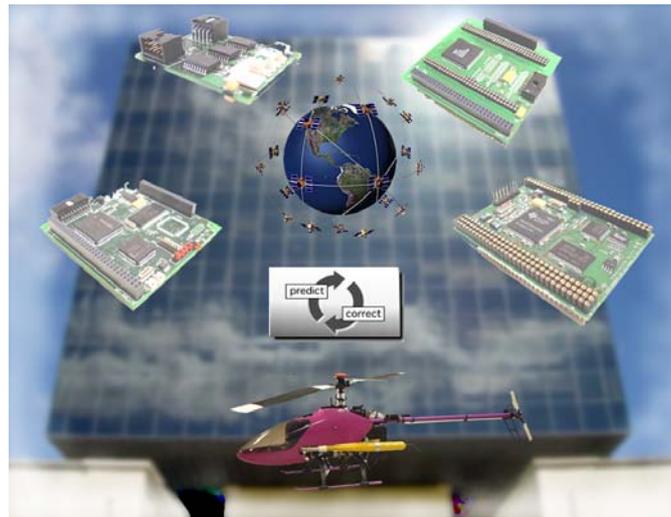




Universidade Técnica de Lisboa

Instituto Superior Técnico



ARQUITECTURAS PARA SISTEMAS DE NAVEGAÇÃO INS/GPS COM APLICAÇÃO A UM HELICÓPTERO AUTÓNOMO

Bruno Carneira n.º 46492, AE de Controlo e Robótica
Filipe Baltazar n.º 46525, AE de Controlo e Robótica

LICENCIATURA EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
Relatório de Trabalho Final de Curso
012/2002/L

Prof. Orientador: Carlos Jorge Ferreira Silvestre
Prof. Acompanhante: Paulo Jorge Oliveira

Janeiro de 2004

Agradecimentos

O presente relatório sumariza um trabalho que se estendeu ao longo de um ano e meio, tendo nesse período, contado com a ajuda e amizade de diversas pessoas às quais gostaríamos de agradecer. As primeiras palavras dirigem-se para os Professores Carlos Silvestre e Paulo Oliveira por todo o apoio e incentivo prestado. Agradece-se também aos engenheiros João Alves, Luís Sebastião, Manuel Rufino e Guilherme Libório pela disponibilidade e ajuda prestadas.

Gostaríamos também de agradecer ao amigo e colega de trabalho, Pedro Dias, pela partilha dos bons momentos e pelo apoio e amizade nos mais delicados. Estendemos o presente agradecimento à “Tropa” que proporcionou um excelente ano de trabalho com discussões construtivas e muitos momentos de boa disposição.

Uma palavra especial para os amigos José Vasconcelos e Carlos Alfaro, cujo apoio em alturas mais críticas do trabalho foi bastante importante.

Ao Eng.º João Serralha pelo seu profissionalismo, disponibilidade, ajuda nos momentos de dificuldade e, acima de tudo, por uma constante boa disposição, dirigimos também uma palavra de apreço.

Não podíamos deixar de agradecer aos nossos pais e familiares por terem respeitado o nosso trabalho, tolerado ausências forçadas em momentos importantes e telefonemas esquecidos, para eles, muito obrigado!

Pelo constante apoio, incentivo e generosidade ao longo deste ano e meio um muito obrigado à Patrícia Mota e Sandra Duarte.

“A teoria e a prática estão mais afastadas que aquilo que se afirma em teoria.”

Alguém que já experimentou ambas

Resumo

O trabalho realizado aborda a integração e teste de arquitecturas de *hardware* para sistemas de navegação INS/GPS com aplicação a um helicóptero autónomo. Para tal foi desenvolvido e testado um simulador de *Hardware-In-The-Loop*. Este tipo de simulador é uma ferramenta com um potencial de utilização bastante elevado, pois permite encontrar e corrigir erros no desenvolvimento do projecto que de outra forma só seriam detectados num teste real, o que poderia levar a custos elevados ou situações de risco. Esta vantagem consegue-se à custa de testar o *software* desenvolvido no *hardware* a utilizar a bordo do veículo em conjunto com a simulação da sua dinâmica e da interacção do mesmo com o ambiente como se de uma aplicação real se tratasse. Para a elaboração do simulador HITL fazer sentido, utilizaram-se placas desenvolvidas no DSOR/ISR (*Dynamical Systems and Ocean Robotics/Instituto Sistemas e Robótica*) a partir das quais se formou uma arquitectura composta por uma parte de aquisição de sinal e outra de processamento digital de sinal.

Apesar da aplicação a que se destina, não são feitas quaisquer restrições ou particularizações quanto ao tipo de veículo, sendo os resultados obtidos válidos para outro tipo de veículo autónomo que não só o helicóptero. A arquitectura de aquisição de sinal é utilizada numa aplicação com requisitos de tempo real e apresenta um desempenho elevado, tendo-se conseguido resoluções efectivas de 19 *bit*. A integração da arquitectura de processamento digital acrescentou capacidade computacional a um sistema que necessita de implementar algoritmos computacionalmente exigentes. Nesta arquitectura de processamento digital foram implementados dois sistemas de navegação: um baseado em filtros de Kalman complementares e outro no filtro de Kalman estendido. O primeiro recorre directamente às medidas dos vários sensores considerando para tal as larguras de banda associadas às medidas disponíveis. O segundo integra um algoritmo de navegação inercial e as medidas de sensores auxiliares, projectando-se um filtro de Kalman estendido segundo uma topologia *direct feedback*.

A complexidade do filtro de Kalman estendido fez com que inicialmente este apresentasse um tempo de processamento superior a 70 ms, o que era incomportável para a aplicação pretendida. Por esta razão procedeu-se a uma optimização do algoritmo, tendo-se alcançado um tempo de processamento de 5,5 ms. Esta optimização permitiu a implementação de algoritmos eficientes, capazes de preencher requisitos de tempo real.

Dos resultados obtidos na simulação dos dois algoritmos no *Hardware-In-The-Loop*, verificou-se que ambos apresentam um bom desempenho, sendo o erro máximo de estimação da posição para o filtro complementar cerca de 3 m e para o estendido cerca de 2 m. Relativamente ao erro máximo de estimação de atitude, este é aproximadamente 0.15 rad para o filtro complementar e 10^{-2} rad para o estendido. Estes resultados são considerados bastante satisfatórios, pois conseguiu-se, com esta arquitectura, um desempenho bastante próximo ao obtido em ambiente de simulação numérica.

Do trabalho desenvolvido concluiu-se que o *Hardware-In-The-Loop* é uma ferramenta eficaz e de grande potencial, tendo possibilitado a verificação de que, tanto a arquitectura de *hardware* como os algoritmos apresentam desempenhos que permitem ter expectativas positivas relativamente à sua utilização numa aplicação real.

Palavras Chave: Sistemas de Navegação; INS; INS/GPS; Filtros de Kalman complementares; Filtro de Kalman estendido; Arquitecturas de *hardware* para navegação de veículos autónomos; *Hardware-In-The-Loop*; Arquitectura de Aquisição de Sinal

Índice

1	INTRODUÇÃO	1
2	ARQUITECTURA DE <i>HARDWARE</i> PARA SISTEMAS DE NAVEGAÇÃO	4
2.1	ENQUADRAMENTO DO PROJECTO E TRABALHO ANTERIOR	4
2.2	CONTRIBUIÇÕES DO PROJECTO	4
3	INTEGRAÇÃO DE SISTEMAS	9
3.1	ARQUITECTURA DE AQUISIÇÃO DE SINAIS DE 24 <i>BIT</i> (AD24B3C)	9
3.1.1	<i>Definição da Arquitectura</i>	9
3.1.2	<i>Definição da Placa AD24B3C</i>	10
3.1.3	<i>Metodologia de Interface de Tempo-Real</i>	12
3.2	PLACA DSPIF: INTERFACE ENTRE MC-XAS3 E DSP (<i>D.MODULE.VC33</i>)	16
3.2.1	<i>Mapas de Memória e Metodologia de Interface</i>	17
3.2.2	<i>Acesso de Leitura à Dual-Port RAM</i>	18
3.2.3	<i>Acesso de Escrita à Dual-Port RAM</i>	19
3.2.4	<i>Interrupções</i>	20
3.2.5	<i>Controlo do pino nRESIN do D.Module.VC33</i>	21
3.2.6	<i>Software de Interação entre a placa MC-XAS3, a DSPIf e D.Module.VC33</i>	21
4	SISTEMA DE NAVEGAÇÃO COM RECURSO A FILTROS DE KALMAN	23
4.1	FILTRO DE KALMAN COMPLEMENTAR	23
4.2	FILTRO DE KALMAN ESTENDIDO	24
4.2.1	<i>Modelação de uma medida do vector de rotação do erro de atitude</i>	26
4.2.1.1	<i>Modelo de Estado do filtro passa-banda</i>	27
4.2.1.2	<i>Discretização</i>	28
4.2.2	<i>Modelo Final</i>	29
5	IMPLEMENTAÇÃO DOS SISTEMAS DE NAVEGAÇÃO NA ARQUITECTURA DE <i>HARDWARE</i>	32
5.1	QUESTÕES DE IMPLEMENTAÇÃO DO FILTRO DE KALMAN ESTENDIDO	32
5.1.1	<i>Factorização UD</i>	32
5.1.2	<i>Implementação em Tempo-Real e Complexidade do Algoritmo</i>	35
5.2	VALIDAÇÃO DOS SISTEMAS DE NAVEGAÇÃO NO COMPUTADOR DE BORDO	38
5.2.1	<i>Sistema de navegação com recurso a filtros de Kalman complementares (CKF)</i>	40
5.2.2	<i>Sistema de navegação com recurso a filtros de Kalman estendidos (EKF)</i>	41
5.3	<i>HARDWARE-IN-THE-LOOP (HITL)</i>	43
5.3.1	<i>Validação do Loop de navegação, Fluxo de Dados e Transduções</i>	44
5.3.2	<i>Sistema de navegação com recurso a filtros de Kalman complementares</i>	46
5.3.3	<i>Sistema de navegação com recurso ao filtro de Kalman estendido</i>	47
6	CONCLUSÕES	50
APÊNDICES		52
A	REPRESENTAÇÃO DA ORIENTAÇÃO E REFERENCIAIS DE NAVEGAÇÃO	52
A.1	REPRESENTAÇÃO DA ORIENTAÇÃO	52
A.1.1	<i>Matriz de Rotação</i>	52
A.1.2	<i>Ângulos de Euler</i>	53
A.1.3	<i>Vector de Rotação</i>	55
A.1.4	<i>Quaterniões</i>	55
A.2	DEFINIÇÃO DOS REFERENCIAIS	57
B	SENSORES	60

B.1	SENSORES INTEGRADOS NO VEÍCULO	60
B.1.1	<i>Acelerómetro</i>	60
B.1.2	<i>Giroscópio</i>	60
B.1.3	<i>Magnetómetro</i>	61
B.1.3.1	Descrição e Modelação das Fontes de Erro	64
B.1.3.2	Modelo do erro das medidas do Magnetómetro	65
B.1.3.3	Algoritmos de calibração de uma magnetómetro	70
B.2	SENSORES AUXILIARES	75
B.2.1	<i>GPS</i>	75
B.2.2	<i>Estimador de Atitude Magneto-Pendular</i>	76
C	SISTEMA DE NAVEGAÇÃO COM RECURSO A FILTROS DE KALMAN COMPLEMENTARES	80
C.1	FILTRO COMPLEMENTAR DE ATITUDE	80
C.2	FILTRO COMPLEMENTAR DE POSIÇÃO	82
C.3	SISTEMA DE NAVEGAÇÃO GLOBAL	84
D	SISTEMA DE NAVEGAÇÃO INERCIAL	85
D.1	ALGORITMO DE ATITUDE	85
D.2	ALGORITMO DE VELOCIDADE	86
D.3	ALGORITMO DE POSIÇÃO	87
E	MODELO DO VEÍCULO	88
E.1	DESCRIÇÃO FÍSICA DO VEÍCULO	88
E.2	EQUAÇÕES DA DINÂMICA	89
E.3	INFLUÊNCIA DAS FORÇAS E DOS BINÁRIOS	90
F	DESENVOLVIMENTO EM <i>MATLAB/SIMULINK</i> DO MODELO DO VEÍCULO, SENSORES E SISTEMA DE NAVEGAÇÃO	93
F.1	MODELO DO VEÍCULO	94
F.2	MODELO DOS SENSORES	95
F.3	MODELO GERAL	96
G	ARQUITECTURA DE AQUISIÇÃO DE SINAL (PLACAS AD24B3C)	97
G.1	CONVERSOR ADS1210	97
G.2	ESCORREGAMENTO DAS FREQUÊNCIAS DE RELÓGIO DOS CRISTAIS	98
G.3	LIGAÇÕES ENTRE AS PLACAS AD24B3C	98
G.4	ESQUEMAS ELÉCTRICOS DA PLACA AD24B3C	100
G.5	CIRCUITOS LÓGICOS IMPLEMENTADOS PELA PAL	101
G.6	CIRCUITO PROGRAMÁVEL (<i>XILINX XCR22V10</i>)	103
G.7	DESCRIÇÃO DO <i>DRIVER</i> ENTRE A PLACA MC-XAS3 E AS PLACAS AD24B3C	104
G.7.1	<i>Descrição da Função de Abertura do Driver</i>	104
G.7.2	<i>Descrição da Função de Leitura dos Conversores</i>	105
G.8	CARACTERIZAÇÃO DO DESEMPENHO DA PLACA ADC24B3C	107
G.9	REGISTOS DO CONVERSOR	109
G.10	PROGRAMA DA PAL DA PLACA ADS24B3C	109
H	PLACA DSPIF	111
H.1	ESQUEMA ELÉCTRICO DA PLACA DSPIF	112
H.2	DESENHO DO PCB DA PLACA DSPIF	113
H.3	MAPA DE MEMÓRIA	114
H.4	CIRCUITOS LÓGICOS SIMBÓLICOS DE ACESSO À DSPIF	115
H.5	DIAGRAMAS TEMPORAIS DOS CIRCUITOS LÓGICOS	117
H.6	PROGRAMA DA PAL DA PLACA DSPIF	117
I	CONVERSOR DIGITAL-ANALÓGICO	123
J	PLACA MC-XAS3	124
K	D.SIGNT.D.MODULE.VC33	126

K.1	CARACTERÍSTICAS DA PLACA <i>D.MODULE.VC33</i>	126
K.2	TEXAS INSTRUMENTS TMS320VC33 DSP	127
BIBLIOGRAFIA		128
REFERÊNCIAS		128

Lista de Figuras

FIGURA 1.1 – DIAGRAMA CONCEPTUAL DO <i>HARDWARE-IN-THE-LOOP</i>	1
FIGURA 1.2 – ESTRUTURA CONCEPTUAL DO RELATÓRIO	2
FIGURA 2.1 – ILUSTRAÇÃO DO SIMULADOR DE <i>HARDWARE-IN-THE-LOOP</i> (HITL) DESENVOLVIDO	5
FIGURA 2.2 – ILUSTRAÇÃO DO CONCEITO DE SIMULAÇÃO COM HITL SEGUNDO A <i>AMERICAN GNC CORPORATION</i> (EXTRAÍDO DE [AGNC])	5
FIGURA 2.3 - LIGAÇÃO DA PLACA MC-XAS3 COM TRÊS PLACAS ADC24B3C	6
FIGURA 2.4 – ARQUITECTURA PARA A INTERFACE ENTRE A PLACA MC-XAS3 COM O <i>D.MODULE.VC33</i>	7
FIGURA 2.5 – EXEMPLO DA DISTORÇÃO DO CAMPO MAGNÉTICO PROVOCADO POR DIFERENTES VEÍCULOS (DE [CaSm98] E [Bowd95])	8
FIGURA 3.1 – DIAGRAMA FUNCIONAL DA ARQUITECTURA DE AQUISIÇÃO DE SINAL	10
FIGURA 3.2 – DIAGRAMA FUNCIONAL DA PLACA AD24B3C	11
FIGURA 3.3 – COMBINAÇÃO LÓGICA DOS SINAIS <i>nDRDY</i> DE TODOS OS <i>ADSI210'S MASTER</i>	11
FIGURA 3.4 – MÁQUINA DE ESTADOS DA INTERFACE DE <i>SOFTWARE</i> COM A AAS	13
FIGURA 3.5 – FLUXOGRAMA DO CÓDIGO EXECUTADO NA ROTINA DE INTERRUPÇÃO	14
FIGURA 3.6 – EFEITO DA SINCRONIZAÇÃO NO TEMPO DOS DADOS DE SAÍDA	15
FIGURA 3.7 – DIAGRAMA TEMPORAL DE UM CICLO DE LEITURA DE DADOS DE TRÊS CONVERSORES	15
FIGURA 3.8 – DIAGRAMA CONCEPTUAL DA INTERFACE ENTRE A PLACA MC-XAS3 E O <i>D.MODULE.VC33</i>	16
FIGURA 3.9 – DIAGRAMA TEMPORAL DE UM ACESSO DE LEITURA DE DADOS DO <i>D.MODULE.VC33</i> NA DP-RAM	18
FIGURA 3.10 – DIAGRAMA TEMPORAL DE UM ACESSO DE ESCRITA DE DADOS DO <i>D.MODULE.VC33</i> NA DP-RAM	19
FIGURA 3.11 – CIRCUITO LÓGICO ASSÍNCRONO QUE GERA A INTERRUPÇÃO NO <i>D.MODULE.VC33</i>	20
FIGURA 3.12 – CIRCUITO LÓGICO ASSÍNCRONO QUE GERA A INTERRUPÇÃO NA MC-XAS3	21
FIGURA 3.13 – CIRCUITO LÓGICO ASSÍNCRONO QUE CONTROLA O ESTADO DA LINHA <i>nRESIN</i> DO <i>D.MODULE.VC33</i>	21
FIGURA 3.14 – DIAGRAMA FUNCIONAL DO PROTOCOLO DO TIPO <i>MAILBOX</i> IMPLEMENTADO COM A DP-RAM	22
FIGURA 4.1 – DIAGRAMA DO SISTEMA DE NAVEGAÇÃO GLOBAL COM FILTROS COMPLEMENTARES	24
FIGURA 4.2 – SISTEMA DE NAVEGAÇÃO EM CONFIGURAÇÃO <i>FEEDBACK</i>	24
FIGURA 4.3 – INTRODUÇÃO DE RUÍDO PASSA-BANDA ARTIFICIAL	27
FIGURA 4.4 – NOVO DIAGRAMA CONCEPTUAL DO SISTEMA DE NAVEGAÇÃO	31
FIGURA 5.1 – <i>LOOP</i> DO FILTRO DE KALMAN	35
FIGURA 5.2 – EVOLUÇÃO DA REDUÇÃO DO CONSUMO TEMPORAL DO ALGORITMO	37
FIGURA 5.3 – DIAGRAMA REPRESENTATIVO DO PROCEDIMENTO PARA A VALIDAÇÃO DOS ALGORITMOS	38
FIGURA 5.4 – PROTOCOLO DE COMUNICAÇÃO ENTRE <i>MATLAB</i> E XAS3	39
FIGURA 5.5 – TRAJECTÓRIA UTILIZADA PARA A SIMULAÇÃO E TESTES.	40
FIGURA 5.6 – DIFERENÇA ENTRE A POSIÇÃO E A ORIENTAÇÃO ESTIMADA NO DSP vs <i>MATLAB</i> (CKF)	40
FIGURA 5.7 – EVOLUÇÃO TEMPORAL DA ATITUDE E POSIÇÃO ESTIMADAS E REAL (CKF)	41
FIGURA 5.8 – EVOLUÇÃO TEMPORAL DA ESTIMAÇÃO E DOS RESPECTIVOS ERROS (DSP vs <i>MATLAB</i> - EKF)	42
FIGURA 5.9 – EVOLUÇÃO TEMPORAL DA ESTIMAÇÃO E DOS RESPECTIVOS ERROS (DSP vs REAL - EKF)	43
FIGURA 5.10 – FLUXO DE INFORMAÇÃO NO <i>HARDWARE</i> NO <i>LOOP</i>	44
FIGURA 5.11 – EVOLUÇÃO TEMPORAL DA MEDIDA SIMULADA DOS SENSORES EM <i>MATLAB</i> E NO DSP	45
FIGURA 5.12 – DIFERENÇA ENTRE OS DADOS SENSORIAIS NA IMPLANTAÇÃO EM <i>MATLAB</i> E OS DADOS DISPONÍVEIS PARA O DSP	46
FIGURA 5.13 – EVOLUÇÃO TEMPORAL DA ESTIMAÇÃO E DOS RESPECTIVOS ERROS (CKF)	46
FIGURA 5.14 – ESTIMAÇÃO DAS POLARIZAÇÕES DOS GIROSCÓPIOS E DOS ACELERÓMETROS (CKF)	47
FIGURA 5.15 – EVOLUÇÃO TEMPORAL DA ESTIMAÇÃO E DOS RESPECTIVOS ERROS (EKF)	48
FIGURA 5.16 – EVOLUÇÃO TEMPORAL DA ESTIMAÇÃO E DOS RESPECTIVOS ERROS (EKF)	49
FIGURA A.1 – ÂNGULOS DE EULER Z-Y-X	53
FIGURA A.2 – REFERENCIAIS NED, ECEF E DO CORPO	58
FIGURA B.1 – ILUSTRAÇÃO DA NÃO SOBREPOSIÇÃO ENTRE OS PÓLOS MAGNÉTICOS E OS PÓLOS GEOGRÁFICOS TERRESTRES	61
FIGURA B.2 – MAPA MUNDIAL DE DECLINAÇÃO MAGNÉTICA (EXTRAÍDO DE [NGDC])	62
FIGURA B.3 – DESCRIÇÃO DO CAMPO MAGNÉTICO TERRESTRE NUM PONTO DA SUPERFÍCIE DA MESMA	63
FIGURA B.4 – LEITURA DE DOIS SENSORES MAGNÉTICOS ORTOGONAIS (INSERIDOS NO PLANO HORIZONTAL) QUANDO RODADOS DE 360° NO PLANO HORIZONTAL: EM RELAÇÃO AO RUMO (A), E UM EM RELAÇÃO AO OUTRO (B)	64

FIGURA B.5 – EVOLUÇÃO DO MODELO DO ERRO EM FUNÇÃO DE <i>YAW</i> MEDIDO COM A PERTURBAÇÃO DE DESALINHAMENTO (A), DE <i>HARD IRON</i> (B), DE <i>SOFT IRON</i> (C) E TOTAL (D)	67
FIGURA B.6 – REPRESENTAÇÃO DA DISTORÇÃO DE <i>HARD IRON</i> NO REFERENCIAL DO SENSOR (A) E NO ESPAÇO DE <i>HEADING</i> (B)	68
FIGURA B.7 – REPRESENTAÇÃO DA DISTORÇÃO DE <i>SOFT IRON</i> NO REFERENCIAL DO SENSOR (A) E NO ESPAÇO DE <i>HEADING</i> (B)	69
FIGURA B.8 - REPRESENTAÇÃO DA DISTORÇÃO DEVIDO A DIFERENTES FACTORES DE ESCALA PARA CADA EIXO DO MAGNETÓMETRO NO REFERENCIAL DO SENSOR (A) E NO ESPAÇO DE <i>HEADING</i> (B)	70
FIGURA B.9 - REPRESENTAÇÃO DA DISTORÇÃO DEVIDO AO DESALINHAMENTO NO PLANO HORIZONTAL ENTRE OS EIXOS DO MAGNETÓMETRO E O REFERENCIAL DO CORPO PARA REFERENCIAL DO SENSOR (A) E PARA O ESPAÇO DE <i>HEADING</i> (B)	70
FIGURA B.10 – FLUXOGRAMA DO PROCEDIMENTO DE CORRECÇÃO DAS LEITURAS DO MAGNETÓMETRO	72
FIGURA B.11 – ILUSTRAÇÃO DO PROCESSO DE CORRECÇÃO DAS PERTURBAÇÕES QUE AFFECTAM AS MEDIDAS DO MAGNETÓMETRO QUANDO O MESMO COMPLETA UMA VOLTA DE 360° NO PLANO HORIZONTAL	73
FIGURA B.12 – ERRO NA OBTENÇÃO DA MEDIDA DE <i>HEADING</i> A PARTIR DO MAGNETÓMETRO ANTES (A) E DEPOIS DA CORRECÇÃO (B)	74
FIGURA B.13 – IMPACTO DA CALIBRAÇÃO DO MAGNETÓMETRO NO CÁLCULO DO <i>HEADING</i>	75
FIGURA B.14 – CONFIGURAÇÃO ORBITAL DOS SATÉLITES GPS (EXTRAÍDO DE [LECTGPS])	76
FIGURA B.15 – TRILATERAÇÃO PLANAR COM DESFASAGEM TEMPORAL (EXTRAÍDO DE [LECTGPS])	76
FIGURA B.16 – INCLINÓMETRO (EXTRAÍDO DE [AZIN02])	77
FIGURA C.1 – FILTRO COMPLEMENTAR DE ATITUDE	82
FIGURA C.2 – FILTRO COMPLEMENTAR DE POSIÇÃO	84
FIGURA C.3 – DIAGRAMA DO SISTEMA DE NAVEGAÇÃO GLOBAL COM FILTROS COMPLEMENTARES	84
FIGURA E.1 – REPRESENTAÇÃO GRÁFICA DO MODELO DO VEÍCULO (EXTRAÍDO DE [CAVA03])	88
FIGURA F.1 – PLATAFORMA COMUM COM O VEÍCULO E SENSORES	93
FIGURA F.2 – MODELO DO VEÍCULO EM <i>SIMULINK</i> COM REFERÊNCIA ÀS EQUAÇÕES DE CADA BLOCO	94
FIGURA F.3 – DINÂMICA DO CORPO RÍGIDO	94
FIGURA F.4 – GERAÇÃO DAS MEDIDAS DE POSIÇÃO E DE ORIENTAÇÃO	94
FIGURA F.5 – MODELO DOS SENSORES EM <i>SIMULINK</i>	95
FIGURA F.6 – MODELO DOS VÁRIOS SENSORES EM <i>SIMULINK</i>	96
FIGURA F.7 – MODELO GERAL EM <i>SIMULINK</i>	96
FIGURA G.1 – PLACA AD24B3C	97
FIGURA G.2 – DIAGRAMA DE LIGAÇÕES DO <i>BUFFER</i> EM CADA PLACA AD24B3C	99
FIGURA G.3 – LIGAÇÕES PARA O SINCRONISMO EM CADA PLACA AD24B3C	100
FIGURA G.4 – ESQUEMÁTICO DO FILTRO RC	100
FIGURA G.5 – CONFIGURAÇÃO DE ENTRADA USANDO V_{BIAS} (EXTRAÍDO DE [ADDS00])	101
FIGURA G.6 – ESQUEMA LÓGICO QUE PERMITE SELECIONAR CADA ADC	101
FIGURA G.7 – ESQUEMA LÓGICO QUE PERMITE A MULTIPLEXAGEM DO <i>SDOUT</i>	103
FIGURA G.8 – FLUXOGRAMA DA FUNÇÃO <i>DRIVER OPEN</i>	105
FIGURA G.9 – FLUXOGRAMA DA LEITURA DOS CONVERSORES	106
FIGURA G.10 – HISTOGRAMA DO ADC 0 PLACA AD24B3C 0	108
FIGURA G.11 – HISTOGRAMA DO ADC 1 PLACA AD24B3C 0	108
FIGURA G.12 – HISTOGRAMA DO ADC 2 PLACA AD24B3C 0	108
FIGURA H.1 – DIAGRAMA FUNCIONAL DA PLACA DSPIf	111
FIGURA H.2 – ESQUEMA ELÉCTRICO DA PLACA DSPIf	112
FIGURA H.3 – CAMADA SUPERIOR DA DSPIf	113
FIGURA H.4 – CAMADA INFERIOR DA DSPIf	113
FIGURA H.5 – CIRCUITO LÓGICO SIMBÓLICO IMPLEMENTADO PELA XPLA NUM ACESSO DE ESCRITA/LEITURA DO <i>D.MODULE.VC33</i> À DP-RAM	115
FIGURA H.6 – CIRCUITO LÓGICO SIMBÓLICO IMPLEMENTADO PELA XPLA NUM ACESSO DE ESCRITA/LEITURA DA MC-XAS3 À DP-RAM	115
FIGURA H.7 – CIRCUITO LÓGICO SIMBÓLICO IMPLEMENTADO PELA XPLA NUM ACESSO AO SEGMENTO DE COMANDOS DA MC-XAS3 E DO <i>D.MODULE.VC33</i>	116
FIGURA H.8 – DIAGRAMA TEMPORAL DO CIRCUITO LÓGICO ASSÍNCRONO QUE GERA A INTERRUPÇÃO NA MC-XAS3	117
FIGURA H.9 – DIAGRAMA TEMPORAL CIRCUITO LÓGICO ASSÍNCRONO QUE GERA A INTERRUPÇÃO NO <i>D.MODULE.VC33</i>	117
FIGURA H.10 – DIAGRAMA TEMPORAL DO CIRCUITO LÓGICO DE CONTROLO DO PINO <i>NRESIN</i> DO <i>D.MODULE.VC33</i>	117
FIGURA I.1 – DIAGRAMA FUNCIONAL DA PLACA ADDAOCT	123
FIGURA J.1 – DIAGRAMA DE BLOCOS DA PLACA MC-XAS3 [XARI]	125
FIGURA K.1 – ASPECTO EXTERIOR DO <i>D.MODULE.VC33</i>	126

Lista de Tabelas

TABELA 3.1 – TEMPOS PARA O DIAGRAMA TEMPORAL DA FIGURA 3.7	16
TABELA 3.2 – COMANDOS DISPONÍVEIS PARA A MC-XAS3 E PARA O <i>D.MODULE.VC33</i>	17
TABELA 3.3 – TEMPOS PARA O DIAGRAMA TEMPORAL DA FIGURA 3.9	19
TABELA 3.4 – TEMPOS PARA O DIAGRAMA TEMPORAL DA FIGURA 3.10	20
TABELA 3.5 – MENSAGENS DISPONÍVEIS A AMBOS OS DISPOSITIVOS NO PROTOCOLO IMPLEMENTADO DO TIPO <i>MAILBOX</i>	22
TABELA 3.6 – SEGMENTO 0 DA MC-XAS3 E SEGMENTO DO <i>D.MODULE.VC33</i>	22
TABELA 5.1 – COMPLEXIDADE DO ALGORITMO NÃO OPTIMIZADO VS ALGORITMO OPTIMIZADO	36
TABELA 5.2 – PARÂMETROS DA MODELAÇÃO DOS SENSORES	39
TABELA 5.3 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO DSP (CKF)	41
TABELA 5.4 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO <i>MATLAB</i> (CKF)	41
TABELA 5.5 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO DSP (EKF)	43
TABELA 5.6 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO <i>MATLAB</i> (EKF)	43
TABELA 5.7 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO DSP (CKF)	47
TABELA 5.8 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO DSP (EKF)	48
TABELA 5.9 – ESTATÍSTICAS DO ERRO DO FILTRO EXECUTADO NO DSP (EKF)	49
TABELA B.1 – SENSORES UTILIZADOS PARA A NAVEGAÇÃO	60
TABELA E.1 – CARACTERÍSTICAS FÍSICAS DO VEÍCULO	88
TABELA F.1 – POLARIZAÇÕES E VARIÂNCIAS DOS RUÍDOS DOS SENSORES	95
TABELA G.1 – ENDEREÇAMENTO PARA A SELECÇÃO DE CADA CONVERSOR NA ARQUITECTURA COMPLETA	102
TABELA G.2 – PERFORMANCE DA PLACA AD24B3C 0	108
TABELA H.1 – CONFIGURAÇÃO DOS PINOS DO BARRAMENTO EXTERNO DO <i>D.MODULE.VC33</i> UTILIZADOS NA INTERFACE COM A PLACA DSPIf (EXTRAÍDO DE [VCUG01])	111
TABELA H.2 – CONFIGURAÇÃO DOS PINOS DO BARRAMENTO EXTERNO DA PLACA MC-XAS3 UTILIZADOS NA INTERFACE COM A PLACA DSPIf (EXTRAÍDO DE [VCUG01])	112
TABELA H.3 – MAPA DE MEMÓRIA DA PLACA MC-XAS3	114
TABELA H.4 – MAPA DE MEMÓRIA DO <i>D.MODULE.VC33</i>	114
TABELA J.1 – PINOS DISPONÍVEIS NA FICHA P1	125

Lista de Siglas

<i>AAS</i>	<i>Arquitetura de Aquisição de Sinal</i>
<i>ADC</i>	<i>Conversor Analógico/Digital</i>
<i>AH</i>	<i>Arquitetura de Hardware</i>
<i>ALU</i>	<i>Arithmetic and Logic Unit</i>
<i>CAN</i>	<i>Controller Area Network</i>
<i>CKF</i>	<i>Filtro de Kalman Complementar</i>
<i>CPU</i>	<i>Central Processing Unit</i>
<i>DMA</i>	<i>Direct Memory Access</i>
<i>DP-RAM</i>	<i>Dual Port RAM</i>
<i>DSOR</i>	<i>Dynamical Systems and Ocean Robotics Lab</i>
<i>DSP</i>	<i>Digital Signal Processor</i>
<i>EKF</i>	<i>Filtro de Kalman Estendido</i>
<i>EMP</i>	<i>Estimador Magneto-Pendular</i>
<i>GPS</i>	<i>Sistema Global de Posição</i>
<i>HITL</i>	<i>Hardware In The Loop</i>
<i>INS</i>	<i>Sistema de Navegação Inercial</i>
<i>ISR</i>	<i>Instituto de Sistemas e Robótica</i>
<i>MFLOPS</i>	<i>Milhões de Operações de Vírgula Flutuante por Segundo</i>
<i>PAL</i>	<i>Programmable Array Logic</i>
<i>PC</i>	<i>Computador Pessoal</i>
<i>PCA</i>	<i>Programmable Counter Array</i>
<i>PCMCIA</i>	<i>Personal Computer Memory Card International Association</i>
<i>PGA</i>	<i>Programable Gain Amplifier</i>
<i>PLD</i>	<i>Programmable Logic Device</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>RS232</i>	<i>Recommended Standard 232</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>SSI</i>	<i>Server Side Includes</i>
<i>TMR</i>	<i>Turbo Mode Rate</i>
<i>UART</i>	<i>Universal Asynchronous Receiver/Transmitter</i>
<i>XPLA</i>	<i>Extended Programable Logic Array</i>

1 Introdução

A investigação na área dos veículos autónomos apresenta um elevado interesse académico, económico e militar fruto do seu vasto leque de aplicações que se estendem desde operações em condições adversas (e.g. centrais atómicas) até à exploração aeroespacial. Este tipo de veículos tem uma grande utilidade no cumprimento de missões onde é arriscada e desaconselhada a presença de operadores humanos, sendo muito utilizados em aplicações militares. É provavelmente nestas, onde se encontra o topo da tecnologia deste tipo de plataformas que incorporam alguns dos sistemas de navegação com melhor desempenho.

Este tipo de veículos são geralmente dispendiosos, razão pela qual surgiu a necessidade de desenvolver um simulador de *Hardware-In-The-Loop* (HITL) que permite o teste de algoritmos de navegação e controlo na Arquitectura de *Hardware* (a utilizar no veículo) em conjunto com o modelo da dinâmica do veículo. Este avançado cenário de simulação/teste permite criar condições de operação muito próximas das reais. Um simulador deste tipo é bastante importante devido às consequências que um erro no *software* ou no *hardware* desenvolvido possa causar ao veículo e ao ambiente envolvente. Consegue-se assim, reduzir ao mínimo as probabilidades de insucesso num teste real. O HITL envolve a comunicação entre a arquitectura de hardware de tempo real, o modelo do veículo e os diversos modelos dos sensores [CaVa03]. Com esse objectivo foi desenvolvida a arquitectura apresentada na Figura 1.1.

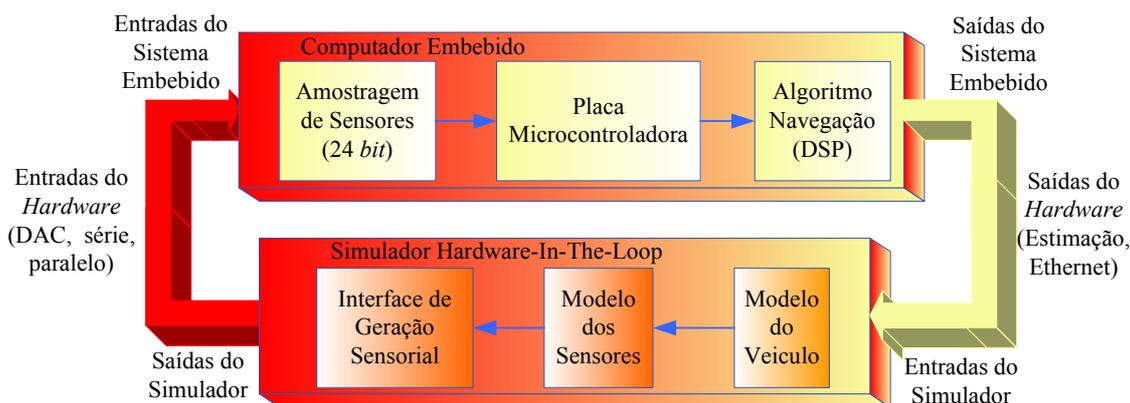


Figura 1.1 – Diagrama conceptual do *Hardware-In-The-Loop*

A arquitectura de *Hardware*, ou computador embebido (Figura 1.1), é parte fundamental de um veículo autónomo, sendo responsável por todo o processo de aquisição e processamento de sinal. A proposta teve como objectivo evoluir de uma já existente [PiFe02], dotando esta de uma maior capacidade computacional e exactidão na aquisição de sinais analógicos. Esta arquitectura pode assim ser dividida em duas partes: uma de aquisição de sinal (constituída por uma placa microcontroladora e por quatro placas, cada uma com três conversores Analógico/Digital) e outra de processamento digital de sinal (formada por uma placa microcontroladora, uma de processamento e outra para a interface entre estas). A arquitectura de aquisição de sinal é utilizada em sistemas com restrições de tempo real, não havendo no código desenvolvido qualquer espera activa. Para a arquitectura de processamento digital foi desenvolvido um protocolo de comunicação entre esta e a placa microcontroladora recorrendo a uma memória de dois portos (Dual-Port RAM). Este funciona assincronamente, não havendo também nenhuma espera activa.

Um sistema de navegação é responsável pela determinação da orientação, posição e velocidade lineares de um corpo relativamente a um referencial, recorrendo para tal a um conjunto de sensores. Historicamente, este é um problema que tem acompanhado a evolução científica. O recurso à posição das estrelas, o desenvolvimento de instrumentos como a Bússola ou o Astrolábio e, mais recentemente, os sistemas de navegação por satélite demonstram que esta é uma área de investigação dinâmica e actual.

Os INS (*Inertial Navigation Systems*) constituem uma das arquitecturas mais comuns entre os sistemas de navegação, pois permitem obter uma estimativa da orientação, velocidade e posição através das leituras de giroscópios e acelerómetros. Existem três classes de INS, diferenciadas pela forma como os sensores são instalados no veículo. No presente trabalho utilizaram-se os *Strapdown Systems* que apresentam os sensores montados directamente na estrutura do veículo e estão alinhados com os seus eixos. Este tipo de sistema é caracterizado por baixos custos e algoritmos evoluídos sendo por isso computacionalmente exigente. Os INS são sistemas dinâmicos instáveis, independentemente da qualidade dos sensores e algoritmos utilizados, para lidar com este problema introduzem-se correcções a partir de sensores auxiliares. Neste trabalho implementaram-se algoritmos de navegação, filtragem e fusão sensorial para sistemas *Strapdown* integrados com GPS e medidas auxiliares de orientação [CaVa03]. Pretendeu-se que estes algoritmos fossem aplicados à arquitectura proposta, tendo sido necessário desenvolver técnicas de implementação eficiente de algoritmos.

O relatório está estruturado conforme a Figura 1.2.

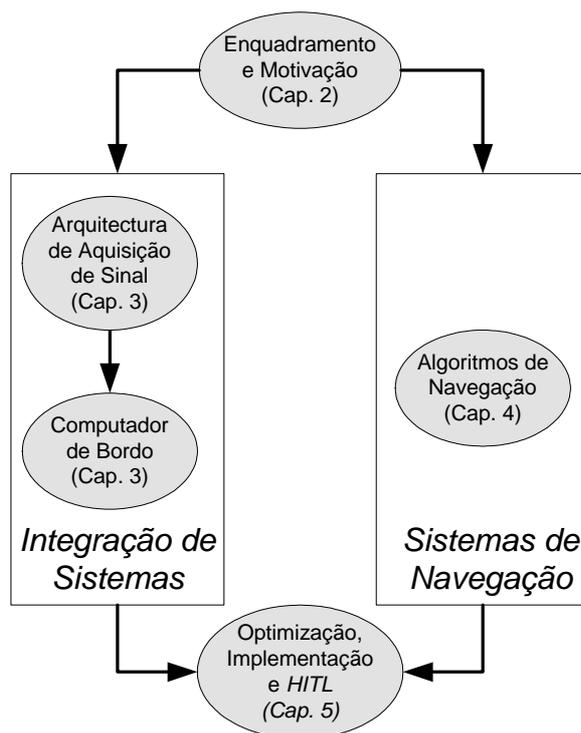


Figura 1.2 – Estrutura conceptual do relatório

No Capítulo 2 enquadra-se o presente trabalho nos objectivos do laboratório DSOR/ISR. Apresentam-se alguns projectos desenvolvidos no passado assim como a importância destes para a elaboração do presente projecto, referindo ainda as principais contribuições do trabalho descrito neste relatório.

No Capítulo 3 apresentam-se a integração do *hardware* e as metodologias a que se recorreram no desenvolvimento do *software* e *firmware*. Ambas as arquitecturas são utilizadas em aplicações com restrições de tempo real.

No Capítulo 4 descrevem-se os algoritmos de navegação implementados: filtro de Kalman complementar e filtro de Kalman estendido. O primeiro recorre directamente às medidas dos sensores disponíveis (acelerómetros, giroscópios, magnetómetro e GPS) combinando duas medidas da mesma grandeza que apresentem larguras de bandas diferentes e complementares. O segundo funde as estimativas de um INS com medidas auxiliares, cuja diferença entre ambas produz um erro de posição e de orientação que é introduzido num filtro de Kalman estendido projectado no espaço do erro segundo uma configuração *direct feedback*.

No Capítulo 5 é apresentado o desenvolvimento de técnicas de implementação eficiente de algoritmos de navegação. De seguida, descreve-se a implementação dos algoritmos na arquitectura de *hardware*. São apresentados os resultados relativos à validação destes algoritmos e do desempenho dos mesmos utilizando o simulador *Hardware-In-The-Loop*. A implementação e validação de algoritmos numa arquitectura de *hardware* constitui uma das principais contribuições do presente trabalho.

No Capítulo 6 apresentam-se as principais conclusões a retirar do trabalho realizado e sugerem-se possíveis desenvolvimentos futuros.

Adicionalmente chama-se a atenção do leitor para o Apêndice B onde se sugere uma evolução de um modelo para o erro do magnetómetro e um algoritmo de calibração que permite aumentar a performance do sensor em causa.

2 Arquitectura de *Hardware* para Sistemas de Navegação

2.1 Enquadramento do Projecto e Trabalho Anterior

A actividade científica desenvolvida no (DSORlab) está relacionada com o projecto e operação de veículos autónomos, quer submarinos quer aéreos. As áreas de interesse do referido grupo vão desde a navegação, controlo, passando pelo controlo de missão e integração de sistemas nesses veículos. Foi desenvolvida neste laboratório, uma arquitectura digital distribuída de tempo real que vem a ser utilizada e testada no âmbito de projectos com veículos autónomos. O presente trabalho enquadra-se na área de investigação deste laboratório relacionada com a integração de sistemas de *hardware* analógico e digital, calibração de sensores, implementação eficiente de algoritmos de navegação e validação por *Hardware-In-The-Loop* (HITL) de todo o sistema de navegação.

Enquadrados com os objectivos do DSOR, existiram no passado contribuições valiosas no âmbito da implementação de algoritmos de navegação [PiFe02], tendo sido documentados bons resultados quer ao nível da simulação, quer ao nível da implementação real de uma plataforma inercial. Com este trabalho, pretende-se ganhar experiência ao nível do desempenho de técnicas e algoritmos de filtragem criando para tal, uma ferramenta valiosa (HITL) que permite testar, em condições mais realistas que no passado, os referidos algoritmos trazendo para o ambiente de simulação um grau crescente de realismo.

2.2 Contribuições do Projecto

- **Hardware-In-The-Loop (HITL)**

O conceito por detrás de uma simulação dita de *Hardware-In-The-Loop* (HITL) é a utilização do *hardware* que é parte do sistema final e testá-lo com dados originados em modelos numéricos que tentam descrever o processo a controlar ou monitorizar. A simulação com HITL encontra aplicações em praticamente todas as áreas da engenharia, sendo de especial interesse para este projecto, as aplicações aos sistemas de navegação, condução e controlo de veículos autónomos, devido ao risco envolvido no teste da plataforma.

No que à navegação de veículos autónomos diz respeito, uma ferramenta deste género representa uma mais valia no sentido em que se consegue reproduzir num ambiente seguro, todos os fenómenos que se consiga modelar. O interesse neste tipo de simuladores vai muito para além dos objectivos académicos, visto que, os mesmos reduzem drasticamente a necessidade de efectuar testes de campo que em geral são muito dispendiosos, com elevado risco para a plataforma e consomem grandes quantidades de tempo.

Na Figura 2.1 ilustra-se a disposição do simulador de HITL desenvolvido. Este simulador incorpora um variado leque de componentes individuais, entre os quais se destacam placas desenvolvidas no DSOR mas integradas no âmbito deste projecto, tais como a placa DSPIf e a placa AD24B3C (ver Capítulo 3). Outros componentes que fazem parte da arquitectura de tempo real desenvolvida e integrada pelo DSOR, são a placa microcontroladora MC-XAS3 (ver Apêndice J), a placa PCMCIA (ver [BCD04]), a placa PWROPTO485 (ver [BCD04]) e a placa ADDAOCT (ver Apêndice I). A geração da dinâmica do veículo e modelação sensorial é conseguida em ambiente *MATLAB & Simulink Release 13* da *The MathWorks, Inc* num Computador Pessoal (PC) com o sistema operativo *MicroSoft Windows 2000*.

Neste simulador implementou-se a modelação de toda a interface do Computador de Bordo (CB) com os sensores. Os acelerómetros, os giroscópios e os magnetómetros (ver Apêndice B)

disponibilizam as suas leituras sob a forma de um sinal eléctrico analógico, pelo que, foi simulada a sua interface recorrendo a DAC's (*Digital to Analog Converters*). Como a interface com a unidade de GPS (*Global Positioning System*) real é por comunicação série (*RS-232*), simulou-se também essa característica. Assim, o CB fica exposto a todas as entradas de informação presentes numa missão com um veículo real.

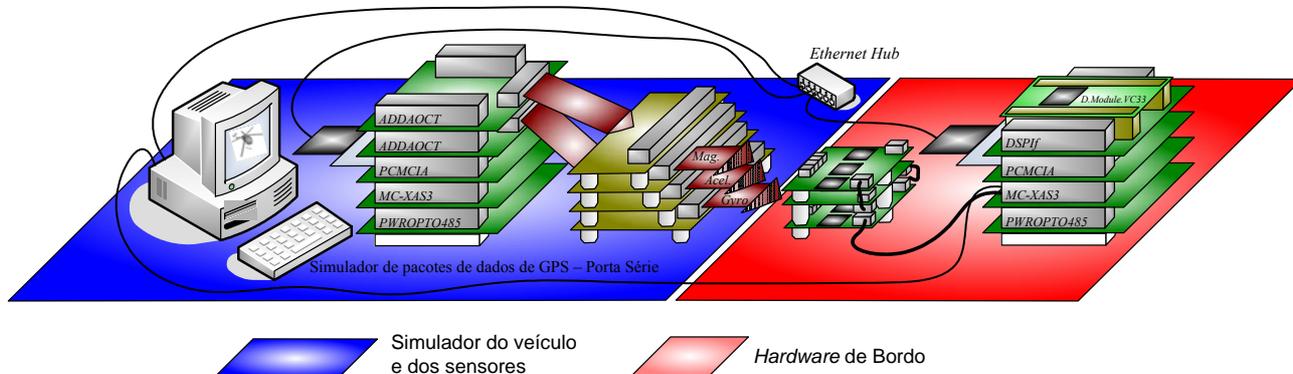


Figura 2.1 – Ilustração do simulador de *Hardware-In-The-Loop* (HITL) desenvolvido

Existem no mercado soluções com os mesmos objectivos que os que se pretendem atingir com o desenvolvimento desta ferramenta. Um fabricante de sistemas deste tipo, sugere de forma gráfica as aplicações deste conceito da forma que se apresenta na Figura 2.2. Estes sistemas são muito dispendiosos atingindo, em geral, várias centenas de milhar de euro. O trabalho efectuado justifica-se, enquadrando este simulador nas necessidades específicas do DSOR no teste de veículos autónomos, com ferramentas sobre as quais possua a capacidade de modificação ou *upgrade* de funcionalidades.

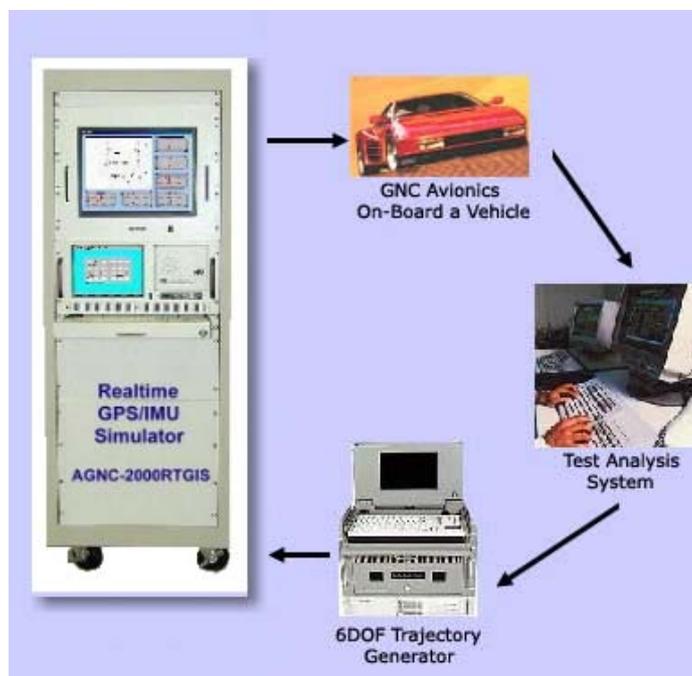


Figura 2.2 – Ilustração do conceito de simulação com HITL segundo a *American GNC Corporation* (extraído de [AGNC])

- **Integração de Sistemas**

No início deste projecto, estava em desenvolvimento no âmbito de outro trabalho no DSOR [CoMo02], uma placa de aquisição de sinal de alta resolução (24 *bit*) a integrar na arquitectura do referido grupo. A integração dessa placa (ADC24B3C descrita no Apêndice G) na arquitectura (a interface com a placa MC-XAS3) tornou-se um dos objectivos iniciais deste projecto. Cada placa ADC24B3C tem três ADC's (*Analog to Digital Converters*) ADS1210 e permite uma utilização em simultâneo de três placas iguais mas com endereços diferentes. A Figura 2.3 apresenta uma ilustração da ligação de três placas ADC24B3C com a placa MC-XAS3.

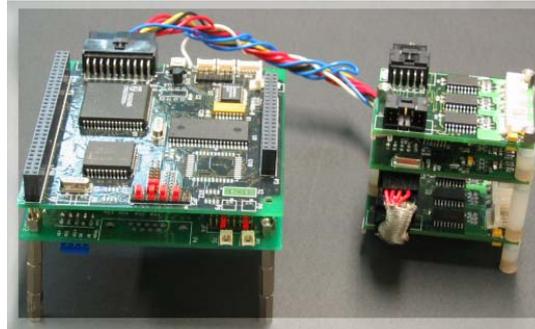


Figura 2.3 - Ligação da placa MC-XAS3 com três placas ADC24B3C

Para fazer interface entre as placas ADC e o microcontrolador foi desenvolvido um *driver* utilizando a linguagem ANSI C e a linguagem *Assembly* do microcontrolador *Philips XA-S3* (placa MC-XAS3), tendo também sido revisto e melhorado o *firmware* da placa ADC24B3C. No desenvolvimento desta interface fez-se uso de metodologias e estratégias de programação de dispositivos com requisitos de tempo real.

Encontrava-se em desenvolvimento no DSOR, no decorrer deste trabalho, uma placa DSPIf (descrito no Apêndice H), que permitiria a interface entre o barramento da placa MC-XAS3 e o barramento de uma nova placa, *D.Module.VC33* da *D.SignT*, fazendo uso de uma *Dual-Port RAM* (DP-RAM). O *D.Module.VC33* consiste num Computador Modular de alto desempenho e baixo consumo baseado no DSP (*Digital Signal Processing*) *TMS320VC33* da *Texas Instruments* (ver Apêndice K). No âmbito deste projecto, conseguiu-se fazer a integração da placa DSPIf, através do desenvolvimento de dois *drivers* escritos em ANSI C e em *Assembly* das respectivas máquinas. Foi também desenvolvido *firmware* que implementa a descodificação, na DSPIf, de endereços para acessos aos dois portos da memória partilhada e a troca de mensagens entre os dois barramentos que são o suporte do protocolo de transferências de dados. Na Figura 2.4 ilustra-se a filosofia de interface adoptada. Com esta metodologia de interface desenvolvida no DSOR e integrada no âmbito deste projecto, é possível ter na mesma pilha de placas vários DSP's.

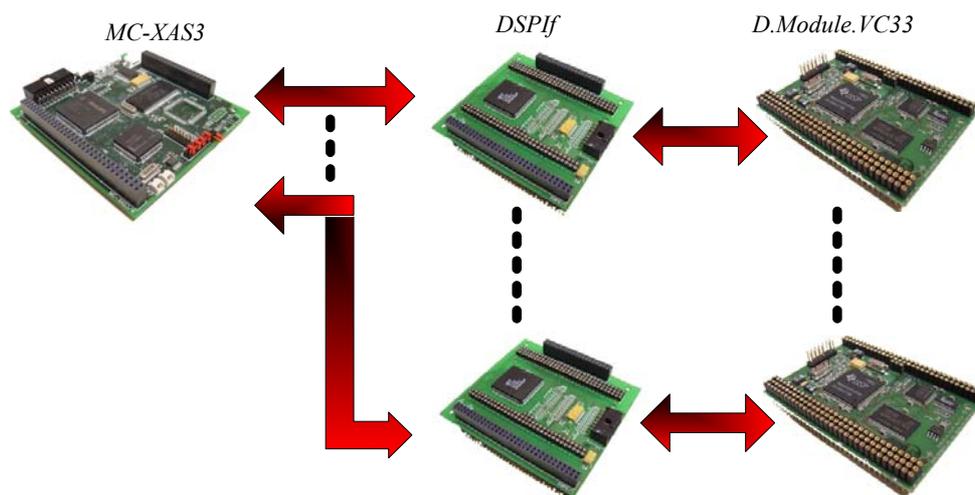


Figura 2.4 – Arquitectura para a interface entre a placa MC-XAS3 com o D.Module.VC33

- **Implementação Eficiente de Algoritmos de Navegação**

Os primeiros passos no desenvolvimento de algoritmos são conseguidos em ambientes de simulação, onde a codificação dos mesmos é facilitada devido ao grau de funcionalidades que estes oferecem. O código desenvolvido, salvo algumas exceções, não é automaticamente portavel para os dispositivos computacionais que equipam os veículos ou plataformas reais. Mesmo quando existem ferramentas para o fazer, estas não conseguem um grau de optimização, que permita aos computadores de bordo de um veículo autónomo atingir o desempenho pretendido.

Avançou-se no sentido da optimização da implementação eficiente de algoritmos, cujo desenvolvimento foi realizado somente em ambiente de simulação. A única linguagem de alto nível para a qual se dispunha de um compilador para o DSP era a ANSI C, linguagem que se revelou uma base sólida para a construção de uma biblioteca de cálculo numérico, não só pela sua portabilidade, mas também pela sua utilização generalizada pela comunidade científica. Assim, foram estudadas e implementadas formas optimizadas de efectuar cálculos numéricos, nomeadamente de cálculo matricial. Para ainda se conseguir um aumento do desempenho, foram implementadas na linguagem *Assembly* do DSP, algumas das operações matemáticas que mais vezes são utilizadas nos algoritmos, garantindo-se que a codificação de mais baixo nível fique o mais optimizada possível.

- **Modelação e Calibração de um Magnetómetro**

O magnetismo terrestre desempenha um papel fundamental na obtenção do rumo de plataformas móveis. Este campo magnético, quando comparado com outras fontes de campo magnético presentes ou originadas em veículos, revela-se de pouca intensidade. Para se conseguir obter informação a partir do vector campo magnético terrestre, é necessário eliminar ou minimizar as outras fontes de perturbação. O fenómeno de distorção magnética provocado pela estrutura dos veículos é um problema real, justificando a investigação nesta área. A Figura 2.5 exemplifica o fenómeno de distorção de um campo magnético por interacção com uma estrutura de diferentes veículos com propriedades magnéticas.

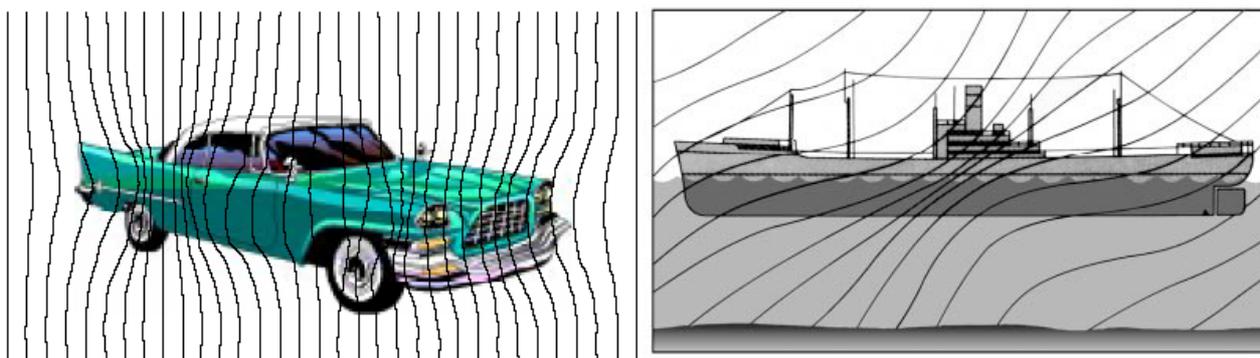


Figura 2.5 – Exemplo da distorção do campo magnético provocado por diferentes veículos (de [CaSm98] e [Bowd95])

Desenvolver uma simulação de HITL só faz sentido quando existem modelos matemáticos que consigam descrever de uma forma o mais próxima possível da realidade, os processos que se pretendem modelar. No caso do magnetómetro (ver Apêndice B), existem modelos para os erros dos mesmo, de utilização e aceitação generalizada. Estes modelos caracterizam os erros de forma aproximada no espaço de uma variável que é obtida matematicamente a partir das leituras directas do sensor. Esta solução não era adequada para modelar as perturbações no espaço das variáveis que um sensor real disponibiliza. Assim, foi feita uma pesquisa no sentido de encontrar outras formas de modelação das perturbações de um magnetómetro directamente nas variáveis do mesmo. Foi encontrada na literatura uma solução que preenchia os requisitos pretendidos. O referido modelo foi adoptado sendo sugeridas algumas evoluções ao mesmo. Depois de obtida uma descrição adequada às necessidades foi concebido no âmbito deste projecto, um algoritmo de calibração que permite melhorar o desempenho do magnetómetro (Apêndice B).

3 Integração de Sistemas

Uma forte componente deste trabalho prende-se com a integração de arquitecturas de *hardware*. Partindo da arquitectura existente no laboratório DSOR, foi proposta uma evolução no sentido de aumentar a performance dos sistemas de navegação. O aumento do desempenho foi conseguido em duas vertentes distintas: aquisição de sinal de alta resolução (24 *bit*) e elevada capacidade de processamento. Estava em desenvolvimento no DSOR, uma arquitectura de aquisição de sinal (AAS) baseada em quatro placas (ADC24B3C, ver Apêndice G) cada uma com três ADC's (*Análogo to Digital Converters*). No âmbito deste trabalho foi desenvolvido um *driver* de interface entre as referidas placas e a placa microcontroladora MC-XAS3. Uma vez assegurada a exactidão da transição das grandezas analógicas para o domínio digital, criou-se a capacidade de processamento que possibilite a implementação em tempo real de algoritmos de navegação. Tal consegue-se à custa de equipar a arquitectura em causa com um DSP de elevada performance e baixo consumo. Para realizar a interface entre o DSP e a placa MC-XAS3 integrou-se uma placa (DSPIf, ver Apêndice H) projectada no DSOR, para a qual foram desenvolvidos *firmware* e dois *drivers* de *software* (um para a placa MC-XAS3 e o outro para o DSP).

3.1 Arquitectura de Aquisição de Sinais de 24 *bit* (AD24B3C)

Os conversores Analógico/Digitais a utilizar desempenham um papel fundamental em toda a arquitectura a desenvolver. Os mesmos terão que apresentar resultados fiáveis e precisos de modo a que se possa prosseguir com todo o sistema de navegação a implementar.

Existiu a necessidade de optar por conversores de elevada resolução, dado que os anteriormente utilizados não conseguiam tirar partido das potencialidades dos sensores existentes. Ou seja, os sensores faziam uma leitura suficientemente próxima da realidade, mas a mesma era posteriormente degradada no processo de discretização pelos ADC's existentes, perdendo-se a exactidão desejada.

O que se pretende é diminuir a importância de uma fonte de erros que está presente sempre que se efectue a conversão analógica/digital. Assim, a escolha recaiu sobre os conversores Analógico/Digitais ADS1210 (consultar Apêndice G.1) da *Burr-Brown* (agora *Texas Instruments*), sendo um conversor *Sigma-Delta* ([BCD04]) de 24 *bit* que permite atingir em determinadas condições resoluções efectivas de 23 *bit*.

3.1.1 Definição da Arquitectura

Habitualmente, é necessário medir grandezas como acelerações lineares e velocidades angulares segundo os três eixos ortogonais, daí a opção de dotar cada placa com três conversores. A filosofia subjacente traduz-se em utilizar “uma placa por cada sensor tri-axial”.

A topologia da arquitectura teve como objectivo satisfazer as necessidades específicas da aquisição de sinais de sensores utilizados na navegação inercial (acelerómetros e giroscópios) e não inercial (magnetómetro). Esta arquitectura tem também utilidade na conversão de outras grandezas, como sejam, pressão, temperatura, entre outras.

Para o controlo e supervisão da Arquitectura de Aquisição de Sinal foi utilizada a placa microcontroladora MC-XAS3 (consultar Apêndice H), desenvolvida no laboratório DSOR/ISR.

Na Figura 3.1 apresenta-se o diagrama funcional da AAS e a sua interface com a placa MC-XAS3. A opção da distribuição de conversores por placas individuais é reflexo dos requisitos das aplicações às quais esta arquitectura se destina, como é o caso dos sistemas distribuídos de tempo real.

A utilização desta arquitectura permite adquirir um total de 12 sinais sem recurso a multiplexagem.

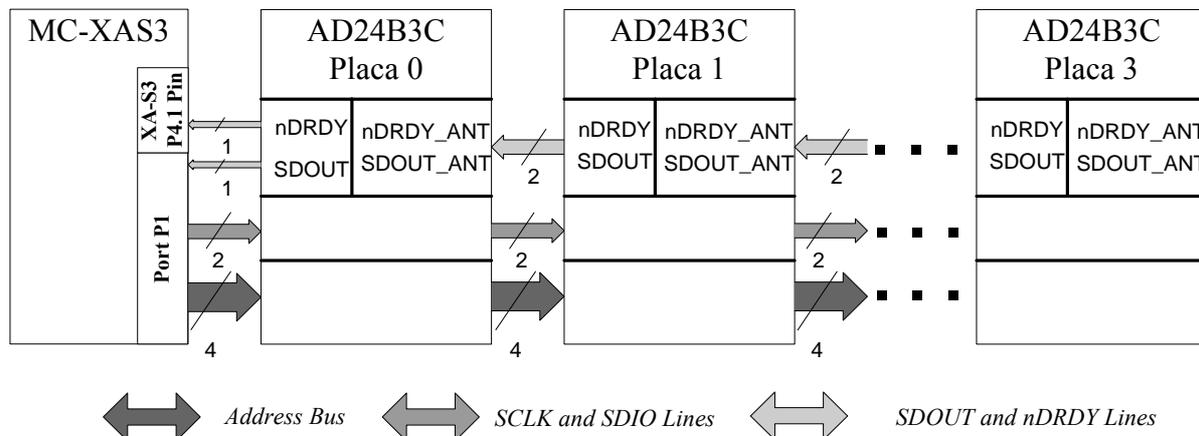


Figura 3.1 – Diagrama funcional da arquitectura de aquisição de sinal

Os sinais de interface são: *Serial Data Input (SDIO)*, *Serial Data Output (SDOUT)*, *Not Data Ready (nDRDY)* e *Clock Input/Output (SCLK)* (Apêndice G.3). Os sinais com o sufixo “_ANT” referem-se aos sinais respectivos da placa anterior.

3.1.2 Definição da Placa AD24B3C

Em sistemas de aquisição de sinal de elevada resolução onde coexistem circuitos analógicos e digitais é essencial a separação por isolamento galvânico, de modo a garantir que os sinais de alta-frequência da parte digital não introduzam ruído na parte analógica de elevada precisão. Este isolamento minimiza a transmissão de ruídos parasitas entre circuitos. O ruído gerado na parte digital deve-se à existência de flancos de tensão.

- **Parte Analógica**

Cada conversor ADS1210 possui um canal diferencial, sendo as duas entradas filtradas analogicamente por um filtro RC passa-baixo, *anti-aliasing*. Dispõe também de um circuito que permite adicionar uma tensão de *offset* a cada entrada analógica de cada conversor, possibilitando deste modo a conversão de valores de tensão pertencente ao intervalo de -10 a +10V (ver Apêndice G.4).

- **Parte Digital**

A parte digital é gerida pelo circuito lógico programável, PLD XCR22V10 (ver Apêndice G.6), que é responsável pela descodificação de endereços, geração do impulso de sincronismo, multiplexagem dos dados série da respectiva placa e das seguintes, bem como da geração dos sinais de *nCS (Not Chip Select)* de cada ADS1210. Existe ainda um *buffer* de sinais digitais (ver Apêndice G.3) para fazer o *drive* das linhas *SDIO* e *SCLK*.

O único canal de comunicação que serve de espinha dorsal para a transferência de dados na Arquitectura de Aquisição de Sinal é baseado em dois tipos de linhas:

- Barramento de endereços
- Sinais de interface

O barramento de endereços permite a selecção individual de um ADS1210, a selecção de todos os ADS1210, a geração dos impulsos de sincronismo e a não selecção de todos os ADS1210's (consultar Tabela G.1). Na Figura 3.2 apresenta-se o diagrama funcional da placa AD24B3C.

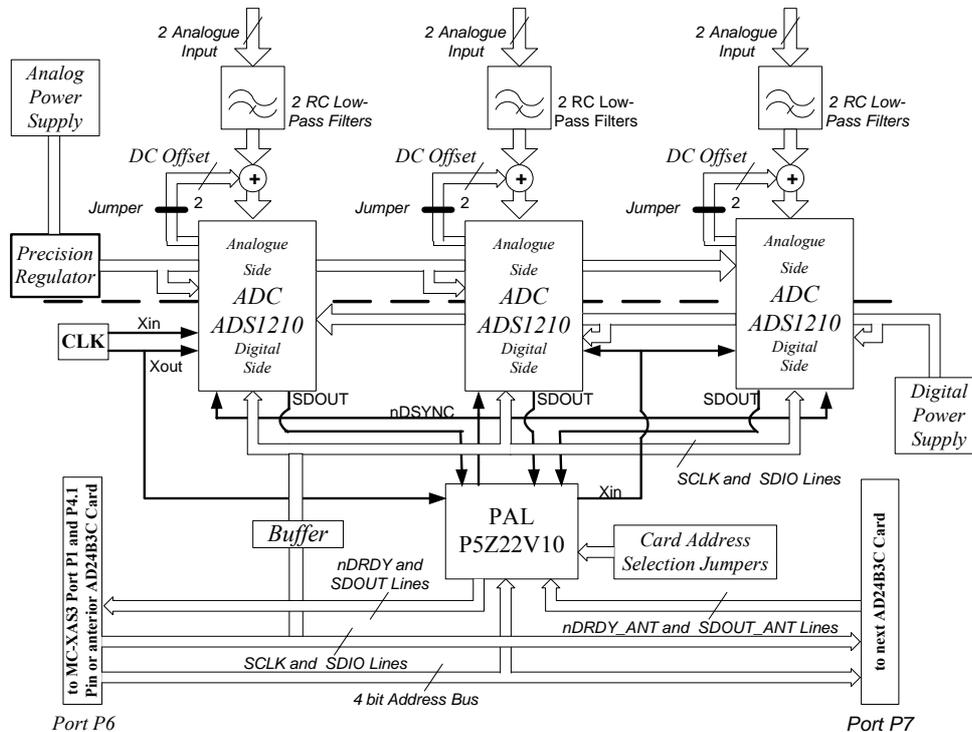


Figura 3.2 – Diagrama funcional da Placa AD24B3C

Quando se faz referência aos sinais *SDOUT* e *nDRDY*, não se pretende referir um pino específico de um qualquer ADS1210, mas sim a sua combinação lógica resultante da multiplexagem. Em cada placa estes sinais são multiplexados com os mesmos das placas anteriores, sendo estes designados *SDOUT_ANT* e *nDRDY_ANT*. De salientar que só um dos três ADS1210 de cada placa apresenta o pino *nDRDY* conectado, o qual foi designado por ADS1210 *Master* (Figura 3.3). Esta situação não representa um problema tal como vai ser demonstrado na Subsecção 3.1.3.

O circuito lógico implementado na PLD de cada placa que multiplexa os sinais *SDOUT* apresenta-se no Apêndice G.5.

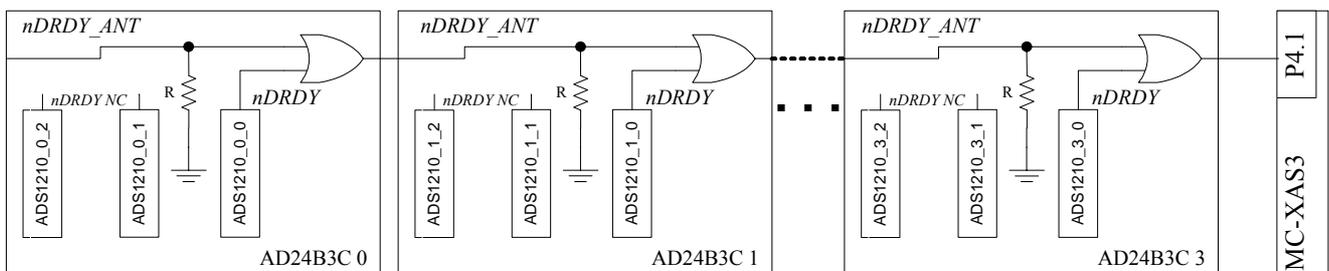


Figura 3.3 – Combinação lógica dos sinais *nDRDY* de todos os ADS1210's *Master*

3.1.3 Metodologia de Interface de Tempo-Real

A interface com a AAS ao nível de *software* foi desenvolvida de modo a garantir que os requisitos de tempo real são cumpridos. Assim, é totalmente proibida a existência de esperas activas bem como a existência de longos blocos de execução que não possam ser interrompidos assincronamente.

A ocorrência de eventos significativos na AAS é transmitida assincronamente à placa MC-XAS3 na forma de um pedido de interrupção, mecanismo descrito com mais pormenor em seguida (Interrupção na Placa MC-XAS3). Os eventos fazem evoluir uma máquina de estados que traduz o funcionamento da arquitectura de *software* que faz a interface com a AAS, evitando assim por completo as esperas activas.

- **Máquina de Estados**

Em termos gerais, pode-se descrever a máquina de estados através do diagrama da Figura 3.4, que apresenta todas as opções de calibração. Os objectivos e implicações de se fazer uma ou outra calibração são descritos em [BCD04]. Está patente na figura a filosofia de verificação de todos os dados escritos nos microcontroladores de todos os ADS1210's. Assim, tudo o que é escrito nos ADS1210's é em seguida lido e verificado no entanto, nunca se interrompe o processo, apenas se notifica o utilizador.

- **Interrupção na Placa MC-XAS3**

A única forma que os conversores ADS1210's dispõem para sinalizar o final de alguma tarefa é com um flanco descendente do pino *nDRDY*. Assim, quando é detectado um flanco descendente (no pino P4.1 – pino de entrada do modulo 0 (zero) do PCA do microcontrolador *Philips XA-S3*) na combinação lógica dos pinos *nDRDY* dos ADS1210's *Master* de cada placa, é disparada uma interrupção na MC-XAS3 cujo tratamento, segue os passos apresentados no fluxograma da Figura 3.5.

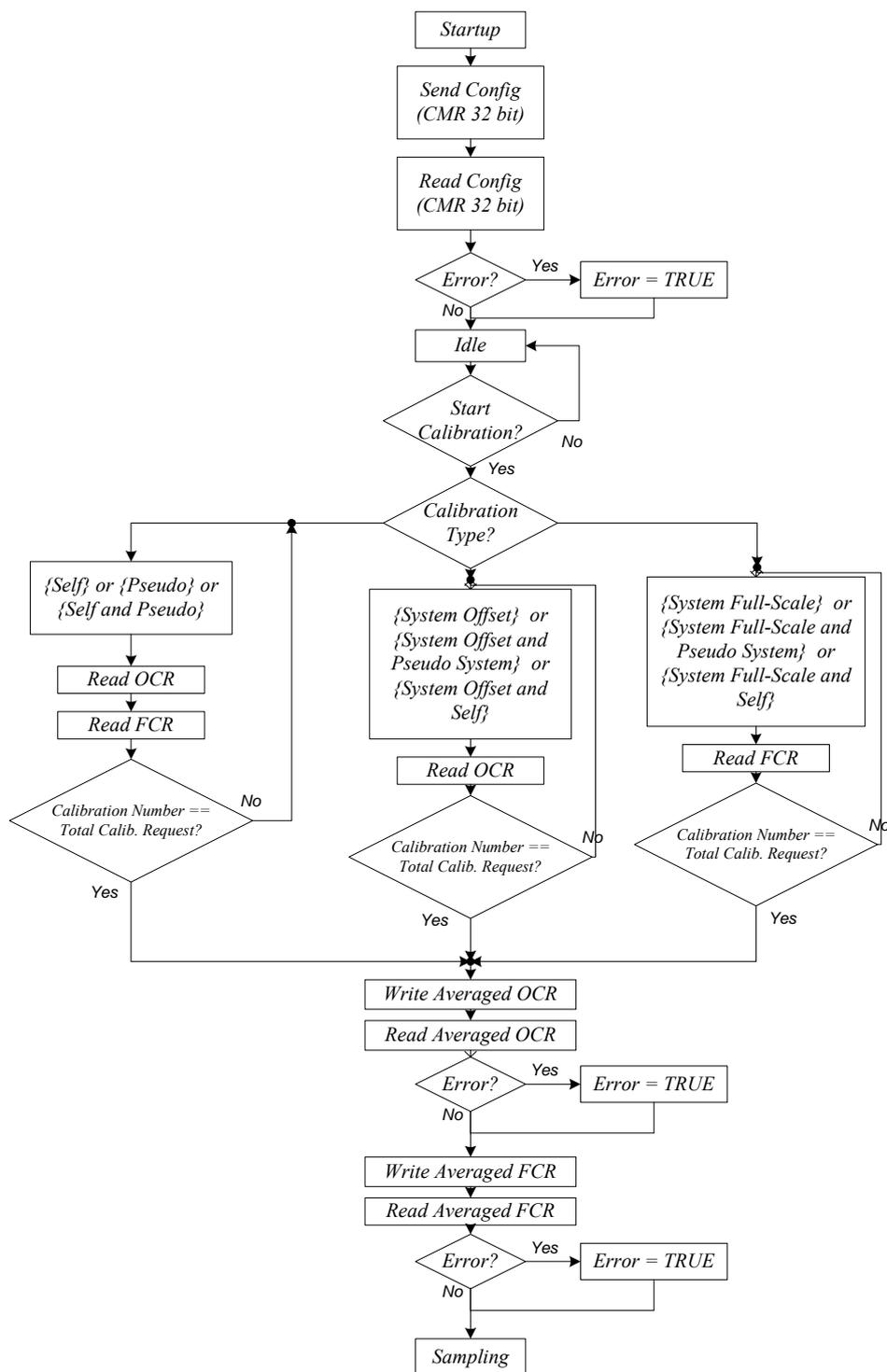


Figura 3.4 – Máquina de estados da interface de *software* com a AAS

O OCR, FCR e CMR são registos internos do conversor ADS1210, apresentados no Apêndice G

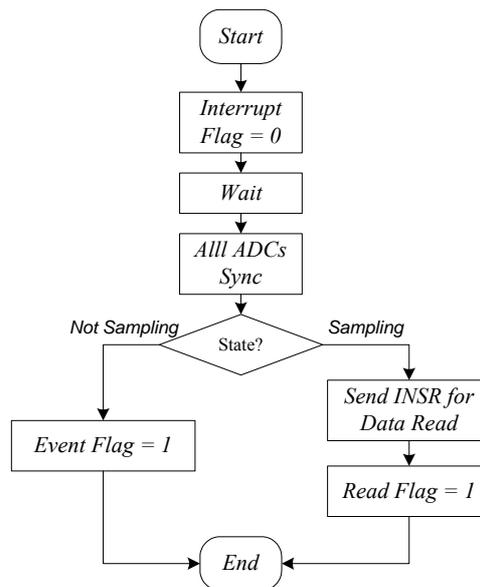


Figura 3.5 – Fluxograma do código executado na rotina de interrupção

- **Sincronização e Leitura de Todos os Conversores Analógico/Digitais**

A sincronização de todos os ADS1210's na arquitectura é conseguida a dois níveis. No primeiro nível, consegue-se um sincronismo exacto entre todos os conversores na mesma placa (daí o facto de não haver problema, dois dos conversores não terem o pino *nDRDY* conectado). Este sincronismo exacto é conseguido através do circuito apresentado no Apêndice G.3 que é implementado na PLD.

Verificou-se experimentalmente que embora cada placa utilize cristais (para gerar o sinal de relógio necessário ao funcionamento dos ADS1210's) com o mesmo valor nominal de frequência (10MHz), na realidade, estes têm uma tolerância de cerca de $\pm 30\text{ppm}$ a 25°C . O resultado destas diferenças é um deslizamento entre os diferentes sinais de relógio de cada placa, como se demonstra no Apêndice G.2.

O problema foi solucionado fazendo a cada ciclo de dados dos ADS1210's, uma sincronização simultânea (colocar a contagem do modulador de todos os conversores a zero) a todos os conversores, o que constitui o segundo nível de sincronização. O pino de entrada *nDSYNC* coloca a contagem actual do modulador (ver [BCD04]) a zero quando recebe um flanco descendente. Este pino permite sincronizar os dados de saída de múltiplos conversores, tal como se pretende. A sincronização envolve configurar cada ADS1210 para o mesmo Rácio de Decimação, Taxa de Sobreamostragem (*Turbo Mode Rate* - TMR) e disponibilizar um sinal comum para as entradas X_{IN} . Consegue-se assim, que o sinal *nDRDY Low* de todos os conversores (inclusive os que se encontram em placas distintas) transite para o nível lógico zero, aproximadamente, no mesmo instante de tempo (ver Figura 3.6).

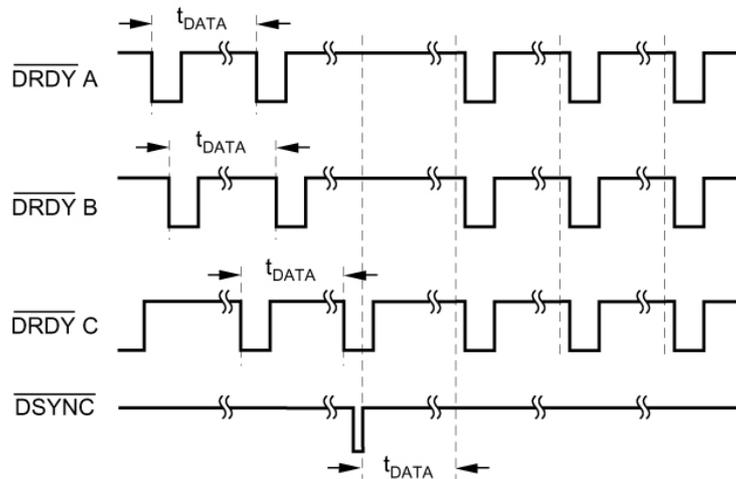


Figura 3.6 – Efeito da sincronização no tempo dos dados de saída

No que diz respeito à leitura de dados de amostragem, quando todos os conversores têm dados novos envia-se o pedido de leitura (Apêndice G.7.2) do *Data Output Register (DOR)* aguardando-se em seguida que se torne possível a recepção dos mesmos. Este funcionamento é demonstrado no diagrama temporal da Figura 3.7 e os tempos respectivos na Tabela 3.1.

Depois da leitura do DOR de cada conversor, é necessário fornecer ao mesmo um impulso no seu pino *nCS* para que este não entre em Modo de Leitura Contínua [ADDS00]. Se o estado do *nCS* é deixado a zero lógico depois de realizada uma leitura ao DOR, a próxima vez que o *nDRDY* tiver um flanco descendente, o conteúdo do Registo de Instrução do ADC será reutilizado e os novos conteúdos do DOR, ou parte deles, serão transmitidos.

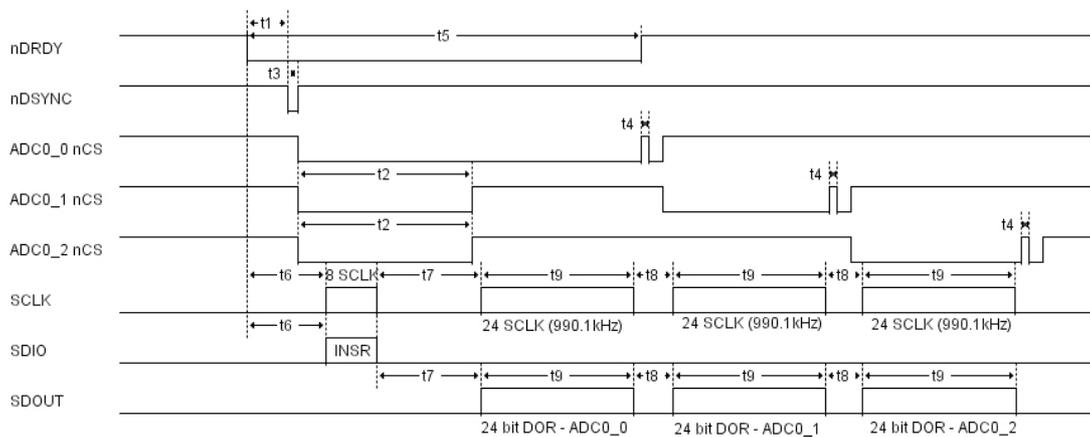


Figura 3.7 – Diagrama temporal de um ciclo de leitura de dados de três conversores

Tabela 3.1 – Tempos para o diagrama temporal da Figura 3.7

Símbolo	Descrição	Tempo(μs)	Comentário
t1	Intervalo de tempo desde o flanco descendente do último <i>nDRDY</i> até ao envio do impulso de sincronização	1	-
t2	Intervalo de tempo de activação do pino <i>nCS</i> de todos os ADC's para envio do <i>INSR</i>	Ind.	(1)
t3	Duração do impulso de sincronização	1.610	-
t4	Duração do impulso do <i>nCS</i> , para não se entrar no Modo de Conversão Contínua	1.360	-
t5	Intervalo de tempo que a linha <i>nDRDY</i> (“OU” Lógico de todos os <i>nDRDY Master</i>) fica a “0” durante a leitura de um dos dados de conversão	Ind	(1)
t6	Intervalo de tempo desde o flanco descendente do <i>nDRDY</i> (“OU” Lógico de todos os <i>nDRDY Master</i>) até ao envio do comando para o <i>INSR</i> (ver G.1)	12.4	-
t7	Intervalo de tempo desde o último flanco descendente de <i>SCLK</i> (envio do <i>INSR</i>) até ao início da leitura do <i>DOR</i> do primeiro ADC	Ind.	(1)
t8	Intervalo de tempo entre o fim da leitura do <i>DOR</i> de um ADC e o início da leitura do <i>DOR</i> do seguinte	6.320	(2)
t9	Intervalo de tempo para leitura de 24 bit do <i>DOR</i>	24.24	(2)

Legenda: (1) - Depende da execução do ciclo principal do programa do utilizador
 (2) - Pode ser maior se ocorrer uma interrupção

3.2 Placa DSPIf: Interface entre MC-XAS3 e DSP (*D.Module.VC33*)

No âmbito deste projecto, fez-se a integração da placa *D.Module.VC33* com a placa MC-XAS3 recorrendo à placa DSPIf desenvolvida no DSOR/ISR. A razão que levou ao desenvolvimento desta placa prende-se com a falta de capacidade de processamento da placa MC-XAS3 (baseada num microcontrolador de 16 bit). Com a placa *D.Module.VC33*, incorporou-se nesta arquitectura de tempo real toda a capacidade computacional de um a sete DSP's de 32 bit, de vírgula flutuante com desempenho até 150 MFLOPS. A filosofia do conceito de interface ilustra-se na Figura 3.8, que se caracteriza por um canal de comunicação implementado com uma memória RAM de dois portos.

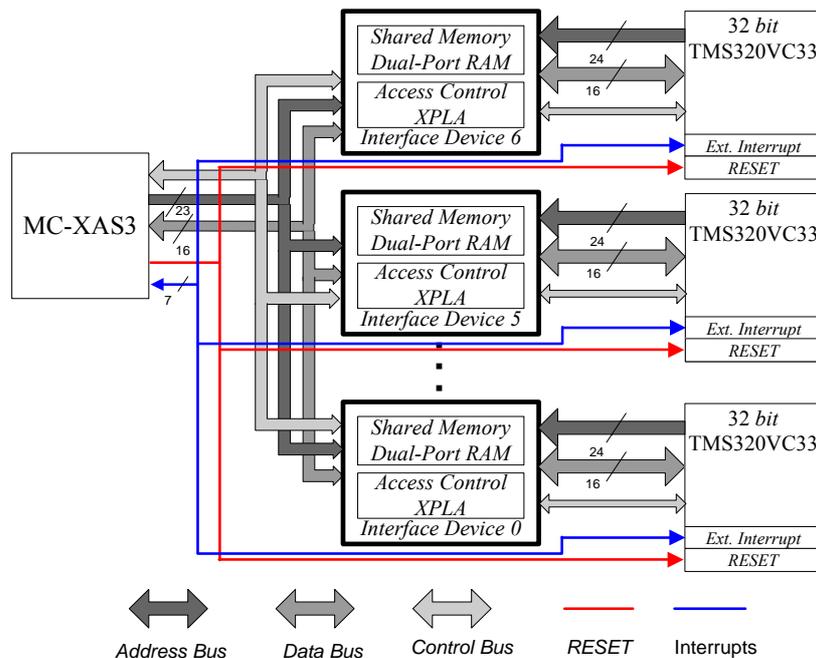


Figura 3.8 – Diagrama conceptual da interface entre a placa MC-XAS3 e o *D.Module.VC33*

Com a integração da placa DSPIf, conseguiu-se criar um canal de comunicação entre dois sistemas com características bastante diferentes, tendo-se recorrido a uma memória RAM de dois portos. Ao contrário da maioria dos sistemas de memória partilhada (que são de acesso multiplexado), este tipo de memória tem dois portos independentes fisicamente disponíveis, com as respectivas linhas de controlo, barramentos de dados e de endereçamento. Na Figura H.1 apresenta-se o diagrama funcional da placa DSPIf, com a memória de dois portos que, a partir deste momento vai ser designada por DP-RAM.

A XPLA ([BCD04]) funciona como ponte na interface entre a placa MC-XAS3 e a DP-RAM, e entre o *D.Module.VC33* e a DP-RAM, gerando para cada tipo de acesso, os sinais de controlo correspondentes. A placa DSPIf oferece um canal de informação entre o *D.Module.VC33* e a placa MC-XAS3 que possibilita do lado da MC-XAS3 uma taxa de transferência de dados mínima de cerca de 1 MB/s, podendo a do DSP ser mais de 10 vezes maior.

3.2.1 Mapas de Memória e Metodologia de Interface

Para uma maior flexibilidade, em termos da utilização da memória da MC-XAS3, dá-se ao utilizador a possibilidade de alocar memória para uma Placa DSPIf em 256 segmentos distintos, até um máximo de 7 placas em simultâneo. Estes 256 segmentos de 32 kB, cada, estão na região alta do espaço de endereçamento da MC-XAS3 (acima do endereço 0x800000), tal como o espaço de endereçamento do Barramento PCMCIA. Para se ter uma ideia visual do mapa de memória consultar Apêndice H.3. Do lado do *D.Module.VC33* não existe a necessidade da criação de vários segmentos, já que este não pode aceder a mais que uma placa DSPIf. O mapa de memória do *D.Module.VC33* encontra-se, também, no Apêndice H.3.

Para além de acessos de leitura e de escrita de dados também é necessário o envio de comandos; portanto, foi necessário dividir cada segmento em dois subsegmentos. O primeiro é o subsegmento de comandos, ou seja, quando se efectua um acesso de leitura ou escrita para este segmento, na realidade faz-se à XPLA um pedido de execução de um comando para o outro dispositivo, não activando a DP-RAM. O segundo subsegmento está efectivamente mapeado na DP-RAM, sendo o resultado dos acessos ao mesmo, a escrita ou leitura de dados na DP-RAM. Tal como se pode verificar através da Tabela 3.2, a placa MC-XAS3 tem ao seu dispor mais comandos, já que, nesta arquitectura, é vista como o dispositivo *Master*. Embora só tenha um subsegmento de comandos, consegue-se ter mais que um comando já que, os acessos são de escrita e o comando é decodificado pela XPLA através da leitura dos pinos de dados *XA D0* a *XA D3* (Figura H.7). Este mesmo mecanismo não pode ser utilizado para o *D.Module.VC33*, porque como se verifica na Figura H.1, as linhas de dados do mesmo não estão conectadas à XPLA.

Tabela 3.2 – Comandos disponíveis para a MC-XAS3 e para o *D.Module.VC33*

Comandos	MC-XAS3	<i>D.Module.VC33</i>
Leitura de dados	Leitura do subsegmento de dados	Leitura do subsegmento de dados
Escrita de dados	Escrita no subsegmento de dados	Escrita no subsegmento de dados
Interrupção no outro dispositivo	Escrita no subsegmento de comandos do dado 0x0002	Leitura do subsegmento de comandos
<i>Interrupt Acknowledge</i>	Escrita no subsegmento de comandos do dado 0x0003	Leitura do subsegmento de dados
<i>Reset</i> ao outro dispositivo	Escrita no subsegmento de comandos do dado 0x0000	-
Ligar o outro dispositivo	Escrita no subsegmento de comandos do dado 0x0001	-
<i>Reset</i> dos circuitos lógicos da <i>XPLA</i>	Escrita no subsegmento de comandos do dado 0x0004	-

As linhas dos dois dispositivos, às quais se recorreu para fazer a interface com a placa DSPIf, são apresentadas no Apêndice H.

3.2.2 Acesso de Leitura à *Dual-Port* RAM

Neste trabalho optou-se pela não utilização das linhas do barramento dos dois dispositivos, que possibilitam a extensão dos ciclos de leitura e escrita na DP-RAM, porque a mesma não dispõe de nenhum mecanismo de sinalização de fim de operação. O ajuste temporal é conseguido à custa da utilização de *waitstates* gerados pelos próprios dispositivos. Todavia, de modo a aumentar as funcionalidades do módulo DSPIf uma nova versão foi concebida incluindo já a manipulação das linhas de *wait* da memória.

- **Placa MC-XAS3**

Os acessos com origem na MC-XAS3 têm a duração superior a 100 ns, não representando assim, qualquer problema para a DP-RAM, não sendo necessária a introdução de *waitstates*.

- ***D.Module.VC33***

Para serem respeitados os tempos de acesso de leitura à DP-RAM, utilizou-se *waitstates* do barramento externo do *D.Module.VC33*. Tal como se pode observar pela Figura 3.9 e tendo em conta que $T_{HI} = 13.33$ ns, a DP-RAM só dispõe de 17.66 ns para apresentar os dados no barramento depois de se ter colocado o endereço válido no barramento de endereços.

Sendo este tempo insuficiente ([BCD04]), foi necessária a introdução de três *waitstates* (a duração de 44.32 ns de dois *waitstates* não é suficiente tal como se pode verificar na Figura 3.9) que fazem com que a duração deste intervalo de espera, até ser feita a leitura dos dados, passe para 57.65 ns, o que é superior à soma dos 35 ns de duração de um ciclo de leitura da DP-RAM com o atraso de propagação da XPLA.

O diagrama simbólico da descodificação dos acessos de leitura do *D.Module.VC33* à DP-RAM apresenta-se na Figura H.5.

Os tempos do diagrama temporal da Figura 3.9 apresentam-se na Tabela 3.3:

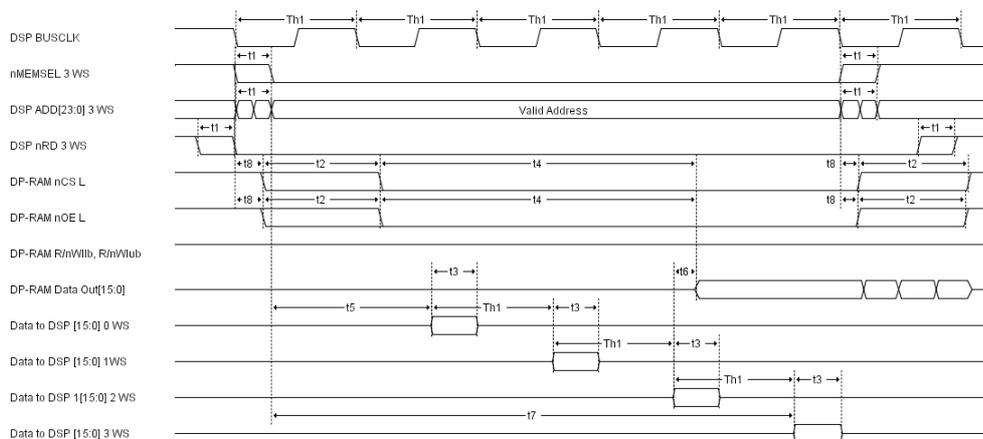


Figura 3.9 – Diagrama temporal de um acesso de leitura de dados do *D.Module.VC33* na DP-RAM

Tabela 3.3 – Tempos para o diagrama temporal da Figura 3.9

Símbolo	Descrição	Tempo	Unidades
Th1	Relógio H1 do TMS320VC33	13.33	ns
t1	Intervalo de incerteza na activação do <i>nMEMSEL</i>	< 4	ns
t2	Incerteza no atraso de propagação da XPLA + t1	0 a 14	ns
t3	Entrada de dados válidos no <i>D.Module.VC33</i>	> 5	ns
t4	Saída de dados válidos da DP-RAM depois de <i>nCS</i> activado	> 35	ns
t5	Entrada de dados válida no <i>D.Module.VC33</i> desde a activação do <i>nMEMSEL</i> (0 <i>waitstates</i>)	17.66	ns
t6	Antecipação da leitura do <i>D.Module.VC33</i> em relação à saída de dados da DP-RAM (2 <i>waitstates</i>)	< 2	ns
t7	t5 + 3 x Th1	57.65	ns
t8	Atraso de propagação mínimo da XPLA	2	ns

3.2.3 Acesso de Escrita à *Dual-Port* RAM

Neste tipo de dispositivos, podem surgir conflitos quando se escrevem dados num porto em simultâneo com uma leitura ou escrita no outro porto. Especificamente para a DP-RAM utilizada, este problema pode ser resolvido fazendo uso da lógica de arbitragem que esta disponibiliza. Neste caso, não se utilizou esta funcionalidade porque os pinos da DP-RAM são do tipo *open-drain*, necessitando de um *pull-up* externo, que nem a XPLA, nem a placa DSPIf dispõem de momento, estando porém contemplada numa nova versão.

A não existência de conflitos é assegurada por *software*, tal como é exposto na Secção 3.2.6

- **Placa MC-XAS3**

Como no caso do acesso de leitura, os acessos de escrita com origem na MC-XAS3 têm a duração de mais de 100 ns, o que não representa qualquer problema para a DP-RAM, não sendo necessária a introdução de *waitstates*.

- ***D.Module.VC33***

Para os acessos de escrita, embora não sejam necessários, decidiu-se por motivos de implementação utilizar três *waitstates*. Apresenta-se na Figura 3.10 o diagrama temporal de um acesso de escrita na DP-RAM e na Tabela 3.4 as respectivas temporizações.

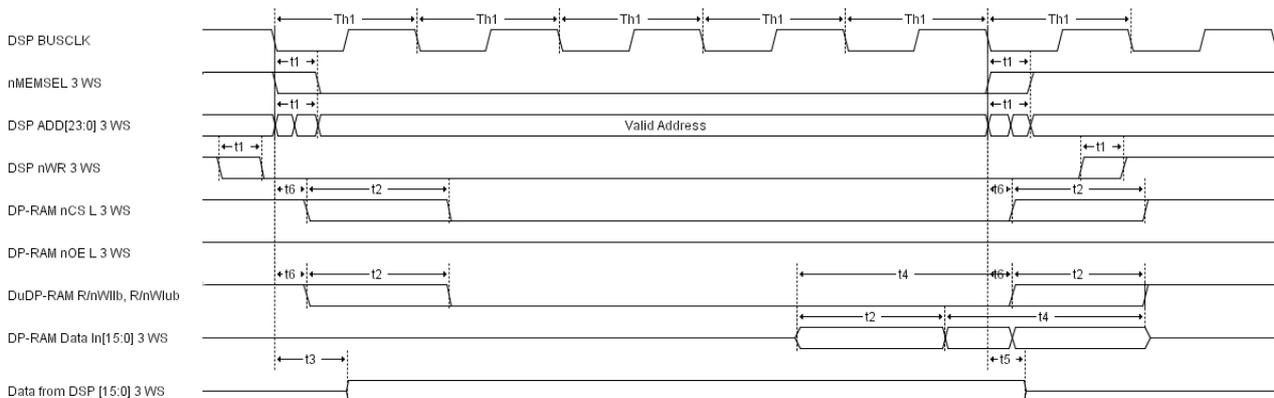


Figura 3.10 – Diagrama temporal de um acesso de escrita de dados do *D.Module.VC33* na DP-RAM

O diagrama simbólico da descodificação dos acessos de escrita do *D.Module.VC33* à DP-RAM apresenta-se na Figura H.5.

Tabela 3.4 – Tempos para o diagrama temporal da Figura 3.10

Simbolo	Descrição	Tempo	Unidades
Th1	Relógio H1 do TMS320VC33	13.33	ns
t1	Intervalo de incerteza na activação do <i>nMEMSEL</i>	< 4	ns
t2	Incerteza no atraso de propagação da XPLA + t1	0 a 14	ns
t3	Saída de dados válidos do <i>D.Module.VC33</i> desde o flanco descendente de DSP BUSCLK	<7	ns
t4	Entrada de dados válidos na DP-RAM antes de <i>nCS</i> desactivado	< 20	ns
t5	Saída de dados válidos do <i>D.Module.VC33</i> desde o último flanco descendente de DSP BUSCLK	0 a 7	ns
t6	Atraso de propagação mínimo da XPLA	2	ns

3.2.4 Interrupções

A única forma de notificar assincronamente o outro dispositivo da ocorrência de um evento é provocar-lhe uma interrupção. É também essencial que o dispositivo interrompido sinalize de alguma forma o conhecimento da interrupção, evitando assim perda de pedidos de interrupção. Este mecanismo é implementado pela XPLA e é descrito de seguida.

- **Placa MC-XAS3 para o *D.Module.VC33***

O circuito lógico assíncrono que gera a interrupção no *D.Module.VC33*, a pedido da MC-XAS3, e que faz o conhecimento da mesma apresenta-se na Figura 3.11:

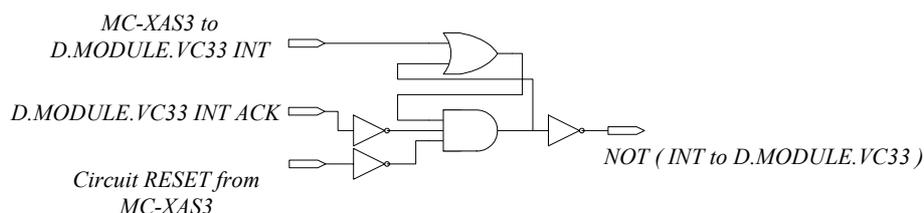


Figura 3.11 – Circuito lógico assíncrono que gera a interrupção no *D.Module.VC33*

Este circuito mantém a linha de interrupção do *D.Module.VC33* no nível lógico zero, até este fazer o conhecimento da interrupção. O *D.Module.VC33* dispõe de um pino (*IACK* pino B22) com o objectivo de notificar dispositivos externos da recepção de uma interrupção. Optou-se por não utilizar esta funcionalidade, porque a duração do impulso de *IACK* é de 13.33 ns, correndo-se assim o risco de não ser detectado pela XPLA.

Neste circuito também é possível reiniciar para um estado conhecido, já que, o estado inicial depois de alimentado é indeterminado. Este comando é da responsabilidade da MC-XAS3. O diagrama temporal deste circuito apresenta-se na Figura H.9

- **D.Module.VC33 para a Placa MC-XAS3**

O circuito apresentado na Figura 3.12 é a análogo ao da Figura 3.11, estando as diferenças na origem do pedido de interrupção, do conhecimento da mesma e do seu destinatário. O diagrama temporal deste circuito apresenta-se na Figura H.8.

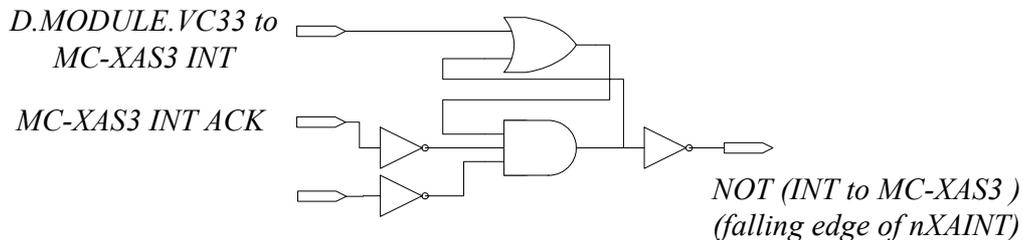


Figura 3.12 – Circuito lógico assíncrono que gera a interrupção na MC-XAS3

3.2.5 Controlo do pino *nRESIN* do D.Module.VC33

Como mencionado, a placa MC-XAS3 desempenha o papel de *Master* na interface com *D.Module.VC33*, assim, tem ao seu dispor a possibilidade de o reiniciar. O pino de *reset* externo do *D.Module.VC33* é o *nRESIN*. O circuito que possibilita esta funcionalidade apresenta-se na Figura 3.13:

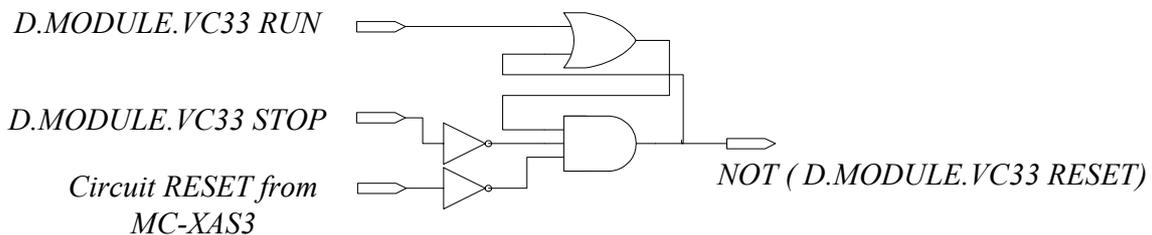


Figura 3.13 – Circuito lógico assíncrono que controla o estado da linha *nRESIN* do *D.Module.VC33*

O diagrama temporal deste circuito apresenta-se na Figura H.10.

3.2.6 Software de Interação entre a placa MC-XAS3, a DSPIf e D.Module.VC33

Foi desenvolvida uma arquitectura dual de *software*, i.e., as funcionalidades do *driver* do lado da MC-XAS3 são análogas às oferecidas no lado do *D.Module.VC33*. O canal de comunicação entre estes dois dispositivos é baseado numa memória partilhada que se encontra na placa de interface DSPIf. Esta memória dispõe fisicamente de dois portos de acesso, o que possibilita acessos de leitura simultâneos com origem em dois dispositivos diferentes.

Existem memórias deste género que disponibilizam funcionalidades acrescidas de troca de mensagens entre dispositivos, o que não é o caso da IDT7133. Assim, foi criado um protocolo baseado na troca de mensagens a que normalmente se atribui a designação de *mailbox*. O diagrama funcional deste protocolo apresenta-se na Figura 3.14:

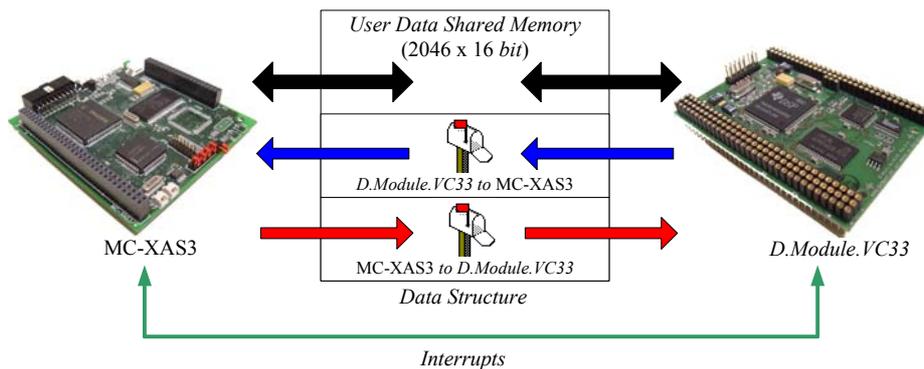


Figura 3.14 – Diagrama funcional do protocolo do tipo *mailbox* implementado com a DP-RAM

Sempre que um dispositivo realiza uma operação (de escrita ou leitura de dados) envia uma mensagem ao outro. O envio desta mensagem é na realidade a escrita dessa mesma operação na posição de memória reservada para tal (*mailbox* de envio), seguida do envio de uma interrupção ao outro dispositivo. O dispositivo que recebe a interrupção irá ler a essa mesma posição de memória (que é a sua *mailbox* de recepção) a operação que o outro acaba de finalizar. Assim, é possível garantir que não ocorrem acessos que gerem conflitos, como é o caso de acessos simultâneos de escrita, ou de escrita e leitura. As mensagens que cada dispositivo pode enviar ao outro para sinalizar um evento apresentam-se na Tabela 3.5.

Criou-se uma estrutura de dados, em ambos os lados, que dá suporte físico a este protocolo bem como à transferência de dados úteis, sendo o seu mapa de memória feito de modo ligeiramente diferente de ambos os lados.

Tabela 3.5 – Mensagens disponíveis a ambos os dispositivos no protocolo implementado do tipo *mailbox*

MC-XAS3	D.Module.VC33
XA2DSP DATA COPY FINISH	DSP2XA DATA COPY FINISH
XA2DSP NEW DATA TO READ	DSP2XA NEW DATA TO READ
XA2DSP NO ACTION	DSP2XA NO ACTION

Estes dois mapas de memória apresentam-se na Tabela 3.6:

Tabela 3.6 – Segmento 0 da MC-XAS3 e segmento do D.Module.VC33

Mapa de memória do D.Module.VC33		Mapa de memória da MC-XAS3	
0xFFFFF	Subsegmento de comandos	0x801FFF	
0xFFFF800		0x801000	
0xFFFF7FF	Subsegmento de memória (2046 × 16 bit de memória física)	0x800FFF	
0xFFFF002		0x800004	
0xFFFF001		Protocolo D.Module.VC33 → MC-XAS3	0x800002
0xFFFF000		Protocolo MC-XAS3 → D.Module.VC33	0x800000

Assim, a MC-XAS3 endereça a memória partilhada num modo contínuo de 16 bit, enquanto o D.Module.VC33 faz num modo de 32 bit, sendo a memória mapeada no *byte* duplo menos significativo (*lower double byte*).

4 Sistema de Navegação com recurso a Filtros de Kalman

4.1 Filtro de Kalman Complementar

A filtragem complementar consiste em utilizar as características da resposta em frequência de dois tipos de sensores de modo a minimizar o erro quadrático médio de estimação. Ou seja, um filtro complementar é um meio de combinar leituras de sensores redundantes em que a informação se encontra em frequências distintas e complementares ([BrHw97]).

O conceito de filtragem complementar baseia-se no problema de combinar duas medidas ruidosas, $s_1(t)$ e $s_2(t)$ do mesmo sinal $s(t)$ que são filtradas através de funções de transferência complementares, i.e., $H_2(s) = 1 - H_1(s)$. Considerando $H_1(s)$ um filtro passa-baixo de primeira ordem obtém-se:

$$\hat{S} = H_1(s)(S + N_1) + H_2(s)(S + N_2) = S + \frac{k}{s+k} N_1 + \frac{s}{s+k} N_2 \quad (4.1)$$

onde S e \hat{S} são, respectivamente, a transformada de *Laplace* do sinal a estimar e da sua estimativa e, N_1 e N_2 são as transformadas de *Laplace* do ruído aditivo às medidas $s_1(t)$ e $s_2(t)$. O parâmetro k determina a frequência de corte e deverá ser ajustado de acordo com as características da largura de banda do ruído dos sensores. O filtro de Kalman complementar recorre a um filtro de atitude e a outro de posição que se baseiam nesta teoria (Apêndice C ou [CaVa03]).

A característica da largura de banda complementar está presente nos sensores utilizados. Para a posição, o sinal do GPS e a dupla integração da força específica são complementares. Para a atitude sucede o mesmo entre as medidas obtidas a partir do sensor magneto-pendular e as medidas da integração das leituras dos giroscópios.

O filtro funciona determinando um sinal de erro entre a estimativa e uma referência calculada directamente a partir de um dos sensores. Um valor escalado deste erro é de seguida realimentado.

Sendo sensores inerciais, quer os giroscópios, quer os acelerómetros fornecem medidas relativas ao referencial inercial expressas no referencial do corpo. Deste modo, o filtro complementar apresenta a estimação da posição e da orientação em relação a um referencial inercial.

A medida dos giroscópios e dos acelerómetros são afectadas por polarizações que têm de ser compensadas introduzindo a dinâmica baseado nas relações cinemáticas de um corpo rígido no modelo do filtro. Ambos os filtros, de posição e de atitude, são implementados com esta abordagem.

O sistema de navegação para além de estimar a posição e a atitude, estima as polarizações dos acelerómetros e giroscópios, como apresenta a Figura 4.1. No entanto, apenas a polarização estimada dos giroscópios, $\hat{\Gamma}_{giros}$, é realimentada e desta forma, utilizada para corrigir as leituras dos mesmos. Isto porque a dupla realimentação não garante que o único ponto de equilíbrio do sistema seja o desejado. Este facto foi comprovado experimentalmente.

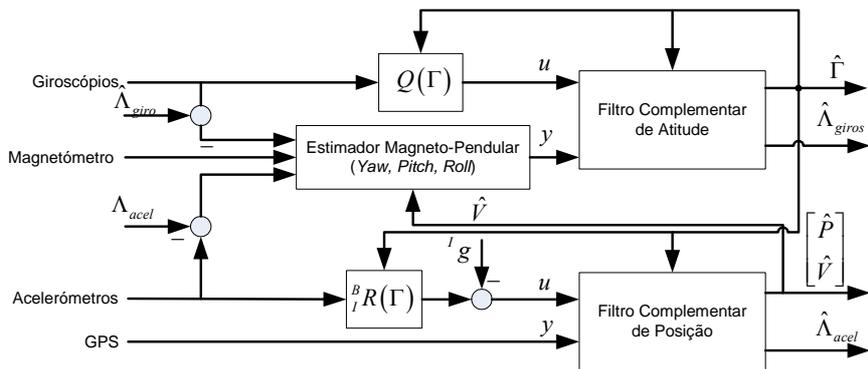


Figura 4.1 – Diagrama do Sistema de Navegação Global com filtros complementares

4.2 Filtro de Kalman Estendido

A designação de “filtro” de Kalman pode causar alguma confusão, sendo esta denominação usada apenas por razões históricas. O filtro de Kalman é apenas um algoritmo que processa observações discretas (entrada) com o objectivo de calcular estimativas óptimas (saída) [BrHw97].

O algoritmo aqui apresentado consiste numa evolução do filtro de Kalman estendido apresentado em [CaVa03] tendo em vista uma implementação real do mesmo descrita em 5.1. O filtro de Kalman estendido é projectado no espaço de erro segundo uma configuração *direct feedback*. Nesta topologia, são estimados os erros das grandezas, sendo introduzidas no filtro medidas obtidas através da diferença entre os valores do bloco INS e dos sensores auxiliares. A Figura 4.2 que se segue pretende ilustrar o funcionamento do filtro:

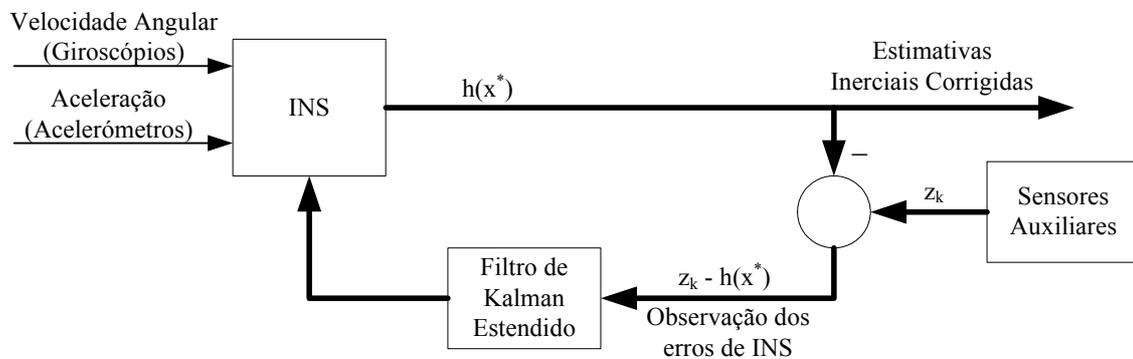


Figura 4.2 – Sistema de Navegação em configuração *feedback*

As estimativas do filtro são utilizadas para corrigir internamente o bloco de INS, impedindo o erro deste de crescer ilimitadamente. Esta topologia, implica que sejam determinadas as dinâmicas dos erros do INS, o que pode ser consultado em [CaVa03], e donde se transcreve aqui a expressão final.

$$\begin{cases} \delta \dot{P} = \delta V \\ \delta \dot{V} = - \left[{}^E \tilde{a} \times \right] \delta \Psi + {}^E_B R \delta^B a \\ \delta \dot{\Psi} = {}^E_B R \delta^B \left({}^I \omega_B \right) \end{cases} \quad (4.2)$$

onde:

- ${}^E \tilde{a} = {}^E R {}^B \tilde{a}$ é a aceleração estimada pelo INS de {B} em relação a {I} expressa em {E}) e

$$a \times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}, \text{ matriz de produto externo [Silv95]}$$
- $\delta P = [\delta p_N \quad \delta p_E \quad \delta p_D]^T = \delta {}^E P_{BORG}$ (erro de posição da origem de {B} em {E} expressa em {E})
- $\delta V = [\delta v_N \quad \delta v_E \quad \delta v_D]^T = \delta {}^E V_{BORG}$ (erro de velocidade de {B} em {E} expressa em {E})
- $\delta \Psi = [\delta \lambda_N \quad \delta \lambda_E \quad \delta \lambda_D]^T = \delta {}^E ({}^B \lambda_{\tilde{B}})$ (vector de rotação do erro de atitude de {B} expressa em {E})

Estas equações de erro resultam de expressões não lineares, pelo que, quando implementadas numa configuração *direct feedback* motivam o recurso a um filtro de Kalman estendido.

O modelo de estado presente foi obtido pela linearização dos erros de posição, velocidade e orientação do veículo. Apenas é observada a posição do veículo através de GPS, cujo ruído é modelado por um processo de Gauss-Markov, introduzindo três variáveis de estado adicionais ($x_{1GPS}, x_{2GPS}, x_{3GPS}$). O sistema é dado por:

$$\begin{cases} x_{k+1} = \phi_k x_k + w_k \\ z_k = H_k x_k + v_k \end{cases} \quad (4.3)$$

onde, de [CaVa03]:

$$x_k = [\delta P \quad \delta V \quad \delta \Psi \quad x_{1GPS} \quad x_{2GPS} \quad x_{3GPS}]^T$$

$$\phi_k = \begin{bmatrix} \phi'_k & 0 \\ 0 & I_{3 \times 3} e^{-\beta T} \end{bmatrix}, \phi'_k = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} T & -[{}^E \tilde{a} \times] \frac{T^2}{2} \\ 0 & I_{3 \times 3} & -[{}^E \tilde{a} \times] T \\ 0 & 0 & I_{3 \times 3} \end{bmatrix} \quad (4.4)$$

$$H_k = [I_{3 \times 3} \quad 0 \quad 0 \quad I_{3 \times 3}]$$

As matrizes de covariância dos ruídos de estado e dos ruídos de observação são dadas por

$$Q_k = E[w_k w_k^T] = \begin{bmatrix} Q'_k & 0 \\ 0 & I_{3 \times 3} \sigma^2 (1 - e^{-2\beta T}) \end{bmatrix}, \quad R_k = E[v_k v_k^T] = \begin{bmatrix} I_{3 \times 3} & A_p \\ & T \end{bmatrix} \quad (4.5)$$

onde

$$Q'_k = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \phi'(t_{k+1}, \xi) G'(\xi) E[w_c(\xi) w_c^T(\eta)] G'^T(\eta) \phi'^T(t_{k+1}, \eta) d\xi d\eta \quad (4.6)$$

sendo w_c o ruído de estado contínuo, $G'(\tau)$ a matriz que introduz w_c no modelo contínuo e $\phi'(t_{k+1}, \tau)$ a matriz de transição entre os instantes τ e t_{k+1} .

O cálculo da matriz Q'_k depende da formulação da matriz $\phi'(t_{k+1}, \tau)$, pelo que nem sempre pode ser obtida pelo desenvolvimento directo do integral. Frequentemente recorre-se a aproximações de primeira ordem [Atha01] ou de segunda ordem [Mayb79], sendo esta última dada por:

$$Q'_k \approx \frac{1}{2} [\phi'_k G'_k Q'_c G'^T_k \phi'^T_k + G'_k Q'_c G'^T_k] T \quad (4.7)$$

onde, para o modelo corrente

$$G'_k = \begin{bmatrix} I_{3 \times 3} & 0 & 0 \\ 0 & {}^E R_k & 0 \\ 0 & 0 & {}^E R_k \end{bmatrix}, Q'_c = E[w_c(\xi)w_c^T(\eta)] = \begin{bmatrix} \sigma_{u_{vel}}^2 & 0 & 0 \\ 0 & \sigma_{acel}^2 & 0 \\ 0 & 0 & \sigma_{giro}^2 \end{bmatrix}$$

As equações de actualização das estimativas de estado são dadas por:

$$\begin{cases} K_k = (P_k^- H_k^T) [H_k P_k^- H_k^T + R_k]^{-1} \\ P_k = (I - K_k H_k) P_k^- \\ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \\ \hat{x}_{k+1}^- = \phi_k \hat{x}_k \\ P_{k+1}^- = \phi_k P_k \phi_k^T + Q_k \end{cases} \quad (4.8)$$

Devido à não observabilidade do sistema associada a um grau de liberdade na atitude, pretende-se introduzir uma medida de atitude de modo a que o sistema seja localmente observável. A introdução da nova medida deve ter em consideração que as leituras do estimador magneto-pendular (EMP) são válidas especialmente para baixas frequências e que se pretende implementar métodos de estabilização algébrica do algoritmo de Kalman como a factorização UD da matriz de covariância dos erros.

4.2.1 Modelação de uma Medida do Vector de Rotação do Erro de Atitude

A medida auxiliar de atitude é válida apenas quando a resultante das acelerações tangenciais é desprezável¹. Devido à existência de atrito, o EMP só está sujeito a essa resultante durante períodos de tempo limitados, pelo que se pode considerar que as medidas são válidas para frequências muito baixas.

De modo a valorizar as componentes de baixa frequência da saída do EMP, modela-se um ruído artificial de alta-frequência nas leituras de atitude. Este ruído apenas se encontra modelado no filtro de Kalman, não tendo presença real nas medidas. Assim, o filtro dará maior peso às componentes de baixa-frequência das leituras do EMP, atribuindo as perturbações de alta-frequência ao ruído.

¹ A componente centrípeta da aceleração é estimada e removida.

Na Figura 4.3 encontra-se ilustrado o processo resultante visto pelo filtro de Kalman, onde se constata que o ruído de alta-frequência é, na verdade, formado pela filtragem passa-banda de um ruído branco. Este artifício pretende evitar uma correlação entre o ruído de estado e o ruído de observação que decorreria da modelação de um filtro passa-alto cuja função de transferência é própria mas não estritamente própria². Esta correlação, embora tratável para a forma clássica do filtro de Kalman [BrHw97], afectaria a expressão de actualização da matriz P e dificultaria a aplicação do método de decomposição UD.

A colocação da frequência superior do filtro passa-banda perto de metade da frequência de Nyquist resulta numa aproximação eficaz do filtro passa-banda a um filtro passa-alto dado que o filtro é funcional apenas para sistemas com frequências de funcionamentos até metade da frequência de amostragem. Adicionalmente, é introduzido um ruído de observação que afectará a confiança em geral do filtro nas leituras do EMP.

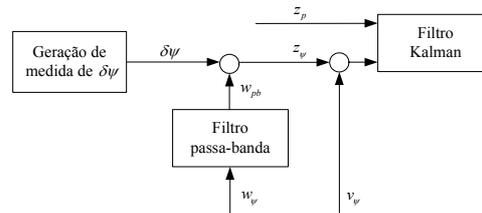


Figura 4.3 – Introdução de ruído passa-banda artificial

4.2.1.1 Modelo de Estado do Filtro Passa-Banda

A introdução da dinâmica do filtro passa-banda escalar no modelo de estado final passa pelo cálculo da função de transferência deste, sendo dada por:

$$w_{pb} = \frac{p_a s}{(s + p_a)(s + p_b)} w_{\psi} \quad (4.9)$$

onde p_b é a frequência inferior da banda de passagem e p_a é a frequência superior da banda de passagem, onde o filtro apresenta ganho unitário.

O modelo de estado correspondente pode ser calculado utilizando as variáveis de estado:

$$x_1 = \frac{p_a w_{\psi}}{(s + p_a)(s + p_b)}, \quad x_2 = s x_1, \quad w_{pb} = x_2 \quad (4.10)$$

o que resulta no modelo de estado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = F_{pb} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + G_{pb} w_{\psi}, \quad w_{pb} = H_{pb} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.11)$$

² Uma função de transferência é própria quando o grau do polinómio numerador é menor ou igual que o grau do polinómio denominador. Caso seja menor, diz-se estritamente própria. [Ribe02]

onde

$$F_{pb} = \begin{bmatrix} 0 & 1 \\ -p_a p_b & -(p_a + p_b) \end{bmatrix}, G_{pb} = \begin{bmatrix} 0 \\ p_a \end{bmatrix}, H_{pb} = [0 \quad 1] \quad (4.12)$$

A observação de cada uma das componentes do vector de erro de atitude é dada por:

$$z_{i,\Psi} = \delta\Psi_i + H_{pb} \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix} + v_i \quad (4.13)$$

em que i é o índice da componente do vector de observações e do vector de erro de atitude e $[x_{i,1} \quad x_{i,2}]^T$ são as variáveis do modelo de estado de (4.11) correspondentes à componente i do vector de observação.

4.2.1.2 Discretização

Para que o filtro possa ser implementado digitalmente, é necessário proceder à sua discretização para a forma:

$$\begin{cases} x_{k+1} = \phi_k x_k + w_k \\ z_k = H_k x_k + v_k \end{cases} \quad (4.14)$$

A matriz da dinâmica ϕ_k é dada por:

$$\phi_k = e^{F_{pbk}T} \quad (4.15)$$

O cálculo da matriz da dinâmica pode ser realizado recorrendo à Transformada de Laplace:

$$e^{F_{pbk}T} = \mathcal{L}^{-1} \left[\left(sI_{2 \times 2} - F_{pbk} \right)^{-1} \right] \Big|_{t=T} = \frac{1}{p_a - p_b} \begin{bmatrix} p_a e^{-p_b T} - p_b e^{-p_a T} & e^{-p_b T} - e^{-p_a T} \\ p_a p_b (e^{-p_a T} - e^{-p_b T}) & p_a e^{-p_a T} - p_b e^{-p_b T} \end{bmatrix} \quad (4.16)$$

Como a matriz F_{pbk} se encontra na forma companheira e, devido aos pólos serem distintos, tem estrutura simples, uma expressão alternativa para cálculo de (4.15) é [Ribe02]:

$$e^{F_k T} = \begin{bmatrix} 1 & 1 \\ -p_a & -p_b \end{bmatrix} \begin{bmatrix} e^{-p_a T} & 0 \\ 0 & e^{-p_b T} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -p_a & -p_b \end{bmatrix}^{-1} \quad (4.17)$$

A matriz Q pode ser calculada através da expressão (4.7), onde:

$$Q_c = \sigma_{w_\Psi}^2, \quad G_k = \begin{bmatrix} 0 \\ p_a \end{bmatrix} \quad (4.18)$$

e $\sigma_{w_\Psi}^2$ é a variância do ruído artificial w_Ψ .

A matriz da covariância dos ruídos de observação é dada por:

$$R_k = E[v_{i_k} v_{i_k}^T] = \frac{1}{T^2} \int_T \int_T E[v_i(\xi) v_i^T(\eta)] d\xi d\eta = \frac{\sigma_{v_\Psi}^2}{T} \quad (4.19)$$

e $\sigma_{v_\Psi}^2$ é a variância do ruído de observação v_Ψ .

4.2.2 Modelo Final

Como modelo final e em jeito de resumo apresentam-se as equações do filtro de Kalman estendido:

$$\begin{cases} K_k = (P_k^- H_k^T) [H_k P_k^- H_k^T + R_k]^{-1} \\ P_k = (I - K_k H_k) P_k^- \\ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \\ \hat{x}_{k+1}^- = \phi_k \hat{x}_k \\ P_{k+1}^- = \phi_k P_k \phi_k^T + Q_k \end{cases} \quad (4.20)$$

O modelo de estado do sistema em causa:

$$\begin{cases} x_{k+1} = \phi_k x_k + w_k \\ z_k = H_k x_k + v_k \end{cases} \quad (4.21)$$

onde, o vector de estado é:

$$x_k = [\delta P \quad \delta V \quad \delta \Psi \quad x_{1GPS} \quad x_{2GPS} \quad x_{3GPS} \mid x_{1,1\Psi} \quad x_{1,2\Psi} \quad x_{2,1\Psi} \quad x_{2,2\Psi} \quad x_{3,1\Psi} \quad x_{3,2\Psi}]^T \quad (4.22)$$

a matriz da Dinâmica é dada pela expressão:

$$\phi_k = \begin{bmatrix} \phi'_k & 0 & 0 & 0 & 0 \\ 0 & I_{3 \times 3} e^{-\beta T} & 0 & 0 & 0 \\ 0 & 0 & \phi''_k & 0 & 0 \\ 0 & 0 & 0 & \phi''_k & 0 \\ 0 & 0 & 0 & 0 & \phi''_k \end{bmatrix}, \quad \phi'_k = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} T & -[{}^E \tilde{a} \times] \frac{T^2}{2} \\ 0 & I_{3 \times 3} & -[{}^E \tilde{a} \times] T \\ 0 & 0 & I_{3 \times 3} \end{bmatrix} \quad (4.23)$$

$$e \quad \phi''_k = \frac{1}{p_a - p_b} \begin{bmatrix} p_a e^{-p_b T} - p_b e^{-p_a T} & e^{-p_b T} - e^{-p_a T} \\ p_a p_b (e^{-p_a T} - e^{-p_b T}) & p_a e^{-p_a T} - p_b e^{-p_b T} \end{bmatrix}$$

a matriz de Observação:

$$H_k = \left[\begin{array}{ccc|ccc} I_{3 \times 3} & 0 & 0 & I_{3 \times 3} & & \\ \hline & & & & 0 & \\ 0 & 0 & I_{3 \times 3} & 0 & \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \end{array} \right] \quad (4.24)$$

a matriz de Covariância dos Ruídos de Estado:

$$Q_k = E[w_k w_k^T] = \left[\begin{array}{c|cc} Q'_k & 0 & 0 \ 0 \ 0 \\ \hline 0 & I_{3 \times 3} \sigma^2 (1 - e^{-2\beta T}) & 0 \ 0 \ 0 \\ \hline 0 & 0 & Q''_k \ 0 \ 0 \\ 0 & 0 & 0 \ Q''_k \ 0 \\ 0 & 0 & 0 \ 0 \ Q''_k \end{array} \right] \quad (4.25)$$

$$Q'_k = \frac{1}{2} [\phi'_k G'_k Q'_c G_k{}^T + G'_k Q'_c G_k{}^T] T$$

$$Q''_k = \frac{1}{2} [\phi''_k G''_k Q''_c G_k{}^T + G''_k Q''_c G_k{}^T] T$$

onde

$$G'_k = \begin{bmatrix} I_{3 \times 3} & 0 & 0 \\ 0 & {}^E R_k & 0 \\ 0 & 0 & {}^E R_k \end{bmatrix}, G''_k = \begin{bmatrix} 0 \\ p_a \end{bmatrix} \quad e \quad Q'_c = \begin{bmatrix} \sigma_{u_{vel}}^2 & 0 & 0 \\ 0 & \sigma_{acel}^2 & 0 \\ 0 & 0 & \sigma_{giro}^2 \end{bmatrix}, Q''_c = \sigma_{w_p}^2 \quad (4.26)$$

a matriz de Covariância dos Ruídos de Observação:

$$R_k = \begin{bmatrix} I_{3 \times 3} \frac{A_p}{T} & 0 \\ 0 & I_{3 \times 3} \frac{\sigma_{v_{\psi}}^2}{T} \end{bmatrix} \quad (4.27)$$

Obtidas as equações, pretende-se com o esquema da Figura 5.1, tornar mais compreensível o funcionamento da solução do filtro de Kalman estendido implementada.

A topologia implementada envolve um INS a operar em conjunto com o filtro de Kalman estendido e um bloco de correcção das estimativas, conforme a Figura 4.4 procura ilustrar. O INS calcula as estimativas de posição, velocidade e atitude a partir das leituras (aceleração e velocidade angular) dos sensores, sendo depois corrigidas pelo bloco de correcção que tem em conta os erros estimados pelo filtro de Kalman estendido. As estimativas corrigidas são realimentadas para o INS, para que este não comece a divergir, devido ao ruído e às polarizações existentes nas medidas dos sensores.

De acordo com a Figura 4.2, a configuração *direct feedback* corrige os erros de INS de modo a manter o ponto de linearização do filtro.

A partir das variáveis de estado $[\delta P_k \ \delta V_k \ \delta \Psi_k]^T$ e da definição de erro, a correcção da estimação de posição e de velocidade calculadas pelo INS, é dada por:

$$x_k = \tilde{x}_k - \delta P_k \quad \text{e} \quad v_k = \tilde{v}_k - \delta V_k \quad (4.28)$$

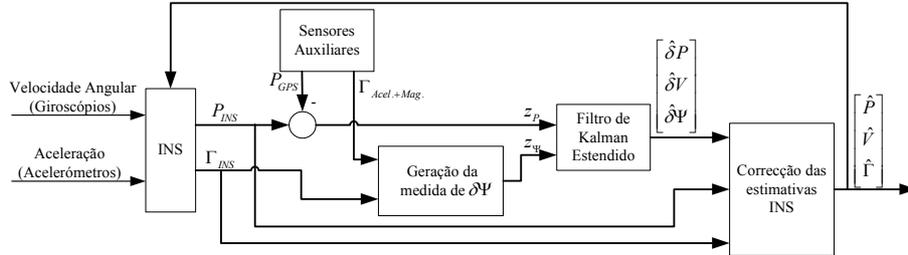


Figura 4.4 – Novo diagrama conceptual do sistema de navegação

onde \tilde{x}_k é a posição estimada pelo INS e x_k é o novo valor de posição depois de corrigida. A relação é análoga para \tilde{v}_k e v_k sendo k o índice do instante de execução t_k .

A correcção de atitude recorre à equação do erro da matriz de rotação:

$${}^E_B R = R_{\delta \Psi}^{-1} {}^E_B \tilde{R}, \quad R_{\delta \Psi} = [I_{3 \times 3} + [\delta \Psi \times]] \quad (4.29)$$

Das propriedades da inversa da matriz de rotação (A.3) e do produto externo (consultar [CaVa03]), tem-se:

$$\begin{aligned} R_{\delta \Psi}^{-1} = R_{\delta \Psi}^T &\Rightarrow R_{\delta \Psi}^{-1} = [I_{3 \times 3} + [\delta \Psi \times]]^T = [I_{3 \times 3} + [\delta \Psi \times]^T] = [I_{3 \times 3} - [\delta \Psi \times]] = R_{-\delta \Psi} \\ &\Rightarrow R_{\delta \Psi}^{-1} = R_{-\delta \Psi} \end{aligned} \quad (4.30)$$

Reescrevendo (4.29) no espaço dos quaterniões, obtém-se a expressão da correcção da orientação:

$${}^E_B q_k = \delta q_k {}^E_B \tilde{q}_k \Rightarrow {}^E_B q_k = \Xi(\delta q_k) {}^E_B \tilde{q}_k = \Upsilon({}^E_B \tilde{q}_k) \delta q_k \quad (4.31)$$

onde ${}^E_B \tilde{q}_k$ é a orientação estimada pelo INS, ${}^E_B q_k$ é o novo valor de orientação no INS e o vector $\delta q = (\delta q_s, \delta q_v)$ equivale à rotação $R_{-\delta \Psi}$ de (4.30). Recorrendo à definição (A.11) do vector de rotação e à sua relação com os quaterniões (A.15):

$$\begin{cases} \delta q_v = -\sin\left(\frac{\Lambda}{2}\right) \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \\ \delta q_s = \cos\left(\frac{\Lambda}{2}\right) \end{cases} \quad (4.32)$$

Através de (4.28) e de (4.31), a remoção dos erros estimados no INS anula as variáveis de estado $[\delta P_k \ \delta V_k \ \delta \Psi_k]^T$.

5 Implementação dos Sistemas de Navegação na Arquitectura de *Hardware*

Depois de validados os algoritmos num ambiente de simulação (consultar Apêndice F), procedeu-se à implementação dos mesmos na arquitectura de *Hardware*. Este passo não foi dado de uma só vez. Primeiro executou-se o algoritmo de navegação no DSP utilizando os valores dos sensores, em grandezas físicas, directamente fornecidos pelo simulador. De seguida, procedeu-se à implementação na arquitectura de *hardware* desenvolvida.

Nas secções seguintes explicam-se estas duas implementações.

5.1 Questões de Implementação do Filtro de Kalman Estendido

5.1.1 Factorização UD

Sendo o filtro de Kalman discreto recursivo, os ciclos poderão, em teoria, correr infinitamente. No entanto, existem limites práticos e sob certas condições, podem existir problemas de divergência. É um facto bem conhecido que o filtro de Kalman na sua formulação original é sensível a imprecisões numéricas, o que pode ser uma causa de instabilidade para o filtro de Kalman estendido.

Uma das causas destes problemas prende-se com erros de arredondamento, o que acontece em qualquer procedimento numérico e pode tornar-se um problema quando o número de iterações é elevado. Algumas técnicas conhecidas para contornar/prevenir este problema consistem em usar aritmética de precisão, propagar a matriz P em vários e pequenos passos entre medições, evitar processos determinísticos na modelação do filtro (e.g. uma constante aleatória). Tal situação pode levar a que a matriz P deixe de ser positiva definida e tem como solução a adição de pequenas quantidades aos termos da diagonal principal da matriz Q . Esta solução leva a um grau de suboptimabilidade, o que não sendo desejável, é no entanto melhor do que ter o filtro a divergir. Uma outra solução tem a ver com o facto de a matriz P ser simétrica. Se a dada altura a matriz da covariância perder esta característica, sabe-se que tal se deve a erros de arredondamento e, se nada for feito para o impedir, a assimetria pode aumentar cada vez mais. Como solução, pode optar-se por efectuar todos os cálculos apenas com a triangular superior, ou inferior, da matriz P .

A existência de uma grande incerteza nas estimações iniciais, pode também conduzir a problemas numéricos. Por exemplo, num problema de navegação, se iniciarmos a matriz P_0^- com valores muito elevados na diagonal principal e se, existir uma medição muito precisa para $t = 0$, a matriz P transita de um valor muito elevado para um valor próximo de zero numa iteração. A alternativa consiste em impor os elementos da matriz P_0^- artificialmente mais pequenos e estar consciente que nas primeiras iterações o filtro terá um desempenho subóptimo.

Este tipo de problemas não merece tanta preocupação nos dias de hoje, como acontecia no início da década de 60, devido ao enorme progresso da tecnologia computacional. Ainda assim, existem aplicações onde os erros de arredondamento se podem tornar num problema, pelo que, neste trabalho foi implementado um algoritmo de factorização *UD*. Este algoritmo é indicado para aplicações em que o filtro opera por longos períodos e onde a estabilidade numérica merece especial atenção. Com este algoritmo, propagam-se os factores de matriz P , ao invés de propagar a própria matriz. Por isto, este algoritmo insere-se na classe de algoritmos de “raiz quadrada” – tipo de algoritmos em que se propaga

a \sqrt{P} , em vez de propagar a matriz P . Esta forma do filtro de Kalman é mais difícil de implementar, mas tem um comportamento numérico bastante melhor.

Este algoritmo assenta na base em que, tendo uma matriz P simétrica, positiva definida, consegue-se sempre decompô-la na sua forma factorizada:

$$P = UDU^T \quad (5.1)$$

onde U é uma matriz triangular superior com diagonal unitária e D é uma matriz diagonal.

Partindo da estimativa *à priori*:

$$P^- = U^- D^- U^{-T} \quad (5.2)$$

os ganhos de Kalman são calculados da maneira tradicional, substituindo P^- pelos seus factores U^- e D^- .

A principal alteração em relação ao algoritmo usual tem a ver com a actualização da matriz P , que neste caso não é directamente actualizada, mas sim os seus factores U e D . Desta forma, garante-se que a estrutura definida positiva da matriz da covariância é preservada. Para tal, é utilizado um algoritmo iterativo em que as observações são processadas uma de cada vez de forma série, o que se pode fazer dado que as observações no instante t_k são não correlacionadas e, por isso a matriz R tem apenas elementos não nulos na diagonal. De referir que, com este algoritmo deixa de ser necessário o cálculo da inversa da matriz S , o que poderia introduzir dificuldades acrescidas.

De (4.20), pode-se escrever que:

$$P_k = P_k^- - K_k H_k P_k^- = P_k^- - P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} H_k P_k^- \quad (5.3)$$

o que em termos dos factores da matriz P fica:

$$U_k D_k U_k^T = U_k^- D_k^- (U_k^-)^T - U_k^- D_k^- (U_k^-)^T H_k^T [H_k U_k^- D_k^- (U_k^-)^T H_k^T + R_k]^{-1} H_k U_k^- D_k^- (U_k^-)^T \quad (5.4)$$

definindo $v = (U_k^-)^T H_k^T$ como um vector de dimensão n , onde n é a dimensão do vector de estado e omitindo o índice k para facilitar a leitura, pode-se reescrever (5.4) como:

$$U^- D^- (U^-)^T = U^- \underbrace{\left(D^- - D^- v (v^T D^- v + R)^{-1} v^T D^- \right)}_{(1)} (U^-)^T \quad (5.5)$$

Pode-se escrever (1) da expressão anterior, como:

$$D^- - D^- v (v^T D^- v + R)^{-1} v^T D^- = B D^+ B \quad (5.6)$$

onde D^+ é o factor D *a posteriori* de P , pelo que:

$$U^+ D^+ (U^+)^T = U^- (B D^+ B) (U^+)^T = (U^- B) D^+ (U^- B)^T \quad (5.7)$$

donde resulta que o factor U *a posteriori* é dado por:

$$U^+ = U^- B \quad (5.8)$$

de [Bier77]:

$$B_{ij} = -\frac{D_{jj}^- v_i v_j}{R + \sum_{k=1}^{j-1} v_k^2 D_{kk}^-}, \quad D_{jj}^+ = D_{jj}^- \left[\frac{R + \sum_{k=1}^{j-1} v_k^2 D_{kk}^-}{R + \sum_{k=1}^j v_k^2 D_{kk}^-} \right] \quad (5.9)$$

De mencionar que a matriz H_t nas expressões anteriores corresponde a h_0, h_1, \dots , do mesmo modo que a R corresponde a r_0, r_1, \dots , tal como mostra a expressão (5.10) (onde t substitui o índice k até agora utilizado por conflitos de notação na Figura 5.1):

$$H_t = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \end{bmatrix}, \quad R_t = \begin{bmatrix} r_0 & 0 & 0 \\ 0 & r_1 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \quad (5.10)$$

A matriz R_t é, no mínimo, diagonal por blocos. Fisicamente, isto significa que as observações disponíveis no instante t podem ser agrupadas de modo a que os erros entre os blocos r_i , com $i=0, \dots, p$ (sendo p o número de observações), são não correlacionados. Este é geralmente o caso quando medidas redundantes provêm de instrumentos diferentes.

Neste algoritmo, a actualização dos factores da matriz P são calculados iterativamente, pelo que, por exemplo, as matrizes U e D *a posteriori* correspondentes ao bloco r_0 serão as matrizes *a priori* para os cálculos relativos ao bloco r_1 . A estimação final (depois de feitos os cálculos para cada um dos blocos) é assim igual à que seria obtida pelo método tradicional.

A Figura 5.1 mostra o funcionamento do filtro de Kalman estendido implementado:

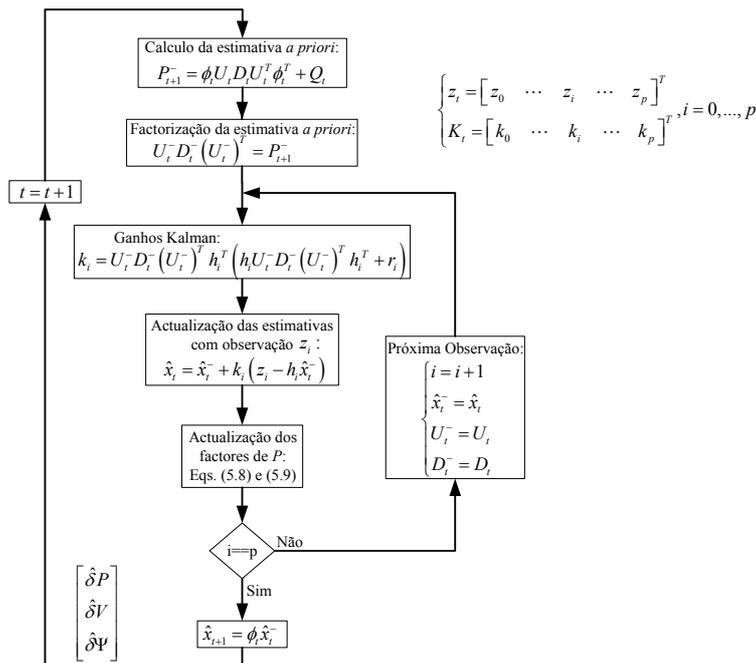


Figura 5.1 – Loop do filtro de Kalman

De referir que o matriz de ganhos Kalman, K_t , é aquele que se obteria se as observações fossem processadas simultaneamente.

5.1.2 Implementação em Tempo-Real e Complexidade do Algoritmo

Uma das características mais apreciadas no filtro de Kalman consiste na sua natureza recursiva e no baixo uso de memória. Este aspecto torna o filtro numa ferramenta de processamento muito útil em aplicações de tempo real. Nos casos em que os cálculos relativos ao filtro de Kalman demorem mais do que um período de amostragem a ser efectuados, uma de duas coisas pode acontecer: o filtro de Kalman processa apenas um número menor de medidas e ignora as restantes, ou o filtro insiste em processar a totalidade dos dados e gradualmente, as suas saídas estarão cada vez mais atrasadas. Na primeira alternativa, o filtro continua ainda a ser de tempo real, embora seja uma solução não óptima, enquanto que, na segunda, o filtro deixa de ser considerado de tempo real.

Nas aplicações reais, é impossível esperar que o processamento das saídas seja completamente instantâneo aos tempos de amostragem. Existe sempre um atraso finito, designado por tempo de latência, que está associado com os tempos gastos nas comunicações e com o tempo do processamento do filtro de Kalman. Este tempo poderá tornar-se uma fracção importante do período de amostragem, o que terá que ser tido em conta em aplicações de controlo. Uma solução para contornar este problema, consiste em projectar a solução do filtro de Kalman para um instante no futuro. Assim, o resultado apresentado calculado pelo filtro em cada instante não teria atraso. No entanto, isto leva a que a covariância do erro não seja a menor possível, porque existe um erro adicional de predição. Assim, neste trabalho optou-se por minimizar o tempo de latência, o que foi feito tendo em conta teorias para a análise da complexidade de algoritmos.

A análise da complexidade de algoritmos preocupa-se com a questão de, para que problemas de decisão existem algoritmos eficientes, o que leva a definir o significado de “algoritmo eficiente” e saber que recursos se querem usar de modo eficiente. Os dois recursos, ou medidas de complexidade, de interesse correspondem ao tempo de computação, i.e. o número de passos que um algoritmo

necessita (no pior dos casos) para devolver um resultado e ao espaço de memória requerido para executar o algoritmo.

O tempo computacional de um algoritmo é proporcional ao número de operações algébricas que este realiza. Assim, define-se $T(n, p) = M(n, p) + A(n, p)$ como o número de operações em função do número de variáveis de estado, n , e do número de observações, p :

$$\begin{cases} M(n, p) = 5n^3 + 2\sum_{i=1}^n (n+1-i) + \left[7n^3 + 3n^2 + 6n + 9\sum_{i=1}^{n-1} 2i \right] p \\ A(n, p) = 5n^2(n-1) + n + \sum_{i=1}^n (n+1-i) + \left[7n^2(n-1) + 3n(n-1) + 4n + 3\sum_{i=1}^{n-1} 2i \right] p \end{cases} \quad (5.11)$$

onde $M(n, p)$ e $A(n, p)$ correspondem ao número de multiplicações e de adições para o algoritmo descrito na Figura 5.1.

Embora a expressão (5.11) de uma indicação do tempo computacional que o algoritmo necessita, esta não tem em conta a implementação concreta do filtro de Kalman estendido, onde se utilizam rotinas óptimas de multiplicação, como se descreveu na secção anterior. Assim, foi necessário calcular uma expressão análoga que desse a mesma indicação de uma forma mais realista para esta implementação em concreto. Considerando as rotinas que envolviam um maior número de operações (pela dimensão das matrizes envolvidas e pelos cálculos com elas realizados), ou seja, onde o algoritmo gastaria mais tempo, obteve-se a seguinte expressão para o número de multiplicações e de adições, novamente em função do número de variáveis de estado, n , e do número de observações, p :

$$\begin{cases} M(n, p) \approx 3n^3 + 2n^2 + 2\sum_{i=1}^n (n+1-i) + \left[2n^3 + 2n^2 + 5n + 9\sum_{i=1}^{n-1} 2i \right] p \\ A(n, p) \approx 3n^2(n-1) + n^2 + 2n(n-1) + \sum_{i=1}^n (n+1-i) + \left[2n^2(n-1) + n(n-1) + 4n + 3\sum_{i=1}^{n-1} 2i \right] p \end{cases} \quad (5.12)$$

Partindo das expressões (5.11) e (5.12) (que não é exacta por não terem sido contabilizadas operações matriciais de ordem reduzida), construiu-se a Tabela 5.1, de forma a ganhar maior sensibilidade em relação à variação do número de operações/tempo consumido com o aumento do número de variáveis de estado (correspondentes às colunas da tabela) e/ou número de observações (correspondentes às linhas da tabela). Esta tabela confronta a complexidade de um algoritmo que não explora a estrutura algébrica dos operadores (algoritmo não otimizado) com o algoritmo implementado.

Tabela 5.1 – Complexidade do algoritmo não otimizado vs algoritmo otimizado

	Algoritmo Não Otimizado					Algoritmo Otimizado				
	3	4	5	6	7	3	4	5	6	7
18	313893	399447	485001	570555	656109	119817	147717	175617	203517	231417
19	368448	468806	569164	669522	769880	139935	172444	204953	237462	269971
20	428990	545770	662550	779330	896110	162190	199790	237390	274990	312590

Da Tabela 5.1, conclui-se que a complexidade aumenta de forma linear com o aumento do número de observações e de forma cúbica com o número de variáveis de estado. Permite também ter uma ideia quantificada das optimizações produzidas no algoritmo. Verifica-se que, apenas contabilizando

operações algébricas, a redução foi superior a 50%, reflectindo-se a optimização realizada não apenas neste tipo de operações.

Com base nesta teoria, tentou-se minimizar o tempo de latência, sem que para isso o algoritmo perdesse a sua generalidade. Os passos mais importantes que foram dados para a resolução deste problema foram:

1. Uso de registos do DSP
2. Melhorar indexação matricial
3. Tirar partido das propriedades matriciais
4. Implementação em Assembly

O DSP apresenta um determinado número de registos para a execução de cada rotina. Uma vez que não era possível a colocação de todas as variáveis usadas em cada rotina nos registos, colocaram-se aquelas que eram utilizadas com maior frequência, o que levou a uma redução do número de acessos à memória e se traduziu num ganho de 10 ms.

Os tempos gastos na indexação matricial revelaram ser de extrema importância, já que a sua optimização levou a um ganho de 20 ms (passou de 42 ms para 22 ms para iterações em que não existem leituras de GPS). Como exemplo destes efeitos, apresentamos a multiplicação matricial que, como é sabido, tem uma complexidade algébrica de n^3 quando se multiplicam duas matrizes de dimensão $n \times n$. O que se tentou reduzir foi o número de operações auxiliares, como é o caso do acesso aos índices das matrizes. Assim, o código inicial da multiplicação matricial necessita de quatro operações (duas somas e duas multiplicações) apenas para aceder aos índices das matrizes pretendidos. Através de uma indexação mais eficiente consegue-se, apenas através de somas, aceder aos índices das matrizes, reduzindo para metade o número de operações auxiliares. O peso computacional desta optimização é elevado, uma vez que se trata de algo que é executado n^3 vezes e várias vezes por instante de amostragem.

Relativamente ao tirar partido das propriedades matriciais, foram utilizadas rotinas especializadas que aproveitavam o facto de algumas matrizes terem uma forma conhecida (triangular, diagonal, ...) independentemente da ordem do filtro de Kalman a usar.

Apesar de todas estas optimizações que levaram a uma redução drástica do tempo consumido pelo algoritmo, tal não foi, contudo, suficiente. Assim, as rotinas mais utilizadas e que representavam um maior gasto de tempo computacional foram implementadas de forma óptima em linguagem *Assembly*.

O impacto que as várias optimizações tiveram, mostra-se na Figura 5.2:

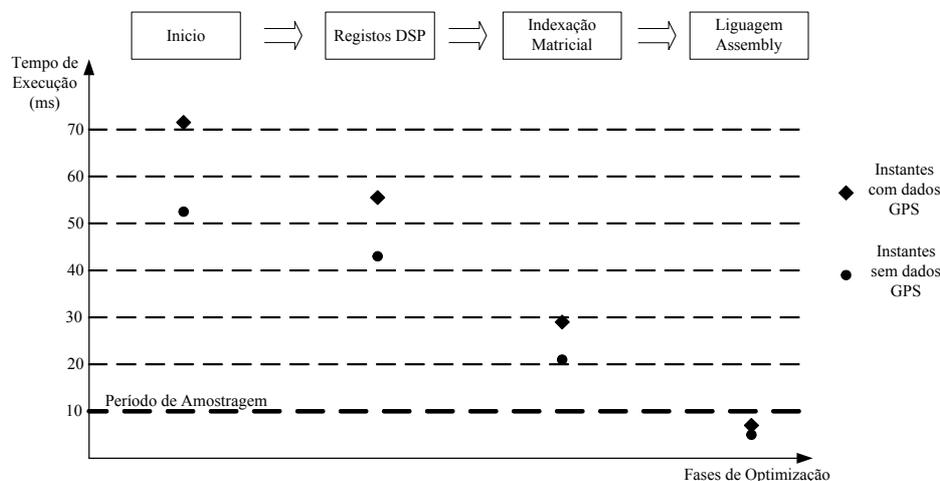


Figura 5.2 – Evolução da redução do consumo temporal do algoritmo

De mencionar que a redução dos 22 ms para os 5.5 ms de tempo de execução para instantes de amostragem sem dados de GPS foi conseguida à custa de optimizações a vários níveis, recorrendo também às propriedades deste algoritmo em particular. Este tipo de optimizações faz o algoritmo perder a generalidade, já que, para um outro caso diferente (diferente número de variáveis de estado, diferente número de observações, ...) o algoritmo tem que ser revisto, o que não acontece com as optimizações já descritas. Recorreu-se então, às propriedades que determinadas matrizes apresentam neste algoritmo em particular e concluiu-se que existem matrizes de dimensão elevada mas que, de instante de amostragem para instante de amostragem, apenas um reduzido número dos seus elementos sofrem alterações. Ao invés do que era feito inicialmente (essas matrizes eram inteiramente construídas a cada ciclo de execução), optou-se por inicializar essas estruturas antes de o algoritmo entrar em execução e, a cada instante de amostragem, apenas actualizar os elementos que sofrem alterações ao longo do tempo. Tal minimiza bastante o número de passos de computação, visto estes elementos representarem uma parte diminuta da totalidade dos elementos de cada uma dessas matrizes. Operações com matrizes que, pela sua estrutura (e.g. matriz H que apresenta apenas alguns elementos unitários, sendo os restantes nulos), têm o seu resultado conhecido à partida, passaram também a ser realizadas sem recorrer à operação em si.

Com isto chegou-se a um ponto em que se tinha como tempo de execução 7.4 ms e 10.5 ms para instantes de amostragem sem e com GPS, respectivamente. Optou-se então, por realizar as rotinas de multiplicação especializadas para as propriedades das matrizes, em linguagem *Assembly*. Embora esta implementação não tirasse partido das propriedades matriciais conhecidas e, por isso, não minimizasse o número de operações, revelou ser mais eficiente do que as rotinas especializadas, implementadas em C. Com estas melhorias e com a implementação também em linguagem *Assembly* de algumas rotinas de acesso à memória, obtiveram-se os resultados finais apresentados na Figura 5.2, que correspondem a um tempo de execução de 7.4 ms e de 5.5 ms para instantes de amostragem em que existem e não existem dados GPS, respectivamente.

5.2 Validação dos Sistemas de Navegação no Computador de Bordo

Neste projecto, o computador de bordo (CB) é composto pela placa MC-XAS3, a placa DSPif e a placa *D.Module.VC33* onde está incorporado o *TMS320VC33* apresentado no Apêndice K. Este teste visou validar os algoritmos no CB e medir os tempos de processamento, isto porque o objectivo é funcionar com o algoritmo a correr 100 vezes por segundo, o que leva a que tenha que ser executado em menos de 10 ms. Por esta razão foi necessário proceder a optimizações do código desenvolvido em ANSI C. Este procedimento apresenta-se na secção 5.1.

Atente-se à Figura 5.3 onde se pode observar o procedimento adoptado.

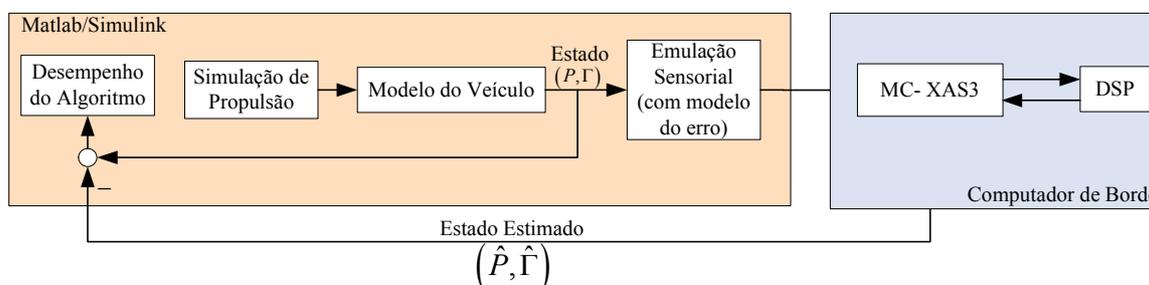


Figura 5.3 – Diagrama representativo do procedimento para a validação dos algoritmos

A troca de dados entre o *MATLAB* e a placa MC-XAS3 foi conseguida recorrendo à troca de pacotes *UDP-IP Ethernet*. No entanto, teve que ser desenvolvido um protocolo auxiliar para sincronizar e validar a troca de pacotes de dados (visto que o protocolo *UDP* não garante a sua entrega). Este protocolo é simples, o *MATLAB* envia um pacote novo sempre que recebe uma mensagem “válida” do XAS3. Se receber uma mensagem “errada” ou se passar um certo tempo (*Timeout*) repete o envio do pacote. Do lado do XAS3 o protocolo evolui de modo semelhante, se receber uma mensagem “válida” (pacote certo) coloca os dados no DSP. Depois de este terminar o processamento, o XAS3 envia uma mensagem ao *MATLAB* para sinalizar que está disponível para receber um novo pacote de dados. Se o pacote recebido não for o esperado, o XAS3 envia para o *MATLAB* o último pacote que recebeu correcto. Atente ao diagrama de estados na Figura 5.4.

No procedimento adoptado, a única diferença para o ambiente de simulação está no processador que faz os cálculos para a navegação. Estas diferenças são mínimas e devem-se a diferentes implementações de operações de vírgula flutuante de cada processador, portanto, não é correcto afirmar-se que uma é mais válida que a outra, apenas são diferentes. Tal pode ser constatado através dos resultados apresentados em figuras e dos valores estatísticos dos erros de estimação.

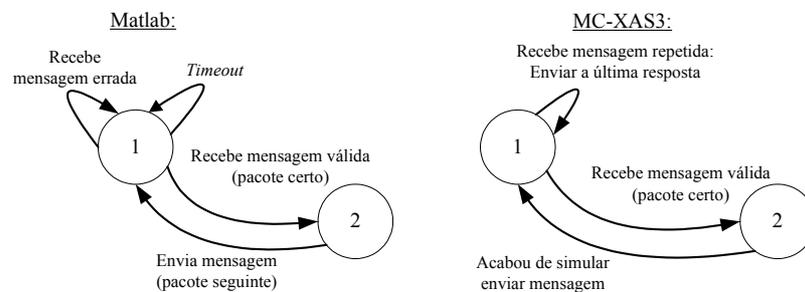


Figura 5.4 – Protocolo de comunicação entre *MATLAB* e XAS3

Para os testes modelaram-se as polarizações e desvios padrão dos ruídos dos sensores com valores reais típicos, apresentados na Tabela 5.2. Considerou-se, por razões de precisão, a existência de um GPS RTK (*Real Time Kinematics*).

Para a simulação utilizou-se um veículo genérico com 6 graus de liberdade (Apêndice E) e uma manobra que consiste numa propulsão positiva em modo comum segundo o eixo do x do referencial do veículo com duração de 20 s, seguida duma espiral ascendente até aos 191 s antes de retomar o modo comum inicial até ao final da simulação. No espaço 3D, a trajectória tem a forma apresentada na Figura 5.5.

Tabela 5.2 – Parâmetros da modelação dos sensores

	Giroscópios [$^{\circ}/s$]	Acelerómetros [m/s^2]	Magnetómetro [G]	GPS [m]
Polarização (<i>Bias</i>)	0,05	0,1	---	---
Desvio Padrão do Ruído (σ)	0,02	6×10^{-3}	1×10^{-6}	0,7

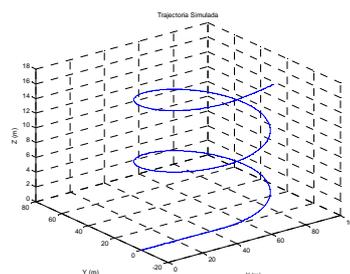


Figura 5.5 – Trajectória utilizada para a simulação e testes.

5.2.1 Sistema de navegação com recurso a filtros de Kalman complementares (CKF)

Na Figura 5.6 apresentam-se as diferenças, a cada instante de execução, entre a estimação calculada no *MATLAB* e a calculada no DSP. Como se comprova, as diferenças são muito pequenas, tendo em conta que estamos a trabalhar com grandezas na ordem das dezenas de metro e da unidade de radiano.

A Tabela 5.3 e a Tabela 5.4 apresentam os resultados do filtro executado no DSP e *MATLAB*, respectivamente. Como se verifica, a performance relativa ao real é muito semelhante.

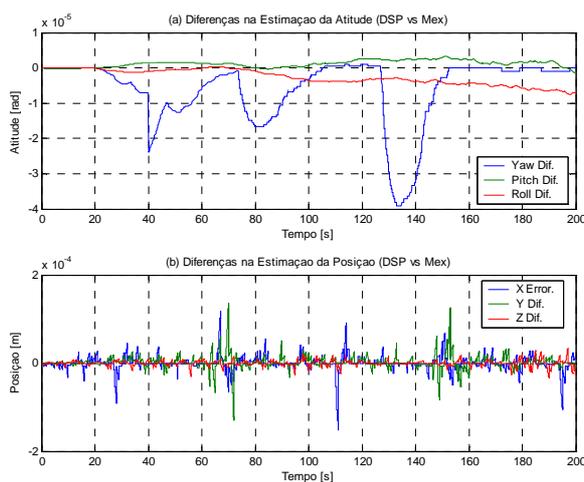


Figura 5.6 – Diferença entre a posição e a orientação estimada no DSP vs *MATLAB* (CKF)

Para uma avaliação do desempenho do sistema, comparou-se a estimação calculada no DSP com a trajectória real. A atitude e a posição são apresentadas, em função do tempo, na Figura 5.7. De notar, nesta figura, que se apresentam três gráficos, dois dos quais ((a) e (b)) referentes à atitude. Estes foram separados para dar a possibilidade de observar com detalhe a estimação de *Pitch* e *Roll*.

A estimação de atitude é calculada de forma correcta, no entanto, e devido a forças externas não gravíticas na medida do estimador magneto-pendular, surge um pequeno erro na estimação de atitude para acelerações relativamente bruscas, como se pode observar na Figura 5.7(a), na estimação de *Pitch* e *Roll*. Este erro afecta a compensação da aceleração gravítica, contribuindo para erros na estimativa de posição Figura 5.7(c).

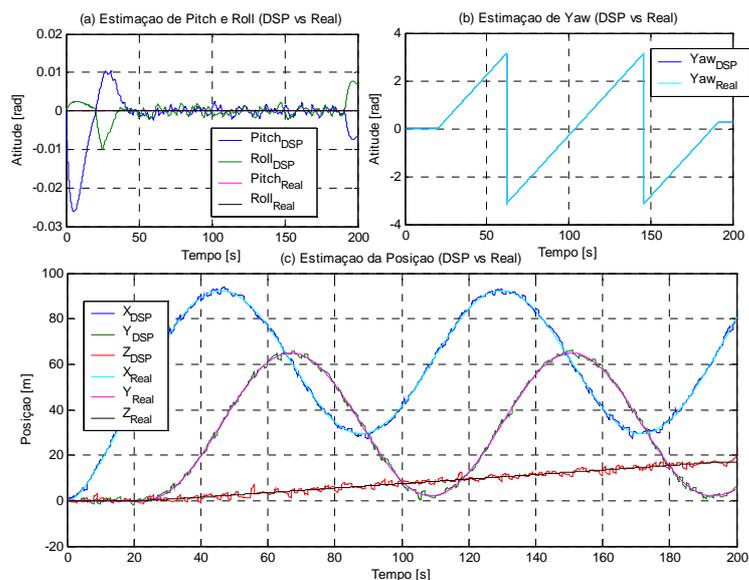


Figura 5.7 – Evolução temporal da atitude e posição estimadas e real (CKF)

A Tabela 5.3 e a Tabela 5.4 comprovam a boa performance do filtro; por exemplo, observa-se que o desvio padrão do erro de posição é da ordem de grandeza do desvio padrão do ruído do GPS. O que seria de esperar, pois as matrizes Q e R são tais que o filtro dá grande peso à medida de GPS, situação que poderia ser melhorada com o cálculo de novos ganhos. Portanto, neste ambiente, seria possível melhorar o desempenho do filtro, levando a que o sistema confiasse mais nas medidas dos outros sensores. Neste caso, não seria tão perceptível a medida de GPS, levando a que os resultados da estimação não se alterassem significativamente com a frequência de GPS.

Tabela 5.3 – Estatísticas do erro do filtro executado no DSP (CKF)

	Real – DSP					
	Erro de Posição [m]			Erro de Orientação [rad]		
Média	[-0,0403 -0,0490 -0,0267]			[-1,984 11,8 0,389] × 10 ⁻⁴		
Desvio Padrão	[0,8381 0,8314 0,9131]			[0,600 5,976 2,342] × 10 ⁻³		

Tabela 5.4 – Estatísticas do erro do filtro executado no MATLAB (CKF)

	Real – C (mex)					
	Erro de Posição [m]			Erro de Orientação [rad]		
Média	[-0.0403 -0.0490 -0.0268]			[-2,044 11,85 0,370] × 10 ⁻⁴		
Desvio Padrão	[0.8381 0.8315 0.9131]			[0,598 5,983 2,343] × 10 ⁻³		

5.2.2 Sistema de navegação com recurso a filtros de Kalman estendidos (EKF)

Na Figura 5.8 apresentam-se as diferenças, a cada instante de execução, entre a estimação calculada no MATLAB e a calculada no CB. Como se observa, as diferenças são pequenas, no entanto maiores que no caso anterior, facto que é explicável. No sistema baseado no filtro de Kalman estendido utiliza-

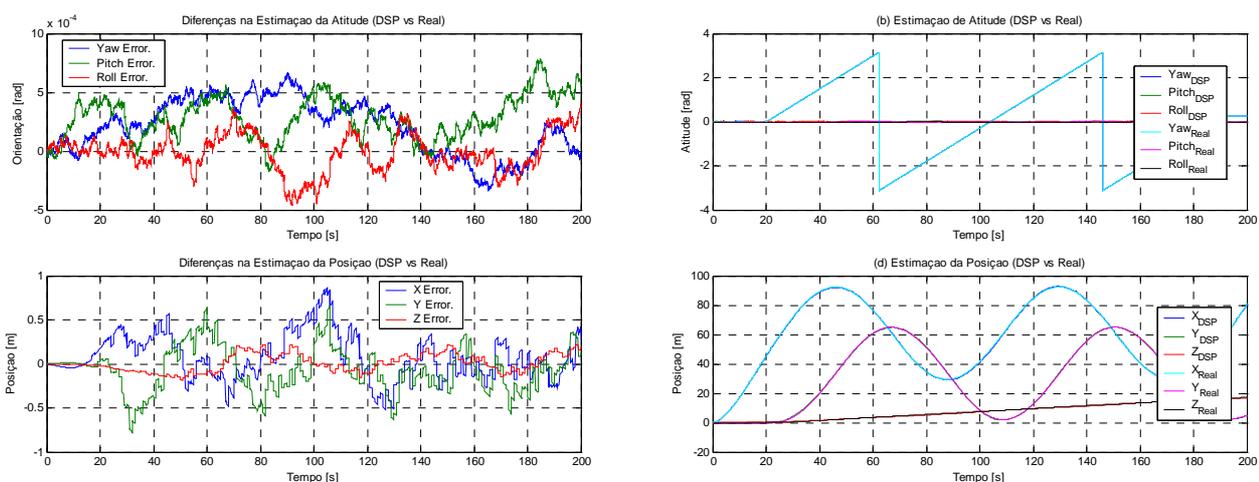


Figura 5.9 – Evolução Temporal da estimação e dos respectivos erros (DSP vs Real - EKF)

Tabela 5.5 – Estatísticas do erro do filtro executado no DSP (EKF)

	Real – DSP						
	Erro de Posição [m]			Erro de Orientação [rad]			
Média	[0,0739	-0,0727	-0,0093]	[2,096	2,847	-0,129]	$\times 10^{-4}$
Desvio Padrão	[0,2564	0,2525	0,1002]	[2,338	1,824	1,626]	$\times 10^{-4}$

Tabela 5.6 – Estatísticas do erro do filtro executado no MATLAB (EKF)

	Real – C (mex)						
	Erro de Posição [m]			Erro de Orientação [rad]			
Média	[0,0742	-0,0723	-0,0074]	[4,869	2,822	-0,103]	$\times 10^{-4}$
Desvio Padrão	[0,2566	0,2529	0,1000]	[2,856	1,820	1,625]	$\times 10^{-4}$

5.3 Hardware-In-The-Loop (HITL)

Dada a fragilidade e o custo deste tipo de veículos, surgiu a necessidade de desenvolver um sistema de HITL que permita realizar testes de algoritmos de navegação e controlo num ambiente seguro. Um sistema com estas características possibilita que o *software* em teste seja executado no *hardware* a ser utilizado na aplicação real.

Durante uma simulação do HITL, o *software* de bordo é executado na Arquitectura de *Hardware*, ao passo que todas as simulações (sensores, dinâmica e trajectória do veículo) são realizadas em Matlab/Simulink (ver Figura 5.10). Em relação ao que foi apresentado em 5.2, esta nova filosofia, possibilita a validação de toda a parte de aquisição de sinal, quer analógica, quer de dados de GPS.

Deste modo, a Arquitectura de *Hardware* interage com o exterior exactamente da mesma forma como se estivesse verdadeiramente a bordo do veículo, uma vez que todas as configurações/fluxo de dados são idênticas a uma montagem para um voo autónomo real. Assim, este procedimento demonstra

se a arquitectura tem ou não capacidade de fazer a aquisição de sinal e a execução de todo o *software* de bordo em tempo real, i.e. mostra se todas as componentes são executadas de forma suficientemente rápida.

Um *setup* experimental deste tipo necessita de um CB, de um PC, de um gerador analógico de sensores (composto por uma placa MC-XAS3, por nove conversores Digital-Analógicos com interface *Ethernet UDP-IP*). Na Figura 5.10 apresenta-se o fluxo de informação que foi implementado com o objectivo de validar a arquitectura completa.

O *setup* experimental concebido é ilustrado conceptualmente na Figura 2.1 onde se podem destacar alguns componentes desenvolvidos ou testados no âmbito deste projecto, tais como a Arquitectura de Aquisição de Sinal (descrita em 3) e a interface com o *D.Module.VC33* (descrito no Apêndice K).

A placa PCMCIA ([BCD04]) com interface *Ethernet*, representada na Figura 2.1 na pilha da (AH) não faz parte da arquitectura final, sendo substituída por uma carta também com interface PCMCIA, mas de memória *flash*, com o objectivo de armazenar os dados da missão. Numa missão real, é necessário acrescentar mais uma placa à AH, que possibilita a obtenção de dados de GPS. A geração dos sinais analógicos de tensão fica a cargo de duas placas ADDAOCT (descrita no Apêndice I) que possuem cada uma delas 8 DACs.

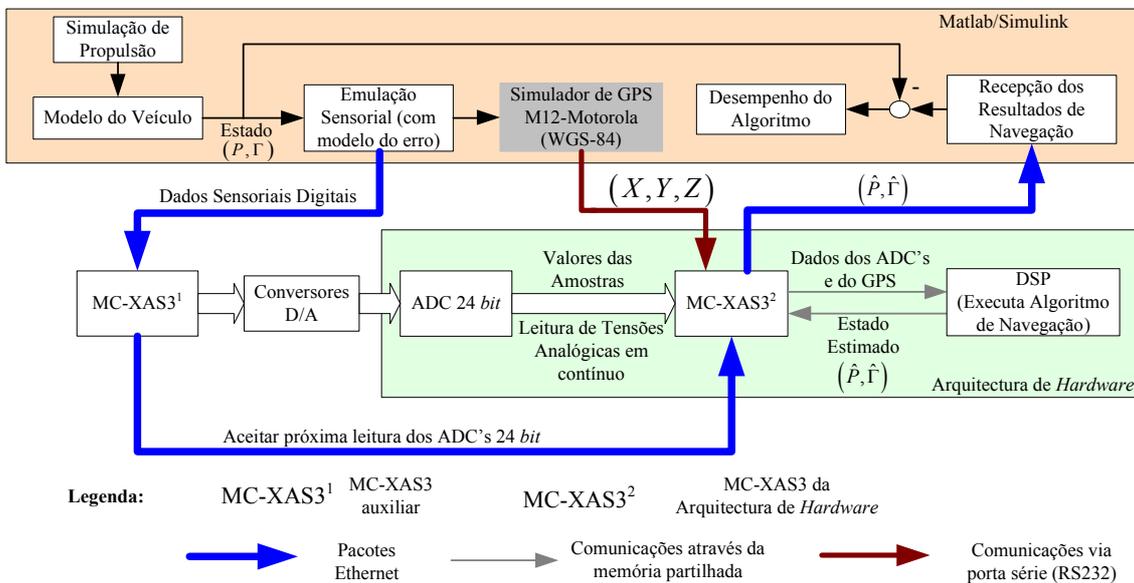


Figura 5.10 – Fluxo de informação no *Hardware no Loop*

5.3.1 Validação do *Loop* de navegação, Fluxo de Dados e Transduções

Quando se desenvolve um sistema deste género, onde se pretende modelar, o mais fielmente possível, todos os dispositivos e fenómenos envolvidos no processo é necessário ter o cuidado de validar todo o fluxo de informação que percorre o sistema. Por exemplo, simular um sensor com saídas analógicas, envolve o cálculo de transduções entre unidades no sentido inverso ao habitual. Torna-se necessária a transdução da unidade física que este sensor mede, para o valor de tensão em Volts que este colocaria nos seus terminais, se estivesse de facto sob o efeito dessa grandeza. Para ser possível avaliar o desempenho de um filtro recorrendo a um sistema do tipo HITL, é necessário primeiro que tudo, garantir que o filtro implementado opere sobre as mesmas entradas (até um grau de exactidão desejado), que o mesmo filtro implementado num ambiente puro de simulação. Na Figura 5.11

apresentam-se os dados sensoriais disponíveis para a implementação dos filtros em ambiente exclusivamente de simulação, e para a implementação no DSP depois destes terem passado pelo domínio de tensão analógica (acelerómetro, giroscópio e magnetómetro) e por mudanças de coordenadas e representação (GPS). Os sinais apresentados correspondem às medidas de sensor quando o veículo é sujeito à trajectória Figura 5.5.

Na Figura 5.12 observaram-se erros da ordem das centésimas e das milésimas nas unidades respectivas para o acelerómetro, giroscópio e para o magnetómetro. Os erros nas medidas de GPS também se encontram dentro dos limites esperados e são no máximo na ordem das décimas de metro. Tal como já foi referido anteriormente, a simulação dos sensores com saídas em tensão analógica, passa pela transdução da unidade física em questão, para um valor em Volts. Este valor é de seguida discretizado (com 14 *bits* de resolução) para que o DAC possa gerar um sinal analógico de tensão. Em seguida, o valor de tensão é novamente discretizado pela AAS, sendo de esperar que os dados sensoriais sejam ligeiramente diferentes, já que sofrem duplamente o ruído de discretização, são expostos a ruído electromagnético, afectados com o ruído intrínseco dos DACs e dos ADCs e sofrem por acumulação de erros numéricos.

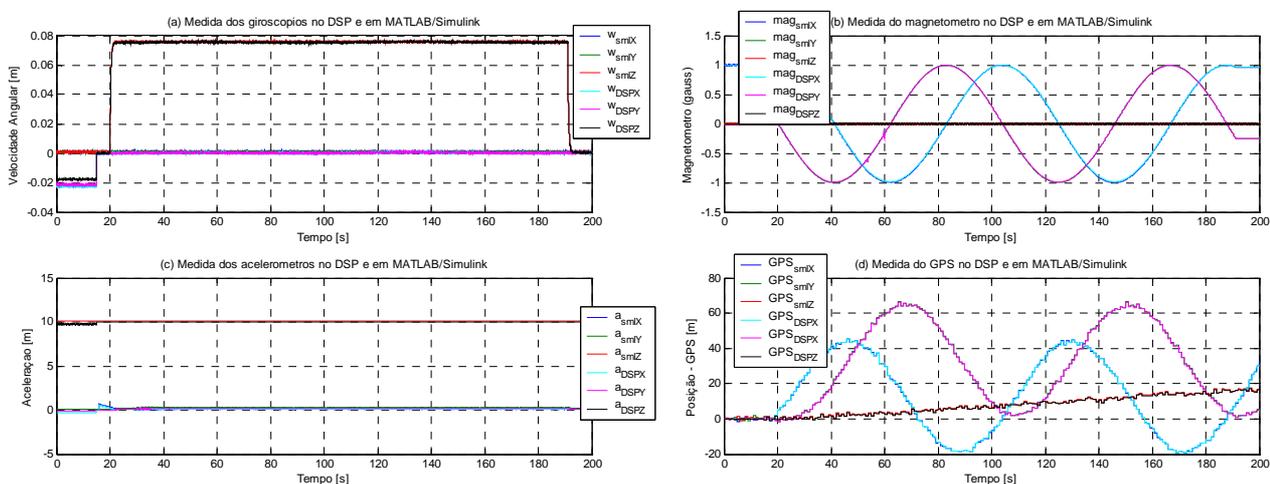


Figura 5.11 – Evolução temporal da medida simulada dos sensores em *MATLAB* e no DSP

De salientar, nas Figura 5.11 e Figura 5.12, os primeiros 15 segundos de simulação, pois são de elevada importância para o funcionamento dos filtros (ver em 4), sendo necessários para a correcta remoção das polarizações introduzidas não só devido à simulação deste fenómeno nos sensores, mas também devido aos *offsets* que os DACs das duas placas ADDAOCT (Apêndice I) introduzem.

Depois de validado o fluxo de dados, garante-se que as duas implementações dos filtros (em ambiente de simulação e HITL) trabalham sobre dados semelhantes, esperando-se então que a performance seja aproximada. Para tal desenvolveram-se testes com o filtro de Kalman complementar e filtro de Kalman estendido, que se apresentam nas secções 5.3.2 e 5.3.3, respectivamente.

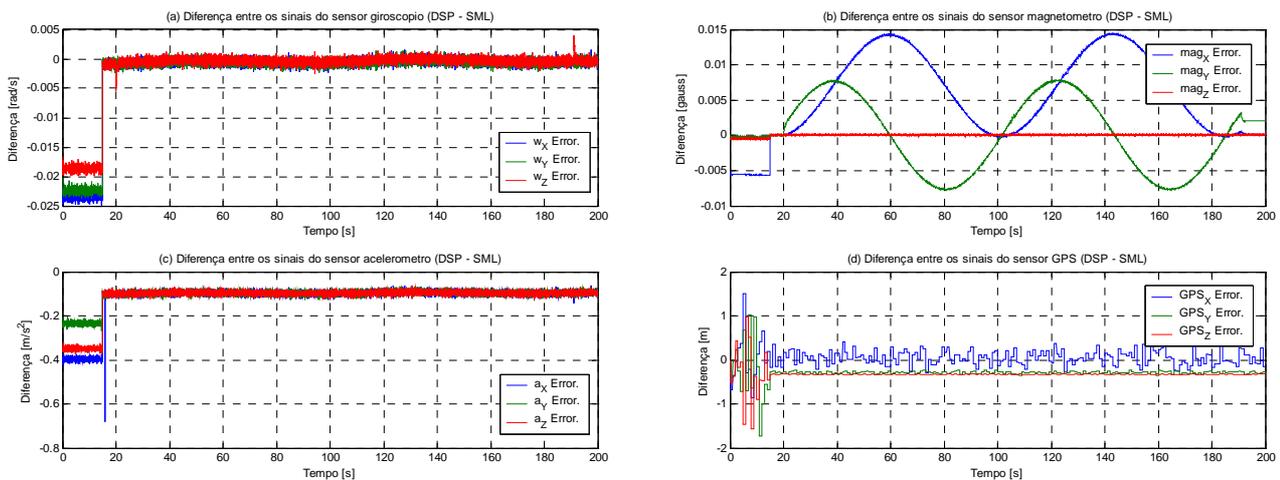


Figura 5.12 – Diferença entre os dados sensoriais na implementação em *MATLAB* e os dados disponíveis para o DSP

5.3.2 Sistema de navegação com recurso a filtros de Kalman complementares

Na secção 5.2.1, conclui-se que o filtro de Kalman complementar atribui grande importância às medidas de posição fornecidas pelo GPS, como demonstra a Figura 5.13 onde o seu desempenho não foi significativamente afectado pela correcção não exacta das polarizações introduzidas nas leituras dos sensores, quer as originadas pelo modelo dos sensores quer pelo *offset* gerado pelas não idealidades dos DAC's.

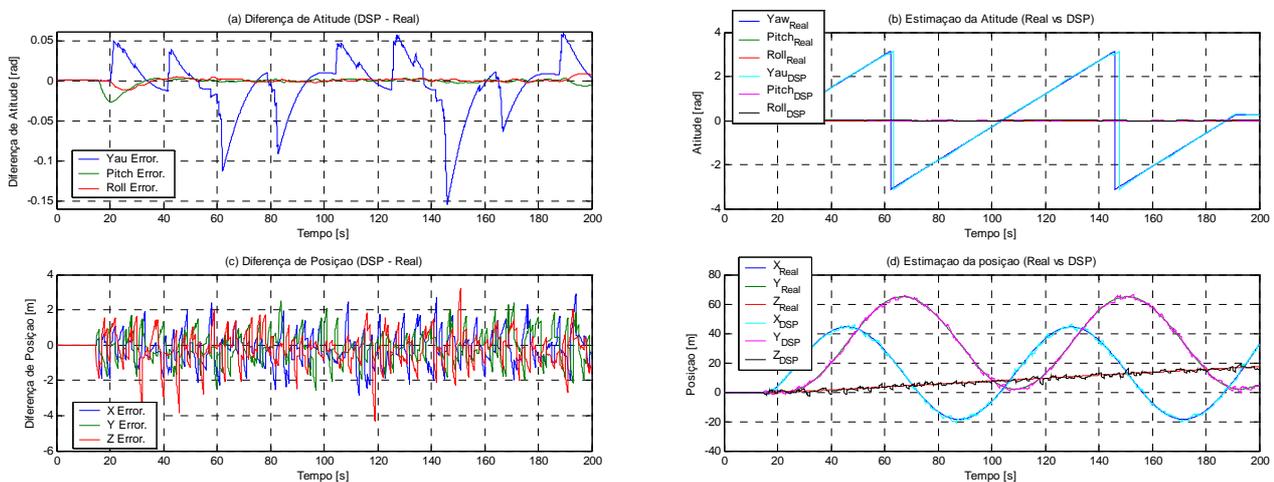


Figura 5.13 – Evolução Temporal da estimação e dos respectivos erros (CKF)

Um dos motivos para o bom desempenho que o filtro de Kalman complementar apresentou, mesmo sob condições não ideais, está relacionado com o facto de, tal como foi dito anteriormente (ver 4.1), este filtro estimar internamente as referidas polarizações conseguindo assim uma melhor correcção para as mesmas.

Observando a Figura 5.14, pode-se constatar que as polarizações estimadas em ambiente de simulação e em HITL são diferentes, ficando este facto a dever-se à calibração efectuada nos primeiros 15 s da simulação, onde se calcula a média do erro entre as medidas fornecidas pelos sensores e as medidas que estes deveriam fornecer quando o veículo se encontra parado no plano horizontal. Esta correcção elimina tanto as polarizações introduzidas no modelo do sensor como as introduzidas pelos DAC's. Portanto, as polarizações estimadas pelas duas implementações pelo filtro têm que ser forçosamente diferentes.

Os resultados da Tabela 5.7 evidenciam o que foi dito anteriormente, o desvio padrão do erro de posição é da ordem de $10^{-1} m$ e o desvio padrão de atitude da ordem de $10^{-3} rad$.

Tabela 5.7 – Estatísticas do erro do filtro executado no DSP (CKF)

	Real – DSP	
	Erro de Posição [m]	Erro de Orientação [rad]
Média	$[-1.883 \times 10^{-1} \quad -1.034 \times 10^{-1} \quad -1.915 \times 10^{-1}]$	$[-4.959 \times 10^{-3} \quad -1.442 \times 10^{-3} \quad -1.889 \times 10^{-5}]$
Desvio Padrão	$[8.747 \times 10^{-1} \quad 8.178 \times 10^{-1} \quad 8.736 \times 10^{-1}]$	$[3.427 \times 10^{-2} \quad 4.960 \times 10^{-3} \quad 3.103 \times 10^{-3}]$

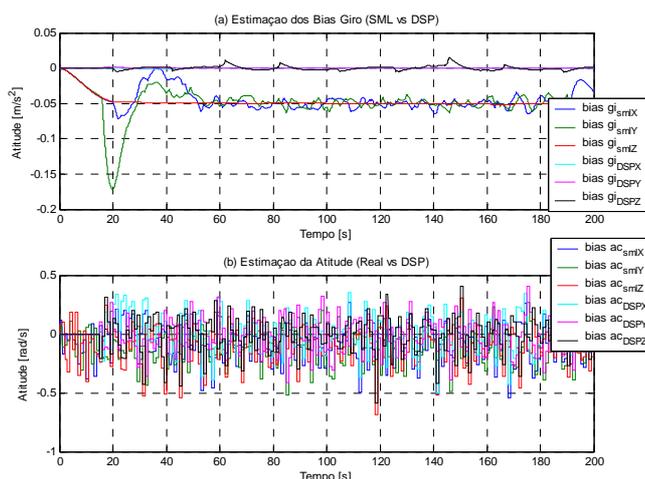


Figura 5.14 – Estimação das polarizações dos giroscópios e dos acelerómetros (CKF)

5.3.3 Sistema de navegação com recurso ao filtro de Kalman estendido

O filtro de Kalman estendido apresenta, para as mesmas condições de simulação, uma performance diferente da do filtro complementar, verificando-se uma degradação do desempenho; em que se evidencia características já identificadas anteriormente (5.2.2). O facto do filtro recorrer a uma dupla integração das medidas dos sensores, não estimar a polarização dos mesmos e por estas não serem exactamente corrigidas verifica-se, tal como se pode observar na Figura 5.15, uma divergência acentuada da trajectória estimada para a trajectória real (fenómeno identificado em [CaVa03]).

Este fenómeno também se explica pelo facto dos parâmetros do filtro não modelarem as não idealidades introduzidas pelos DAC's.

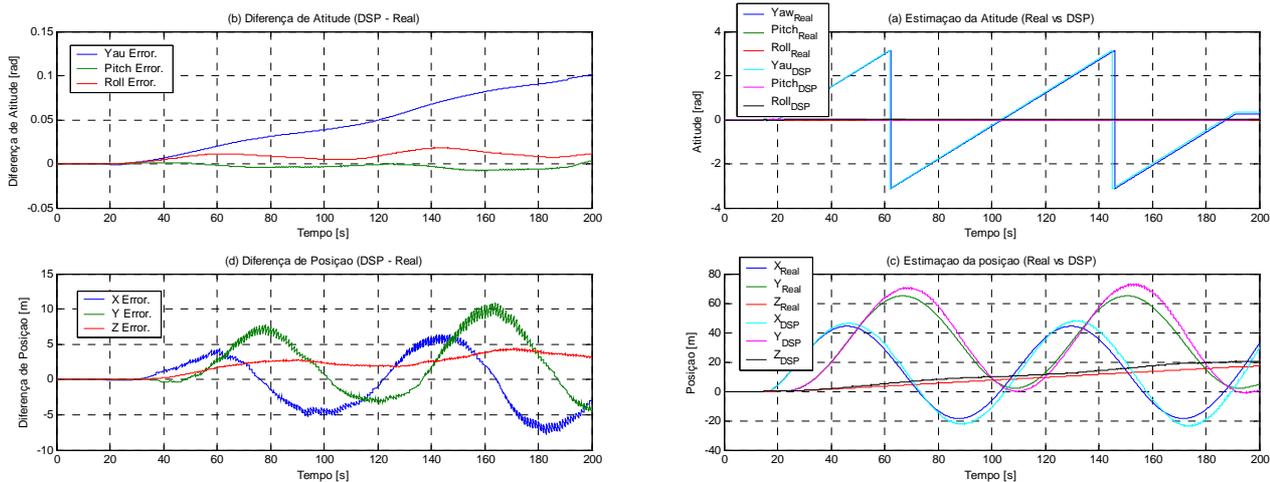


Figura 5.15 – Evolução Temporal da estimação e dos respectivos erros (EKF)

A degradação da performance também é evidente na estimação da atitude do veículo, por exemplo, o erro na estimação do ângulo de *Yaw* (Figura 5.15(a)) apresenta no final da trajetória um valor de aproximadamente 0.1 rad , e uma evolução temporal que aparenta ser sempre crescente, levando o filtro a divergir.

A Tabela 5.8 comprova o que foi anteriormente referido, notando-se que os erros de estimação de posição apresentados são da ordem das dezenas de metro, não estando estes de acordo com os obtidos em 5.2.2. Concluiu-se assim, que o fraco desempenho do filtro se deve ao facto de a formulação do mesmo não modelar todos os fenómenos envolvidos no processo e os ganhos calculados não serem adequados à realidade do sistema.

Tabela 5.8 – Estatísticas do erro do filtro executado no DSP (EKF)

	Real – DSP	
	Erro de Posição [m]	Erro de Orientação [rad]
Média	$[-3.534 \times 10^{-1} \ 1.980 \times 10^0 \ 2.071 \times 10^0]$	$[4.339 \times 10^{-2} \ -2.459 \times 10^{-3} \ 8.292 \times 10^{-3}]$
Desvio Padrão	$[3.572 \times 10^0 \ 3.751 \times 10^0 \ 1.344 \times 10^0]$	$[3.286 \times 10^{-2} \ 2.619 \times 10^{-3} \ 4.981 \times 10^{-3}]$

Por esta razão, aumentaram-se as variâncias dos sensores inerciais, com o objectivo do filtro lhes atribuir menos importância, dado que as polarizações não são eliminadas correctamente, como referido anteriormente. Este problema assume maior relevo pelo facto do sensor GPS apresentar uma variância maior que a dos outros sensores, não lhe sendo atribuída importância na correcção dos erros originados pela dupla integração das polarizações.

Após o aumento das variâncias dos sensores, acelerómetro e giroscópio, o desempenho do filtro foi substancialmente melhor como a Figura 5.16 e Tabela 5.9 comprovam.

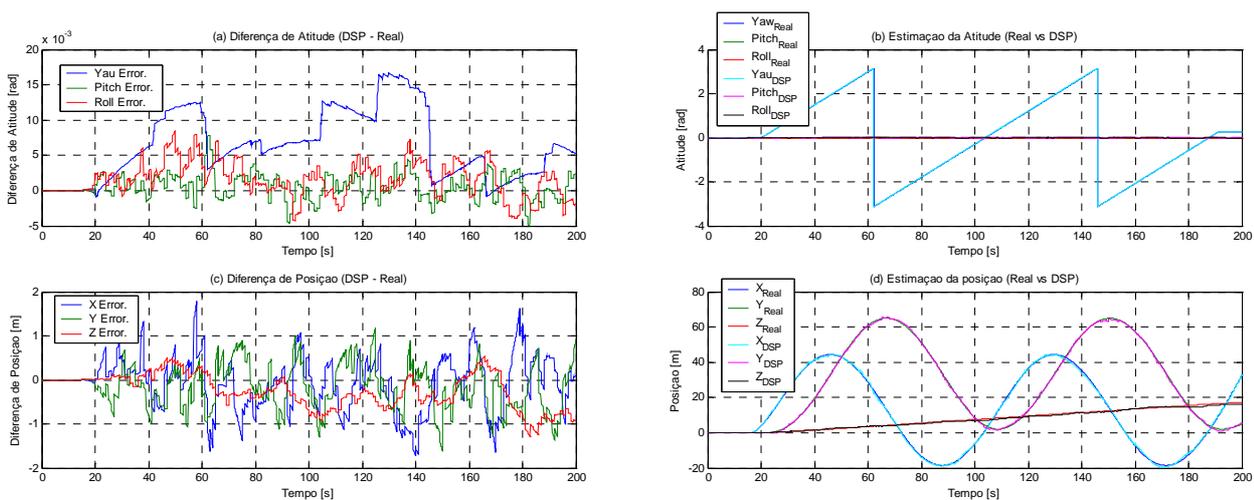


Figura 5.16 – Evolução Temporal da estimação e dos respectivos erros (EKF)

Tabela 5.9 – Estatísticas do erro do filtro executado no DSP (EKF)

	Real – DSP	
	Erro de Posição [m]	Erro de Orientação [rad]
Média	$[-7.219 \times 10^{-2} \quad -1.644 \times 10^{-1} \quad -2.479 \times 10^{-1}]$	$[6.296 \times 10^{-3} \quad 5.425 \times 10^{-3} \quad 1.531 \times 10^{-3}]$
Desvio Padrão	$[5.942 \times 10^{-1} \quad 5.011 \times 10^{-1} \quad 3.851 \times 10^{-1}]$	$[4.825 \times 10^{-3} \quad 1.797 \times 10^{-3} \quad 2.435 \times 10^{-3}]$

6 Conclusões

Com o trabalho descrito neste relatório foram desenvolvidas arquitecturas de *hardware* para sistemas de navegação, assim como um simulador de *Hardware-In-The-Loop* (HITL) que possibilita testar estes sistemas num ambiente seguro, mas com condições semelhantes às reais. Um simulador deste tipo permite uma maior facilidade no teste de algoritmos de navegação e de controlo num ambiente onde a maioria dos requisitos de um teste real são cumpridos, nomeadamente requisitos de tempo real. Uma das grandes vantagens deste tipo de simulador consiste em utilizar todo o *hardware* que irá ser colocado a bordo do veículo para a implementação dos sistemas de navegação. Recorrendo a um modelo do veículo e dos sensores utilizados numa aplicação real, consegue-se simular as grandezas que a Arquitectura de Aquisição de Sinal receberia, caso estivesse a bordo do veículo numa missão real.

Neste trabalho foi dada uma contribuição importante para a evolução da Arquitectura de *Hardware*. Conseguiu-se que esta não representasse uma limitação para o bom desempenho dos algoritmos de navegação testados em ambiente de simulação (Matlab/Simulink). Estas limitações poderiam ter origem na Arquitectura de Aquisição de Sinal, se esta fosse uma fonte de ruído/incerteza nas leituras feitas às medidas dos sensores. Provou-se que tal não é o caso, uma vez que se obteve uma resolução efectiva de aproximadamente 19 *bit* para a frequência de trabalho (100Hz).

Uma vez garantido que as leituras dos sensores não condicionavam o desempenho dos algoritmos de navegação, foi necessário introduzir uma capacidade computacional acrescida, de modo a conseguir operar à frequência desejada, o que se conseguiu recorrendo a um DSP de alto desempenho (150 MFLOPS) e baixo consumo. A introdução deste DSP foi realizada à custa da placa DSPiF (desenvolvida no DSOR) que faz a interface entre dois barramentos distintos, conseguida por uma *Dual-Port RAM* de acesso rápido.

Obtida a descrição matemática dos algoritmos de navegação a implementar (filtro de Kalman complementar e filtro de Kalman estendido), conseguiu-se a validação destes em linguagem ANSI C. Tal foi feito recorrendo ao modelo do simulador de [CaVa03] e à implementação em Matlab *Mex-Functions* dos algoritmos. A comparação entre as estimativas produzidas pelos algoritmos implementados revelou que as diferenças eram praticamente inexistentes. No entanto, a ordem de grandeza desta diferença está dependente se a implementação em ANSI C é feita utilizando uma representação numérica de *double precision* (64 *bit*) ou *single precision* (32 *bit*). No primeiro caso o desvio-padrão das diferenças das estimativas são na ordem de grandeza 10^{-13} m e 10^{-14} rad e para o segundo as mesmas diferenças são na ordem de 10^{-3} m e 10^{-5} rad.

Os mesmos algoritmos tiveram o seu desempenho avaliado no computador de bordo através da comparação entre a trajectória real, obtida pelo simulador e a estimação calculada por este e pela implementação feita através de Matlab *Mex-Functions* (em *single precision*). Obteve-se um desvio-padrão na ordem dos 10^{-1} m e de 10^{-3} rad para as duas implementações, o que leva a concluir que não há uma degradação do desempenho do algoritmo quando executado no CB, apesar de os desvios-padrão entre as duas implementações (Matlab *Mex-Functions* vs CB) se situar na ordem dos 10^{-3} m e 10^{-4} rad. Tal situação explica-se dado que o DSP e o Matlab têm diferentes implementações numéricas de vírgula flutuante, embora sejam ambas de 32 *bit*.

A implementação do EKF levantou algumas questões de possível instabilidade numérica, pelo que se optou por implementar o método conhecido por factorização UD. Este método conserva as propriedades da matriz da covariância, nomeadamente a sua simetria. Isto foi comprovado em simulação, constatando-se que, para o algoritmo, onde não era tida qualquer precaução, esta matriz apresentava diferenças, para a matriz esperada, na ordem dos 10^{-5} ao fim de algum tempo, com

tendência a aumentar. Com o método da factorização UD, devido à estrutura do algoritmo, estes erros são negligenciáveis independentemente do tempo de operação do filtro.

A arquitectura de sistemas de navegação ficou completa com o simulador HITL que provou que o sistema desenvolvido se encontra preparado para uma substituição dos modelos dos sensores em Matlab/Simulink por sensores reais. Para ser possível a execução do filtro de Kalman estendido (EKF) na arquitectura, foi necessário proceder a uma optimização da implementação realizada em ANSI C, já que este, na sua versão original, apresentava um tempo computacional superior a 70 ms. Com o desenvolvimento de técnicas de implementação eficientes de algoritmos de navegação, conseguiu-se um tempo de processamento de cerca de 7 ms. Com este sistema, obtiveram-se resultados de navegação bastante satisfatórios levando a crer que a arquitectura de *Hardware* funcione a bordo de um veículo real, igualmente com bom desempenho.

Da pesquisa relacionada com a modelação de perturbações num magnetómetro, surgiu uma nova proposta para a descrição dos erros que afectam as leituras do referido sensor. Esta caracteriza o erro no espaço das variáveis do sensor, o que se considera ser um modelo mais realista. No que diz respeito à calibração do magnetómetro, visando a correcção de perturbações geradas segundo o modelo do erro sugerido, conseguiu-se a validação em ambiente de simulação de um novo algoritmo de calibração para trajectórias no plano horizontal. Reduziu-se em cerca de duas ordens de grandeza (em radianos), o erro na obtenção do rumo (*Yaw*) de um veículo, recorrendo às leituras de um magnetómetro corrigido com parâmetros obtidos pelo referido algoritmo de calibração.

Uma continuação da optimização do algoritmo EKF é algo que pode ser feito futuramente tendo em vista uma maior redução do seu tempo de processamento, através da implementação, em linguagem *Assembly*, de rotinas que representem um tempo computacional elevado, quer pelo número de operações realizadas, quer pelo número de vezes que estas são executadas em cada instante de amostragem. Esta implementação pode tirar partido da estrutura algébrica do algoritmo.

Sugere-se ainda o desenvolvimento do filtro de Kalman estendido, de modo a que este passe a estimar as polarizações dos sensores. Actualmente, parte-se do pressuposto que todas as medidas de sensores que entram para filtro não apresentam polarizações. O filtro só exhibe um bom comportamento, se estas polarizações forem compensadas na perfeição, caso contrário, apresenta uma degradação no desempenho levando-o à instabilidade. Este desenvolvimento no filtro aumentaria assim a sua robustez.

Como trabalho futuro, sugere-se também a validação dos algoritmos de navegação, recorrendo a sensores reais, sendo este o último passo a dar antes de testar a Arquitectura de *Hardware* e o sistema de navegação no helicóptero num voo não autónomo.

No que diz respeito à calibração e modelação do erro do magnetómetro, sugere-se como trabalho futuro, a realização de testes experimentais que possibilitem validar os modelos e os algoritmos apresentados neste trabalho. De salientar a importância de se obter uma medição de *heading* de grande exactidão (recorrendo a manobras que tirem partido das leituras de GPS) para se poder aferir a validade dos modelos. Sugere-se também o estudo das perturbações no espaço dos três eixos do magnetómetro quando o veículo perfaz trajectórias no espaço de atitude com ângulos de *Roll* e de *Pitch* diferentes de zero. Quando representada no espaço das três coordenadas do sensor, esta perturbação traduz-se com o aparecimento de um elipsóide onde, em condições ideais deveria aparecer uma superfície esférica.

Apêndices

A Representação da Orientação e Referenciais de Navegação

É necessária uma forma matemática para descrever a orientação do veículo e dos referenciais utilizados. Neste Apêndice apresentam-se algumas alternativas para descrição da orientação dos referenciais, sendo posteriormente definidos os que foram utilizados neste trabalho.

A.1 Representação da Orientação

A abordagem que se segue tem em conta o facto desta representação ser necessária para algoritmos de navegação, focando aspectos como a composição de transformações, existência de singularidades e dinâmica de representação.

A.1.1 Matriz de Rotação

Esta matriz descreve rotações de vectores e efectua o mapeamento de coordenadas entre referenciais. De acordo com a notação seguida em [CaVa03], a transformação de coordenadas de um vector representado no referencial {B} para o referencial {A} é escrita como:

$${}^A P = {}^A R {}^B P \quad (\text{A.1})$$

A matriz de rotação ${}^A R$ pode ser obtida pela projecção, disposta por colunas, dos vectores ortonormados do referencial {B} segundo os vectores ortonormais do referencial {A}.

$${}^A R = \begin{bmatrix} \langle X_B, X_A \rangle & \langle Y_B, X_A \rangle & \langle Z_B, X_A \rangle \\ \langle X_B, Y_A \rangle & \langle Y_B, Y_A \rangle & \langle Z_B, Y_A \rangle \\ \langle X_B, Z_A \rangle & \langle Y_B, Z_A \rangle & \langle Z_B, Z_A \rangle \end{bmatrix} \quad (\text{A.2})$$

onde X, Y, Z são os vectores ortonormais de cada um dos referenciais e \langle , \rangle representa o operador produto interno. Dado que o produto interno de dois vectores de norma unitária é igual ao coseno do ângulo formado pelos dois vectores, torna-se clara a razão dos componentes desta matriz serem designados por cosenos directores.

De (A.1), é directa a relação de ${}^A R$ com a transformação inversa ${}^B R$. Sabe-se da álgebra linear que a inversa de uma matriz com colunas ortonormadas é igual à sua transposta.

$${}^B R = ({}^A R)^{-1} = ({}^A R)^{-T} \quad (\text{A.3})$$

A transformação entre referenciais pode ser obtida através das transformações intermédias simples:

$${}^A_C R = {}^A_B R {}^B_C R \quad (\text{A.4})$$

envolvendo o referencial {B} na rotação entre o referencial {A} e o referencial {C}.

A dinâmica da matriz de rotação é dada por [CaVa03]:

$$\frac{d}{{}^A_B R} = {}^A_B R \left[{}^B ({}^A \omega_B) \times \right] \quad (\text{A.5})$$

onde ${}^B ({}^A \omega_B) = [p \ q \ r]^T$ corresponde à velocidade angular do referencial {B} em relação ao referencial {A}, vista no referencial {B} e $\left[{}^B ({}^A \omega_B) \times \right]$ é a matriz *skew-symmetric* do operador produto externo, definido como:

$$\left[{}^B ({}^A \omega_B) \times \right] = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (\text{A.6})$$

A.1.2 Ângulos de Euler

Nesta representação, cada rotação é realizada em torno de um eixo do referencial em movimento, {B}, em vez de um referencial fixo, {A}. Este conjunto de três rotações, em que a orientação de cada eixo é alterada após cada rotação, designa-se por ângulos de Euler (Figura A.1)

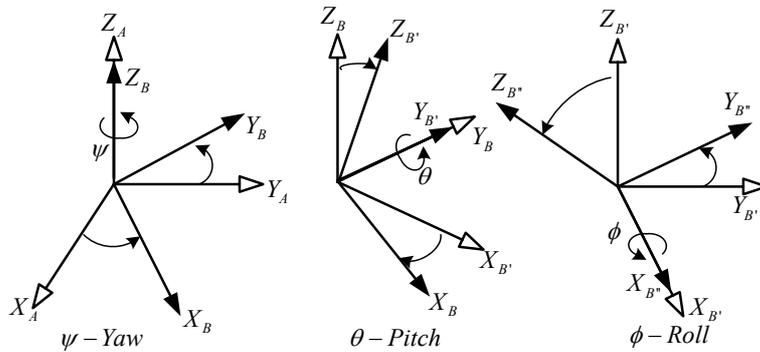


Figura A.1 – Ângulos de Euler Z-Y-X

Uma vez que as três rotações ocorrem segundo os eixos Z-Y-X, esta representação denomina-se por ângulos de Euler Z-Y-X, cuja matriz de rotação, obtida pela sequência de rotações em torno de cada eixo, ${}^A_B R_{ZYX} = R_Z(\psi)R_Y(\theta)R_X(\phi)$, é dada por:

$$R_{ZYX}(\Gamma) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi - c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (\text{A.7})$$

onde $s \equiv \sin$, $c \equiv \cos$ e $\Gamma = [\psi \ \theta \ \phi]^T$ representa a atitude em ângulos de rotação *Yaw*, *Pitch* e *Roll*, os quais correspondem a rotações em torno dos eixos Z, Y e X, respectivamente. A solução para os ângulos de Euler Z-Y-X é apresentada em [Crai89]:

$$\begin{aligned} \theta &= \text{atan} \left(\frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}} \right) \\ \psi &= \text{atan2} \left(\frac{r_{21}}{c\theta}, \frac{r_{11}}{c\theta} \right) \\ \phi &= \text{atan2} \left(\frac{r_{31}}{c\theta}, \frac{r_{32}}{c\theta} \right) \end{aligned} \quad (\text{A.8})$$

onde r_{ij} representa o elemento de ${}^A_B R$, correspondente à linha i , coluna j . Dada a existência de uma raiz quadrada para o cálculo de θ , assumem-se apenas como soluções que respeitam a condição $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Esta representação apresenta ainda problemas com singularidades para $\theta = \pm \frac{\pi}{2}$, para as quais se encontram solução em [Crai89]:

$$\begin{cases} \theta = \frac{\pi}{2} \\ \psi = 0 \\ \phi = \text{atan2}(r_{21}, r_{22}) \end{cases}, \quad \begin{cases} \theta = -\frac{\pi}{2} \\ \psi = 0 \\ \phi = -\text{atan2}(r_{21}, r_{22}) \end{cases} \quad (\text{A.9})$$

A dinâmica dos ângulos de Euler obtém-se escrevendo explicitamente os elementos da matriz de rotação como função dos ângulos de Euler e desenvolvendo a dinâmica da matriz de rotação em função da velocidade angular, resultando:

$$\dot{\Gamma} = Q(\Gamma)^B ({}^I \omega_B) \quad (\text{A.10})$$

$$Q(\Gamma) = \begin{bmatrix} 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \\ 0 & \cos \phi & -\sin \phi \\ 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \end{bmatrix}$$

A.1.3 Vector de Rotação

Como alternativa ao uso de rotações sequenciais, o referencial {B} pode ser obtido directamente rodando o referencial {A} de um ângulo Λ em torno do eixo K . O vector de rotação é definido como:

$${}^A\lambda_B = \Lambda \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \quad (\text{A.11})$$

onde $\Lambda \in [0, \pi]$ é o ângulo de rotação e $K = [k_x \ k_y \ k_z]^T$ é o vector de norma unitária que define o eixo de rotação. De [CaVa03], tem-se que a matriz de rotação pode ser obtida a partir do vector de rotação pela expressão:

$$R({}^A\lambda_B) = I_{3 \times 3} + \frac{\sin \Lambda}{\Lambda} ({}^A\lambda_B \times) + \frac{1 - \cos \Lambda}{\Lambda^2} ({}^A\lambda_B \times) ({}^A\lambda_B \times) \quad (\text{A.12})$$

onde $(\lambda \times)$ é o operador produto externo definido de forma análoga a (A.6).

A dinâmica do vector de rotação é apresentada em [CaVa03]:

$${}^A\dot{\lambda}_B = {}^B({}^A\omega_B) + \frac{1}{2} {}^A\lambda_B \times {}^B({}^A\omega_B) + \frac{1}{\Lambda^2} \left(1 - \frac{\Lambda \sin \Lambda}{2(1 - \cos \Lambda)} \right) {}^A\lambda_B \times ({}^A\lambda_B \times {}^B({}^A\omega_B)) \quad (\text{A.13})$$

Esta notação apresenta problemas de precisão para rotações muito pequenas e está também sujeito a singularidades para $\Lambda=0$ e $\Lambda=\pi$. Para mais pormenores recorrer a [CaVa03].

A.1.4 Quaterniões

Um quaternião é um formalismo matemático que permite descrever rotações em torno de um eixo. É descrito por uma componente escalar q_s e outra vectorial q_v :

$$q = (q_s, q_v) = q_4 + iq_1 + jq_2 + kq_3 \quad (\text{A.14})$$

onde i, j e k são as componentes da parte vectorial $q_s = (q_1, q_2, q_3)$ e q_4 é a parte escalar $q_s = q_4$.

A partir de uma rotação em torno de um eixo, obtêm-se as componentes do quaternião através da expressão:

$$q_v = \sin\left(\frac{\Lambda}{2}\right) \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \quad (\text{A.15})$$

$$q_s = \cos\left(\frac{\Lambda}{2}\right)$$

onde $\Lambda \in [0, \pi]$ é o ângulo de rotação e $K = [k_x \quad k_y \quad k_z]^T$ é o vector de norma unitária que define o eixo de rotação.

A partir do quaternião é possível definir a matriz de rotação [CaVa03]:

$$R(q) = (q_s^2 - \|q_v\|^2) I_3 + 2q_v q_v^T + 2q_s (q_s \times) \quad (\text{A.16})$$

De [CaVa03], obtém-se a expressão do quaternião a partir dos elementos da matriz de rotação:

$$q_v = \frac{1}{4q_s} \begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix} \quad (\text{A.17})$$

$$q_s = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}}$$

Esta representação tem a vantagem de não apresentar singularidades e permitir calcular directamente rotações e mapeamento de coordenadas sem recorrer à matriz de rotação. Para isto, [CaVa03] define o produto interno como:

$$q_1 q_2 = \Xi(q_1) q_2 = \Upsilon(q_2) q_1 \quad (\text{A.18})$$

onde

$$\Xi(q) = \begin{bmatrix} q_s + q_v \times & q_v \\ -q_v^T & q_s \end{bmatrix} \quad \text{e} \quad \Upsilon(q) = \begin{bmatrix} q_s - q_v \times & q_v \\ -q_v^T & q_s \end{bmatrix}$$

Recorrendo à estrutura do quaternião, a transformação de um vector visto no referencial {B} para coordenadas do referencial {A}, é dada por:

$${}^A P_q = {}^A q {}^B P_q {}^A q^{-1} \quad (\text{A.19})$$

onde $P_q = (0, P)$ representa o quaternião auxiliar com parte vectorial correspondente às coordenadas do ponto e parte escalar nula, ${}^A q$ é o quaternião que descreve a orientação do referencial {B} em relação ao referencial {A} e ${}^A q^{-1}$ é o recíproco do quaternião ${}^A q$. O recíproco de um quaternião, devido à sua norma unitária, é dado por:

$$q^{-1} = \frac{q^*}{\|q\|} = q^* = (q_s, -q_v) \quad (\text{A.20})$$

onde q^* representa o conjugado do quaternião.

A composição e a inversa das transformções é dada por (de [CaVa03]):

$$\begin{aligned} {}^A_C q &= {}^A_B q {}^B_C q \\ \left({}^A_B q {}^B_C q \right)^{-1} &= {}^B_C q^{-1} {}^A_B q^{-1} \end{aligned} \quad (\text{A.21})$$

De referir a linearidade da equação da dinâmica do quaternião, que possibilita uma solução exacta para a actualização da atitude e é dada por:

$$\frac{d^A_B q}{dt} = \frac{1}{2} A(\Omega)^A_B q \quad (\text{A.22})$$

onde ${}^A_B q$ é o quaternião que representa a orientação do referencial {B} em relação ao referencial {A}, $\Omega = {}^B \left({}^A \omega_B \right) = [p \ q \ r]^T$ é a velocidade angular do referencial {B} em relação ao referencial {A}, vista do referencial {B} e $A(\Omega)$ é dado por:

$$A(\Omega) = \begin{bmatrix} -\Omega \times & \Omega \\ -\Omega^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & r & -q & p \\ -r & 0 & p & q \\ q & -p & 0 & r \\ -p & -q & -r & 0 \end{bmatrix} \quad (\text{A.23})$$

A.2 Definição dos Referenciais

Referencial é o mesmo que sistema de coordenadas. Para navegar é preciso, no mínimo, dois referenciais, um para o veículo e outro para representar a navegação (mapa).

Lista de alguns referenciais:

- Referencial do Corpo

Este é o referencial base para os sensores inerciais e é o mesmo que o do veículo. O eixo x aponta para a frente, o y para a direita e o eixo z aponta para baixo, formando, portanto, um sistema ortogonal. A origem coincide com a posição dos sensores, topologia *strapdown* (Figura A.2).

- Referencial NED na superfície terrestre

Este referencial é usado para a navegação. Existem três eixos que formam um sistema ortogonal da mão direita. O vector N aponta para norte, E aponta para este e D aponta na direcção do vector gravidade. Os eixos N e E formam um plano tangente à superfície da Terra e D é mapeado da origem deste plano (Figura A.2). Ter em atenção que este referencial resulta bem desde que o veículo não se afaste muito da origem.

- WGS-84, *World Geodetic System 1984*

O *Geodetic System* lembra bastante o mapeamento elíptico, usando dois ângulos e a altura, que é medida relativamente ao geóide definido.

- Referencial ECI – *Earth Centered Inertial*

Este sistema de coordenadas é usado para referência inercial. O referencial inercial é fixo com alguns dos eixos a apontar para estrelas distantes. Este referencial não acompanha a rotação da Terra e tem a sua origem no centro de massa da Terra.

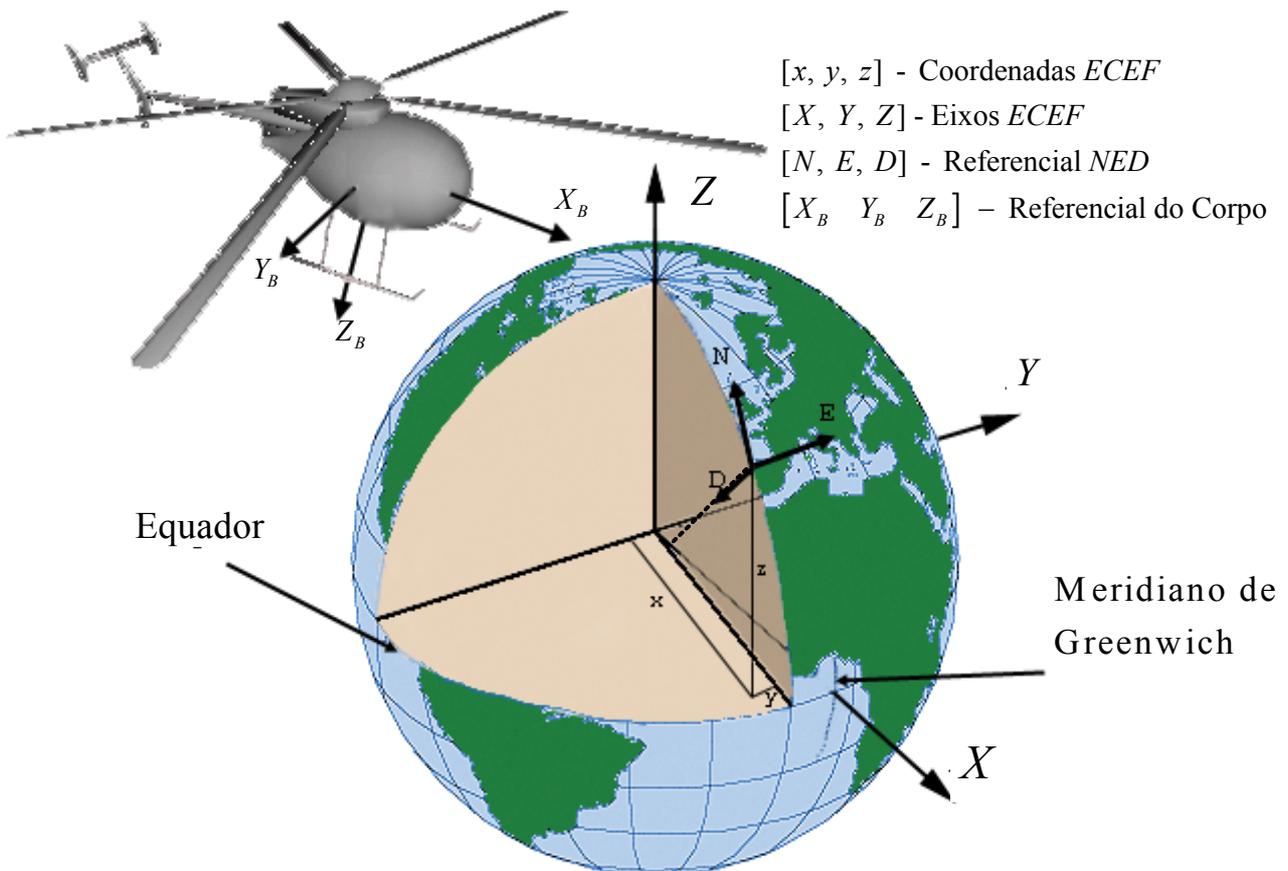


Figura A.2 – Referenciais NED, ECEF e do corpo

- Referencial ECEF – *Earth Centered Earth Fixed*

Este referencial é fixo com respeito à Terra e acompanha a sua rotação. O ECEF é usado no sistema do GPS, chamado de WGS-84. Portanto as observações do GPS são representadas neste referencial. O ECEF pode, também, ser chamado de referencial cartesiano da Terra, por causa das coordenadas cartesianas [x, y, z]_{ECEF} (ver Figura A.2).

Para o projecto em questão são apenas necessários alguns referenciais. Utiliza-se o referencial do corpo {B}, o referencial da Terra {E} e um referencial inercial de interesse (local) {I}. De seguida passa-se à explicação e à razão do uso destes referenciais.

Como referido em [CaVa03], não é possível definir a Terra como referencial inercial, pelo que, para lidar com estes fenómenos num sistema de navegação, a orientação do corpo rígido em relação a um referencial de interesse é obtida através do encadeamento de referenciais. Assim, segundo [CaVa03], é apenas necessário considerar o referencial inercial, {I}, de interesse para descrever a orientação do veículo e o referencial da Terra (*Earth*), {E}, usado para acompanhar o movimento de rotação da Terra. Segundo o teorema de Coriolis, a velocidade observada no referencial {I} é igual à velocidade em relação ao referencial {E}, afectada pela velocidade de rotação entre referenciais:

$${}^I V_{BORG} = {}^I \left({}^E V_{Borg} \right) + {}^I \omega_E \times {}^I P_{BORG} \quad (\text{A.24})$$

onde ${}^I \omega_E = [0 \ 0 \ \Omega]$ é o vector da velocidade angular resultante da rotação da Terra, com velocidade angular Ω constante.

O referencial {B} diz respeito ao referencial do corpo na topologia *strapdown*, cujos eixos coincidem com a colocação dos sensores e cuja origem não coincide, em geral, com o centro de massa do corpo. Por último, o referencial {G} consiste no referencial do centro de massa, cuja origem coincide com o centro de massa e cuja orientação é idêntica à do referencial {B}.

Neste trabalho, a escolha dos referenciais permite contabilizar a rotação terrestre num ambiente de navegação local através do uso dos referenciais Inercial, Terra, Corpo e Centro de Massa.

B Sensores

Neste capítulo apresentam-se e descrevem-se os sensores utilizados para medir as grandezas necessárias para a navegação. Estes podem ser divididos em dois grupos, sensores agregados ao veículo, que fornecem medidas expressas no referencial do corpo {B} e sensores auxiliares que fornecem medidas directas das grandezas a estimar. A lista de sensores apresenta-se na Tabela B.1.

Tabela B.1 – Sensores utilizados para a navegação

Sensores Instalados no Veículo	<ul style="list-style-type: none">• Acelerómetros• Giroscópios• Magnetómetro
Sensores Auxiliares	<ul style="list-style-type: none">• Sistema Global de Posição (GPS)• Estimador de Atitude Magneto-Pendular

Os sensores agregados ao veículo são montados triaxialmente, medindo grandezas vectoriais. De referir que é fundamental serem colocados, entre si, alinhados, ou seja, os três eixos de cada sensor estarem o mais alinhados possível.

B.1 Sensores Integrados no Veículo

B.1.1 Acelerómetro

O princípio de funcionamento do acelerómetro pode socorrer-se de diversos conceitos físicos. No entanto, a grandeza medida é independente da tecnologia utilizada portanto, conceptualmente, o dispositivo equivale ao sistema físico que consiste numa massa ligada a uma mola cuja frequência natural é muito superior à frequência máxima de funcionamento do acelerómetro, como referido em [Kell94].

Da segunda lei de Newton, uma massa sujeita a acelerações sofre uma força que resulta na distensão da mola, originando uma força elástica de sentido contrário. Medindo o comprimento da mola determina-se a aceleração segundo o eixo de leitura do acelerómetro.

Como qualquer massa está constantemente sujeita à força gravítica, os acelerómetros medem uma aceleração que não existe, de sentido contrário à gravidade. Assim a medida, designada por força específica, resulta da aceleração do veículo adicionada à aceleração gravítica [Kell94]:

$$a = \left(\frac{d^1 V_{BORG}}{dt} \right) + {}^B g \quad (\text{B.1})$$

Onde a é a força específica e ${}^B g$ é o vector aceleração gravítica observada no referencial do corpo.

B.1.2 Giroscópio

No projecto utiliza-se um giroscópio de velocidade (*rate gyro*) para medir a velocidade angular do referencial {B} em relação ao referencial {I}, expressa em {B}:

$${}^B({}^I\omega_B) = [p \quad q \quad r]^T \quad (\text{B.2})$$

A variação de atitude segundo ângulos de Euler é obtida directamente a partir da leitura do giroscópio:

$$\dot{\Gamma} = Q(\Gamma) {}^B({}^I\omega_B) \quad (\text{B.3})$$

onde:

$$Q(\Gamma) = \begin{bmatrix} 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \\ 0 & \cos \phi & -\sin \phi \\ 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \end{bmatrix} \quad (\text{B.4})$$

B.1.3 Magnetómetro

Recorre-se a um magnetómetro para determinação da direcção do campo magnético terrestre. O magnetómetro de fluxo (*fluxgate*) é um sensor que mede o campo magnético através de indução de correntes eléctricas numa bobine cujo núcleo é saturado periodicamente. A existência de um campo magnético não nulo distorce a simetria associada à histerese, provocando perturbações na corrente induzida, que podem ser detectadas com uma análise em frequência.

Os compassos magnéticos são facilmente afectados por perturbações do meio envolvente, que surgem com campos magnéticos de outros dispositivos e com a proximidade de materiais de permeabilidade diversa.

- **Características do Campo Magnético Terrestre**

O planeta Terra pode ser considerado como um íman gigante que se encontra envolvido por linhas de fluxo magnético que conectam dois pólos magnéticos. Estes pólos magnéticos estão perto, embora não coincidentes, com os pólos geográficos da Terra.

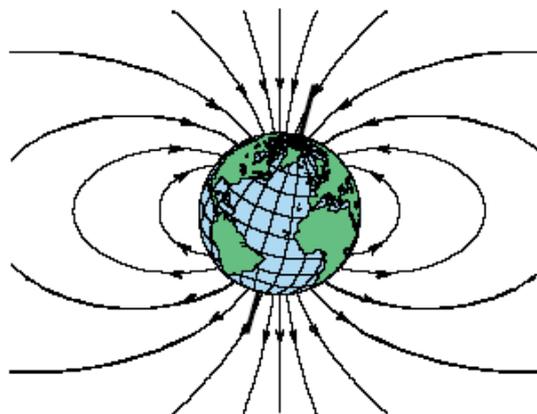


Figura B.1 – Ilustração da não sobreposição entre os pólos magnéticos e os pólos geográficos terrestres

Como se ilustra na Figura B.1, as linhas de força do campo magnético da Terra, penetram a superfície terrestre com diferentes ângulos com a horizontal local. Este ângulo com a horizontal, é designado por ângulo de *dip*. Este ângulo aumenta em valor absoluto de 0°, desde o equador magnético, até 90° nos pólos magnéticos.

Devido à não correspondência entre os pólos magnéticos e os pólos geográficos³, qualquer dispositivo magnético (bússola, magnetómetro, ou outro) não calibrado, não vai indicar o Norte geográfico, mas sim o Norte magnético. A diferença angular entre o meridiano verdadeiro (circulo que une os pólos geográficos) e o meridiano magnético (tem a direcção das linhas de fluxo do campo magnético) é designada por declinação. Esta declinação tem valores diferentes, em locais diferentes na superfície da Terra.

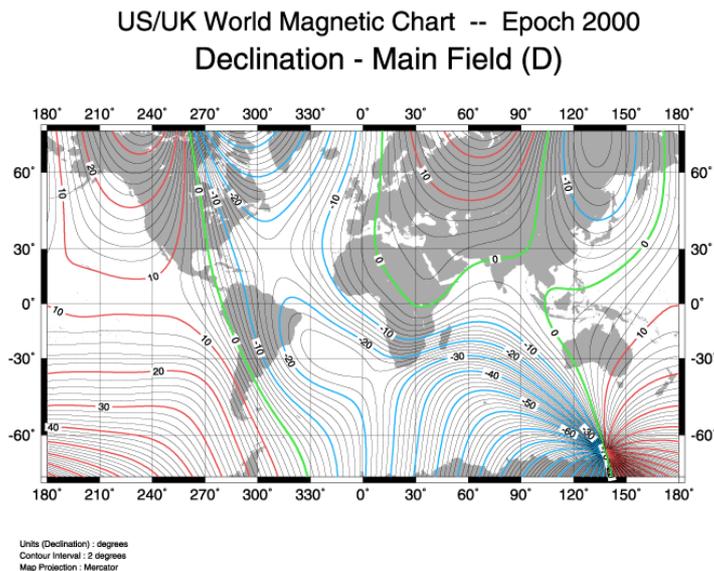


Figura B.2 – Mapa Mundial de declinação magnética (extraído de [NGDC])

É usual definir o campo magnético terrestre num determinado ponto da superfície terrestre em relação ao referencial geográfico recorrendo a dois ângulos: declinação, o ângulo entre a componente horizontal do campo magnético em relação ao Norte verdadeiro, e inclinação, o ângulo entre o vector campo magnético terrestre e o plano horizontal. A intensidade, que é independente da orientação do referencial, é representada como a intensidade do vector campo magnético terrestre.

³ O pólo Norte geomagnético está localizado no antártico Canadiano com coordenadas 82° N de latitude e 248°E de longitude. O pólo Sul geomagnético está aproximadamente localizado na coordenadas 65° Sul de Latitude e 138°E de Longitude. De notar que os pólos geomagnéticos não são anti-odais, senso uma assimetria que também mede a complexidade do campo magnético terrestre

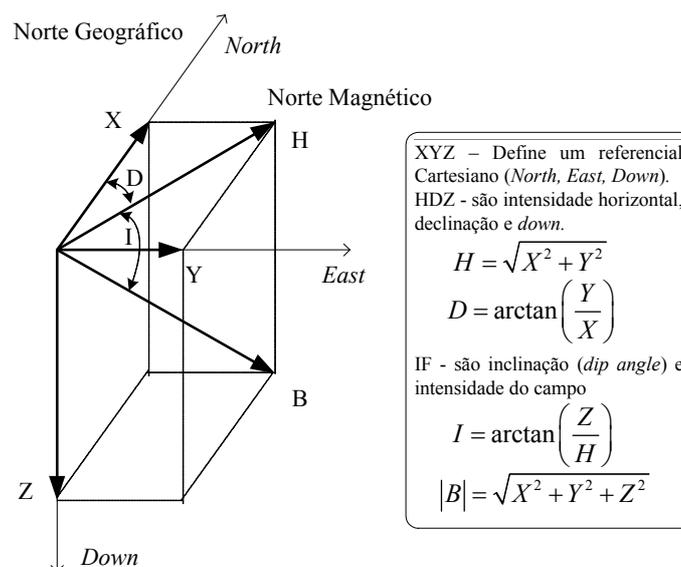


Figura B.3 – Descrição do campo magnético terrestre num ponto da superfície da mesma

- **Utilização de um Magnetómetro para Navegação**

Passaram já muitos séculos, desde a primeira utilização do campo magnético terrestre por um ser humano, para obter informação sobre o seu rumo (*Yaw* ou *heading*). É legítimo afirmar que a maioria, senão totalidade dos veículos navais, aeronáuticos e alguns terrestres, utiliza de alguma forma o campo magnético terrestre para o cálculo do rumo da plataforma em questão.

Os dispositivos de medição de campos magnéticos utilizados na actualidade (como por exemplo um magnetómetro) surgem em geral sob a forma de sensores de 2 ou 3 eixos ortogonais entre si, possibilitando medir componentes do vector campo magnético terrestre. São necessárias as componentes horizontais do referido campo para se conseguir obter uma medida de *heading* em relação ao Norte magnético.

Se o veículo, ao qual o magnetómetro está solidário, tiver ângulos de *Roll* ou de *Pitch* diferentes de zero, os eixos horizontais vão sentir componentes verticais. Por este motivo, um magnetómetro de somente dois eixos, nas referidas condições, não tem a capacidade de determinar o ângulo de *Yaw*. Um sensor com três eixos pode determinar a orientação dos seus eixos somente se a magnitude do campo magnético for conhecida ou se assumir um valor para a mesma. Somente quando se dispõe de um inclinómetro, é que se torna possível a determinação da magnitude e da orientação do campo magnético em relação aos eixos do sensor.

Um magnetómetro bi-axial, em que os seus dois eixos ortogonais se encontram no plano horizontal (designados de forma simplificada por X_h e Y_h), sob o efeito do campo magnético terrestre sem perturbações, origina as leituras apresentadas na Figura B.4, quando rodado de 360°, no plano horizontal.

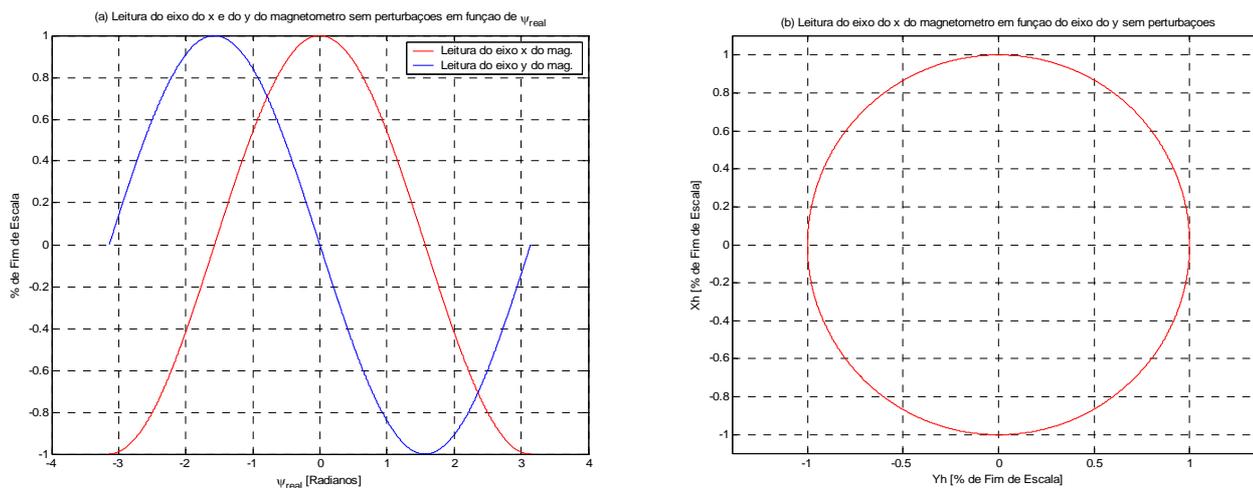


Figura B.4 – Leitura de dois sensores magnéticos ortogonais (inseridos no plano horizontal) quando rodados de 360° no plano horizontal: em relação ao rumo (a), e um em relação ao outro (b)

B.1.3.1 Descrição e Modelação das Fontes de Erro

- **Declinação Magnética**

Tal como foi referido anteriormente, o não conhecimento da declinação magnética na região onde se está a navegar, pode levar a erros graves na obtenção de uma medida de *heading* (rumo). Em geral, o fenómeno de declinação magnética não representa um problema de difícil resolução, visto que é relativamente fácil a obtenção dos valores de correcção para um determinado local de interesse.

- **Distorção *Hard Iron***

Neste tipo de distorção, estão envolvidos os materiais ferromagnéticos com propriedades magnéticas permanentes, isto é, materiais que sofreram, no passado, o efeito de um campo magnético (deliberadamente ou não) que os magnetizou. Devido a serem materiais duros (*hard iron*), com menor permeabilidade magnética relativa, são de difícil desmagnetização, comportando-se assim como ímanes permanentes. Estes materiais geram campos magnéticos que se sobrepõem ao campo magnético da Terra perturbando a sua leitura.

Os materiais que podem gerar erros de leitura podem existir em dois locais, ou estão no ambiente ou pertencem à estrutura do veículo. Das fontes de campo magnético perturbador do meio envolvente, identificam-se os materiais com propriedades magnéticas que se encontram no solo e no subsolo, e possíveis estruturas metálicas que existam perto das trajectórias do veículo.

Consoante a sua localização, as perturbações fazem-se sentir de forma diferente pelo sensor. As ambientais, de origem geológica, originam perturbações invariantes no tempo e em geral pode-se afirmar que também o são no espaço, já que estes fenómenos abrangem áreas normalmente superiores às cobertas por uma missão normal de um veículo autónomo. Estes fenómenos, em geral, estão mapeados o que possibilita a sua calibração em *offline*.

- **Distorção *Soft Iron***

Os materiais ferromagnéticos, ditos “macios”, têm a capacidade de gerar campos magnéticos quando estão sob o efeito de um campo magnético exterior. Têm a referida característica, devido à sua elevada permeabilidade magnética.

Em relação às perturbações produzidas por estes campos magnéticos, estas são diferentes das originadas pelo *hard iron* no sentido em que dependem da atitude do veículo, já que, o campo magnético terrestre penetra o veículo de forma não uniforme, quando por exemplo o veículo tem um movimento de rotação, no plano horizontal.

- **Distorção de Desalinhamento**

Surgem erros de desalinhamento quando, numa montagem não ideal, não se consegue um alinhamento perfeito da tríade de magnetómetros com o referencial do corpo do veículo. Esta fonte de erro, não sendo fácil de eliminar totalmente, pode ser bastante minimizada. No espaço de *heading* (*Yaw*), este erro, revela-se como uma constante que é somada à medida *heading* verdadeira.

- **Distorção Devido a Erros de Factor de Escala**

Em condições ditas ideais, os três magnetómetros da tríade são exactamente iguais. Na realidade, não se consegue obter os três dispositivos com uma sensibilidade perfeitamente igual. As consequências deste fenómeno são, por exemplo, leituras diferentes quando são sujeitos ao mesmo campo magnético.

- **Distorção Devido a Equipamentos Eléctricos**

A existência de cablagem que transporta corrente eléctrica, motores eléctricos ou outros dispositivos electrónicos no veículo, gera campos magnéticos que se sobrepõem de forma corruptiva ao campo magnético terrestre.

Este tipo de distorção não tem uma correcção algorítmica trivial, devido ao carácter não determinístico e variante no tempo dos campos magnéticos que corrompem as medidas. Assim, a solução para este problema está na colocação cuidada dos sensores em zonas do veículo com menos ruído electromagnético de forma a minimizar a exposição às referidas perturbações.

- **Distorção Devido à Não Ortogonalidade nos Eixos do Sensor**

Os erros produzidos por este fenómeno podem ser desprezados, já que, a não ortogonalidade dos eixos dos magnetómetros aparece nos manuais dos mesmos como sendo da ordem de $\pm 0.1^\circ$, não produzindo desvios que se comparem aos originados por outras fontes de erro.

B.1.3.2 Modelo do Erro das Medidas do Magnetómetro

Em seguida apresentam-se os efeitos de algumas destas distorções no espaço de *heading* e no espaço das coordenadas do magnetómetro. A análise, que se apresenta, pretende caracterizar os fenómenos de distorção somente para manobras que mantenham o veículo na horizontal, portanto, com os ângulos de *Roll* e de *Pitch* nulos.

- **Espaço de *Heading***

Na literatura [Bowd95], [Mach03] e [Denn79], a perturbação que aparece na medida de *Yaw* obtida a partir das leituras do magnetómetro pode ser aproximada por:

$$E_{HI}(\psi_m) \approx R_{HI} \sin(\psi_m + \gamma_{HI}) \quad (\text{B.5})$$

que se pode reescrever da seguinte forma:

$$E_{HI}(\psi_m) \approx B \sin(\psi_m) + C \cos(\psi_m) \quad (\text{B.6})$$

Este efeito é frequentemente designado por erro de ciclo único, já que a cada rotação completa, o erro é aproximado por um período de uma sinusóide.

Os efeitos dos materiais ditos do tipo *soft iron* são responsáveis pelo aparecimento de uma sinusóide de frequência dupla, em relação à que aparece nos efeitos *hard iron*. Este fenómeno modela-se no espaço de *heading* da seguinte forma:

$$E_{SI}(\psi_m) \approx R_{SI} \sin(2\psi_m + \gamma_{SI}) \quad (\text{B.7})$$

que é equivalente a:

$$E_{SI}(\psi_m) \approx D \sin(2\psi_m) + E \cos(2\psi_m) \quad (\text{B.8})$$

O desalinhamento horizontal entre o referencial do corpo e o referencial do sensor são modelados de forma exacta no espaço de *heading* como um *offset*:

$$E_M(\psi_m) = A \quad (\text{B.9})$$

Ao juntar os efeitos das fontes de erros apresentadas anteriormente, define-se um modelo aproximado para o erro no espaço de *heading*:

$$\psi_m = \psi_{real} + E_{HI+SI+M}(\psi_m) \quad (\text{B.10})$$

com

$$E_{HI+SI+M}(\psi_m) \approx A + B \sin(\psi_m) + C \cos(\psi_m) + D \sin(2\psi_m) + E \cos(2\psi_m) \quad (\text{B.11})$$

Na Figura B.5(a), (b) e (c) faz-se um resumo gráfico das 3 perturbações modeladas, sendo a Figura B.5(d) uma ilustração do modelo da perturbação total, definida na equação (B.11), quando o veículo faz uma rotação de 360°, no plano horizontal.

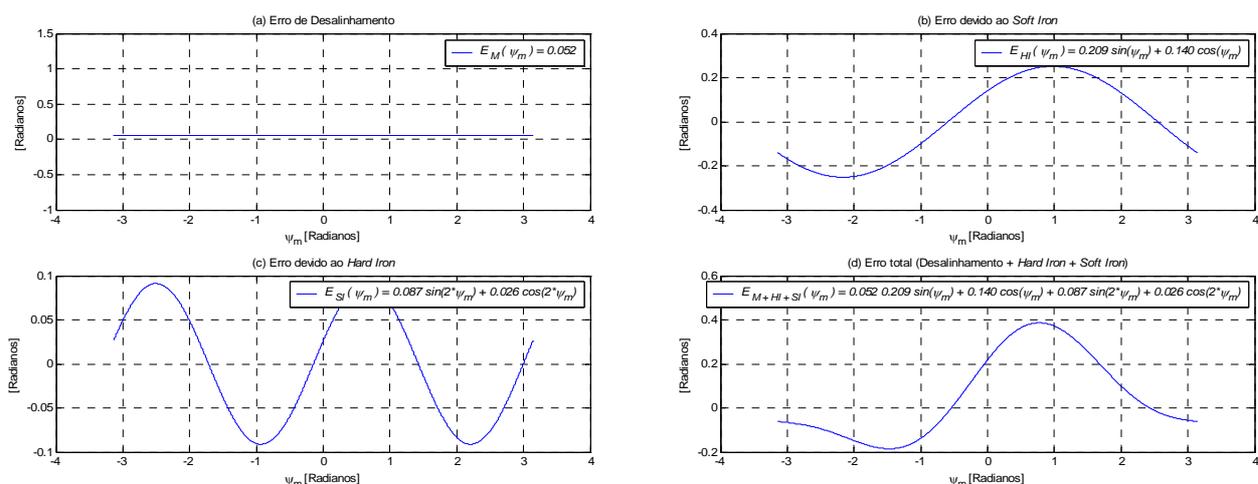


Figura B.5 – Evolução do modelo do erro em função de Yaw medido com a perturbação de desalinhamento (a), de *hard iron* (b), de *soft iron* (c) e total (d)

- **Espaço do Sensor**

Espera-se de um modelo de erro, no espaço do sensor, para além de uma descrição mais rica do fenómeno, a capacidade de modelar o sensor de uma forma mais realista, visto que se actua directamente nas medidas do sensor e não numa qualquer relação entre as mesmas, ou parte delas.

Um novo modelo para o erro no espaço do sensor que alia a teoria apresentada em [Denn79] e em [ElGe01] é sugerido neste trabalho:

$${}^B B_{\text{não ideal}} = {}^M R_B {}^M C_{SF} {}^M R_B \left({}^B C_{SI} {}^B B_{\text{ideal}} + {}^B \delta B_{HI} \right) \quad (\text{B.12})$$

onde os índices B e M designam respectivamente descrição no referencial do corpo $\{B\}$ e descrição no referencial do magnetómetro $\{M\}$ (sensor). As matrizes ${}^M R_B$ e ${}^M C_{SF}$ são definidas mais à frente. Este modelo é apresentado com mais detalhe em seguida.

Em [Denn79] sugere-se a modelação dos efeitos de *hard iron* e de *soft iron* da seguinte forma:

$${}^B B_{HI+SI} = {}^B C_{SI} {}^B B_{\text{ideal}} + {}^B \delta B_{HI} \quad (\text{B.13})$$

onde a matriz ${}^B C_{SI}$ modela os efeitos de *soft iron* em $\{B\}$, ${}^B \delta B$ modela o *offset* provocado pelo fenómeno de *hard iron* em $\{B\}$ e ${}^B B_{\text{ideal}}$ representa o campo magnético verdadeiro sem as perturbações descrito também em $\{B\}$. Em seguida, apresenta-se o modelo anterior evidenciando a estrutura da matriz ${}^B C_{SI}$:

$$\begin{bmatrix} {}^B B_{HI+SI x} \\ {}^B B_{HI+SI y} \\ {}^B B_{HI+SI z} \end{bmatrix} = \begin{bmatrix} (1+a) & b & c \\ d & (1+e) & f \\ g & h & (1+k) \end{bmatrix} \begin{bmatrix} {}^B B_{\text{ideal } x} \\ {}^B B_{\text{ideal } y} \\ {}^B B_{\text{ideal } z} \end{bmatrix} + \begin{bmatrix} {}^B \delta B_{HI x} \\ {}^B \delta B_{HI y} \\ {}^B \delta B_{HI z} \end{bmatrix} \quad (\text{B.14})$$

Os parâmetros a , b , c , d , e , f , g , h e k são responsáveis por modelar os efeitos de *soft iron* e ${}^B\delta B_{HIx}$, ${}^B\delta B_{HIy}$ e ${}^B\delta B_{HIz}$ modelam os efeitos de *hard iron*.

Para trajetórias no plano horizontal (*Roll* e *Pitch* nulos), e para um ângulo *dip* que se supõe igual a 0 (zero), pode-se afirmar que ${}^B B_{idealz}$ é também nulo. Assim sendo, é permitida a redução da dimensão do modelo para:

$$\begin{bmatrix} {}^B B_{HI+SIx} \\ {}^B B_{HI+SIy} \end{bmatrix} = \begin{bmatrix} (1+a) & b \\ d & (1+e) \end{bmatrix} \begin{bmatrix} {}^B B_{idealx} \\ {}^B B_{idealy} \end{bmatrix} + \begin{bmatrix} {}^B \delta B_{HIx} \\ {}^B \delta B_{HIy} \end{bmatrix} \quad (B.15)$$

Os pontos cardeais representados nas Figura B.6, Figura B.7, Figura B.8 e Figura B.9 representam a leitura do magnetómetro quando este está orientado segundo esses mesmos pontos cardeais, sob o efeito da distorção respectiva. Para a análise das perturbações de *hard iron* e de *soft iron*, supõe-se que o referencial $\{B\}$ é coincidente com referencial $\{M\}$, sendo indiferente a utilização do índice B ou do índice M .

Na Figura B.6 apresenta-se, para os parâmetros do modelo descrito anteriormente (${}^B\delta B_{HIx} = 0.3$ e ${}^B\delta B_{HIy} = 0.3$), as perturbações devido ao *hard iron*, respectivamente no referencial do sensor e no espaço de *heading*. No referencial do sensor, observa-se uma translação da circunferência, que está directamente relacionada com a adição às componentes horizontais do vector do campo magnético das componentes horizontais do vector de perturbação de *hard iron*.

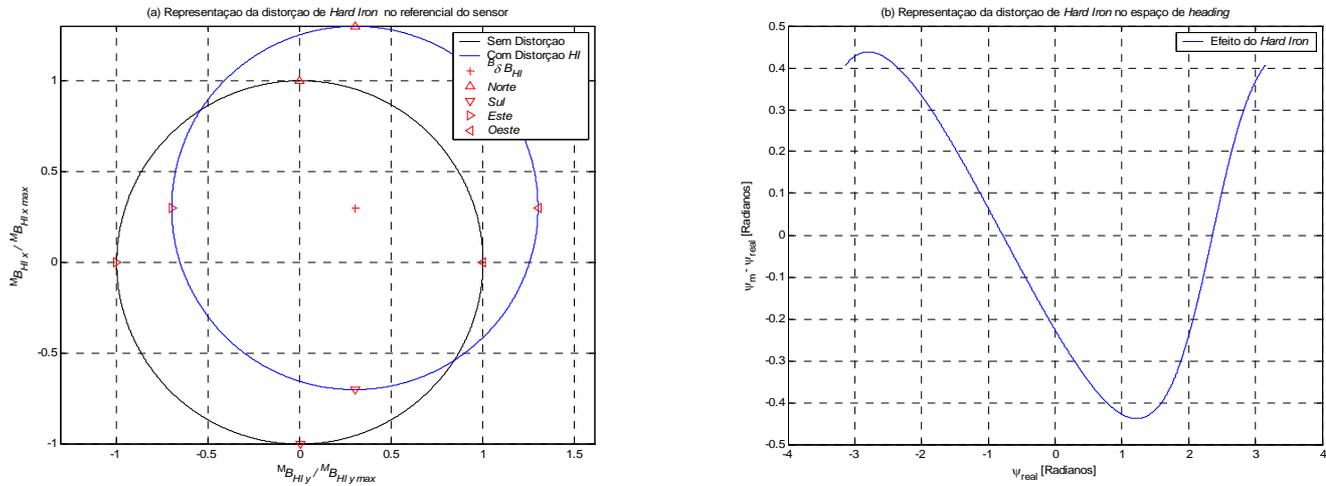


Figura B.6 – Representação da distorção de *Hard Iron* no referencial do sensor (a) e no espaço de *heading* (b)

De salientar, a diferença entre a distorção produzida pela modelação no espaço do sensor e a distorção que se sugere para o espaço de *heading* quando se tenta representar os efeitos de *hard iron*. Ao fazer-se uso dos dados produzidos pelo modelo (B.15) e se obtém os valores de *heading* respectivos, o erro originado não é uma sinusóide perfeita, o que leva a concluir que o modelo no espaço de *heading* é um modelo com aproximações mais grosseiras que o do espaço do sensor [Denn79].

Na Figura B.7 apresenta-se, para os parâmetros do modelo descrito pela equação (B.15) ($a = 0.4$; $b = 0.2$; $d = 0.2$; $e = 0.4$), as perturbações devido ao *soft iron*, respectivamente no referencial do sensor e no espaço de *heading*.

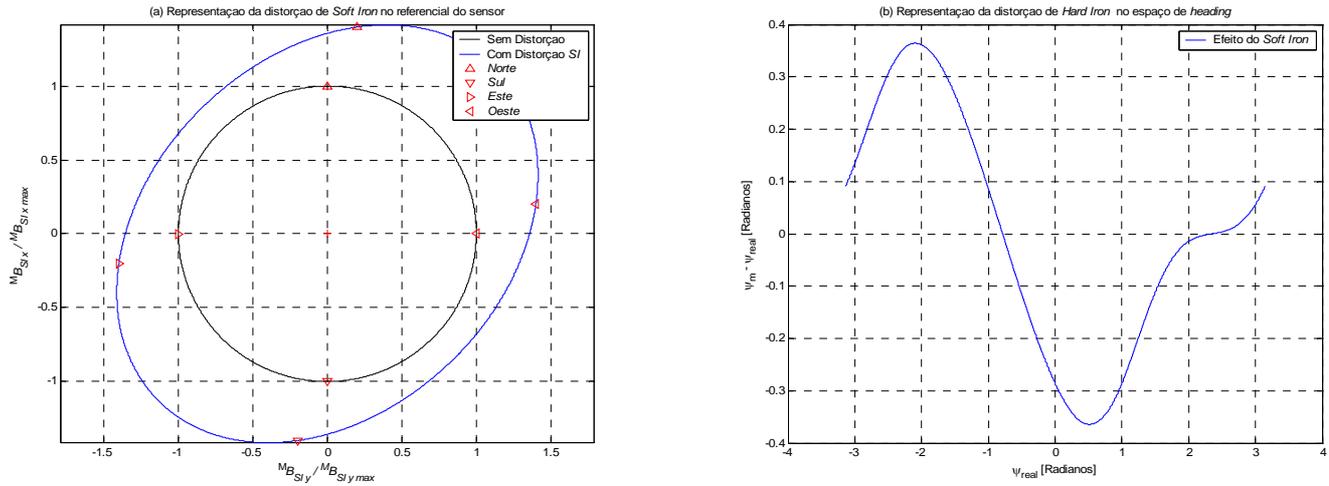


Figura B.7 – Representação da distorção de *Soft Iron* no referencial do sensor (a) e no espaço de *heading* (b)

A perturbação relacionada com o efeito *soft iron* manifesta-se de uma forma diferente da que é originada pelo *hard iron*. Embora também introduza uma polarização nas medições, esta polarização depende do rumo, ao contrário da que é introduzida pelo *hard iron*, que é constante quando descrita no referencial do veículo.

Para se modelar a distorção introduzida por diferentes factores de escala em cada eixo do sensor, modifica-se o modelo do erro, multiplicando pela matriz ${}^M C_{SF}$. Para o conseguir, e supondo que pode existir um desalinhamento entre o referencial do sensor e o referencial do corpo, é necessário descrever ${}^B B_{HI+SI}$ em $\{M\}$. Para tal, é necessário recorrer-se à matriz de rotação ${}^M R_B$:

$${}^M B_{medido} = {}^M C_{SF} {}^M R_B {}^B B_{HI+SI}, \quad \text{com} \quad {}^M C_{SF} = \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} e \quad {}^M R_B = \begin{bmatrix} c\beta & s\beta \\ -s\beta & c\beta \end{bmatrix} \quad (B.16)$$

onde β representa o desalinhamento horizontal entre o referencial do corpo e o referencial do sensor e K_x e K_y representam os factores de escala do magnetómetro para os eixos do x e do y respectivamente.

Na Figura B.8, apresenta-se somente os efeitos de diferentes factores de escala para cada eixo do magnetómetro, quer no espaço de sensor, quer no espaço de *heading*. No caso apresentado ($K_x > K_y$) observa-se que o eixo maior da elipse fica alinhado segundo o eixo do x do sensor, caso contrário ($K_x < K_y$), este ficaria alinhado segundo o eixo do y do sensor.

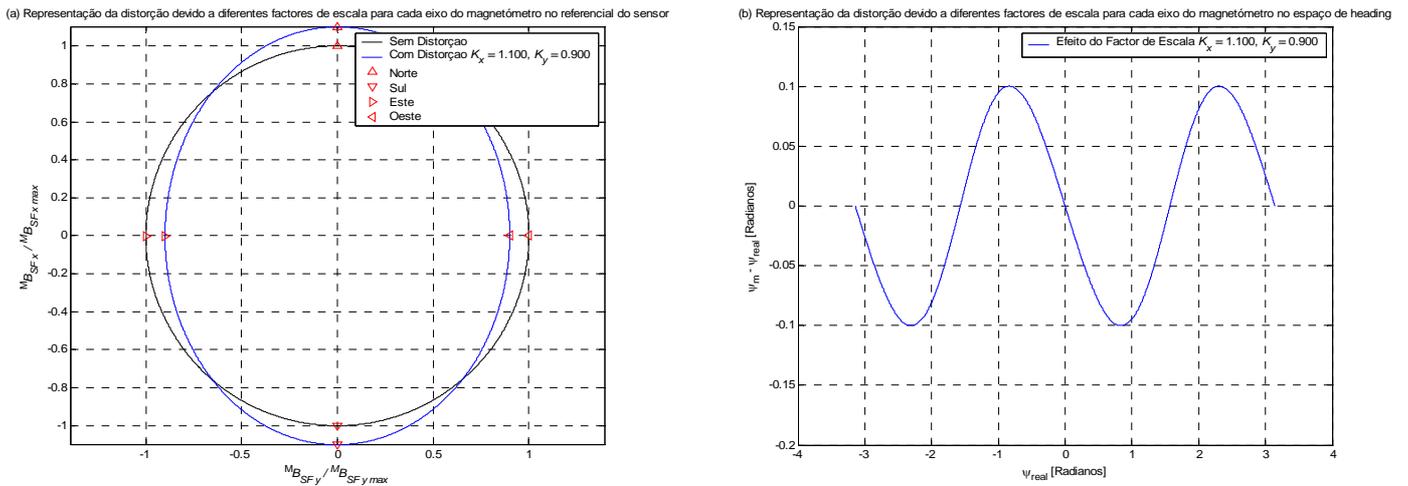


Figura B.8 - Representação da distorção devido a diferentes factores de escala para cada eixo do magnetómetro no referencial do sensor (a) e no espaço de heading (b)

Na Figura B.9, apresenta-se somente os efeitos de uma instalação não rigorosa, isto é, não se conseguir um alinhamento perfeito entre o referencial do sensor e o referencial do corpo. A Figura B.9 (a) ilustra este fenómeno no espaço de sensor, e a Figura B.9 (b) ilustra o mesmo fenómeno no espaço de heading.

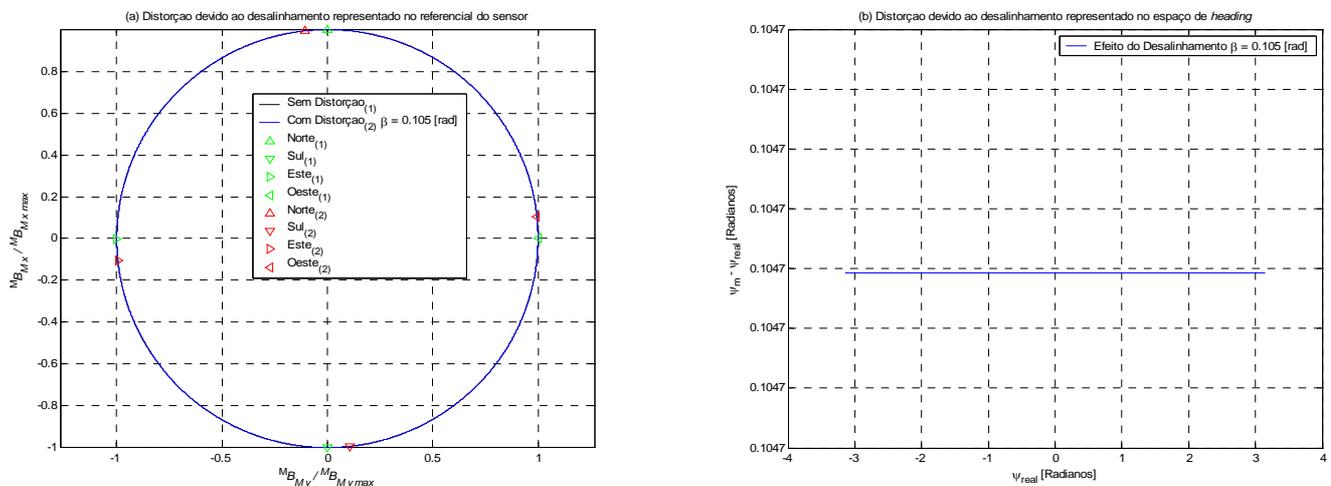


Figura B.9 - Representação da distorção devido ao desalinhamento no plano horizontal entre os eixos do magnetómetro e o referencial do corpo para referencial do sensor (a) e para o espaço de heading (b)

B.1.3.3 Algoritmos de Calibração de um Magnetómetro

No presente trabalho, entende-se por algoritmos de calibração, algoritmos matemáticos, implementados em *software* com o objectivo de obter parâmetros que permitam a correcção, segundo um determinado modelo de erro, das leituras de um magnetómetro solidário com um veículo. Embora existam procedimentos que, por ajuste, ou adição de peças magnéticas, consigam calibrar uma bússola ou magnetómetro, esse não é o objectivo deste trabalho.

- **Critérios de Avaliação de um Algoritmo de Calibração**

- Utilização de referências externas

São desaconselháveis visto que não é simples a determinação, numa qualquer localização remota, de um conjunto mínimo de pontos cardeais com um nível de exactidão elevado.

- Duração da calibração

Não é muito importante como critério de desenvolvimento de algoritmos, desde que se mantenha dentro de limites de duração razoáveis de poucos minutos.

- Utilização somente de dados do magnetómetro ou de outros sensores

É um critério importante, já que se pode estar na presença de *setups* de calibração em que os sensores que, supostamente poderiam ser utilizados como referência, não oferecem um grau de exactidão desejada, ou mesmo a não existência de outro sensor adequado. No caso em que se tem ao dispor sensores que possam contribuir para uma melhor calibração do magnetómetro, é aconselhado o desenvolvimento de algoritmos que tirem partido dessas capacidades.

- Viabilidade de execução da manobra de calibração

Para a extracção de leituras do magnetómetro para calibração, pressupõe-se que se desenvolveu uma manobra que origine valores, no mesmo, que possibilitam o funcionamento do algoritmo de calibração. Estas manobras têm que ser factíveis para o veículo em questão, sendo este um factor determinante para viabilização do algoritmo de calibração.

- **Algoritmos de Calibração para o Espaço de *Heading***

Em geral, os algoritmos de calibração presentes na literatura ([ElGe01] e [Mach03]), passam pela identificação dos parâmetros do modelo do erro definido na equação (B.11), normalmente recorrendo a medidas de *heading* fornecidas por outros sensores previamente calibrados (como por exemplo da integração de um giroscópio ou outros), ou referências externas fazendo depois uma aproximação segundo um critério de minimização que origine os parâmetros que caracterizam o modelo.

Em geral, estes algoritmos são sensíveis a erros originados pela dificuldade de geração de referências externas. Outra característica menos favorável está relacionada com o facto destes métodos só servirem nas aplicações em que o magnetómetro é necessário exclusivamente para a obtenção de medidas de *Yaw*.

- **Algoritmos de Calibração para o Espaço do Sensor**

Pretende-se com o algoritmo que se apresenta em seguida, obter parâmetros que possibilitem a correcção das leituras do magnetómetro. A ênfase vai para a correcção das perturbações causadas pelos fenómenos de *hard iron* e de *soft iron*.

Assume-se que os erros de factor de escala são eliminados por calibração do magnetómetro sem este estar acoplado a um veículo (os factores de escala são característicos do magnetómetro e não do seu ambiente).

Neste caso, o modelo que vai gerar as perturbações nas saídas do magnetómetro para posterior calibração, reduz-se a:

$$\begin{bmatrix} {}^B B_{HI+SI\ x} \\ {}^B B_{HI+SI\ y} \end{bmatrix} = \begin{bmatrix} (1+a) & b \\ d & (1+e) \end{bmatrix} \begin{bmatrix} {}^B B_{ideal\ x} \\ {}^B B_{ideal\ y} \end{bmatrix} + \begin{bmatrix} {}^B \delta B_{HI\ x} \\ {}^B \delta B_{HI\ y} \end{bmatrix} \quad (B.17)$$

Supõe-se que o erro de desalinhamento foi eliminado por uma instalação rigorosa (${}^M R_B$ é uma matriz identidade), e que é negligenciável a diferença entre os factores de escala dos dois eixos do sensor (${}^M C_{SF}$ é uma matriz identidade). Para estas condições, os referenciais $\{B\}$ e $\{M\}$ são coincidentes, sendo indiferente a utilização do índice B ou do índice M .

Este algoritmo de calibração requer que o veículo experimente o maior número possível de valores de *Yaw*, sempre com os ângulos de *Roll* e de *Pitch* nulos (rotação no plano horizontal). Idealmente, os valores obtidos das leituras do eixo do x e do y descreveriam uma circunferência quando representados no plano xy do referencial do sensor (ver Figura B.4), na realidade isso não sucede.

Este algoritmo, tenta transformar o que na realidade se pode obter (uma elipse descentrada e rodada no caso em que as perturbações são mais complexas), no que em condições ideais se deveria obter (uma circunferência centrada na origem), não pretendendo identificar os parâmetros do modelo da equação (B.17).

Para o conseguir, recorre-se ao algoritmo de ajuste de elipses descrito em [FiPi99]. Este algoritmo fornece parâmetros que caracterizam a referida elipse (deslocamento do seu centro em relação à origem - ${}^M \delta \hat{B}_{HI\ x}$ e ${}^M \delta \hat{B}_{HI\ y}$ -, dimensão do eixo maior e do eixo menor e o ângulo - η - do eixo maior em relação ao eixo Y_h do referencial do sensor) que melhor se ajusta aos dados obtidos, segundo o critério dos mínimos quadrados. Depois de obtidos estes parâmetros, consegue-se corrigir ainda no espaço das variáveis do sensor, todas as futuras leituras. Para tal, aplica-se o seguinte conjunto de operações a cada leitura:

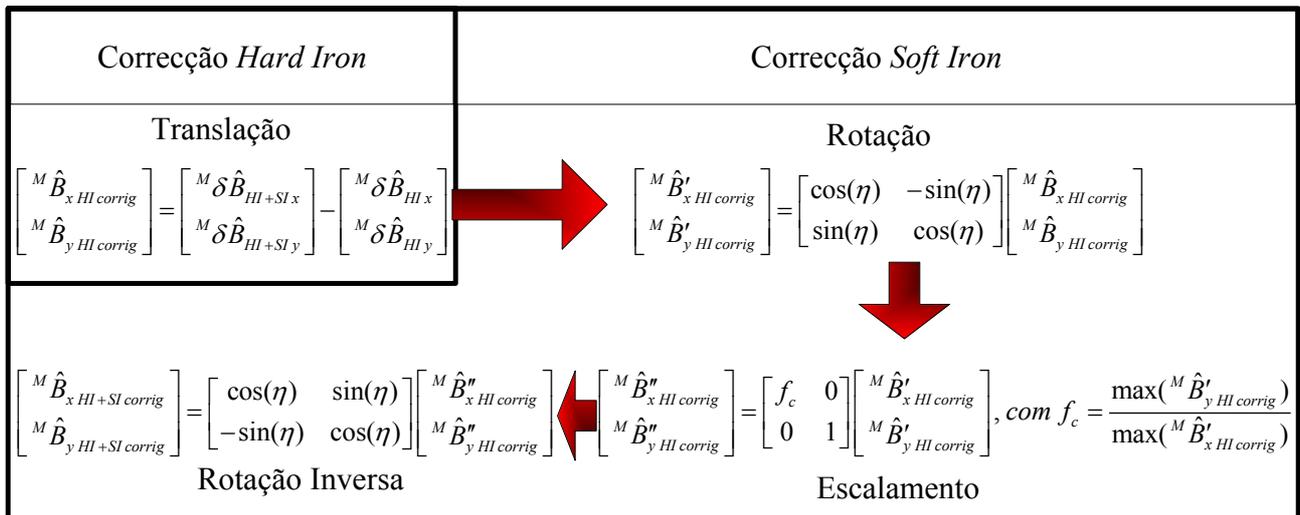


Figura B.10 – Fluxograma do procedimento de correcção das leituras do magnetómetro

Em seguida, na Figura B.11, apresenta-se este procedimento de uma forma gráfica. Para tornar a notação mais simples designa-se o eixo do sensor que coincide com o eixo do x do referencial do corpo por X_h e por Y_h o eixo do sensor que coincide com o referencial do y do mesmo referencial.

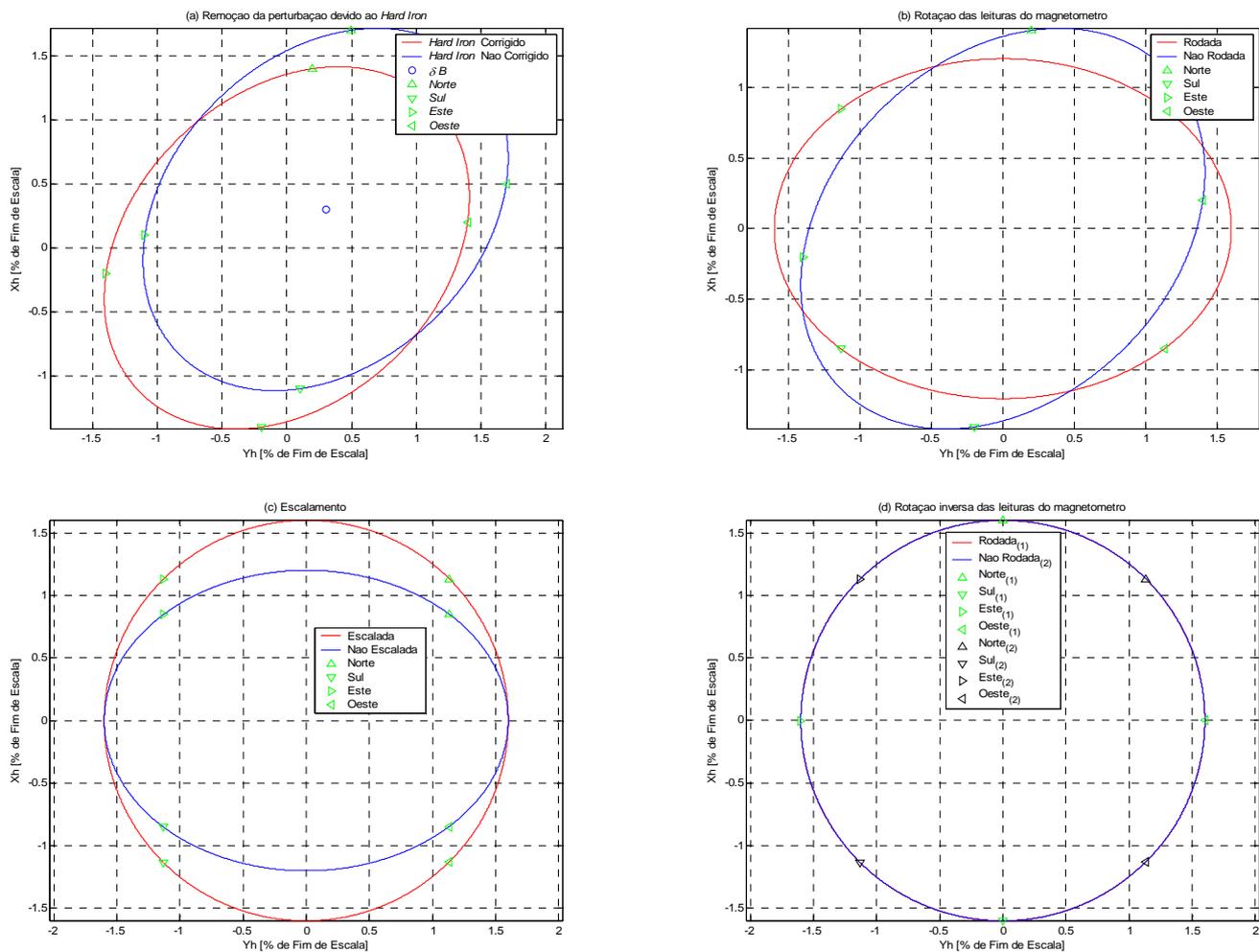


Figura B.11 – Ilustração do processo de correcção das perturbações que afectam as medidas do magnetómetro quando o mesmo completa uma volta de 360° no plano horizontal

Os pontos cardeais representados na Figura B.11 representam a leitura do magnetómetro quando este está orientado segundo esses mesmos pontos cardeais, sob o efeito da distorção e depois da respectiva transformação.

Quando se obtém o valor de *heading* a partir das leituras do magnetómetro, antes e depois da correcção, observa-se uma melhoria significativa, tal como se demonstra na Figura B.12.

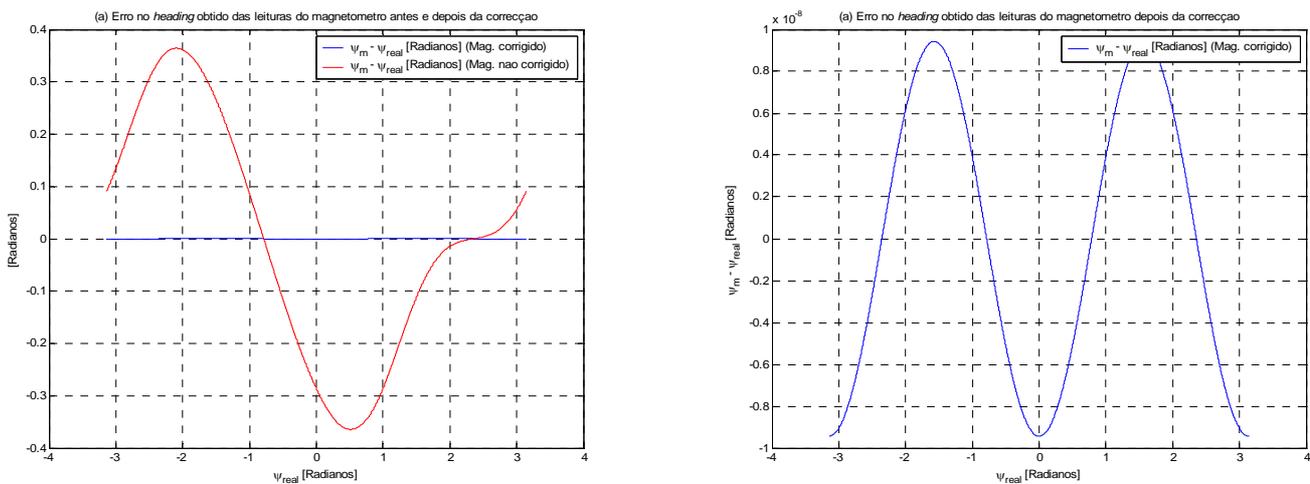


Figura B.12 – Erro na obtenção da medida de heading a partir do magnetómetro antes (a) e depois da correcção (b)

Os resultados provam a validade algorítmica deste método de calibração. Mais à frente, apresentam-se resultados de simulação que evidenciam o aumento da performance ao nível da obtenção do heading recorrendo às medidas corrigidas do magnetómetro.

Este algoritmo, para além de não se basear num modelo de erro aproximado, tem as seguintes vantagens:

- As correcções são feitas directamente nas saídas do magnetómetro e não em outras medidas obtidas com base nas primeiras
- Não necessita de outros sensores
- Obriga à execução de uma manobra de calibração, que para a generalidade dos veículos, é simples de executar

Como desvantagens, pode-se afirmar que:

- O algoritmo de calibração pode ser pesado quer ao nível da capacidade computacional, quer ao nível de utilização de memória, que conforme o sistema utilizado, se pode tornar num problema ou não
- A performance do algoritmo degrada-se com o aumento dos ângulos de Roll e de Pitch

Considera-se que, para veículos com trajectórias predominantemente no plano horizontal, este algoritmo de calibração produza bons resultados.

• **Implementação e Testes em Ambiente de Simulação**

Depois de se obter a validação algorítmica do método de calibração e posterior correcção, interessa testar, em ambiente de simulação, o referido procedimento. Mais uma vez, recorreu-se ao simulador apresentado no Apêndice F para gerar uma trajectória, bem como para modelar os sensores. A trajectória permite testar o algoritmo nas condições para as quais este foi desenvolvido (ângulos de Roll e de Pitch nulos). Depois de executada a manobra de calibração e de se terem obtido os parâmetros necessários à correcção das leituras do magnetómetro ao longo da trajectória definida, pode-se iniciar a sua simulação. Na Figura B.13(a) apresenta-se a orientação do veículo (o magnetómetro está solidário com o veículo) durante a trajectória executada. A Figura B.13 (b) e a Figura B.13 (c) mostram os erros no cálculo de heading em função do heading verdadeiro e em função do tempo respectivamente.

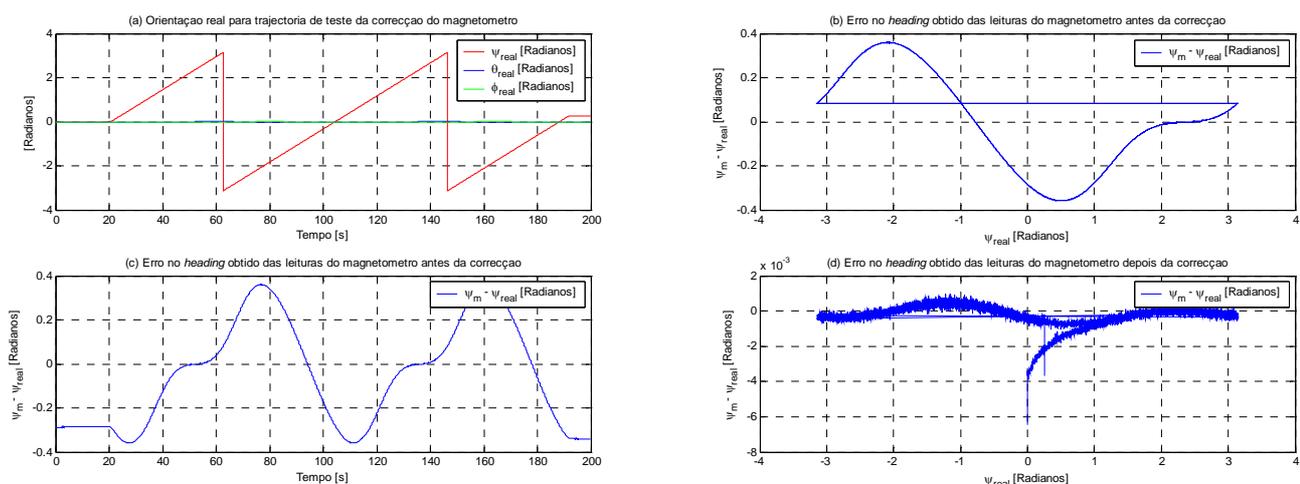


Figura B.13 – Impacto da calibração do magnetómetro no cálculo do *heading*

Comparando a Figura B.13 (b) e Figura B.13 (d) observa-se, para as condições da simulação, uma eliminação quase total do erro no cálculo do *heading* da plataforma, a partir das medições dos eixos do magnetómetro no plano horizontal. Embora estes testes tenham sido efectuados em ambiente de simulação, a codificação do algoritmo foi feita em ANSI C. Portanto, permite que, em trabalhos futuros, seja implementado no Computador de Bordo. O algoritmo foi também testado com leituras reais do magnetómetro. Estes testes sofriam de uma limitação ao nível da capacidade de validar os resultados produzidos pelo algoritmo, já que, não se dispunha de uma medida de *heading* fiável.

- **Considerações Finais sobre Algoritmos de Calibração**

Quando se executa um procedimento de calibração, é aconselhável ter presente que os resultados do mesmo só são válidos para as mesmas condições, ou seja, para os mesmos sensores, veículo e instalação nesse mesmo veículo. Quando se altera a instalação do magnetómetro, é de esperar que o procedimento de calibração produza resultados diferentes.

B.2 Sensores Auxiliares

B.2.1 GPS

GPS é uma rede de 24 satélites numa órbita da Terra de aproximadamente 20'000km, que pode ser usada para fornecer informação sobre a localização de um receptor de sinal na superfície da Terra (Figura B.14). Cada satélite emite um sinal, em duas frequências, contendo informação sobre a sua posição e instante temporal do envio do sinal.



Figura B.14 – Configuração Orbital dos Satélites GPS (extraído de [lectGPS])

O receptor recebe o sinal de cada satélite que permite determinar a distância ao satélite. As coordenadas de um receptor de GPS são determinadas por trilateração, no plano ou no espaço, em função do número de satélites a que se tem acesso, dependendo da localização espaço-temporal do receptor (Figura B.15). Se quatro destas distâncias podem ser medidas, então é possível calcular a localização tri-dimensional do receptor em relação aos satélites, e sendo a posição dos satélites conhecida então pode-se determinar a posição absoluta.

A disponibilidade do sistema significa que, actualmente, qualquer pessoa pode determinar a sua localização na superfície da terra, ou mesmo, navegar.

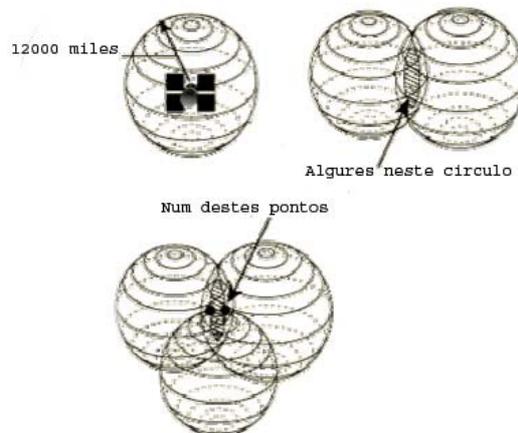


Figura B.15 – Trilateração planar com desfasagem temporal (extraído de [lectGPS])

O receptor utilizado disponibiliza para o sistema de navegação as medidas de latitude, longitude e altitude. Neste caso apenas se tem de converter estas medidas para XYZ no referencial inercial local.

B.2.2 Estimador de Atitude Magneto-Pendular

O estimador de atitude baseia-se no princípio do inclinómetro, segundo o qual se pode determinar a inclinação de um corpo através do ângulo formado entre um pêndulo e a vertical (ver Figura B.16). Para o trabalho presente, em lugar do pêndulo utiliza-se uma tríade de acelerómetros, em que a aceleração gravítica corresponde à vertical.

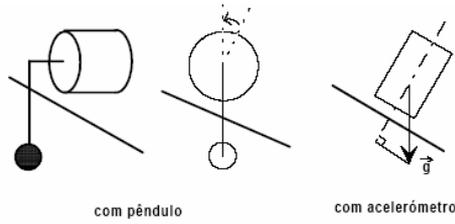


Figura B.16 – Inclinómetro (extraído de [Azin02])

Utilizando este princípio apenas é possível determinar, em ângulos de Euler, os ângulos de *Pitch* e *Roll*, porque a informação fornecida por este é independente da rotação segundo o eixo normal à superfície terrestre. Para determinar o ângulo *Yaw* recorreu-se à componente horizontal do campo magnético, fornecida pelo magnetómetro.

- **Leitura da Componente Gravítica nos Acelerómetros**

Geralmente, o movimento oscilatório do pêndulo é amortecido de modo a estabilizar a leitura do inclinómetro sujeito a acelerações externas. Assim, optou-se por filtrar o sinal dos acelerómetros de modo a atenuar este efeito.

No entanto, a leitura do acelerómetro não poderá ser directa, pois se o movimento for uniforme e circular, a leitura vem afectada pela aceleração centrípeta ${}^B({}^I\omega_B) \times {}^B({}^IV_{BORG})$. Portanto, para que os resultados do estimador sejam válidos para qualquer movimento, rectilíneo ou de rotação, é necessário remover a aceleração centrípeta:

$$a_p = a - {}^B({}^I\omega_B) \times {}^B({}^IV_{BORG}) \quad (B.18)$$

onde a_p é a entrada do estimador e a a leitura do acelerómetro.

O termo ${}^B({}^I\omega_B)$ é dado pelas medidas dos giroscópios e o termo ${}^B({}^IV_{BORG})$ pode, por exemplo, ser estimado recorrendo ao filtro complementar de posição (Apêndice C) ou, no caso do filtro de kalman estendido, à estimativa existente no INS.

A remoção da aceleração centrípeta não está livre de erros, no entanto, analisando as condições de operação do veículo, observa-se que as acelerações tangenciais são de curta duração devido à compensação dos atritos e que a aceleração gravítica é muito superior às acelerações de funcionamento do veículo, tornando viáveis as leituras do sensor para baixas frequências.

- **Determinação da Inclinação**

Através da matriz de rotação (A.7) e da propriedade da sua inversa (A.3), obtém-se a expressão das projecções da força gravítica nos acelerómetros:

$$a_p = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \left({}^E R({}^I\Gamma)^{-1} \right)^E g = \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{bmatrix} \quad (B.19)$$

onde ${}^E g = [0 \ 0 \ g]^T$ é a aceleração gravítica vista no referencial da Terra.

Por manipulação de (B.19), obtém-se as expressões dos ângulos:

○ *Roll*:

$$\frac{a_y}{a_z} = \frac{g \cos \theta \sin \phi}{g \cos \theta \cos \phi} \Rightarrow \phi = \text{atan} \left(a_y, a_z \right) \quad (\text{B.20})$$

○ *Pitch*:

$$\begin{cases} \sin \theta = -\frac{a_x}{g} \\ \cos \theta = \frac{a_y}{g \sin \phi} \\ \cos \theta = \frac{a_z}{g \cos \phi} \end{cases} \Rightarrow \begin{cases} \theta = a \tan 2 \left(-a_x, \frac{a_y}{\sin \phi} \right), & \sin \phi \neq 0 \\ \theta = a \tan 2 \left(-a_x, \frac{a_z}{\cos \phi} \right), & \cos \phi \neq 0 \end{cases} \quad (\text{B.21})$$

Os ângulos de Euler permitem duas soluções para a mesma orientação, portanto é necessário em (B.20) convencionar $\phi \in \left[-\pi/2, \pi/2 \right]$. As expressões em (B.21) são equivalentes, optando-se pela que tiver maior precisão.

• Determinação do Ângulo de *Yaw*

A estimação de *Yaw* é obtida através da projecção da leitura do magnetómetro num referencial auxiliar $\{E_a\}$. Os referenciais $\{E_a\}$ e $\{E\}$ apenas diferem numa rotação em torno de z (*Yaw*).

A componente horizontal da leitura do magnetómetro é obtida projectando o campo magnético segundo os ângulos *Pitch* e *Roll* determinados anteriormente:

$${}^E_a B = R_Y(\theta) R_X(\phi) {}^B B \quad (\text{B.22})$$

onde ${}^B B = [B_x \ B_y \ B_z]^T$ é a leitura do magnetómetro e ${}^E_a B$ é a leitura do magnetómetro vista em $\{E_a\}$.

O ângulo de *Yaw* é dado pela rotação do vector campo magnético, sendo necessário compensar a declinação da terra α_{dec} :

$$\psi = \arctan 2 \left(-{}^E_a B_y, {}^E_a B_x \right) - \alpha_{dec} \quad (\text{B.23})$$

É necessário converter a estimação de *Yaw* do referencial $\{E\}$ para o referencial $\{I\}$. A orientação do referencial $\{E\}$ em relação a $\{I\}$ é afectada por uma velocidade angular constante e conhecida, resultante da rotação da Terra ${}^I \omega_E = [0 \ 0 \ \Omega]$. O vector de rotação surge da integração directa da

velocidade angular ${}^I \lambda(t) = \int_0^t {}^I \omega_E d\tau = {}^I \omega_E$. A discretização é directa:

$$\begin{aligned}
{}^I_E \lambda(t_k) &= {}^I_E \lambda(t_{k-1}) + \Delta {}^I_E \lambda(t_k) \\
\Delta {}^I_E \lambda(t_k) &= {}^I \omega_E T_k
\end{aligned}
\tag{B.24}$$

onde k é o índice do instante de amostragem e $T_k = t_k - t_{k-1}$ é o período de amostragem.

A matriz de rotação é então calculada de acordo com (A.12). Usando a composição de transformações, obtém-se a orientação do referencial $\{B\}$ em relação a $\{I\}$, ${}^I_B R = {}^I_E R {}^E_B R$.

A matriz ${}^E_B R$ é calculada de acordo com (A.7), usando os ângulos estimados. Por sua vez, a matriz ${}^I_B R$ pode ser convertida em ângulos de Euler através de (A.8).

C Sistema de Navegação com recurso a Filtros de Kalman Complementares

Aqui apresenta-se um complemento ao Capítulo 4, descrevendo com algum pormenor os filtros de atitude e posição para o filtro de Kalman complementar. Para maior detalhe sobre os algoritmos consultar [CaVa03].

C.1 Filtro Complementar de Atitude

Para determinar a atitude do veículo foi implementado um filtro complementar para cada eixo de rotação. Os giroscópios medem a velocidade de rotação do referencial do corpo {B} em relação ao referencial inercial {I}, vista no referencial {B}:

$${}^B \left({}^I \omega_B \right) = [p \quad q \quad r]^T \quad (C.1)$$

para obter uma medida da variação de atitude em ângulos de Euler utiliza-se a relação:

$$\dot{\Gamma} = Q(\Gamma) {}^B \left({}^I \omega_B \right) \quad (C.2)$$

Fazendo uso do sensor magneto-pendular obtém-se uma medida directa da atitude (*Yaw*, *Pitch* e *Roll*). Estas medidas estão sujeitas a perturbações, que podem ser representadas por um termo aditivo, para além desta perturbação os giroscópios apresentam ainda polarizações $\Lambda = [\alpha_p \quad \alpha_q \quad \alpha_r]^T$, que podem ser representadas na medida de variação de atitude, também por um termo aditivo, $\Upsilon = Q(\Gamma)\Lambda$. Resultando:

$$y = \Gamma + y_d \quad u = \dot{\Gamma} + u_d + \Upsilon \quad (C.3)$$

onde y é a medida de atitude perturbada por y_d e u é a medida de variação de atitude perturbada por u_d e pelas polarizações dos giroscópios Υ .

A partir de (4.1) e introduzindo as medidas das grandezas (C.3), obtém-se uma estimativa possível para Γ :

$$\hat{\Gamma}(s) = \Gamma(s) + \frac{k}{s+k} TL\{y_d\} + \frac{1}{s+k} TL\{u_d\} \quad (C.4)$$

O filtro funciona como um observador de Luenberger, ou seja, utiliza-se a leitura de um sensor como entrada do sistema e outro como a saída do mesmo [Luen64]. Após introdução de estados adicionais para modelar e estimar o termo aditivo Υ , obtém-se o modelo:

$$\begin{cases} \begin{bmatrix} \dot{\hat{\Gamma}} \\ \dot{\hat{Y}} \end{bmatrix} = \begin{bmatrix} 0 & I_{3 \times 3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\Gamma} \\ \hat{Y} \end{bmatrix} + \begin{bmatrix} I_{3 \times 3} \\ 0 \end{bmatrix} u + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} (y - \hat{\Gamma}) \\ \hat{\Gamma} = \begin{bmatrix} I_{3 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \hat{\Gamma} \\ \hat{Y} \end{bmatrix} \end{cases} \quad (\text{C.5})$$

em que, u e y são as medidas apresentadas ao filtro, $\hat{x} = \begin{bmatrix} \hat{\psi} & \hat{\theta} & \hat{\phi} & | & \hat{Y} \end{bmatrix}^T$ são as variáveis de estado do modelo, $K_1 = I_{3 \times 3} k_{1\Gamma}$ e $K_2 = I_{3 \times 3} k_{2\Gamma}$ são os ganhos de cada realimentação.

A discretização do filtro é efectuada por aproximação *Zero Order Hold* (ZOH) [Ribe02], obtendo-se:

$$\begin{cases} \begin{bmatrix} \hat{\Gamma}_{k+1} \\ \hat{Y}_{k+1} \end{bmatrix} = \Phi_k \begin{bmatrix} \hat{\Gamma}_k \\ \hat{Y}_k \end{bmatrix} + B_k u_k + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} (y_k - \hat{\Gamma}_k) \\ \hat{\Gamma}_k = \begin{bmatrix} I_{3 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \hat{\Gamma}_k \\ \hat{Y}_k \end{bmatrix} \end{cases} \quad (\text{C.6})$$

onde $\Phi_k = e^{AT} = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} T \\ 0 & I_{3 \times 3} \end{bmatrix}$, $B_k = \int_0^T e^{A\sigma} B d\sigma = \begin{bmatrix} I_{3 \times 3} T \\ 0 \end{bmatrix}$ e T é o período de amostragem do sistema.

Apresentado o filtro, o seu desenho é então reduzido ao cálculo do vector de ganhos $K = [K_1 \quad K_2]^T$ que são calculados recorrendo às expressões para ganhos óptimos estacionários de um observador de Kalman $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$, onde P_k é a matriz de covariância do erro, determinada pela equação discreta de Riccati, a matriz H_k é obtida pelo modelo de estado discreto e Q , R são matrizes que pesam as entradas dos sistema, que se usam para afinar os ganhos (extraído de [CaVa03]).

Como o filtro trabalha com ângulos de Euler Z-Y-X e a polarização dos giroscópios é definida no espaço do vector velocidade angular, é necessário proceder a uma mudança de espaço de integração da polarização através de (C.2) para que esta seja calculada no espaço original.

A introdução desta transformação variante no tempo e dependente das variáveis de estado afecta a estabilidade do filtro e torna-o não linear. No entanto, testes exaustivos ao filtro revelaram a ausência de instabilidade. Outro problema da alteração introduzida é levar a que os ganhos calculados deixem de ser óptimos [CaVa03]. O modelo final do filtro de atitude apresenta-se na Figura C.1:

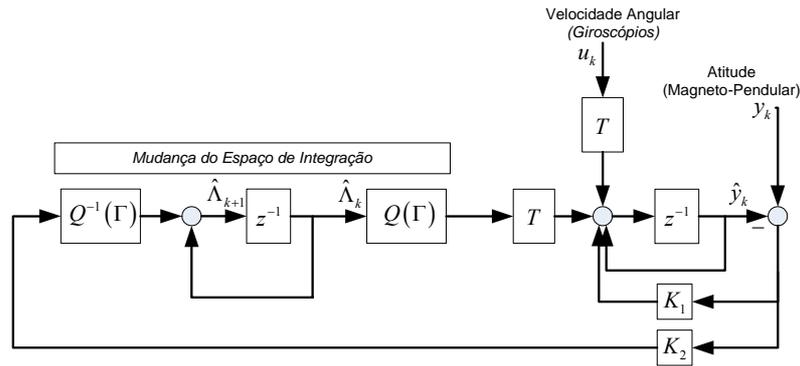


Figura C.1 – Filtro complementar de atitude

C.2 Filtro Complementar de Posição

O filtro de posição recorre ao GPS, obtendo assim uma medida de posição, e aos acelerómetros, que permitem obter uma medida de aceleração que terá que ser duplamente integrada. Os diferentes ritmos de amostragem das medidas exigem uma solução multi-ritmo.

Os acelerómetros medem a força específica, i.e., a partir do referencial $\{B\}$, determinam a variação da velocidade de $\{B\}$ em relação a $\{I\}$, afectada pela aceleração gravítica, $\left(a = \left(\frac{d^1 V_{BORG}}{dt} \right) + {}^B g \right)$.

Conhecendo uma estimativa da atitude é possível remover a componente gravítica, obtendo a medida a introduzir no filtro:

$$u = \left(\frac{d^1 V_{BORG}}{dt} \right) = {}^I R(\Gamma) (a - {}^B R(\Gamma) {}^I g) = {}^I R(\Gamma) a - {}^I g \quad (C.7)$$

sendo ${}^I g$ o vector aceleração gravítica representado no referencial $\{I\}$. À semelhança dos giroscópios, assume-se que as polarizações dos acelerómetros se podem representar por um termo aditivo, $\Upsilon = {}^I R(\Gamma) \Lambda$. Resultando:

$$u = {}^I R(\Gamma) a - {}^I g + \Upsilon \quad (C.8)$$

Partindo de considerações análogas ao modelo do filtro de atitude e, depois de introduzir os estados adicionais associados às polarizações Λ , obtém-se o filtro complementar de posição contínuo:

$$\left\{ \begin{array}{l} \begin{bmatrix} \dot{\hat{P}} \\ \dot{\hat{V}} \\ \dot{\hat{\Gamma}} \end{bmatrix} = \begin{bmatrix} 0 & I_{3 \times 3} & 0 \\ 0 & 0 & I_{3 \times 3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\Gamma} \\ \hat{V} \\ \hat{Y} \end{bmatrix} + \begin{bmatrix} 0 \\ I_{3 \times 3} \\ 0 \end{bmatrix} u + \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} (y - \hat{P}) \\ \\ \hat{P} = [I_{3 \times 3} \quad 0 \quad 0] \begin{bmatrix} \hat{\Gamma} \\ \hat{V} \\ \hat{Y} \end{bmatrix} \end{array} \right. \quad (C.9)$$

em que, u e y são as medidas de aceleração e de posição apresentadas ao filtro, $\hat{x} = \left[\hat{p}_x \quad \hat{p}_y \quad \hat{p}_z \mid \hat{v}_x \quad \hat{v}_y \quad \hat{v}_z \mid \hat{\Gamma} \right]^T$ são as variáveis de estado do filtro e $K_1 = I_{3 \times 3} k_{1p}$, $K_2 = I_{3 \times 3} k_{2p}$ e $K_3 = I_{3 \times 3} k_{3p}$ são os ganhos de cada realimentação.

A discretização do filtro é efectuada por aproximação *Zero* (ZOH), obtendo-se:

$$\left\{ \begin{array}{l} \begin{bmatrix} \hat{P}_{k+1} \\ \hat{V}_{k+1} \\ \hat{\Gamma}_{k+1} \end{bmatrix} = \Phi_k \begin{bmatrix} \hat{P}_k \\ \hat{V}_k \\ \hat{\Gamma}_k \end{bmatrix} + B_k u_k + \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} (y_k - \hat{P}_k) \\ \\ \hat{P}_k = [I_{3 \times 3} \quad 0 \quad 0] \begin{bmatrix} \hat{P}_k \\ \hat{V}_k \\ \hat{\Gamma}_k \end{bmatrix} \end{array} \right. \quad (C.10)$$

onde $\Phi_k = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} T & I_{3 \times 3} \frac{T^2}{2} \\ 0 & I_{3 \times 3} & I_{3 \times 3} T \\ 0 & 0 & I_{3 \times 3} \end{bmatrix}$, $B_k = \int_0^T e^{A\sigma} B d\sigma = \begin{bmatrix} I_{3 \times 3} \frac{T^2}{2} \\ I_{3 \times 3} T \\ 0 \end{bmatrix}$ e T é o período de amostragem do sistema.

Os ganhos de realimentação K_1 , K_2 e K_3 são calculados recorrendo às expressões para ganhos óptimos de um observador de Kalman multi-ritmo [CaVa03].

O filtro trabalha no referencial $\{I\}$ e a polarização dos acelerómetros é definida no referencial $\{B\}$, portanto é necessário proceder a uma mudança de espaço recorrendo à matriz de rotação ${}^I_B R(\Gamma)$ [CaVa03].

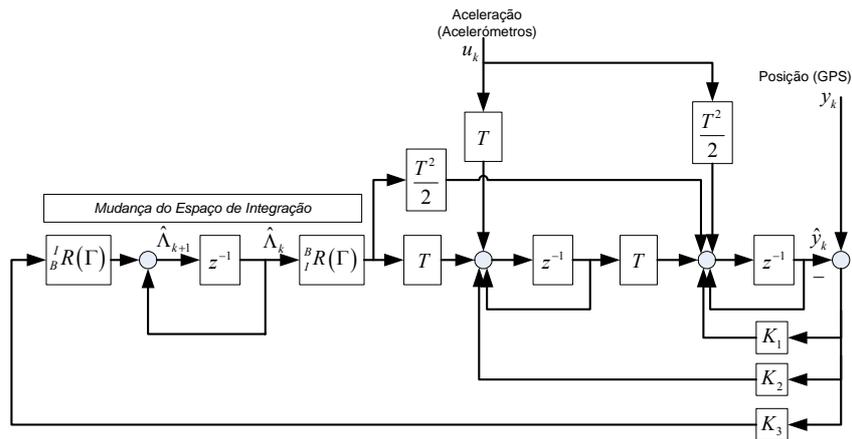


Figura C.2 – Filtro complementar de posição

C.3 Sistema de Navegação Global

O sistema de navegação, para além de estimar a posição e a atitude, estima as polarizações dos acelerómetros e giroscópios. Observe-se o diagrama da Figura C.3. No entanto, só a polarização estimada dos giroscópios, $\hat{\Gamma}_{giros}$, é realimentada e, desta forma, utilizada para corrigir as leituras dos mesmos. Isto porque a dupla realimentação levaria o sistema a evoluir para um ponto de equilíbrio diferente do pretendido.

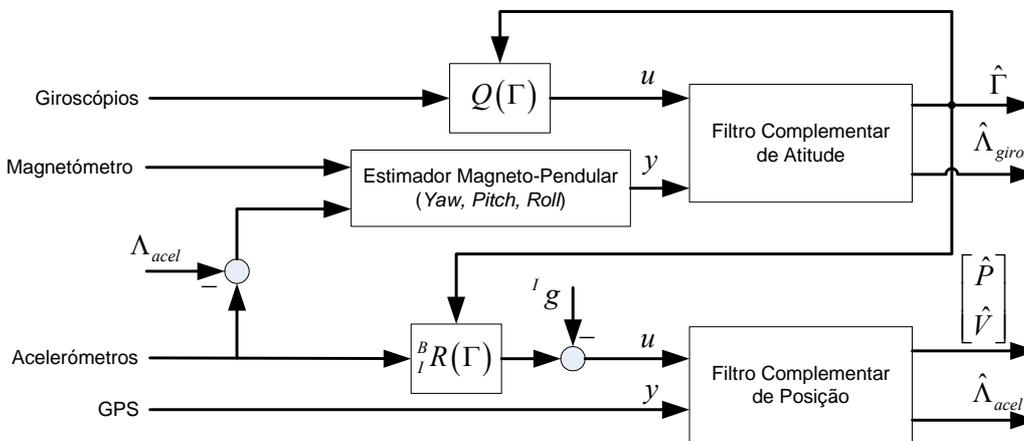


Figura C.3 – Diagrama do Sistema de Navegação Global com filtros complementares

D Sistema de Navegação Inercial

Ao longo dos tempos, a síntese de algoritmos para INS tem resultado de um compromisso entre a exactidão e a capacidade computacional disponível. A maioria das soluções recorre a uma estrutura multi-ritmo, com um algoritmo de frequência elevada, mas de ordem reduzida, de modo a contabilizar a dinâmica de alta-frequência e um algoritmo de frequência moderada para introduzir computações de ordem superior, mais lentas, baseadas nos resultados da do ritmo de alta-frequência.

A solução de INS utilizada encontra-se descrita com mais pormenor em [CaVa03] e descreve a atitude através de quaterniões. Relativamente a outras soluções, em que a estimativa da atitude é feita recorrendo à Matriz dos Cosenos Directores (DCM), esta permite uma equação de dinâmica linear, cuja solução resultante é exacta e requer uma capacidade computacional bem mais reduzida, o que permite que seja implementada a um só ritmo. A exactidão dos resultados depende, em última análise, do *hardware* disponível, já que o que está em causa é a ordem de aproximação dos termos trigonométricos.

Integrando os valores dos acelerómetros consegue-se obter a velocidade e a posição vistos no referencial do corpo, que são posteriormente convertidos em velocidade e posição no referencial inercial, através da orientação calculada no algoritmo de atitude.

A simplicidade computacional do algoritmo de atitude permite resultados exactos a um ritmo elevado, disponibilizando valores de atitude com uma frequência suficiente para que o cálculo da velocidade e da posição sejam também executados a um só ritmo, sem recurso a extrapolações.

D.1 Algoritmo de Atitude

A seguinte equação diferencial, variante no tempo, traduz a dinâmica da atitude descrita por quaterniões:

$$\frac{d {}^I_B q}{dt} = \frac{1}{2} A(\Omega) {}^I_B q, \quad A(\Omega) = \begin{bmatrix} 0 & r & -q & p \\ -r & 0 & p & q \\ q & -p & 0 & r \\ -p & -q & -r & 0 \end{bmatrix} \quad (D.1)$$

e ${}^I_B q = [q_1 \quad q_2 \quad q_3 \quad q_4]^T$ é o quaterniões que representa a orientação do referencial {B} em relação ao referencial inercial, $\Omega = {}^B ({}^I \omega_B) = [p \quad q \quad r]^T$ é a velocidade angular do referencial {B} em relação ao referencial {I} vista no referencial {B}. A expressão (D.1) permite relacionar as variações de atitude com as leituras dos giroscópios. A sua solução, retirada de [CaVa03], é aqui apresentada:

$$\Phi_k = \cos\left(\frac{\|W_k\|}{2}\right) I_3 + \frac{1}{\|W_k\|} \sin\left(\frac{\|W_k\|}{2}\right) A(W_k), \quad W_k = \int_{t_{k-1}}^{t_k} \Omega d\tau \quad (D.2)$$

Se a frequência de amostragem for elevada relativamente à dinâmica do corpo, a variação da velocidade angular do corpo, num período de amostragem, será sempre muito reduzida. Isto faz com que o integral de (D.2) possa ser calculado pela aproximação trapezoidal:

$$W_k \approx \frac{1}{2}(\Omega_k + \Omega_{k-1})T_k \quad (D.3)$$

onde $T_k = t_k - t_{k-1}$ corresponde ao período de amostragem.

Esta solução permite determinar a orientação do referencial {B} em relação ao referencial inercial, representado por um quaternião e, conseqüentemente, sem a existência de singularidades, ao contrário do que acontece com a formulação DCM. De referir ainda que, o grau de exactidão dos resultados depende principalmente da ordem de aproximação das funções trigonométricas de (D.2) (pelo que existe uma dependência indirecta do *hardware* utilizado) e pela frequência de amostragem que, deverá ser suficientemente elevada de modo a entrar em conta a dinâmica de alta-frequência do corpo.

D.2 Algoritmo de Velocidade

Este algoritmo recorre à transformação da força específica para o referencial inercial e à modelação da componente gravítica para integrar a aceleração total em velocidade e posição. A força específica consiste nas leituras dos acelerómetros, pelo que se trata da soma da aceleração inercial com a componente gravítica:

$$a = \left(\frac{d^1 V_{BORG}}{dt} \right) + {}^B g \quad (D.4)$$

Assim, integrando o resultado correspondente a retirar a componente gravítica à leitura dos acelerómetros, obtém-se a velocidade do veículo vista no referencial inercial. Reescrevendo (D.4) em função da velocidade:

$$v(t) = \int_0^t {}^I_B R (a - {}^B g) d\tau \quad (D.5)$$

onde $v = {}^I V_{BORG}$ é a velocidade da origem do referencial do corpo (P_{BORG}) no referencial inercial. De (D.5) resulta que a relação da velocidade entre dois instantes de amostragem é dada por:

$$v_k = v_{k-1} + \Delta v_{G,k} + \Delta v_{SF,k}, \quad \Delta v_{G,k} = -\int_{t_{k-1}}^{t_k} {}^I g d\tau \quad \text{e} \quad \Delta v_{SF,k} = \int_{t_{k-1}}^{t_k} {}^I_B R d\tau \quad (D.6)$$

onde k é o índice do instante de amostragem t_k .

Tal como no caso anterior, a discretização do algoritmo é desenvolvida recorrendo à aproximação trapezoidal:

$$\Delta v_{G,k} \approx \frac{1}{2}({}^I g_k + {}^I g_{k-1})T_k, \quad \Delta v_{SF,k} \approx \frac{1}{2}({}^I a_k + {}^I a_{k-1})T_k \quad (D.7)$$

onde ${}^I a = {}^I R a$ é a leitura dos acelerómetros vista no referencial inercial e T_k representa o período de amostragem.

Novamente, este resultado pressupõe que a frequência de amostragem é suficientemente elevada de modo a contabilizar as componentes de alta-frequência da velocidade em $\Delta v_{SF,k}$. Por sua vez, $\Delta v_{G,k}$ é correctamente calculada para qualquer período de amostragem projectado para $\Delta v_{SF,k}$, já que a dinâmica de ${}^I g_k$ se relaciona directamente com a velocidade de rotação terrestre, ${}^I \omega_E$, muito inferior à do veículo.

D.3 Algoritmo de Posição

Este algoritmo baseia-se no algoritmo de velocidade e os seus resultados vêm da integração da velocidade:

$$x(t) = \int_0^t v d\tau \quad (D.8)$$

onde $x = {}^I P_{BORG}$ corresponde à posição da origem do referencial do corpo no referencial inercial.

A discretização é idêntica à apresentada para o algoritmo de velocidade:

$$x_k = x_{k-1} + \Delta x_k, \quad \Delta x_k = \int_{t_{k-1}}^{t_k} v d\tau \quad (D.9)$$

Recorrendo, novamente, à aproximação trapezoidal, tem-se:

$$\Delta x_k \approx \frac{1}{2}(v_k + v_{k-1})T_k \quad (D.10)$$

onde k é índice do instante de amostragem t_k e $T_k = t_k - t_{k-1}$.

E Modelo do Veículo

Para simular o comportamento de veículo foi utilizado o modelo que aqui se apresenta. Este modelo matemático gera, através da simulação da sua dinâmica, a informação a ser registada pelos sensores, adicionada das não idealidades inerentes. A medida de desempenho do sistema é feita através da comparação dos resultados de posição, velocidade e atitude gerados pelo modelo do veículo (valores reais) e os mesmos dados gerados pelo algoritmo de navegação no *Hardware* no *loop* (valores estimados).

E.1 Descrição Física do Veículo

Foi utilizado um veículo simétrico e uniforme relativamente ao centro de massa (forma paralelepipedica). Este veículo apresenta seis propulsores (dois por face) que permitem uma gama completa de manobras de translação (três graus de liberdade) e de rotação (três graus de liberdade). Este modelo simples e fácil de comandar apresenta-se na Figura E.1:

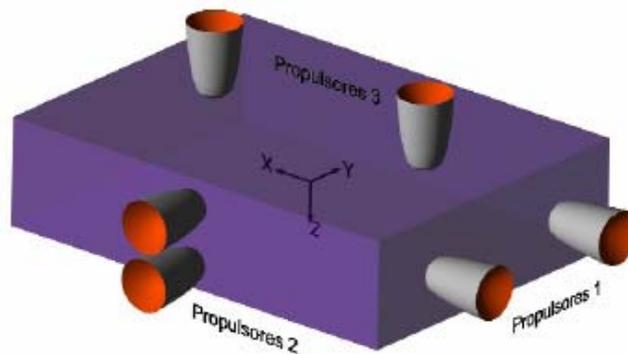


Figura E.1 – Representação gráfica do modelo do veículo (extraído de [CaVa03])

A Tabela E.1 apresenta as características físicas do veículo em coordenadas do referencial do centro de massa. A localização dos sensores corresponde à origem do referencial $\{B\}$, $({}^G P_{BORG})$, que em geral não coincide com a origem do referencial do centro de massa $\{G\}$. Uma vez que os referenciais têm a mesma orientação, tem-se ${}^G P_{BORG} = -{}^B P_{GORG}$.

Tabela E.1 – Características físicas do veículo

	Valor
Massa	10 kg
Altura (z)	0,25 m
Comprimento (y)	0,75 m
Largura (x)	1,00 m
Propulsores 1 (eixo y)	$[-0,50; \pm 0,30; 0]^T$ m
Propulsores 2 (eixo z)	$[0; \pm 0,375; \pm 0,10]^T$ m
Propulsores 3 (eixo x)	$[\pm 0,40; 0; \pm 0,125]^T$ m
Sensores (origem $\{B\}$)	$[-0,1; -0,2; -0,3]^T$ m

E.2 Equações da Dinâmica

Visto os sensores estarem localizados no referencial {B}, a dedução das equações da dinâmica é descrita no espaço desse referencial.

Da terceira lei de Newton sobre a translação do ponto de um corpo rígido em relação a um referencial inercial [CaVa03] e da equação de Euler para a rotação em torno de (${}^G P_{BORG}$):

$$\frac{d^I L_{BORG}}{dt} = {}^I N_{BORG} + m({}^I P_{BORG} - {}^I P_{BORG}) \times \frac{d({}^I V_{BORG})}{dt} \quad (E.1)$$

$${}^I F = m \frac{d^I V_G}{dt}$$

onde L_{BORG} representa o momento angular, N_{BORG} é o binário das forças exteriores, ambos em torno da origem de {B} e ${}^I F$ é a resultante das forças externas aplicadas ao centro de massa. Todas as grandezas são vistas no referencial {I}.

De (E.1), obtêm-se as expressões que descrevem a influência das forças e binários exteriores na rotação e translação do corpo rígido [CaVa03]:

$$N = M_R \dot{\Omega} + M_{RT} + \Omega \times (M_{RT} U + M_R \Omega) + U \times (M_{RT} \Omega) \quad (E.2)$$

$$F = M_T \dot{U} + M_{TR} \dot{\Omega} + \Omega \times (M_{TR} \Omega + M_T U)$$

onde:

- $N = {}^B N_{BORG}$ é o momento das forças em torno da origem de {B}, visto em {B}.
- $F = {}^B F$ é a resultante das forças externas aplicada no centro de massa, vista em {B}.
- $\Omega = {}^B ({}^I \omega_B)$ é a velocidade angular do referencial {B} em relação ao referencial {I}, vista no referencial {B}.
- $\dot{U} = \frac{d^B ({}^I V_{BORG})}{dt}$ é a aceleração resultante da variação de velocidade da origem de {B} em relação a {I}, vista no referencial {B}.
- $M_R = I_B$ é a matriz dos efeitos inerciais de rotação devido à rotação.
- $M_T = mI_3$ é a matriz dos efeitos inerciais de translação devido à translação.
- $M_{RT} = m({}^B P_{GORG} \times)$ é a matriz dos efeitos inerciais de rotação devido à translação.
- $M_{TR} = -m({}^B P_{GORG} \times) = M_{RT}^T$ é a matriz dos efeitos inerciais de translação devido à rotação.

O tensor de inércia no referencial do centro de massa, I_G , é definido por [Crai89]:

$$I_G = \iiint_V \rho \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \quad (E.3)$$

onde ρ é a densidade do veículo em cada elemento diferencial de volume dv .

Particularizando para o veículo modelado:

$$I_G = \frac{m}{12} \begin{bmatrix} h^2 + l^2 & 0 & 0 \\ 0 & w^2 + h^2 & 0 \\ 0 & 0 & l^2 + w^2 \end{bmatrix} \quad (\text{E.4})$$

onde h , l e w são a altura, comprimento e largura do veículo, respectivamente.

Uma vez que os referenciais $\{G\}$ e $\{B\}$ têm a mesma orientação, o tensor de inércia no referencial do corpo I_B é definido em função de I_G através do Teorema dos Eixos Paralelos [Crai89]:

$$I_B = I_G + m \left[{}^B P_{GORG}^T {}^B P_{GORG} I_{3 \times 3} - {}^B P_{GORG} {}^B P_{GORG}^T \right] \quad (\text{E.5})$$

Consegue-se obter as equações fundamentais para simulação da dinâmica do veículo, reescrevendo explicitamente (E.2) como equações diferenciais de U e Ω [CaVa03]:

$$\begin{aligned} \dot{U} &= Y^{-1} \left\{ F - M_{TR} M_R^{-1} \left[N - \Omega \times (M_R \Omega + M_{RT} U) - U \times (M_{TR} \Omega) \right] - \Omega \times (M_T U + M_{TR} \Omega) \right\} \\ \dot{\Omega} &= X^{-1} \left\{ N - M_{RT} M_T^{-1} \left[F - \Omega \times (M_T U + M_{TR} \Omega) \right] - \Omega \times (M_R \Omega + M_{RT} U) - U \times (M_{TR} \Omega) \right\} \end{aligned} \quad (\text{E.6})$$

onde:

$$X = M_R - M_{RT} M_T^{-1} M_{TR}$$

$$Y = M_T - M_{TR} M_R^{-1} M_{RT}$$

E.3 Influência das forças e dos binários

O veículo é afectado por diversas forças exteriores, gerando binários que dependem da localização do referencial $\{B\}$. Visto de um referencial genérico $\{A\}$, o binário originado por forças exteriores em torno da origem do referencial $\{B\}$ é dado por:

$${}^A N_{BORG} = {}^A R \sum_i {}^B P_i \times {}^B F_i \quad (\text{E.7})$$

onde i é o índice da força ${}^B F$ com ponto de aplicação ${}^B P$, vistos no referencial $\{B\}$.

É necessário converter as forças e os binários para o referencial do corpo, visto que as equações da dinâmica de (E.2) se situam no espaço do referencial $\{B\}$ e as forças externas são exercidas directamente no centro de massa (referencial $\{G\}$):

- A conversão das forças obtém-se através de uma rotação entre os referenciais $\{G\}$ e $\{B\}$. Como ambos têm a mesma orientação ${}^B R = I_3$, verifica-se:

$${}^B F = {}^G F \quad (\text{E.8})$$

- De (E.7), a translação entre os referenciais $\{G\}$ e $\{B\}$ origina binário em torno da origem de $\{B\}$. Partindo da equação genérica de conversão de coordenadas de um ponto, tem-se:

$${}^B P = {}^R R {}^G P + {}^B P_{GORG} \quad (E.9)$$

Fazendo $\{A\}$ coincidir com $\{B\}$ em (E.7) e recorrendo a (E.9) e às propriedades do produto externo [CaVa03] e das matrizes de rotação:

$$\begin{aligned} {}^B N_{BORG} &= \sum_i {}^B P_i \times {}^B F_i = \sum_n \left({}^R R {}^G P_i + {}^B P_{GORG} \right) \times {}^B F_i = \\ &= {}^B R \sum_i \left({}^G P_i \times {}^G F_i \right) + \sum_i \left({}^B P_{GORG} \times {}^B F_i \right) = {}^B R {}^G N_{GORG} + {}^B P_{GORG} \times {}^B F \end{aligned} \quad (E.10)$$

De seguida, descrevem-se as forças e binários que actuam sobre o veículo:

- Força gravítica: está relacionada com a massa do corpo:

$${}^G F_g = m {}^G g \quad (E.11)$$

onde ${}^G g$ representa a aceleração gravítica expressa no referencial $\{G\}$.

- Força e binário de atrito: são proporcionais às velocidades linear e angular do veículo:

$$\begin{cases} {}^G F_a = -K_{lin} {}^G \left({}^I V_{GORG} \right) \\ {}^G N_a = -K_{ang} {}^G \left({}^I \omega_G \right) \end{cases} \quad (E.12)$$

De mencionar que o binário de atrito N_a é distinto do binário gerado por F_a e invariante do referencial $\{G\}$ para o referencial $\{B\}$. Recorrendo ao Teorema de Coriolis e sabendo que ambos os referenciais têm a mesma orientação e rodam com a mesma velocidade angular (${}^B \omega_G = 0, {}^B R = I_3$), calculam-se as velocidades de (E.12):

$$\begin{aligned} {}^G \left({}^I V_{GORG} \right) &= {}^B \left({}^I V_{GORG} \right) = U + [\Omega \times] {}^B P_{GORG} \\ {}^G \left({}^I \omega_G \right) &= {}^G \left({}^I \omega_G \right) + {}^G \left({}^I \omega_G \right) = \Omega \end{aligned} \quad (E.13)$$

- As forças e os binários de propulsão são expressos por:

$$\begin{aligned} {}^G F_{prop} &= \sum_n {}^G F_n \\ {}^G N_{GORG prop} &= \sum_n {}^G P_n \times {}^G F_n \end{aligned} \quad (E.14)$$

onde ${}^G P_n$ representa a coordenada do propulsor n do referencial $\{G\}$.

A força e binário resultantes no centro de massa são descritos por:

$$\begin{aligned}
{}^G F_{ext} &= {}^G F_g + {}^G F_a + {}^G F_{prop} \\
{}^G N_{G\ ORG\ ext} &= {}^G N_{G\ ORG\ prop} + N_a
\end{aligned}
\tag{E.15}$$

De (E.8), (E.10), (E.15) e ${}^B R = I_3$ obtém-se, por fim, as equações que descrevem a força e binário resultantes no referencial {B}:

$$\begin{aligned}
{}^B F_{ext} &= {}^G F_g + {}^G F_a + {}^G F_{prop} \\
{}^B N_{B\ ORG\ est} &= {}^G N_{G\ ORG\ prop} + N_a + {}^B P_{G\ ORG} \times {}^B F_{ext}
\end{aligned}
\tag{E.16}$$

A força gravítica é anulada por um propulsor virtual, para uma maior simplicidade no comando do veículo. Esta compensação não impede, no entanto, a existência duma força específica nos acelerómetros, já que a componente gravítica da força específica não está relacionada com a actuação do propulsor virtual.

F Desenvolvimento em *Matlab/Simulink* do Modelo do Veículo, Sensores e Sistema de Navegação

Neste Apêndice descreve-se o simulador desenvolvido *Simulink* segundo a óptica do utilizador. Este simulador baseia-se no modelo apresentado em [CaVa03], tendo sido desenvolvido o modelo relativo aos Sistemas de Navegação. Tanto para a validação dos Sistemas de Navegação no computador de bordo (secção 5.2), como para a implementação destes no HITL (secção 5.3), os testes foram realizados tendo como base uma plataforma comum, correspondente ao modelo do veículo e sensores, que se apresenta na Figura F.1:

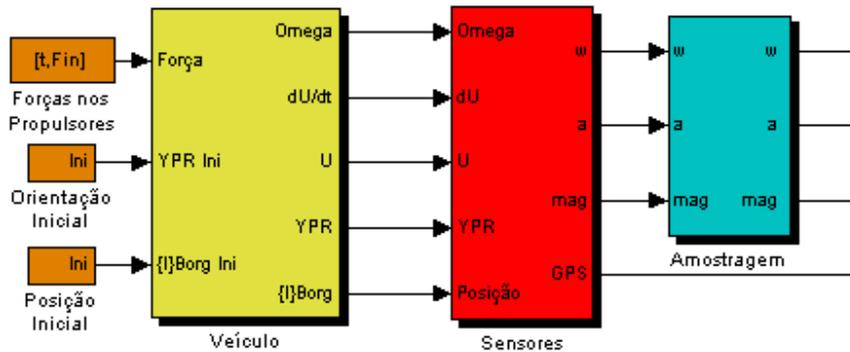


Figura F.1 – Plataforma comum com o veículo e sensores

Este modelo tem o objectivo de gerar valores realistas do comportamento do veículo no espaço tridimensional, bem como dos sensores instalados a bordo, de forma a poderem ser testados os sistemas de navegação desenvolvidos.

Antes de se iniciar qualquer simulação, é necessário proceder à inicialização de um conjunto de variáveis.

As características do modelo do veículo e parâmetros associados aos vários sistemas de navegação encontram-se no ficheiro “inicializa.m”. As forças aplicadas ao veículo pelos seis propulsores instalados, definem-se no ficheiro “propulsão.m”, através do vector de forças, Fin e de vector de tempos, t , cujos elementos correspondem aos instantes de tempo em que ocorrem as forças:

$$Fin = \begin{bmatrix} F_k^1 & F_k^2 & F_k^3 & F_k^4 & F_k^5 & F_k^6 \end{bmatrix} \quad \text{e} \quad t = \begin{bmatrix} \vdots \\ t_k \\ \vdots \end{bmatrix} \quad (\text{F.1})$$

onde F_k^i tem dimensões iguais ao vector t e representa a força aplicada pelo propulsor i deste o instante t_k até t_{k+1} .

F.1 Modelo do Veículo

Do modelo do veículo resulta o modelo em *Simulink* apresentado na Figura F.2, no qual se fazem referência às equações apresentadas no Apêndice E:

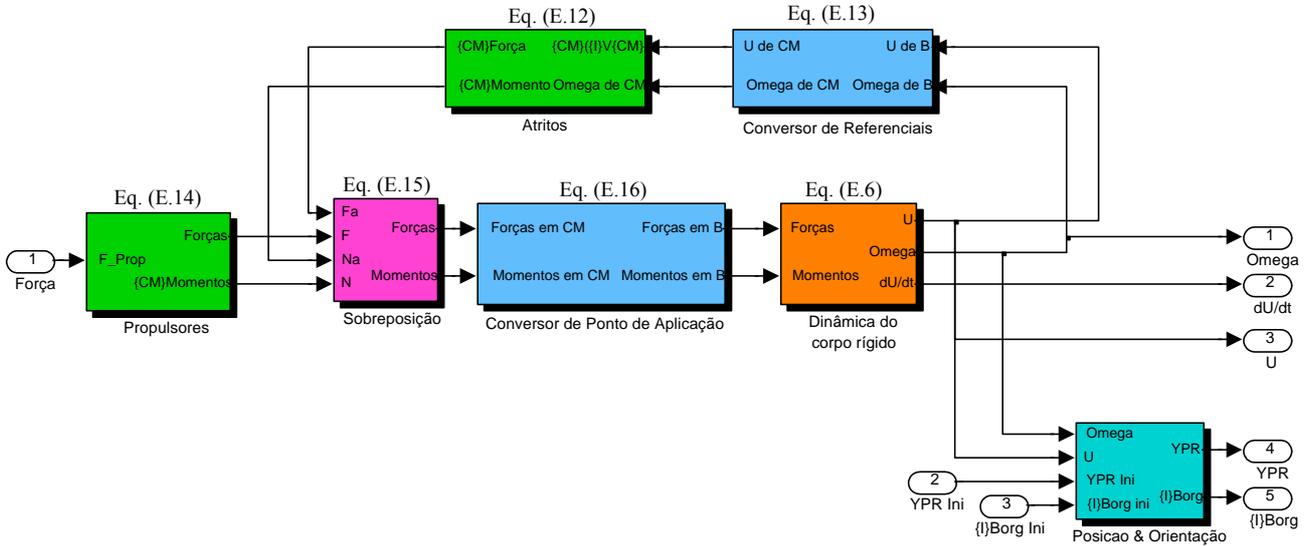


Figura F.2 – Modelo do veículo em *Simulink* com referência às equações de cada bloco

As forças de atrito introduzidas no modelo podem facilmente ser alteradas através de bloco correspondente. A dinâmica do corpo rígido e a geração da medida de posição e orientação são implementadas em subsistemas apresentados na Figura F.3:

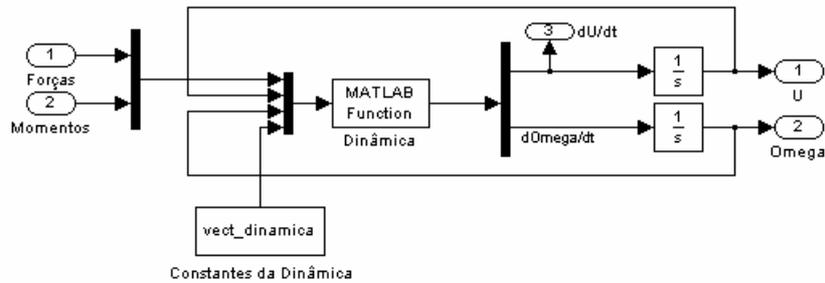


Figura F.3 – Dinâmica do corpo rígido

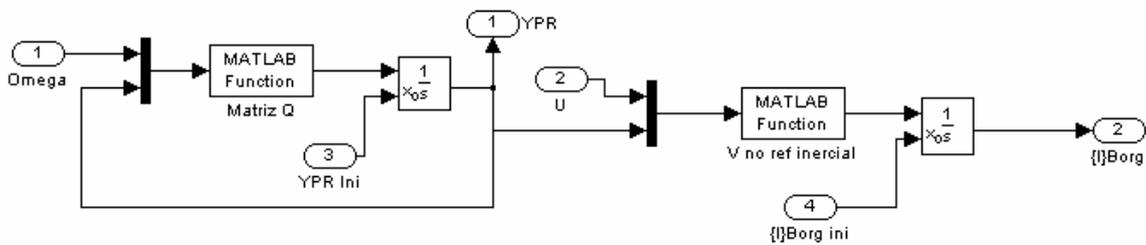


Figura F.4 – Geração das medidas de posição e de orientação

Tal como foi referido no Apêndice E, a força gravítica é anulada por um propulsor virtual, o que permite uma maior facilidade na geração de trajectórias para teste dos sistemas de navegação e não resulta de qualquer limitação introduzida no simulador ou no desenvolvimento do modelo do veículo.

F.2 Modelo dos Sensores

O objectivo é simular as leituras de uma tríade de acelerómetros, de magnetómetros, de três giroscópios e de um GPS.

A partir dos dados produzidos pelo modelo do veículo, as medidas dos sensores são geradas através da adição de não idealidades, de acordo com a Tabela F.1:

Tabela F.1 – Polarizações e variâncias dos ruídos dos sensores

	Giroscópio	Acelerómetro	Magnetómetro	GPS
Polarização	0.05°/s	10mg	-	-
Variância do Ruído (σ^2)	(0.02°/s) ²	(0.6mg) ²	(1μG) ²	(0.7m) ² /(2.5m) ²

A Figura F.5 mostra o modelo dos sensores utilizado:

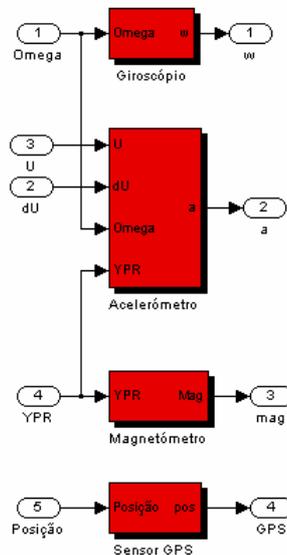


Figura F.5 – Modelo dos sensores em *Simulink*

A teoria subjacente a cada sensor é apresentada no Apêndice B, sendo a sua implementação em *Simulink* de acordo com a Figura F.6.

Foram definidos dois modelos para o GPS, dos quais apenas se apresenta o modelo utilizado para o filtro de Kalman estendido que apresenta uma perturbação mais realista devido à introdução de uma componente adicional correspondente a ruído branco filtrado por um processo Gauss-Markov. O

modelo utilizado para o filtro de Kalman complementar recorre a um modelo mais simples, assumindo a existência de ruído branco Gaussiano de média nula.

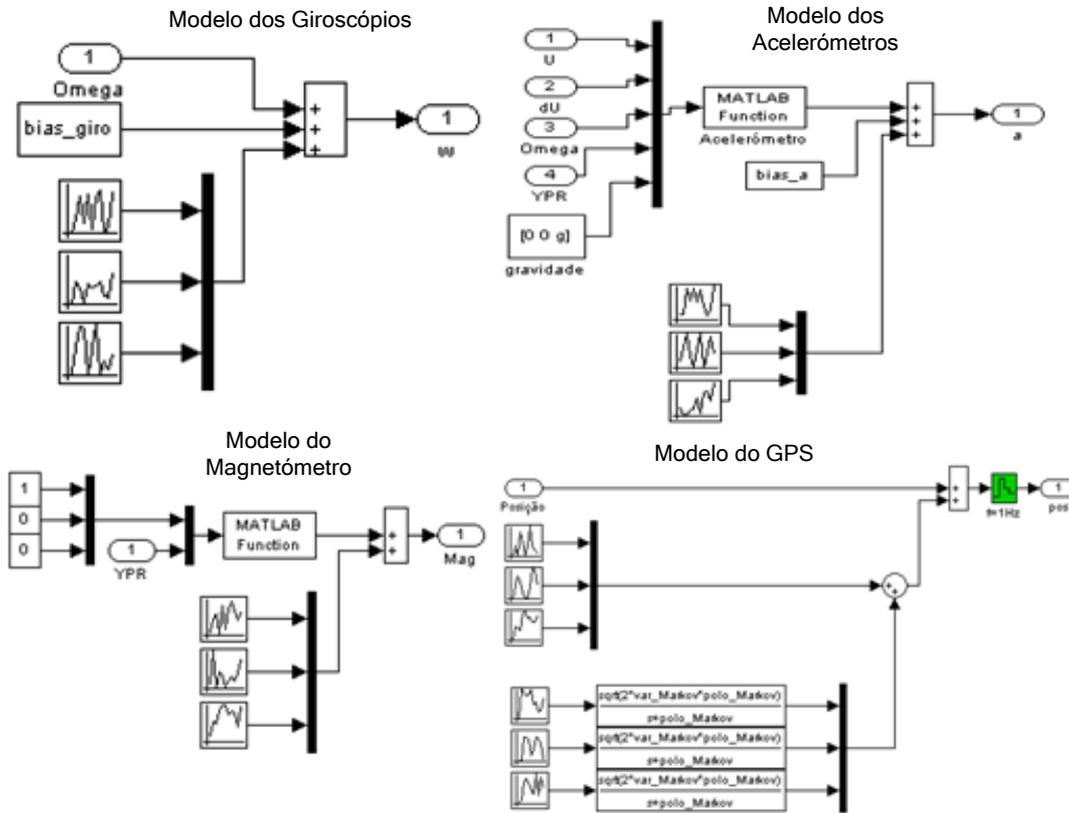


Figura F.6 - Modelo dos vários sensores em *Simulink*

F.3 Modelo Geral

Este modelo, que se mostra na Figura F.7, apresenta uma visão geral do simulador utilizado. De referir que o bloco que executa o sistema de navegação é feito à custa de uma *Matlab C-Mex Function*. Tal tem o objectivo de minimizar as diferenças entre a execução do algoritmo em ambiente *Simulink* e a execução deste no DSP, uma vez que a linguagem de implementação é a mesma (ANSI C). Embora este modelo tenha uma apresentação única, o bloco correspondente ao sistema de navegação executa uma ou outra *Mex Function* dependendo se se pretende testar o filtro de Kalman complementar ou o filtro de Kalman estendido. O modelo que implementa este último encontra-se preparado para receber uma medida auxiliar de velocidade, caso esteja disponível.

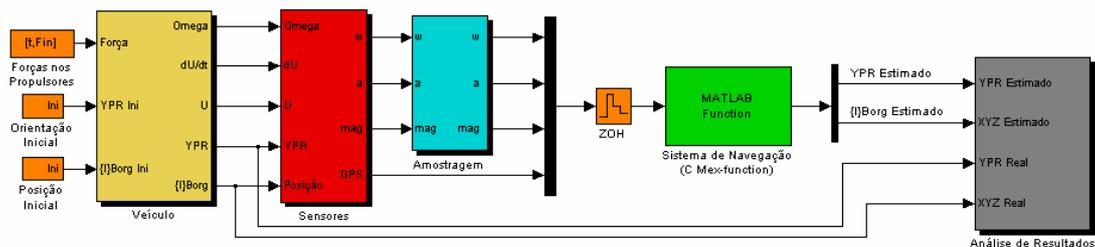


Figura F.7 – Modelo geral em *Simulink*

G Arquitectura de Aquisição de Sinal (Placas AD24B3C)

Nesta subsecção expõe-se a Arquitectura de Aquisição de Sinal e alguns dos componentes que a constituem. Na Figura G.1 ilustra-se a placa AD24B3C, elemento principal da AAS.

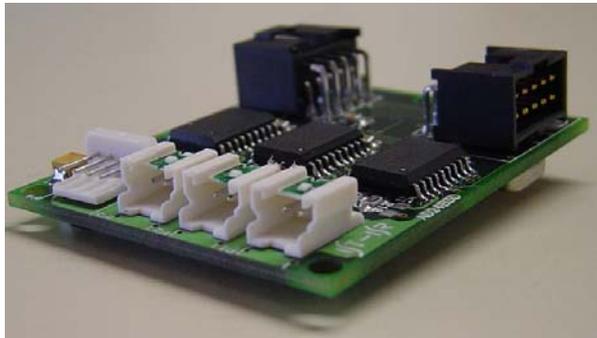


Figura G.1 – Placa AD24B3C

G.1 Conversor ADS1210

- **Características Gerais**

Estes circuitos são desenhados para aplicações em que sejam necessárias medições de elevada resolução. Tratam-se de conversores do tipo Delta-Sigma de alta resolução (24 *bit*), funcionando com uma única fonte de alimentação de +5V. Aceitam entradas diferenciais, que são ideais para ligações directas a transdutores ou sinais de baixa tensão.

Consegue-se uma resolução efectiva de 23 *bit* usando um amplificador de baixo ruído em taxas de conversão até aos 10Hz. Resoluções efectivas de 20 *bit* conseguem-se até uma frequência de amostragem de 1kHz, através do uso do modo Turbo (*Turbo Mode*).

- **Características Específicas (de [ADDS00])**

- Conversor A/D Delta-Sigma
- Resolução efectiva de 23 *bit* a 10 Hz e de 20 *bit* a 1000 Hz
- Entradas Diferenciais
- Amplificador de Ganho Programável
- SPI flexível, compatível com modo de interface SSI com duas linhas
- Frequência de corte programável até 15.6 kHz
- Referência interna/externa
- *Self-Calibration* no circuito

Cada conversor é constituído por um amplificador de ganho programável (PGA), um modulador Delta-Sigma de segunda ordem, um filtro digital programável, um microcontrolador, um circuito gerador de relógio e uma tensão de referência. Para mais detalhe consultar o manual elaborado [BCD04].

G.2 Escorregamento das Frequências de Relógio dos Cristais

Suponhamos que estamos a operar a uma frequência de sinal:

$$f_{\text{sinal}} = 100\text{Hz} \Rightarrow T = 0,01\text{s}$$

Temos que a frequência de oscilação do cristal é de:

$$f_{\text{osc}} = 10\text{MHz} \Rightarrow T = 10^{-7}\text{s}$$

Para cada ciclo do sinal ($T = 0,01\text{s}$) existem 100 000 períodos do sinal de oscilação do cristal. É sabido que a tolerância dos cristais utilizados é de $\pm 30\text{ppm}$ a 25°C , o que faz com que o número de ciclos deste sinal em $0,01\text{s}$ varie de 99 997 a 100 003. Assim, no pior dos casos, pode-se obter uma diferença de 6 ciclos de relógio num período de $0,01\text{s}$. Sendo o período dessas oscilações de 10^{-7}s , tem-se:

$$6 \times 10^{-7} = 0,6\mu\text{s}$$

Conclui-se então que, no pior dos casos, para cada $0,01\text{s}$ (período do sinal), existe uma desfasagem de $0,6\mu\text{s}$.

G.3 Ligações entre as Placas AD24B3C

- **Sinais de Interface**

No grupo de sinais de interface, existem:

- *Serial Data Output* – *SDIO*
- *Clock Input/Output* – *SCLK*
- *Serial Data Output* – *SDOUT*
- *Data Ready* – *DRDY*

O sinal *SDIO* é utilizado para a escrita de dados provenientes do microcontrolador XAS3 para os ADC's. Tais dados incluem as configurações de cada ADC (ganho, modo turbo, taxa de dados), bem como a ordem da leitura dos resultados de conversão. A escrita de dados no ADC é feita através do Registo de Instrução (INSR) do microcontrolador de cada conversor. Em cada placa AD24B3C do sistema o sinal passa numa primeira fase por um *buffer*, seguindo depois para cada um dos ADC's.

O sinal *SCLK* consiste num sinal de relógio de entrada que determina a taxa da transferência de dados. Estando os conversores a operar em *Slave Mode*, a frequência do sinal *SCLK* não poderá exceder a frequência de relógio (X_{IN}) a dividir por cinco.

O sinal de interface *SDOUT* possibilita a transferência dos resultados da conversão para a placa MC-XAS3. Estes resultados provêm do Registo de Dados de Saída (DOR), onde foram previamente escritos. Esta situação é descrita com maior detalhe em G.7.2. O sinal que chega à placa microcontroladora é o resultado de uma combinação lógica dos sinais *SDOUT* provenientes de cada conversor de uma mesma placa AD24B3C (*SDOUT1*, *SDOUT2*, *SDOUT3*) com o *SDOUT* resultante da Placa AD24B3C anterior (*SDOUT_ANT*), como se esquematiza na Figura G.7.

Finalmente, o sinal *nDRDY* indica a existência de novos resultados de conversão válidos, indicando ao microcontrolador que este já pode aceder aos valores existentes no DOR. A transição deste sinal para o nível “0”, anuncia a existência de novos dados disponíveis. Ao contrário do que acontece com o sinal *SDOUT*, apenas o *nDRDY* do ADC *Master* de cada Placa AD24B3C é “visto” pela placa AD24B3C seguinte ou pelo microcontrolador. Consequentemente, a placa microcontroladora recebe uma combinação lógica dos ADC *Master’s* existentes (ver Figura 3.3).

- **Barramento de Endereços**

Os sinais de endereço existentes são:

- Linha de endereço 0 – ADDR0
- Linha de endereço 1 – ADDR1
- Linha de endereço 2 – ADDR2
- Linha de endereço 3 – ADDR3

As duas linhas de endereço menos significativas (ADDR1 e ADDR0) são responsáveis pela selecção de cada ADC dentro da respectiva Placa AD24B3C. As duas linhas de endereço restantes determinam qual a Placa AD24B3C seleccionada. Assim, são necessários quatro *bits* para seleccionar um dos possíveis doze (recorde-se que poderão existir quatro Placas AD24B3C) conversores existentes.

- **Buffer**

A existência de um *buffer* deve-se aos requisitos em termos de corrente para cada um dos níveis lógicos, de modo a que estes não sejam deturpados. Para que este requisito seja satisfeito é necessário ter em atenção o *fan-out*, i.e., o número máximo de entradas que uma porta pode accionar.

Uma vez que os sinais provenientes do microcontrolador XAS3 são aplicados a várias entradas dos ADC’s, a introdução de um *buffer* foi necessária de modo a satisfazer os requisitos acima referidos, garantindo que não existe carga excessiva para a saída do microcontrolador.

Os sinais do microcontrolador que passam pelo *buffer* são o SCLK e o *SDIO*, estando as ligações representadas na Figura G.2:

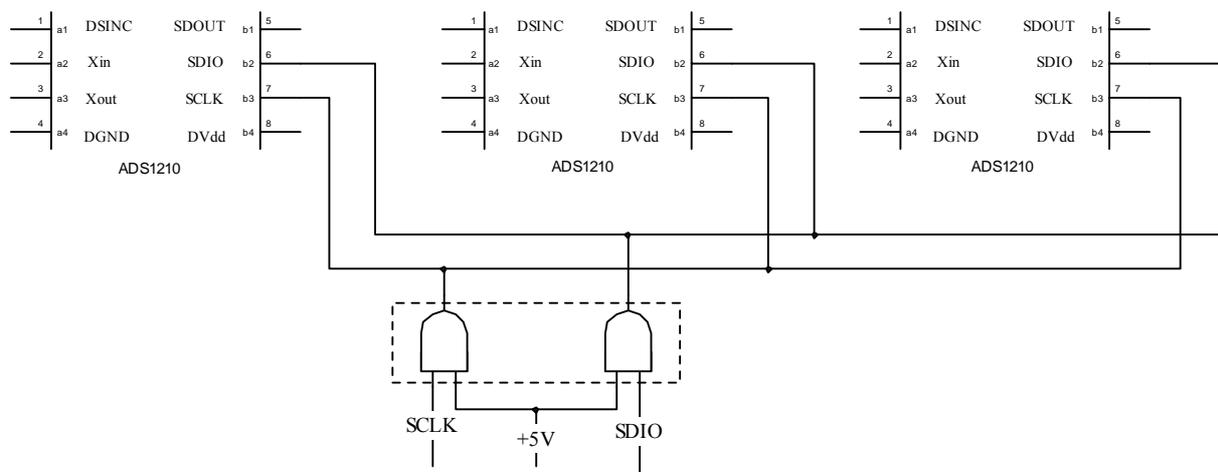


Figura G.2 – Diagrama de ligações do *buffer* em cada Placa AD24B3C

- **Ligações para o Sincronismo**

Com o circuito da Figura G.3 garante-se que o sinal assíncrono *nDSYNC* assume o valor de “1” lógico no flanco descendente do sinal de relógio (CLK), i.e., todos os ADC’s o recebem no mesmo instante. Com isto é assegurado o sincronismo exacto entre os conversores de uma placa AD24B3C.

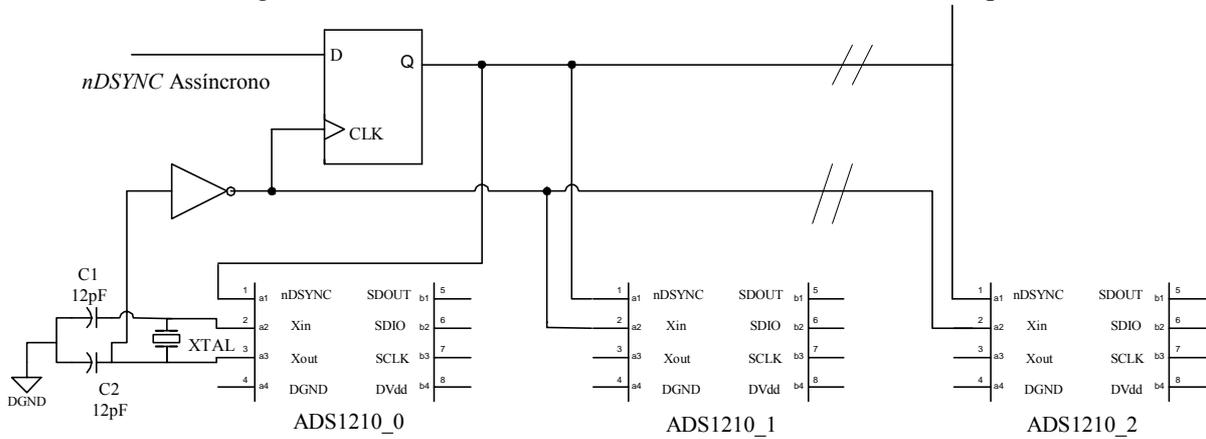


Figura G.3 – Ligações para o Sincronismo em cada placa AD24B3C

G.4 Esquemas Eléctricos da placa AD24B3C

- **Filtro RC Anti-Aliasing**

Pelo teorema de Nyquist, é sabido que um sinal deve ser amostrado a uma frequência igual ou superior ao dobro da frequência máxima do sinal, para que este mesmo sinal possa depois ser recuperado integralmente.

Dadas as características do ruído adicionado pelos componentes electrónicos, não é possível garantir que o sinal de entrada não apresente frequências acima de metade da frequência de amostragem, o que torna necessário a existência de um filtro passa-baixo com frequência de corte igual ou inferior à frequência de Nyquist.

Assim, projectou-se o filtro, cujo esquemático se apresenta na Figura G.4, que é aplicado às entradas de todos os conversores ADC:

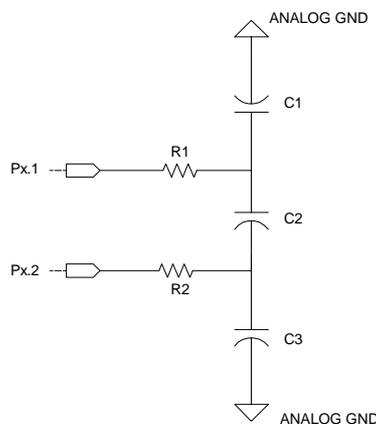


Figura G.4 – Esquemático do Filtro RC

- **Tensão de Offset**

A Figura G.5 mostra o circuito que possibilita a conversão de sinais numa gama que varia entre $\pm 10V$. A saída V_{BIAS} depende da referência da tensão de entrada (REF_{IN}) e é aproximadamente 1.33 vezes superior.

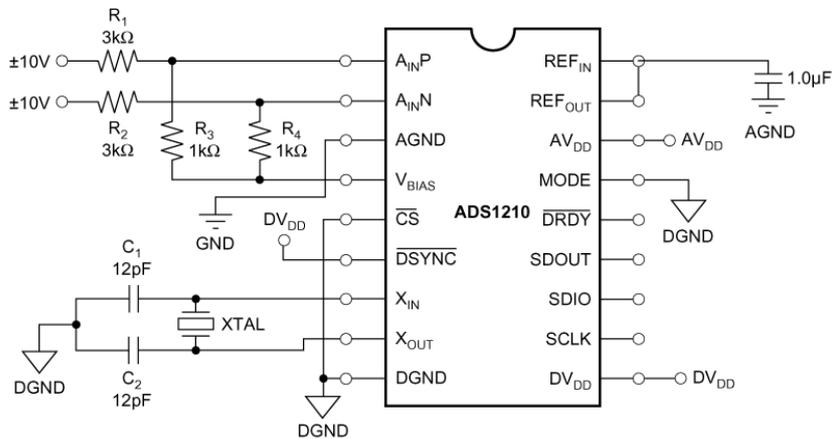


Figura G.5 – Configuração de entrada usando V_{BIAS} (extraído de [ADDS00])

G.5 Circuitos Lógicos Implementados pela PAL

- **Seleção dos ADC's**

A PAL é responsável pela selecção dos ADC's dentro de cada Placa AD24B3C. A Figura G.6 exhibe o circuito lógico respeitante a esta selecção:

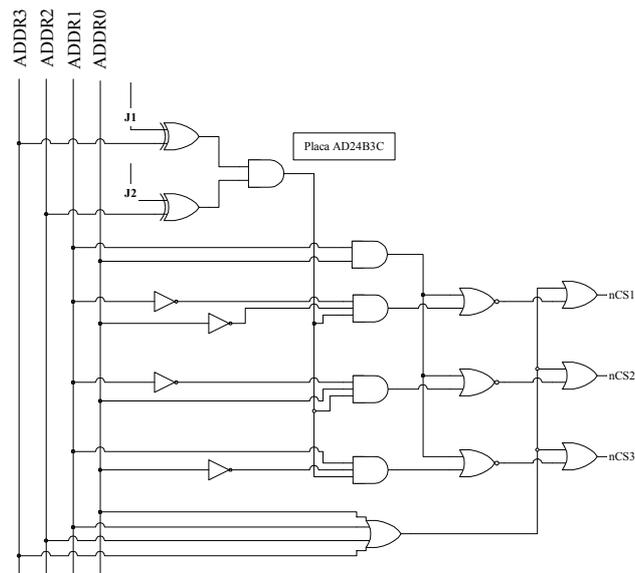


Figura G.6 – Esquema lógico que permite seleccionar cada ADC

Como complemento, apresenta-se a Tabela G.1, onde se pode verificar qual o endereço para seleccionar um determinado ADC na AAS:

Tabela G.1 – Endereçamento para a selecção de cada conversor na arquitectura completa

	ADDR3	ADDR2	ADDR1	ADDR0		
0	0	0	0	0	ADC0	Placa ADC24B3C 0
1	0	0	0	1	ADC1	
2	0	0	1	0	ADC2	
3	0	0	1	1	$\overline{CS}=1$ para todos	
4	0	1	0	0	ADC0	Placa ADC24B3C 1
5	0	1	0	1	ADC1	
6	0	1	1	0	ADC2	
7	0	1	1	1	\overline{DSYNC}	
8	1	0	0	0	ADC0	Placa ADC24B3C 2
9	1	0	0	1	ADC1	
10	1	0	1	0	ADC2	
11	1	0	1	1	<i>Sem função</i>	
12	1	1	0	0	ADC0	Placa ADC24B3C 3
13	1	1	0	1	ADC1	
14	1	1	1	0	ADC2	
15	1	1	1	1	$\overline{CS}=0$ para todos	

- **Multiplexagem dos Sinais *nDRDY***

O sinal *nDRDY* é um sinal gerado pelo ADS1210 sempre que acaba uma conversão. Este sinal é utilizado pelo microcontrolador para saber exactamente o instante em que tem dados disponíveis para efectuar o pedido de envio dos valores.

Como na arquitectura final, vão existir, no máximo, quatro placas AD24B3C e, portanto 12 conversores ADS1210, é necessário garantir que a leitura dos dados é feita só depois de todos os ADC's terem concluído a conversão. Esta situação é garantida com um OR lógico entre os sinais *nDRDY* dos conversores sucessivamente (ver Figura 3.3). O microcontrolador só é "autorizado" a fazer a leitura quando o sinal à sua entrada tem o valor lógico de "0".

- **Multiplexagem dos Sinais *SDOUT***

O pino/sinal *SDOUT* contém o resultado da conversão ADC. Esse sinal chega ao microcontrolador através do esquema da Figura G.7. Como na arquitectura completa haverá quatro placas AD24B3C, e portanto, 12 conversores, é necessário que o microcontrolador tenha controlo sobre os dados a receber e que saiba de qual a sua origem. Para tal é necessário que o microcontrolador seleccione primeiramente o ADC e de seguida faça a leitura, realizando isto consecutivamente para todos os ADC's. Isto é feito usando o sinal *nCS* que, quando colocado a "1", torna o pino *SDOUT* inactivo.

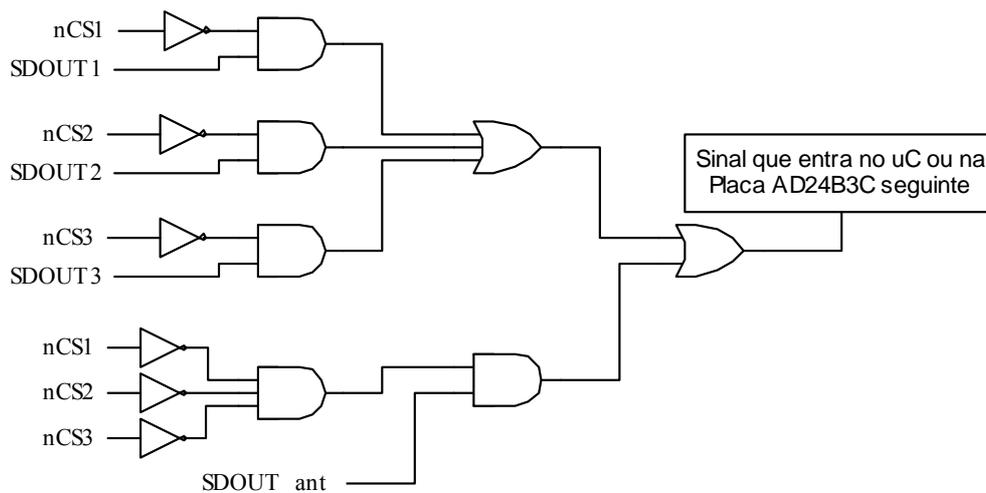


Figura G.7 – Esquema lógico que permite a multiplexagem do *SDOUT*

G.6 Circuito Programável (*Xilinx XCR22V10*)

- **Características Gerais**

O circuito para gerar a lógica da placa MC-XAS3 e da placa AD24B3C é o XCRZ22V10 da *Xilinx*. Este circuito tem por função gerar alguns sinais de controlo para os dispositivos ligados, quer ao microcontrolador, quer aos ADC's, desenvolver as máquinas de estados necessárias e permitir alguma flexibilidade ao sistema.

- **Características Específicas**

- 5V de tensão de alimentação
- Tecnologia CMOS
- Baixo consumo de potência e elevada rapidez
- Permite 1000 escritas/apagamentos
- Reprogramável usando *hardware* adequado
- Capacidade de provocar um *reset* assíncrono
- Sinal de relógio comum a todos os flip-flops

A informação foi extraída de [DSXi00].

G.7 Descrição do *Driver* entre a placa MC-XAS3 e as placas AD24B3C

```
<Variáveis fundamentais>
static AdConfigStruct AdConf[12];
AdDataStruct AdData;

<Inicio do corpo do programa>
Void main()

<Seleccionar os Conversores>
AdData.Selection = CARD_0_ALL_ADC + CARD_3_ALL_ADC;

<Chamar a função que constrói a estrutura com as opções do utilizador (CMR)>
Build_Config_Struct();

<Abertura do Driver das Placa AD24B3C>
Adc24DrvOpen(AdConf, &AdData);

<Devolve o Error Status e o estado da maquina de estados>
Adc24Status(StatusVec); //

<Permite que a maquina de estados evolua para as calibrações>
Adc24Calibration(AdConf);

Do{
    if (AdData.EventFlag)
    {
        <faz-se evoluir a maquina de estados>
        Adc24StateMachineEvHand();
    }

    if (AdData.ReadFlag)
    {
        <Leitura dos conversores>
        Adc24RD();
    }
}While (1) <ciclo infinito>
```

G.7.1 Descrição da Função de Abertura do *Driver*

O utilizador começa a interface com o *driver* da AAS, através da função *Adc24DrvOpen()* fornecendo ao mesmo, uma estrutura com as configurações que deseja para os ADCs e um ponteiro para a localização de memória onde pretende que os dados das leituras sejam colocados. Os parâmetros são utilizados para construir as mensagens de configuração a enviar para a AAS. É nesta função, que são inicializadas todas as estruturas de dados, configurados os portos do microcontrolador XA-S3 e as interrupções e se faz RESET aos ADCs. Depois dos ADCs estarem preparados para começar a aquisição de amostras, são sincronizados permitindo, em seguida, que a sua linha de *nDRDY* provoque uma interrupção no XA-S3. Quando se retorna desta função, consegue-se preservar o sincronismo dos ADCs, visto que, sempre que ocorre uma interrupção, esta é tratada enviando a todos os conversores o pulso de sincronismo. Embora não se esteja a receber as amostras que os ADCs adquirem, o *driver* continua a controlar e supervisionar o estado da AAS. Na Figura G.8 apresenta-se um fluxograma que ilustra as várias etapas da execução desta função.

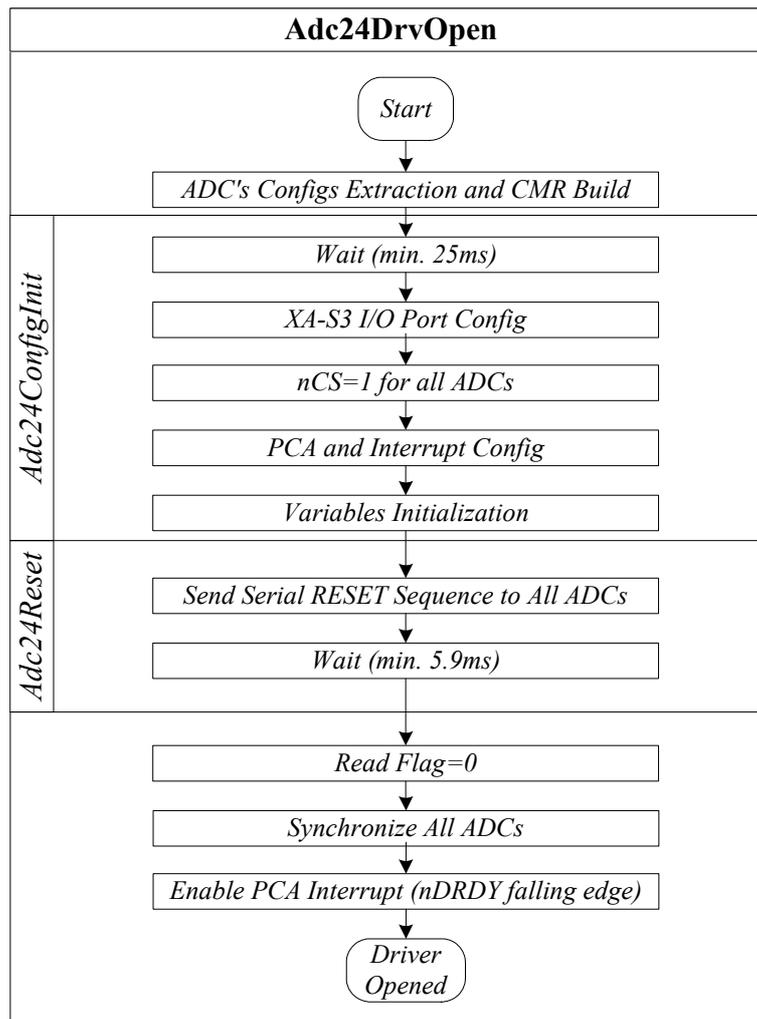


Figura G.8 – Fluxograma da função *Driver Open*

G.7.2 Descrição da Função de Leitura dos Conversores

Nesta secção, descreve-se o funcionamento e filosofia seguida no desenvolvimento da função de leitura das amostras (*Adc24RD()*) da AAS. O bom desempenho desta função, é crítico, já que, numa aplicação onde se faça uso desta arquitectura para aquisição de sinal de sensores (por exemplo acelerómetros, giroscópios ou magnetómetros), é de extrema importância que essa informação seja adquirida de forma o mais rápida e fiável possíveis. Por motivos que estão relacionados com a optimização da performance, esta rotina foi implementada na linguagem *Assembly* do microcontrolador XA-S3, para se conseguir atingir níveis de desempenho que uma linguagem de alto nível, como a linguagem C, poderia possibilitar. Na Figura G.9 apresenta-se o fluxograma que demonstra a sequência de acontecimentos quando se executa esta função.

Como se pretende que a interface com a AAS, obedeça a requisitos que lhe permitam ser inserida num sistema de tempo real, e porque a leitura dos 12 conversores pode ter a duração de aproximadamente 380µs, optou-se por tornar um bloco de leitura dos conversores seleccionados como uma secção de execução não atómica (pode ser interrompida), levando a que entre a leitura de cada conversor se permita que outras interrupções sejam atendidas. De salientar, que a duração da

interrupção tenha que ser forçosamente de curta duração (inferior a poucas centenas de μs), caso contrário, pode provocar o mau funcionamento da interface com a AAS.

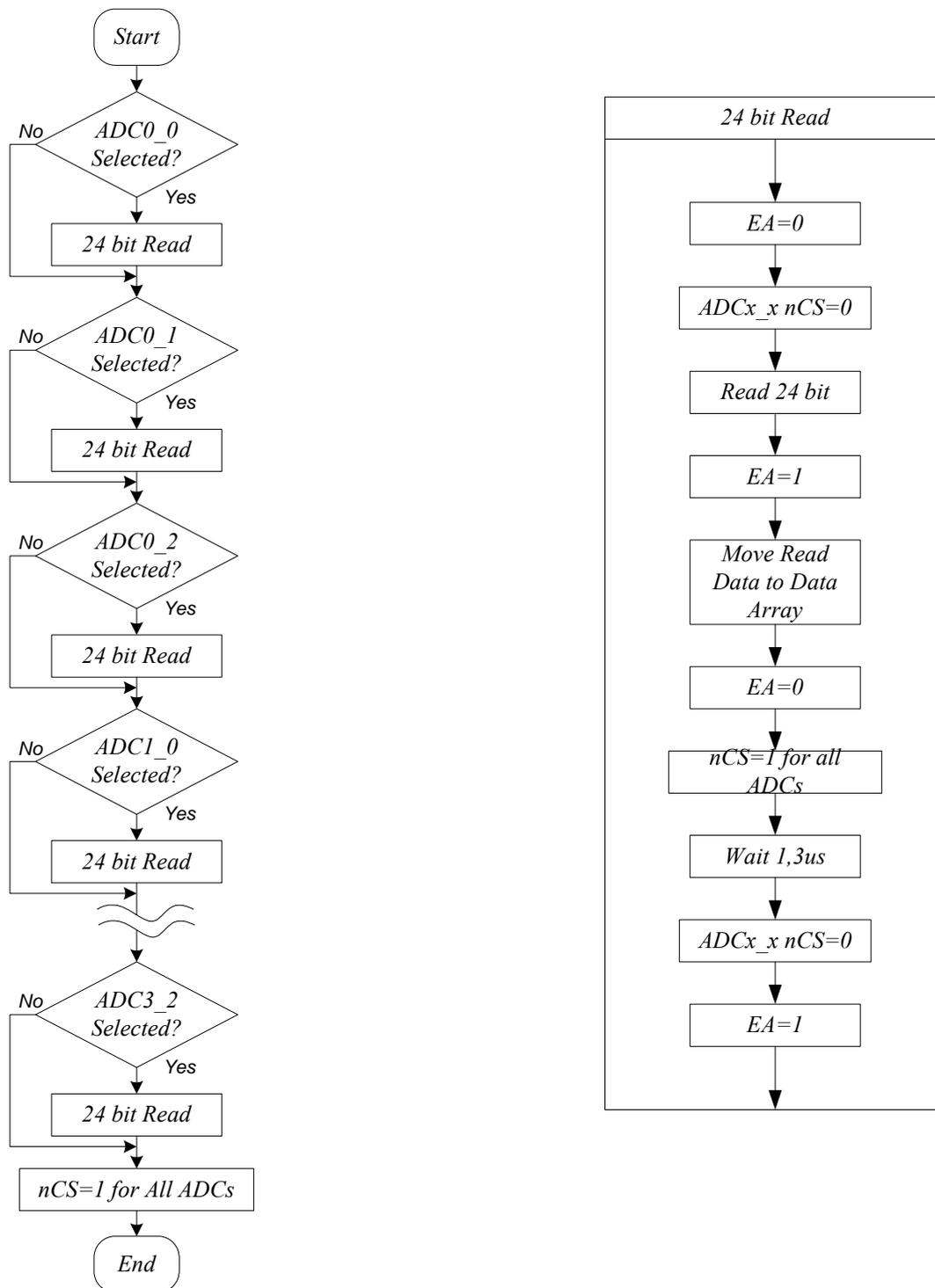


Figura G.9 – Fluxograma da leitura dos conversores

G.8 Caracterização do Desempenho da Placa ADC24B3C

Existem técnicas específicas para medir e quantificar a performance de conversores AD. Dois testes básicos incluem o uso de histogramas e a transformada *Fast Fourier* (FFT). Cada tipo de teste fornece uma perspectiva única da performance do circuito.

O tipo de teste a usar é baseado no parâmetro a medir. Por exemplo, os histogramas são utilizados para medir a exactidão em DC ou características estáticas como o *offset*, por outro lado as FFT's são uma ferramenta para medir a performance dinâmica, como por exemplo a linearidade.

Neste caso, os sinais são na sua maioria de baixa frequência e filtrados (filtro passa-baixo), por isso optou-se por apenas usar os histogramas para medir e quantificar a performance dos conversores.

A saída de um ADC varia para uma entrada constante devido ao ruído. O ruído é definido por uma Função Densidade de Probabilidade (PDF), que representa a probabilidade de eventos discretos. A forma da PDF descreve a exactidão da saída do ADC e a sua característica de ruído.

Os testes realizados às três placas utilizadas foram feitos nas mesmas condições, fios de sinal blindados, afastados de possíveis fonte de ruído e curto-circuitados entre si.

Para caracterizar cada ADC utilizaram-se medidas como a Resolução Efectiva (*Effective Resolution* (ER)), Bits livres de ruído (*Noise Free Bits or Peak-to-Peak Resolution* (NFb)) e Relação Sinal Ruído (SNR). As equações, para calcular estes valores, apresentam-se de seguida:

$$rms = std\left(data_ADC \times \frac{10}{2^{24}}\right) \quad (G.1)$$

onde std representa o cálculo do desvio padrão (*Standard Deviation*).

$$ER = \frac{\left(20 \cdot \log\left(\frac{10}{rms}\right) - 1.76\right)}{6.02} \quad (G.2)$$

$$NFb = ER - 2.723 \quad (G.3)$$

$$SNR = 20 \cdot \log\left(\frac{6.6 \times rms}{10}\right) \quad (G.4)$$

Neste apêndice apenas se apresenta os resultados do teste da placa 0. Para informação sobre a performance das restantes placas consultar [BCD04].

Analisando os histogramas da Figura G.10, Figura G.11, Figura G.12 (depois de subtraída a média das leituras) e a Tabela G.2, conclui-se que se desenvolveu uma placa de aquisição de sinal analógico de elevada performance. Para a navegação, é fundamental poder contar com conversores de elevada resolução, pois todo o processamento de sinal e cálculo posterior é baseado nos valores fornecidos por estes. Para além disso é fundamental que a limitação de exactidão não esteja do lado dos conversores, mas sim dos sensores utilizados, de modo a tirarmos todo o partido das medidas fornecidas por estes.

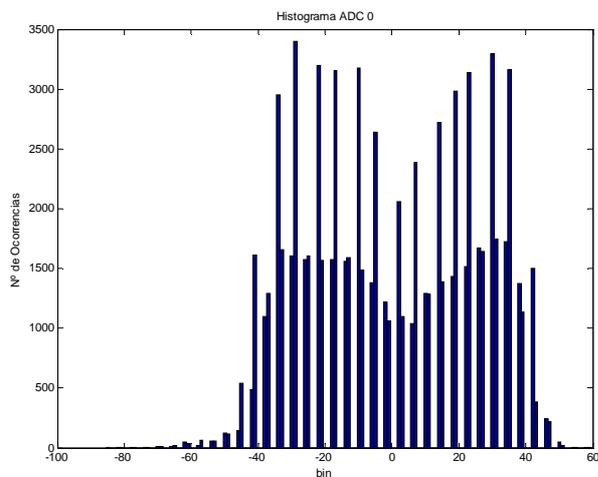


Figura G.10 – Histograma do ADC 0 Placa AD24B3C 0

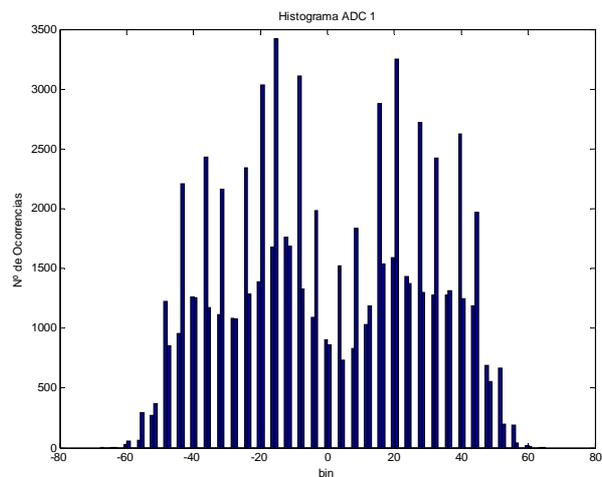


Figura G.11 – Histograma do ADC 1 Placa AD24B3C 0

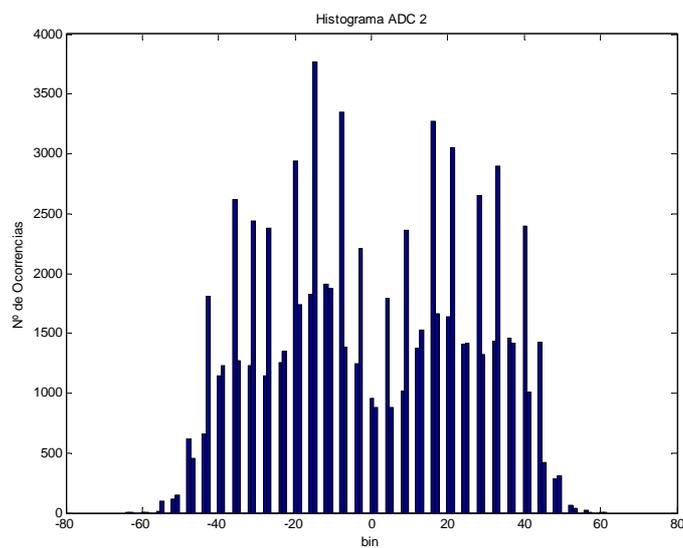


Figura G.12 – Histograma do ADC 2 Placa AD24B3C 0

Tabela G.2 – Performance da Placa AD24B3C 0

	ADC0	ADC1	ADC2
Effective Resolution <i>bits</i>	19.0488	18.872	18.997619
SNR (dB)	-100.043	-98.9788	-99.734787
Noise Free <i>bits</i>	16.326	16.1493	16.274882

G.9 Registos do conversor

Aqui apresenta-se a descrição de cada um dos registos do conversor ADS1210. Para mais detalhe consultar [BCD04].

- **Registo de Comando (CMR)**

O CMR controla todas as funcionalidades do ADS1210. Uma nova configuração terá efeito na transição negativa do *SCLK* para o ultimo *bit* de cada byte dos dados que estão a ser escritos para o CMR.

- **Registo de Dados de Saída (DOR)**

Trata-se de um registo de 24 *bit* que contém o resultado mais recente da conversão.

- **Registo de Instrução (INSR)**

Trata-se de um registo de 8 *bit* que comanda a interface série, quer seja para ler, quer seja para escrever “n” *bytes* começando na localização especificada do registo.

- **Registo de Calibração de *Offset* (OCR)**

Trata-se de um registo de 24 *bit* que contém o *offset* do factor de correcção que é aplicado ao resultado da conversão antes que este seja colocado no DOR

- **Registo de Calibração de Fim de Escala (FCR)**

Este registo é de 24 *bit* que contém o factor de correcção de fim de escala que é aplicado ao resultado da conversão antes de este ser colocado no DOR

G.10 Programa da PAL da placa ADS24B3C

```
MODULE pal_tfc
TITLE 'programa da pal para tfc'
```

```
DECLARATIONS
```

```
``INPUTS
```

```
IN_DRDY0    PIN    25;
IN_DRDY     PIN    5;
IN_SDOOUT   PIN    13;
ADDR0       PIN    4;
ADDR1       PIN    9;
ADDR2       PIN    7;
ADDR3       PIN    6;
SDOUTAD0    PIN    18;
```

```

SDOUTAD1  PIN    17;
SDOUTAD2  PIN    16;
ADDR3AUX  PIN    10;
ADDR2AUX  PIN     3;
CLK        PIN     2;
XIN        PIN    11;

```

“OUTPUTS

```

DRDY      PIN    26;
SDOUT     PIN    27;
CSAD0     PIN    23;
CSAD1     PIN    21;
CSAD2     PIN    20;
DSYNC     PIN    24;
XIN_OUT   PIN    19;

```

EQUATIONS

```

CSAD0=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&!ADDR1&!ADDR0)!$
$(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

```

```

CSAD1=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&!ADDR1&ADDR0)!$
(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

```

```

CSAD2=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&ADDR1&!ADDR0)!$
(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

```

```

SDOUT=(!CSAD0&SDOUTAD0#!CSAD1&SDOUTAD1#!CSAD2&SDOUTAD2)#(
CSAD0&CSAD1&CSAD2&IN_SDOUT);
DRDY=(IN_DRDY#IN_DRDY0);

```

```

DSYNC.clk=CLK;

```

```

DSYNC:=(!ADDR3&ADDR2&ADDR1&ADDR0);

```

```

XIN_OUT=!XIN;

```

```

END

```

H Placa DSPIf

Neste capítulo apresenta-se a Placa DSPIf. Na Figura H.1 apresenta-se o diagrama funcional da placa DSPIf, com a memória de dois portos que, a partir deste momento vai ser designada por DP-RAM.

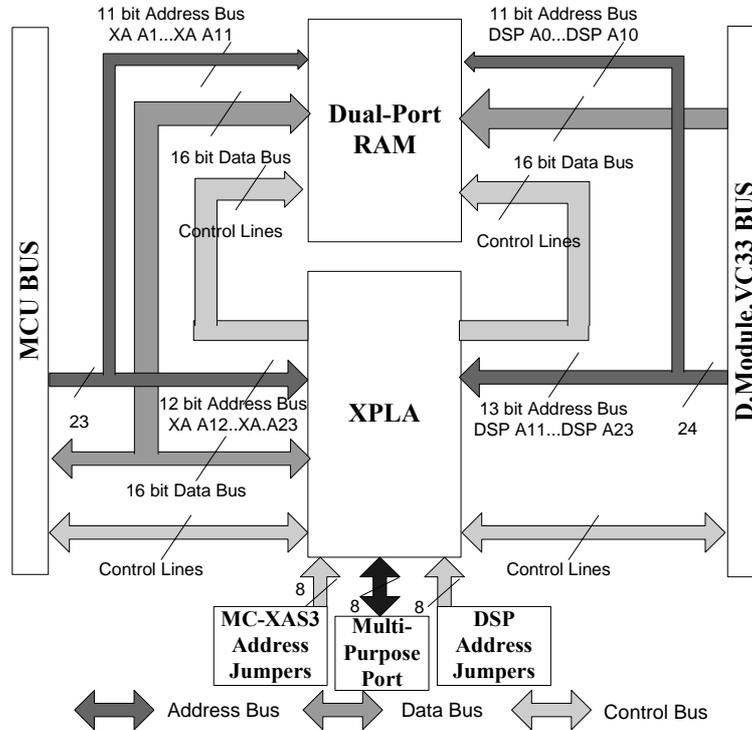


Figura H.1 – Diagrama funcional da Placa DSPIf

Começemos com a descrição da configuração dos pinos do barramento externo do *D.Module.VC33* (ver Tabela H.1) e da Placa MC-XAS3 (ver Tabela H.2) utilizados na interface com a Placa DSPIf:

Tabela H.1 – Configuração dos pinos do barramento externo do *D.Module.VC33* utilizados na interface com a placa DSPIf (extraído de[VCUG01])

Sinal	Pino	Tipo	Descrição
D0..D15	V15..V30	I/O	<i>data bus bit 0..15</i>
A0..A5	U9..U14	O	<i>address bus bit 0..5</i>
A6..A18	V2..V14	O	<i>address bus bit 6..18</i>
A19..A23	A14..A18	O	<i>address bus bit 19..23</i>
nMEMSEL	V31	O	<i>MEMSEL memory select, active low</i>
nRD	U2	O	<i>read select (read from ext. bus), active low</i>
nWR	U5	O	<i>write select (write to ext. bus), active low</i>
nINT0	U3	I	<i>External Interrupt Request inputs, falling edge, on-board pull-up resistor</i>
nRESOUT	U7	O	<i>reset input, active low</i>

Nota da Tabela H.2:

D0..D31 are active only if IOSEL or MEMSEL is accessed, they are disconnected during on-board accesses.

Tabela H.2 – Configuração dos pinos do barramento externo da placa MC-XAS3 utilizados na interface com a placa DSPIf (extraído de [VCUG01])

Sinal	Tipo	Descrição
D0..D15	I/O	<i>data bus bit 0..15</i>
A1..A23	O	<i>address bus bit 0..23</i>
nCS16	O	<i>MEMSEL memory select, active low</i>
XA RD	O	<i>read select (read from ext. bus), active low</i>
XA WRL	O	<i>write select (write to ext. bus) low byte, active low</i>
XA WRH	O	<i>write select (write to ext. bus) high byte, active low</i>
XA INT1..XA INT7	I	<i>External Interrupt Request inputs</i>

H.1 Esquema Eléctrico da Placa DSPIf

O esquema do circuito eléctrico da Placa DSPIf (desenvolvida no DSOR) encontra-se na Figura H.2:

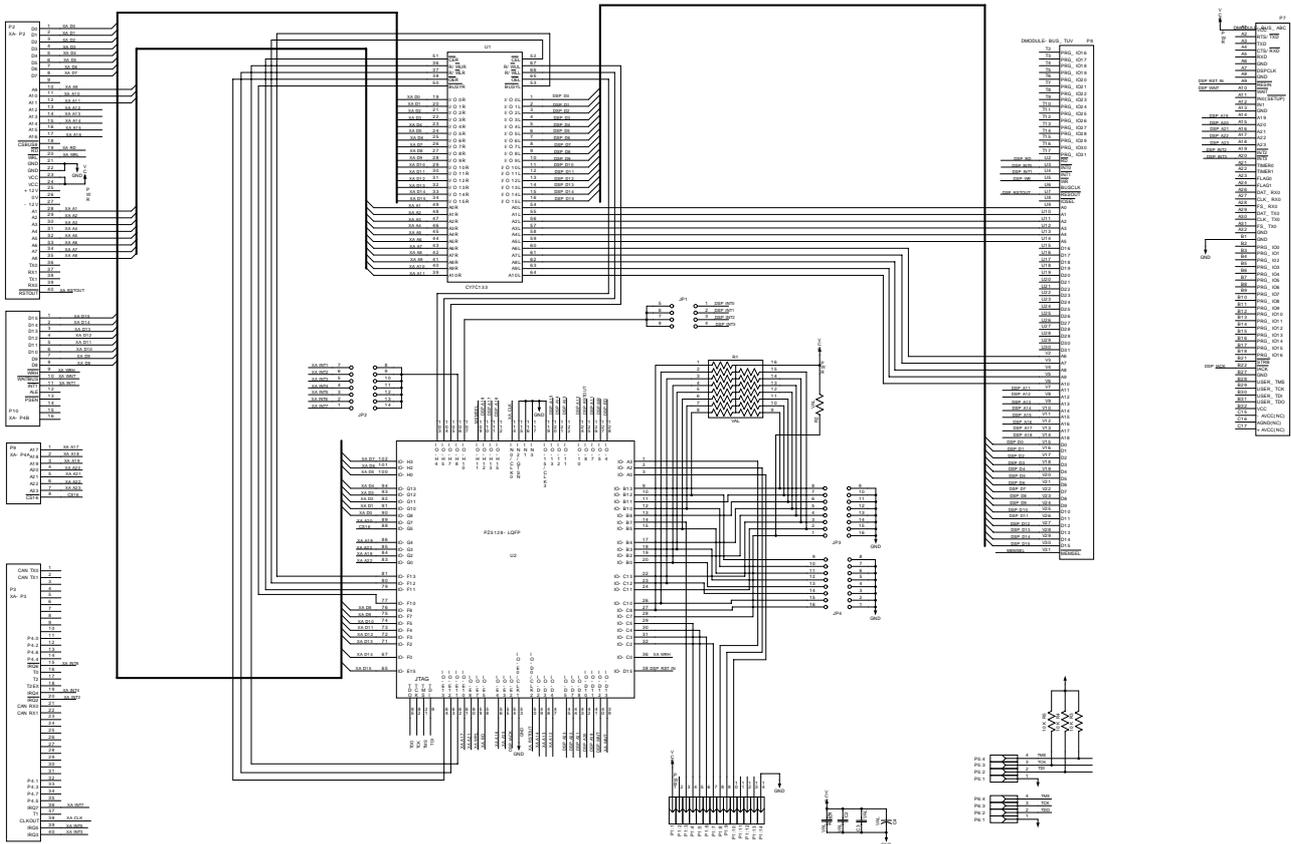


Figura H.2 – Esquema eléctrico da placa DSPIf

H.2 Desenho do PCB da Placa DSPIf

Os desenhos das duas camadas de sinal da Placa DSPIf (desenvolvida no DSOR) encontram-se nas Figura H.3 e Figura H.4:

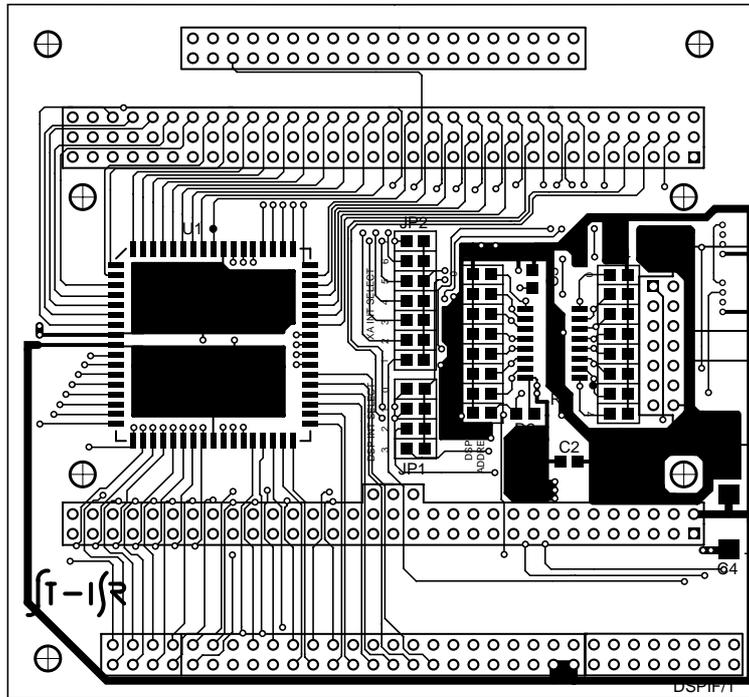


Figura H.3 – Camada Superior da DSPIf

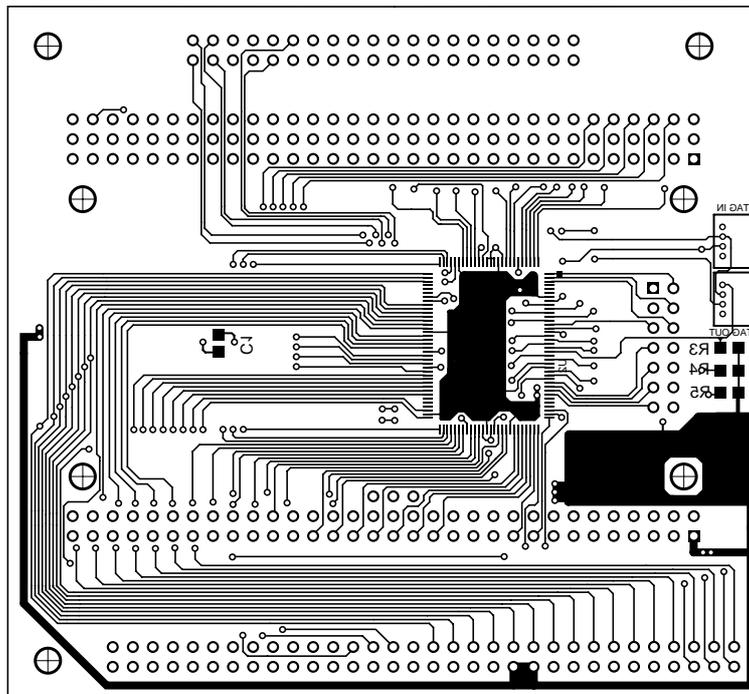


Figura H.4 – Camada inferior da DSPIf

H.3 Mapa de Memória

Tabela H.3 – Mapa de memória da placa MC-XAS3

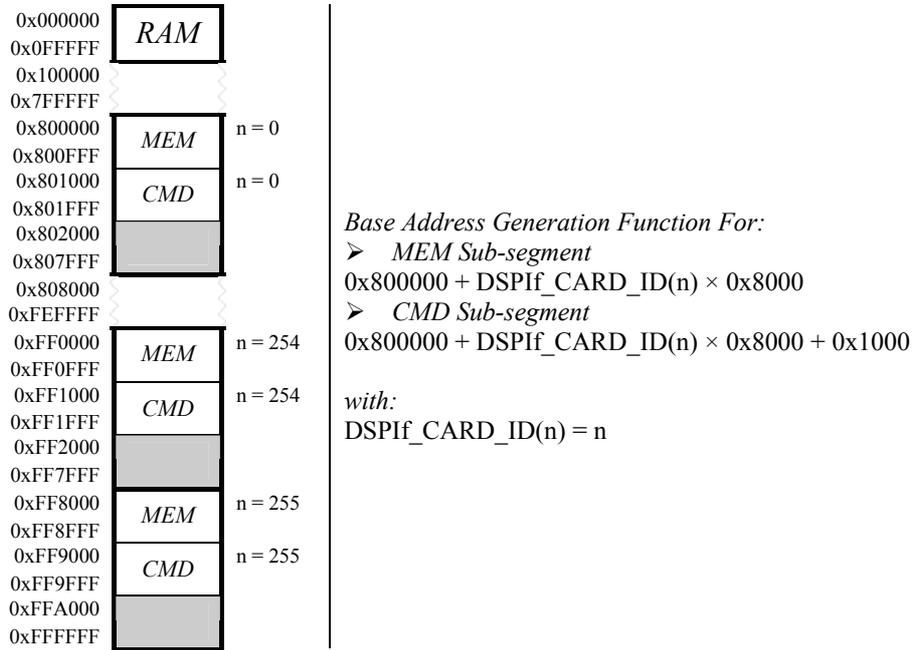


Tabela H.4 – Mapa de memória do D.Module.VC33

0x000000	ROM			
0x000FFF				
0x001000	SRAM			
0x010FFF				
0x400000	FLASH			
0x47FFFF				
0x500000	UART			
0x500007				
0x600000	Module Configuration			
0x600003				
0x700000	CPLD			
0x7000FF				
0x800000	int. RAM2			
0x803FFF				
0x804000	int. RAM3			
0x807FFF				
0x808000	int. Peripherals			
0x8097FF				
0x809800	int. RAM0			
0x809BFF				
0x809C00	int. RAM1			
0x809FBF				
0x809FC0	int. RAM1			
0x809FFF				
0x80A000	nIOSEL			
0xBFFFFFF				
0xC00000	nMEMSEL	Dual-Port RAM in DSPIf	0xFFFF00	Command Sub-Segment
0xFFEFFF				
0xFFFF00				
0xFFFF00				
0xFFFF00			0xFFFF7FF	
0xFFFF00			0xFFFF800	Data Sub-Segment
0xFFFF00			0xFFFFFFF	

H.4 Circuitos Lógicos Simbólicos de Acesso à DSPIf

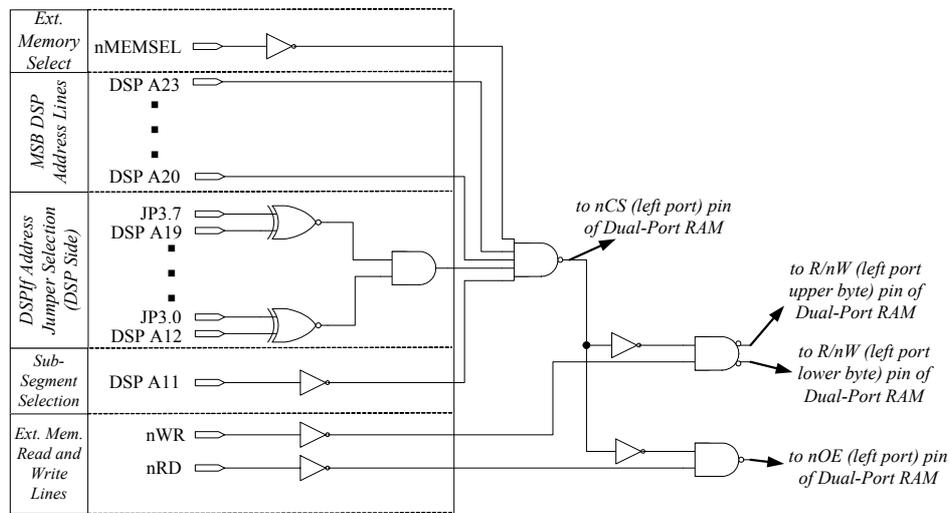


Figura H.5 – Circuito lógico simbólico implementado pela XPLA num acesso de escrita/leitura do D.Module.VC33 à DP-RAM

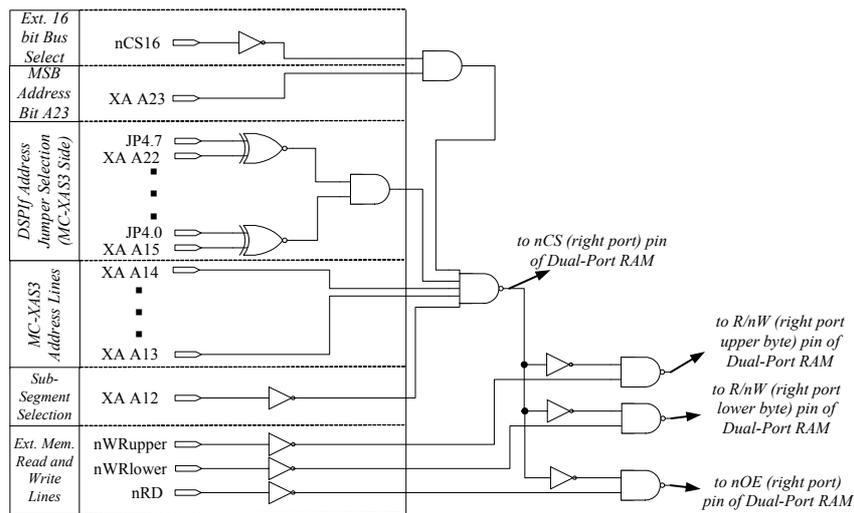


Figura H.6 – Circuito lógico simbólico implementado pela XPLA num acesso de escrita/leitura da MC-XAS3 à DP-RAM

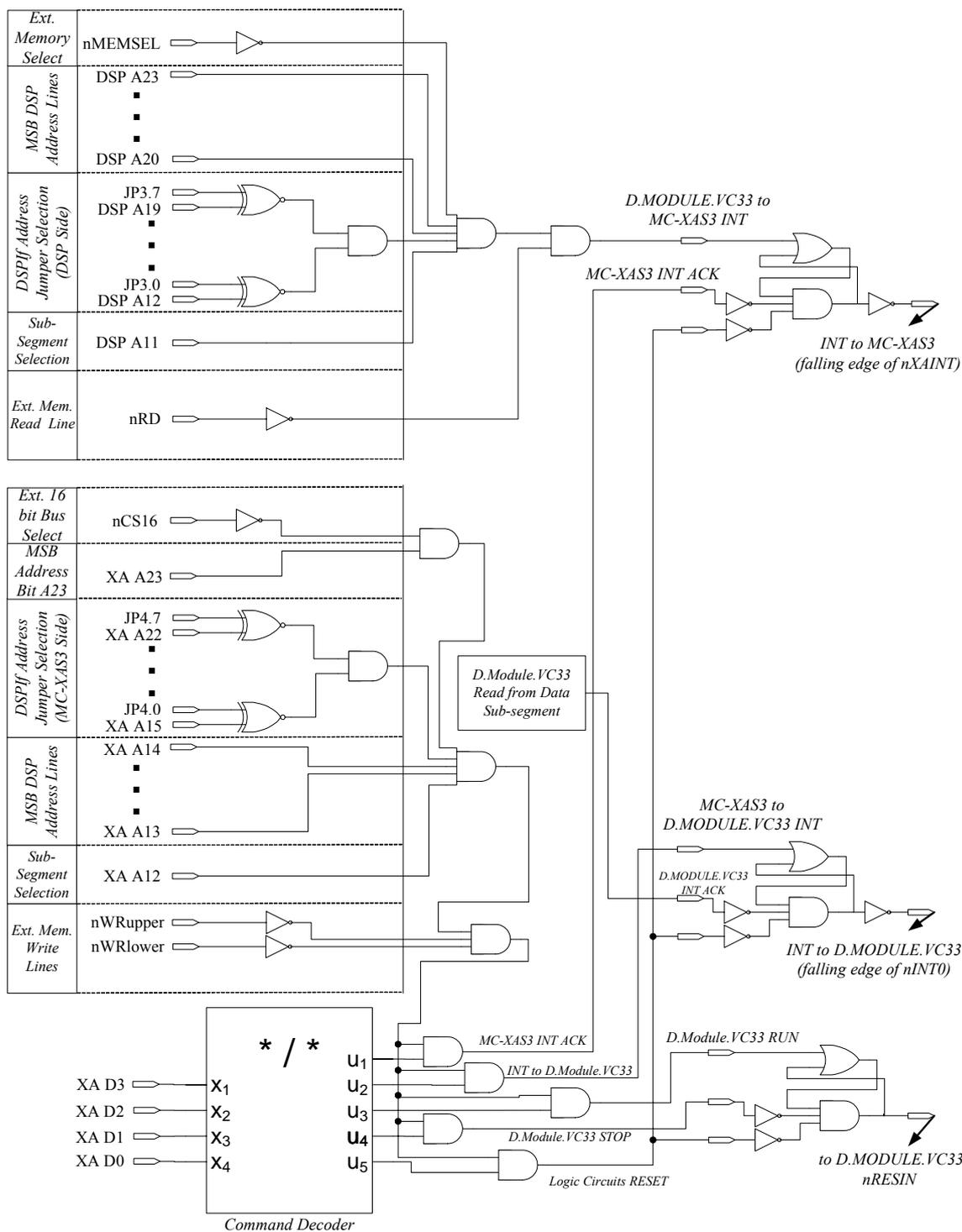


Figura H.7 – Circuito lógico simbólico implementado pela XPLA num acesso ao segmento de comandos da MC-XAS3 e do D.Module.VC33

H.5 Diagramas Temporais dos Circuitos Lógicos

Nesta subsecção ilustra-se os diagramas temporais referidos na Subsecção 3.2.4.

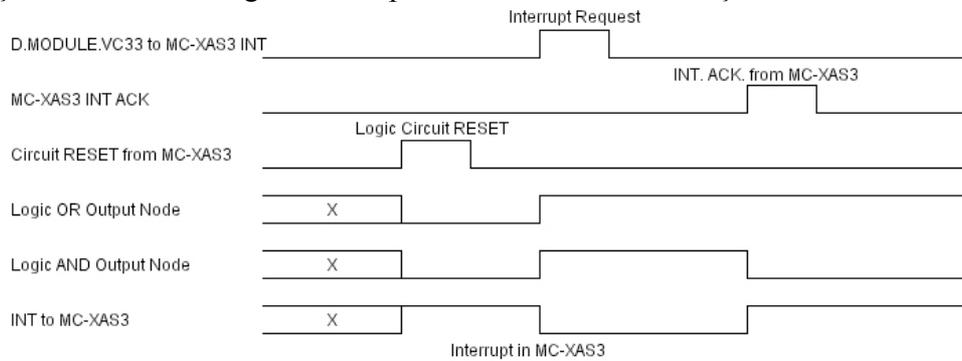


Figura H.8 – Diagrama temporal do circuito lógico assíncrono que gera a interrupção na MC-XAS3

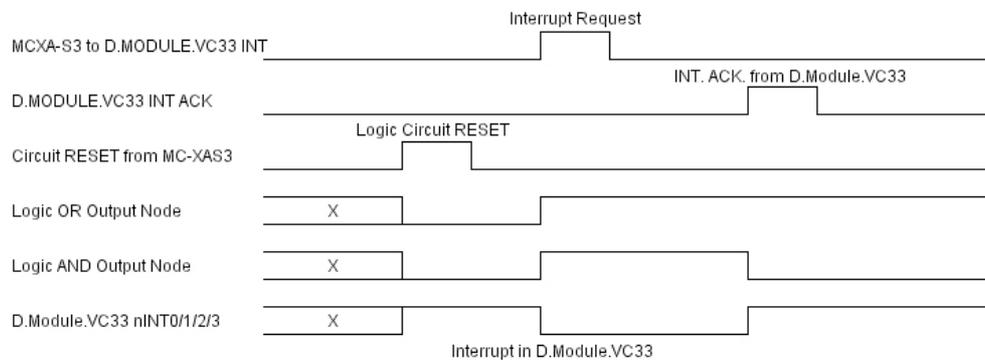


Figura H.9 – Diagrama temporal Circuito lógico assíncrono que gera a interrupção no D.Module.VC33

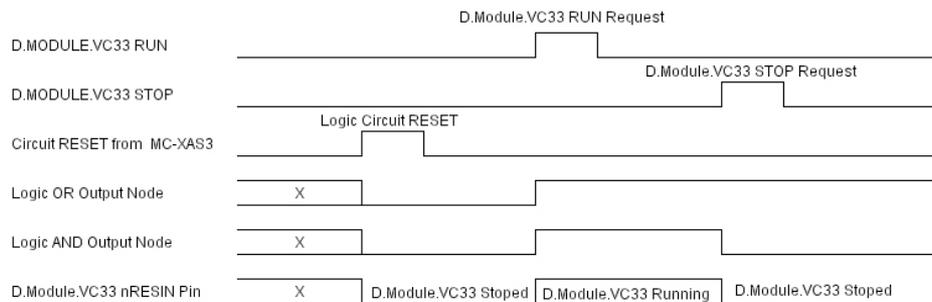


Figura H.10 – Diagrama temporal do circuito lógico de controlo do pino nRESIN do D.Module.VC33

H.6 Programa da PAL da placa DSPIf

```
MODULE template
TITLE 'Template'
```

```
XPLA PROPERTY 'tri-state XAD4:94';
XPLA PROPERTY 'tri-state XAD5:100';
```

```

XPLA PROPERTY 'tri-state XAD6:101';
XPLA PROPERTY 'tri-state XAD7:102';
XPLA PROPERTY 'tri-state XAD8:76';
XPLA PROPERTY 'tri-state XAD9:75';
XPLA PROPERTY 'tri-state XAD10:74';
XPLA PROPERTY 'tri-state XAD11:73';
XPLA PROPERTY 'tri-state XAD12:72';
XPLA PROPERTY 'tri-state XAD13:71';
XPLA PROPERTY 'tri-state XAD14:67';
XPLA PROPERTY 'tri-state XAD15:65';

```

DECLARATIONS

```

// *****
// Debug Lines
// *****
P0..P7          PIN   14,29,30,31,32,1,2,3;

// *****
// Jumper Lines Input
// *****
" XA-S3
XAJMPADDR0..XAJMPADDR7 PIN   10,12,15,18,20,23,26,27;
" DSP (TMS320VC33)
DSPJMPADDR0..DSPJMPADDR7 PIN   9,11,13,17,19,22,24,28;

// *****
// Address Lines MSBs
// *****
" XA-S3
XAA12..XAA23      PIN   47,48,49,55,56,61,84,86,89,60,83,85;
" DSP (TMS320VC33)
DSPA11..DSPA23    PIN   126,124,122,121,120,112,111,110,41,42,43,44,45;

// *****
// Data Lines
// *****
" XA-S3
XAD0..XAD15       PIN   90,91,92,93,94,100,101,102,76,75,74,73,72,71,67,65;

// *****
// XA-S3 External Access Lines
// *****
XARD              PIN   58;
XAWRL             PIN   59;
XAWRH            PIN   36;
XAWAIT           PIN   39;
XARSTOUT         PIN   50;
XACS16           PIN   88;
XACLK            PIN  114;
XAIRQ            PIN  106;

// *****
// DSP (TMS320VC33) External Access Lines
// *****
DSPRD            PIN  128;
DSPWR            PIN  127;
DSPRSTIN        PIN   38;

```

```

DSPRSTOUT      PIN 125;
DSPWAIT        PIN 40;
DSPIRQ         PIN 107;
DSPIACK        PIN 54;
DSPMEMSEL      PIN 109;

// *****
// Right Side of Dual-Port RAM Control Lines, XA-S3 Side
// *****
CER            PIN 79;
RWUR           PIN 62;
RWLR           PIN 63;
OER            PIN 64;
BUSYR          PIN 77;

// *****
// Left Side of Dual-Port RAM Control Lines, DSP (TMS320VC33) Side
// *****
CEL            PIN 80;
RWUL           PIN 105;
RWLL           PIN 104;
OEL            PIN 103;
BUSYL          PIN 81;

DSP_CMD_ADDR_VALID  NODE ISTYPE 'com';
XA_CMD_ADDR_VALID   NODE ISTYPE 'com';

A_NODE            NODE ISTYPE 'com';
B_NODE            NODE ISTYPE 'com';
AUX1              NODE ISTYPE 'com';
AUX2              NODE ISTYPE 'com';
A_NODE_2          NODE ISTYPE 'com';
B_NODE_2          NODE ISTYPE 'com';
AUX1_2            NODE ISTYPE 'com';
AUX2_2            NODE ISTYPE 'com';

C_NODE            NODE ISTYPE 'com';
D_NODE            NODE ISTYPE 'com';

// *****
// BUS Declarations
// *****
JMPXA=[XAJMPADDR7..XAJMPADDR0];
ADDRTESTXA=[XAA22..XAA15];
ADDRTESTXA_A23=[XAA23];
ADDRTESTXA_A14_TO_A13=[XAA14..XAA13];
ADDRTESTXA_A12=[XAA12];

DATAXA_D3_TO_D0=[XAD3..XAD0];

JMPDSP=[DSPJMPADDR7..DSPJMPADDR0];
ADDRTESTDSP=[DSPA19..DSPA12];
ADDRTESTDSP_A23_TO_A20=[DSPA23..DSPA20];
ADDRTESTDSP_A11=[DSPA11];

// *****
// Other Declarations
// *****

```

" MSB's to decode memory access

XA_A14_A13=[^]b00;

DSP_A23_TO_A20=[^]b1111;

XA_CMD_DATA_nRESIN_0=[^]h0; " Command to out the pin nRESIN=0

XA_CMD_DATA_nRESIN_1=[^]h1; " Command to out the pin nRESIN=1

XA_CMD_DATA_DSP_INTERRUPT=[^]h2; " Command to generate an interrupt in the DSP

XA_CMD_DATA_INT_ACK=[^]h3; " Command to generate an interrupt ACK from XA to the DSP

XA_CMD_DATA_CTRL_CIRCUITS_RESET=[^]h4; " Command to generate an interrupt ACK from XA to the DSP

EQUATIONS

XAWAIT=1;

// *****

// DSP Code

// *****

WHEN (DSPMEMSEL==0) THEN

{

P3=1;

WHEN (JMPDSP==ADDRTESTDSP) & (ADDRTESTDSP_A23_TO_A20== DSP_A23_TO_A20) &
(ADDRTESTDSP_A11==0) THEN

{

P4=1;

CEL=0;

WHEN (DSPRD==0) THEN

{

OEL=0;

}

ELSE

{

OEL=1;

}

WHEN (DSPWR==0) THEN

{

RWLL=0;

RWUL=0;

}

ELSE

{

RWLL=1;

RWUL=1;

}

}

ELSE

{

P4=0;

CEL=1;

RWUL=1;

RWLL=1;

OEL=1;

}

}

ELSE

{

{

```

P3=0;
CEL=1;
RWUL=1;
RWLL=1;
OEL=1;
}

// *****
// XA Code
// *****
WHEN (XACS16==0) THEN
{
P1=1;
WHEN (JMPXA==ADDRTESTXA) & (ADDRTESTXA_A14_TO_A13== XA_A14_A13) &
(ADDRTESTXA_A12==0) & (ADDRTESTXA_A23==1) THEN
{
P2=1;
CER=0;
WHEN (XARD==0) THEN
{
OER=0;
}
ELSE
{
OER=1;
}
WHEN (XAWRL==0) THEN
{
RWLR=0;
}
ELSE
{
RWLR=1;
}
WHEN (XAWRH==0) THEN
{
RWUR=0;
}
ELSE
{
RWUR=1;
}
}
ELSE
{
P2=0;
CER=1;
RWUR=1;
RWLR=1;
OER=1;
}
}
ELSE
{
P1=0;
CER=1;
RWUR=1;
RWLR=1;
OER=1;
}
}

```

```

}

// *****
// Interrupts
// *****

WHEN (XACS16==0) & (JMPXA==ADDRTESTXA) & (ADDRTESTXA_A14_TO_A13== XA_A14_A13) &
(ADDRTESTXA_A12==1) & (ADDRTESTXA_A23==1) & (XAWRL==0) & (XAWRH==0) THEN
{
  P5=1;
  XA_CMD_ADDR_VALID=1;
}
ELSE
{
  P5=0;
  XA_CMD_ADDR_VALID=0;
}

" INT ACK of XA
AUX1=!((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 == XA_CMD_DATA_INT_ACK));

" Interrupt from DSP to XA
AUX2=((DSPMEMSEL==0) & (JMPDSP==ADDRTESTDSP) &
(ADDRTESTDSP_A23_TO_A20==DSP_A23_TO_A20) & (ADDRTESTDSP_A11==1) & (DSPRD==0));

A_NODE=B_NODE # AUX2;
B_NODE=A_NODE & AUX1 & !((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 ==
XA_CMD_DATA_CTRL_CIRCUITS_RESET));
XAIRQ=! (B_NODE);

" ***

" INT ACK of DSP
AUX1_2=!((DSPMEMSEL==0) & (JMPDSP==ADDRTESTDSP) &
(ADDRTESTDSP_A23_TO_A20==DSP_A23_TO_A20) & (ADDRTESTDSP_A11==0) & (DSPRD==0));

" Interrupt from XA to DSP
AUX2_2=(XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 == XA_CMD_DATA_DSP_INTERRUPT);

A_NODE_2=B_NODE_2 # AUX2_2;
B_NODE_2=A_NODE_2 & AUX1_2 & !((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 ==
XA_CMD_DATA_CTRL_CIRCUITS_RESET));
DSPIRQ=! (B_NODE_2);

" ***

// *****
// DSP nRESIN Control
// *****

D_NODE= C_NODE & !((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 ==
XA_CMD_DATA_nRESIN_0)) & !((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 ==
XA_CMD_DATA_CTRL_CIRCUITS_RESET));
C_NODE=D_NODE # ((XA_CMD_ADDR_VALID==1) & (DATA_XA_D3_TO_D0 == XA_CMD_DATA_nRESIN_1));
DSPRSTIN = D_NODE;

END

```

I Conversor Digital-Analógico

Os conversores Digital/Analógico (DAC's) utilizados para o gerador analógico de sensores foram os implementados na Placa ADDAOCT desenvolvida no DSOR/ISR. O diagrama funcional desta placa ilustra-se na Figura I.1. Os DAC's disponíveis são da *Analog Devices* e têm a referência AD7841. São DAC's de 14-*Bit*. As principais características deste circuito integrado são apresentadas de seguida.

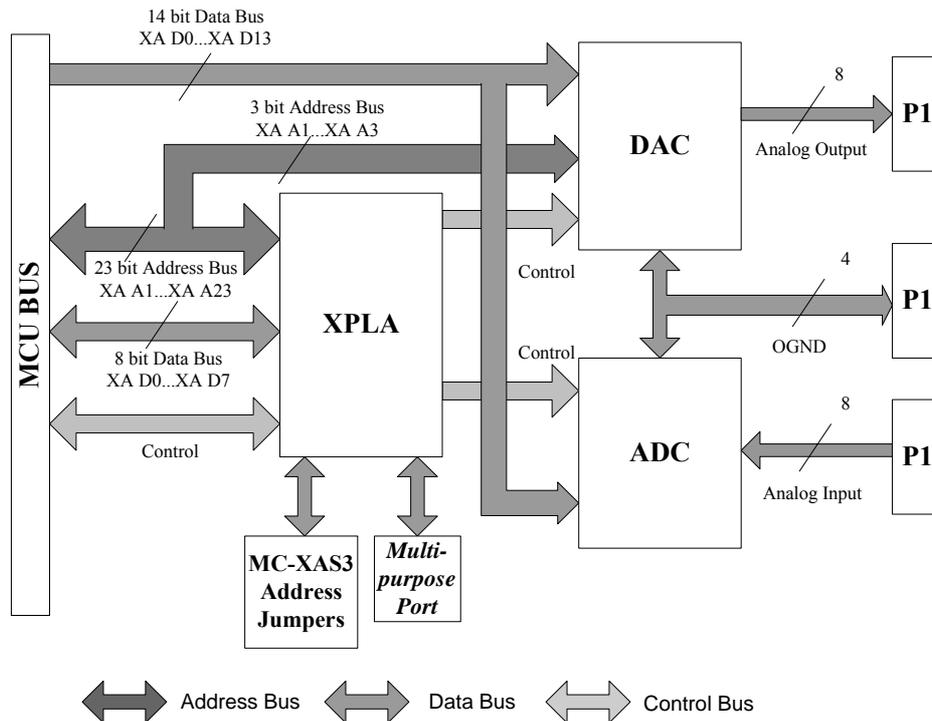


Figura I.1 – Diagrama Funcional da Placa ADDAOCT

• Características Gerais

O circuito integrado AD7841 é constituído por oito DAC's de 14-bit e permite na sua saída tensões de $\pm 10V$ com tensões de referência de $\pm 5V$. Aceita uma trama de dados paralela de 14-bit com origem no barramento externo para os registos internos quando controlado pelo pino \overline{WR} , \overline{CS} , e pelos pinos de endereçamento, $A0 - A2$.

A saída dos DAC's é actualizada sempre que chegam dados novos aos registos, para além disso, as saídas podem ser actualizadas simultaneamente, levando o sinal \overline{LDAC} a zero.

• Características Específicas

- Oito DAC's de 14-bit num único circuito integrado;
- Ajuste da tensão de *offset* para cada par de DAC's;
- Tensão de referência num intervalo de $\pm 5V$;
- Intervalo máximo na tensão de saída de $\pm 10V$;
- Tensão de funcionamento de $\pm 15V \pm 10\%$;
- 44 – Lead Metric Quad Flat Pack (MQFP).

J Placa MC-XAS3

A placa MC-XAS3 é parte de uma arquitetura, com uma topologia caracterizada pelo empilhamento de várias placas de igual dimensão partilhando o mesmo barramento, que contém dispositivos com as mais diversas aplicações, o que torna o sistema muito flexível.

- **Características Gerais**

Trata-se de uma placa microcontroladora de uso geral, com capacidade de comunicação série de tempo real (CAN 2.0) e série RS-232, sendo vocacionada para aplicações de controlo distribuído em tempo real onde o tamanho e o consumo energético são de extrema importância.

É utilizado o microcontrolador Philips XA-S3. Este circuito pertence à família *Semiconductors'XA (eXtendend Architecture)* da *Philips* e consiste num microcontrolador de 16 *bit* com 24 linhas de endereços a funcionar a 30 MHz. O microcontrolador contém um conversor AD, três *timers/counters*, a função de *Watchdog*, PCA (*Programmable Counter Array*), uma interface com o barramento I²C, duas UART's e alguns portos entrada/saída que o tornam adequado para um leque variado de funções.

- **Características Específicas**

- Um microcontrolador de 16 *bit Philips XAS3*
- Dois controladores CAN *Intel 82527*
- Uma XPLA *Philips/Xilinx PZ5128*
- Até 512 kbytes de memória *FLASH* e permite 1MB de memória RAM estática
- Uma única fonte alimentação de 5V
- Até 15 Mbytes de memória externa mapeada para periféricos
- Compatibilidade com dispositivos de 8-*bit* utilizados na placa MCCAN

A informação anterior foi extraída de [XADS01].

Na Figura J.1 apresenta-se o diagrama de blocos da placa.

Uma das formas de interface da placa com o exterior (entrada e saída digital e entrada analógica) é feita pelo porto 5 do *Philips XAS3*. O porto 5 (do XA-S3) tem todos os seus 8 pinos disponíveis na Ficha P1 da MC-XAS3. Este porto pode realizar interface digital com o exterior (I/O) e conversão Analógica/Digital (I) de 8 ou 10 *bits*.

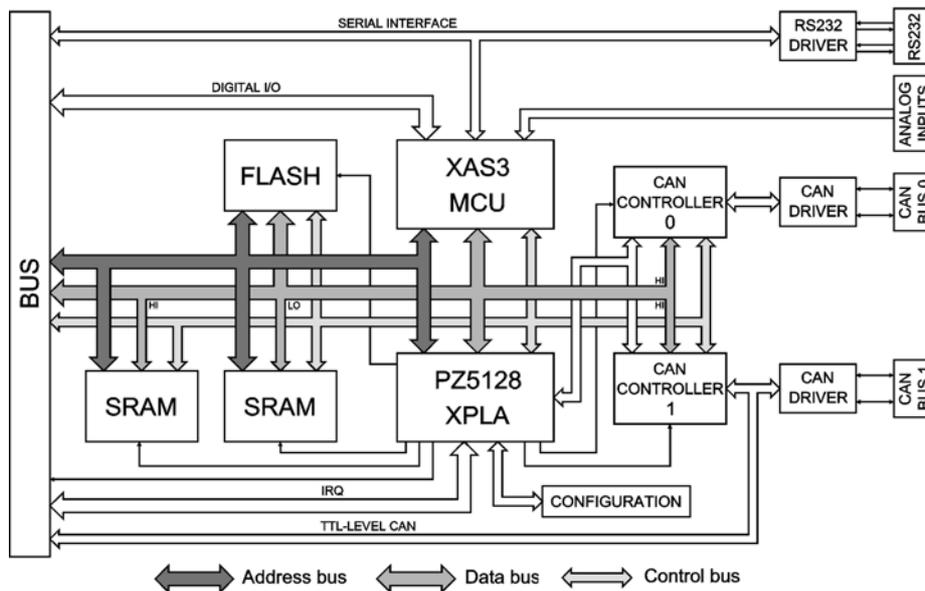


Figura J.1 – Diagrama de blocos da placa MC-XAS3 [XARI]

Tabela J.1 – Pinos Disponíveis na Ficha P1

Pino	11	12	13	14	15	16	17	18	19	20
Função	AVDD				AVSS	P5.1/AD1	P5.3/AD3	P5.5/AD5	P5.7/AD7	AVREF+
Pino	1	2	3	4	5	6	7	8	9	10
Função		AVREF-			AVSS	P5.0/AD0	P5.2/AD2	P5.4/AD4	P5.6/AD6/SCL	AVSS

- **Características do microcontrolador Philips XAS3**

- Compatibilidade com núcleo 8051.
- Espaço de endereçamento de 24 *bits* (16 MB de código e dados).
- CPU estático de 16-*bit*.
- Instruções rápidas de multiplicação e divisão.
- Conversor analógico – digital de 8-*bit* de resolução e 8 canais.
- Duas *enhanced UARTs* com *baud rates* independentes.
- Frequência de relógio do microprocessador até 30 MHz.
- Duração típica de 100ns por instrução.
- Modos de operação com poupança de energia.
- Interface CAN

Suporte para CAN na Especificação 2.0.

- Comunicações CAN até 1 Mbit/segundo.
- Duas interfaces CAN independentes.
- Controlador CAN INTEL 82527 a 16MHz.
- Philips 82C250 CAN transceiver com *slope control*.
- Capacidades de endereçamento.
- Interface de 16-*bit*.

K D.SignT.D.Module.VC33

K.1 Características da Placa D.Module.VC33

- **Características Gerais**

O *D.Module.VC33* é uma placa de processamento digital de sinal de vírgula flutuante de desempenho médio, que pertence à família *D.Module*. O seu objectivo são aplicações embebidas do tipo *stand-alone* que requerem funcionalidade e flexibilidade máximas e baixo consumo de potência. Os requisitos de baixo consumo tornam possível a sua utilização em aplicações móveis alimentadas a baterias.

Dispõem de um conjunto de funções denominado por *D.Module.BIOS* que permite tratar da programação de baixo nível de *hardware*. Uma *CPLD* programável pelo utilizador providencia 32 sinais de entrada/saída que possibilitam o interface flexível com dispositivos ou barramentos *standard*. A manutenção em campo é facilitada pela *Set-Up Utility* que permite armazenar parâmetros específicos a cada aplicação por meio de ficheiros de texto recorrendo à memória *Flash*.

É compatível ao nível de código objecto com o *DSP TMS320C31* o que permite duplicar a performance de sistemas antigos já que se passa de 75 *MFLOPS* (*C31*) para 150 *MFLOPS* (*VC33*).



Figura K.1 – Aspecto exterior do *D.Module.VC33*

- **Características Específicas**

- *DSP TMS320VC33* a 150 MHz, vírgula flutuante 32/40 Bit, vírgula fixa 24/32 Bit, *DMA*;
- Memória
 - 34k x 32 Bit de memória de acesso dual interna ao *DSP*;
 - 64k ou 256k x 32 Bit de memória *SRAM* na Placa, 1 wait state (33ns);
 - 512 kByte de memória *Flash* não volátil , arquitectura de sectores uniformes;
- 1 *UART* 16C650 com *baudrates* até 460kBaud, *FIFOs* de transmissão e recepção 32 words , *RTS/CTS* e *Xon/Xoff Auto-Flow-Control*, suporte *DMA*, *DSP wake-up* no mode *idle2*, interface *RS232* (opt. *RS422/485*);
- 2 *Timers* internos do *DSP* de 32 Bit, geração de relógio, pulsos ou contagem de eventos externos;
- 1 porto série síncrono de transmissão e 1 porto série síncrono de recepção interno ao *DSP* com taxa de transmissão até 32 *Mbits/s*, 2 *timers* de 16 bits para geração de relógio e sincronização de tramas ou relógios externos;
- Barramento externo de 32 bits de dados 8M de espaço de endereçamento;
- 32 portos de entrada/saída programáveis pelo utilizador (*CPLD XCR3064XL* programável *in-system* com 64 macrocélulas e pinos de entrada, saída ou bidirecionais);

- 4 interrupções externas, 7 interrupções internas com disparo ao flanco descendente multiplexadas com a *UART* e a *CPLD*.
- Tensões de alimentação únicas de 3.3 a 5V, com consumo em modo *idle* de 6 mA, consumo típico de 100 mA com todas as interfaces tolerantes a 5V;
- *Software D.Module.BIOS* que permite fazer inicialização e configuração do módulo, inicialização da *UART* e das comunicações, programar a memória *Flash* e com funções de *Boot-Loader*;
- *Software Set-Up Utility* que permite fazer o *upload* de ficheiros ASCII e binários, programar a *CPLD*, configurar o módulo, *memory dump*, alterar a memória e executar programas /funções;
- *Watchdog* e supervisor de tensões de alimentação, interface *JTAG* para *in-circuit emulation*

K.2 Texas Instruments TMS320VC33 DSP

• Características Gerais

O *TMS320VC33* é um processador digital de sinal de vírgula flutuante de 32 *bits*, fabricado com a tecnologia 0.18 μm *four-level-metal CMOS Timeline*. O *TMS320VC33* faz parte da geração *TMS320C3X* de *DSP's* da *Texas Instruments*.

O barramento interno dos *TMS320C3X's* e o conjunto de instruções especiais de processamento digital de sinal permite executar 150 *MFLOPS*. A optimização, em termos de velocidade de execução, é fruto da implementação em *hardware* de funções habitualmente implementadas em *software* ou *microcode*. Este tipo de abordagem permite performances até agora inexistentes num único *chip*. O *TMS320VC33* pode executar multiplicações em paralelo com operações da *ALU* em inteiros ou vírgula flutuante num ciclo único. O *TMS320VC33* suporta uma vasta gama de aplicações que vão desde processador principal a co-processador dedicado, disponibilizando um largo espaço de endereçamento, suporte de linguagens de alto nível, modos de endereçamento e conjunto de instruções flexíveis. Com o *TMS320VC33* ganha-se em relação ao *TMS320C31* 1Mbits de *SRAM* interna. Dispõe também de um porto série, 2 *timers* de 32 *bit* entre outros.

• Características Específicas

- Ciclo de instrução de 13 ns;
- 150 Milhões de Operações de Vírgula Flutuante por Segundo;
- 34K×32 *bit* (1.1 *Mbit*) de *SRAM* de acesso dual configurada em 2×16K mais 2×1K blocos para aumentar a performance interna;
- Baixo consumo: < 200 mW @ 150 *MFLOPS*;
- Operações de inteiros de 16-/32 *bit* e de vírgula flutuante de 32-/40 *bit*;
- Interrupções externas seleccionáveis *EDGEMODE*;
- Endereços de 24 *bit* e instruções de *words* de 32 *bit*;
- 8 Registos de precisão extendida;
- Periféricos mapeados internamente;
- 1 porto série;
- 2 *timers* de 32 *bit*;
- Acesso directo à memória (*DMA*);
- Execução paralela da *ALU* e do multiplicador num único ciclo;
- 2 Modos de baixo consumo;
- *Boot-Program Loader*;

A informação anterior foi extraída de [TIUG] e [TIDs].

Bibliografia

Pais, M. Alves, P., *Sistema distribuído de Microcontroladores para Controlo em Tempo Real*, Trabalho de Final de Curso, Instituto Superior Técnico, Portugal, 1999.

Nise, N., *Control Systems Engineering*, Third Edition, John, Wiley & Sons, 2000

Referências

- [ADDS00] ADS1210 *Data Sheet*. Burr-Brown, Maio 2000
- [AGNC] *American GNC Corporation*. (<http://www.americangnc.com>)
- [ASPa00] Alves, J., Silvestre, C., Pascoal, A. *A Distributed Architecture For Real Time Control*. IST-ISR, Janeiro 2000.
- [Atha01] Athans, M., *Approximation of Continuous-Time Linear Stochastic Systems by Discrete-Time Equivalents*, MIT & IST, 2001.
- [Azin02] Azinheira, J., *Sensores e Actuadores*, Instituto Superior Técnico, 2002.
- [BCD04] Baltazar, F., Cardeira, B., Dias, P., *Manuais da Arquitectura de Navegação*
- [Bier77] G. J. Bierman; *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.
- [Bowd95] Bowditch, N., *The American Practical Navigator*. Defense Mapping Agency, Hydrographic/Topographic Center, Bethesda, Maryland, USA, 1995.
- [BrHw97] Brown, R. and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering*, Third Edition, John Wiley & Sons, 1997.
- [CaVa03] Calvário, J. e Vasconcelos, J., *Estratégias de Fusão Sensorial para Sistemas de Navegação com Aplicação a Helicópteros*, Trabalho de Final de Curso, Instituto Superior Técnico, Portugal, 2003.

- [CaSm98] Caruso, M. J., Smith, C. H., *A New Perspective on Magnetic Field Sensing*, Honeywell, SSEC Nonvolatile Electronics, Inc. 12001 State Highway 55 11409 Valley View Road Plymouth, MN 55441 Eden Prairie, MN 55344 (http://www.ssec.honeywell.com/magnetic/datasheets/new_pers.pdf)
- [CoMo02] Morgado, R., Costa, T. *Implementação da Instrumentação de um Helicóptero Autônomo*, Trabalho de Final de Curso, Instituto Superior Técnico, Portugal, 2002.
- [Crai89] Craig, J., *Introduction to Robotics, Mechanics and Control*, Second Edition, Addison-Wesley, New York, 1989.
- [DSXi00] XCR5128 *Data Sheet*. Xilinx, Agosto 2000
- [Denn79] Denne W., *Magnetic Compass Deviation and Correction*, Brown, Son & Ferguson, Ltd., Glasgow 1979.
- [ElGe01] Elkaim, G.H., Gebre-Egziabher, D., Powell, J.D., Parkinson, W.B., *A Non-Linear, Two-Step estimation Algorithm for Calibrating Solid-State Strapdown Magnetometers*, 8th St. Petersburg Conference on Navigation Systems (IEEE/AIAA), St. Petersburg, Russia, May 27-May 31, 2001.
- [FiPi99] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher, *Direct least-squares fitting of ellipses*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5), 476--480, May 1999
- [Kell94] Kelly, A., *Modern Inertial and Satellite Navigation Systems*, Carnegie Mellon University, 1994.
- [lectGPS] G338 – Introduction to GIS, October 6, 2003 (http://www.indiana.edu/~gis/courses/g338/lectures/lecture_gps.pdf)
- [Lips02] Lipson, B., *Introduction to Sensors*, Stanford University, 2002.
- [Luen64] Luenberger, D., "Observing the state of a linear system", *IEEE Transactions on Military Electronics*, Vol. 8, pp. 74-80, 1964.
- [Mach03] Mach, J. J., *Toward Auto-Calibration of Navigation Sensors for Miniature Autonomous Underwater Vehicles*, Master Thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003
- [Mayb79] Maybeck, P. S. *Stochastic Models, Estimation, and Control, Vol. 1*. Academic Press, 1979.
- [NGDC] National Oceanic and Atmospheric Administration, National Geophysical Data Center (<http://www.ngdc.noaa.gov/>)

- [PiFe02] Pires, A., Ferraz, N., *Sistema de Navegação para Helicóptero Autónomo por Integração INS/GPS*, Trabalho de Final de Curso, Instituto Superior Técnico, Portugal, 2002.
- [Ribe02] Ribeiro, I., *Análise de Sistemas Lineares*, IST Press, Lisboa, 2002.
- [Silv95] Silvestre, C., *Modelação e Controlo de Veículos Submarinos Autónomos*, Tese de Mestrado, Universidade Técnica de Lisboa, Lisboa 1995
- [TIUG] *Texas Instruments TMS320C3X User's Guide (literature number spru031e)*
- [TIDs] *Texas Instruments TMS320VC33 Datasheet (literature number SPRS087D)*
- [XCDS00] *XCR5128 Data Sheet*. Xilinx, Agosto 2000
- [VCUG01] *D.Module.VC33 User's Guide, Doc Rev. 1.3*, 2001
- [XADS01] *XA-S3 Data Sheet*. Philips, Dezembro 2001
- [XARI] *MCXAS3 Microcontroller and CAN 2.0 B interface board*, Relatório Interno, Instituto de Sistemas e Robótica, Instituto Superior Técnico, Portugal.