500 (tabela 4.2), que é usada em plataformas baseadas em **FPGA**, mais baratas e menos consumidoras do que a de desenvolvimento.

Da análise da tabela 4.2 conclui-se que o projecto pode ser implementado numa **FPGA** bem mais pequena sem nenhum esforço. Devido à ainda pouca utilização da **FPGA**, o Microblaze poderá ser acrescentado ao design final para realizar o protocolo de comunicação e calcular os parâmetros dos filtros em tempo real usando aritmética de virgula flutuante, caso seja necessário.

Utilização lógica	Usado	Disponível	Utilização
Número de Slices	826	4656	17%
Número de BRAMs	6	20	30%
Número de Multiplicadores	8	20	40%

Tabela 4.2: Sumario da utilização da Spartan3E-500.

4.2 Validação Experimental

4.2.1 Curto Alcance

Para validar o correcto funcionamento de todo o hardware de detecção (detectores de envolvente, filtros, níveis de detecção), ligaram-se duas placas de emissão e recepção do MARES a dois transdutores e ligaram-se as interfaces acústicas. No sistema foi criado um programa de testes que enviava um sinal, guardava o tempo de envio, e esperava por uma resposta de qualquer canal, calculando, com o tempo de propagação, a distância entre os dois transdutores. Os filtros foram calibrados com uma largura de banda de 100Hz e frequência central dos 20 aos 27kHz, com espaçamento de 1kHz, e níveis de detecção coerentes com o nível de ruído existente. Esperava-se receber resposta no mesmo canal em que houve o envio.

Ao executar o primeiro teste com os transdutores no ar (mudando a velocidade do som para mais ou menos 300m/s), com um espaçamento entre os transdutores de mais ou menos 1m, houve ram detecções no canal esperado. Os tempos não coincidiam com o esperado porque as detecções aconteciam sempre 1ms depois do envio do sinal (que era sensivelmente o tempo de resposta dos filtros), independentemente da distância entre os transdutores. Este fenómeno, veio-se depois de descobrir, era resultado da interferência electromagnética que as placas de envio e recepção do MARES induziam nelas próprias.

Executou-se novo teste, agora com as placas de emissão e recepção a uma distância segura, mas desta vez não houve nenhuma detecção. Concluindo-se que os transdutores não tinham impedância acústica suficiente para que a transmissão se sucedesse. De notar que o esquema montado para este teste não possuía amplificação do sinal. Medido no osciloscópio, o sinal recebido atingia alguns μV de pico estando completamente distorcido.

O teste seguinte foi feito com os transdutores mergulhados em água, usando para isso um balde. Foram realizadas detecções neste caso e os tempos medidos mantinham coerência, embora não se pudessem fazer testes a mais de meio metro. Deste teste resultaram, algumas vezes, detecções noutras canais que não o enviado. Isto não poderia acontecer se apenas fosse recebido um sinal com uma única frequência porque invalidaria o sistema. Para resolver o caso, tiraram-se amostras do sinal (figura 4.1), e dos detectores de envolvente dos canais recebidos (figuras 4.2 e 4.3), até à passagem, pelo nível de detecção, do canal onde foi recebida uma detecção errada.

O que acontece é que ao enviar um sinal nos 21kHz ,cuja recepção é mostrada na figura 4.1, a envolvente que sai do filtro adaptado à frequência de envio (21kHz) é a que se mostra na figura 4.2.

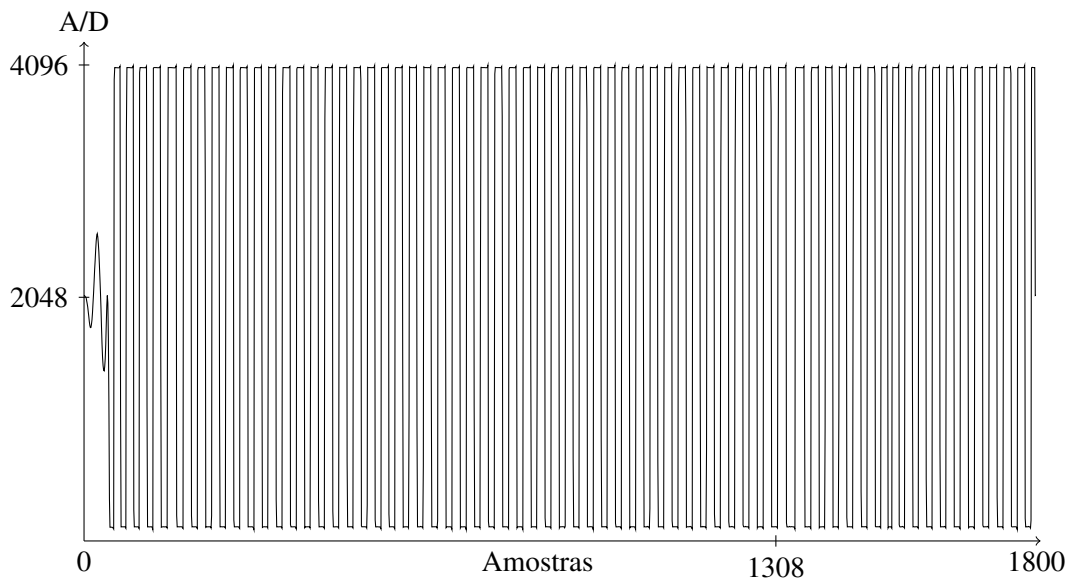


Figura 4.1: Amostras de um sinal enviado nos 21kHz.

Esta envolvente mostra que por volta da amostra 1300 existe uma mudança na frequência recebida e, como consequência, a saída do filtro cai.

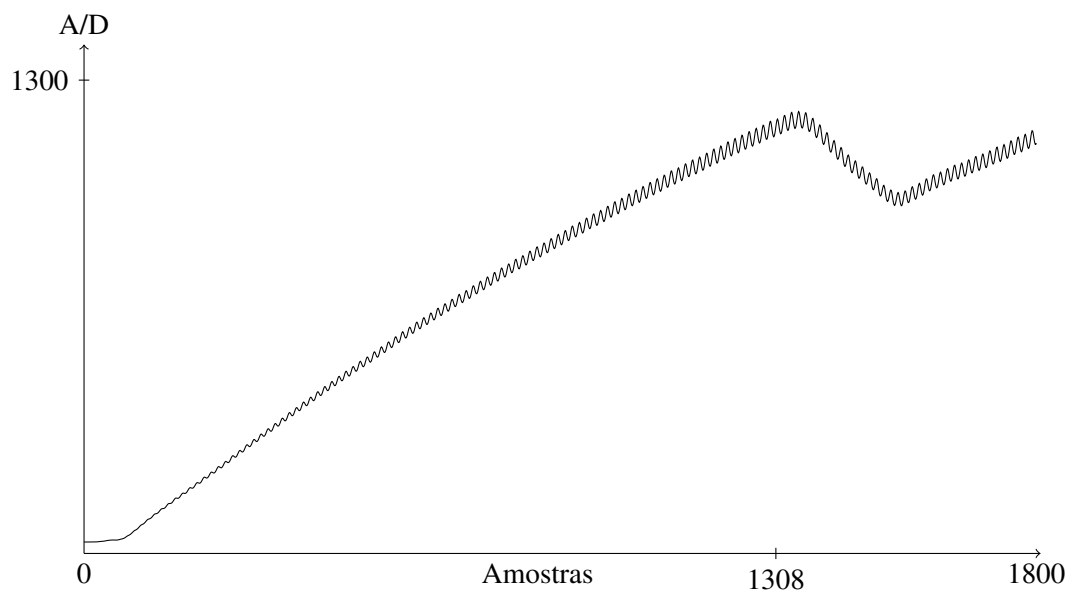


Figura 4.2: Amostras do detector de envolvente de 21kHz.

Em contra partida, no filtro dos 22kHz (figura 4.3), embora inicialmente a saída do filtro seja oscilatória mostrando uma resposta desadaptada, a partir da amostra 1300 o filtro começa a responder exponencialmente, mostrando que a entrada está agora adaptada a ele.

A frequência enviada pela placa de emissão do **MARES** não muda de frequência porque a sua

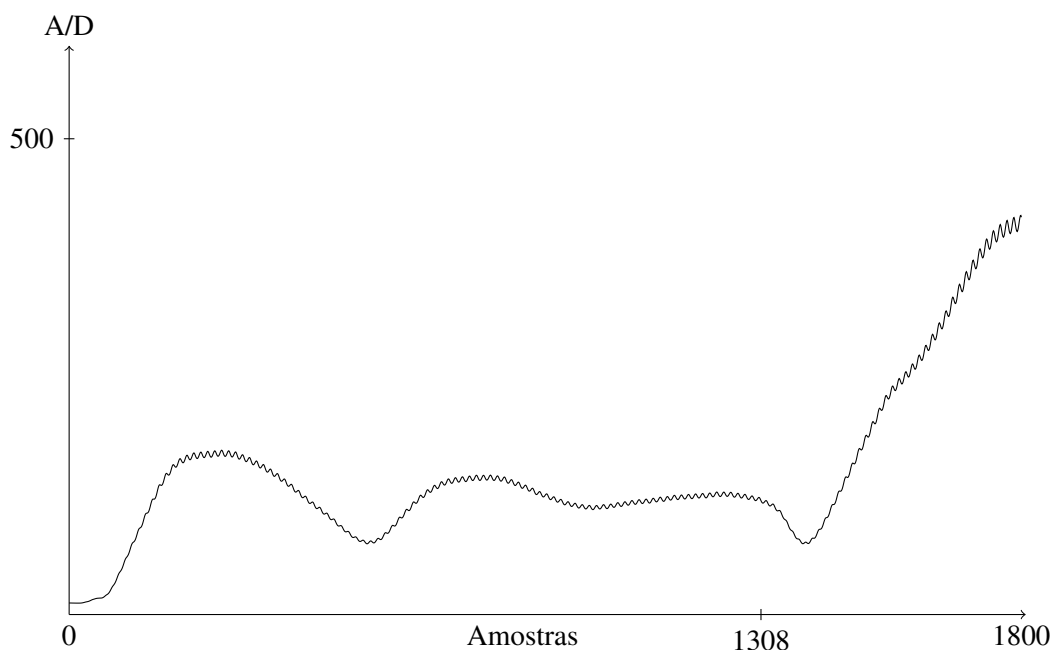


Figura 4.3: Amostras do detector de envoltório de 22kHz.

saída foi medida. O que pode estar a acontecer é que, como existem muitas reflexões no balde, a saturação da mistura de todos os sinais (fenómeno não linear) provoca mudanças na frequência.

4.2.2 Médio Alcance

No tanque da Faculdade de Engenharia da Universidade do Porto foi realizado o mesmo teste realizado na secção anterior, mas desta vez com os transdutores afastados alguns metros. A tabela 4.3 mostra as distâncias medidas de segundo a segundo. O que se quer medir aqui é a repetibilidade do sistema, ou seja, a precisão sem contar com efeito de *Doppler* e decaimento do sinal sobrando apenas ruído. É de notar que o efeito visível no teste anterior, onde se verificava uma mudança na frequência do sinal, não se verificou aqui, provavelmente porque as reflexões dos sinais, embora ainda existam, chegam ao transdutor em intervalos de tempo maiores diminuindo a mistura entre sinais e reflexões.

A tabela 4.3 mostra também a repetibilidade obtida em cada frequência. Existem diferenças significativas entre as medições de diferentes frequências, embora todos os filtros tenham a mesma largura de banda. Isto deve-se ao desvio entre a frequência de envio e a frequência central do filtro que podiam ser corrigidas, mas com o efeito de *Doppler* não se iria obter melhoria. O erro máximo entre todas as medidas é de 0.57metros.

4.2.3 Integração com o MARES

O último teste foi uma integração com o MARES. O sistema foi configurado com os mesmos valores do que nos testes anteriores mas agora controlado pela ligação RS-232 do processador

	Distâncias em metros					
	23kHz	24kHz	25kHz	26kHz	20kHz	21kHz
	3.016	3.768	2.979	2.870	2.824	2.666
	3.003	3.781	2.963	2.795	2.805	2.688
	3.024	3.797	2.963	2.776	2.830	2.674
	3.008	3.784	2.977	2.840	2.813	2.661
	3.027	3.776	2.963	2.988	2.835	2.680
	3.011	3.787	2.977	2.974	2.821	2.666
	3.030	3.776	2.960	2.872	2.808	2.672
	3.017	3.763	2.979	2.803	2.829	2.690
	3.003	3.779	2.960	2.854	2.811	2.677
	3.025	3.765	2.979	2.821	2.835	2.666
	3.009	3.752	2.960	2.835	2.821	2.685
	3.027	3.768	2.976	2.886	2.805	2.669
	3.014	3.755	2.960	2.843	2.827	2.688
	3.030	3.771	2.976	2.813	2.811	2.674
	3.019	3.757	2.958	2.862	2.832	2.664
Repetibilidade(m)	±0.015	±0.023	±0.011	±0.106	±0.015	±0.015

Tabela 4.3: Repetibilidade do Sistema.

central do [MARES](#). Houve um pequeno problema ao fazer o teste, algumas bóias não estavam a emitir com potência suficiente para os sinais serem detectados com nenhuma amplificação, como era com os testes anteriores. Por isso, só depois de aplicado o mesmo circuito de amplificação que está no [MARES](#) é que foi realizado o teste. O sistema comunicava correctamente com o processador central, enviando correctamente, o canal e o tempo das detecções. Os comandos de mudança de nível de detecção não estavam a ser implementados correctamente pelo que terá de ser feita uma nova revisão ao software.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Satisfação dos Objectivos

Tendo em conta os objectivos do trabalho, pode-se dizer que se atingiu um nível de satisfação elevado. Ficou provado que a implementação digital do sistema acústico do **MARES** é possível e com ele todas as vantagens de um sistema digital. Para além da eliminação dos problemas apresentados pelos circuitos analógicos, foi possível acrescentar mais flexibilidade na configuração dos filtros e em mais quantidade. O esquema de detecção por si só não permite que hajam melhorias significativas na precisão mesmo sendo implementado digitalmente, embora estejam abertas agora novas portas para novos sistemas de detecção que podem melhorar muito a precisão e a cadência das medições.

5.2 Trabalho Futuro

Futuramente, o projecto para ser instalado definitivamente no **MARES** ou outro **AUV**, terá de ser transportado para uma plataforma mais simples do que a Suzaku (por causa do seu custo), como por exemplo uma GODIL da OHO-Elektronik [15], e conseqüentemente para uma **FPGA** de menor capacidade. O sistema operativo também não precisará de ser usado, pelo que o software também terá de ser modificado. A nova plataforma poderá depois ser instalada no **AUV** e nas bóias.

Com todo o sistema instalado vai se poder explorar novas técnicas de detecção para permitir mais precisão e mais rapidez entre aquisições de posição. As novas técnicas a ser exploradas são por exemplo, detectores coerentes ou não coerentes [5], que permitem melhor precisão embora não ofereçam melhoria entre tempos de aquisição. As técnicas mais interessantes, porém, são técnicas de espalhamento espectral como as de *Chirp Spread Spectrum* [16], *Direct Sequence Spread Spectrum* [17] ou *Frequency-Hopping* [18] que, para além de melhorarem imenso a precisão do

sistema, diminuem muito os tempos de aquisição por descorrelacionarem as reflexões. Estas técnicas podem ser realizadas em **FPGA** actualmente sem dificuldade, tendo, algumas delas, já sido implementadas [19].

Outro caminho que pode surgir, como continuação deste trabalho, é uma implementação de comunicações acústicas subaquáticas. Isto alargaria o leque de operação dos **AUV** permitindo que se possa controla-los remotamente sem fios alargando a área de operação e permitindo a implementação de outras técnicas de navegação.

Referências

- [1] National Science Foundation. United states antarctic program, April 1996. <http://www.nsf.gov/pubs/1996/nstc96rp/sb10.htm>.
- [2] Nuno Cruz, Luís Madureira, Aníbal Matos, e Fernando Lobo Pereira. A versatile acoustic beacon for navigation and remote tracking of multiple underwater vehicles. *MTS/IEEE Oceans 2001*, páginas 1829–1834, Novembro 2001.
- [3] Azizul H. Quazi e William L. Konrad. Underwater acoustic communications. *ieee communications magazine*, páginas 24–30, Novembro 1982.
- [4] Alex Bahr. *Cooperative Localization for Autonomous Underwater Vehicles*. Tese de doutoramento, Massachusetts Institute of Technology, 2009.
- [5] Donald Christiansen e Charles Alexander. *Standard Handbook of Electronic Engineering*. McGraw-Hill, quinta edição, 2004.
- [6] W.D. Wilson. Equation for the speed of sound in sea water. *J. Acoust. Soc. Amer.*, páginas 641–644, 1960.
- [7] Hwee-Pink Tan, Roe Diamant, Winston K.G. Seah, e Marc Waldmeyer. A survey of techniques and challenges in underwater localization. 2010.
- [8] Brian Bingham, David Mindell, Thomax Wilcox, e Andy Bowen. Integrating precision relative positioning into jason/medea rovs operations. *MTS Journal*, páginas 87–96, Janeiro 2006.
- [9] Nuno Cruz e Aníbal Matos. The mares auv, a modular autonomous robot for environment sampling. *MTS/IEEE Oceans 2008*, páginas 1–6, Setembro 2008.
- [10] Simon Haykin. *Communication Systems*. John Wiley & Sons, Fourth edição, 2001.
- [11] Xilinx. Spartan-3e fpga family: Data sheet. Relatório técnico, Xilinx, Agosto 2009.
- [12] Atmark-Techno. Suzaku-s sz-130-u00 hardware manual. Relatório técnico, Atmark-Techno, Agosto 2006.
- [13] Xilinx. Microblaze processor reference guide. Relatório técnico, Xilinx, Março 2011.
- [14] Analog Devices. Ad7912/ad7922. Relatório técnico, Analog Devices, Abril 2004.
- [15] M.Randelzhofer. Godil user manual. Relatório técnico, OHO-Elektronik, 2009.
- [16] Jianguo Huang, Chengbing He, e Qunfei Zhang. M-ary chirp spread spectrum modulation for underwater acoustic communication. *TENCOM IEEE region 10*, páginas 1–4, Novembro 2005.

- [17] E.M.Sozer, J.G.Proakis, M.Stojanovic, M.Hatch, J.A.Rice, e A.Benson. Direct sequence spread spectrum based modem for under water acoustic communication and channel measurements. *MTS/IEEE Oceans 1999*, páginas 228–233, Setembro 1999.
- [18] Feng Tong, Xiaomei Xu, e Tianzeng Xu. Frequency hopping underwater data communication system's synchronization processing. *IEEE 2002 International Conference on Communications*, páginas 277–281, Julho 2002.
- [19] Sérgio Rui dos Santos Barbosa Oliveira da Silva. *Interferometric Synthetic Aperture Sonar System supported by Satellite*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 2009.