

**GILBERTO TAKASHI UEHARA
RYNALDO ZANOTELE HEMERLY DE ALMEIDA**

Nota final

8,7 (oitava nota)

KAU

22/01/04

**ARQUITETURA DE CONTROLE PARA
VEÍCULOS SUBMARINOS NÃO TRIPULADOS**

Trabalho apresentado à Escola Politécnica da
Universidade de São Paulo para a obtenção de
Títulos de Engenheiro

São Paulo

2003

**GILBERTO TAKASHI UEHARA
RYNALDO ZANOTELE HEMERLY DE ALMEIDA**

**ARQUITETURA DE CONTROLE PARA
VEÍCULOS SUBMARINOS NÃO TRIPULADOS**

Trabalho apresentado à Escola Politécnica da
Universidade de São Paulo para a obtenção de
Títulos de Engenheiro

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Newton Maruyama

São Paulo

2003

A todos aqueles que acreditam

AGRADECIMENTOS

Aos nossos pais e irmãos, fonte eterna de amor e carinho, por sempre estarem ao nosso lado em todas as buscas em que nos projetamos.

Ao nosso orientador Prof. Dr. Newton Maruyama por nos ter mostrado o caminho da liberdade de criar.

Aos colegas de projeto e amigos (em ordem alfabética), Douglas, Eric, Frederico, Juan, Luiz Henrique, Marcelo, Paulo e Vitor, pelo suporte técnico e, principalmente, humano.

E finalmente, a todos os inestimáveis amigos da Turma de Engenharia Mecatrônica 98 que compartilharam momentos de alegria e mais alegria e se fortaleceram em apoio mútuo frente aos obstáculos desta grande jornada de formação de engenheiros, cidadãos, pessoas, ..., enfim, grandes nomes do século XXI.

RESUMO

O presente trabalho trata do estudo, projeto e implementação de Arquitetura de Controle para veículo submarino não tripulado. Realiza-se um levantamento histórico de utilização desses veículos e seus grandes potenciais de utilização na exploração submarina, principalmente em plataformas petrolíferas de alta profundidade. Verificada essa possibilidade ou mesmo necessidade, estuda-se as características necessárias em uma arquitetura de controle e se faz um projeto abstrato o suficiente para utilização em diversos veículos. A implementação da arquitetura é realizada para o veículo submarino em desenvolvimento na Escola Politécnica utilizando-se ferramentas para sistemas em tempo real disponíveis no pacote Simulink do MatLab. Foram realizados testes secos demonstrando a correta integração dos dispositivos do sistema e testes de simulação computacional que verificaram a capacidade do veículo em responder corretamente a comandos de movimentação manual (enviados por *joystick*), estabilização de inclinação e cumprimento de trajetória em linha reta.

ABSTRACT

This work is a study, project and implementation of a Control Architecture for an Unmanned Underwater Vehicle. It's done a study of utilization of those vehicles from their origin up to now and its potential use in the oil exploitation in deepwater offshore production systems. The possibility or necessity of use underwater vehicles justifies the development of a Control Architecture. It's achieved an abstract project that may be employed in many vehicles. The implementation of the architecture concerns the vehicle in development at the Escola Politécnica (University of São Paulo) and it's made use of real-time systems tools available in the Simulink/MatLab software. Out of the water tests were done to verify the integration of the system's parts and computational simulation was used to verify the capability of the vehicle to follow movement commands (by joystick input), attitude stabilization and ordered movements in strait horizontal line.

SUMÁRIO

LISTA DE FIGURAS	iv
LISTA DE TABELAS	vii
LISTA DE TABELAS	vii
LISTA DE ABREVIATURAS E SIGLAS.....	viii
LISTA DE SÍMBOLOS.....	ix
LISTA DE EQUAÇÕES.....	xi
1 INTRODUÇÃO.....	1
2 HISTÓRICO	3
3 ATUALIDADE.....	5
3.1 Veículos	5
3.2 Aplicações.....	7
4 ESTUDO DE VIABILIDADE DE VEÍCULOS SUBMARINOS NÃO TRIPULADOS E NECESSIDADE DE ARQUITETURAS DE CONTROLE	9
4.1 Características do Ambiente e Forças Envolvidas.....	10
4.2 Verificação da capacidade de atuadores e sensores.....	11
4.3 Necessidade de uma Arquitetura de Controle.....	11
5 OBJETIVO E ESPECIFICAÇÃO DOS REQUISITOS	12
6 DESCRIÇÃO DO PROJETO EM DESENVOLVIMENTO NA ESCOLA POLITÉCNICA.....	13
6.1 Estrutura Tubular e Vasos de Pressão.....	14
6.2 Dispositivos de Atuação	16
6.3 Dispositivos de Sensoriamento	18
6.4 Dispositivo de Realização do Controle.....	20
6.5 Conjunto de Baterias.....	21
6.6 Dispositivos de Comunicação com o Meio Externo e Cabo Umbilical.....	22
7 FUNCIONAMENTO E COMPORTAMENTO ESPERADOS DO VEÍCULO.....	24
7.1 Modos de Funcionamento.....	24
7.1.1 Inicialização e Modo Inativo	24
7.1.2 Modo Manual.....	24

7.1.3	Modo de <i>Hovering</i>	25
7.1.4	Modo Automático.....	25
7.1.5	Modo de Teste.....	25
7.2	Alternância entre Modos.....	26
7.3	Interface com o Operador.....	27
7.3.1	Monitoração.....	27
7.3.2	Comando.....	27
8	ESTUDO E PROJETO DE ARQUITETURA DE CONTROLE	30
8.1	Características desejadas numa Arquitetura de Controle.....	30
8.2	Arquitetura de Controle Proposta.....	31
8.3	Módulo Gerenciador do Fluxo de Informações.....	33
8.4	Módulo Supervisor.....	35
8.5	Módulo de Alocação de Propulsão.....	37
8.6	Módulo Gerador de Referência.....	39
9	SELEÇÃO DE FERRAMENTAS	41
9.1	Simulink / Stateflow.....	41
9.1.1	Real-Time Workshop.....	41
9.2	Constellation e VxWorks.....	42
9.2.1	Constellation.....	42
9.2.2	VxWorks.....	43
10	IMPLEMENTAÇÃO DA ARQUITETURA DE CONTROLE	46
10.1	Ambiente de Simulação.....	46
10.2	Plataforma real de utilização.....	49
10.3	Interface com o operador.....	50
10.3.1	Interface no Ambiente de Simulação.....	51
10.3.2	Interface na Plataforma Real.....	52
10.4	Driver do Joystick.....	53
10.5	Módulos de Comunicação.....	53
10.6	Módulo Supervisor.....	55
10.7	Módulo Controlador.....	57
10.7.1	Cálculo dos parâmetros dos controladores.....	60
10.8	Módulo Gerador de Referência.....	71

10.9	Módulo de Alocação de Propulsão	73
10.10	Driver dos Propulsores.....	74
10.11	Driver dos Sensores	76
11	RESULTADOS	79
11.1	Estabilização de Inclinação.....	79
11.2	Resposta a comandos de manuais de movimentação.....	80
11.3	Cumprimento de Missão	82
11.3.1	Missão Utilizando Oito Propulsores	82
11.3.2	Missão com oito propulsores e compensação de correnteza	84
11.3.3	Missão com sete propulsores e compensação de correnteza	86
12	CONCLUSÃO	89
	REFERÊNCIAS BIBLIOGRÁFICAS.....	91

LISTA DE FIGURAS

Figura 1.1 - Ilustração do UUV da Escola Politécnica.....	2
Figura 2.1 - Veículo KAIKO (franco-japonês) detentor do recorde de 10911,4 metros de profundidade	4
Figura 3.1 – Veículo de baixo custo Quest.....	6
Figura 3.2 - Veículo de baixo custo Hyball	6
Figura 3.3 - Veículo Grande Voyager.....	6
Figura 3.4 - Veículo Grande MaxROVER	6
Figura 3.5 - Recuperação de Dispositivo com auxílio de um ROV.....	8
Figura 6.1 - Montagem Experimental com estrutura do veículo, eletrônica a ser embarcada, baterias e propulsores	13
Figura 6.2 - Detalhe da eletrônica embarcada (PC104, sensores, etc).....	14
Figura 6.3 - Vista Frontal do UUV	15
Figura 6.4 - Vista Superior do UUV.....	15
Figura 6.5 - Propulsor	17
Figura 6.6 - Bússola.....	18
Figura 6.7 - Inclinômetros.....	19
Figura 6.8 - PC104.....	21
Figura 8.1 – Diagrama lógico do software da arquitetura de controle.....	32
Figura 9.1 – Esquema do Ciclo de Desenvolvimento com o Constellation.....	43
Figura 10.1 - Sistema de Simulação da Arquitetura de Controle	47
Figura 10.2 - Diagrama do Controle a ser embarcado.....	47
Figura 10.3 - Esquema de montagem física do sistema.....	49
Figura 10.4 - Esquema do funcionamento lógico do sistema.....	49
Figura 10.5 - Tela de Interface com operador.....	51
Figura 10.6 - Tela do programa de comunicação do computador de apoio.....	52
Figura 10.7 - Diagrama de Estados implementado no StateFlow.....	56
Figura 10.8 - Ilustração com sistemas de referência inercial e solidário ao veículo .	58
Figura 10.9 - Diagrama do módulo controlador	59
Figura 10.10 - Lugar das raízes para rotação em malha fechada para um controlador do tipo P	61

Figura 10.11 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo P.....	62
Figura 10.12 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar)	63
Figura 10.13 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD.....	63
Figura 10.14 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar) e incerteza de 10% em parâmetro da planta.....	65
Figura 10.15 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD e incerteza de 10% em parâmetro da planta.....	65
Figura 10.16 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar) e alteração no pólo da planta	66
Figura 10.17 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD e alteração no pólo da planta.....	67
Figura 10.18 - Lugar das raízes para velocidade angular em malha fechada para um controlador do tipo PI	68
Figura 10.19 - Resposta a degrau unitário para velocidade angular em malha fechada para um controlador do tipo PI	68
Figura 10.20 - Resposta a rampa para posição linear em malha fechada	69
Figura 10.21 - Lugar das raízes para velocidade linear em malha fechada para um controlador do tipo PI	70
Figura 10.22 - Resposta a degrau unitário para velocidade linear em malha fechada para um controlador do tipo PI	71
Figura 10.23 - Diagrama do módulo gerador de referência.....	72
Figura 10.24 - Diagrama do módulo alocador de propulsão	74
Figura 10.25 - Detalhe do driver dos propulsores e dos parâmetros	75
Figura 10.26 - Detalhe do driver dos sensores e dos parâmetros.....	77
Figura 11.1 - Estabilização de Inclinação entrando em Hovering.....	80
Figura 11.2 - Referência e resposta de velocidades lineares em modo manual.....	81
Figura 11.3 - Referência e resposta de velocidades angulares em modo manual.....	82

Figura 11.4 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores).....	83
Figura 11.5 - Referência e resposta de posições lineares em missão (oito propulsores)	84
Figura 11.6 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores e compensação de correnteza)	85
Figura 11.7 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores e compensação de correnteza)	86
Figura 11.8 - Referência e resposta de posições lineares em função do tempo em missão (sete propulsores e compensação de correnteza)	87
Figura 11.9 - Referência e resposta de posições lineares em função do tempo em missão (sete propulsores e compensação de correnteza)	88

LISTA DE TABELAS

Tabela 3.1 - Classificação dos veículos submarinos.....	5
Tabela 7.1 - Alternância entre modos de funcionamento	26
Tabela 7.2 - Relações entre o movimento do joystick e o respectivo comando de ação	28
Tabela 8.1 - Sinais de entrada e saída do módulo gerenciador do fluxo de informações.....	35

LISTA DE ABREVIATURAS E SIGLAS

- **AUV** veículo submarino autônomo (*Autonomous Underwater Vehicle*);
- **ROV** veículo operado remotamente (*Remotely Operated Vehicle*);
- **UAV** veículo aéreo não tripulado (*Unmanned Aerial Vehicle*);
- **UGV** veículo terrestre não tripulado (*Unmanned Ground Vehicle*);
- **UV** veículo não tripulado (*Unmanned Vehicle*);
- **UUV** veículo submarino não tripulado (*Unmanned Underwater Vehicle*);

LISTA DE SÍMBOLOS

- $\mathbf{B}$ matriz de configuração de propulsores;
- $\mathbf{B}^T$ matriz transposta de $\mathbf{B}$;
- $\mathbf{B}^+$ pseudo-inversa de $\mathbf{B}$, para matriz $\mathbf{W}$ igual a matriz identidade;
- $\mathbf{B}_W^+$ pseudo-inversa de $\mathbf{B}$, em função da matriz $\mathbf{W}$;
- $\mathbf{F}$ resultante das forças restaurativas e de propulsão;
- $\mathbf{F}_{\text{res}}$ componente de força (ou momento) restaurativo num grau de liberdade;
- $\mathbf{F}_{\text{prop}}$ componente de força (ou momento) de propulsão num grau de liberdade;
- $\mathbf{F}_{\text{prop}_i}$ força de propulsão de um dos propulsores;
- $\mathbf{G}_c$ função de transferência dos controladores;
- $\mathbf{J}$ energia associada ao conjunto de acionamento de propulsores;
- $\mathbf{J}$ momento de inércia em cada grau de rotação;
- $\mathbf{M}$ inércia (linear ou angular) num grau de liberdade;
- $\mathbf{T}$ resultante dos momentos restaurativos e de propulsão;
- $\mathbf{U}$ velocidade da corrente marítima;
- $\mathbf{U}_d$ componente da velocidade da corrente marítima devida diferença de densidade em diferentes porções do oceano;
- $\mathbf{U}_m$ componente da velocidade da corrente marítima devida a circulação de água na região conforme a geografia local;
- $\mathbf{U}_s$ componente de Stokes da velocidade da corrente marítima;
- $\mathbf{U}_{\text{set-up}}$ componente da velocidade da corrente marítima devida a fenômenos como tempestades;
- $\mathbf{U}_t$ componente da velocidade da corrente marítima devido às ondas;
- $\mathbf{U}_w$ componente da velocidade da corrente marítima devido à ação do vento local;
- $V_{\text{água}}$ volume de água deslocado pelo veículo (empuxo);
- $\mathbf{W}$ matriz de pesos de utilização de propulsores;
- k_p constante de propulsão
- k_v constante de arrasto hidrodinâmico em cada grau de liberdade;
- k_{va} constante de arrasto hidrodinâmico em cada grau de translação;

- $k_{v,i}$ constante de arrasto hidrodinâmico em cada grau de rotação;
- u sinais de atuação dos propulsores;
- u_i sinais de atuação de um dos propulsores;
- τ conjunto de força e momento desejados;

LISTA DE EQUAÇÕES

$\tau = B \cdot u$	Equação 8.1.....	37
$J = \frac{1}{2}u^T W u$	Equação 8.2.....	38
$B_W^+ = W^{-1} B^T (B W^{-1} B^T)^{-1}$	Equação 8.3.....	38
$B^+ = B^T (B B^T)^{-1}$	Equação 8.4.....	38
$u = B^+ \cdot \tau$	Equação 8.5.....	38
$M \ddot{x} = -k_v \left(\dot{x} - \dot{x}_c \right) + F_{res} + F_{prop}$	Equação 10.1.....	48
$F_{prop_i} = k_p u_i u_i $	Equação 10.2.....	48
$X = \frac{F}{s(s.M + k_{vl})}$	Equação 10.3.....	60
$\Theta = \frac{T}{s(s.J + k_{va})}$	Equação 10.4.....	60
$V = \frac{F}{s.M + k_{vl}}$	Equação 10.5.....	60
$\Omega = \frac{T}{s.J + k_{va}}$	Equação 10.6.....	60
$\Theta = \frac{T}{s(s.8,425 + 12)}$	Equação 10.7.....	60
$G_C = 12 \cdot \left(\frac{s+2}{s+7} \right)$	Equação 10.8.....	64
$\Omega = \frac{T}{s.8,425 + 12}$	Equação 10.9.....	67
$G_C = 29,778 \cdot \frac{(s+0,9)}{s}$	Equação 10.10.....	69
$G_C = 4200 \cdot \frac{(s+2)}{(s+7)}$	Equação 10.11.....	69

$$V = \frac{F}{s \cdot 50,55 + 20}$$

Equação 10.12..... 70

$$G_c = 62 \cdot \frac{(s + 0,5)}{s}$$

Equação 10.13..... 71

1 INTRODUÇÃO

No constante esforço para a ampliação de seu domínio no mundo e no universo o homem cria novas tecnologias que permitem sua locomoção a ambientes cada vez mais longínquos e hostis, e a realização de trabalho sobre esses ambientes. Mas percebe-se que a presença humana não é definitiva nesse processo de conquista, mas sim a possibilidade de realização de um determinado trabalho requerido pelo próprio homem. Nesse contexto surgem os *Unmanned Vehicles* (UV) ou Veículos Não Tripulados.

Os UVs são veículos não tripulados construídos afim de que sejam capazes de realizar alguma tarefa, podendo ser teleoperados ou autônomos. Primeiramente os UVs foram utilizados com fins militares e em seguida tiveram suas aplicações expandidas para exploração do espaço, de regiões submarinas de alta profundidade, de áreas contaminadas por lixo nuclear e para instalação e inspeção de dispositivos nesses ambientes. Os UVs são divididos basicamente em três grupos conforme o tipo de ambiente no qual ele pode ser utilizado. O primeiro grupo é dos Veículos Aéreos Não Tripulados ou *Unmanned Aerial Vehicles* (UAV) como o *Pioneer* utilizado na Guerra do Golfo em operações de reconhecimento e espionagem. O segundo grupo é dos Veículos Terrestres Não Tripulados ou *Unmanned Ground Vehicles* (UGV) como o UGV *Sojourner* que foi enviado em missão de reconhecimento ao planeta Marte e foi desenvolvido pelo *Jet Propulsion Laboratory Mars Rover*. E o terceiro grupo é dos Veículos Submarinos Não Tripulados ou *Unmanned Underwater Vehicles* como o OTTER desenvolvido pela Universidade de Stanford e pelo *Monterey Bay Aquarium Research Institute*, o Ocean Voyager II desenvolvido pelo Departamento de Engenharia Oceânica da *Florida Atlantic University*, o Odyssey desenvolvido pelo *Laboratory at MIT Sea Grant*, o Kambara desenvolvido pelo Laboratório de Sistemas Robóticos da *Australian National University* e o PTEROA desenvolvido pelo Instituto de Ciência Industrial da Universidade de Tokyo. Esses UUVs foram concebidos para aplicações de pesquisa biológica submarina, para a instalação e inspeção de cabos submarinos e na exploração de petróleo *off-shore*.

Dentro dos UUVs existem os Veículos Operados Remotamente ou Remotely Operated Vehicles (ROV) e os Veículos Submarinos Autônomos ou Autonomous

Underwater Vehicles (AUV). Como o próprio nome indica, os ROV são operados remotamente por um ser humano e geralmente são ligados a um navio por um cabo umbilical que tem a função de fornecer energia ao ROV e estabelecer um canal de comunicação entre o ROV e o operador. Já os AUVs são autônomos, ou seja, eles são auto-suficientes para a realização de alguma missão previamente estabelecida. Seu controle é feito por microcomputadores embarcados e a fonte de energia é interna, geralmente provinda de baterias elétricas.

Reconhecidas as diversas aplicações possíveis para os Veículos Não Tripulados, encontra-se em desenvolvimento o projeto de um Veículo Submarino Não Tripulado no Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da Escola Politécnica da Universidade de São Paulo. Fazendo parte desse projeto, o presente trabalho busca desenvolver o software da Arquitetura de Controle do UUV. Entende-se por Arquitetura de Controle o conjunto de hardware e software necessário para execução, controle e coordenação de diversas funções como percepção do ambiente externo, planejamento e execução de ações, etc, que juntas são capazes de fazer com que o UUV realize uma missão pré-determinada.

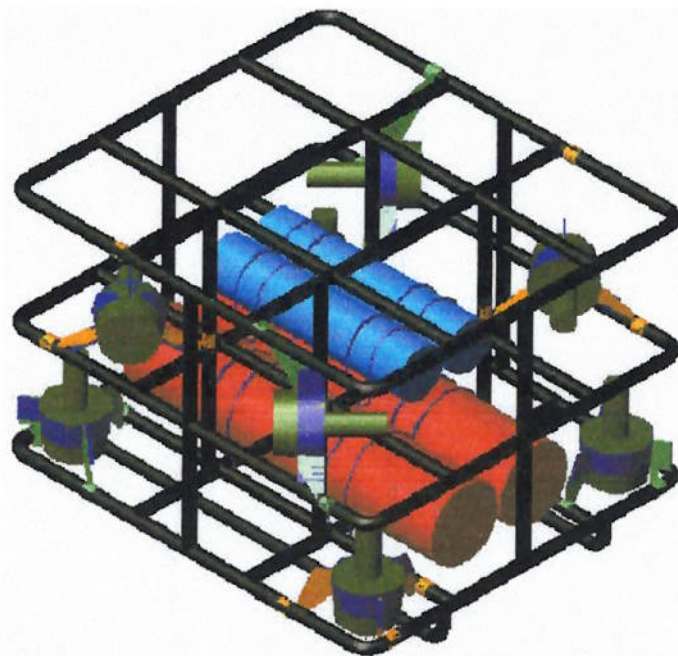


Figura 1.1 - Ilustração do UUV da Escola Politécnica

2 HISTÓRICO

O início exato dos experimentos com UUVs (*Unmanned Underwater Vehicle*) não pode ser bem definido mas grandes acontecimentos foram registrados desde os anos sessenta. Em 1966 o veículo CURV (*Cable Controlled Underwater Recovery Vehicle*) da marinha dos Estados Unidos foi utilizado para recuperação de uma bomba atômica perdida após um acidente em Palomares, Espanha. Esse mesmo veículo foi também a peça chave no resgate de pilotos de um submarino naufragado em Cork, Irlanda, em 1973.

Nos anos setenta houve o reconhecimento pelos governos de diversos países quanto a utilidade de desenvolvimento de veículos submarinos não tripulados e assim foram fundados diversos centros de como o JAMSTEC (*Japan Marine Science and Technology Center*) no Japão e a parceria franco-americana FAMOUS. Nesse período foram realizadas expedições em profundidades de 30 a 100 metros, mas sempre com o intuito de atingir maiores profundidades. Algumas sondas submarinas já estavam atingindo profundidades de 6000 metros.

O processo de desenvolvimento foi contínuo e nos anos oitenta percebeu-se a possível utilização de veículos submarinos não tripulados na exploração de petróleo. Dessa forma surgiram parcerias não governamentais e os veículos RCV-225 e RCV-150 projetados e construídos pela HydroProducts (Estados Unidos) foram os primeiros utilizados no trabalho *offshore*. Em 1984 foi fundado o instituto francês INFREMER (*Institut français de recherche pour l'exploitation de la mer*) e foi realizada a parceria franco-japonesa KAIKO. Em 1986 os veículos Alvin e Algo (Estados Unidos) foram capazes de atingir a profundidades de 3750 metros e fotografaram o navio naufragado Titanic. Três anos mais tarde esses mesmos veículos foram enviados em missão de exploração ao naufrágio do navio de guerra alemão Bismark numa profundidade de 4700 metros.

A exploração de regiões cada vez mais profunda do oceano levou em 1995 a missão do veículo KAIKO na fenda submarina de Mariana, ponto de maior profundidade do Globo terrestre. O mergulho de 10911,4 metros foi realizado com sucesso, estabelecendo o recorde máximo.



Figura 2.1 - Veículo KAIKO (franco-japonês) detentor do recorde de 10911,4 metros de profundidade

3 ATUALIDADE

Atualmente as pesquisas se desenvolvem no sentido de redução dos custos de construção e operação dos veículos submarinos não tripulados e aumento da confiabilidade no cumprimento de missões para que esses veículos sejam viáveis na utilização comum de instalação e manutenção de estruturas, cabos e dutos submersos, e na pesquisa geológica, arqueológica e da biomassa submarina.

3.1 Veículos

Os veículos submarinos não tripulados atuais podem ser divididos em categorias segundo o seu tamanho, profundidade de trabalho, potência despendida e tipo de atuação (totalmente elétrica ou eletro-hidráulica). As categorias estão resumidas na tabela abaixo.

Categoria	Capacidade	Profundidade	Potência Max.
Baixo Custo (elétrico)	Observação	Até 100m	5HP
Pequeno (elétrico)	Observação	Até 300m	10HP
Grande (elétrico)	Observação e trabalhos leves	Até 3000m	20HP
Ultra Profundo (elétrico)	Observação e coleta de dados	Maior que 3000m	25HP
Médio (eletro-hidráulico)	Trabalhos leves a médios	Até 2000m	100HP
Grande (eletro-hidráulico)	Trabalhos Pesados	Até 3000m	300HP
Ultra Profundo (eletro-hidráulico)	Trabalhos Pesados	Maior que 3000m	120HP

Tabela 3.1 - Classificação dos veículos submarinos

Os veículos de baixo custo surgiram nos anos oitenta com o ingresso de empresas privadas na construção de robôs submarinos e representam hoje 22% do

total. Seus preços giram na faixa dez mil a cem mil dólares. Alguns exemplos desses veículos são o Quest e o Hyball da Hydrovision.



Figura 3.1 – Veículo de baixo custo Quest



Figura 3.2 - Veículo de baixo custo Hyball

Veículos com maior capacidade e profundidade de trabalho são mais caros e possuem preços que giram em torno de quinhentos mil dólares como o Voyager da Perry Trittech e o MaxROVER da Deep Sea Systems International.



Figura 3.3 - Veículo Grande Voyager



Figura 3.4 - Veículo Grande MaxROVER

3.2 Aplicações

Dentro das expectativas capitalistas do mundo moderno, as aplicações mais freqüentes dos robôs submarinos são na exploração de gás e petróleo. Na Europa os UUV são utilizados principalmente pelos ingleses e noruegueses na exploração do Mar do Norte. As profundidades de trabalho são de aproximadamente 150 metros.

Na Ásia existe uma forte utilização de UUVs no estudo sísmico e procura de novas reservas de gás natural pela Mobil e pela Texaco em regiões oceânicas do Sul da China e Malásia, se estendendo até o Oeste da Austrália. As profundidades de navegação vão de 900 a 1600 metros.

A Rússia está iniciando seus planos de exploração da região Ártica em busca de reservas de gás natural em profundidades de 400 metros. Na África a Exxon prevê a utilização de UUVs em plataformas de petróleo na Nigéria a profundidade de 1474 metros e no Congo a profundidades de até 2000 metros.

E no Brasil a Petrobrás envia robôs submarinos em missões nas plataformas de exploração de petróleo na Bacia de Campos. A plataforma Marlim Sul possui o recorde em profundidade de extração de petróleo a 1747 metros e existem outras reservas já identificadas e com grande possibilidade de aplicação de UUVs a profundidades de até 2000 metros.

A tarefa mais freqüente dos UUVs é o auxílio na recuperação de equipamentos que são depositados no fundo do oceano. Esses equipamentos têm grande autonomia e ficam em operação durante vários anos. Após sua utilização eles são recuperados por navios com sistema de tração de cabos especiais que possuem um tipo de conexão particular para se acoplarem aos dispositivos submersos. Normalmente o UUV é comandado remotamente por um operador, ou seja, trata-se de um ROV (Remoted Operated Vehicle) e tem a tarefa de guiar os cabos até os dispositivos a serem recuperados.

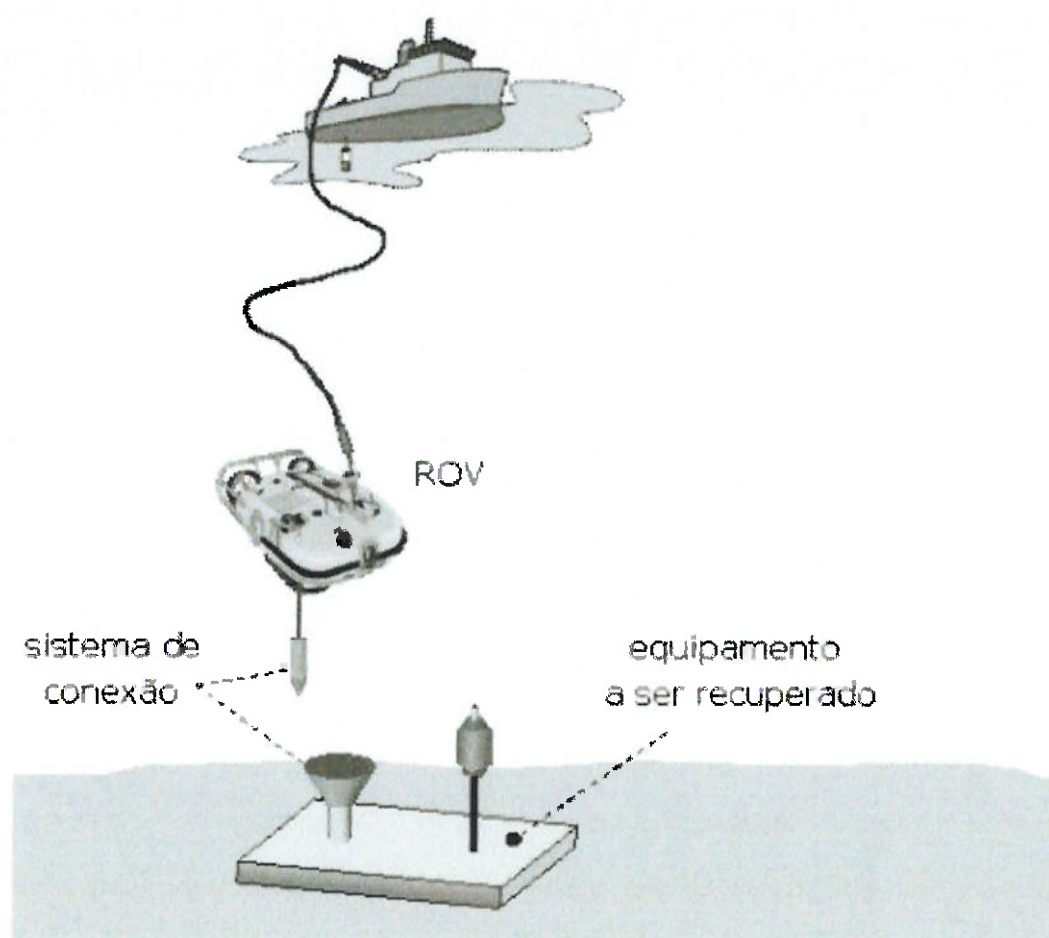


Figura 3.5 - Recuperação de Dispositivo com auxílio de um ROV

Deseja-se dar mais autonomia ao UUV de forma que ele seja capaz de fazer o encaixe do sistema de conexão de forma automática. Isso reduziria os custos devido à redução de tempo de operação e eliminaria a necessidade de pessoal com boas habilidades manuais na condução do robô.

4 ESTUDO DE VIABILIDADE DE VEÍCULOS SUBMARINOS NÃO TRIPULADOS E NECESSIDADE DE ARQUITETURAS DE CONTROLE

A análise anterior dos acontecimentos relativos ao desenvolvimento de UUVs mostra a crescente utilização desses veículos nos mais variados campos de exploração subaquática e evidencia seu principal uso na prospecção e exploração de gás natural e petróleo. A indústria petrolífera brasileira é referência mundial na exploração de petróleo em águas profundas e a cada dia novas reservas petrolíferas são descobertas. Para a instalação e manutenção do maquinário de extração de petróleo é necessário um grande esforço conjunto de mergulhadores e comandantes de barcos de suporte e de manobra. Exige-se bastante precisão de operadores em condições de alto risco. Dessa forma é natural esperar que essas atividades passem a ser desenvolvidas de modo mais seguro e confiável através da utilização de robôs. Assim, não restam dúvidas quanto à necessidade do desenvolvimento de veículos submarinos não tripulados.

Economicamente as missões tradicionais tomam bastante tempo e são bastante caras. Por outro lado, os equipamentos necessários para a construção de robôs submarinos estão cada vez mais baratos e acessíveis. Resta analisar se um robô é realmente capaz de superar as adversidades do ambiente e de alcançar seus objetivos de forma eficaz. Isso depende basicamente de dois fatores: a disponibilidade de componentes mecânicos suficientemente eficientes como fortes propulsores, sensores de alta precisão e, principalmente, da lógica que governa todo o funcionamento de cada dispositivo, o Software da Arquitetura de Controle.

4.1 Características do Ambiente e Forças Envolvidas

Uma grande dificuldade na utilização de robôs submarinos vem da complexidade dos ambientes subaquáticos, e impossibilidade de modelar todas as suas características como correntes marítimas, ondas e disposição topográfica do fundo do mar.

A título de exemplo a fórmula abaixo mostra uma possível modelagem da velocidade de correntes marítimas:

$$U = U_t + U_w + U_s + U_m + U_{\text{set-up}} + U_d$$

onde U_t : componente devido às ondas

U_w : componente devido à ação do vento local

U_s : componente de corrente de Stokes

U_m : componente devida a circulação de água na região conforme a geografia local (golfo, baía, estreito, ...)

$U_{\text{set-up}}$: componente devida a fenômenos como tempestades

U_d : componente devida diferença de densidade em diferentes porções do oceano

Percebe-se que o projeto deve ser capaz de eliminar essas perturbações sem a modelagem específica de cada componente. De uma forma geral é necessário que o veículo seja capaz de vencer a soma dessas componentes. Projetos de estruturas *offshore* no mar do Norte consideram 1 m/s como velocidade típica. Algumas regiões costeiras como dos Fiordes na Noruega possuem correntes que atingem 3m/s. Felizmente as correntes marítimas em regiões tropicais e equatoriais do Globo Terrestre tendem a ser menores e no presente trabalho será utilizado o valor nominal de 0,5 m/s.

As correntes marítimas são as responsáveis por duas das principais forças atuantes no robô: o arrasto do veículo propriamente dito e o arrasto do cabo umbilical.

Outras forças atuantes sobre o veículo são a força peso e o empuxo. Essas forças são menos preocupantes pois o empuxo pode ser regulado de forma a praticamente anular a força peso.

4.2 Verificação da capacidade de atuadores e sensores

É comum a utilização de propulsores com motores elétricos para a movimentação de robôs submarinos. Esses propulsores podem fornecer um empuxo de 200N sem muitas dificuldades, o que garante a superação da força de arrasto.

Como sensores, pode-se utilizar um sistema de localização acústica que pode dar a posição do veículo em relação a um alvo com uma precisão de 5mm, quando se encontram a uma distância de 1m. O alcance do sistema pode chegar a 50m. Quanto a orientação do robô, estão disponíveis no mercado bússolas e inclinômetros com precisão da ordem de $0,3^\circ$ e $0,28^\circ$ respectivamente, que é suficiente.

Dessa forma é constatado que os dispositivos de atuação e sensoramento disponíveis são suficientes para aplicação. Assim, resta apenas que eles sejam corretamente utilizados, conforme o software de controle.

4.3 Necessidade de uma Arquitetura de Controle

A fim de dar a funcionalidade ao UUV é preciso formular uma lógica de controle, que utiliza as informações dos sensores e exige um certo comportamento dos atuadores, e implementá-la. Nota-se que essa lógica não é simples pois o UUV estará exposto a um ambiente parcialmente conhecido, dinâmico, e os dispositivos estão sujeitos a falhas. Dessa forma fica claro que o desenvolvimento do software que será capaz de dar funcionalidade ao UUV deve ser muito bem estruturado para que sejam possíveis o entendimento minucioso e o controle de todos os possíveis estados do sistema, ou seja, necessita-se de uma arquitetura de controle.

5 OBJETIVO E ESPECIFICAÇÃO DOS REQUISITOS

O objetivo deste trabalho é projetar e implementar uma Arquitetura de Controle de um veículo submarino para observação e realização de trabalhos leves como auxílio na recuperação de equipamentos de extração de petróleo, a profundidades de até 2000 metros.

Assim, necessita-se que o veículo seja capaz de se mover com velocidades de até 1m/s, tenha autonomia de 20 minutos no local de trabalho mais 40 minutos para locomoção geral e seja suficientemente manobrável com precisão da ordem de centímetros. Deve ser disponível a escolha entre dois modos movimentação do robô: um automático e um manual. No modo manual o operador deve conduzir o veículo com o auxílio de um *joystick* e no modo automático o robô deve ser capaz de seguir uma trajetória pré-determinada. Deve estar em funcionamento um sistema de previsão e gerenciamento de situações anormais de operação para que o sistema seja robusto e estável. A lógica de controle e sua implementação são os focos principais do trabalho.

6 DESCRIÇÃO DO PROJETO EM DESENVOLVIMENTO NA ESCOLA POLITÉCNICA

O veículo submarino que está sendo desenvolvido na Escola Politécnica se trata de um UUV híbrido pois existe a intenção que ele funcione tanto em modo manual, como um ROV, com comandos dados por um operador através de um *joystick*, ou em modo autônomo, como um AUV (*Autonomous Underwater Vehicle*), em que o veículo procuraria seguir uma trajetória pré-determinada.

A estrutura do robô é aberta, de tubos de alumínio, onde vão instalados os dispositivos de atuação e vasos de pressão em aço inox que acomodam os dispositivos de sensoriamento e de realização de controle, e o jogo de baterias.

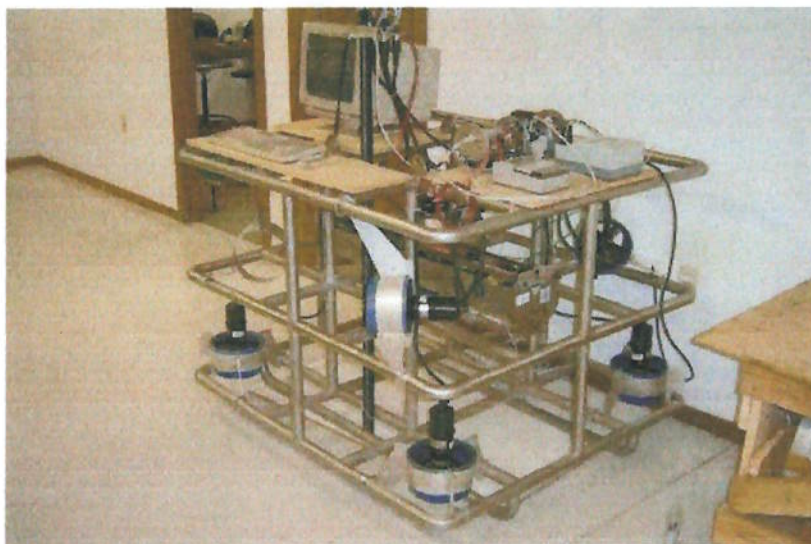


Figura 6.1 - Montagem Experimental com estrutura do veículo, eletrônica a ser embarcada, baterias e propulsores

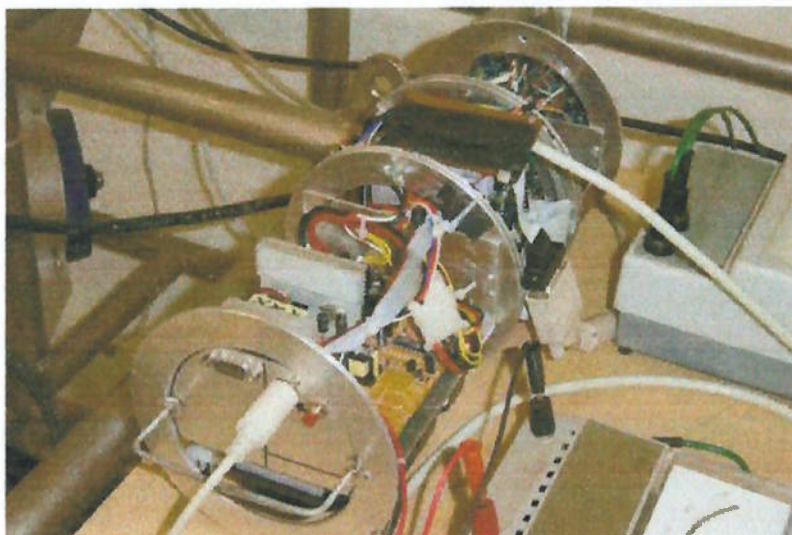


Figura 6.2 - Detalhe da eletrônica embarcada (PC104, sensores, etc)

6.1 Estrutura Tubular e Vasos de Pressão

Como dito anteriormente a estrutura do AUV é tubular e aberta e o projeto final prevê que a estrutura seja de alumínio. Temporariamente a estrutura disponível é feita em aço inox e esta é suficiente para visualização da disposição dos dispositivos na estrutura final do robô.

São quatro os vasos de pressão disponíveis, dois menores para abrigar a eletrônica a ser embarcada e dois maiores para carregar as baterias.

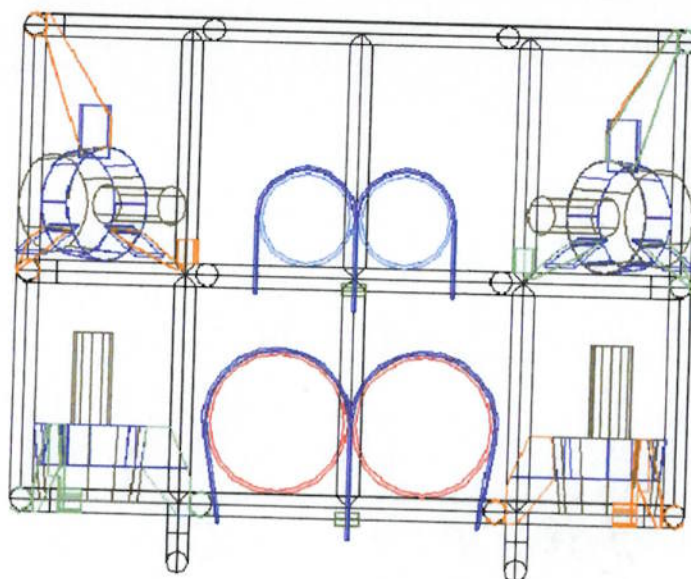


Figura 6.3 - Vista Frontal do UUV

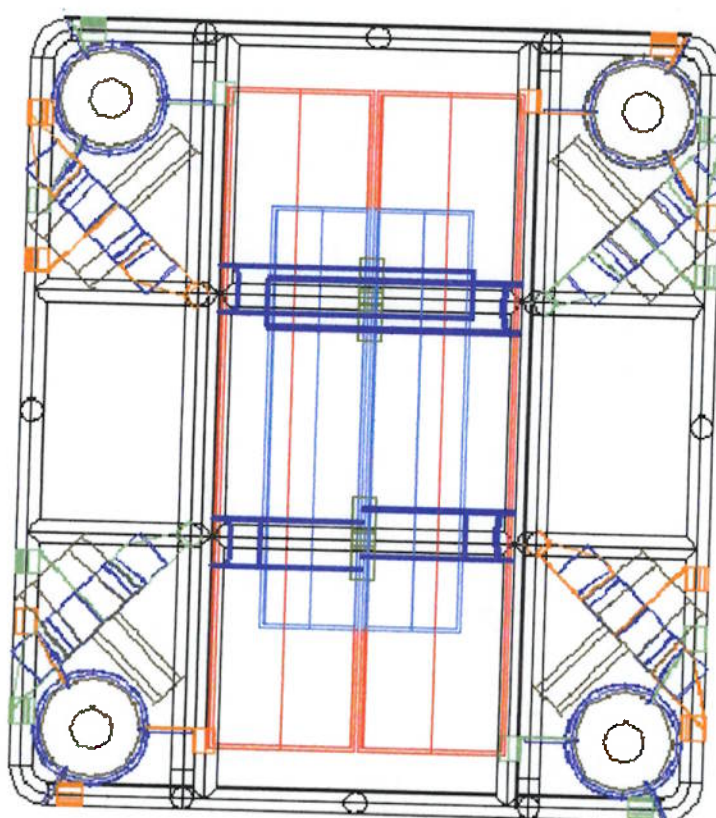


Figura 6.4 - Vista Superior do UUV

Nota-se que os vasos de pressão com as baterias são mais pesados e foram dispostos na parte inferior da estrutura para que o centro de gravidade ficasse abaixo do centro de flutuação, levando a estabilidade do veículo. Seguem abaixo as características físicas e mecânicas da estrutura prevista em alumínio e dos vasos de pressão:

Estrutura

- Dimensões da Estrutura:
 - Diâmetro dos tubos: 38mm
 - Comprimento: 1316mm
 - Largura: 1150mm
 - Altura: 838mm
- Massa: 24,15 Kg

Vasos de Pressão

- Dimensões dos vasos de pressão para eletrônica:
 - Diâmetro: 164mm
 - Comprimento: 700mm
- Dimensões dos vasos de pressão para baterias:
 - Diâmetro: 244mm
 - Comprimento: 1100mm

6.2 Dispositivos de Atuação

Os dispositivos a serem utilizados na movimentação do veículo são oito propulsores de hélice movidos a motores elétricos brushless fornecidos pela Tecnadyne. A controle de propulsão é dado por um sinal variando de -5 a 5 volts,

sendo que o sinal indica o desejado sentido de rotação. Esses propulsores possuem um tacômetro que mede a rotação do rotor em módulo, ou seja, não há diferença na leitura do tacômetro em relação ao sentido de rotação. Seguem abaixo suas principais características físicas, elétricas e mecânicas:



Figura 6.5 - Propulsor

Propulsores

- Dimensões:
 - diâmetro da hélice: 140mm
 - diâmetro máximo externo: 180mm
 - comprimento: 250mm
- Potência Máxima: 1HP = 745 W
- Voltagem: 150 V
- Corrente na Potência Máxima: aprox. 5A
- Empuxo Máximo para frente: 200N
- Empuxo Máximo para trás: 120N

6.3 Dispositivos de Sensoriamento

Para que seja possível fechar uma malha de controle, ou mesmo obter informações do estado do veículo para monitoração e controle manual, estão disponíveis uma bússola, um par de inclinômetros e está em projeto um sensor de posicionamento acústico.

A bússola, modelo TCM2-50 do fabricante PNI Corporation, é composta por três sensores de campo magnético posicionados em três eixos perpendiculares entre si, de um par de inclinômetros, e de um sensor de temperatura. Os inclinômetros da bússola são utilizados para compensar as leituras dos sensores magnéticos caso ela não esteja num plano horizontal. De maneira correspondente o sensor de temperatura está presente para que sejam possíveis compensações por variação de temperatura do dispositivo. O sinal de saída da bússola pode ser lido tanto por uma porta serial RS232C como por uma saída analógica (disponível apenas para compatibilidade com instrumentos exclusivamente analógicos).



Figura 6.6 - Bússola

Bússola

- Faixa de leitura: - 50 a 50°

- Resolução: $0,1^\circ$
- Repetibilidade: $0,3^\circ$
- Voltagem: 5V
- Amperagem: 5 a 20 mA
- Temperatura de trabalho: -20 a 70°C
- Saída Digital
 - tipo serial RS232C
 - frequência de aquisição: 1 a 30 Hz
- Saída Analógica
 - 0 a 5V, resolução de 19,53 mV (256 níveis)

Os inclinômetros são da série 758 da Applied Geomechanics. O dispositivo possui também um sensor de temperatura e é necessária uma placa de condicionamento de sinais para que se possa fazer a leitura dos sensores. A placa modelo 83162 é produzida especialmente para esse fim.

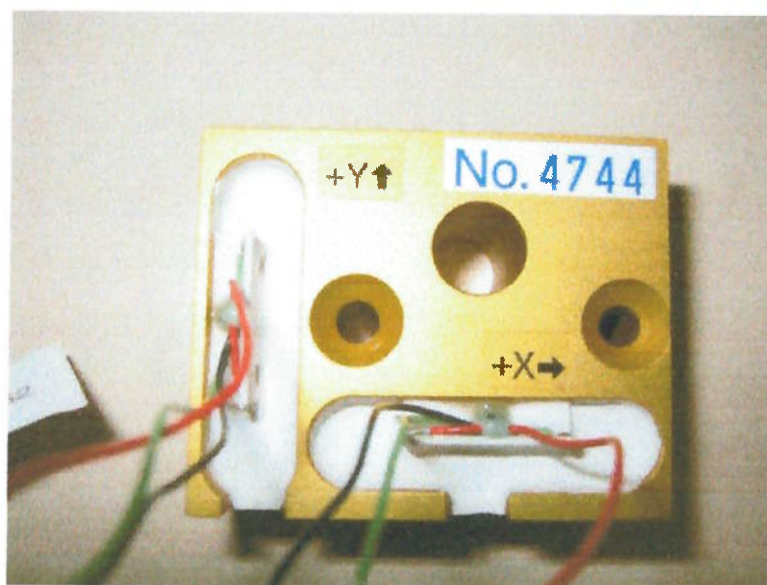


Figura 6.7 - Inclinômetros

Inclinômetros

- Faixa de leitura: -60 a 60°
- Resolução: aprox. $5\ \mu\text{Rad} = 0,28^\circ$

- Repetibilidade: $5 \mu\text{Rad} = 0,28^\circ$
- Voltagem: 11 a 15 V
- Amperagem: 12 mA
- Temperatura de Trabalho: -30 a 70°C
- Sinal de saída: -8 a 8VDC

As informações fornecidas pela bússola em conjunto com as informações fornecidas pelos inclinômetros fazem com que a orientação do veículo esteja determinada. Para a determinação da posição será utilizado um sistema acústico que está em desenvolvimento. Esse sistema é composto de duas partes, uma embarcada no veículo e outra fixada no alvo a ser atingido. Suas características principais são:

Sistema de Localização Acústica

- Alcance: 50 m
- Precisão Máxima: 5 mm (a uma distância de 1m do alvo)
- Voltagem: 12 V
- Amperagem: aprox. 20mA (estimado)

6.4 Dispositivo de Realização do Controle

Como dispositivo de realização do controle será utilizado um microcomputador do tipo PC104, modelo CMR6686GX fornecido pela Real Time Devices. Os PC104 obedecem ao padrão PC/AT e foram especialmente projetados para aplicações onde o espaço físico disponível é limitado, bem como a fonte de energia. Dessa forma, apresentam dimensões reduzidas e baixo consumo, além de ter boa confiabilidade e baixa sensibilidade a ruídos. Outras características estão relacionadas abaixo:

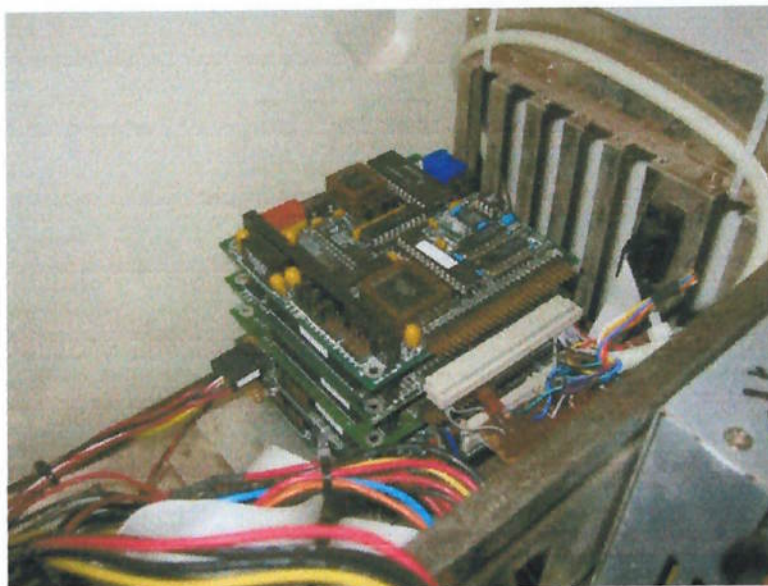


Figura 6.8 - PC104

Microcomputador PC104

- Processador Geode GXLV MMX National Semiconductor operando a 233MHz
- Processador Matemático
- 64 Mb de memória RAM
- Disco de estado sólido DiskOnChip 2000 de 32Mb
- Barramento para módulos de extensão de 8 e 16 bits
- Duas portas de comunicação serial configuráveis para RS232/422/485
- Porta paralela com SPP, ECP ou EPP
- Consumo: 7,5W
- Dimensões: 95 x 100 mm
- Temperatura de operação: -40 a 85° C

6.5 Conjunto de Baterias

Toda a energia requerida para o funcionamento dos dispositivos do UUVs será fornecida por dois conjuntos de doze baterias Chumbo-Ácido. Cada bateria disponibiliza 12V e para ser possível alcançar os 150V necessários para o funcionamento dos propulsores as doze baterias de cada conjunto serão montadas em série. A vantagem dessas baterias é o baixo custo frente a outros tipos oferecidos no mercado, por exemplo as de tecnologia Níquel-Cádmio e a Níquel-Metal-Hidreto. Sua grande desvantagem é o grande peso e a baixa relação carga disponível por massa. Resumindo, tem-se as seguintes características:

Baterias

- Tecnologia: Chumbo-Ácido
- Unidade
 - Voltagem: 12V
 - Capacidade de Carga: 12 Ah
 - Tempo de recarga: 12h
- Conjunto de 12 Baterias em Série
 - Voltagem: 144V
 - Capacidade de Carga: 12Ah
- Dois Conjuntos ligados em paralelo
 - Voltagem: 144V
 - Capacidade de Carga: 24 Ah

6.6 Dispositivos de Comunicação com o Meio Externo e Cabo Umbilical

A comunicação com o meio externo será realizada através de um canal serial operando no padrão RS485. Esse padrão é necessário por trabalhar com níveis de

corrente no envio de informações, e não com níveis de tensão, o que permite alcançar maiores distâncias de comunicação, chegando aos necessários 2000 metros.

O meio físico por onde as informações serão transmitidas será o cabo umbilical. Ainda está em estudo qual será o cabo adotado pois é necessário que o cabo obedeça diversos requisitos como resistência mecânica, flexibilidade, pequeno diâmetro e isolamento de ruídos elétricos. Para isso existem diferentes modelos disponíveis no mercado que utilizam variados materiais de reforço estrutural como fios de aço galvanizados e fios de KEVLAR. Naturalmente o primeiro apresenta a vantagem de ser mais barato mas as desvantagens de ter uma flexibilidade reduzida e o risco de deterioração por corrosão, enquanto que o segundo tem boa relação resistência por densidade e flexibilidade mas é significativamente mais caro. Sendo um elemento de vital importância para o funcionamento do sistema, o cabo umbilical deve ser cuidadosamente estudado e escolhido.

7 FUNCIONAMENTO E COMPORTAMENTO ESPERADOS DO VEÍCULO

Os objetivos especificados no capítulo 5 definem de forma bastante sucinta o que é esperado do funcionamento geral do veículo. A seguir serão detalhadas as etapas de funcionamento e a interação do operador com o veículo.

7.1 Modos de Funcionamento

7.1.1 Inicialização e Modo Inativo

Quando ligado espera-se que primeiramente o veículo seja capaz de preparar seus componentes para o uso. Isso envolve o carregamento do software da arquitetura de controle com seus parâmetros corretamente ajustados e a ativação da bússola, que é o único dispositivo passível de programação. Enquanto o veículo está realizando essas tarefas, diz-se que está em modo de inicialização. Após ter sido concluído o processo de inicialização o veículo entra em modo inativo.

O modo inativo é característico pela total passividade do veículo que apenas monitora seus dispositivos de sensoriamento e não aciona nenhum propulsor. Esse modo é tanto um modo de partida (alcançado logo após a inicialização) como um modo de preparo para desligamento. Serve então para que o veículo possa ser lançado ao mar em missão e ser recuperado a bordo da embarcação de apoio, após o término da missão.

7.1.2 Modo Manual

Nesse modo espera-se que o veículo responda ativamente a comandos dados pelo operador através de um *joystick*. A manipulação do *joystick* deve ser

interpretada como uma ação específica por parte do veículo como “andar para frente” ou “girar em sentido horário” e traduzida ao acionamento dos propulsores necessários para que a ação requisitada seja atingida. Maiores detalhes sobre cada tipo de ação estão dados na descrição da interface operador/veículo.

7.1.3 Modo de *Hovering*

No modo de *hovering* (do inglês, pairar) o veículo deve ser capaz de se manter estacionário num certo ponto, com inclinação zero. Trata-se, portanto, de um modo ativo, pois o veículo deve acionar os propulsores de forma a compensar os esforços da resultante da força peso e do empuxo, além das perturbações externas. Este modo será usado como estado de partida para o modo automático, bem como pós-estado do modo automático.

7.1.4 Modo Automático

Nesse modo o veículo deve responder de forma a acompanhar uma trajetória pré-determinada. Não há portanto intervenção do operador, que pode apenas monitorar a situação do veículo em missão. Esse modo deve ser utilizado em ações específicas como ida ao encontro de equipamento submerso a ser recuperado e, principalmente, manobra para o encaixe da conexão de recuperação.

7.1.5 Modo de Teste

Este modo, como o nome indica, deve ser utilizado para testes como a verificação do funcionamento dos propulsores. Diferentemente do modo manual, em que os comandos dados pelo *joystick* são interpretados como ações de alto nível, no modo de teste cada propulsor pode ser acionado individualmente. Não oferece, assim, meio eficaz para a condução do veículo.

7.2 Alternância entre Modos

Como já especificado anteriormente, quando ligado o veículo passa pelo modo de inicialização e em seguida vai para o modo inativo. O modo inativo é um modo de base para a mudança para o modo manual. A mudança entre modos deve ser feita conforme a solicitação do usuário e a real possibilidade de mudança de modo, conforme a verificação do bom funcionamento dos dispositivos. Para a mudança do modo inativo para o modo manual deve-se verificar o correto funcionamento dos sensores e dos propulsores. Para atingir o modo automático deve-se passar pelo modo de *hovering* e dentro desse último modo deve ser verificada a correta especificação da trajetória a ser acompanhada. A alternância para o modo de testes pode ser efetuada por requisição do operador e condição de que o modo de partida seja o inativo.

Sempre que for detectada a impossibilidade de funcionamento do veículo em modo manual, de *hovering* ou automático, seja por deficiência de funcionamento dos propulsores, seja por falha na leitura dos sensores, o estado do veículo deve ser alternado automaticamente para o modo inativo, a menos que o modo atual seja o de teste. As possíveis alternâncias entre modos estão resumidas na tabela abaixo:

		Modo de origem				
		Inativo	Manual	Hovering	Automático	Teste
Modo de chegada	Inativo	-	● ●	●	●	●
	Manual	●	-	●	-	-
	Hovering	-	●	-	●	-
	Automático	-	-	●	-	-
	Teste	●	-	-	-	-
Legenda: ● Mudança com requisição ● Mudança com requisição e verificação de condições ● Mudança automática por detecção de falha						

Tabela 7.1 - Alternância entre modos de funcionamento

7.3 Interface com o Operador

A interface com o operador deve permitir a monitoração da posição e orientação do veículo, do modo de operação atual e do estado dos dispositivos como propulsores e sensores e o envio de sinais de comando como mudança de modo de operação e comandos de movimentação.

7.3.1 Monitoração

A monitoração do veículo apresenta desafios para ser desenvolvida devido à dificuldade de se representar um objeto de natureza 3D (seis graus de liberdade, três de translação e três de rotação) num dispositivo plano, a tela de um monitor. Existem projetos como o do veículo submarino australiano Kambara que procuram utilizar os avanços da computação gráfica para representar o veículo numa só imagem. Mas a visualização dos estados do veículo nem sempre é fácil num sistema como esse e devido à existência de instrumentos de leitura já consagrados na navegação de outros veículos, como aviões e embarcações, será dada preferência a esses últimos. Assim, planeja-se utilizar um visor na forma de bússola para a visualização da orientação do veículo e mostrador do tipo horizonte artificial para a visualização de suas inclinações. A posição do veículo será dada por um plano XY, como uma vista superior da cena, e sua profundidade por um indicador gráfico de barras.

Serão também utilizados diversos indicadores tipo LED para a visualização do modo de operação do veículo e do estado de seus dispositivos.

7.3.2 Comando

Os comandos a serem enviados ao veículo são as requisições de mudança de modo de operação, as ações definidas pelo joystick, a determinação da trajetória a ser seguida (em modo automático) e, eventualmente, alguns parâmetros de configuração. As requisições de mudança de modo deverão ser feitas através de *push-buttons* e a

determinação da trajetória deverá ser realizada através de caixas de diálogo e leitura de arquivos.

O *joystick* disponível é um *joystick* Microsoft Force FeedBack que apresenta diversos eixos e botões, suficientes para o total controle do veículo. O *manche* do joystick permite os movimentos para frente/trás, esquerda/direita, e torção horária/anti-horária. Existe também um eixo auxiliar (na base do joystick), um pad circular com oito posições de comando e oito botões para uso geral. As relações entre o movimento do joystick e o respectivo comando de ação estão descritas na tabela seguinte.

Movimento do Joystick	Comando de ação
Manche para frente	Navegar para frente
Manche para trás	Navegar para trás
Manche para esquerda	Navegar para a esquerda (translação)
Manche para a direita	Navegar para a direita (translação)
Torção anti-horária do manche	Girar no sentido anti-horário
Torção horária do manche	Girar no sentido horário
Botão 1 (gatilho)	Navegar para baixo
Botão 2	Navegar para cima
Pressionamento superior do Pad	Inclinar veículo, frente para baixo
Pressionamento inferior do Pad	Inclinar veículo, frente para cima
Pressionamento esquerdo do Pad	Inclinar veículo, lado esquerdo para baixo
Pressionamento direito do Pad	Inclinar veículo, lado direito para baixo
Eixo auxiliar	Controle de intensidade de ação

Tabela 7.2 - Relações entre o movimento do joystick e o respectivo comando de ação

O que se deseja com cada comando de ação é uma resposta em velocidade do veículo. Manter o *manche* empurrado para frente, por exemplo, deve equivaler a um

desejo de movimentação para frente do veículo, com velocidade constante, proporcional ao ângulo do *manche*. Como os botões e o *pad* não apresentam sensibilidade, isto é, estão ou ligados ou desligados, as ações de comando requisitadas por eles vão ser proporcionais à posição do eixo auxiliar. Dessa forma, o eixo auxiliar funciona como um ganho sobre os sinais de ação. Mesmo os sinais dados pelo *manche*, que possuem sensibilidade, também serão afetados pelo eixo auxiliar, funcionando como limitadores.

8 ESTUDO E PROJETO DE ARQUITETURA DE CONTROLE

8.1 Características desejadas numa Arquitetura de Controle

A necessidade de uma Arquitetura de Controle foi introduzida a partir da razoável complexidade de funcionamento de um UUV básico. Mas estendendo as funcionalidades da Arquitetura de Controle, são levantadas as seguintes características desejáveis em uma Arquitetura de Controle (FERREIRA, 2003):

- **Extensibilidade:** possibilidade de inclusão de novos dispositivos (sensores, atuadores, etc) sem a necessidade de mudança significativa da estrutura já existente
- **Generalidade:** possibilidade de troca de dispositivos por equivalentes ou transferência da estrutura já existente para um outro UUV sem grandes empecilhos
- **Previsibilidade:** o comportamento do sistema deve ser previsível com base no objetivo da missão e do histórico dos acontecimentos (leitura de sensores e comando para atuadores)
- **Capacidade de Modelagem do Ambiente:** reconhecimento e informações sobre o ambiente suficientes para sua modelagem a fim de ser utilizada no planejamento de trajetórias
- **Reatividade:** as reações do sistema (basicamente acionamento de propulsores) devem ser significativas o suficiente (em intensidade e em tempo de resposta) para permitir a realização da missão pré-determinada
- **Capacidade de Detecção e Recuperação de Erros:** procura verificar o bom funcionamento dos dispositivos de forma que leituras inconsistentes de sensores ou mau funcionamento de atuadores ou outros dispositivos não comprometam a missão (relacionado com capacidade de tolerância a falhas)

- **Capacidade de Tolerância a Falhas:** caso uma falha seja detectada, deve-se na medida do possível não utilizar funcionalidades que sejam relativas à falha, ou seja, procura-se identificar os dispositivos falhos e, de acordo com o nível de redundância, designar outros dispositivos a fazerem as tarefas realizadas anteriormente pelos dispositivos falhos
- **Programabilidade:** facilidade de especificação e verificação de missões, dada pela interface gráfica com o operador.

As características listadas acima podem ser separadas em dois grupos: características de desenvolvimento (as duas primeiras) e características de operação (as demais).

8.2 Arquitetura de Controle Proposta

Para que os objetivos de controle possam ser atingidos procura-se seguir os conceitos genéricos anteriormente apresentados sobre arquitetura de controle e aplicá-los no desenvolvimento de um software capaz de dar funcionalidade aos dispositivos do sistema. Assim, encontra-se uma possível forma de organização lógica, que pode ser observada na figura abaixo:

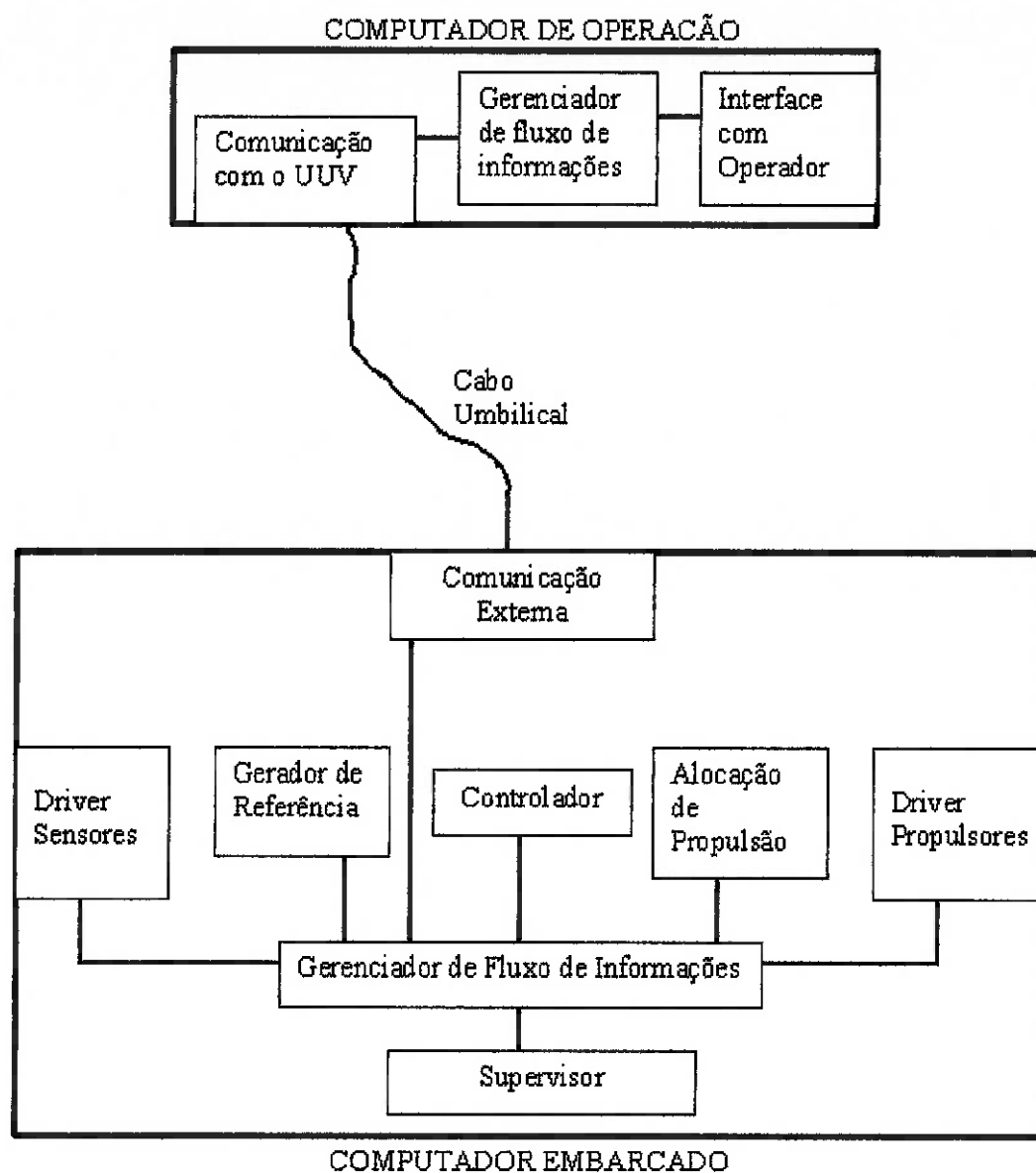


Figura 8.1 – Diagrama lógico do software da arquitetura de controle

Primeiramente percebe-se na figura acima que existe uma clara divisão do sistema entre o veículo submarino e a embarcação de apoio/operação. Na base de apoio é necessário um software que contenha um módulo de comunicação com o veículo, um módulo de gerenciamento do fluxo de informações e um módulo de interface com o operador. O módulo de interface deve possuir um sub-módulo *driver* do *joystick*.

Embarcados no veículo são necessários um módulo de comunicação com o meio externo, um módulo *driver* dos sensores, um módulo *driver* dos atuadores, um módulo de geração de referência, um módulo controlador, um módulo de alocação de propulsão, um módulo gerenciador do fluxo de informações e um módulo supervisor.

Os módulos de comunicação externa e de *drivers* de sensores e atuadores são dependentes do tipo de hardware instalado, que no caso são basicamente portas seriais e placas de aquisição de dados. A robustez exigida do módulo controlador para que o veículo se comporte de maneira previsível leva ao aumento da complexidade desse módulo, de forma que fica necessário um estudo específico para seu desenvolvimento e o assunto não será tratado nesse trabalho. Quanto à implementação, importa-se sobre tudo com os módulos de gerenciamento de fluxo de informações, de supervisão, de alocação de propulsão e de geração de referência.

8.3 Módulo Gerenciador do Fluxo de Informações

Esse módulo é responsável pelas solicitações de leitura de dados dos sensores, e do módulo de comunicação externa bem como o envio dessas informações para cada módulo que delas necessita. As linhas de fluxo de informações básicas são o envio de sinais de referência do gerador de referência e de leitura de sensores para o controlador, o envio de sinais de força e momento desejados sobre veículo entre o controlador e o alocador de propulsão e, finalmente, o envio de sinais de atuação entre o alocador de propulsão e os *drivers* dos motores. No funcionamento em modo manual de movimentação os sinais de referência têm origem no módulo de comunicação externa e são tratados no módulo gerador de referência, o que leva a um forte fluxo de informações entre esses módulos. Em qualquer modo de operação todos os sinais relevantes são enviados ao módulo supervisor para que seja possível a detecção de condições anormais de funcionamento.

Ressalta-se que esse módulo não é apenas uma ponte de comunicação entre os outros módulos, mas algo que, além disso, deve se preocupar com a natureza discreta do controle envolvido, as constantes de tempo do sistema e as limitações de

velocidade de troca de informações, particularmente entre o veículo e a embarcação de apoio.

Os sinais de entrada e saída do módulo gerenciador do fluxo de informações estão detalhados na tabela abaixo.

Sinais de entrada e saída do módulo gerenciador de fluxo de informações			
Entrada / Saída	Módulo Relacionado	Sinal	Tipo
Entrada	Driver dos sensores	Leitura de posição e orientação	Matriz [6x1], ponto flutuante
		Estado da bússola	Booleano
	Gerador de Referência	Referência de posição e velocidade (linear/angular)	Matriz [12x1], ponto flutuante
	Comunicação	Requisição de Mudança de Modo	Inteiro
		Comando do Joystick	Matriz[7x1], ponto flutuante
		Parâmetros de Trajetória	Matriz[2x1]
	Controlador	Forças e Torques necessários	Matriz [6x1], ponto flutuante
	Supervisor	Modo de Operação	Inteiro
		Estado do Veículo	Inteiro
	Alocador de Propulsão	Atuação necessária de cada propulsor	Matriz [8x1], ponto flutuante
Saída	Driver dos propulsores	Estado dos propulsores	Booleano [8x1]
	Driver dos sensores	Configuração da bússola	Inteiro
		Modo de operação	Inteiro
		Comando do Joystick	Matriz[7x1], ponto flutuante
		Parâmetros de Trajetória	Matriz[2x1]

	Comunicação	Modo de Operação	Inteiro
		Leitura de posição e orientação	Matriz [6x1], ponto flutuante
		Estado da bússola	Booleano
		Estado dos propulsores	Booleano [8x1]
	Controlador	Referência de posição e velocidade (linear/angular)	Matriz [12x1], ponto flutuante
		Leitura de posição e orientação	Matriz [6x1], ponto flutuante
	Supervisor	Requisição de Mudança de Modo	Inteiro
		Estado dos propulsores	Booleano [8x1]
		Estado da bússola	Booleano
	Alocador de Propulsão	Forças e Torques necessários	Matriz [6x1], ponto flutuante
	Driver dos propulsores	Atuação necessária de cada propulsor	Matriz [8x1], ponto flutuante

Tabela 8.1 - Sinais de entrada e saída do módulo gerenciador do fluxo de informações

8.4 Módulo Supervisor

O módulo supervisor é responsável gerenciar o estado veículo (modo de operação, estado dos sensores, estados dos atuadores, etc) e a detectar e contornar condições anormais de operação. Por condições anormais de operação entende-se falhas de dispositivos, choque inesperado do veículo, não cumprimento do sinal de trajetória especificado e falta de energia elétrica.

Algumas dessas anomalias podem ser facilmente detectadas enquanto que outras exigem uma análise dinâmica de variáveis associadas. Os propulsores

possuem tacômetros próprios que poderiam indicar quase diretamente a incoerência entre um sinal de atuação e a respectiva resposta do propulsor. Mas não se pode tão facilmente identificar uma falha nos dispositivos de sensoramento, por exemplo.

Após a detecção da falha parte-se para o passo de seu contorno e superação. Novamente existem situações que podem ser contornadas sem grandes dificuldades e outras em que passa a ser exigido um certo grau de inteligência de tomada de decisão. Uma falha de um propulsor, por exemplo, poderia ser facilmente superada com a realocação da propulsão sobre os demais atuadores, uma vez que o número de atuadores é redundante em relação ao número de graus de liberdade do sistema. A seguir procura-se levantar alguns problemas e suas possíveis detecções e contorno.

- Falha de propulsor
 - identificação: incoerência entre sinal de atuação e velocidade de rotação
 - contorno: realocação de propulsão
- Baixa reserva de energia
 - identificação: baixa voltagem disponível pelas baterias
 - contorno: caso o veículo tenha flutuabilidade positiva, garantir que ele não esteja vinculado com o meio e que não haja barreiras na sua subida. Para flutuabilidade negativa, voltar à superfície quando o nível de carga é apenas o suficiente para vencer a profundidade atual.
- Leituras não confiáveis da bússola
 - identificação: sinal de erro gerado pela própria bússola
 - contorno: manter o veículo em estado de espera por alguns instantes aguardando retorno da bússola. Caso o sinal de erro persista fechar a malha de controle apenas com os inclinômetros e dar referências para que o veículo volte à superfície.
- Choque inesperado do veículo
 - identificação: mudança brusca no sinal de posição do veículo no tempo

- contorno: procurar retornar a uma distância segura em relação ao local do choque seguindo o caminho de ida e gerar nova trajetória, contornado esse ponto

8.5 Módulo de Alocação de Propulsão

Dadas as referências desejadas de posição e orientação do veículo e os valores reais em que essas variáveis se encontram o controlador cumpre a tarefa de fornecer a força e o momento a serem realizados pelo conjunto de propulsores. Como o conjunto de propulsores apresenta redundância um único par de força e momento resultantes pode ser obtido com infinitas combinações de distribuição de empuxo para cada propulsor. Dessa forma é necessária a implementação de uma lógica de distribuição de esforços, função dada ao módulo de alocação de propulsão.

Chamando de $\mathbf{u}$ um vetor coluna contendo os sinais de atuação enviados aos propulsores, espera-se que após alguns instantes os motores entrem em regime. Escrevendo as componentes da força e do momento resultante na forma de um vetor coluna $\boldsymbol{\tau}$, fica-se com a seguinte relação:

$$\boldsymbol{\tau} = \mathbf{B} \cdot \mathbf{u}$$

Equação 8.1

onde $\mathbf{B}$ é uma matriz que define a geometria da disposição dos propulsores.

Se o número de propulsores fosse igual ao número de graus de liberdade do veículo, ou seja, igual a seis (três de rotação e três de translação), e se a matriz $\mathbf{B}$ não fosse singular, bastaria multiplicar o vetor de força e momento requeridos pelo controlador pela matriz inversa de $\mathbf{B}$ para obtermos o sinal de atuação necessário. Apenas por esclarecimento, a matriz $\mathbf{B}$ ser singular significa fisicamente que existe pelo menos um grau de liberdade do veículo que não é influenciado pelo conjunto de atuadores. Como já dito anteriormente, no veículo em desenvolvimento na Escola Politécnica teve-se, além do cuidado em dar a possibilidade de controlar todos os graus de liberdade do robô, a preocupação em acrescentar uma redundância. Dessa forma a matriz da geometria de disposição dos propulsores não é inversível pois

apresenta um número de colunas (graus de liberdade) diferente do número de linhas (número de propulsores). Seguindo uma proposta de (Fossen, 1994), pode-se encontrar uma matriz pseudo-inversa de **B** tal que o gasto de energia seja minimizado. Esse gasto seria computado através da seguinte relação:

$$J = \frac{1}{2} u^T W u \quad \text{Equação 8.2}$$

onde **W** é uma matriz de pesos dos atuadores. Minimizando a energia necessária **J** em relação a sinal de atuação **u**, encontra-se a seguinte matriz pseudo inversa:

$$B_w^+ = W^{-1} B^T (B W^{-1} B^T)^{-1} \quad \text{Equação 8.3}$$

Caso se deseje dar o mesmo peso para cada propulsor, **W** será uma matriz identidade e a expressão para a pseudo inversa de **B** fica simplificada a:

$$B^+ = B^T (B B^T)^{-1} \quad \text{Equação 8.4}$$

Assim, é possível realizar a alocação de propulsão de acordo com a atuação requerida pelo controlador da seguinte forma:

$$u = B^+ \cdot \tau \quad \text{Equação 8.5}$$

Para o funcionamento normal do sistema o módulo de alocação de propulsão realizaria simplesmente a multiplicação de matrizes acima. Mas, ampliando as capacidades desse módulo para situações de anormalidade como falha de algum propulsor, pode-se fazer com que o módulo seja constituído de diversas matrizes de configuração da geometria dos propulsores e o módulo de supervisão deve indicar ao

módulo de alocação de propulsão qual matriz deve ser utilizada naquele instante. Além disso, o sistema de supervisão deve ser capaz de verificar se falhas de mais de um atuador não levaram a matriz **B** a ser singular, de forma que não é mais possível o controle de algum grau de liberdade do veículo e, nessa situação, decidir como fazer com que o veículo volte à superfície, se possível.

8.6 Módulo Gerador de Referência

O Módulo Gerador de Referência deve fornecer sinais de referência de posição e velocidade desejadas ao veículo. Então, sua funcionalidade é efetiva quando o veículo está em algum dos modos ativos, ou seja, manual, de *hovering* ou automático.

Quando em modo manual o gerador de referência necessita dos dados de comando enviados pelo joystick. Como esses comandos são em velocidade, como dito no item 7.3.2, o gerador de referência deve apenas filtrar esses dados e passá-los adiante. Como referência de posição pode-se fornecer a integral do sinal de velocidade dado pelo joystick ou, mais simplesmente, o próprio sinal de leitura dos sensores. No segundo caso o sinal de controle seria obtido somente do erro em velocidade do veículo.

Em modo de *hovering*, o veículo deve ficar estacionário. Dessa forma, deve ser dada uma referência de posição e orientação constantes e iguais ao último valor lido nos sensores e referências constantes iguais a zero para as inclinações e velocidades lineares e angulares.

A maior dificuldade desse módulo está na geração de referência para o modo automático. Nesse modo deve-se gerar uma trajetória que seja suficiente para o cumprimento de alguma missão e que seja suave o suficiente para ser fisicamente realizável pelos propulsores. O desenvolvimento do processo de geração de trajetória pode ser tão profundo quanto se queira, acrescentando-se funcionalidades, como o

desvio de obstáculos. Para esse trabalho se deseja apenas que o módulo seja capaz de gerar trajetórias simples, em linha reta, unindo um ponto de partida com um ponto de chegada. A referência de posição em função do tempo será dividida em três partes: aceleração, velocidade constante e desaceleração, com devidos parâmetros para que o movimento seja realizável.

9 SELEÇÃO DE FERRAMENTAS

Para que seja possível o desenvolvimento da Arquitetura de Controle necessita-se de ferramentas adequadas, ou seja, um ambiente de programação e de um sistema operacional para sistemas de tempo real. De acordo com a disponibilidade, estão sendo estudados o MatLab e programas agregados, e o Constellation com o sistema operacional VxWorks.

9.1 Simulink / Stateflow

O Simulink é uma ferramenta interativa que acompanha o Matlab e possibilita a modelagem, simulação e análise de sistemas Dinâmicos. O Simulink pode ser usado para explorar o comportamento de uma grande variedade de sistemas do mundo real, como sistemas elétricos, mecânicos ou termodinâmicos.

Através de sua interface gráfica pode-se construir diagramas de blocos, que representam o sistema estudado de forma simples e rápida. Essa ferramenta é totalmente integrada ao Matlab, que disponibiliza diversas ferramentas de projeto e análise.

O Simulink é uma excelente ferramenta para projeto e simulação de sistemas de controle devido à facilidade de construção de modelos e a disponibilidade de uma extensa biblioteca de funções. Além disso, pode-se inserir funções escritas em linguagem C, flexibilizando o uso dos diagramas de blocos, que podem comportar algoritmos escritos pelo próprio usuário.

Acompanhado o Simulink existe uma outra ferramenta gráfica chamada Stateflow. O Stateflow permite a simulação de sistemas reativos complexos baseados na teoria de máquinas de estado, permitindo o desenvolvimento de lógicas de controle e supervisão em conjunto com os modelos do Simulink.

9.1.1 Real-Time Workshop

O Real-Time Workshop é uma outra ferramenta do Matlab que gera código ANSI otimizado, portátil, e modificável a partir de modelos do Simulink. Ele automaticamente constrói programas de tempo-real executáveis. O código gerado pode rodar em um hardware PC comum, microcontroladores ou em sistemas operacionais comerciais ou proprietários. Essa é uma ferramenta que pode ser utilizada para a prototipagem rápida de uma grande variedade sistemas de tempo-real.

O RTW permite que se utilize toda a extensa biblioteca do Simulink para cálculo e processamento matemático diretamente no hardware disponível, sem a necessidade de se codificar todo o modelo manualmente. A facilidade para criação e modificação dos modelos, assim como o acerto de parâmetros de regulação do sistema agilizam muito a validação de modelos e verificação da comunicação entre as diversas partes do hardware.

9.2 Constellation e VxWorks

9.2.1 Constellation

O *Constellation* é uma ferramenta de alto nível em desenvolvimento de sistemas para aplicações de tempo-real. Ele cobre uma grande parte do ciclo de desenvolvimento, passando pelo projeto conceitual, geração de código e gerenciamento de código.

Fica disponível, assim, uma plataforma completa de desenvolvimento, que fornece suporte para o desenvolvimento de sistemas complexos em tempo-real. Ele oferece uma combinação de facilidades, serviços e ferramentas que agilizam o projeto e implementação do sistema.

Constituintes do *Constellation*:

- **Sistema de programação gráfica:** que fornece uma incrível facilidade de visualização do sistema.
- **Cadeia integrada de ferramentas:** que oferece a ferramenta certa para cada etapa do desenvolvimento do projeto.

Entre suas características, distingue-se:

- **Composição hierárquica:** que fornece uma visão “top-down” da arquitetura do sistema.
- **Estrutura baseada em componentes:** que encoraja o projeto modular do programa e reutilização de código.
- **Coleção de “softwares” robusta:** as ferramentas acompanham serviços essenciais para o programas a ser desenvolvido.

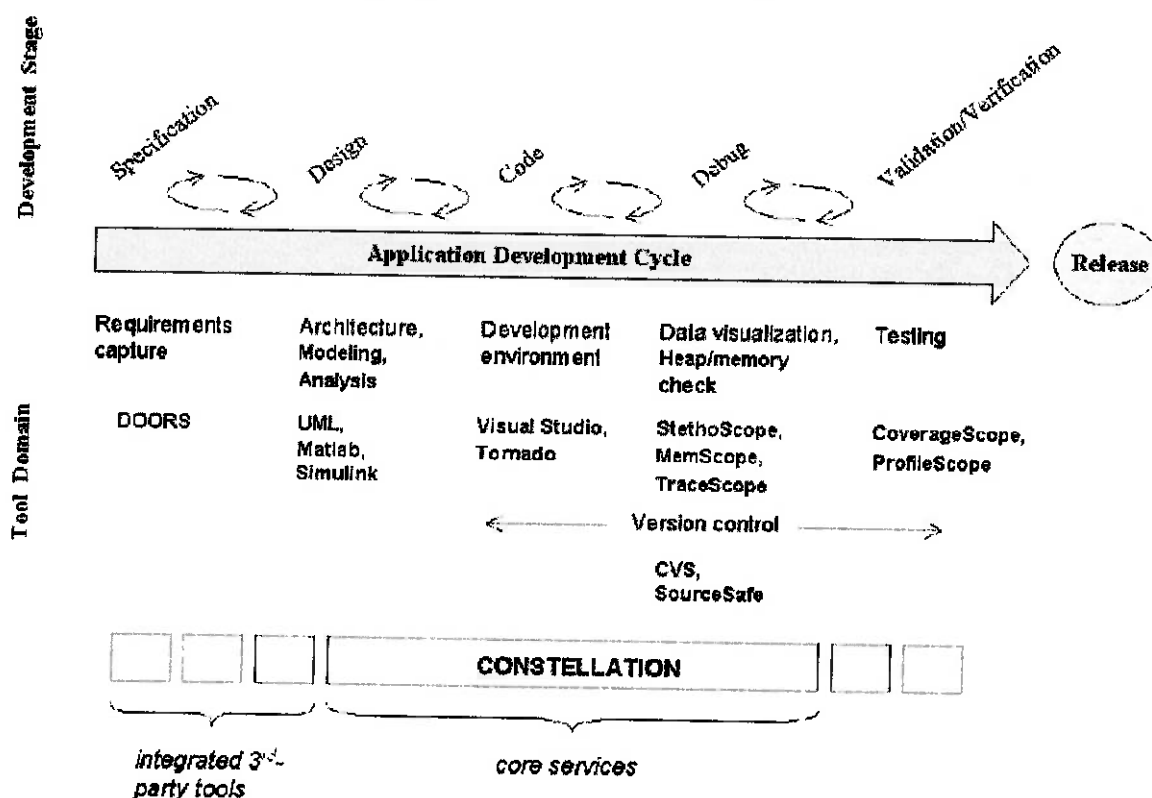


Figura 9.1 – Esquema do Ciclo de Desenvolvimento com o Constellation

9.2.2 VxWorks

O *Unix* e o *Windows* são excelentes sistemas para o desenvolvimento de programas e para muitas aplicações interativas, mas não são apropriados para aplicações em tempo real. Por outro lado, sistemas operacionais de tempo-real tradicionais oferecem ambientes de desenvolvimento pobres ou componentes que não são de tempo-real, como interfaces gráficas (GUIs – *graphical user interfaces*).

Ao invés de criar um sistema operacional que faz tudo isso, a filosofia da *Wind River* é de utilizar dois sistemas complementares, que cooperam entre si (*VxWorks* e *UNIX*, ou *VxWorks* e *Windows*) e deixar que cada um faça o seu melhor. O *VxWorks* as funções críticas de tempo-real, enquanto outras máquinas contendo *UNIX* ou *Windows* são usadas para o desenvolvimento da aplicação ou outras funções que não tem fortes requisitos de tempo.

Características do *VxWorks*:

- **Recursos do núcleo de tempo-real de alta performance:** o núcleo do *VxWorks*, *wind*, inclui multi-tarefa com agendamento de prioridade preemptivo, sincronização de inter-tarefas e recursos de comunicação, suporte a interrupções, temporizadores “*watchdog*”, e gerenciamento de memória.
- **Compatibilidade POSIX:** o *VxWorks* fornece a maioria das interfaces especificadas pelo padrão 1003.1b, simplificando as portas de outros sistemas conformes.
- **Sistema E/S:** o *VxWorks* fornece um sistema de E/S compatível com o ANSI C rápido e flexível.
- **Sistema de arquivos local:** *VxWorks* fornece uma sistema de arquivos rápido especial para aplicações de tempo-real.
- **Objetos de memória compartilhada (*VxMP Option*):** a opção *VxMP* fornece um recurso de semáforos compartilhados, fila de mensagens, e regiões de memória entre tarefas e processos diferentes.
- **Memória virtual (*VxVMI Option*)**

- **Ferramentas de avaliação de desempenho:** entre as ferramentas de avaliação de desempenho estão um temporizador de execução para sincronizar uma rotina ou grupo de rotinas, e utilitários de utilização da *CPU* em porcentagem por tarefa.
- **Suporte para placas:** o pacote de suporte para placas (*BSPs – board support packages*) estão disponíveis para uma variedade de placas e fornecem rotinas de inicialização de *hardware*, *setup* de interrupções, temporizadores, mapeamento de memória, etc.
- **Simulador *VxWorks* (*VxSim*) e analisador lógico (*WindView*):** o ambiente de programação *Tornado*, acompanha um simulador integrado e um analisador lógico para todas as plataformas hospedeiras. O *VxSim* simula um alvo *VxWorks* para uso como protótipo e ambiente de teste. O *WindView* fornece uma ferramenta avançada de *debugagem* para o simulador de ambiente.
- **Recursos de rede:** o *VxWorks* fornece acesso transparente para outros *VxWorks* e sistemas de rede *TCP-IP*. Todos os recursos de rede estão de acordo para os protocolos de *Internet* padrões.

10 IMPLEMENTAÇÃO DA ARQUITETURA DE CONTROLE

Devido a maior facilidade de programação e obtenção de resultados positivos em testes preliminares, o projeto da arquitetura de controle foi implementado em diagramas da ferramenta Simulink do MatLab. Deixa-se claro que existe a possibilidade de transporte do programa para o uso no sistema operacional VxWorks, apesar do Simulink não ser uma ferramenta específica de desenvolvimento de programas para esse sistema operacional como o Constellation.

A implementação foi estruturada de forma a se obter um pacote de simulação de funcionamento e controle do veículo e um outro pacote para funcionamento e controle real formado pelo controle a ser embarcado, desenvolvido no Simulink, e software do computador de apoio, desenvolvido em C++.

10.1 Ambiente de Simulação

Para que a implementação do sistema pudesse ser facilmente testada, sem a necessidade de montagem experimental com o veículo submarino, foi desenvolvido um ambiente de simulação em diagramas do Simulink contendo três blocos principais: o controle a ser embarcado no veículo, o módulo do computador de apoio e um bloco de simulação da dinâmica do veículo. Dessa forma, no ambiente de simulação, o sinal de comando de atuadores é na verdade enviado ao modelo dinâmico do veículo e este retorna os valores que seriam lidos pelos sensores embarcados (posição e orientação), como pode ser observado na figura abaixo.

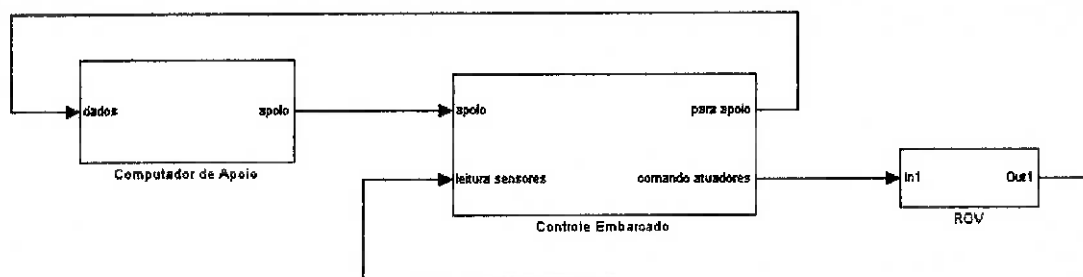


Figura 10.1 - Sistema de Simulação da Arquitetura de Controle

O bloco do controle a ser embarcado está aberto para visualização na próxima figura. Reconhece-se claramente que os blocos apresentados no diagrama são relativos aos módulos descritos no capítulo 8.

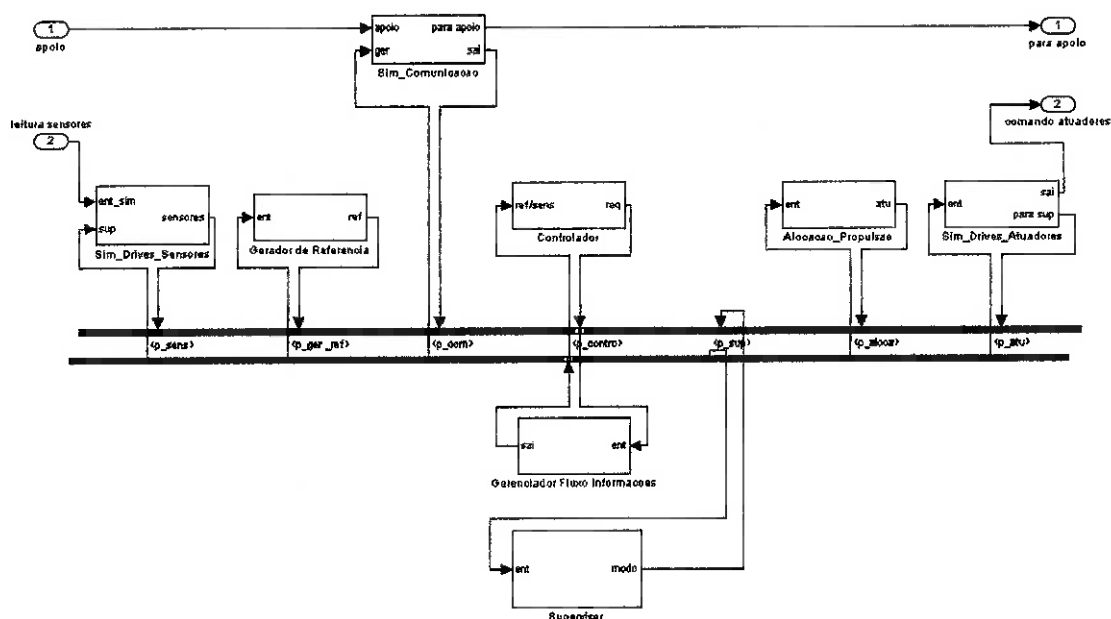


Figura 10.2 - Diagrama do Controle a ser embarcado

O modelo de veículo foi desenvolvido com base no trabalho de (AVILA, 2003) em que a dinâmica do veículo é modelada como seis graus de liberdade (três de translação e três de rotação) independentes. Assim, para cada grau de liberdade tem-se:

$$M \ddot{x} = -k_v \left(\dot{x} - \dot{x}_c \right) + F_{res} + F_{prop}$$

Equação 10.1

Ou seja, a aceleração de uma inércia M é dada pela resultante de uma força de arrasto, função da velocidade relativa entre o veículo e da correnteza, da resultante restaurativa F_{res} devida às forças peso e empuxo e de uma força de propulsão F_{prop} . Para os graus de liberdade lineares a força F_{res} é a projeção sobre cada grau de liberdade da resultante entre a força peso e a força de empuxo (que varia conforme o veículo é inclinado). Para os graus liberdade angulares essa força é na verdade um momento, conseqüente da força de empuxo, existente apenas quando o centro de flutuabilidade difere do centro de gravidade do veículo.

Apesar de (AVILA, 2003) ter estimado os parâmetros do exato veículo utilizado no presente trabalho, a configuração esperada era do veículo completo (quatro vasos de pressão, etc) diferindo da configuração de testes aqui trabalhada. Além disso, não foram estimados parâmetros de rotação. Dessa forma, os parâmetros do veículo foram estimados levando-se em consideração também o trabalho de (INDIVERI, 1998) que trata da estimação dos parâmetros hidrodinâmicos de um veículo bastante similar.

A força de propulsão é obtida com a soma da contribuição de cada propulsor. Os propulsores foram modelados seguindo o trabalho de (BARBARINI, 2003) em que foi encontrada a seguinte relação entre tensão (entrada dos propulsores) e propulsão:

$$F_{prop_i} = k_p u_i |u_i|$$

Equação 10.2

Ou seja, a força obtida por cada propulsor depende do quadrado da tensão de entrada e possui sentido dado pelo sinal da tensão. A referência apresenta que a constante k_p possui valores distintos quando o sinal de atuação é positivo ou negativo.

Os valores adotados para o modelo do veículo estão resumidos abaixo:

- $M = 50,55 \text{ kg}$;
- $k_v \text{ (linear)} = 20 \text{ N/(m/s)}$;
- $k_v \text{ (angular)} = 12 \text{ N.m/(Rad/s)}$;

- k_p (positivo) = $9,66 \text{ N/(m/s)}^2$;
- k_p (negativo) = $6,67 \text{ N/(m/s)}^2$;
- $V_{\text{água}} = 0,0176 \text{ m}^3$;

10.2 Plataforma real de utilização

No caso real de utilização do sistema fica dividido em dois computadores, no computador de apoio (ou Host PC) e no computador embarcado (ou Target PC). As figuras abaixo esquematizam a montagem do sistema:

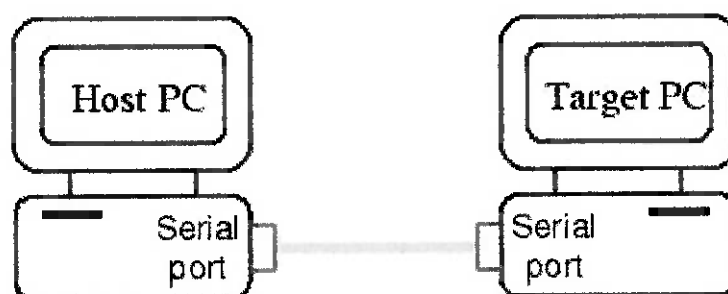


Figura 10.3 - Esquema de montagem física do sistema

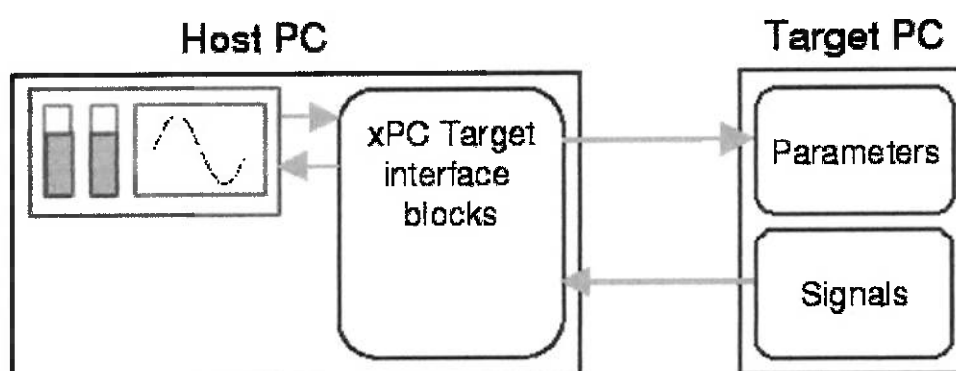


Figura 10.4 - Esquema do funcionamento lógico do sistema

Nesta configuração o Host PC e o Target PC se comunicam através de um cabo serial. O Host PC seria encarregado de mandar os sinais de comando para o

programa rodando no Target PC. O programa seria, então, encarregado de controlar corretamente os motores ligados nos conversores DA do PC104. Ao mesmo tempo, o Target PC pode enviar sinais para a monitoração de volta ao Host PC. Os sinais enviados ao Host PC podem ser utilizados para a monitoração, pilotagem do sistema, ou uma posterior análise de performance com o sistema fora de funcionamento.

Para a montagem do sistema, a primeira coisa a ser desenvolvida é o programa a ser rodado no Target PC em Simulink/Stateflow. Dentro do diagrama estão contidos os diagramas de controle e supervisão do ROV assim como os drivers de pilotagem do hardware. O programa é então compilado utilizando-se o xPC Target do Real-Time Workshop tendo como resultado final um único arquivo executável (controle.dlm). Foi desenvolvido um programa com a finalidade fazer o upload do execução do arquivo controle.dlm no Target PC através da porta serial e realizar a comunicação enviando sinais de comando e lendo os sinais de monitoração. Esse programa foi desenvolvido a partir de uma biblioteca de comunicação do xPC Target. A biblioteca consiste de uma dll cujas funções podem ser acessadas através do Visual C++ da Microsoft.

10.3 Interface com o operador

A interface com o operador constitui-se fisicamente de um computador com um *joystick*. Os comandos dados pelo *joystick* são tratados por um módulo próprio e passados ao módulo de comunicação. Cabe ao módulo de interface disponibilizar informações sobre o estado do veículo na tela do computador de apoio e receber alguns comandos, como requisição de mudança de modo de operação. As interfaces utilizadas no ambiente de simulação e na plataforma real de utilização diferem ligeiramente conforme as funcionalidades disponíveis e as ferramentas de desenvolvimento.

10.3.1 Interface no Ambiente de Simulação

Abaixo está exposta a interface utilizada no ambiente de simulação. Esta utiliza *switches* para a requisição de mudança de modo de operação, caixas para determinação de ponto alvo em missões, LEDs para a informação do modo atual de operação, displays para informação da posição do veículo e relógios para a visualização de sua orientação e inclinação. A orientação do veículo é dada pelo relógio em forma de bússola. Já as inclinações são visualizadas no mostrador de “horizonte” artificial. As coordenadas XY também podem ser visualizadas num plano referente a uma vista superior da posição do veículo.

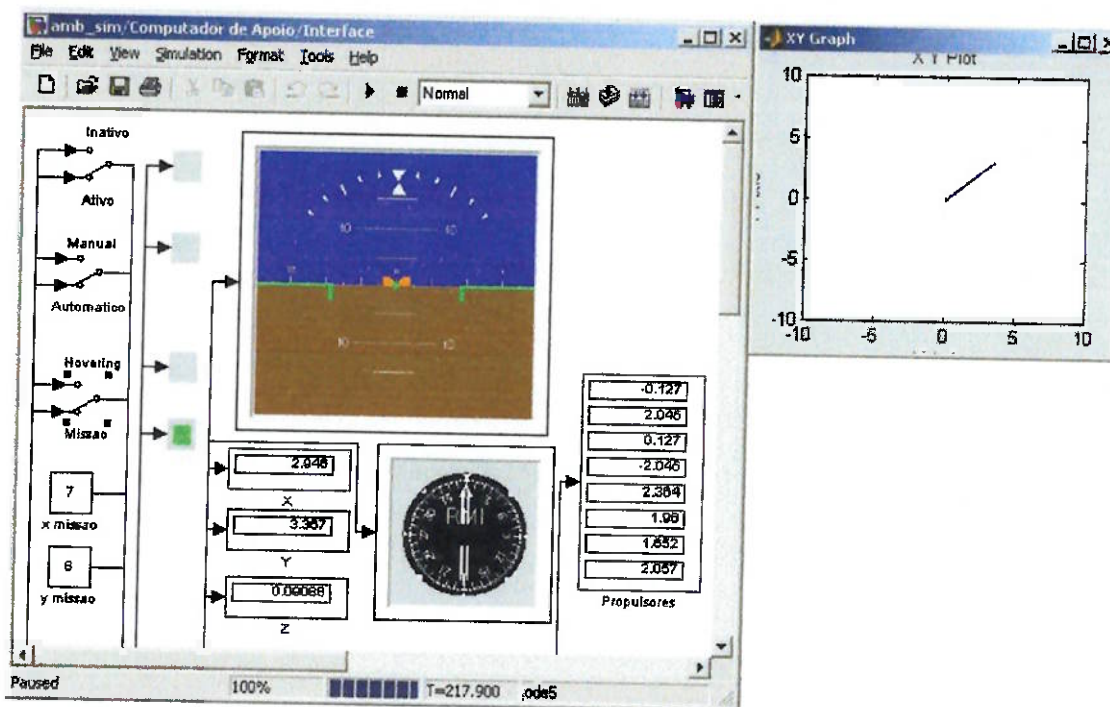


Figura 10.5 - Tela de Interface com operador

Nota-se que a presente interface apresenta alguns defeitos como poluição visual e uso indevido dos *switches* como meio de requisição de mudança de modo de operação. A poluição visual é devida à própria forma em que se desenvolve programas no Simulink, através de diagramas de blocos. Já os *switches* deveriam ser substituídos por *push-buttons*, pois estes representam de forma adequada uma

requisição. Essas são algumas limitações da ferramenta de desenvolvimento que são contornadas com o desenvolvimento da interface com programação C++.

10.3.2 Interface na Plataforma Real

A interface para o uso real de controle do veículo é similar à interface anterior, com a principal diferença de apresentar comandos para execução e carregamento do controle embarcado. Segue uma tela de exemplo:

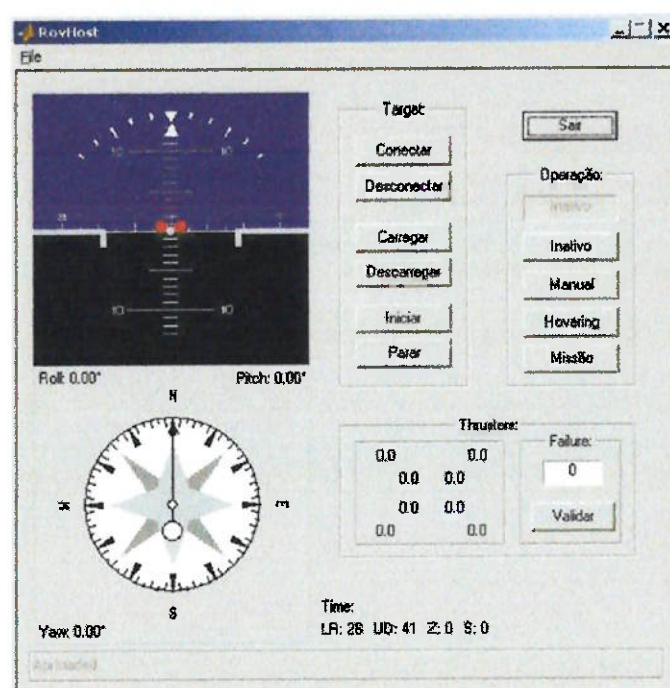


Figura 10.6 - Tela do programa de comunicação do computador de apoio

Através das diversas funções fornecidas pela biblioteca pode-se carregar o programa, atualizar parâmetros no computador embarcado e ler os sinais desejados.

Dentre as funcionalidades acessíveis da interface gráfica do programa estão:

- Conectar-se/desconectar-se da porta RS232;
- Carregar/descarregar o programa no Target PC;
- Iniciar/parar o programa;

- Selecionar o modo de operação do ROV;
- Indicar qual propulsor não funciona na simulação de falha;
- Visualizar o sinal dos sensores (roll, pitch, yaw);
- Visualizar o sinal de comando enviado aos propulsores.

10.4 Driver do Joystick

O Simulink dispõe em sua biblioteca um bloco para a utilização do *joystick*, bastando apenas a adequação da disponibilidade dos sinais conforme o uso desejado de controle do veículo. A única alteração significativa foi a adição de blocos de “*dead zone*” ou zona morta. Esses blocos agem como filtros que barram o sinal quando seu módulo é menor que um parâmetro limiar e deixam passar o sinal caso contrário. Isso evita que pequenas descentralizações do joystick sejam incorretamente interpretadas como ações de comando.

No desenvolvimento em C++ foram utilizados drivers DirectX disponíveis em biblioteca e adicionadas as mesmas propriedades descritas anteriormente.

10.5 Módulos de Comunicação

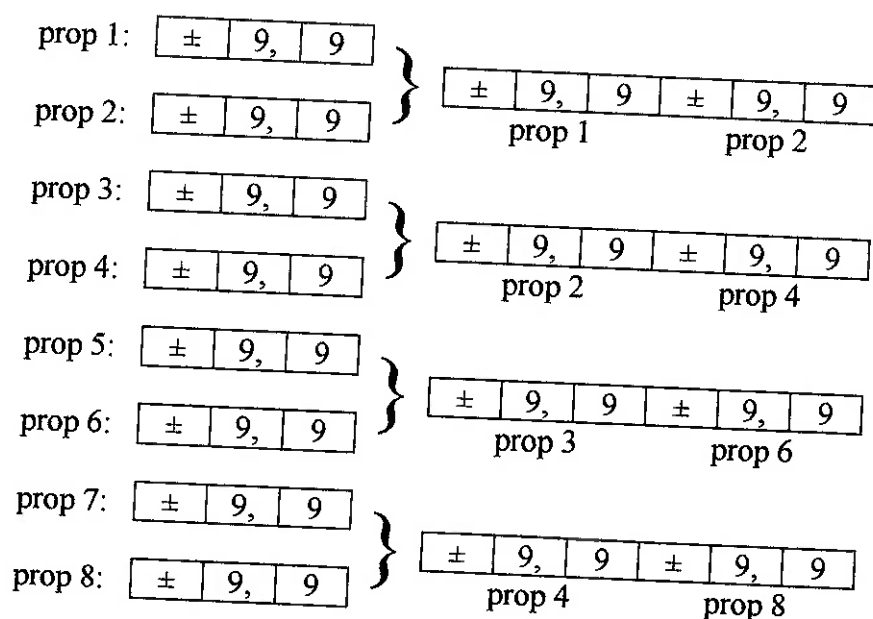
Os módulos de comunicação utilizam funcionalidades disponíveis em biblioteca, sendo necessária apenas sua configuração para utilização. O problema enfrentado durante o desenvolvimento desse módulo surgiu da grande quantidade de sinais que transitariam pelo cabo serial. Mesmo com a porta serial trabalhando a uma taxa de 115200 kbits/s e um período de amostragem de 0.1 segundos não foi possível ler e enviar todos os sinais diretamente.

Segue uma lista se sinais de comunicação:

- Envio de 3 sinais do joytick representando os 3 eixos de movimentação disponíveis no dispositivo;
- Envio do sinal de mudança do modo de operação;
- Envio do sinal de falha em um dos propulsores;
- Leitura dos 3 ângulos fornecidos pela bússola e inclinômetros;
- Leitura do modo de operação;
- Leitura de 8 sinais de comando dos propulsores.

Para não sobrecarregar a porta serial, optou-se por fazer a leitura dos sinais dos sensores de maneira alternada com os sinais dos propulsores. Em um período de amostragem ocorre a leitura dos sensores e no próximo ocorre a leitura dos propulsores, repetindo-se assim o ciclo a cada dois períodos.

Outra atitude tomada para a redução do trânsito de informações pela porta serial foi a codificação da leitura dos propulsores. Como a biblioteca de comunicação do xPC Target não permite o controle do tipo nem do tamanho dos dados enviados, optou-se por codificar a leitura de dois propulsores na leitura de apenas um número; passando a quatro o número total de números para leitura dos oito propulsores. Antes dos dados dos propulsores serem lidos pelo Host PC eles passam pela codificação esquematizada a seguir:



Os números transmitidos desta forma podem variar entre -9,9 a 9,9 sendo que o sinal também é passado na forma de um número (1 para positivo e 2 para negativo). Para ilustrar melhor a codificação dos propulsores seguem alguns exemplos:

- prop 1 = +2,4 V e prop 2 = +4,7 V → Número codificado = 124147
- prop 1 = -2,4 V e prop 2 = -4,7 V → Número codificado = 224247
- prop 1 = +2,4 V e prop 2 = -4,7 V → Número codificado = 124247
- prop 1 = -3,4 V e prop 2 = +0,9 V → Número codificado = 234109

Essas medidas foram suficientes para adequar o fluxo de informações necessário às condições de comunicação disponíveis.

10.6 Módulo Supervisor

O módulo supervisor deve ser capaz de identificar os estados do sistema e dar orientações sobre como os outros módulos devem funcionar. Assim, a forma de implementação desse módulo foi baseada no modelo de máquina de estados, suportado pela ferramenta State-flow do Matlab.

A principal variável de estado associada ao módulo supervisor é o modo de operação em que se encontra o veículo e a maior parte das mudanças de estado estão relacionadas a requisições de mudança de modo de operação.

Na figura abaixo se pode observar a implementação dos possíveis modos de operação e como ocorre a alternância entre eles.

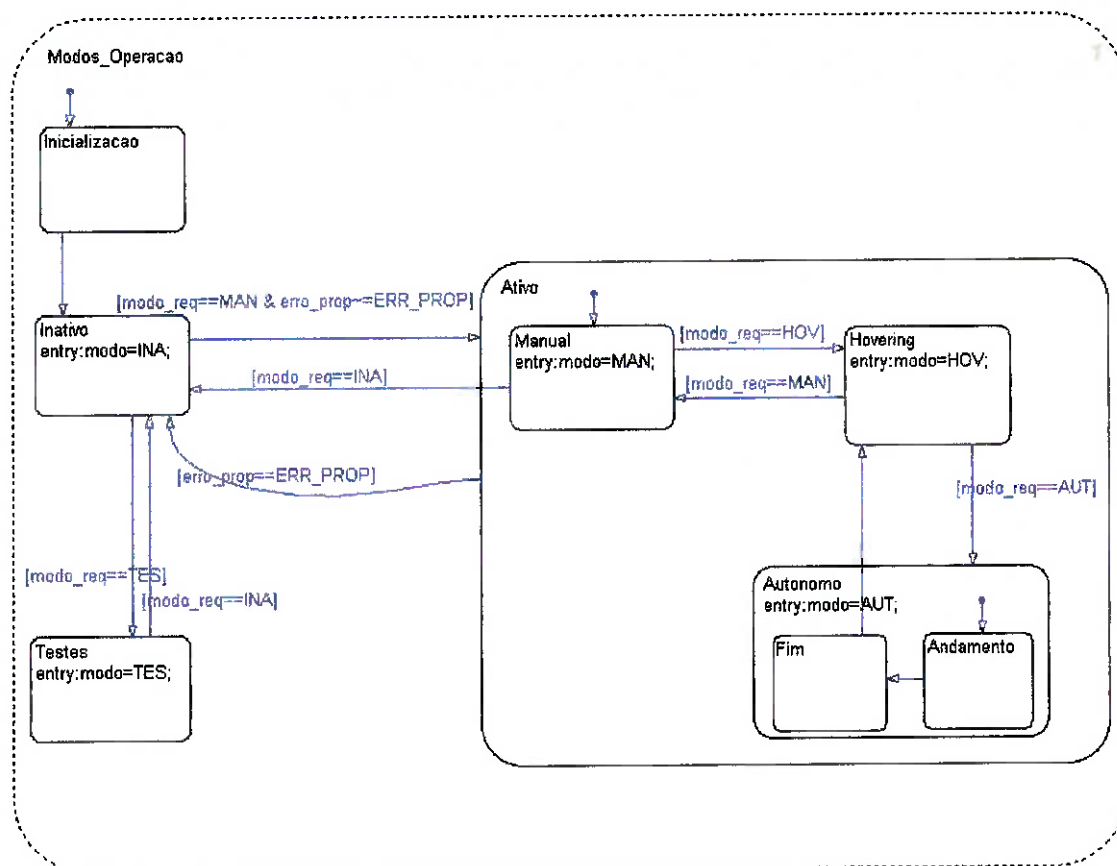


Figura 10.7 - Diagrama de Estados implementado no StateFlow

As caixas do diagrama representam os estados e as setas as possíveis transições. As transições ocorrem segundo a validação dos testes de condições escrita nas expressões entre colchetes. Destaca-se, ainda, que um estado pode conter sub-estados.

Dessa forma, quando ligado o veículo entra em inicialização e em seguida vai para o modo inativo. Segundo a solicitação e a ausência de erro no conjunto de propulsores pode-se alternar para o modo manual, dentro de um estado genérico chamado ativo. Seguindo passos equivalentes pode-se alternar para *hovering* e automático. Caso exista erro no conjunto de propulsores existe uma condição para que o modo seja alternado para inativo, não importa qual sub-estado do estado ativo que se encontre o veículo.

Além do diagrama apresentado acima, o módulo supervisor também é dotado da funcionalidade que avalia o erro no conjunto de propulsores. A função de

detecção de estado de funcionamento ou falha de cada propulsor foi delegada ao módulo *driver* dos propulsores. O módulo supervisor utiliza essa informação para avaliar se o conjunto de propulsores disponíveis é capaz de fornecer as requisições de força e momento resultantes determinadas pelo controlador. Para a configuração particular de oito propulsores do veículo em desenvolvimento avaliou-se que a falha de um propulsor pode ser superada com a realocação do empuxo para os demais propulsores. Já a falha de dois propulsores leva à possibilidade matemática mas à inviabilidade física de realocação de propulsão devido à necessidade de utilização de inversão de matriz quase-singular (ver item 8.5). Então, para efeitos práticos, considera-se que a falha de dois ou mais propulsores gera um estado de erro do conjunto de propulsores.

Na continuação da implementação espera-se ampliar o diagrama de estados do veículo com estados relativos a falha de sensores e baixo nível de energia disponível.

10.7 Módulo Controlador

O projeto de um controlador robusto para um veículo submarino é de alta complexidade e foge ao objetivo desse trabalho. Mas para se fazer o projeto da arquitetura de controle é necessário saber que tipos de informações são normalmente requisitados pelo controlador e precisa-se de algo que feche a malha de controle para que seja possível testar o sistema.

Existem projetos de controladores que utilizam como referência apenas a posição e a orientação do veículo. Da comparação dessas referências com os valores lidos nos sensores são calculados os esforços a serem aplicados sobre o veículo. Por outro lado, o controle em modo manual sugere que a malha de controle seja fechada utilizando-se as variáveis de velocidade. Dessa forma, é proposto um controlador híbrido que, quando em modo manual, utiliza um controlador do tipo PI sobre o sinal de velocidade do veículo e, quando em modo automático, utiliza um controlador do tipo PD sobre a posição.

Antes de prosseguirmos é necessário deixar claro que são utilizados dois sistemas de referência diferentes nos cálculos: um sistema de referência inercial e um outro solidário ao veículo, conforme a figura abaixo.

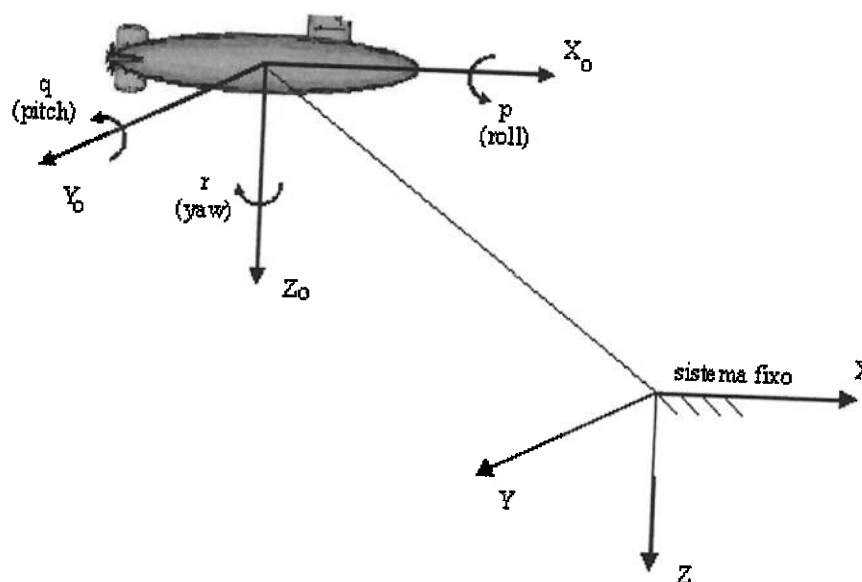


Figura 10.8 - Ilustração com sistemas de referência inercial e solidário ao veículo

A utilização de dois sistemas de referência facilita os cálculos relativos aos esforços necessários sobre o veículo, bem como a alocação de propulsão. Observa-se também que a referência dada pelo *joystick* se trata de uma referência de velocidade segundo o referencial solidário ao veículo, reforçando a necessidade de utilização de dois referenciais.

Agora é possível partir para a análise do controlador proposto cujo diagrama segue na figura seguinte.

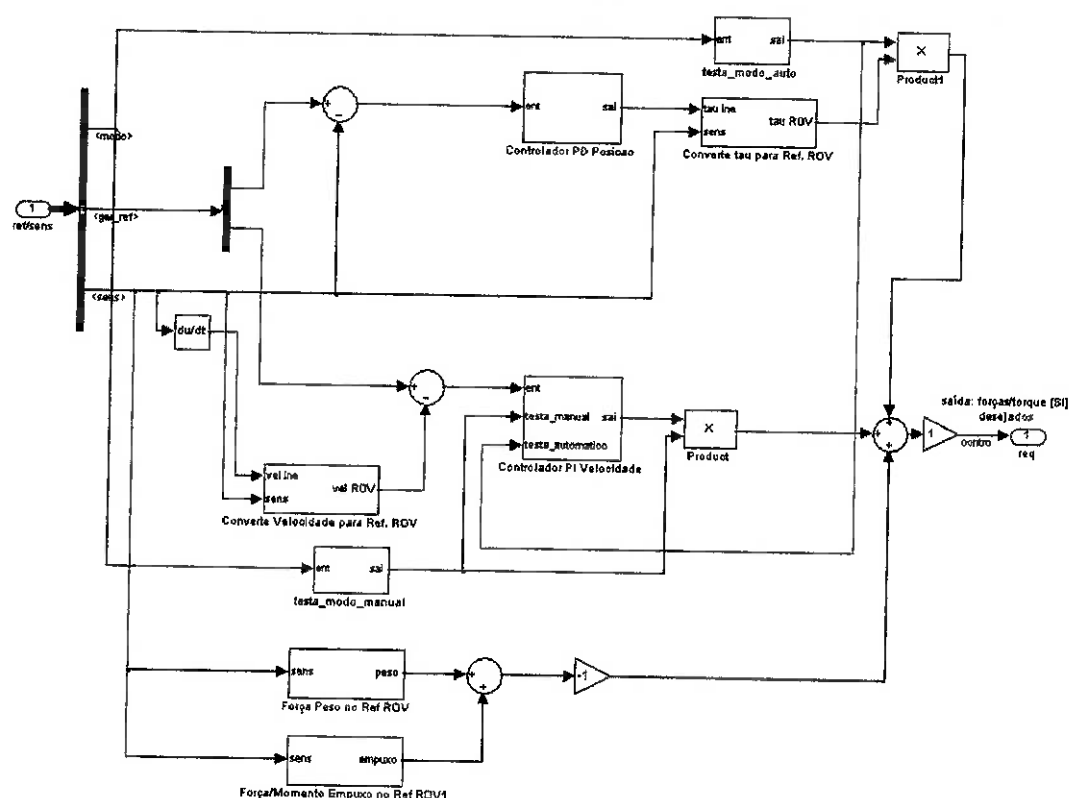


Figura 10.9 - Diagrama do módulo controlador

Primeiramente temos que o sinal de referência é dividido entre referências de posição (e orientação) e velocidade. O sinal de estado de velocidade do veículo é obtido pela derivada do sinal de sua posição. Com esses sinais calcula-se o erro em posição e em velocidade. Lembrando que as referências de posição são utilizadas em modo automático e em relação ao referencial inercial, utiliza-se o erro de posição para o cálculo de esforços necessários, segundo o referencial inercial, sendo necessário em seguida a conversão desses esforços para o referencial solidário ao veículo. Já a referência de velocidade é dada diretamente no referencial do veículo, sendo necessária apenas a transformação do estado de velocidade.

Nota-se também que a compensação das forças restaurativas (peso, empuxo e momento de empuxo) é realizada diretamente, sem a necessidade cálculo de erros de referência, pois podem ser estimadas a priori. Um único porém vem do fato de ser

necessária a constante conversão dessas forças para o referencial do veículo para que a resultante de propulsão seja calculada corretamente.

10.7.1 Cálculo dos parâmetros dos controladores

Os parâmetros dos controladores são projetados com base no modelo apresentado no item 10.1. Aplicando a Transformada de Laplace para cada grau de liberdade tem-se:

Linear:

$$X = \frac{F}{s(s.M + k_{vl})} \quad \text{Equação 10.3}$$

Angular:

$$\Theta = \frac{T}{s(s.J + k_{va})} \quad \text{Equação 10.4}$$

onde **F** é a resultante das forças restaurativas e de propulsão, **T** é a resultante dos momentos restaurativos e de propulsão e **s** é a variável de Laplace.

A função de transferência para o cálculo dos controladores de velocidade possuem um pólo a menos (não possuem o termo integrativo):

Linear:

$$V = \frac{F}{s.M + k_{vl}} \quad \text{Equação 10.5}$$

Angular:

$$\Omega = \frac{T}{s.J + k_{va}} \quad \text{Equação 10.6}$$

No controlador de posição angular do ROV temos para a planta a seguinte função de transferência já com os devidos valores substituídos:

$$\Theta = \frac{T}{s(s.8,425 + 12)} \quad \text{Equação 10.7}$$

Como estamos admitindo que o momento de inércia J e o coeficiente de arrasto hidrodinâmico são os mesmos para os três eixos de rotação, o mesmo controlador servirá para esses três graus de liberdade.

Como esta planta já possui um integrador, o controlador mais simples que poderia ser projetado é do tipo proporcional, desejando-se obter erro em regime zero para entrada degrau. Com a ajuda do Matlab foi traçado o gráfico do Lugar das Raízes desta função de transferência com um controlador proporcional.

Os requisitos de desempenho foram definidos da seguinte forma para um degrau unitário:

- $t_s (98\%) < 3 \text{ s}$
- $M_p < 5 \%$

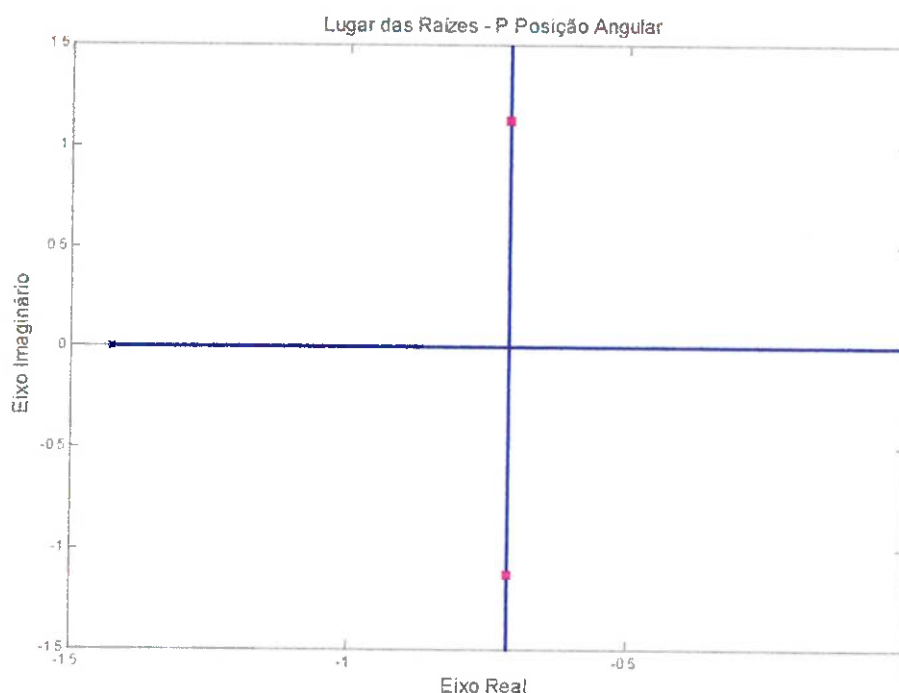


Figura 10.10 - Lugar das raízes para rotação em malha fechada para um controlador do tipo P

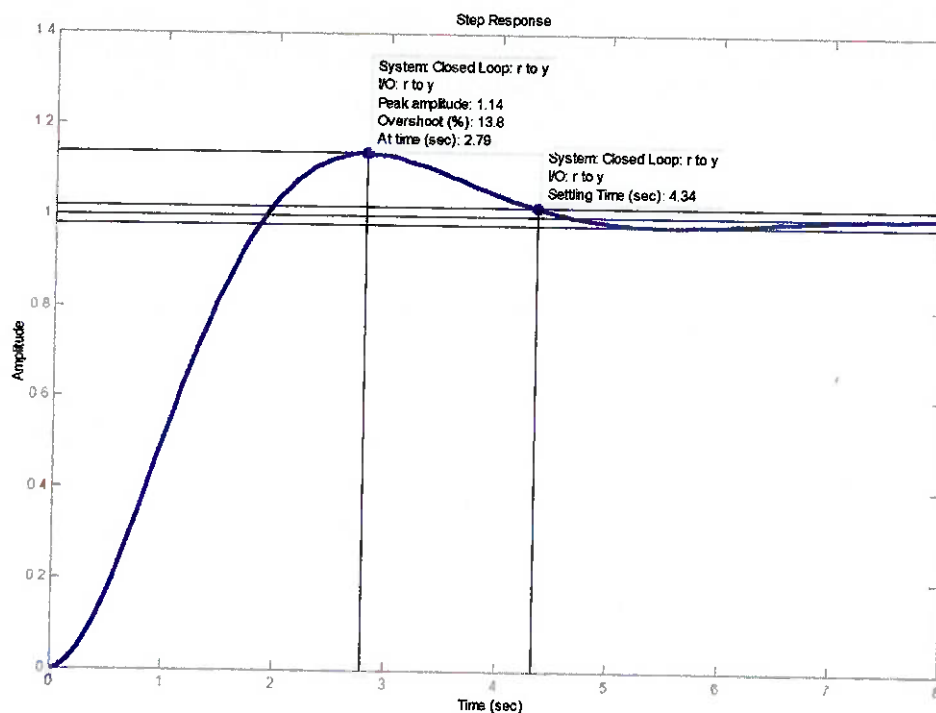


Figura 10.11 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo P

Ao se fazer os pólos da malha fechada se moverem com o ganho K do controlador percebe-se que ele não atinge os requisitos mínimos do projeto. Quando se consegue que o controlador fique mais rápido o sobressinal cresce demasiadamente e quando o controlador atende ao sobressinal máximo o controlador fica muito lento. A figura anterior serve para ilustrar um caso. O controlador acabou ficando muito lento e com alto sobressinal.

Para se atingir os requisitos de desempenho optou-se então por um controlador do tipo PD. O gráfico do Lugar das Raízes e a resposta a um degrau unitário são apresentados a seguir:

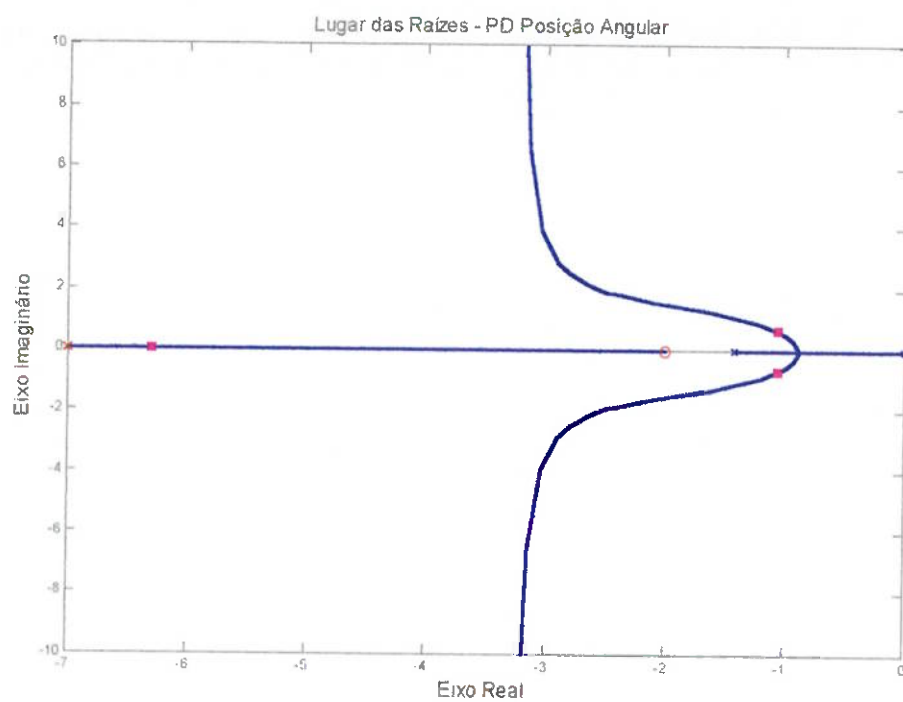


Figura 10.12 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar)

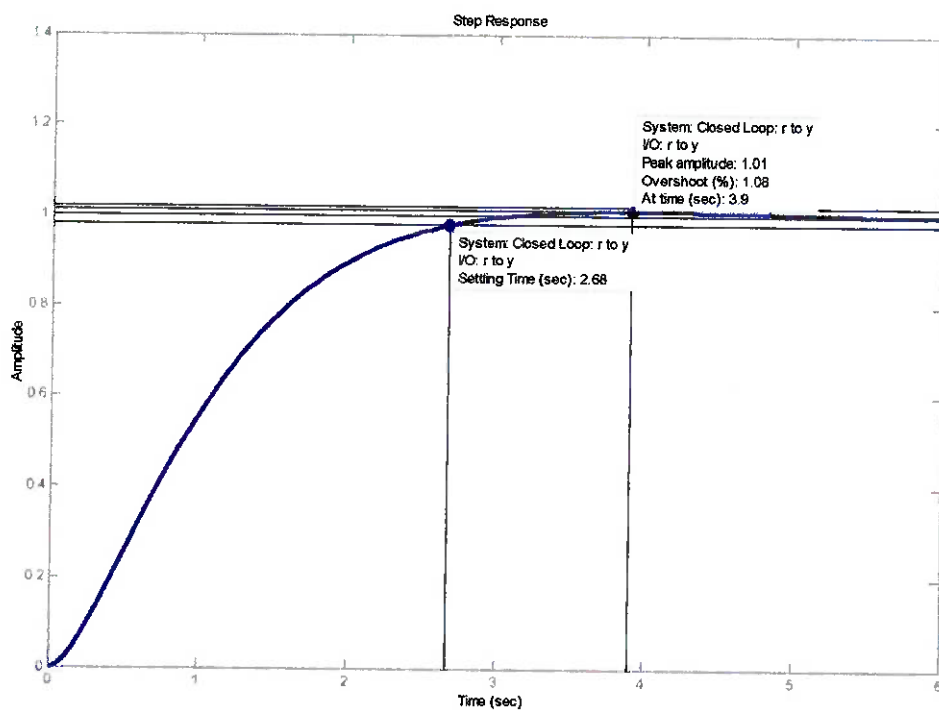


Figura 10.13 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD

Como a função de transferência do controlador PD é imprópria, ou seja, o grau do numerador da função é maior do que o grau do denominador, optou-se por adicionar um pólo rápido ao controlador. Desta forma o tanto o numerador como o denominador da função possuem o mesmo grau.

Como se pode observar a resposta obtida se encaixa perfeitamente nos requisitos, estando o sobressinal com 1,08 % e tempo de assentamento em 2,68 s.

A função de transferência do controlador fica então:

$$G_C = 12 \cdot \left(\frac{s+2}{s+7} \right)$$

Equação 10.8

Na sequência foram feitas algumas experiências com a variação da posição do pólo da planta. Como a incerteza sobre os parâmetros da planta são muito grandes desejou-se verificar o efeito de uma variação nos pólos para o controlador anteriormente projetado.

Para a experiência deu-se um acréscimo de mais 10 % na posição do pólo, simulando por exemplo em erro de 10 % na avaliação dos coeficientes de arrasto supondo que o momento de inércia esteja correto. Seguem os gráficos resultantes:

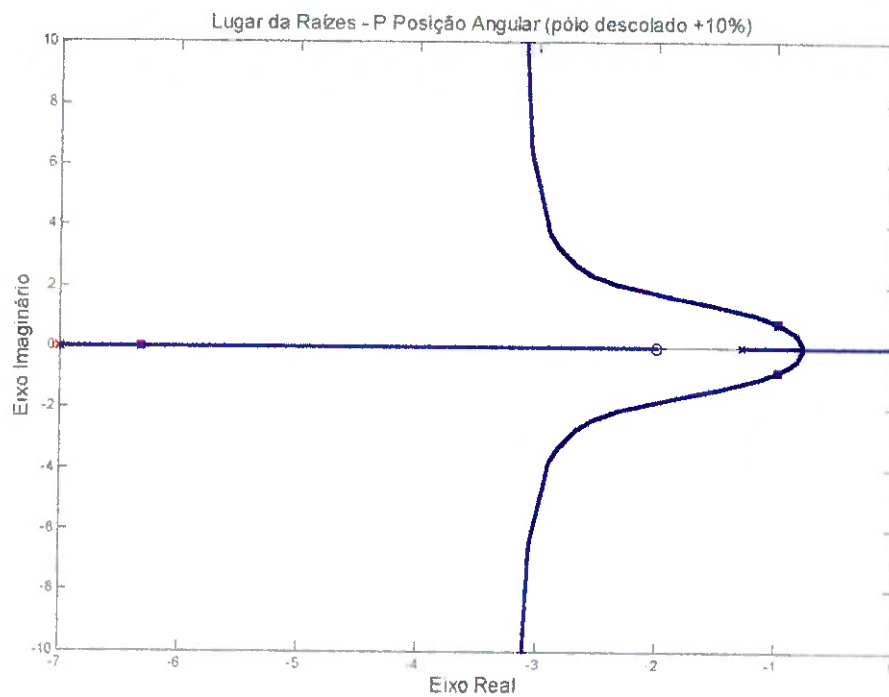


Figura 10.14 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar) e incerteza de 10% em parâmetro da planta

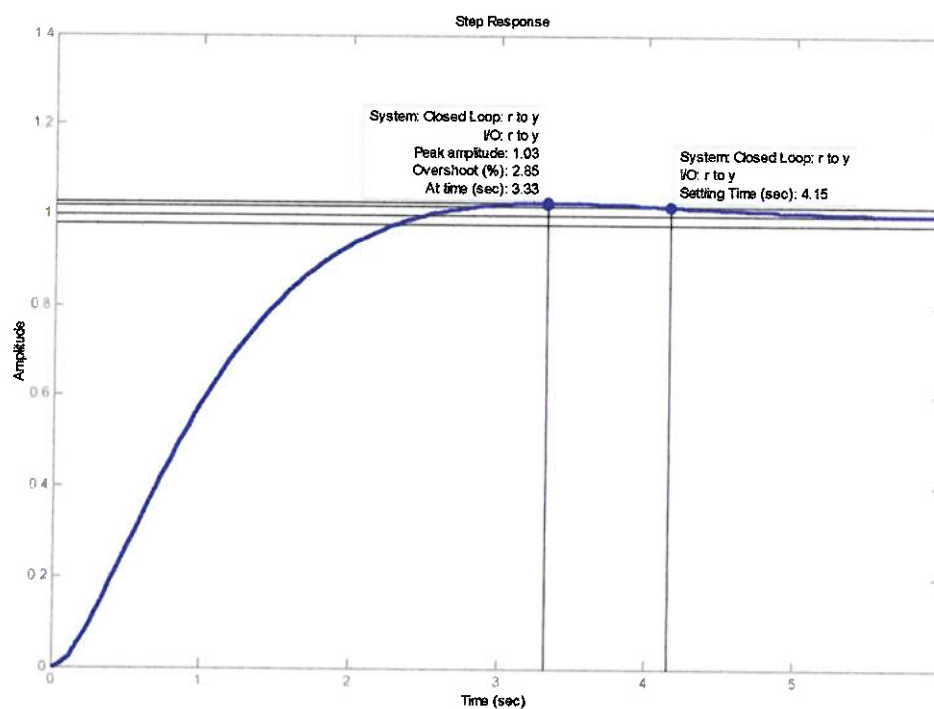


Figura 10.15 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD e incerteza de 10% em parâmetro da planta

Observe que o pólo da planta passou de -1,4243 para -1,2819. Essa mudança representou uma diminuição do desempenho do sistema com o sobressinal indo de 1,08 % para 2,85 % e tempo de assentamento de 2,68 s para 4,15 s. Vemos aí que uma mudança nesse sentido acarretaria no não atendimento dos requisitos de projeto.

Passamos então para a experiência inversa. O pólo da planta foi variado em - 10 %. Os gráficos são apresentados a seguir:

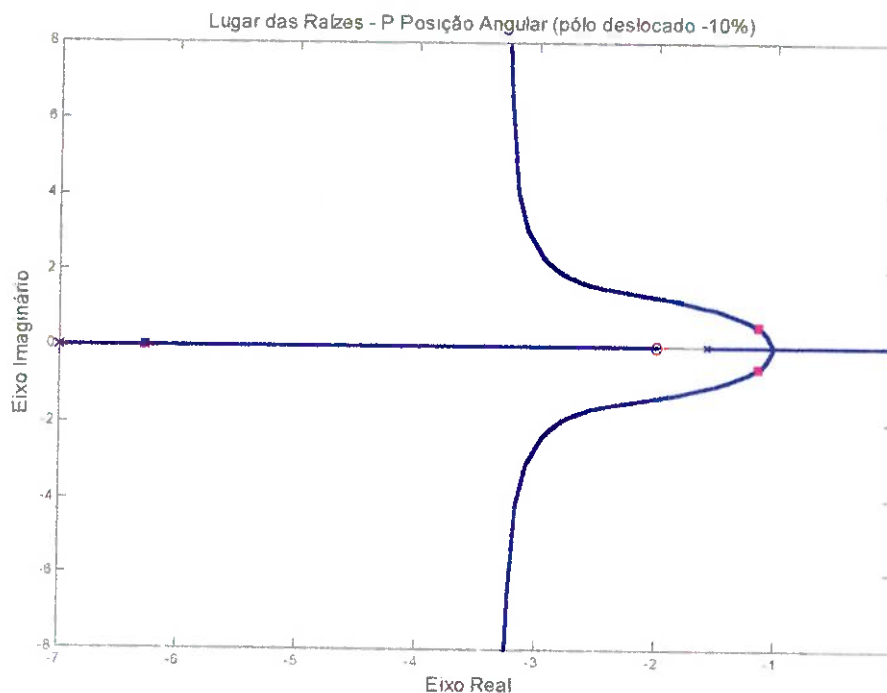


Figura 10.16 - Lugar das raízes para rotação em malha fechada para um controlador do tipo PD (com adição de pólo auxiliar) e alteração no pólo da planta

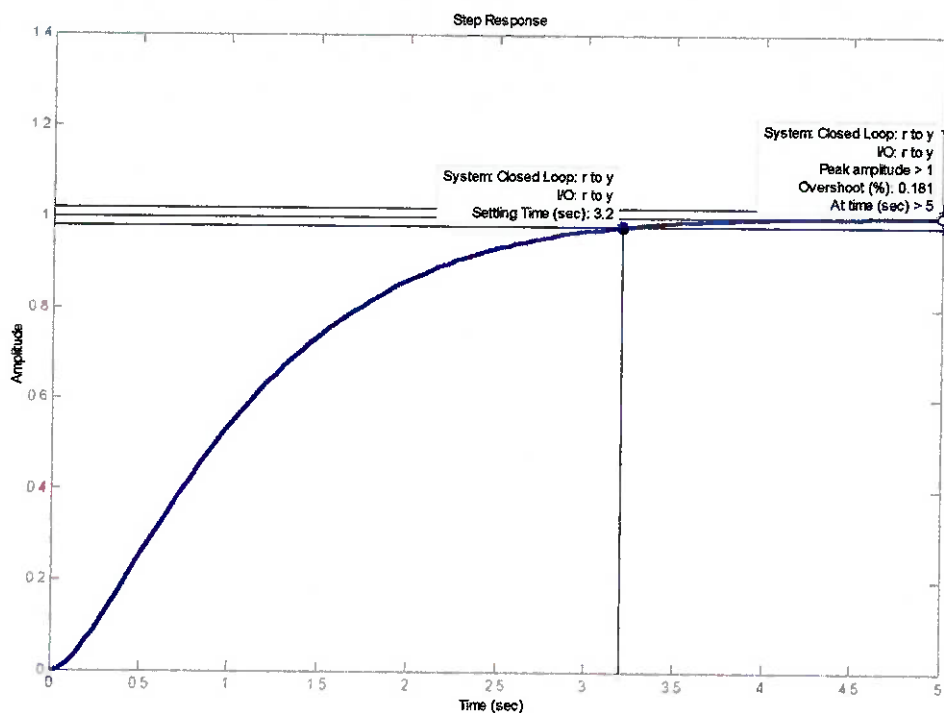


Figura 10.17 - Resposta a degrau unitário em rotação em malha fechada para um controlador do tipo PD e alteração no pólo da planta

Como esta mudança o pólo passou de -1,4243 para -1,5668. Neste caso pode-se ver o prejuízo causado pela incorreta avaliação dos parâmetros do sistema, pois neste também os requisitos do sistema não foram atendidos. O sobre-sinal diminuiu, mas o tempo de assentamento ultrapassou o limite de projeto.

Para o projeto do controlador de velocidade angular temos como função de transferência:

$$\Omega = \frac{T}{s \cdot 8,425 + 12} \quad \text{Equação 10.9}$$

Como a função não possui um integrador optou-se por um controlador do tipo PI, procurando satisfazer os seguintes requisitos para uma resposta degrau unitário:

- $t_s (98\%) < 3 \text{ s}$
- $M_p < 5\%$

O gráfico do Lugar das Raízes resultou no seguinte:

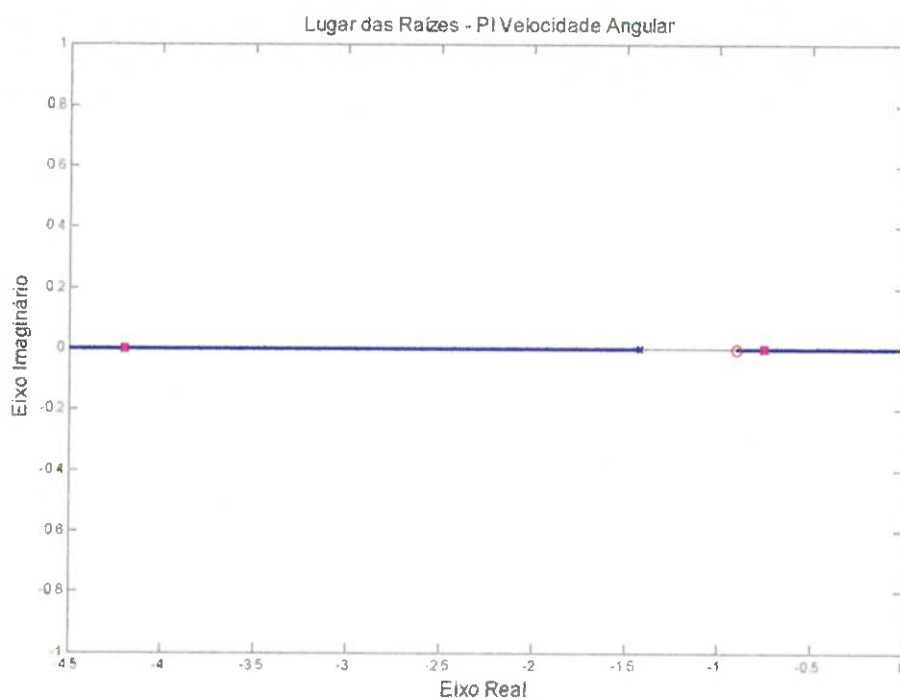


Figura 10.18 - Lugar das raízes para velocidade angular em malha fechada para um controlador do tipo PI

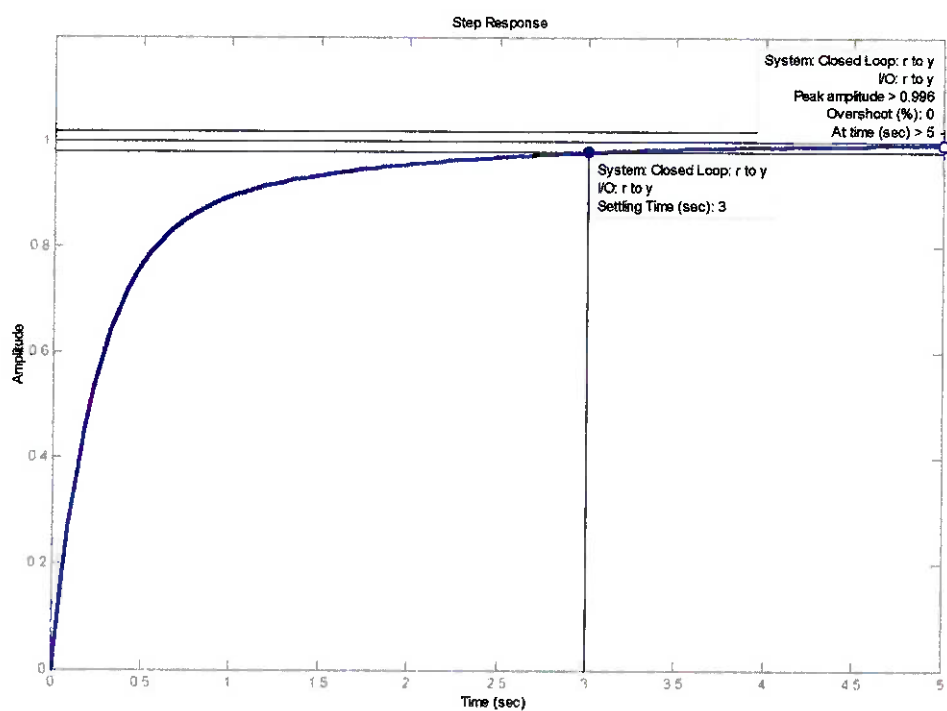


Figura 10.19 - Resposta a degrau unitário para velocidade angular em malha fechada para um controlador do tipo PI

Os requisitos de desempenho foram atingidos com uma certa folga mesmo não tendo componente derivativa no controlador. O tempo de assentamento ficou em exatos 3 s e o sobressinal não existe.

A função de transferência do controlador fica:

$$G_C = 29,778 \cdot \frac{(s + 0,9)}{s} \quad \text{Equação 10.10}$$

O projeto do controlador de posição linear foi um pouco diferente dos outros. O importante para esse controlador é que ele seja muito rápido, não se importando muito com os limites dos propulsores. Fica a cargo do gerador de trajetórias criar curvas aceleração e desaceleração adequadas ao projeto.

Tendo em vista a necessidade de um controlador rápido, chegou-se a esta função para o controlador após algumas tentativas:

$$G_C = 4200 \cdot \frac{(s + 2)}{(s + 7)} \quad \text{Equação 10.11}$$

Para avaliar o desempenho do controlador foi traçado o gráfico de sua resposta a uma rampa unitária:

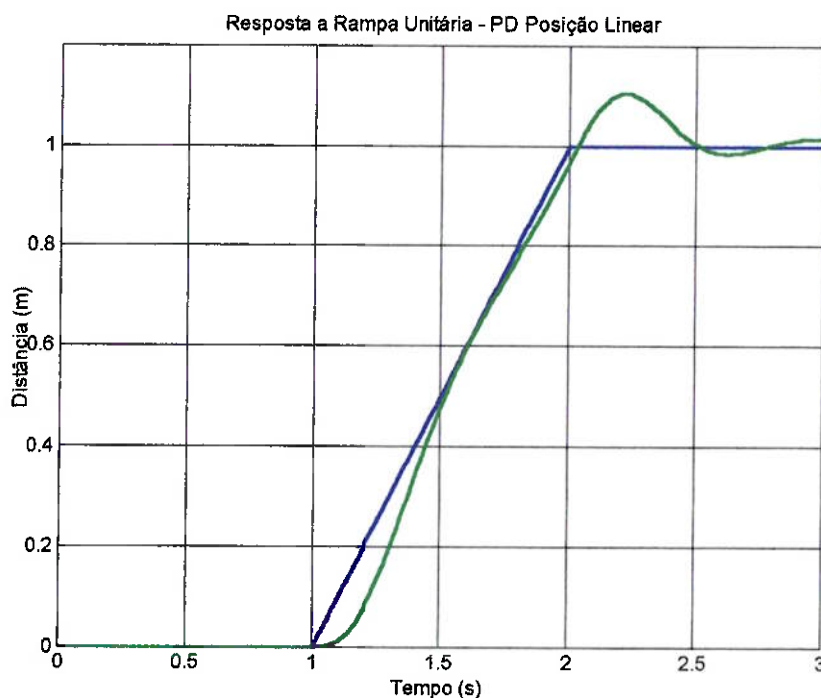


Figura 10.20 - Resposta a rampa para posição linear em malha fechada

Observa-se que no início a função de teste (função rampa) não possui primeira derivada contínua e essa mudança brusca faz com que o sistema não seja capaz de seguir a referência. Isso não é problema pois exatamente essas mudanças bruscas que vão ser evitadas no cálculo de referência dada pelo gerador de trajetória. O importante é notar que enquanto a referência varia continuamente o sistema a acompanha com bastante fidelidade.

Para o cálculo do controlador para velocidade linear considera-se:

$$V = \frac{F}{s \cdot 50,55 + 20} \quad \text{Equação 10.12}$$

Como a planta não possui integrador o controlador escolhido foi do tipo PI para se obter erro zero em regime para entrada degrau. Seguem os gráficos buscando os seguintes requisitos para essa entrada:

- $t_s (98\%) < 3 \text{ s}$
- $M_p < 5 \%$

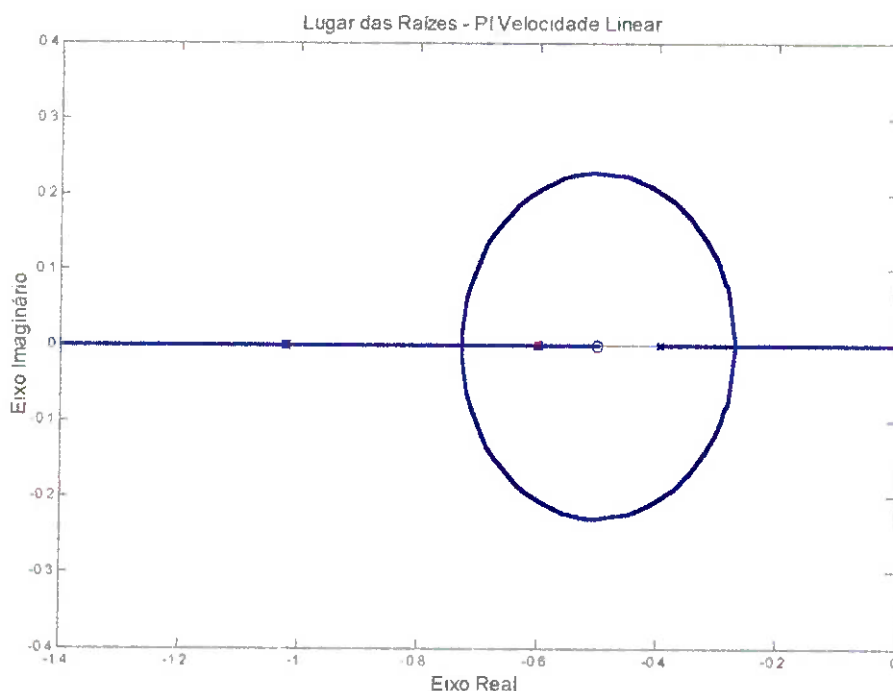


Figura 10.21 - Lugar das raízes para velocidade linear em malha fechada para um controlador do tipo PI

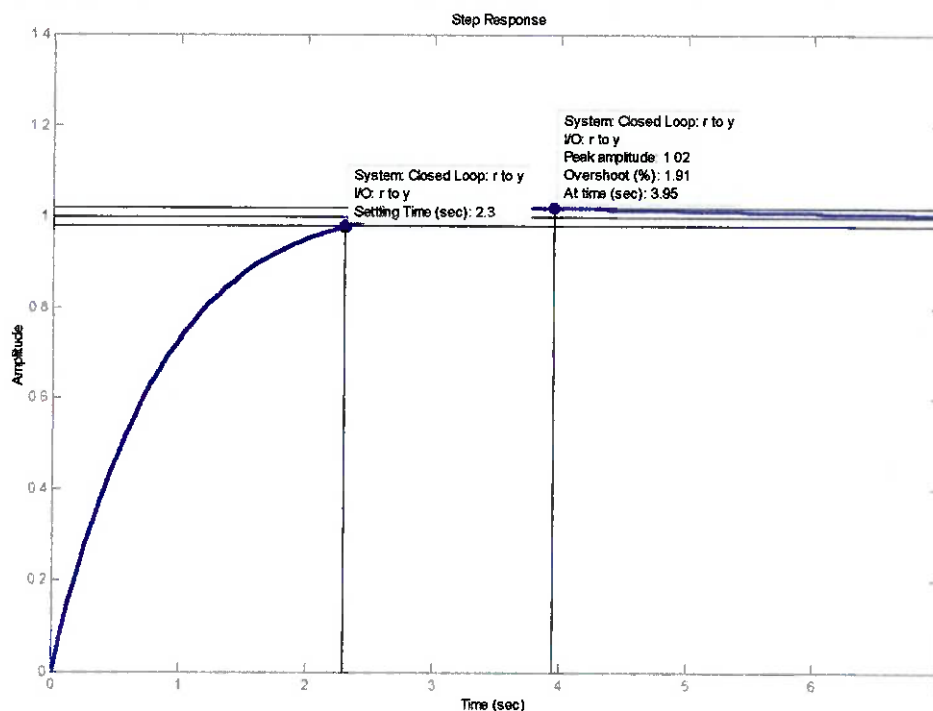


Figura 10.22 - Resposta a degrau unitário para velocidade linear em malha fechada para um controlador do tipo PI

O controlador projetado atende as especificações ficando bem abaixo dos limites máximos estabelecidos. O sobressinal máximo ficou em 1,91 % e o tempo de assentamento em 2,3 s.

A função de transferência do controlador fica:

$$G_c = 62 \cdot \frac{(s + 0,5)}{s}$$

Equação 10.13

10.8 Módulo Gerador de Referência

O módulo gerador de referência está de certo modo atrelado ao módulo controlador e tem seu funcionamento alterado segundo o modo de operação, como pode ser observado em seu diagrama de implementação.

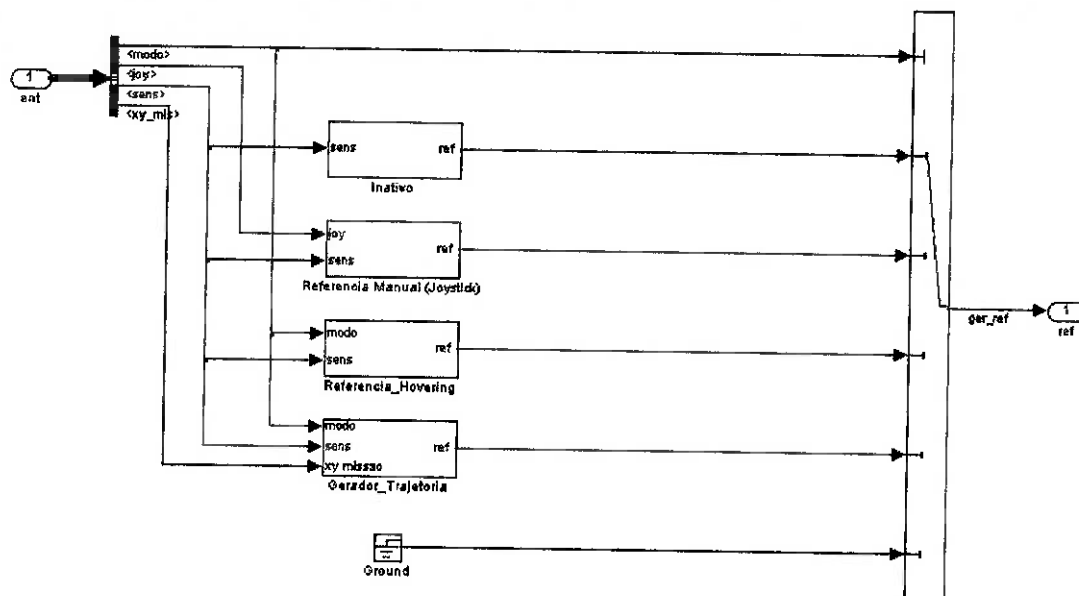


Figura 10.23 - Diagrama do módulo gerador de referência

Para funcionamento em modo inativo, deseja-se que não sejam enviados sinais de atuação para os propulsores. Uma forma de se obter isso é disponibilizando como referência ao controlador o próprio sinal lido dos sensores. Dessa forma o erro gerado é zero, o controlador acredita ter cumprido sua tarefa e nenhum sinal é enviado ao alocador de propulsão e propulsores.

Quando em modo manual, pelo próprio projeto do controlador, as referências de posição dadas são as próprias leituras dos sensores mas como referência de velocidade são dados os sinais obtidos pela movimentação do *joystick*.

Para funcionar em modo de *hovering*, as referências de posição devem ser constantes e as de velocidade devem ser constantes e iguais a zero. As constantes devem ser obtidas da última leitura dos sensores, fazendo com que o veículo pare naquele ponto.

Ao contrário do modo manual, no modo automático utiliza-se as referências de posição, ou uma trajetória, para conduzir o veículo. O sub-módulo gerador de trajetória da como referência um sinal de posição com três regiões características em função do tempo: aceleração, velocidade constante e desaceleração. A aceleração e

desaceleração são realizadas a acelerações constantes e durante um tempo certo para garantir continuidade de velocidade com a região de velocidade constante.

A saída do gerador de trajetória deve ser capaz de fornecer referências com as propriedades acima para movimentação em linha reta do veículo sobre um plano horizontal. Dessa forma as referências para os dois graus de liberdade devem ser sincronizadas. Isso é realizado da seguinte maneira: parte-se do grau de liberdade em que deve ser realizado o maior deslocamento e calcula-se a referência para esse grau de liberdade. Em seguida assume-se que a referência para o outro grau de liberdade deve ser proporcional à referência do grau de liberdade já calculado, obtendo-se a segunda referência sincronizada com a primeira. Para todos os outros graus de liberdade as referências são mantidas constantes, com os últimos valores de sensores lidos no instante imediatamente anterior ao envio do veículo em missão.

10.9 Módulo de Alocação de Propulsão

A disponibilidade de um maior número de atuadores que o número de graus de liberdade permite que um par de força e momento resultantes possa ser obtido com diversas combinações de empuxo em cada propulsor (item 8.5). De forma geral a implementação no Simulink do algoritmo de alocação de propulsão não apresenta grandes dificuldades.

Matlab para utilização com o xPC *target* (*target* para PC padrão do Real Time Workshop).

Para a configuração do *driver* foram utilizados os seguintes parâmetros:

- Base address: 0x300
- Channel: 1 a 8 (um para cada propulsor)
- Range: -5 a +5 V

O *driver* consiste de um bloco do Simulink dotado de oito entradas representando cada propulsor. O sinal pode variar numa faixa entre +5 e -5 V, sendo que +5 V aciona a motor em potência máxima e -5 V aciona o motor em reversão máxima.

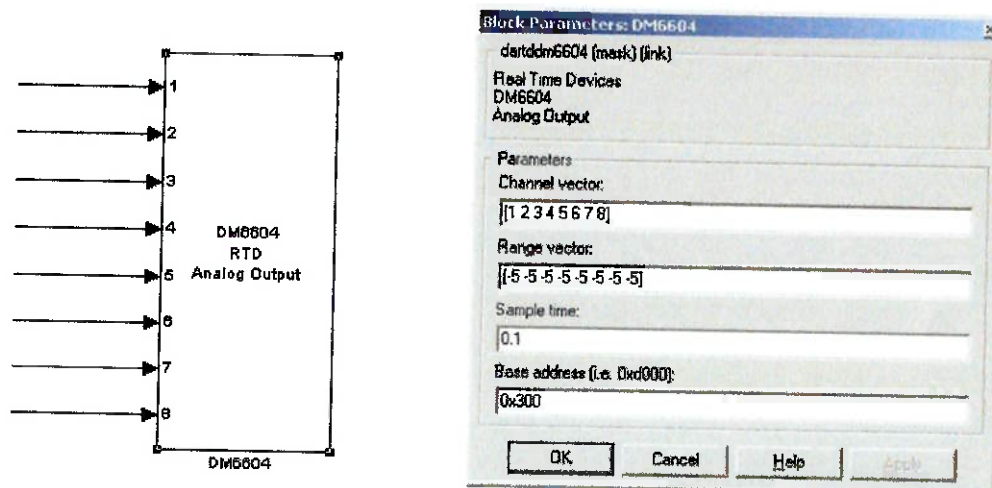


Figura 10.25 - Detalhe do driver dos propulsores e dos parâmetros

A configuração dos parâmetros do bloco do driver dos propulsores utilizada nos testes está ilustrada na figura anterior.

O parâmetro "Channel vector" representa o canal de hardware ao qual cada entrada do bloco está ligada, sendo que neste caso optou-se por ligar as entradas em sequência (porta 1 ligada ao canal 1, porta 2 ligada ao canal 2, ...).

O parâmetro "Range vector" representa a faixa de tensão de cada porta e como se trabalha com a mesma faixa para todas as portas o mesmo número se repete para todos os propulsores. O número -5 quer dizer que tensão vai de -5 a 5 V.

O parâmetro “Sample time” quantifica o período tempo para atualização do valor de saída no driver dos propulsores. Para o teste julgou-se suficiente um período de 0,1 segundos.

O último parâmetro “Base address” representa o endereço base para comunicação com os propulsores. Na caso 0x300.

Para mais detalhes sobre os parâmetros pode-se consultar o manual do Matlab.

10.11 Driver dos Sensores

Como o Matlab não possui um driver específico para operação dos DVAs do PC104 optou-se pela utilização dos inclinômetros da bússola que podem ser lidos pela porta serial. O Matlab já possui uma biblioteca própria para comunicação via serial.

O sensor utilizado é o TCM2-50 da Precision Navigation que possui uma bússola e dois inclinômetros. A comunicação via porta serial é baseada em strings. Ao se passar um comando de inicialização o sensor começa a mandar os dados numa taxa pré-programada.

Os dados são recebidos no formato de uma string contendo os dados de direção da bússola e ângulo dos inclinômetros. A string obedece ao seguinte formato:

`$C<compass>P<pitch>R<roll>E<errorcode>*checksum<cr>`

sendo que:

- `<compass>` : representa o valor medido pela bússola em graus;
- `<pitch>` : representa o ângulo pitch medido pelo inclinômetro;
- `<roll>` : representa o ângulo de roll medido pelo inclinômetro;
- `<errorcode>` : código de erro;
- `<checksum>` : valor de checagem das medidas.

Exemplo de medição:

`$C328.3P28.4R-12.4E001*checksum<cr>`

- bússola: 238,3 graus;

- pitch: 28,4 graus;
- roll: -12,4 graus.

Para a configuração da porta serial (RS232) foram utilizados os seguintes parâmetros:

- dados de 8 bits;
- 1 start bit;
- 1 stop bit;
- sem paridade
- baud rate: 9600.

Para que o sensor comece a enviar os dados ele necessita de um comando de inicialização (“go\n”). Assim que o sensor interpreta a mensagem ele começa a enviar os dados a uma taxa constante.

A biblioteca do *Simulink* para o xPC target possui um *driver* para comunicação serial que é composta por dois blocos: um bloco de configuração e outro de comunicação com as devidas portas de entrada ou saída.

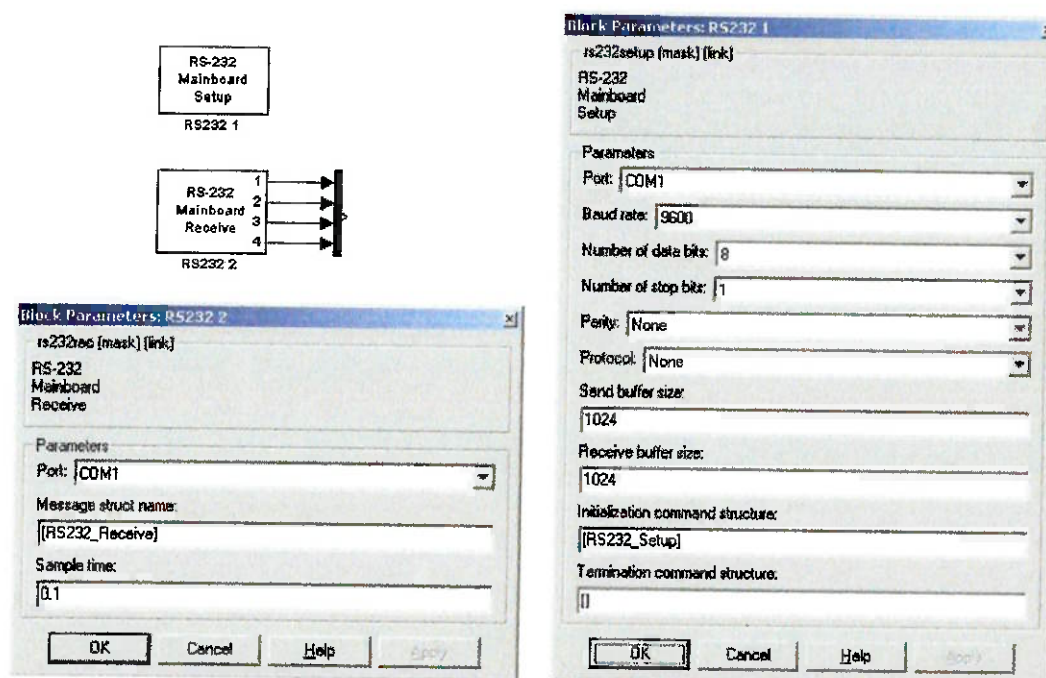


Figura 10.26 - Detalhe do driver dos sensores e dos parâmetros

O parâmetro “Message struct name” deve conter o nome de uma *struct* que deve estar presente no *workspace* do *Matlab*. Esta *struct* possui o formato da string de comunicação e ela foi configurada como a seguir:

```
RS232_Receive(1).RecData = '$C%fP%fR%f*%x\n';  
RS232_Receive(1).OutputPorts = [1 2 3 4];  
RS232_Receive(1).Timeout = 0.01;  
RS232_Receive(1).EOM = 1;
```

As portas de saída do bloco fornecem, nesta ordem: *yaw*, *pitch*, *row* e código de erro. Os ângulos são em graus sendo o *yaw* medido em relação ao norte magnético e o *pitch* e *row* medidos em relação à horizontal. Para mais detalhes sobre a configuração da porta serial utilizando esses blocos do *simulink* pode-se recorrer ao manual do *Matlab*.

11 RESULTADOS

Os testes com o veículo em funcionamento real se limitaram a testes secos devido à impossibilidade de uma montagem que não colocasse em risco os equipamentos eletrônicos, além de não estarem disponíveis, no momento, meios para o sensoriamento da posição linear do veículo. Mesmo assim constatou-se o correto funcionamento do sistema desde do carregamento do controle a ser embarcado até as funções de movimentação manual do veículo, demonstrando que a integração dos equipamentos foi completa. Movimentando-se os sensores verificou-se que o controlador procurava manter o veículo em posição horizontal e que a compensação da força peso era obtida com a melhor alocação de força em cada propulsor em função da inclinação do veículo.

Para efeito de documentação do bom funcionamento dos módulos de controle, geração de referência e alocação de propulsão foram realizados testes no ambiente de simulação. Os resultados são apresentados abaixo.

11.1 Estabilização de Inclinação

Num primeiro teste foi verificada a capacidade do veículo de se manter estável em posição horizontal. Para tanto se utilizou o modo manual para inclinar o veículo e, em seguida, ativou-se o modo de *hovering*.

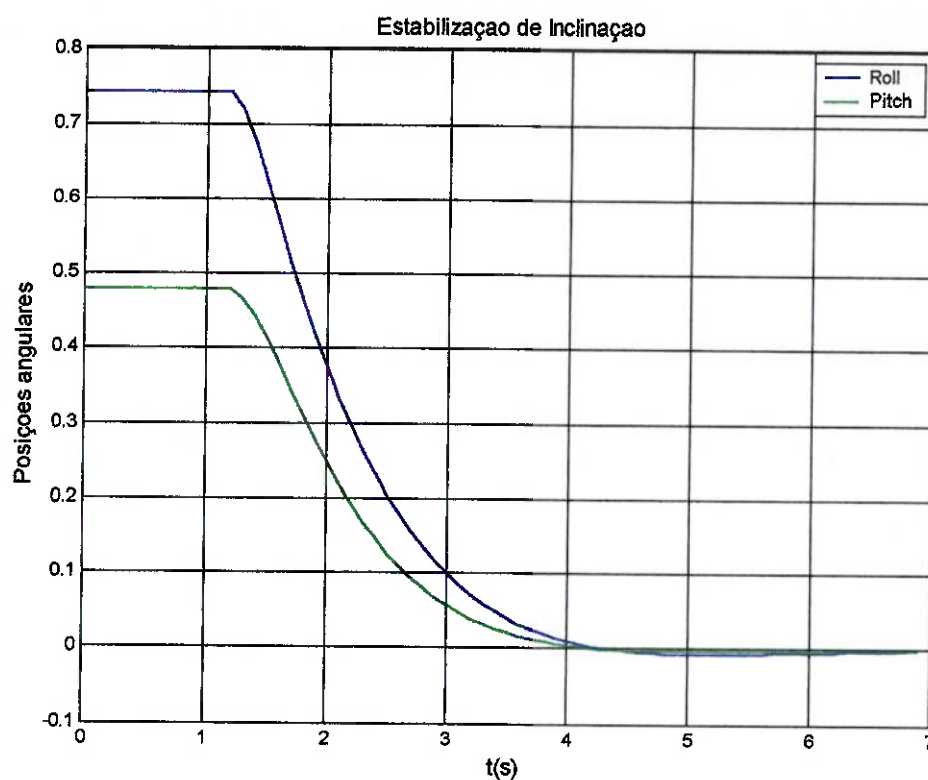


Figura 11.1 - Estabilização de Inclinação entrando em Hovering

Nota-se que por volta do instante 1,25s é requisitado o modo de hovering ao sistema, que leva a estabilização das inclinações em roll e pitch do submarino à referência zero. A estabilização ocorre em pouco menos de 3 segundos, como desejado.

11.2 Resposta a comandos de manuais de movimentação

Os movimentos manuais de movimentação (dados pelo joystick) são utilizados como referência de velocidade para o controlador, que calcula a resultante de forças requeridas e que finalmente são obtidas com a alocação de propulsão. Para movimentos de exemplo obteve-se:

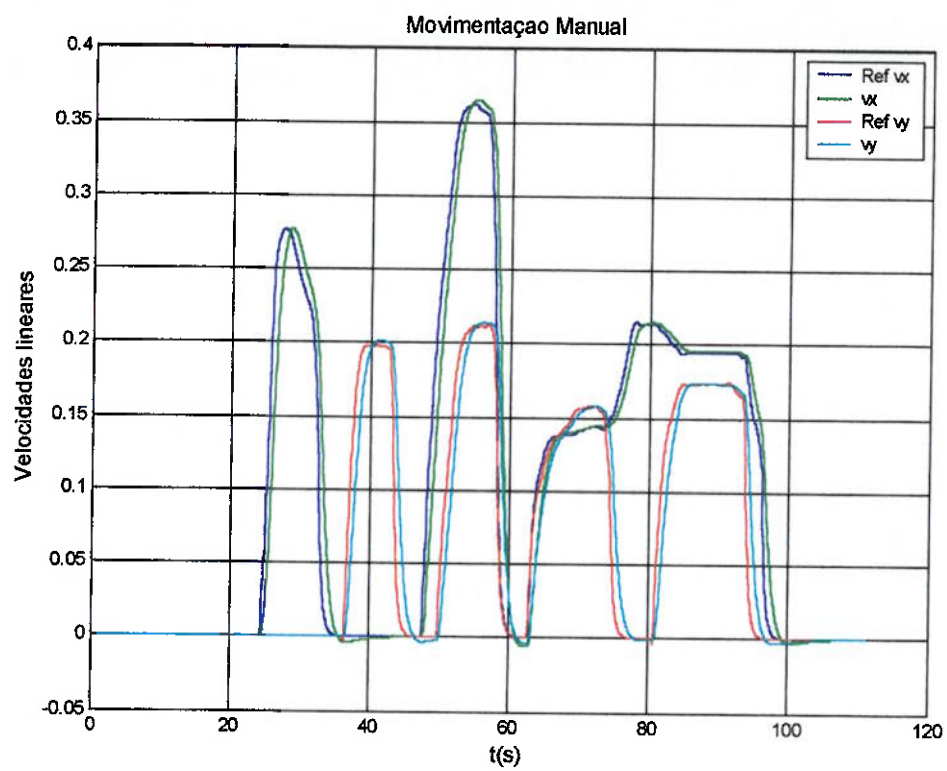


Figura 11.2 - Referência e resposta de velocidades lineares em modo manual

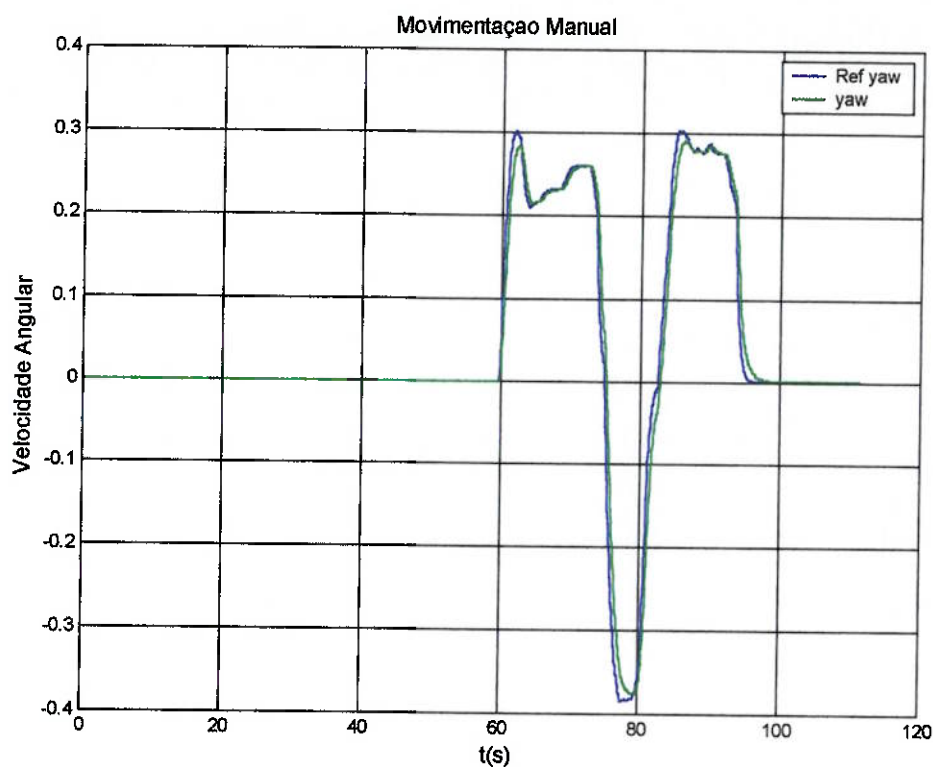


Figura 11.3 - Referência e resposta de velocidades angulares em modo manual

Observa-se que a resposta do veículo é bastante satisfatória, permanecendo sempre próxima a velocidade de referência. Lembra-se que tanto a referência como a velocidade do veículo são dadas no referencial solidário a ele. Dessa forma, caso se queira fazer a reconstrução da trajetória do veículo deve-se fazer a correta transformação de coordenadas.

11.3 Cumprimento de Missão

11.3.1 Missão Utilizando Oito Propulsores

Para a verificação de movimentação do veículo em modo automático foram realizados alguns testes. Esse primeiro teste conta com a capacidade de propulsão de todos os oito propulsores.

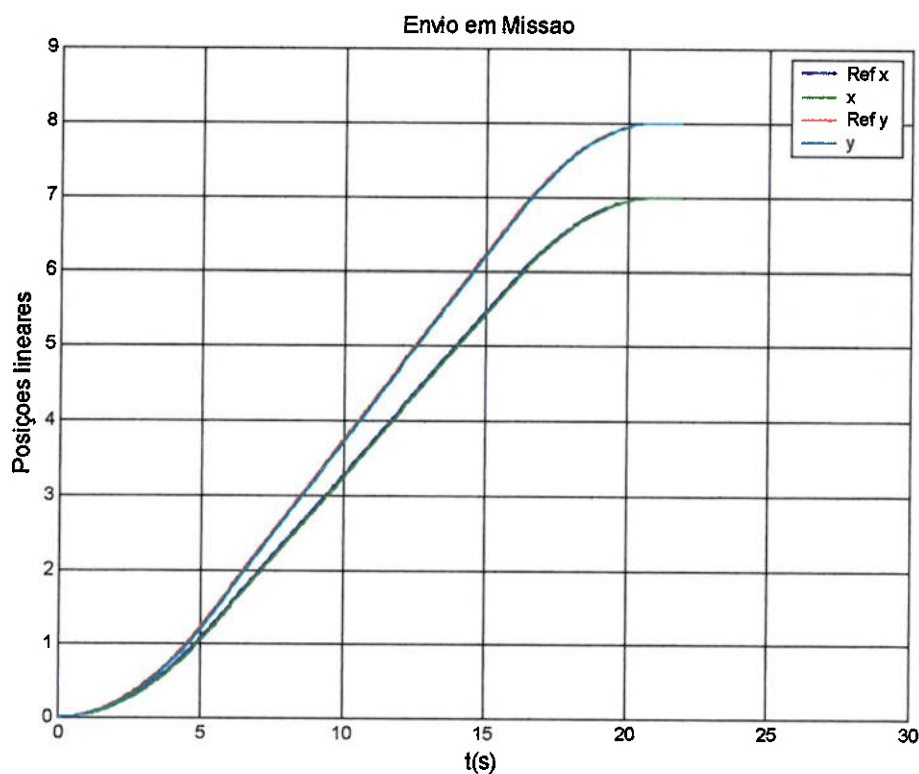


Figura 11.4 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores)

Do gráfico percebe-se claramente que a posição real do veículo seguiu de perto as referências dadas. O movimento, como projetado, foi realizado em três etapas: uma de aceleração (do instante 0 até o instante 4s), uma de velocidade constante ($4 < t < 17s$) e uma de desaceleração ($t > 17s$). Os movimentos em cada grau de liberdade foram sincronizados para que se obtivesse o efeito final de deslocamento em linha reta na composição de movimentos, como pode ser verificado no gráfico abaixo:

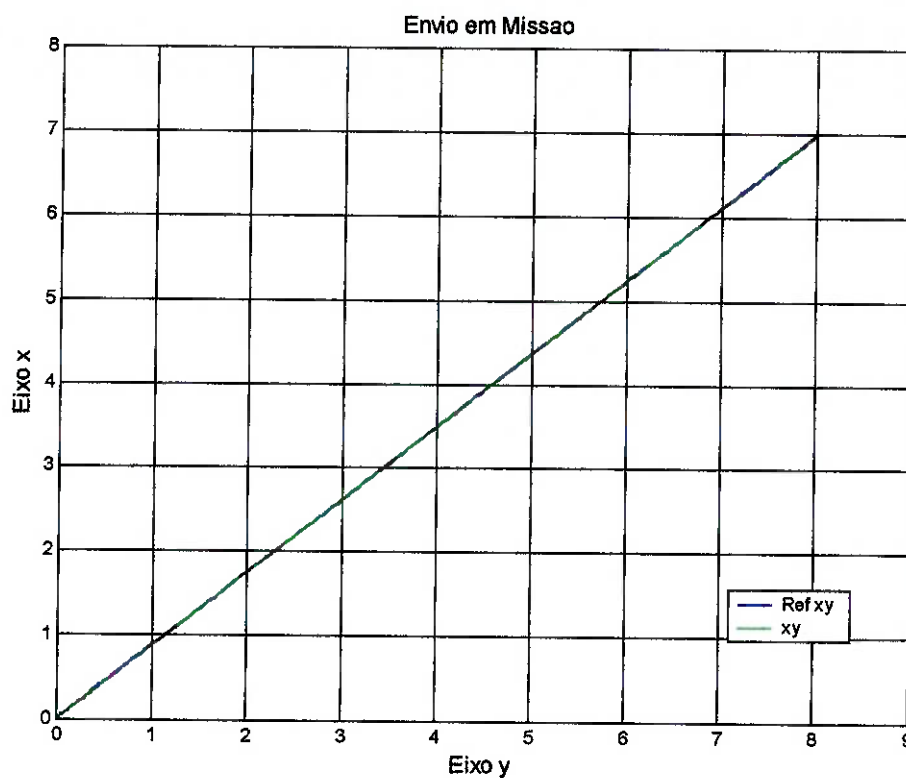


Figura 11.5 - Referência e resposta de posições lineares em missão (oito propulsores)

Novamente fica evidenciado a estreita relação entre a referência e a posição real do veículo, obtida com o controlador do tipo PI com ganhos extremamente elevados.

11.3.2 Missão com oito propulsores e compensação de correnteza

Simula-se agora a situação em que uma correnteza tenderia a levar o veículo a deriva, mas lhe é dada uma missão.

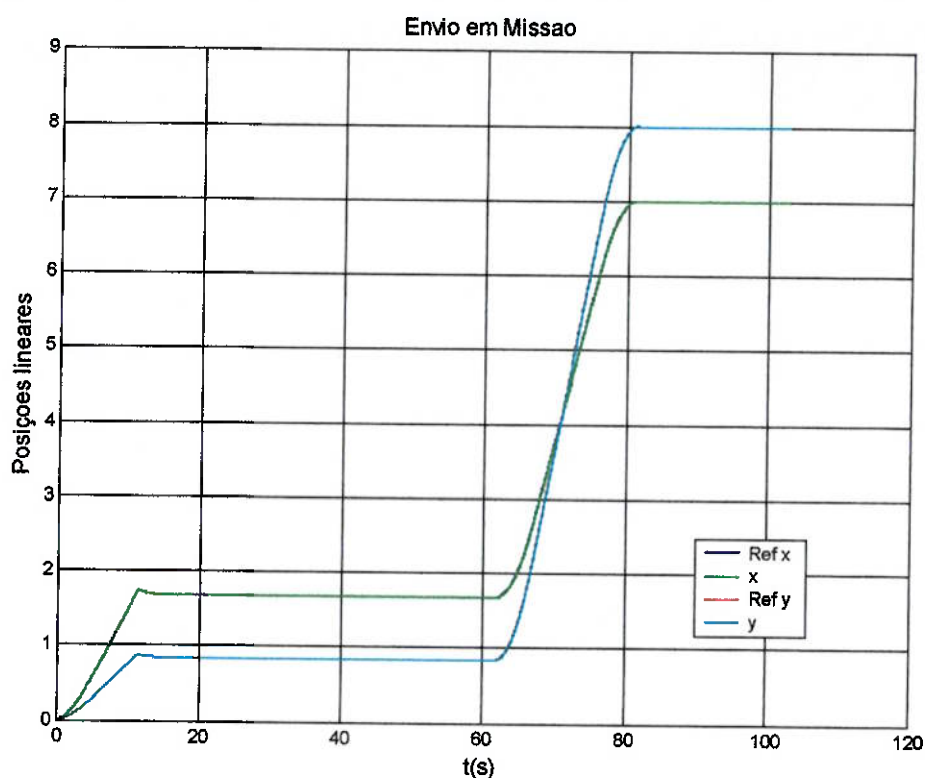


Figura 11.6 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores e compensação de correnteza)

Novamente os sinais de referência e posição se confundem. No início, em modo inativo, a correnteza conduz o veículo para a posição (1,7 ; 0,8), quando é dado o pedido de modo manual e em seguida de modo hovering. Percebe-se que durante esse tempo a posição do veículo permanece constante. Em seguida é dada a missão de busca do ponto (7 ; 8) que é cumprida em aproximadamente 18 segundos. Abaixo visualiza-se a trajetória realizada pelo veículo.



Figura 11.7 - Referência e resposta de posições lineares em função do tempo em missão (oito propulsores e compensação de correnteza)

Como dito anteriormente, o veículo é movido pela correnteza até o pedido de estabilização e depois é enviado em missão.

11.3.3 Missão com sete propulsores e compensação de correnteza

Devido à redundância de propulsão instalada no veículo pode-se obter qualquer movimento em qualquer grau de liberdade mesmo na falta de um dos propulsores. Dessa forma realizou-se o teste mais rígido que pode ser aplicado ao sistema implementado: o envio do veículo em missão, utilizando-se sete propulsores.

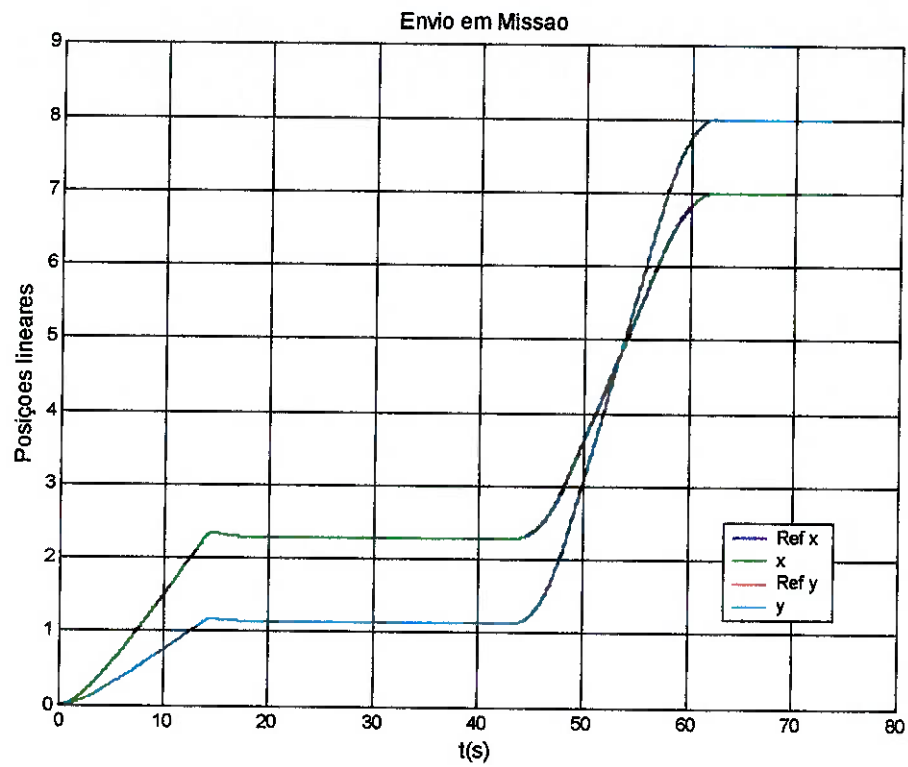


Figura 11.8 - Referência e resposta de posições lineares em função do tempo em missão (sete propulsores e compensação de correnteza)

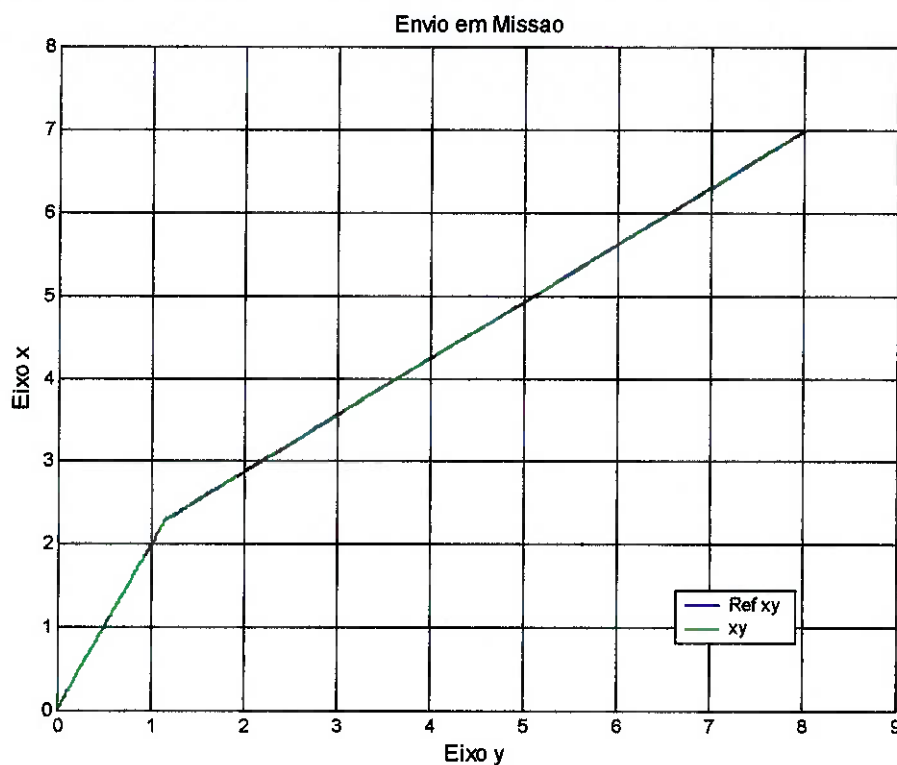


Figura 11.9 - Referência e resposta de posições lineares em função do tempo em missão (sete propulsores e compensação de correnteza)

Como esperado, a capacidade de realocação de propulsão faz com que a resultante de forças exigidas pelo controlador sejam corretamente obtidas, obtendo-se um resultado bastante similar ao caso de missão com oito propulsores e correnteza, apenas com a mudança nos instantes de comutação entre modos (inativo, manual, hovering, missão).

12 CONCLUSÃO

O estudo realizado deixa claro que existe uma forte tendência mundial em utilização de veículos submarinos não tripulados em tarefas de instalação e manutenção de estruturas e dispositivos submarinos como os utilizados em plataformas de extração de petróleo e gás. Além disso, percebe-se a necessidade de uma arquitetura de controle para que seja possível a utilização do robô com grande eficácia e robustez, ou seja, capacidade de locomoção com velocidades de até 1m/s, autonomia de 20 minutos no local de trabalho mais 40 minutos para locomoção geral e manobrabilidade com precisão da ordem de centímetros.

Utilizando a teoria de arquitetura de controle formulou-se uma possível estrutura de software em módulos que têm sua funcionalidade própria que garantem o bom funcionamento de cada recurso do sistema. No computador de operação foram definidos os módulos de interface com o usuário, gerenciador de fluxo de informações e comunicação com o UUV. Já no veículo foram levantados os seguintes módulos: comunicação com a embarcação de operação, *drivers* de sensores, *drivers* de atuadores, gerador de referência, controlador, alocador de propulsão, gerenciador de fluxo de informações e supervisor. Essa divisão permite uma boa estruturação lógica para que o sistema possa ser desenvolvido de forma organizada e que se atinja bom funcionamento em situações normais de operação e em casos de ocorrência de anomalias que devem ser detectadas e contornadas.

A fim de implementar os módulos foram estudados os sistemas operacionais VxWorks e um Kernel próprio do MatLab. A grande vantagem do VxWorks é de ele ser um sistema concebido para funcionar com sistemas em tempo real, como o UUV, e pode ser utilizado em conjunto com um programa de desenvolvimento gráfico, o Constellation. Mas infelizmente não foi possível a realização de testes nesse sistema devido a problemas na sua instalação no veículo. Já o MatLab não oferece todos os recursos oferecidos pelo VxWorks mas funcionou satisfatoriamente nos testes realizados com o sistema executado em tempo real no computador embarcado, demonstrando a integração de todos os dispositivos.

Os testes desenvolvidos no ambiente de simulação verificaram a capacidade do veículo em estabilizar sua inclinação, compensar correntezas, seguir movimentos

dados por joystick e cumprir missão de busca de um determinado ponto em trajetória retilínea horizontal. Dessa forma obteve-se um sistema de controle funcional que deve ser testado efetivamente em testes reais de utilização para que sejam terminantemente validadas as hipóteses aqui adotadas e para que se possa iterativamente aprimorar o sistema até que se torne uma ferramenta confiável para utilização em sistemas de produção.

REFERÊNCIAS BIBLIOGRÁFICAS

FERREIRA, G. A. N. **Desenvolvimento de uma Arquitetura de Controle Baseada em Objetos para um Robô Móvel Aquático**. 2003. Qualificação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2003.

SOARES, F. J. A. **Projeto e Implementação de uma Plataforma de Testes Aplicada ao Desenvolvimento de Veículos Submarinos não Tripulados**. 2002. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2002.

FOSSSEN, T. I. **Guidance and Control of Ocean Vehicles**. John Wiley & Sons, 1994.

SOUZA, E. C. **Modelagem e Controle de Veículos Submarinos Não Tripulados**. 2003. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2003.

AVILA, J. P. J. **Identificação dos Coeficientes Hidrodinâmicos de um Veículo Submarino Semi-autônomo**. 2003. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2003.

INDIVERI, G. **Modeling and Identification of Underwater Robotic Systems**. Genova: Department of Communications, Computer and Systems Science, 1998.

BARBARINI, L. H. M. **Integração Mecânica, Sensoriamento e Obtenção dos Parâmetros Mecânicos do Sistema Robótico Submarino**. São Paulo: Escola Politécnica, Universidade de São Paulo, 2003 (Relatório Fapesp, 02/00581-9).

FALTINSEN, O. M. **Sea Loads on Ships and Offshore Structures**. Cambridge University Press, 1990.

CAYFORD, J. E. **Underwater Work**. Cornell Maritime Press, 1966

LIMA, F. V. F. **Desenvolvimento de um Sistema Acústico de Posicionamento de Submarino**. 2002. Qualificação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2002.

REAL-TIME INNOVATIONS. Manuais do Software Constellation.

WIND RIVER. Manuais do Software VxWorks.

APPLIED GEOMECHANICS. Manual do Usuário dos Inclínômetros modelo 758

PNI CORPORATION. Manual do Usuário da Bússola modelo TCM2-50

MARINE TECHNOLOGY SOCIETY. Materiais disponíveis em
<<http://www.rov.org>>.

JAMSTEC. Materiais disponíveis em <<http://www.jamstec.go.jp/jamstec-e/index-e.html>>.

STANFORD UNIVERSITY. Material sobre o veículo OTTER disponível em
<http://sunvalley.stanford.edu/projects/underwater_robots>.

DEPARTMENT OF OCEAN ENGINEERING AT FLORIDA ATLANTIC UNIVERSITY. Material sobre o veículo Ocean Voyager II disponível em
<<http://www.oe.fau.edu/AMS/auv.html>>.

AUV LAB AT MASSACHUSETTS INSTITUTE OF TECHNOLOGY. Material sobre o veículo Odyssey disponível em <<http://auvlab.mit.edu>>

THE AUSTRALIAN NATIONAL UNIVERSITY. Material sobre o veículo Kambara disponível em <<http://www.syseng.anu.edu.au/rsl/sub>>.

UNDERWATER ROBOTICS & APPLICATION LABORATORY, UNIVERSITY OF TOKYO. Material sobre o veículo PTEROA disponível em
<<http://underwater.iis.u-tokyo.ac.jp>>.