

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

VITOR DE OLIVEIRA VARGAS

Simulação dinâmica da propulsão e
controle de trajetória de um navio patrulha de
pequeno porte em SIMULINK

São Paulo
2010

VITOR DE OLIVEIRA VARGAS

Simulação dinâmica da propulsão e controle de trajetória
de um navio patrulha de pequeno porte em SIMULINK

Trabalho de conclusão de curso
apresentado à Escola Politécnica da
Universidade de São Paulo.

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Eduardo Aoun Tannuri

São Paulo
2010

Ficha Catalográfica

Vargas, Vitor de Oliveira

Simulação dinâmica da propulsão e controle de trajetória de um navio patrulha de pequeno porte em SIMULINK/ Vitor de Oliveira Vargas. – São Paulo, 2010.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e Sistemas Mecânicos.

1. Controle de Trajetória 2. Modelo de Manobra 3. Controle Linear I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e Sistemas Mecânicos II.t

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

AGRADECIMENTOS

A Deus, porque nada acontece se for de sua vontade. Pois, também, ajudou-me nos momentos mais difíceis do curso. Nessas horas, que se pode sentir a presença divina.

A minha noiva e cúmplice Jamille, que sempre foi a minha base, minha motivação e o meu refúgio de paz, de onde eu sempre recarregava minhas forças para seguir à frente.

Aos meus amigos de turma André Barbosa e Jeferson Oliveira, que foram meus companheiros e ombro amigo nessa trajetória.

Ao meu orientador, Professor Tannuri, sempre disposto a ajudar e sempre rigoroso, a cobrar resultados para manter um bom nível do trabalho.

Ao Centro de Coordenação e Estudos da Marinha em São Paulo, que esteve sempre pronto a prestar qualquer auxílio que eu precisa-se.

RESUMO

VARGAS, V. O. **Simulação dinâmica da propulsão e controle de trajetória de um navio patrulha de pequeno porte em SIMULINK.** 2010. 114 f. Trabalho de conclusão de curso – Escola Politécnica, Universidade de São Paulo, São Paulo, 2010.

Esse trabalho tem o propósito de desenvolver um piloto automático de um navio patrulha de pequeno porte projetado para a Marinha do Brasil. De posse de suas características e requisitos, a tarefa do projeto consiste em simular a planta propulsora, desde os motores de combustão interna, passando pelos acoplamentos fluidos, engrenagem redutora, até os hélices. Deverá constar na simulação toda a inércia desse conjunto, onde os motores fornecerão torque ao sistema, que gerará uma aceleração ao navio, necessária para o controle da trajetória. Para a realização desse controle serão utilizados os estudos de controle clássico e de derivadas hidrodinâmicas, no qual a entrada principal é a velocidade fornecida pela propulsão do navio, que irá interagir com o leme, o que irá possibilitar a manutenção da trajetória pretendida. Por fim, o sistema de controle deverá ser capaz de ser manter estável, mesmo sob distúrbios como, por exemplo, vento.

Palavras-chave: piloto automático, navio patrulha, planta propulsora, controle de trajetória.

ABSTRACT

VARGAS, V. O. Dynamic simulation of propulsion and trajectory control of a small patrol vessel in SIMULINK. 2010. 114 f. Trabalho de conclusão de curso – Escola Politécnica, Universidade de São Paulo, São Paulo, 2010.

This work aims to develop an autopilot for a small patrol vessel designed for the Brazilian Navy. Since the ship has its characteristics and its requirements, the design task is to simulate the propulsion plant, since the internal combustion engine, through the fluid couplings, gear reducer, to the propellers. All the inertia of the set must be included in the simulation, where motors provide torque to the system that generates acceleration for the ship, which is needed to control the trajectory. Classical control studies and hydrodynamic derivatives will be used to achieve this control, in which the main entrance is the speed of the ship provided by the propulsion, which will interact with the rudder, which will enable the maintenance of the desired trajectory. Finally, the control system should be capable of maintaining stable, even under disturbances such as, for example, wind.

Keywords: autopilot, patrol vessel, propulsion plant, trajectory control.

LISTA DE FIGURAS

Figura 1.1 - Arranjo do NPa.....	16
Figura 1.2- Direção do navio em relação ao sistema OXYZ.	18
Figura 1.3 - Arranjo da planta propulsora.....	19
Figura 1.4 - Malha de controle.....	20
Figura 2.1 - Tela principal da simulação.....	22
Figura 2.2 - Sistema principal da simulação.....	23
Figura 2.3 - Ampliação de parte do sistema principal de controle (controle)... ..	24
Figura 2.4 - Ampliação de parte do sistema principal de controle (segurança).25	
Figura 2.5 - Ampliação de parte do sistema principal de controle (dinâmica). .	26
Figura 2.6 - Ampliação de parte do sistema principal de controle (desenho)... ..	27
Figura 3.1 - Subsistema CONTROLE DE PROPULSÃO.	28
Figura 3.2 - Ampliação do subsistema Controle da Propulsão.....	29
Figura 3.3 - Rotação em função de PCL.	30
Figura 3.4 - Malha auxiliar de rotação.	31
Figura 3.5 - Subsistema "Controle MCP".	32
Figura 3.6 - Ângulo de passo em função de PCL.....	33
Figura 3.7 - Subsistema "PLANTA DE PROPULSÃO".....	34
Figura 3.8 - Subsistema "PROP MCP".....	36
Figura 3.9 - Subsistema "dinâmica do eixo".....	37
Figura 3.10 - Subsistema "ACOPLAMENTO FLUIDO".	39
Figura 3.11 - Subsistema "dinâmica da redutora (após AF)".....	40
Figura 3.12 - Subsistema "chaveamento entre modos".	42
Figura 3.13 - Subsistema "modelo lógico das embreagens".	43

Figura 3.14 - Subsistema "atuador de passo".	45
Figura 3.15 - P/D em função do ângulo de passo.	45
Figura 4.1 – Resistência ao avanço em função da velocidade.	47
Figura 4.2 – Rotação e K_T em função da velocidade.	48
Figura 4.3 – Rotação e K_Q em função da velocidade.	48
Figura 4.4 – Eficiência relativa rotativa em função da velocidade e do passo.	49
Figura 4.5 – Passo em função da velocidade do navio.	50
Figura 4.6 – Potência em função da rotação do motor.	51
Figura 5.1 - Ângulos entre o Navio e o vento	58
Figura 6.1 – Desempenho do motor.	60
Figura 6.2 - Curvas de giro.	61
Figura 6.3 – Velocidades com 15° de leme (BB).	62
Figura 6.4 – Roll com leme de 15° BB.	62
Figura 6.5 - Característica da manobra de zig-zag	63
Figura 6.6 - Deslocamento em zig-zag.	63
Figura 6.7 - Resposta do navio em "zig-zag"	64
Figura 6.8 – Parada em emergência (velocidade).	65
Figura 6.9 – Parada em emergência (distância).	66
Figura 7.1 – Resposta dinâmica da planta.	67
Figura 7.2 – Máquina do leme.	68
Figura 7.3 – Diagrama de blocos do controlador PD.	69
Figura 7.4 – Alocação de pólos.	70
Figura 7.5 – Resposta dinâmica em função de ω_n .	71
Figura 7.6 - Estabilização da planta (controlador PD).	72
Figura 7.7 – Inclusão de K_i .	73
Figura 7.8 – Piloto automático.	74

Figura 8.1 – Deslocamento para 90°.....	75
Figura 8.2 - Variação da proa para o rumo 90°.....	76
Figura 8.3 – Ângulo do leme.....	77
Figura 8.4 - Roll durante estabilização.....	77
Figura 8.5 – Controle com distúrbio (Vento 0° - 10 nós).....	78
Figura 8.6 – Deslocamento para o norte (0°).....	79
Figura 8.7 – Velocidades de <i>surge</i> e <i>sway</i>	79
Figura 8.8 – Velocidade de <i>sway</i>	80
Figura 8.9 – Variação do rumo (pane em um eixo).....	80
Figura 8.10 – Compensação do leme.....	81

LISTA DE TABELAS

Tabela 1.1 – Navio patrulha de pequeno porte.	17
Tabela 1.2 - Notações usadas.....	18
Tabela 3.1 - Momentos de inércia da planta da propulsão.....	35
Tabela 3.2 - Lógica das engrenagens SSS 124.	44
Tabela 5.1 - Coeficientes e forças adotados no modelo	53
Tabela 5.2 - Tabela de normalização (Prime-System I)	54
Tabela 6.1 – Velocidade do navio em função de PCL.....	59

LISTA DE ABREVIATURAS

GDL	<i>Graus de Liberdade</i>
MCP	<i>Motor de Combustão Principal</i>
HPC	<i>Hélice de Passo Controlado</i>
PCL	<i>Power Control Level</i>
AF	<i>Acoplamento Fluido</i>
RRF	<i>Razão de Redução Fornecida</i>
PD	<i>Proporcional e Derivativo</i>
PID	<i>Proporcional, Integrativo e Derivativo</i>
BE	<i>Boreste</i>
BB	<i>Bombordo</i>
NPa	<i>Navio Patrulha</i>

LISTA DE SÍMBOLOS

ψ_d	Rumo desejado.
δ	Ângulo de leme.
δ_d	Ângulo de leme desejado.
J_{pr_af}	Momento de inércia do primário do acoplamento fluido
N_m	Velocidade angular do eixo do MCP
Q_{MCP}	Torque fornecido pelo motor
Q_{pr_af}	Torque no primário do acoplamento fluido
Q_{pr_f}	Perdas por atrito antes do primário do acoplamento fluido
J_{se_af}	Momento de inércia do secundário do acoplamento fluido
N_h	Velocidade angular do hélice
N_{se_af}	Velocidade angular do secundário do acoplamento fluido
Q_{se_af}	Torque no secundário do acoplamento fluido
Q_h	Torque no propulsor
Q_{se_f}	Perdas por atrito na linha de eixo
I_x	Momento de inércia em relação ao eixo x
I_z	Momento de inércia em relação ao eixo z
I_{xy}	Produto de inércia no plano xy
I_{yz}	Produto de inércia em relação ao plano yz
ω_δ	Velocidade angular do leme (rad/s)
Z	Amortecimento do sistema
ω_n	Frequência natural do sistema

GLOSSÁRIO

Boca	Maior medida transversal do navio.
Boreste	Olhando da popa para a proa, o lado direito.
Bombordo	Olhando da popa para a proa, o lado esquerdo.
Calado	É a medida de altura desde a quilha até a altura d'água.
Coeficiente de bloco	É a razão entre o volume ocupado pelo navio e o volume do paralelepípedo formado pelo comprimento, boca e calado.
Comprimento	Medida longitudinal do navio.
Comprimento entre perpendiculares	Há perpendiculares transversais ao comprimento ao longo de todo o navio. E a medida entre a primeira perpendicular e a última é o comprimento entre perpendiculares.
Milha	1852 m.
Movimento de <i>roll</i>	Rotação em torno do eixo longitudinal do navio (balanço).
Movimento de <i>surge</i>	Translação longitudinal (avanço).
Movimento de <i>sway</i>	Translação lateral (deriva).
Movimento de <i>yaw</i>	Rotação no plano horizontal (guinada).
Nó	Milha/hora.
Passo do HPC	É a medida que a pá do hélice percorre a cada rotação do HPC.
Popa	Parte de trás do navio.
Proa	Parte da frente do navio.
SIMULINK	Toolbox pertencente ao Software Matlab.

Velocidade de cruzeiro

É a velocidade média que o navio foi projetado para navegar a maior parte do tempo.

Volume deslocado

É o volume de água que o navio desloca.

SUMÁRIO

1 Introdução.....	16
1.1 Notações utilizadas	17
1.2 Planta propulsora	19
1.3 Malha de controle	20
2 Controles principais do navio	21
3 Simulação da propulsão	28
3.1 CONTROLE DE PROPULSÃO	28
3.1.1 Subsistema “Controle MCP”	30
3.1.2 Subsistema “Controle do HPC”	33
3.2 PLANTA DE PROPULSÃO	33
3.2.1 Subsistema “PROP MCP”	35
3.2.2 Subsistema “chaveamento entre modos”	40
3.2.3 Subsistema “atuador de passo”	44
4 Resistência ao avanço	46
4.1 Ajuste do passo do HPC	47
5 Modelos Matemáticos	52
5.1 Dinâmica de navios	52
5.1.1 Equação de <i>surge</i>	52
5.2 Equações normalizadas	54
5.3 Derivadas hidrodinâmicas	56
5.4 Vento	57
5.5 Arquivo calc_derivadas.m	58
6 Desempenho do modelo.....	59
6.1 Motores	60
6.2 Desempenho da planta	61
6.2.1 Curva de giro	61
6.2.2 Manobra de zig-zag	63
6.2.3 Parada em emergência.....	64
7 Controle de trajetória	67
7.1 Resposta dinâmica da planta	67
7.2 Máquina do leme	68
7.3 Controle PD	68
7.3.1 Inclusão do termo integrativo	72
7.4 Arquivo NPa_data.m	74
8 Resultados	75
8.1 Piloto automático	75
8.1.1 Piloto automático com distúrbio	78
8.1.2 Navio funcionando com um único eixo	78
9 Conclusões	82
10 Bibliografia.....	83

Anexo A - calc_derivadas.m.....	84
Anexo B - calc_erro.m.....	89
Anexo C - convunit.m.....	90
Anexo D - ctr_proa.....	91
Anexo E - ctr_data.....	94
Anexo F - desenho_NPa.....	96
Anexo G - histerese_embreagens.m.....	103
Anexo H - hpc.m.....	104
Anexo I - Integracao_casco_helice.m.....	105
Anexo J - Integracao_casco_heliceAR.m.....	106
Anexo K - Integração_casco_heliceAV.m.....	112
Anexo L – Arquivos referentes aos botões do modelo.....	118
Anexo M – Arquivos referentes às manobras do modelo.....	119
Anexo N - NPa_data.m.....	121
Anexo O - pressao.m.....	123
Anexo P - programas_demandados_aproximados.m.....	124
Anexo Q - redutora.m.....	125
Anexo R - Vento.m.....	127

1 Introdução

O trabalho visa ser capaz de fornecer subsídios para que sejam desenvolvidos sistemas de piloto automático em navios que a Marinha do Brasil venha construir ou adquirir, ou quem sabe, desenvolvê-los em navios já incorporados à Força.

O controle de trajetória deve ser capaz de manter o Navio no rumo pretendido nas várias condições de máquinas (rotação nos eixos) e vento (distúrbios).

A Figura 1.1 mostra o arranjo do navio, onde mostra o posicionamento de compartimentos e equipamentos importantes como praça de máquinas, leme, hélice e canhões.

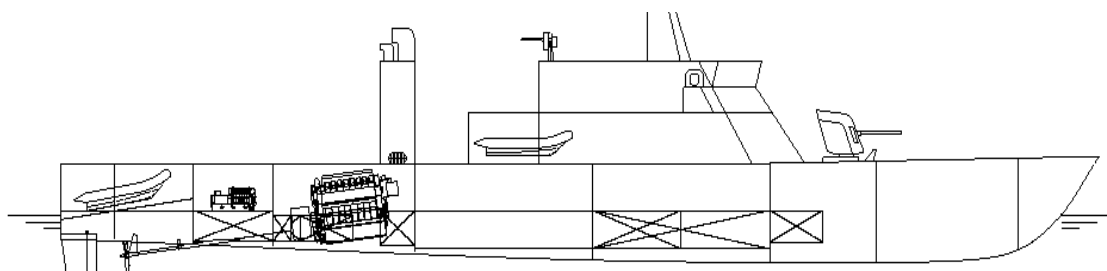


Figura 1.1 - Arranjo do NPa.

A Tabela 1.1 mostra as principais características do navio usado no modelo. Vale ressaltar, que ao ser dito que o navio patrulha projetado é um navio de pequeno porte, que essa afirmação refere-se ao fato de essa classe de navio abranger embarcações de até no máximo 500000 kg.

Tabela 1.1 – Navio patrulha de pequeno porte.

Característica	Valor
Comprimento total (L_{oa})	58,3 m
Comprimento entre perpendiculares (L)	56,5 m
Boca (B)	8,6 m
Calado de projeto (T_s)	2,28 m
Volume deslocado (∇)	476,42 m ³
Velocidade de cruzeiro (u_0)	12 nós
Massa deslocada (m)	488,81.10 ³ kg
Coeficiente de bloco (C_B)	0,43

1.1 Notações utilizadas

A Figura 1.2 exemplifica os eixos coordenados adotados e a Tabela 1.2 as notações. Para simplificar as equações, a origem dos eixos será fixa no centro de giro do navio; o eixo OX será fixo no sentido popa-proa do navio; OY no sentido BB-BE; e OZ de cima para baixo.

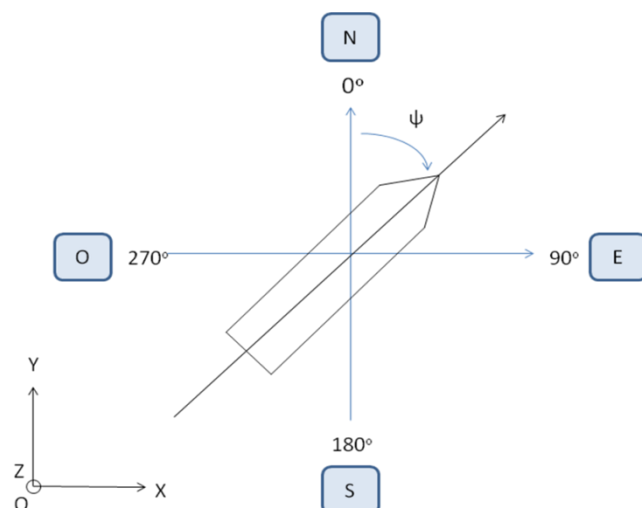


Figura 1.2- Direção do navio em relação ao sistema OXYZ.

Tabela 1.2 - Notações usadas.

Grau de Liberdade (GDL)		Forças e Momentos	Velocidades lineares e angulares	Deslocamentos lineares e angulares
1	Translação na direção x (surge) – positivo na direção OX	X	u	x
2	Translação na direção y (sway) – positivo na direção OY	Y	v	y
3	Rotação em torno de x (roll) – positiva no sentido horário	K	p	ϕ
4	Rotação em torno de z (yaw) – positiva no sentido horário	N	r	ψ

1.2 Planta propulsora

O arranjo da propulsão apresenta-se com duas linhas de eixos idênticas. Cada linha parte de um motor de combustão principal (MCP), seguido de um acoplamento fluido, uma embreagem sincronizadora, um sistema de engrenagens redutoras de rotação e finaliza com um hélice de passo controlado (HPC) como mostra a Figura 1.3.

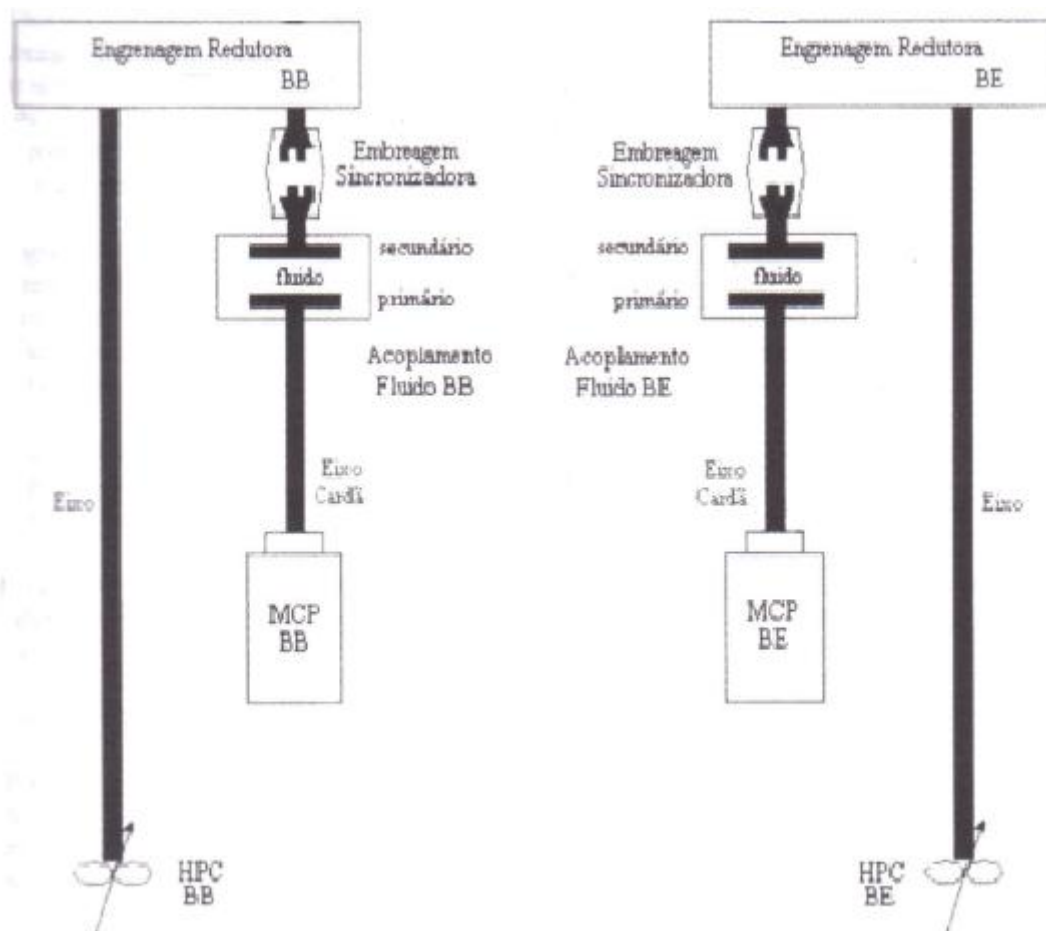


Figura 1.3 - Arranjo da planta propulsora.

1.3 Malha de controle

A Figura 1.4 segue para mostrar, de forma simplificada, como funciona a malha de controle adotada para o desenvolvimento do controle de trajetória do navio.

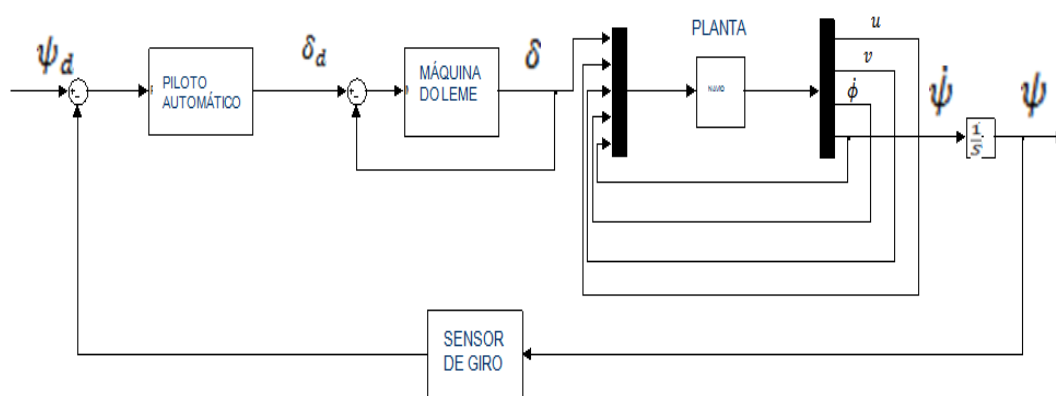


Figura 1.4 - Malha de controle.

2 Controles principais do navio

A Figura 2.1 mostra a tela principal da simulação, onde se pode controlar toda a dinâmica do navio.

No canto superior direito da figura, o estado do navio (velocidade, rumo, ângulo de leme, potência, torque e rotação dos motores, rotação e ângulo de passo dos hélices), o tempo de simulação e o tipo de governo são mostrados.

Logo abaixo, pode-se determinar quanto de demanda, que varia de -100% (marcha à ré) a 100% (marcha para frente), cada eixo deve atender. O campo verde se refere ao eixo de BE e o vermelho de BB. A atualização da demanda só acontece ao ser apertado o botão “PCL_enter”.

De forma análoga, acontece com os campos que atualizam o leme desejado, o rumo desejado, a direção e a intensidade do vento. No que tange ao campo do leme, este varia de -40° (BB) a 40° (BE).

A direção do vento se dá em graus e tem por referência o ponto de onde o vento está partindo. Por exemplo, se o vento está vindo do norte, ele terá a direção 0°. Já a intensidade do vento é dada em nós.

Há dois botões que tratam o tipo de governo que o navio está submetido: “Governo” e “Zig-zag”. O primeiro botão alterna o navio entre os modos “piloto automático” e “controle manual”. Se o NPa estiver sob “piloto automático”, controla-se o navio através da atualização do rumo desejado e a atualização de leme perde sua finalidade. Contudo, se o NPa estiver sob “controle manual”, ocorre o inverso, ou seja, o controle é feito pela atualização de leme. Nesse caso, a simulação também exibirá, abaixo do estado do governo, o raio da curva que o navio estaria percorrendo. O segundo botão irá funcionar somente se o governo do navio for “controle manual”, o que irá fazer com que o navio percorra uma trajetória em “zig-zag”, independente do rumo ou do leme desejados. É importante salientar que toda vez que o rumo desejado é atualizado, este passa a ser o rumo de referência para a manobra de “zig-zag”. O rumo de referência padrão, no início da simulação, é zero.

O botão “parada em emergência” funciona de forma a determinar que os dois eixos atendam com -100% de demanda até que o navio alcance a velocidade de 3 m/s, quando então será determinado que os eixos atendam com 0% de demanda.

Já o botão “centralizar” tem a finalidade apenas de colocar o desenho do navio no centro da simulação, além de atualizar os eixos cartesianos. Vale ressaltar que toda vez que o desenho do NPa ultrapassa um limite de algum eixo, este reaparecerá na extremidade oposta e haverá uma atualização do respectivo eixo.

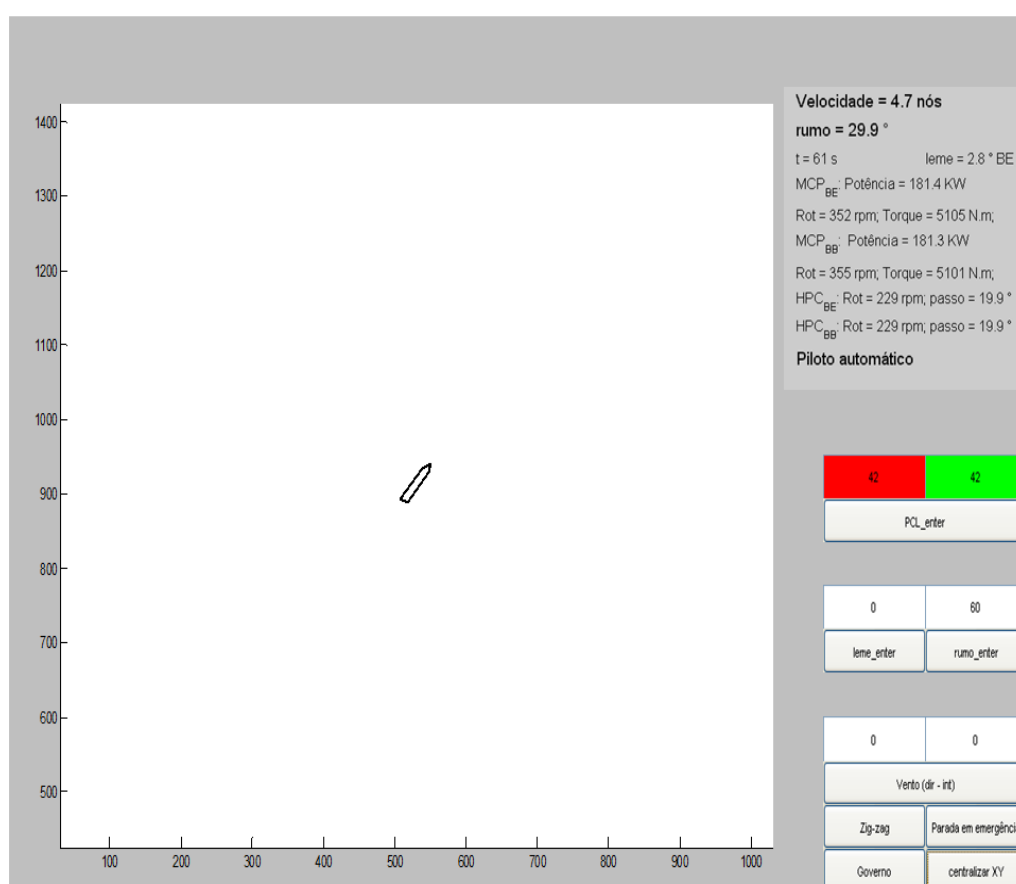


Figura 2.1 - Tela principal da simulação.

A Figura 2.2 mostra o sistema principal de controle da simulação. Essa página aparece ao ser chamado o arquivo sim_NPa.mdl. A partir desse sistema, a simulação recebe todas as suas entradas, realimentações e também gera a tela da Figura 2.1.

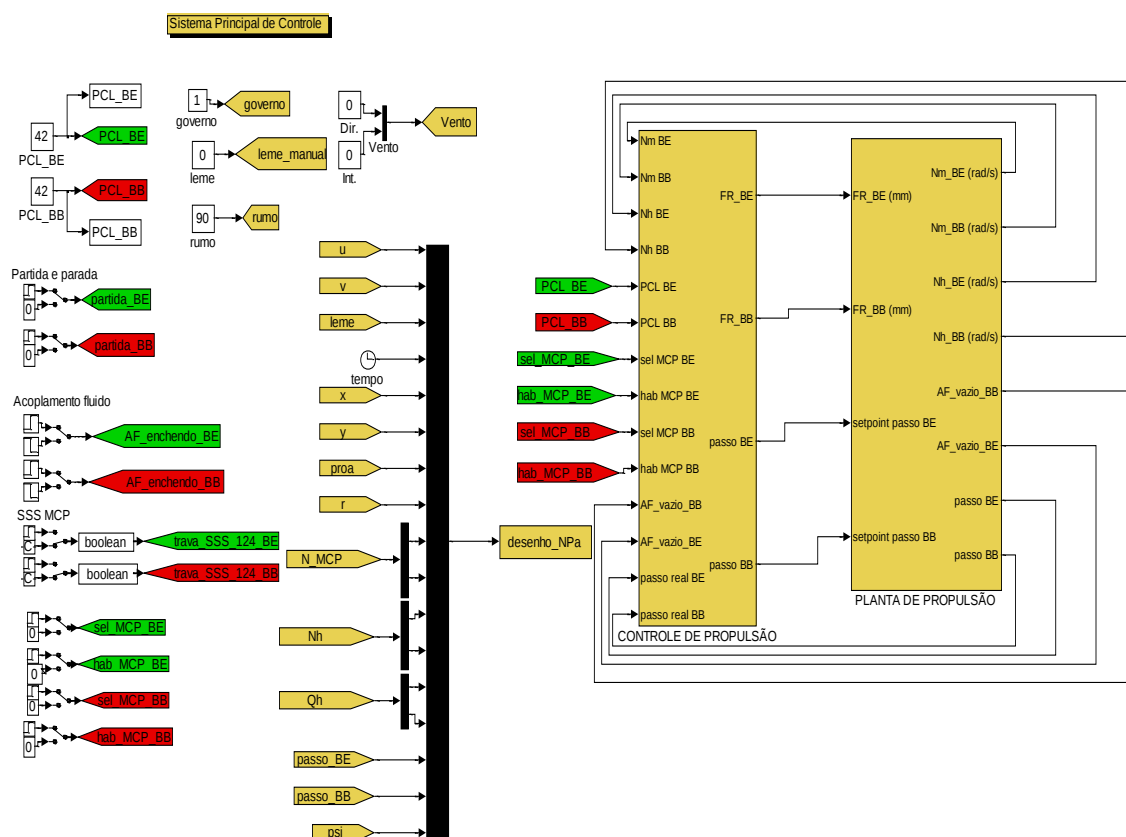


Figura 2.2 - Sistema principal da simulação.

A Figura 2.3 é uma ampliação de parte do sistema principal de controle. Os botões “PCL_enter”, “leme_enter”, “rumo_enter”, “Vento (dir – int)” e “Governo” da Figura 2.1 atualizam as constantes da Figura 2.3. Em relação à constante “governo”, será zero quando o NPa estiver operando no modo “Controle manual” e igual a um quando em “Piloto automático”.

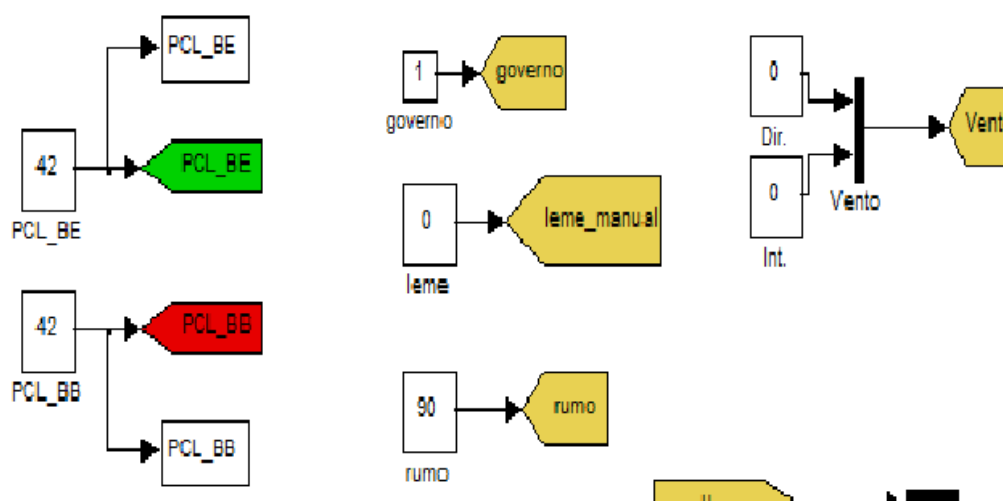


Figura 2.3 - Ampliação de parte do sistema principal de controle (controle).

A Figura 2.4 é também uma ampliação do sistema principal de controle, mas esta se refere a algumas chaves de segurança. As cor verde se refere ao eixo de BE e a vermelha ao de BB. Para que um eixo funcione normalmente, é preciso que seja dada partida no seu MCP, seu acoplamento fluido esteja cheio, seu eixo não esteja travado e o seu motor esteja selecionado e habilitado. Essas chaves foram implementadas nos moldes do Sistema de Propulsão da Corveta Barroso (navio da Marinha do Brasil) [1].

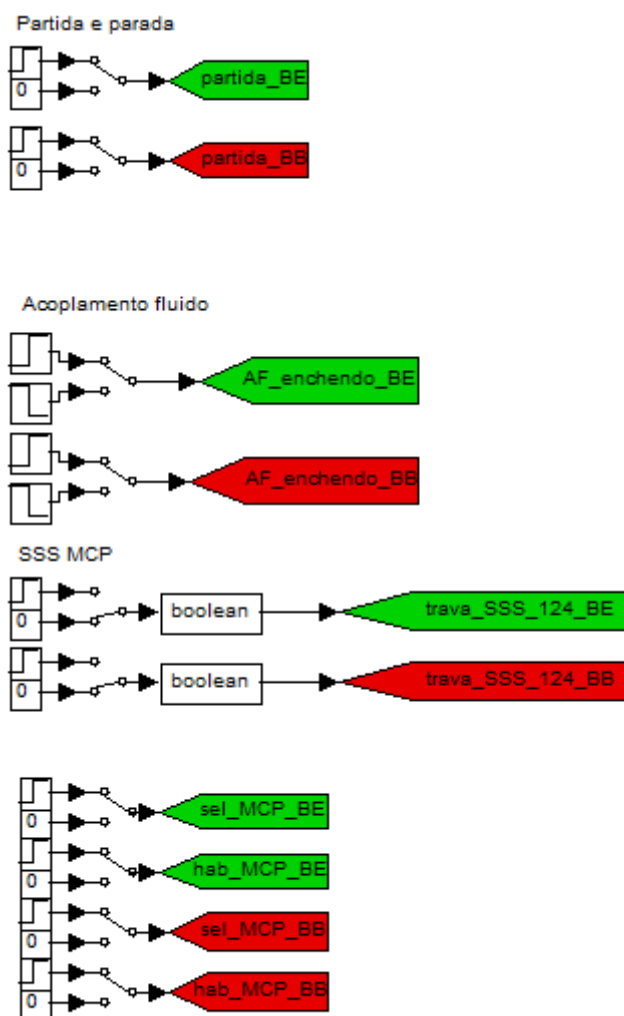


Figura 2.4 - Ampliação de parte do sistema principal de controle (segurança).

A Figura 2.5 se refere também a uma ampliação do sistema principal de controle e mostra os subsistemas que realizam efetivamente toda a simulação do modelo.

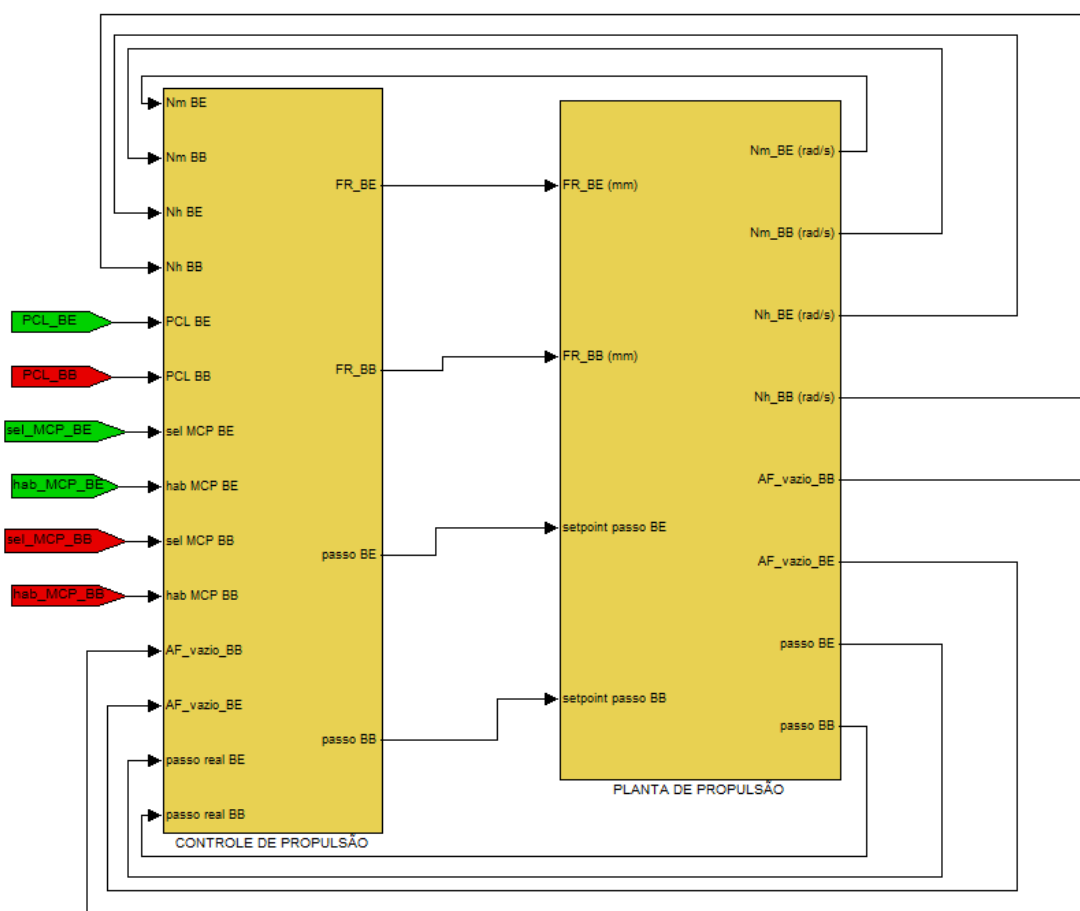


Figura 2.5 - Ampliação de parte do sistema principal de controle (dinâmica).

Por fim, a Figura 2.6 mostra a última parte do sistema principal de controle ampliada. Nela, reúnem-se inúmeras variáveis do modelo e gera-se a tela da simulação da Figura 2.1.

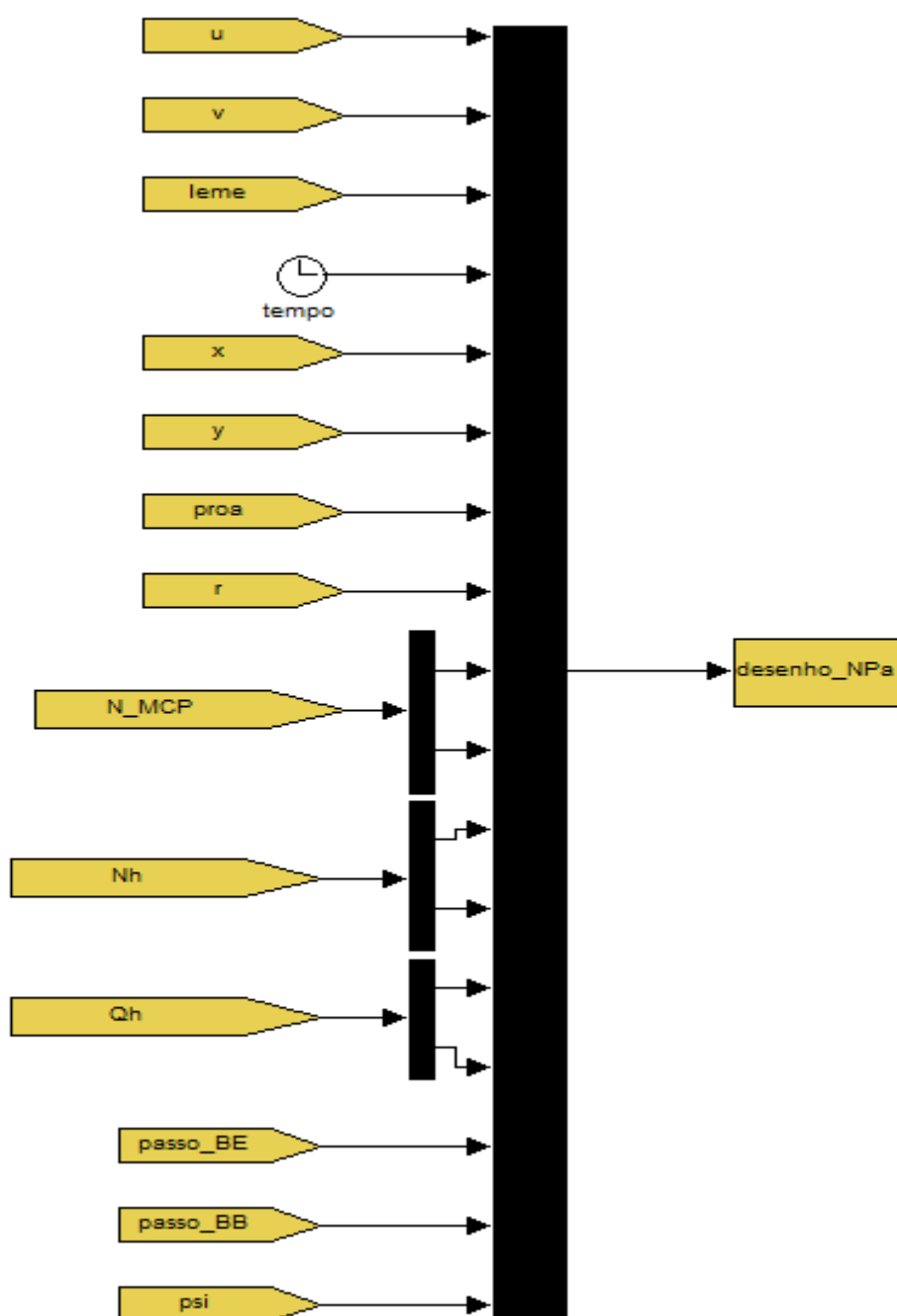


Figura 2.6 - Ampliação de parte do sistema principal de controle (desenho).

3 Simulação da propulsão

3.1 CONTROLE DE PROPULSÃO

É importante ressaltar que toda simulação da propulsão foi extraída do modelo da propulsão da Corveta Barroso desenvolvido pelo Laboratório de Automação e Controle da Escola Politécnica da USP[1].

A Figura 3.1 mostra o subsistema que realiza o controle de cada MCP e de cada HPC.

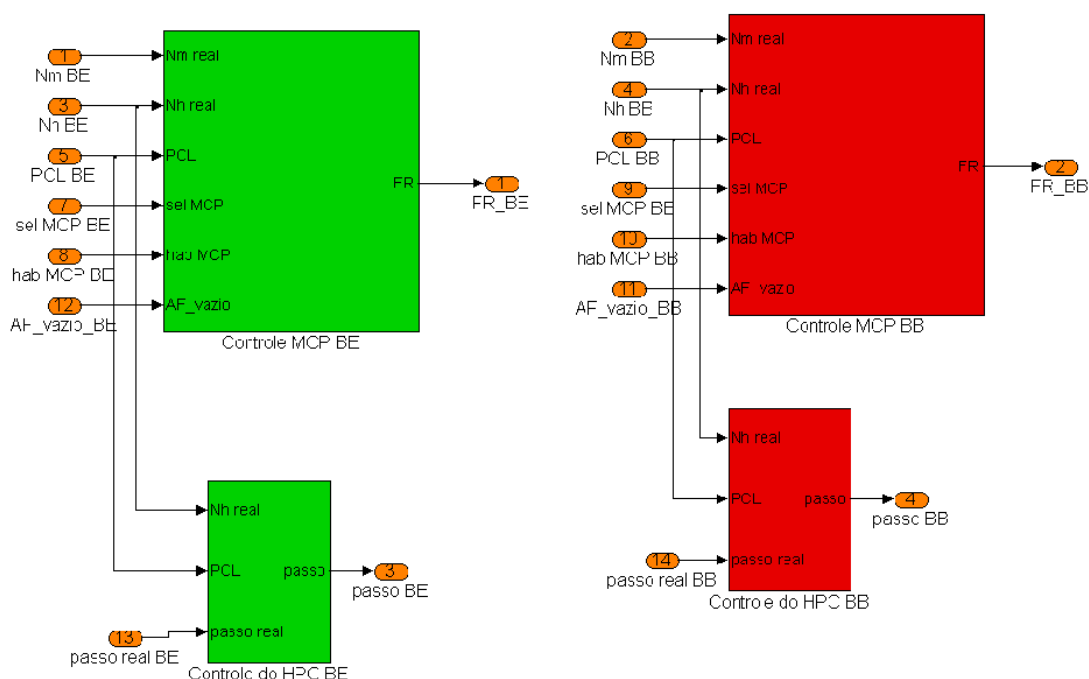


Figura 3.1 - Subsistema CONTROLE DE PROPULSÃO.

A Figura 3.2 amplia parte da Figura 3.1 que mostra o controle da propulsão do eixo de BE. Note que, ao observar a Figura 3.1, a parte que se refere ao eixo de BB é idêntico ao de BE. As entradas desse subsistema são rotações dos motores, rotações dos eixos, valores de Power Control Level (PCL), as informações se os

3.1.1 Subsistema “Controle MCP”

No subsistema “Controle MCP”, a referência de rotação varia em função do PCL ordenado e varia de acordo com o gráfico da Figura 3.3. Caso o motor não esteja habilitado, a referência de rotação será de 350 rpm (rotação mínima do motor).

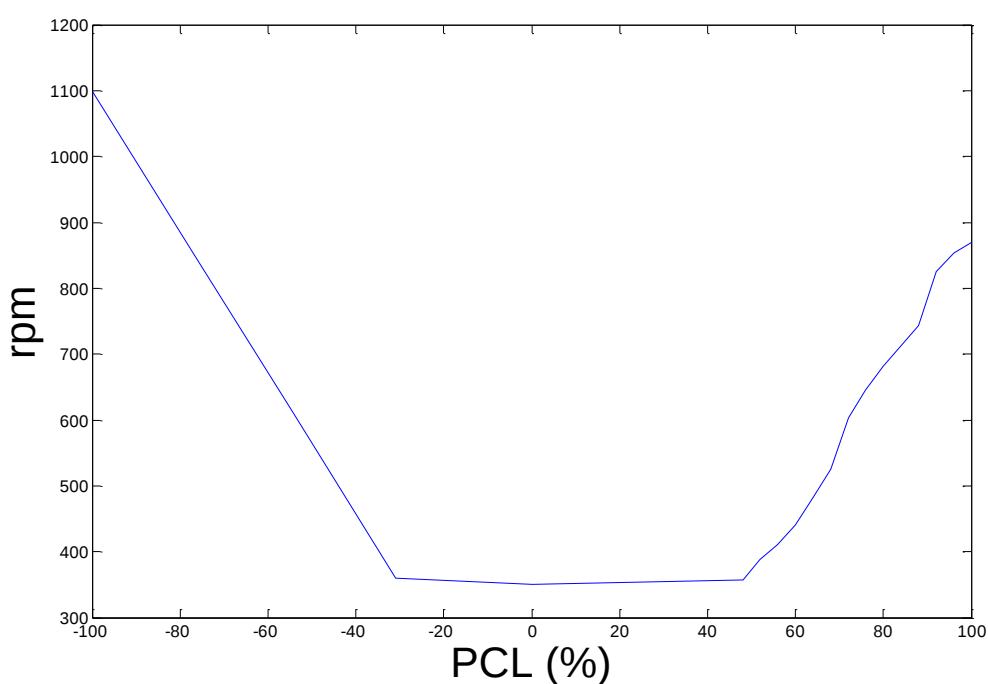


Figura 3.3 - Rotação em função de PCL.

Analogamente à Figura 3.3, há uma referência de rotação do HPC em função do PCL. Como há perdas na planta de propulsão, soma-se à referência de rotação do motor um valor proporcional à diferença entre a referência de rotação do HPC e o valor de sua rotação real, como mostra a Figura 3.4.

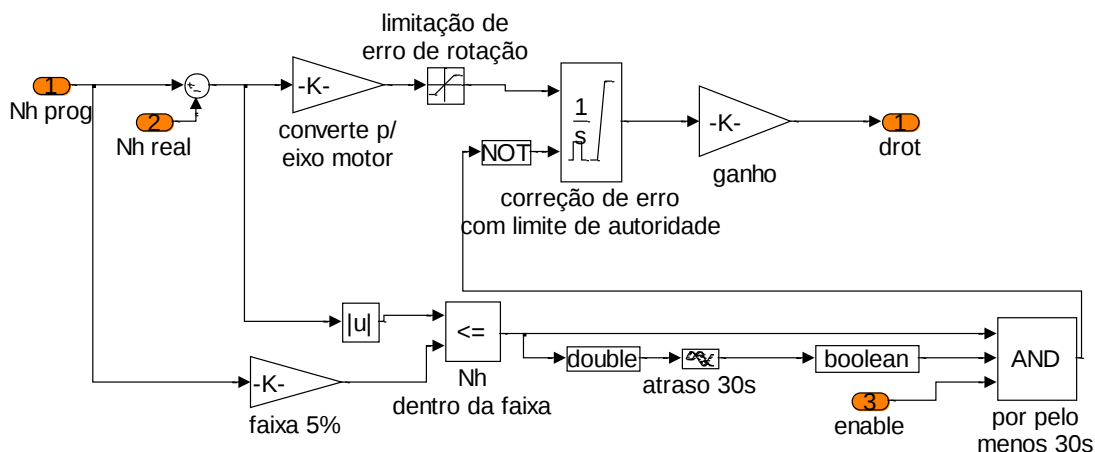


Figura 3.4 - Malha auxiliar de rotação.

Caso o acoplamento fluido esteja vazio, o sistema de controle não permite que os motores ultrapassem 500 rpm.

O controle de aceleração dos motores dá-se por taxas variáveis de valor de referência de rotação para que não haja uma defasagem muito grande entre a rotação real e a referência de rotação. As taxas de aceleração variam entre 2 e 30 rpm/s e de desaceleração de 50 e 200 rpm/s. Por fim, a diferença entre a referência de rotação e a rotação real fornecerá a posição da cremalheira do motor (regulador do motor). Abaixo, a Figura 3.5 mostra esse sistema de controle.

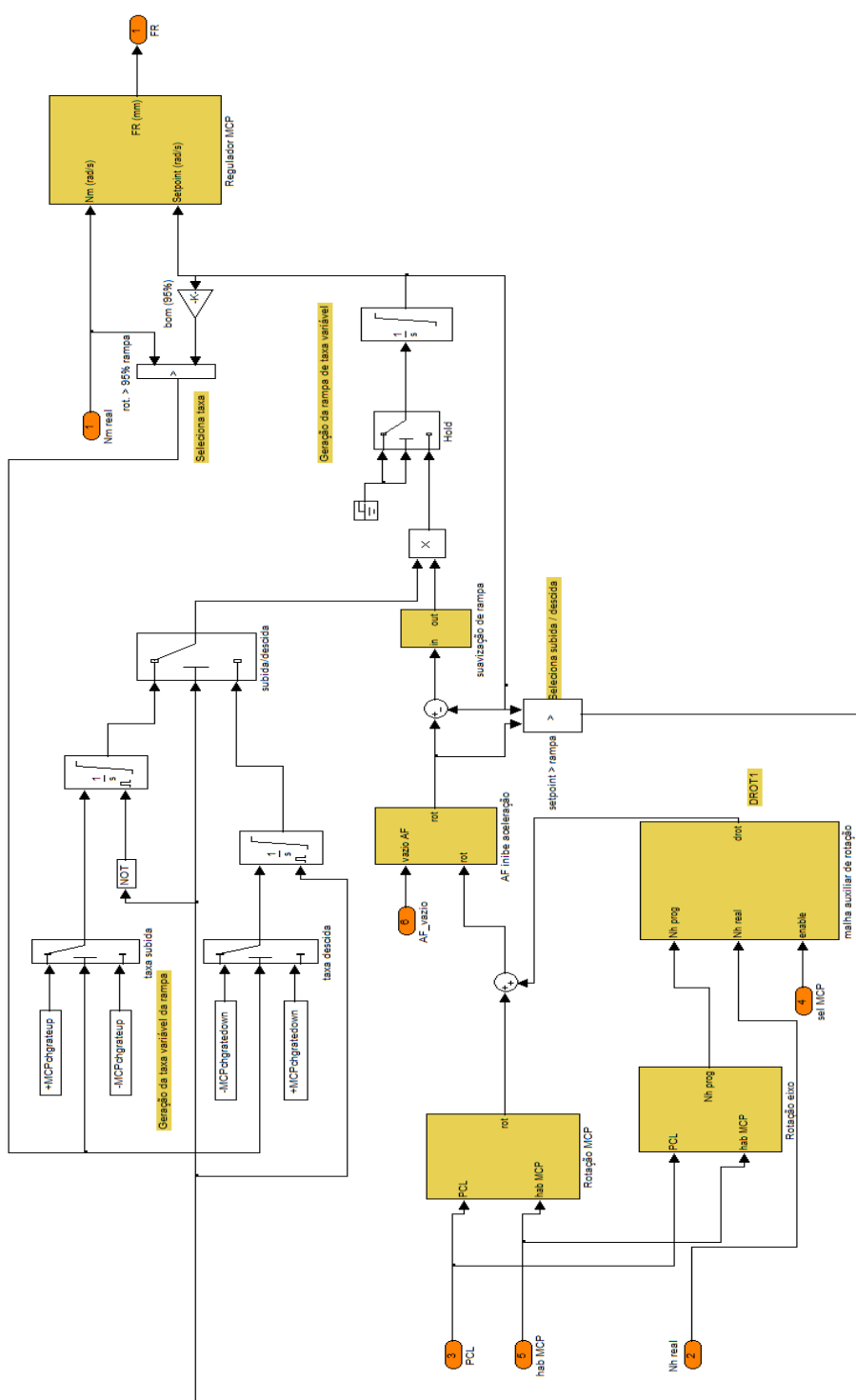


Figura 3.5 - Subsistema "Controle MCP".

3.1.2 Subsistema “Controle do HPC”

Assim como no controle do MCP, o ângulo de passo de referência varia em função do PCL, como mostra a Figura 3.6.

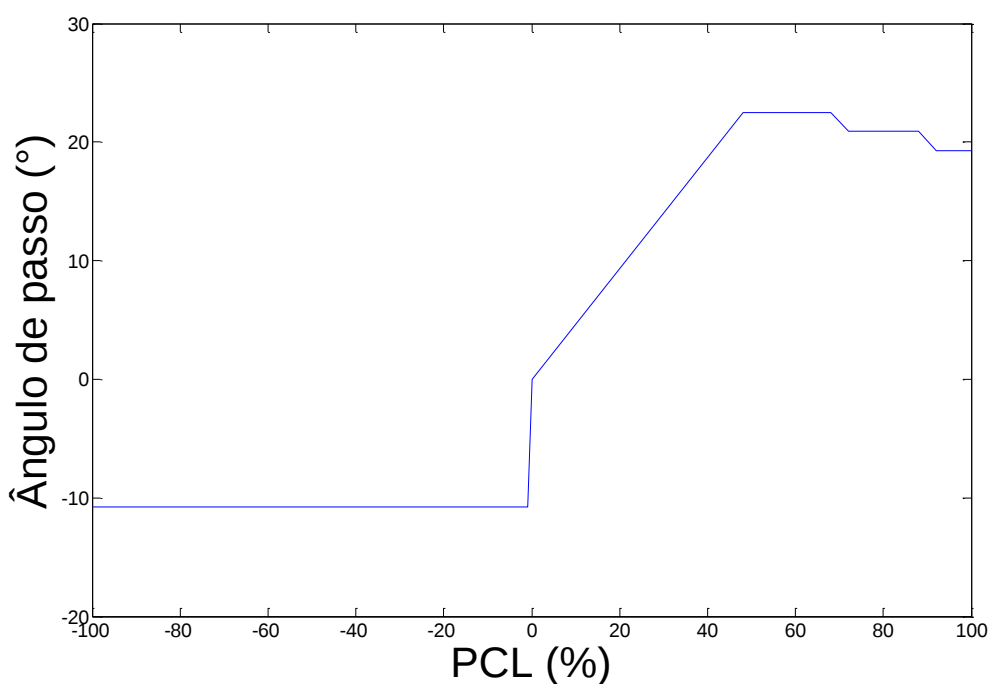


Figura 3.6 - Ângulo de passo em função de PCL.

3.2 PLANTA DE PROPULSÃO

É no subsistema “PLANTA DE PROPULSÃO” que são simuladas as dinâmicas desde os motores, acoplamentos fluidos, engrenagem redutora até os hélices, implementado na Figura 3.7.

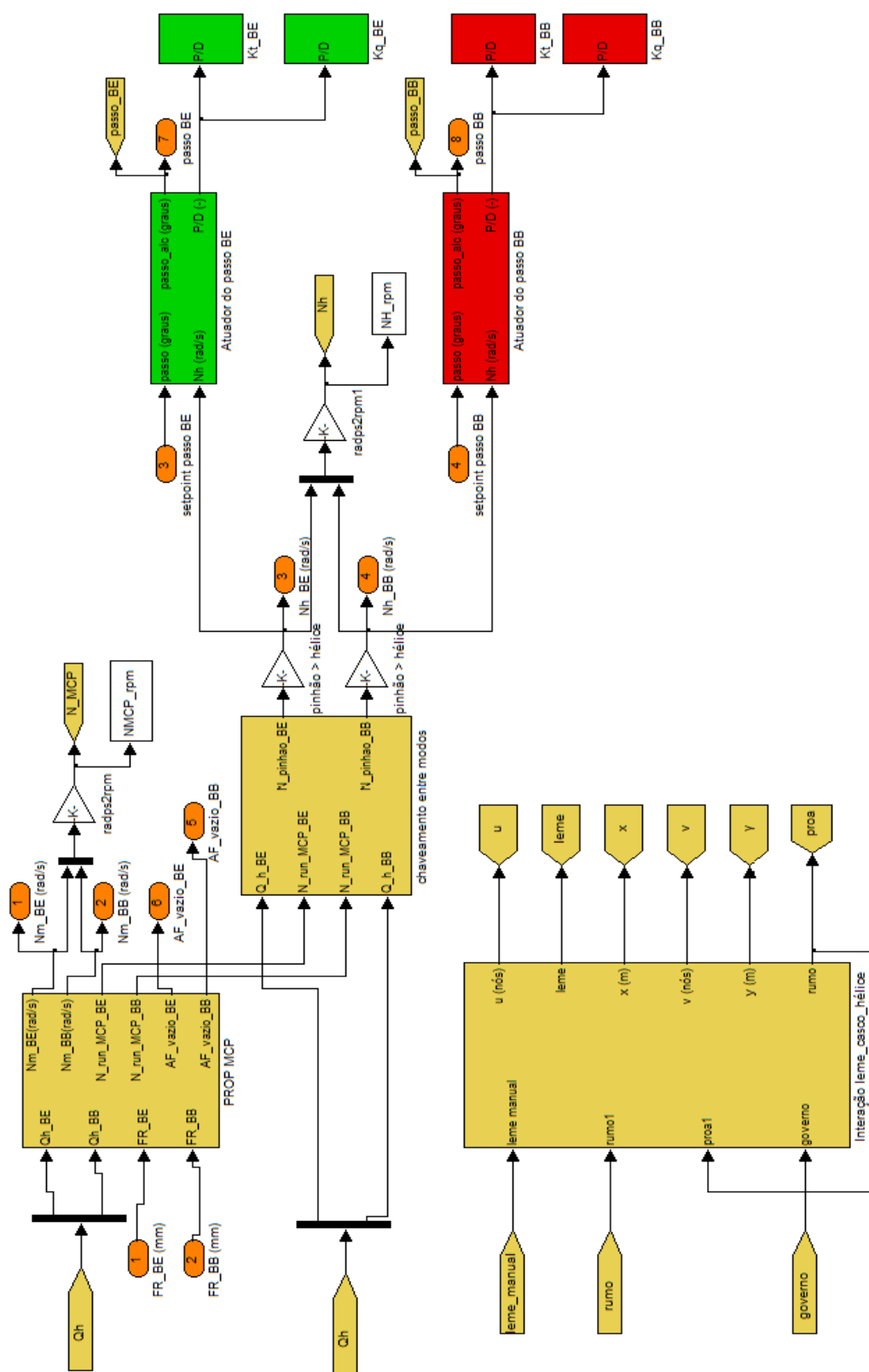


Figura 3.7 - Subsistema "PLANTA DE PROPULSÃO".

A Tabela 3.1 - Momentos de inércia da planta da propulsão. Tabela 3.1 mostra os momentos de inércia das partes da planta da propulsão.

Tabela 3.1 - Momentos de inércia da planta da propulsão.

Componente	Momento de inércia (kg.m²)
MCP	54,95
Eixo até o acoplamento fluido	2,6
Bomba do acoplamento fluido	90,03
Runner do motor	38,42
Embreagem sincronizadora (lado do motor)	6,88
Embreagem sincronizadora (lado do HPC)	3,43
Engrenagem redutora (pinhão)	11,0
Engrenagem redutora (coroa)	5082,2
Eixo	573,08
HPC	3900

3.2.1 Subsistema “PROP MCP”

A Figura 3.8 mostra que esse subsistema divide a dinâmica em duas partes: do MCP até módulo primário do acoplamento fluido e do secundário do acoplamento fluido até o HPC.

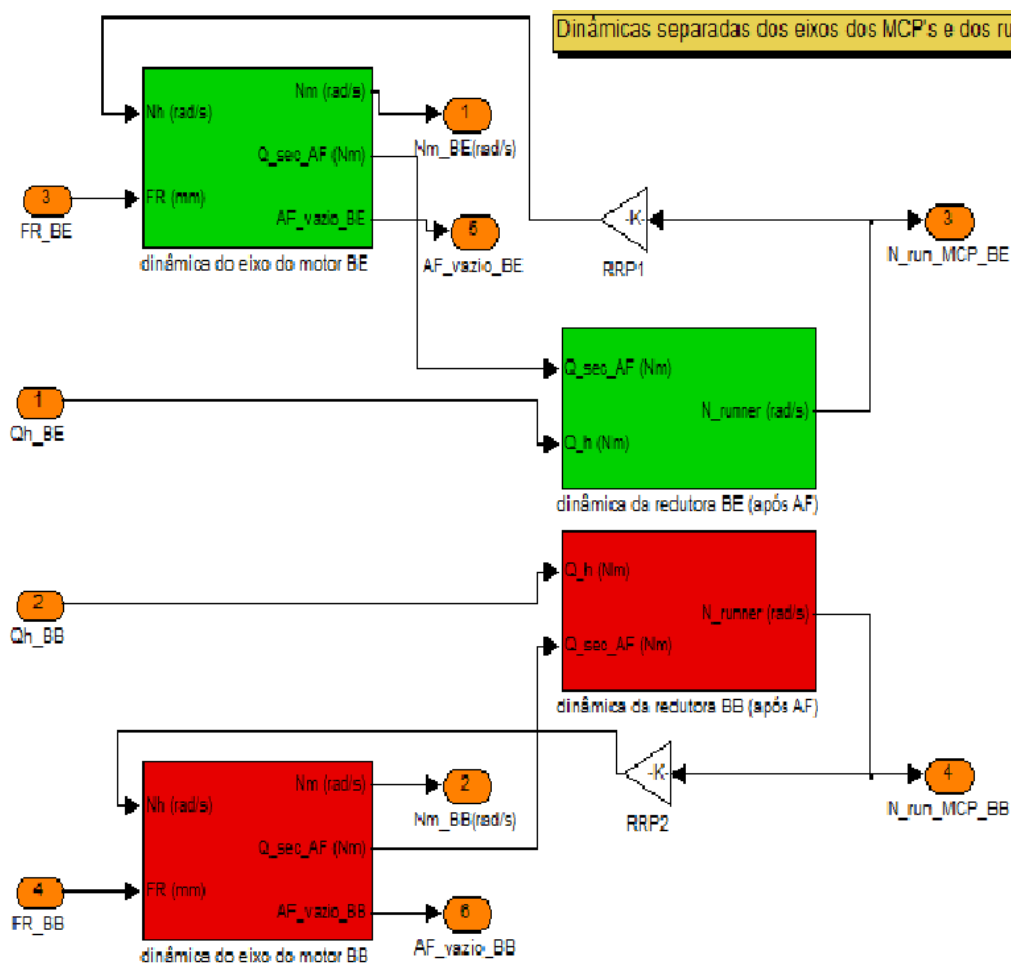


Figura 3.8 - Subsistema "PROP MCP".

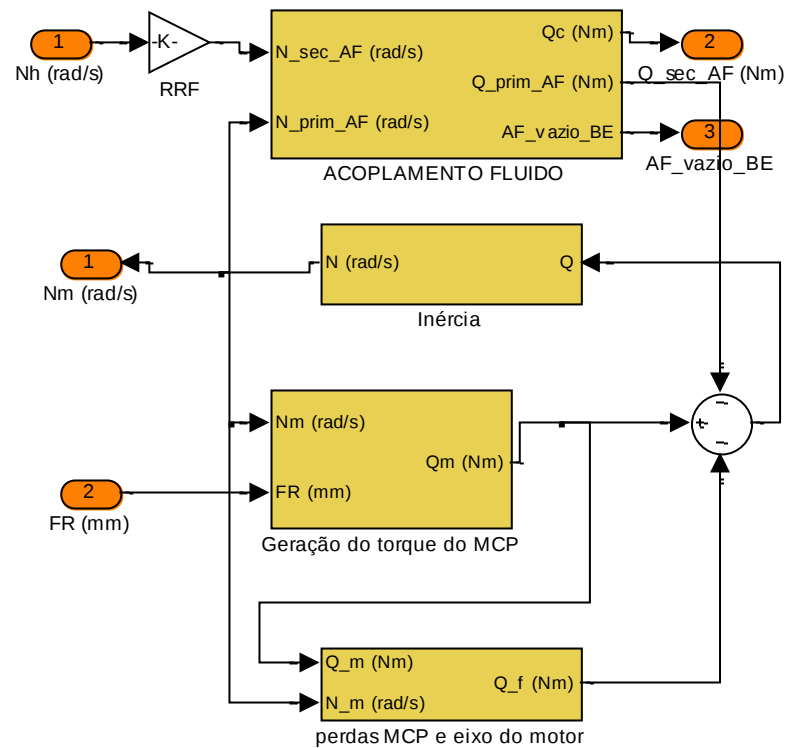
3.2.1.1 Subsistema "dinâmica do eixo"

A Figura 3.9 mostra o subsistema que segue o seguinte equacionamento:

$$J_{pr_af} \frac{dN_m}{dt} = Q_{MCP} - Q_{pr_af} - Q_{pr_f} \quad (3.1)$$

$$N_m = \frac{1}{J_{pr_af}} \int (Q_{MCP} - Q_{pr_af} - Q_{pr_f}) dt \quad (3.2)$$

Vale ressaltar que as perdas no MCP e no seu eixo são da ordem de 1% do torque fornecido pelo motor.



Dinâmica do eixo do MCP

Figura 3.9 - Subsistema "dinâmica do eixo".

3.2.1.1.1. Subsistema “ACOPLAMENTO FLUIDO”

A Figura 3.10 como foi implementado o equacionamento do acoplamento fluido, que é nada mais do que um dispositivo hidráulico, composto por uma bomba (primário) e uma turbina (secundário) e entre elas um mesmo fluido. Sua função é transmitir torque, sem contudo, transmitir vibrações. Seguem as equações:

$$Q_c = \frac{1}{c} [\text{sgn}(\text{esc}) \sqrt{\text{esc}}] n_1^2 \quad (3.3)$$

Onde:

- $Q_c \rightarrow$ torque transmitido à turbina;
- $\text{esc} \rightarrow$ escorregamento:

$$\text{esc} = \frac{n_1 - n_2}{\max(n_1, n_2)} \quad (3.4)$$

- $n_1 \rightarrow$ rotação da bomba;
- $n_2 \rightarrow$ rotação da turbina;
- $c \rightarrow$ constante, que foi ajustada para que haja um escorregamento de 3% quando o motor estiver fornecendo torque para que o navio assuma a velocidade de 12 nós (6,17 m/s). Assim $c = 0,0285$;
- $\text{sgn}(\text{esc})$:
 - +1, se $n_1 > n_2$;
 - 0, se $n_1 = n_2$;
 - -1, se $n_1 < n_2$.

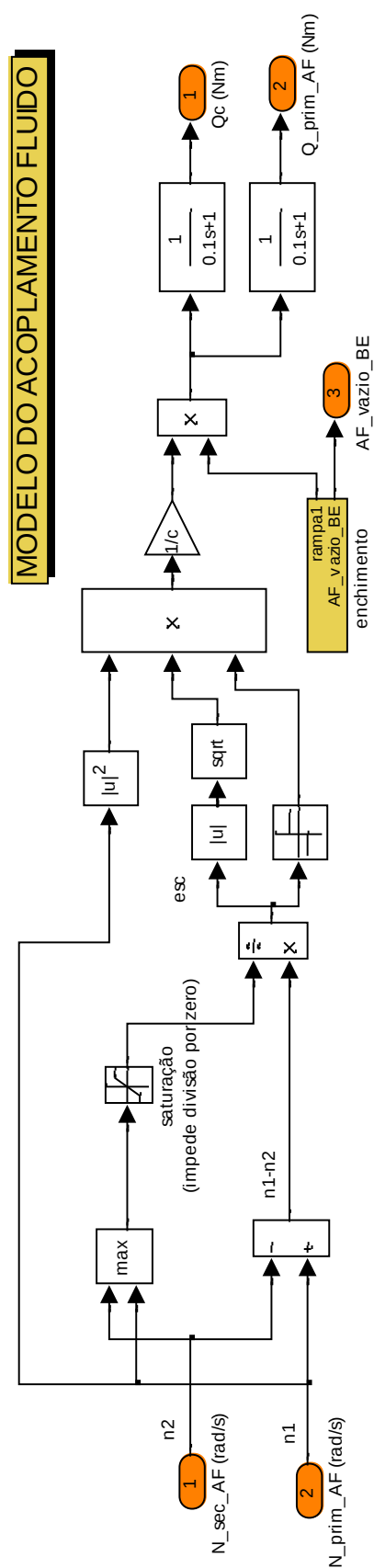


Figura 3.10 - Subsistema "ACOPLAMENTO FLUIDO".

3.2.1.2 Subsistema “dinâmica da redutora (após AF)”

A Figura 3.11 mostra a forma como foi implementado a dinâmica do secundário do acoplamento fluido até o HPC. Esse subsistema é equacionado da seguinte forma:

$$J_{se_af} \frac{dN_m}{dt} = Q_{se_af} - \frac{Q_h}{(RRF)} - Q_{se_f} \quad (3.5)$$

$$N_m = \frac{1}{J_{se_af}} \int \left(Q_{se_af} - \frac{Q_h}{(RRF)} - Q_{se_f} \right) dt \quad (3.6)$$

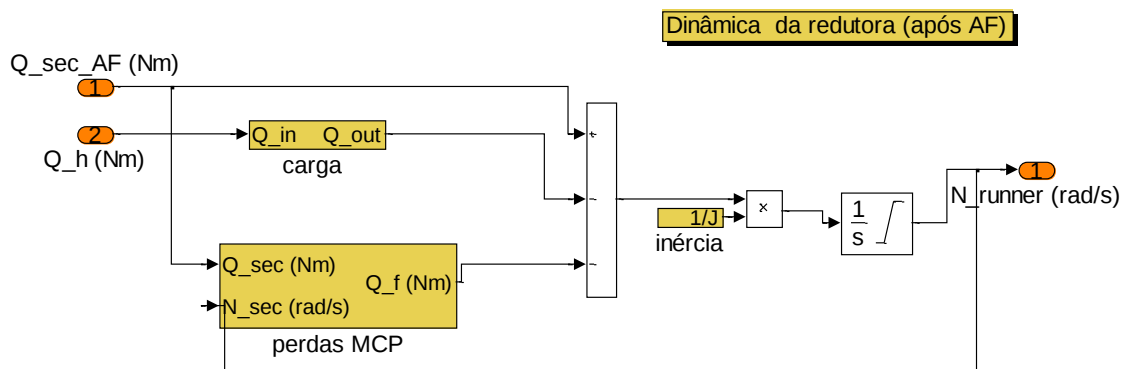


Figura 3.11 - Subsistema "dinâmica da redutora (após AF)".

3.2.2 Subsistema “chaveamento entre modos”

Esse subsistema, basicamente, tem como entradas os torques nos hélices. Esses torques serão usados caso as embreagens sincronizadoras estejam abertas, ou seja, os hélices iriam desacelerar. As rotações dos hélices, nesse caso, são chamadas de rotações de eixo livre. É no subsistema “lógica de chaveamento” que se determina quando as embreagens sincronizadoras estão engrazadas. De acordo com o estado das embreagens, o subsistema “roteamento de velocidades” decidirá

se a velocidade do secundário do AF se refere à dinâmica completa do sistema ou se o eixo está livre. Caso as embreagens estejam engrazadas, as rotações dos hélices serão:

$$N_h = \frac{N_{se_af}}{RRF} \quad (3.7)$$

A Figura 3.12 mostra a implementação do subsistema.

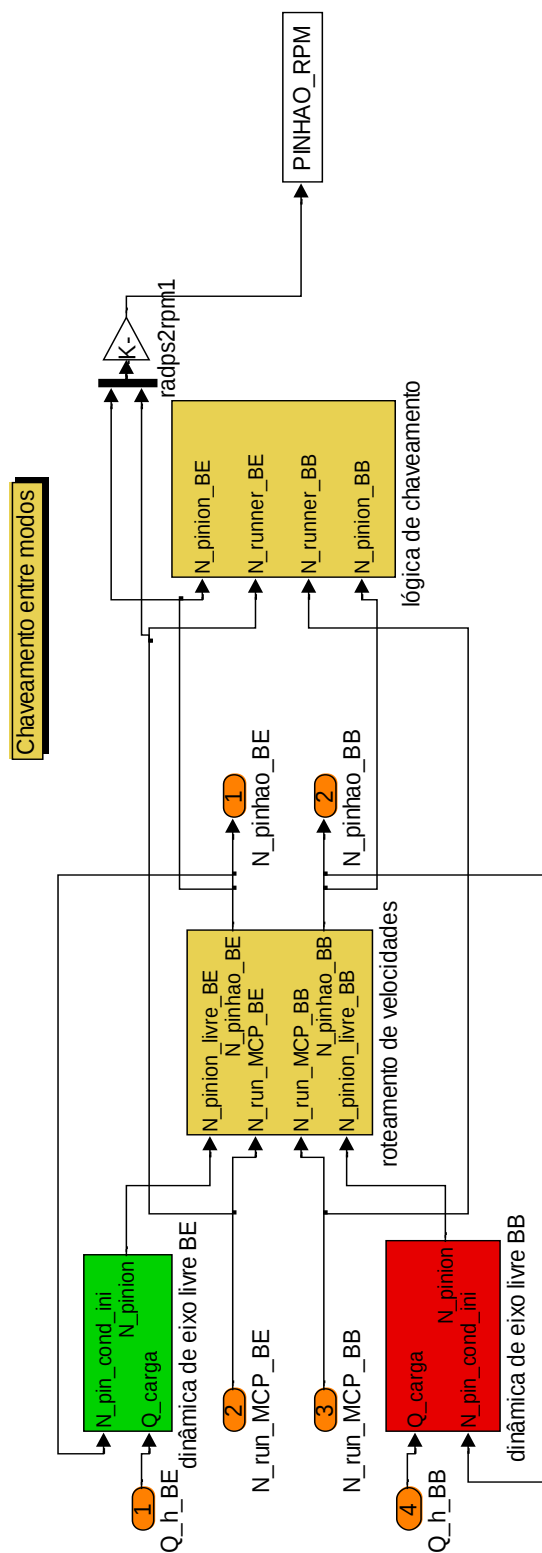


Figura 3.12 - Subsistema "chaveamento entre modos".

3.2.2.1 Subsistema “lógica de chaveamento”

Nesse subsistema, há duas lógicas, uma para cada eixo, mostrada na Figura 3.13.

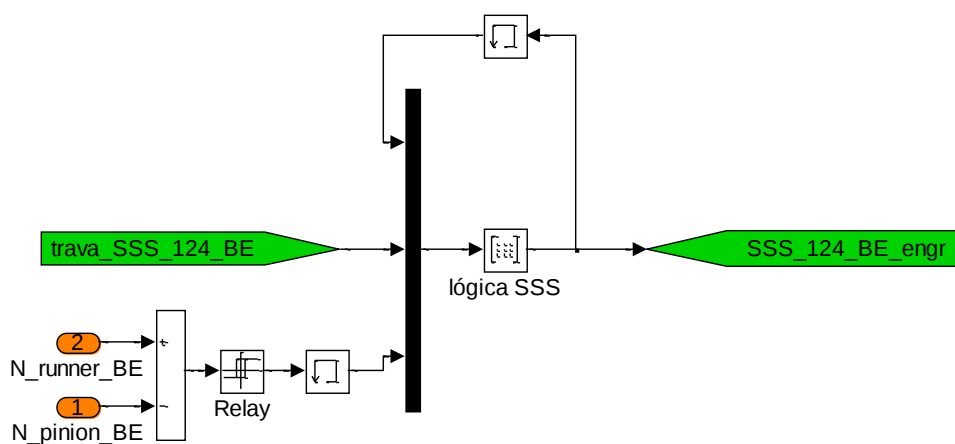


Figura 3.13 - Subsistema "modelo lógico das embreagens".

A lógica funciona de acordo com a Tabela 3.2. Se a trava da embreagem for acionada, ela automaticamente desengrenará a embreagem. O mesmo acontecerá se a diferença entre as rotações do MCP e do secundário do acoplamento fluido for maior que 0,5 rad/s.

Tabela 3.2 - Lógica das engrenagens SSS 124.

Estado da SSS 124	Trava da SSS 124	Limite de diferença de rotação permitido	Estado da SSS 124
F	F	F	F
F	F	V	V
F	V	F	F
F	V	V	F
V	F	F	F
V	F	V	V
V	V	F	F
V	V	V	F

3.2.3 Subsistema “atuador de passo”

A Figura 3.14 mostra o funcionamento do atuador hidráulico do passo. A bomba usada no sistema possui seu eixo dependente ao eixo do hélice. Ou seja, quanto mais rápido o hélice estiver girando, mais rápido o sistema atuará.

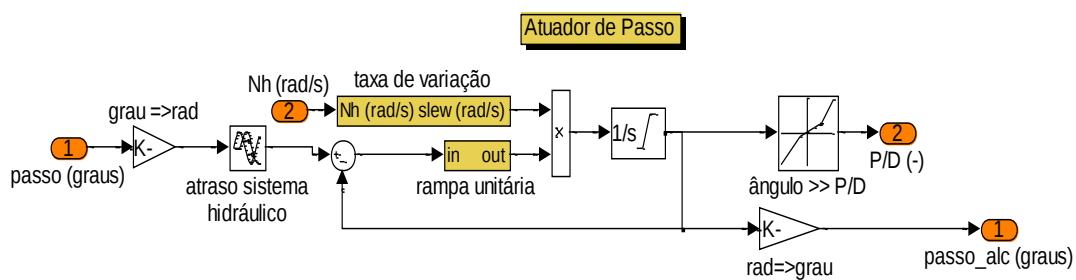


Figura 3.14 - Subsistema "atuador de passo".

A Figura 3.15 mostra como varia a razão entre o passo do hélice pelo seu diâmetro em função do ângulo de passo.

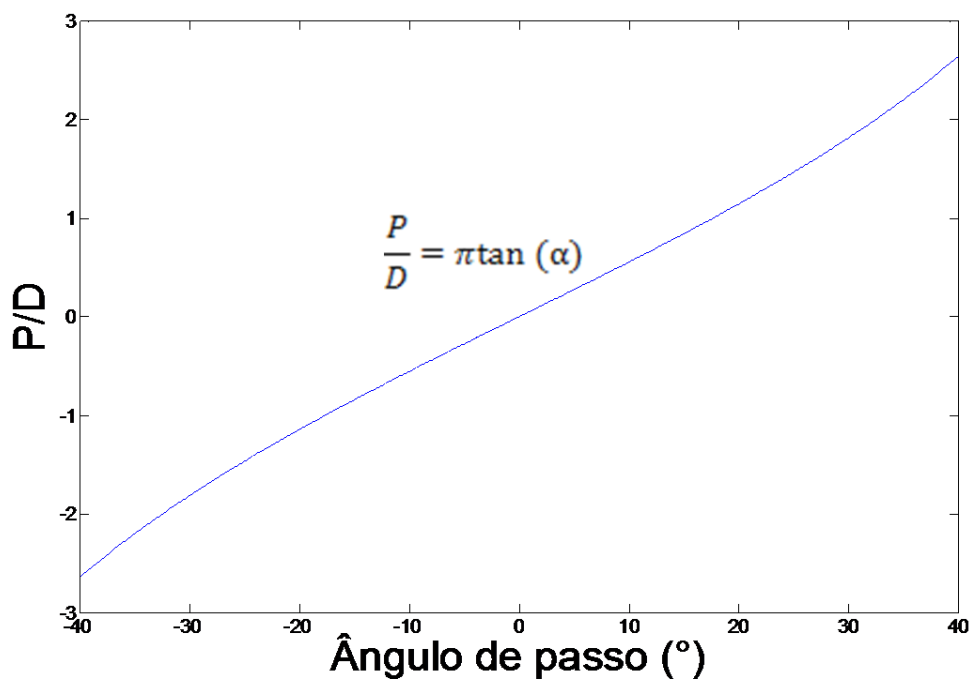


Figura 3.15 - P/D em função do ângulo de passo.

4 Resistência ao avanço

A resistência ao avanço, como o nome sugere, refere-se à força de reação que a água do mar exerce no casco do navio devido ao seu movimento. Esse esforço foi modelado com dois enfoques, quais sejam, o navio indo para frente ou indo para trás.

No primeiro caso, a resistência é aproximada pelas seguintes equações, onde a velocidade é dada em nós e a resistência em KN:

- Velocidade do navio entre 0 e 15 nós (7,72 m/s):

$$Res = 1,01 \cdot 10^{-4}v^6 - 4,71 \cdot 10^{-3}v^5 + 7,85 \cdot 10^{-2}v^4 - 5,55 \cdot 10^{-1}v^3 + 1,83v^2 - 1,28v \quad (4.1)$$

- Velocidade do navio acima de 15 nós (7,72 m/s):

$$Res = 6,75 \cdot 10^{-3}v^4 - 6,07 \cdot 10^{-1}v^3 + 19,11v^2 - 233,79v + 981,788 \quad (4.2)$$

Para o segundo caso, fez-se uso do texto da Federação Portuguesa de Aeromodelismo[2], a modelagem do Navio indo para trás ficou da seguinte forma:

$$Res = \frac{1}{2}K\rho Sv^2 \quad (4.3)$$

Onde k e S são o coeficiente de forma e a área da seção submetida ao escoamento do fluido respectivamente. Desse modo, o Navio foi modelado como um cone, o que fornece um k=0,56. Já área foi modelada como S=B.T_s.

Nota-se na Figura 4.1, que a resistência ao avanço para o navio indo para trás é muito maior do que a referente ao navio indo para frente, para uma mesma velocidade.

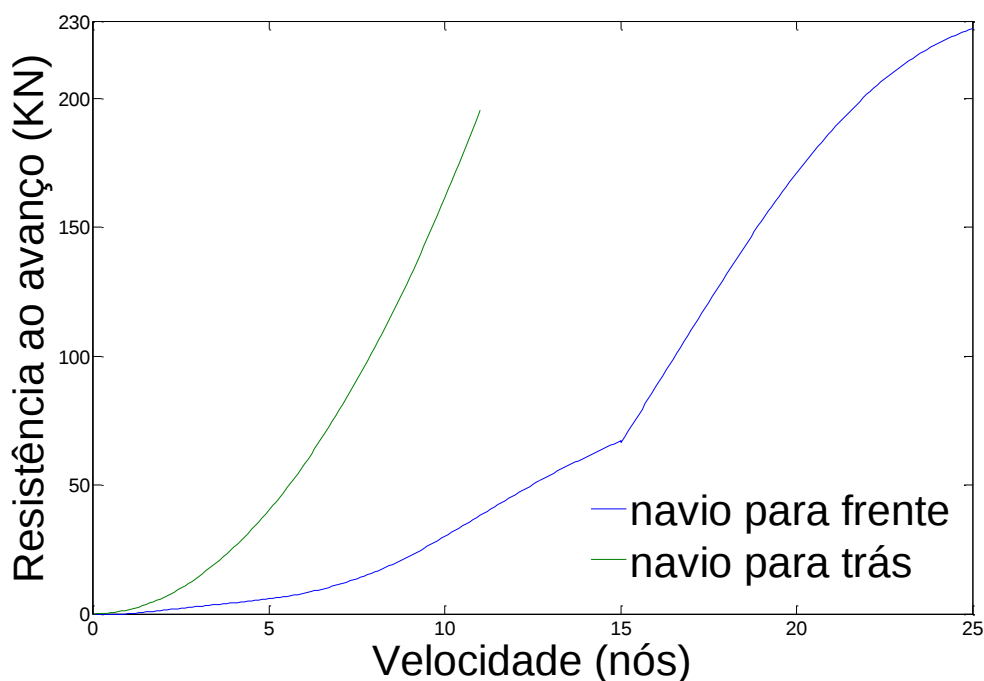


Figura 4.1 – Resistência ao avanço em função da velocidade.

4.1 Ajuste do passo do HPC

Uma vez modelada a resistência ao avanço do navio, o passo deve ser ajustado em função do empuxo necessário para vencer a resistência contrária e da velocidade do navio. Além disso, para cada configuração de empuxo e velocidade, é requerido um torque ao hélice.

Na equação ((4.4), nota-se que o empuxo no hélice depende da constante K_T e da rotação do hélice.

$$T = K_T \rho D^4 n^2 \text{ (empuxo)} \quad (4.4)$$

A Figura 4.2 mostra as configurações de rotação do motor e K_T , associados a um passo e a uma velocidade do navio. É importante ressaltar que, nos pontos onde a rotação requerida e K_T são zero, são configurações em que haveria grande probabilidade de cavitação do hélice.

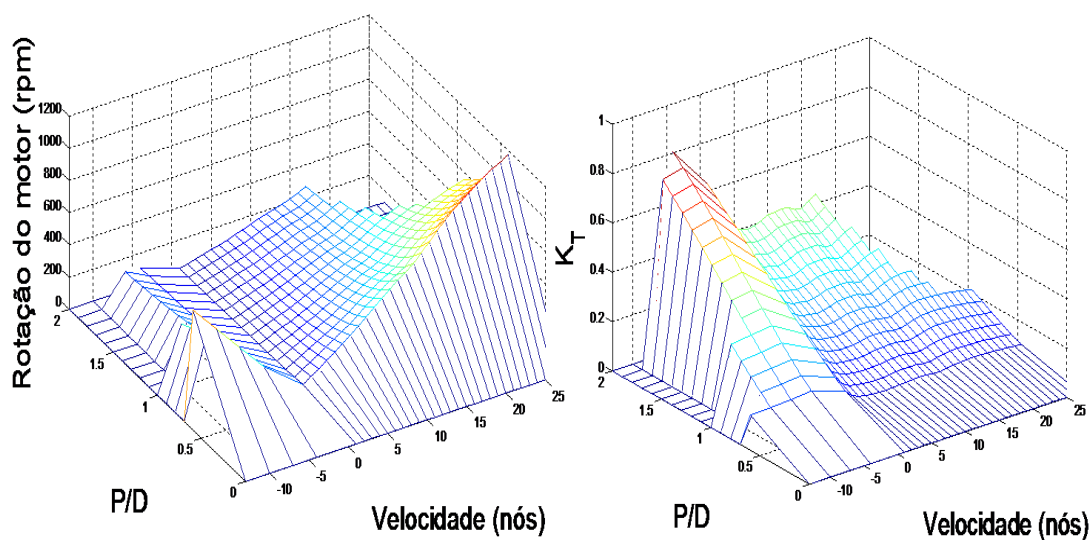


Figura 4.2 – Rotação e K_T em função da velocidade.

Analogamente à equação ((4.4), a equação ((4.5) modela o empuxo no hélice:

$$Q = K_Q \rho D^5 n^2 \text{ (torque)} \quad (4.5)$$

De forma semelhante à Figura 4.2, a Figura 4.3 mostra as configurações de rotação do motor e K_Q , associados a um passo e a uma velocidade do navio.

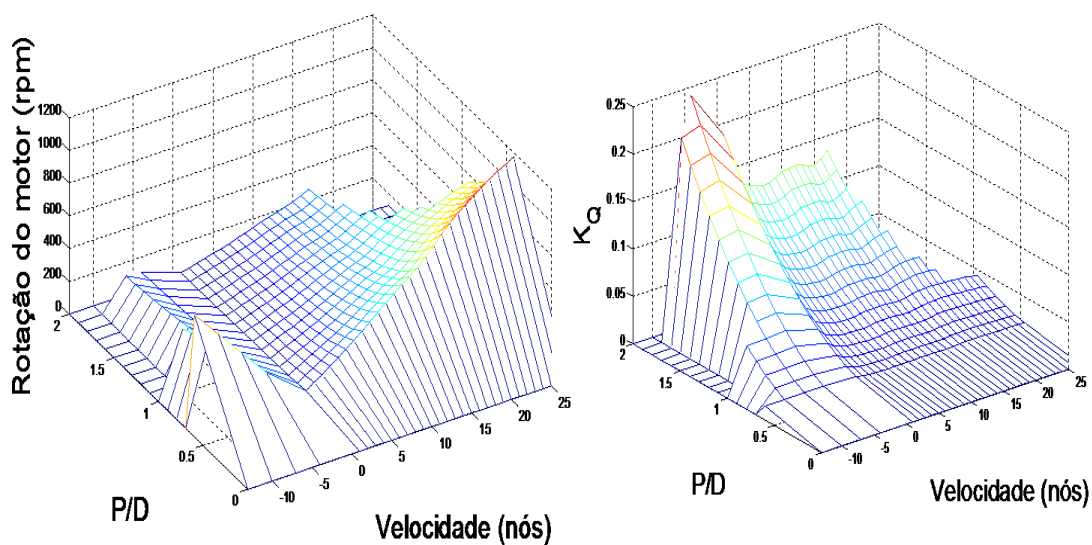


Figura 4.3 – Rotação e K_Q em função da velocidade.

A constante que relaciona K_T e K_Q é a eficiência relativa rotativa:

$$ef_{rr} = \frac{K_T J}{K_Q 2\pi} \quad (4.6)$$

J (número de avanço) é uma constante que depende da velocidade do fluxo no hélice. A Figura 4.4 mostra o comportamento dessa eficiência em função da velocidade e do passo. Dessa forma, para cada velocidade amostrada, foi associado o valor de passo que desse a maior ef_{rr} , ou seja, buscou-se o maior empuxo possível, com o menor torque possível, em cada velocidade.

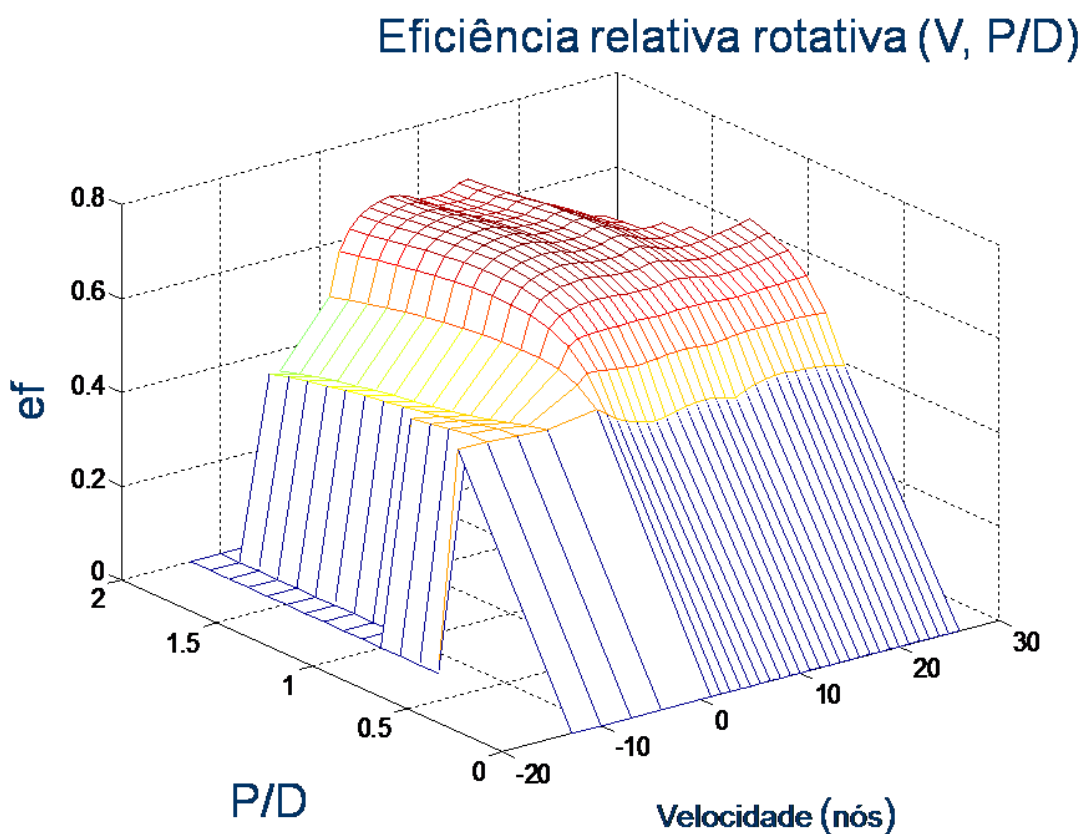


Figura 4.4 – Eficiência relativa rotativa em função da velocidade e do passo.

Assim, em busca da maior eficiência relativa rotativa, o sistema de controle do passo fixou, para cada PCL e em consequencia, para cada rotação do motor, o passo, como mostrou a Figura 3.3, a Figura 3.6 e mostra a Figura 4.5. Ao observar-se essa última Figura, deve ser levado em conta que, da faixa de velocidade de -4 a 12 nós (6,17 m/s), o motor mantém sua rotação constante em 350 rpm, sendo o navio acelerado apenas pelo aumento do passo do hélice.

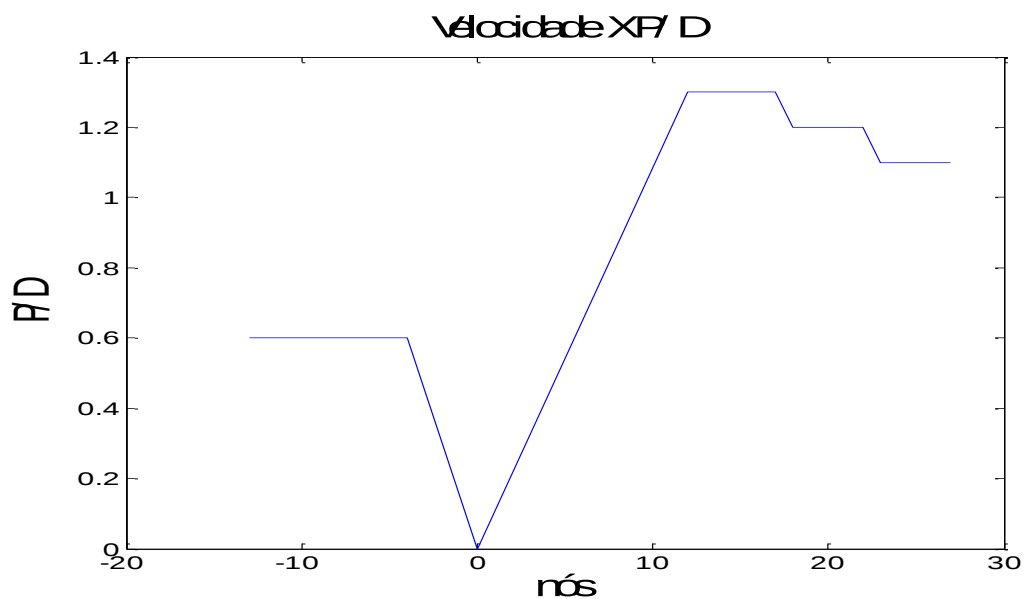


Figura 4.5 – Passo em função da velocidade do navio.

Finalmente, a Figura 4.6 mostra que, para toda configuração de rotação do motor e o passo do hélice associado, a potência requerida ao MCP não ultrapassa a curva de potência máxima do motor.

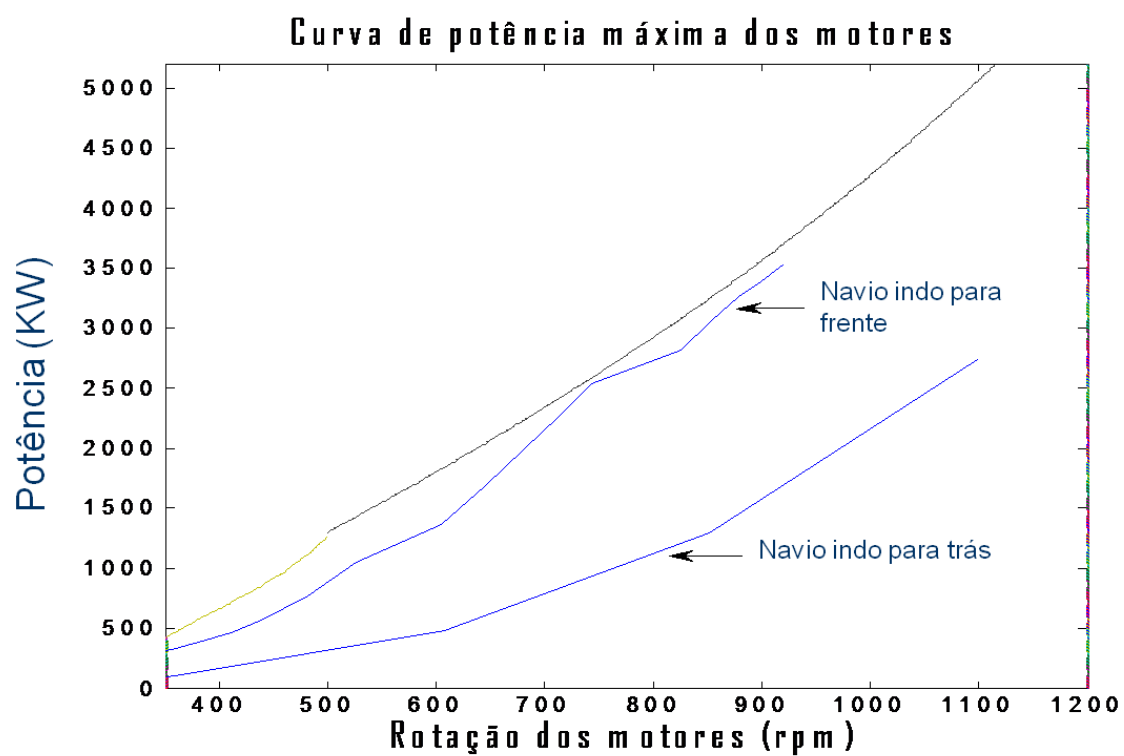


Figura 4.6 – Potência em função da rotação do motor.

5 Modelos Matemáticos

5.1 Dinâmica de navios

Neste capítulo, será discutida a modelagem não linear para a dinâmica de navios com base no modelo do Anexo E do Fossen [3]. Nesse desenvolvimento, serão abordadas equações padrões de movimento e derivadas hidrodinâmicas.

Hipóteses: - distribuição de massa homogênea em torno do eixo de simetria xz $\Rightarrow I_{xy} = I_{yz} = 0$;

- sistema com 4 graus de liberdade.

$$X = m(\dot{u} - vr - x_G r^2) \quad (5.1)$$

$$Y = m[\dot{v} + ur - z_G \dot{p} + x_G \dot{r}] \quad (5.2)$$

$$K = I_x \dot{p} - I_{xz} \dot{r} + m[-z_G(\dot{v} + ur)] \quad (5.3)$$

$$N = I_z \dot{r} - I_{xz} \dot{p} + m[x_G(\dot{v} + ur)] \quad (5.4)$$

Onde

$$X = X(u, v, r, \dot{u}, \emptyset, \delta, T_p) \quad (5.5)$$

$$Y = Y(v, r, \dot{v}, \dot{r}, \delta, p, \emptyset) \quad (5.6)$$

$$K = K(v, r, \dot{v}, \dot{r}, \delta, p, \emptyset) \quad (5.7)$$

$$N = N(v, r, \dot{v}, \dot{r}, \delta, p, \emptyset) \quad (5.8)$$

5.1.1 Equação de surge

Igualando as equações (5.1) e (5.5), apenas para *surge*, vem:

$$m(\dot{u} - vr - x_G r^2) = X(u, v, r, \dot{u}, \phi, \delta, T_p) \quad (5.9)$$

E desenvolvendo a equação (5.9):

$$X = X_{\dot{u}}\dot{u} + X_{vr}vr + X_{|u|u}|u|u + X_{rr}r^2 + (1 - t)T_p + X_{cc\delta\delta}c^2\delta^2 + X_{ext} \quad (5.10)$$

$$(m - X_{\dot{u}})\dot{u} = X_{|u|u}|u|u + (1 - t)T_p + (m + X_{vr})vr + X_{cc\delta\delta}c^2\delta^2 + (X_{rr} + mx_G)r^2 + X_{ext} \quad (5.11)$$

A Tabela 5.1 segue explicando os coeficientes das equações (5.10) e ((5.11):

Tabela 5.1 - Coeficientes e forças adotados no modelo

Coeficiente	Nomenclatura
$X_{\dot{u}}$	Massa adicional em surge
$X_{ u u}$	Resistência ao avanço
t	Coeficiente de redução da força propulsora
$X_{cc\delta\delta}$	Resistência devido à deflexão do leme
T_p	Torque do propulsor
c	Velocidade do fluxo no leme
$m + X_{vr}$	Força de arrasto adicional devido à combinação de sway e yaw
$X_{rr} + mx_G$	Força de arrasto adicional devido a yaw
X_{ext}	Distúrbios (vento, onda)

5.2 Equações normalizadas

Para facilitar o trabalho com as equações desenvolvidas acima, serão usadas equações normalizadas. A Tabela 5.2 mostra os parâmetros de normalização do Prime-System I, retirados do livro do Fossen[3].

Tabela 5.2 - Tabela de normalização (Prime-System I)

Variável	Parâmetro	Variável	Parâmetro
Comprimento	L	Velocidade linear	U
Massa	$\frac{\rho}{2}L^3$	Velocidade angular	$\frac{U}{L}$
Momento de inércia	$\frac{\rho}{2}L^5$	Aceleração linear	$\frac{U^2}{L}$
Tempo	$\frac{L}{U}$	Aceleração angular	$\frac{U^2}{L^2}$
Área	L^2	Força	$\frac{\rho}{2}U^2L^2$
Posição	L	Momento	$\frac{\rho}{2}U^2L^3$
Ângulo	1		

Onde

$$U = \sqrt{u^2 + v^2} \quad (5.12)$$

$$v = Uv' \quad (5.13)$$

$$r = \frac{U}{L}r' \quad (5.14)$$

Uma vez explanados os critérios de normalização, seguem as equações normalizadas:

$$(m' + m'_x)\dot{u}' - (m' + m'_y)v'r' = X' \quad (5.15)$$

$$(m' + m'_y)\dot{v}' + (m' + m'_x)u'r' + m'_y\alpha'_y\dot{r}' - m'_yl'_y\dot{p}' = Y' \quad (5.16)$$

$$(I'_x + J'_x)\dot{p}' - m'_yl'_y\dot{v}' - m'_xl'_x\dot{u}'r' + W'\overline{GM}'\phi' = K' \quad (5.17)$$

$$(I'_z + J'_z)\dot{r}' + m'_y\alpha'_y\dot{v}' = N' - Y'\chi_G' \quad (5.18)$$

Onde W é o peso (N) devido ao deslocamento do Navio e GM é a distância do centro de gravidade ao centro de carena do Navio. Além disso, m'_x , m'_y , J'_z e J'_x são as massas e momentos de inércia adicionais, nas direções x e y , em relação aos eixos z e x , respectivamente. Já α'_y corresponde à coordenada em x de m'_y , l'_x e l'_y correspondem às coordenadas em z de m'_x e m'_y respectivamente.

$$X' = X'(u') + (1 - t)T' + X'_{vr}v'r' + X'_{vv}v'^2 + X'_{rr}r'^2 + X'_{\phi\phi}\phi'^2 + c_{RX}F'_N \sin\delta \quad (5.19)$$

$$Y' = Y'_vv' + Y'_rr' + Y'_pp' + Y'_\phi\phi' + Y'_{vvv}v'^3 + Y'_{rrr}r'^3 + Y'_{vvr}v'^2r' + Y'_{vrr}v'r'^2 + Y'_{vv\phi}v'^2\phi' + Y'_{v\phi\phi}v'\phi'^2 + Y'_{rr\phi}r'^2\phi' + Y'_{r\phi\phi}r'\phi'^2 + (1 + a_H)F'_N \cos\delta \quad (5.20)$$

$$K' = K'_vv' + K'_rr' + K'_pp' + K'_\phi\phi' + K'_{vvv}v'^3 + K'_{rrr}r'^3 + K'_{vvr}v'^2r' + K'_{vrr}v'r'^2 + K'_{vv\phi}v'^2\phi' + K'_{v\phi\phi}v'\phi'^2 + K'_{rr\phi}r'^2\phi' + K'_{r\phi\phi}r'\phi'^2 - (1 + a_H)Z'_R F'_N \cos\delta \quad (5.21)$$

$$N' = N'_vv' + N'_rr' + N'_pp' + N'_\phi\phi' + N'_{vvv}v'^3 + N'_{rrr}r'^3 + N'_{vvr}v'^2r' + N'_{vrr}v'r'^2 + N'_{vv\phi}v'^2\phi' + N'_{v\phi\phi}v'\phi'^2 + N'_{rr\phi}r'^2\phi' + N'_{r\phi\phi}r'\phi'^2 + (x'_H + a_H x'_H)F'_N \cos\delta' \quad (5.22)$$

Da equação (5.19) a (5.22), aparecem variáveis como X'_{vr} , que são as derivadas hidrodinâmicas. Essa variável significa o coeficiente que atua em X' devido às variáveis v e r .

O termo $X'(u')$ representa a resistência ao avanço, calculada no Item 4.

T' refere-se à força propulsora (empuxo), modelada pela equação (4.4), normalizada segundo o critério da Tabela 5.2.

No que tange aos termos devido ao ângulo de leme, c_{RX} é uma constante que depende da forma do leme; a_H é uma constante que relaciona a força de resistência no leme devido ao seu ângulo com o esforço de reação que essa força causará no

casco; z_R' é a coordenada normalizada no eixo z do leme; e x_H' é a coordenada normalizada no eixo x que a reação no casco é aplicada.

$$F'_N = -\frac{6,13\Delta}{\Delta + 2,25} \cdot \frac{A_R}{L^2} \cdot (u_R'^2 + v_R'^2) \cdot \text{sen}(\alpha_R) \quad (5.23)$$

Onde A_R é a área do leme e Δ é a sua razão de aspecto.

$$u_R' = u_P' \varepsilon \sqrt{1 + \frac{8\kappa K_T}{\pi J^2}} \quad (5.24)$$

Onde u_R' é a velocidade na direção x do fluxo no leme normalizada; e κ é uma constante referente ao hélice.

$$u_P' = \cos(v) \left((1 - w) + \tau((v + x_P'r)^2 + c_{Pv}v + c_{Pr}r) \right) \quad (5.25)$$

Onde u_P' é a velocidade do fluxo no propulsor normalizada; w é o coeficiente de esteira; τ , c_{Pv} e c_{Pr} são constantes ligadas a condições de cavitação do propulsor; e x_P' corresponde a coordenada normalizada do propulsor em relação ao eixo x.

$$J = \frac{u_P U}{nD} \quad (5.26)$$

Onde J é o número de avanço, n é a rotação do eixo e D é o diâmetro do hélice.

$$v_R' = \gamma v + c_{Rr} + c_{Rrrr}r^3 + c_{Rrrv}r^2v \quad (5.27)$$

Onde v_R' é a velocidade na direção y do fluxo no leme normalizada; γ , c_{Rr} , c_{Rrrr} e c_{Rrrv} são constantes relacionadas à forma do leme.

$$\alpha_R = \delta + \tan\left(\frac{v_R'}{u_R'}\right)^{-1} \quad (5.28)$$

5.3 Derivadas hidrodinâmicas

As derivadas hidrodinâmicas que se seguem foram modeladas de acordo com Bertram[4]. Todas as outras derivadas hidrodinâmicas e constantes do modelo foram

mantidas iguais ao modelo do portacontainer do Anexo E do Fossen[3], pois para determiná-las seriam necessários testes com protótipos em tanques de manobras.

$$m'_x = \frac{m'}{\pi \sqrt{\frac{L^3}{V} - 14}} = 1,1221 \cdot 10^{-4} \quad (5.29)$$

$$m'_y = Y'_v = -\pi \left(\frac{T_s}{L}\right)^2 \left(1 + 0,16 C_B \frac{B}{T_s} - 5,1 \left(\frac{B}{L}\right)^2\right) = 5,839 \cdot 10^{-3} \quad (5.30)$$

$$\alpha'_y = \frac{Y'_r}{m'_y} = -\frac{\pi}{m'_y} \left(\frac{T_s}{L}\right)^2 \left(0,67 \frac{B}{L} - 0,0033 \left(\frac{B}{T_s}\right)^2\right) = 4,8217 \cdot 10^{-2} \quad (5.31)$$

$$Y'_v = -\pi \left(\frac{T_s}{L}\right)^2 \left(1 + 0,4 C_B \frac{B}{T_s}\right) = -8,435 \cdot 10^{-3} \quad (5.32)$$

$$Y'_r = -\pi \left(\frac{T_s}{L}\right)^2 \left(-0,5 + 2,2 \frac{B}{L} - 0,08 \frac{B}{T_s}\right) = 2,3886 \cdot 10^{-3} \quad (5.33)$$

$$N'_v = -\pi \left(\frac{T_s}{L}\right)^2 \left(0,5 + 2,4 \frac{T_s}{L}\right) = -3,0534 \cdot 10^{-3} \quad (5.34)$$

$$N'_r = -\pi \left(\frac{T_s}{L}\right)^2 \left(0,25 + 0,039 \frac{B}{T_s} - 0,056 \frac{B}{L}\right) = -1,9874 \cdot 10^{-3} \quad (5.35)$$

5.4 Vento

Sendo o Navio no rumo ψ , o sentido e intensidade do vento ψ_w e V_w , como mostra a Figura 5.1, vem:

$$u_R = V_w \cos \gamma_R - u \quad (5.36)$$

$$v_R = V_w \sin \gamma_R - v \quad (5.37)$$

$$V_R = \sqrt{u_R^2 + v_R^2} \quad (5.38)$$

$$\gamma_R = \arctan \left(\frac{v_R}{u_R} \right) \quad (5.39)$$

Segundo OCIMF[5]

$$X_w = \hat{C}_{X_w} \cos(\gamma_R) \frac{\rho_a}{7.6} V_R^2 A_T \quad (5.40)$$

$$Y_w = \hat{C}_{Y_w} \sin(\gamma_R) \frac{\rho_a}{7.6} V_R^2 A_L \quad (5.41)$$

$$N_w = \hat{C}_{N_w} \sin(2\gamma_R) \frac{\rho_a}{7.6} V_R^2 A_L L \quad (5.42)$$

Onde $\hat{C}_{X_w} \in \{-1, -0.8\}$, $\hat{C}_{Y_w} \in \{-1, -0.7\}$ e $\hat{C}_{N_w} \in \{-0.2, -0.05\}$. Dessa forma, foram utilizados os valores do meio dos intervalos no modelo.

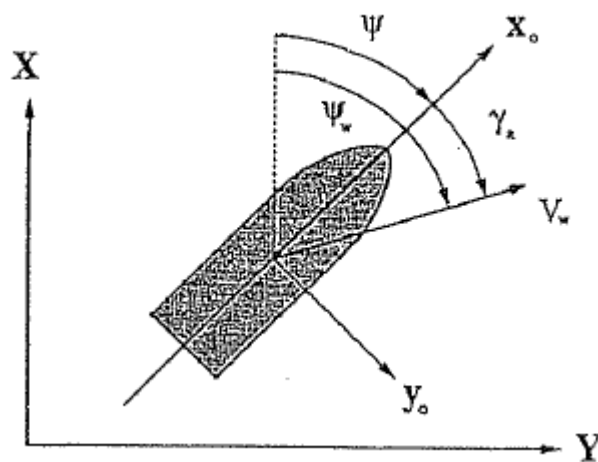


Figura 5.1 - Ângulos entre o Navio e o vento

5.5 Arquivo calc_derivadas.m

Esse arquivo é utilizado pelo Matlab e é através dele que as equações do item 5 são processadas. O arquivo encontra-se no Anexo A deste trabalho.

Já o Anexo R refere-se ao distúrbio causado pelo vento, utilizado pelo “calc_derivadas.m”.

6 Desempenho do modelo

A Tabela 6.1 tem a função de mostrar a relação entre as demandas entre os motores, desde que sejam iguais, com a velocidade assumida pelo navio. Para uma conversão de nós para m/s, deve-se multiplicar o valor por 0,5144.

Tabela 6.1 – Velocidade do navio em função de PCL.

PCL (Power Control Level)	velocidade (nós)
10	4,8
20	7,0
30	9,2
40	11,5
50	14,2
60	15,9
70	18,9
80	22,0
90	22,4
100	26,1

6.1 Motores

A Figura 6.1 mostra o desempenho do MCP acelerando até o navio atingir 12 nós (6,17 m/s). Nota-se que o motor mantém uma rotação constante até a embreagem engrazar. Então sofre uma queda de rotação devido ao incremento de torque; acelera até cerca de 500 rpm, fornecendo torque necessário para a aceleração do navio; e por fim estabiliza em cerca de 380 rpm.

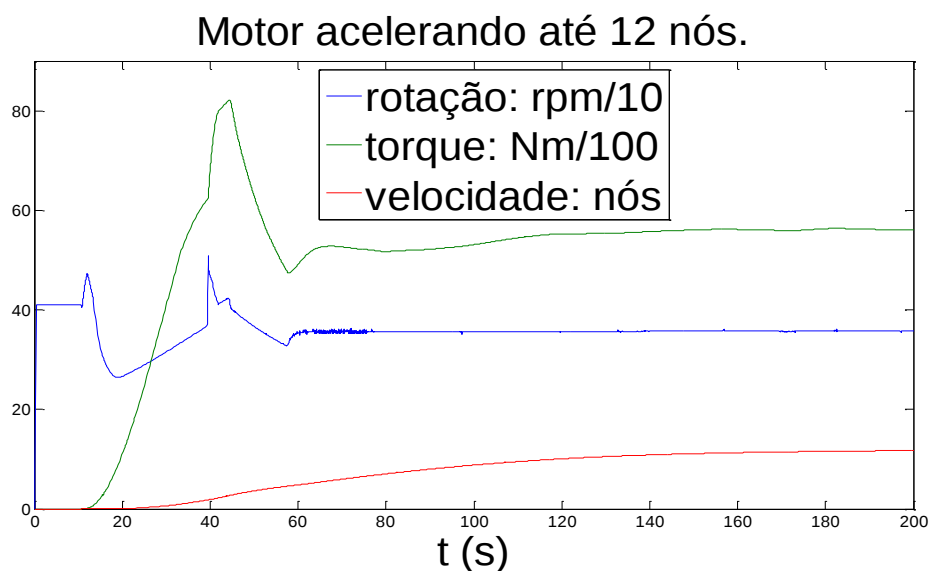


Figura 6.1 – Desempenho do motor.

6.2 Desempenho da planta

6.2.1 Curva de giro

A Figura 6.2 mostra o desempenho do navio com quatro configurações diferentes de leme.

Curvas de giro com 12 nós de velocidade.

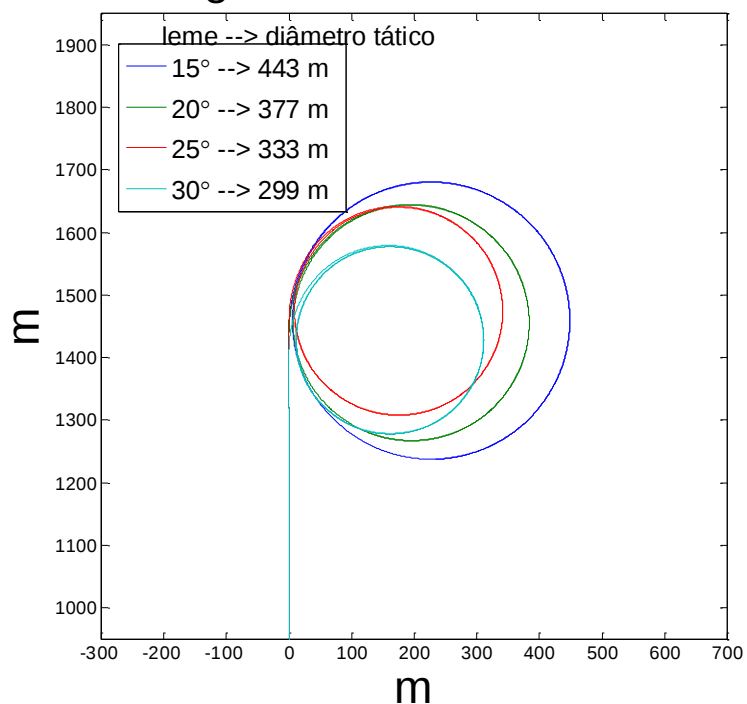


Figura 6.2 - Curvas de giro

A Figura 6.3 mostra que a velocidade de *surge* decresce quando o navio inicia o giro, além da velocidade de *sway* estabilizar em um valor positivo, ou seja, aponta para fora da curva de giro.

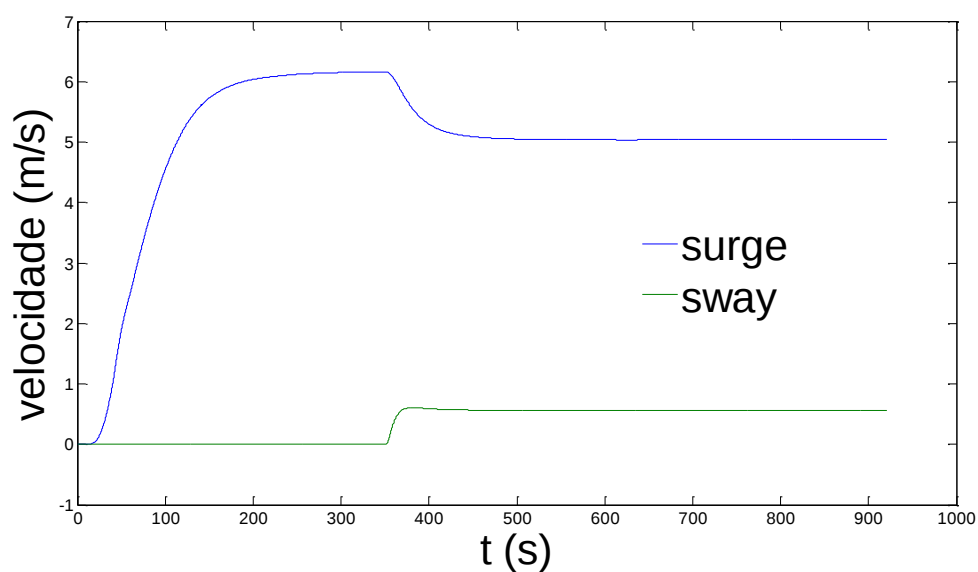


Figura 6.3 – Velocidades com 15° de leme (BB).

A Figura 6.4 ilustra o comportamento do movimento de *roll* do navio por ocasião do giro, a uma velocidade de 12 nós (6,17 m/s), que se estabiliza em cerca de 0,61°.

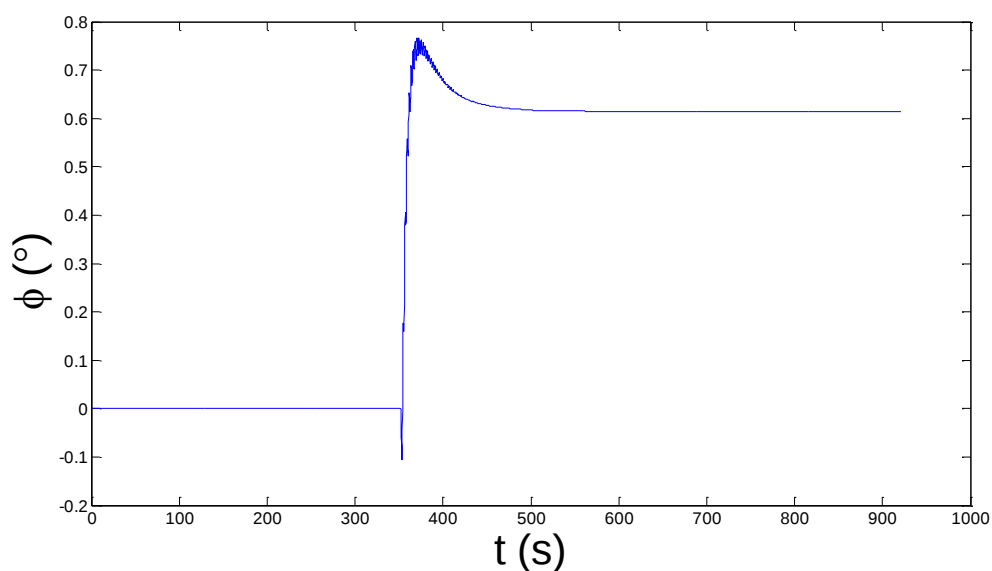


Figura 6.4 – Roll com leme de 15° BB.

6.2.2 Manobra de zig-zag

Tem caráter comparativo das dinâmicas de classes de navios. A Figura 6.5 mostra o seu princípio de funcionamento e a Figura 6.6 mostra o deslocamento do navio em “zig-zag”.

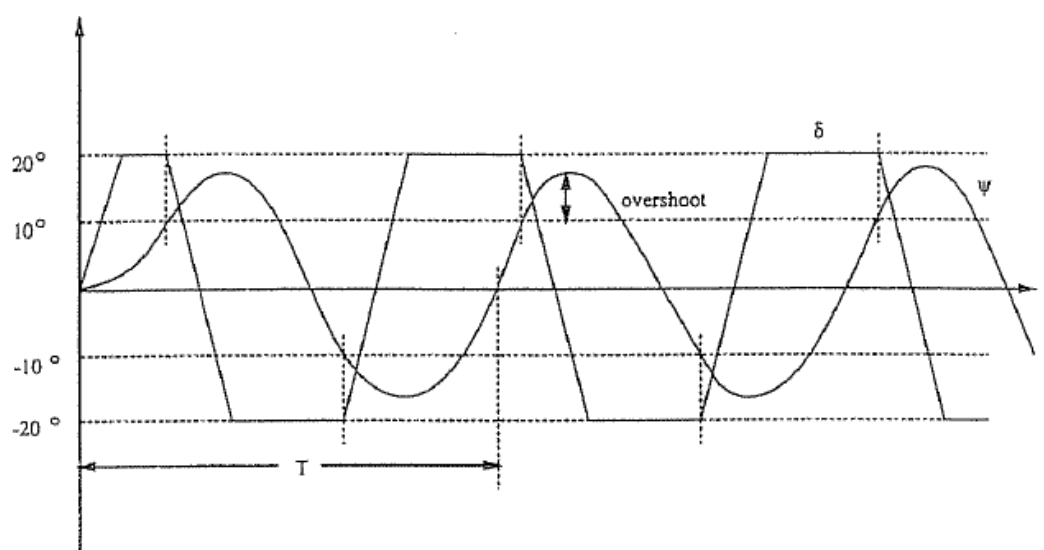


Figura 6.5 - Característica da manobra de zig-zag

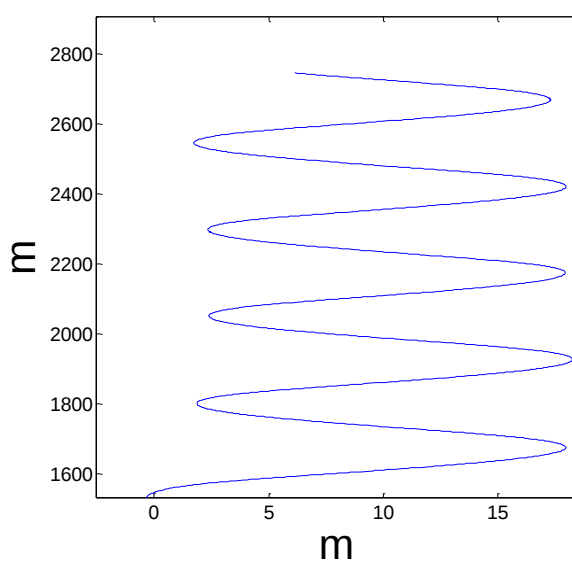


Figura 6.6 - Deslocamento em zig-zag

Da Figura 6.7, é possível descobrir uma característica importante do navio, que é a constante de tempo (T). Posteriormente, T será usado no projeto do controle de trajetória do navio.

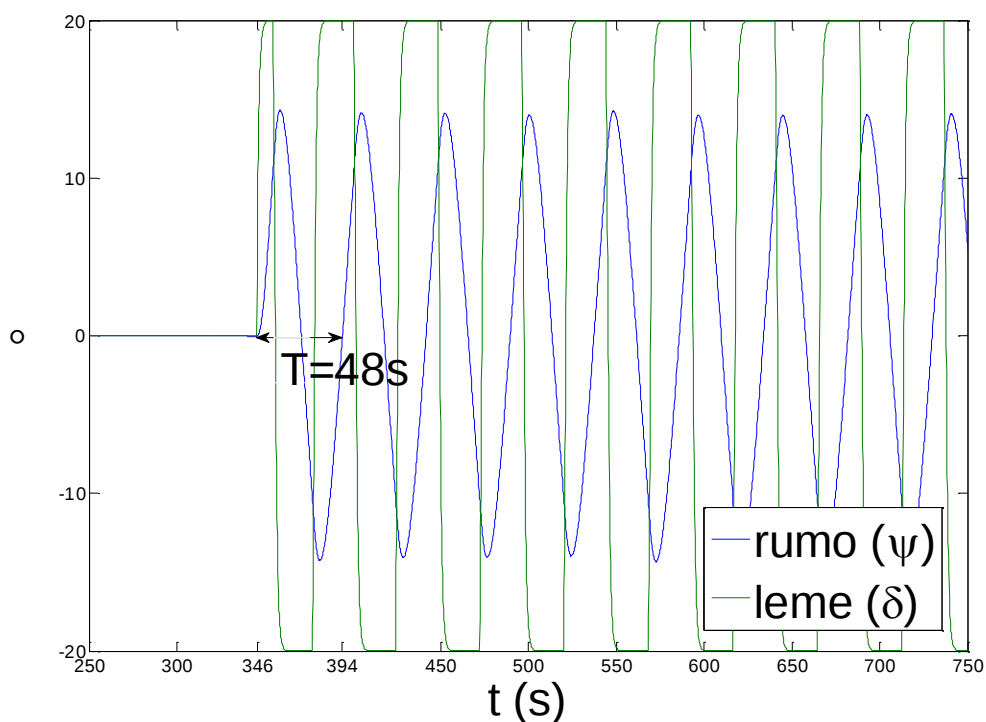


Figura 6.7 - Resposta do navio em “zig-zag”.

6.2.3 Parada em emergência

A parada em emergência funciona de forma a solicitar -100% de demanda nos dois eixos até a velocidade do navio chegar a 3 m/s, quando, então, solicita-se 0% aos motores. A Figura 6.8 mostra a dinâmica do navio quando utilizada a parada em emergência.

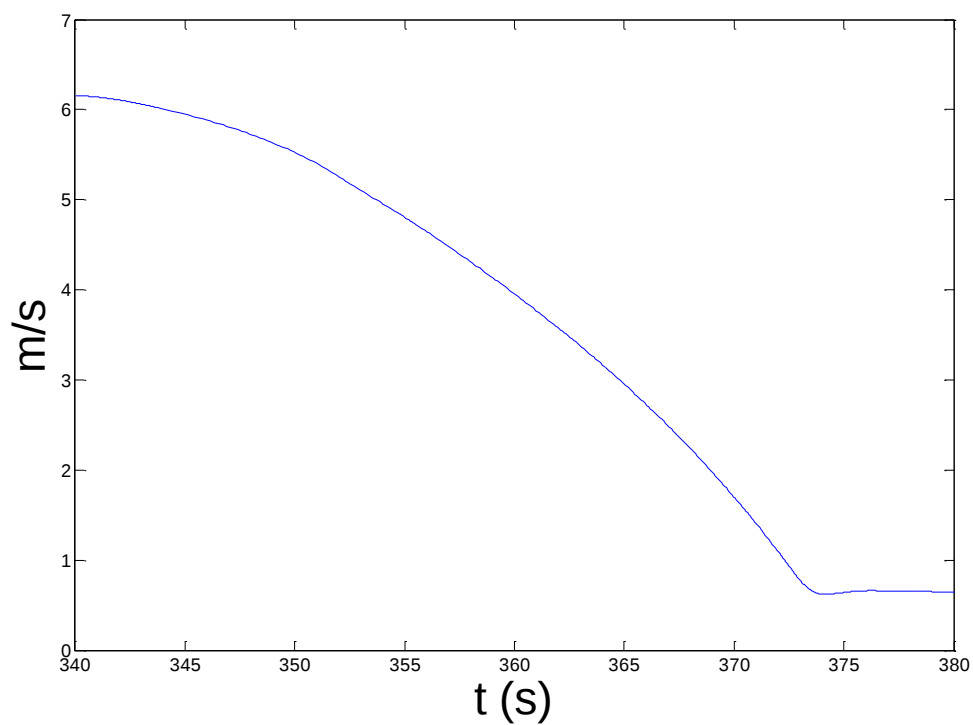


Figura 6.8 – Parada em emergência (velocidade).

Já a Figura 6.9 mostra que se o navio estiver navegando a 12 nós (6,17 m/s), só conseguirá parar o navio cerca de 140 m de distância do ponto de início da frenagem.

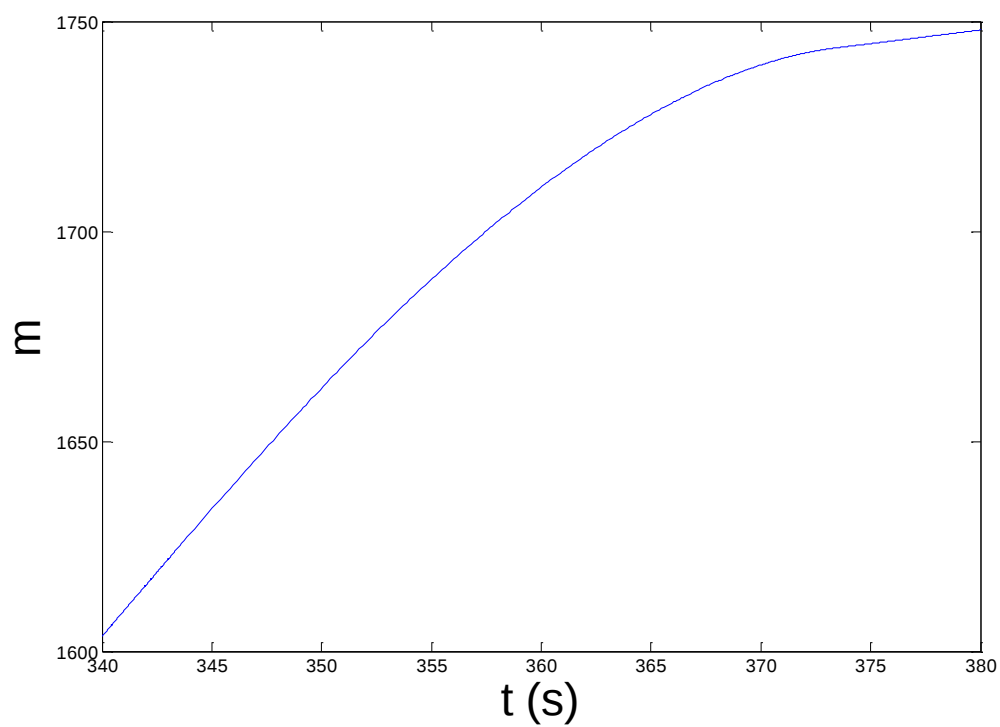


Figura 6.9 – Parada em emergência (distância).

7 Controle de trajetória

Após ser modelada a planta do modelo, o próximo passo é implementar o piloto automático.

7.1 Resposta dinâmica da planta

A Figura 7.1 mostra a resposta dinâmica da planta e uma aproximação por uma função de transferência. Essa aproximação deu-se através de uma rotina de otimização dada pelo arquivo “otimiza.m” do Anexo B.

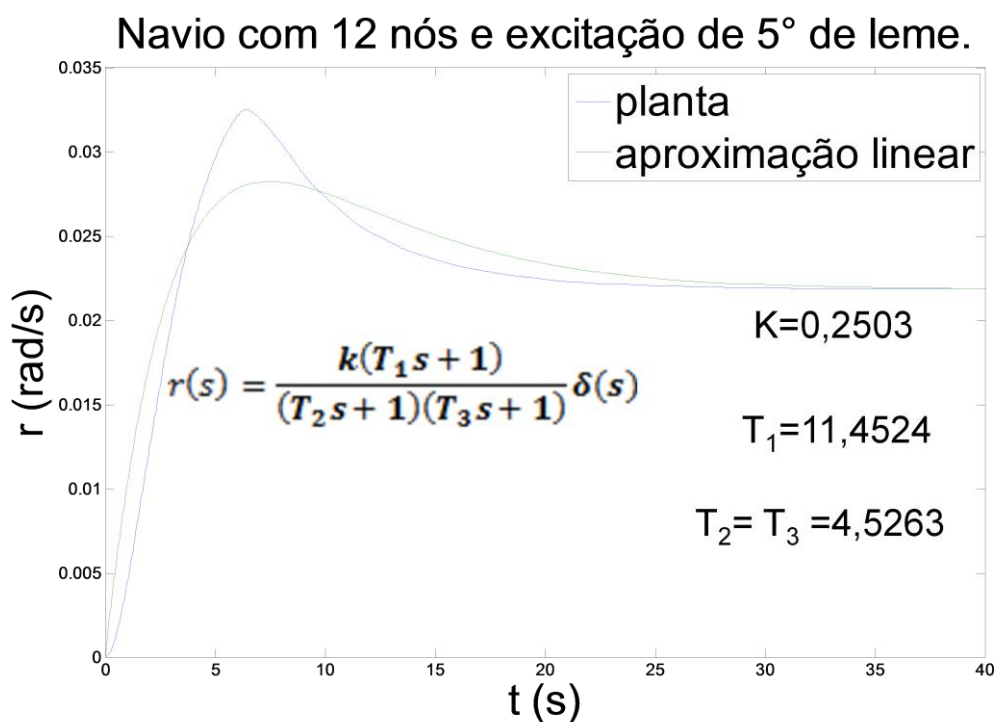


Figura 7.1 – Resposta dinâmica da planta.

7.2 Máquina do leme

A Figura 7.2 mostra a implementação da máquina do leme. A chave do subsistema, de acordo com o governo, se referencia de leme virá do piloto automático ou do leme manual. Esse sistema tem por limites a velocidade máxima de $15^\circ/\text{s}$ e 40° de ângulo de leme máximo.

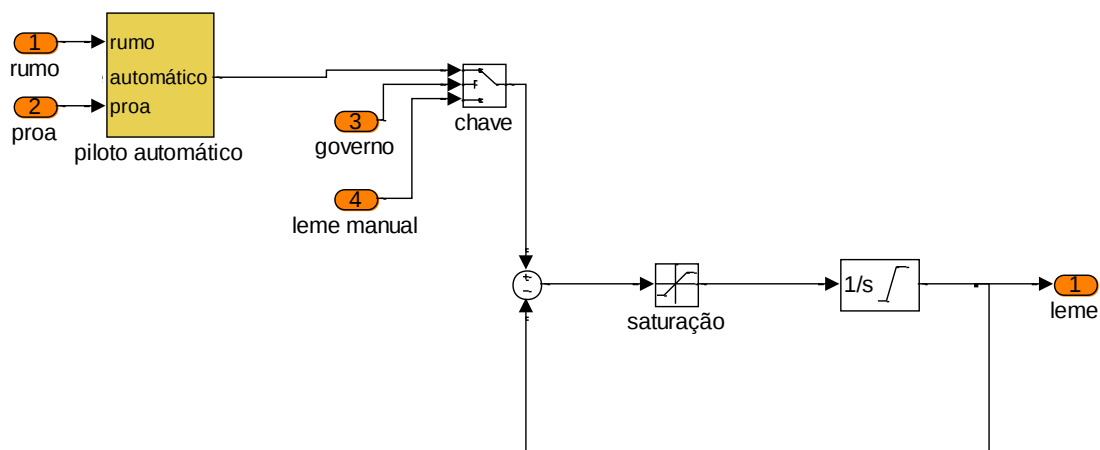


Figura 7.2 – Máquina do leme.

7.3 Controle PD

Inicialmente, foi implementado no leme um controlador PD.

$$\delta(s) = (K_d s + K_p) \psi(s) \quad (7.1)$$

$$r(s) = s \psi(s) = \frac{K(T_1 s + 1)}{(T_2 s + 1)(T_3 s + 1)} (K_d s + K_p) \quad (7.2)$$

A Figura 7.3 representa, em diagrama de blocos, como as equações (7.1) e (7.2) se relacionam no modelo.

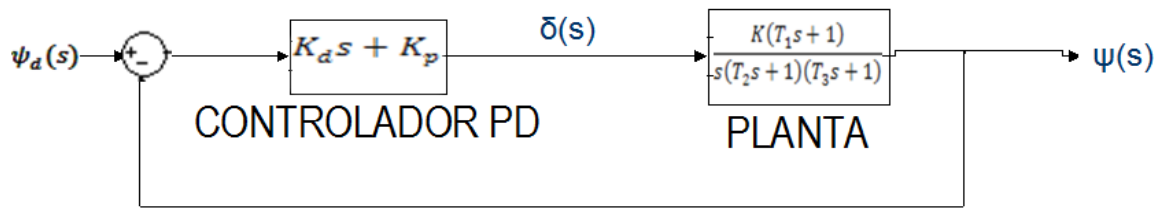


Figura 7.3 – Diagrama de blocos do controlador PD.

A equação (7.2) representa a função de transferência da planta em malha aberta.

$$\psi(s) = \frac{K(T_1 s + 1)(K_d s + K_p)}{T_2 T_3 s^3 + (T_1 K K_d + T_2 + T_3)s^2 + (K K_d + T_1 K K_p + 1)s + K K_p} \quad (7.3)$$

A equação (7.3) representa a função de transferência da planta em malha fechada. Em seguida, decompõe-se o polinômio do denominador:

$$den = (s^2 + 2\zeta\omega_n + \omega_n^2)(T_2 T_3 s + (T_1 K K_d + T_2 + T_3 - 2\zeta\omega_n T_2 T_3)) \quad (7.4)$$

Essa decomposição só será possível se a divisão entre o denominador da equação (7.3) e o termo de 2º da equação (7.4) for exata. Para tal, iguala-se o resto dessa divisão a zero:

$$\begin{bmatrix} T_1 K & (1 - 2\zeta\omega_n T_1)K \\ K & -\omega_n^2 T_1 K \end{bmatrix} \begin{bmatrix} K_p \\ K_d \end{bmatrix} = \begin{bmatrix} 2\zeta\omega_n(T_2 + T_3) - (4\zeta^2\omega_n^2 - \omega_n^2)T_2 T_3 \\ \omega_n^2(T_2 + T_3) - 2\zeta\omega_n^3 T_2 T_3 \end{bmatrix} \quad (7.5)$$

O valor de ζ escolhido foi de 0,9. Dessa forma, K_p e K_d variam em função de ω_n . Além disso,

$$\frac{1}{T} < \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} < \omega_\delta \quad (7.6)$$

Onde $\omega_\delta = 15^\circ/\text{s}$ e $T = 48\text{s}$ (manobra de “zig-zag”), então $0,028 < \omega_n < 0,3509$.

A Figura 7.4 mostra como a alocação de pólos varia com o valor de ω_n . Foram testados, no modelo completo, os seguintes valores de ω_n : 0,118 rad/s, 0,138 rad/s, 0,218 rad/s e 0,348. Esses valores de ω_n fornecem as seguintes razões entre a parte real dos pólos conjugados e o outro pólo: 9,311, 5,0794, 1,1052 e 0,3954, respectivamente.

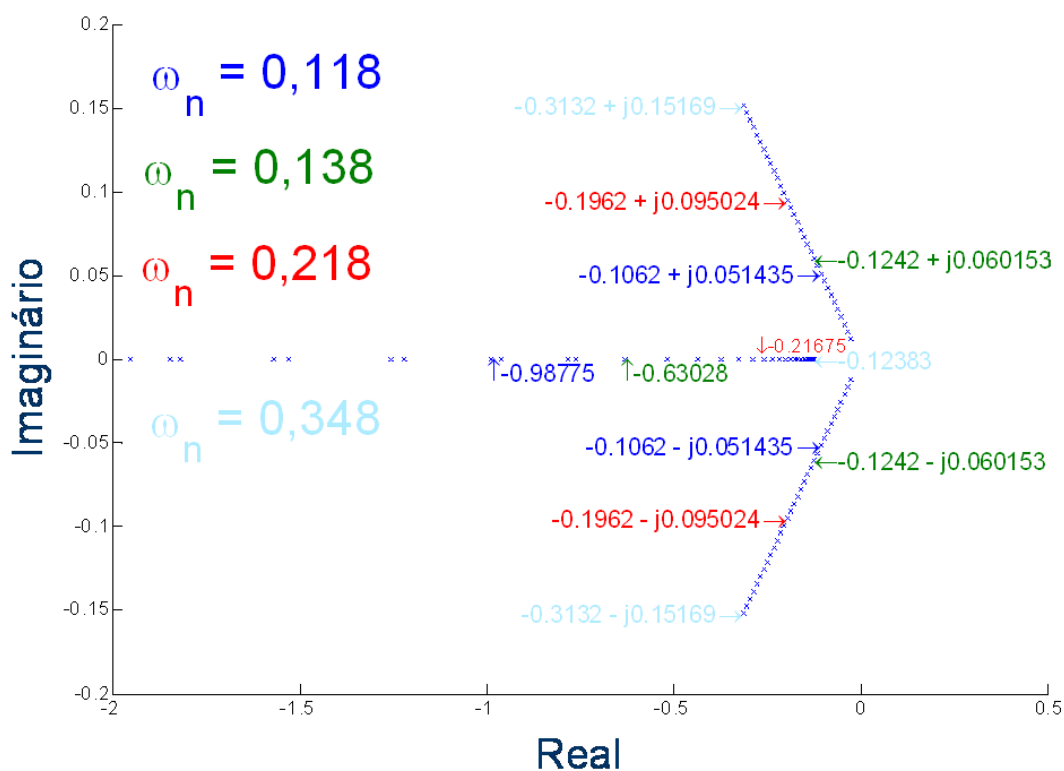


Figura 7.4 – Alocação de pólos.

A Figura 7.5 mostra a resposta dinâmica do modelo, navegando com uma velocidade de 12 nós (6,17 m/s), submetido a um controlador PD, variando-se o valor de ω_n e consequentemente de K_p e K_d .

Modelo completo submetido a um controlador PD.

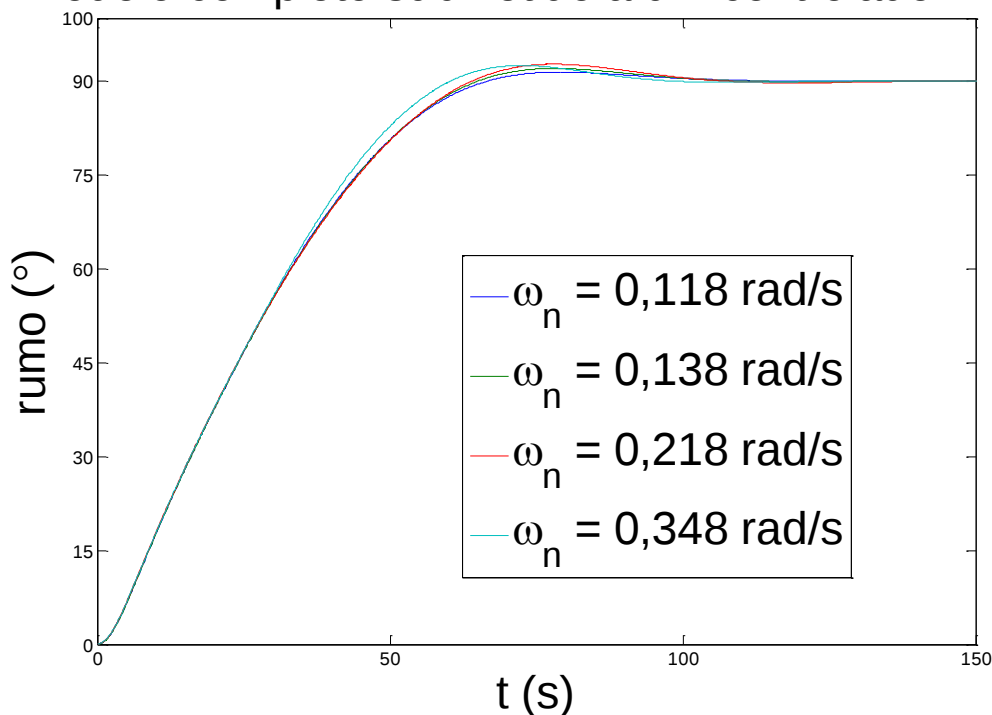


Figura 7.5 – Resposta dinâmica em função de ω_n .

A Figura 7.6 nada mais é que uma amplificação da Figura 7.5.

A quarta curva foi a primeira a ser descartada. Isso porque, para o referido valor de ω_n , os pólos conjugados deixam de ser dominantes, o que causa oscilação à resposta do modelo, o que se agravaria com a adição de um ganho integrador ao controlador.

O fato de a terceira curva suscitar certa oscilação ao modelo, até porque este está ainda muito influenciado pelo pólo de valor real, pois possui praticamente o mesmo valor da parte real dos pólos conjugados, faz com que esta curva seja descartada.

Tanto na primeira quanto na segunda curva, os pólos dominantes são os conjugados. Assim o valor de ω_n selecionado foi de 0,138 rad/s, pois este é o mínimo valor que garante que a parte real dos pólos conjugados seja, pelo menos, cinco vezes menor que o outro pólo.

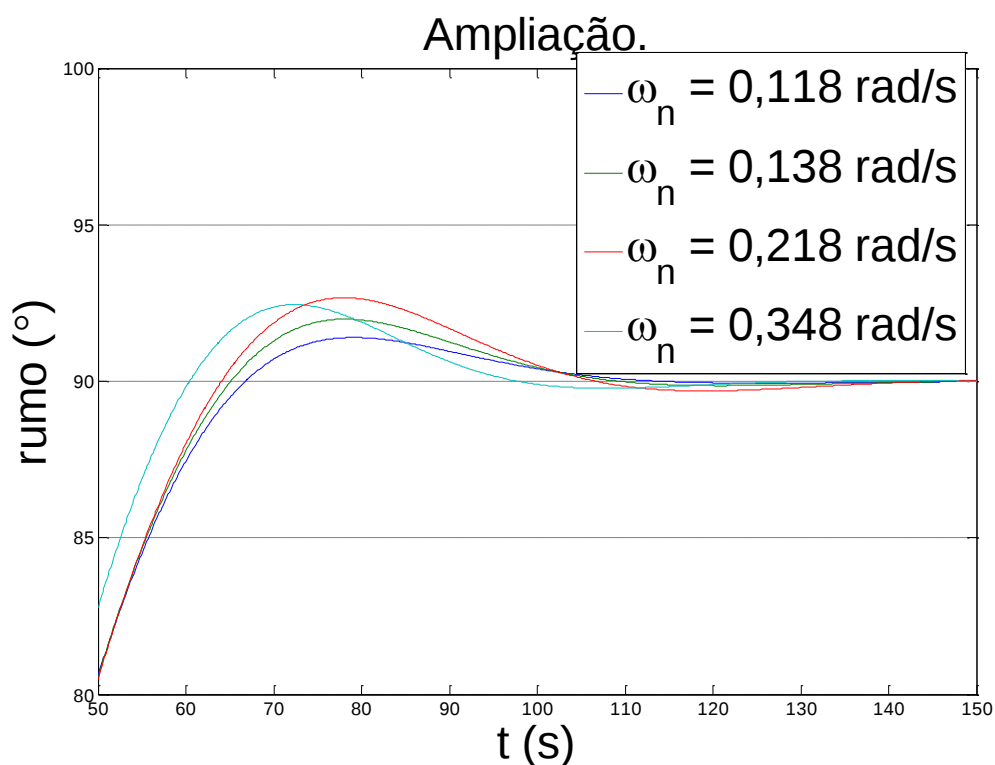


Figura 7.6 - Estabilização da planta (controlador PD).

Por fim, $K_p=0,9825$ e $K_d=3,1219$.

7.3.1 Inclusão do termo integrativo

Foi testado, no modelo completo, o navio rumando para o norte (0°) e, a partir de um certo instante, submetido a um vento da direção 90° e com 30 nós (15,43 m/s) de intensidade. A implementação de um controlador proporcional, integrativo e derivativo (PID), em detrimento a um proporcional e derivativo (PD), deve-se ao fato de que o modelo tem de ser capaz de se manter estável mesmo sob distúrbios, ou seja, anular erros de regime permanente. A Figura 7.7 mostra que um controlador PD não é capaz de anular esses erros. Mostra também que, a partir de $K_i=K_p/40$, o retorno para o rumo de referência acontece sem sobressinal. Dessa forma, adotou-se no modelo esse valor de K_i (0,0246). Além disso, a estratégia adotada de controle

prevê que o termo integrador somente atuará quando a diferença entre a proa e o rumo desejado for menor do que 10° . Essa estratégia elimina o sobressinal que aconteceria à resposta dinâmica da planta devido ao leme possuir um ângulo máximo de 40° .

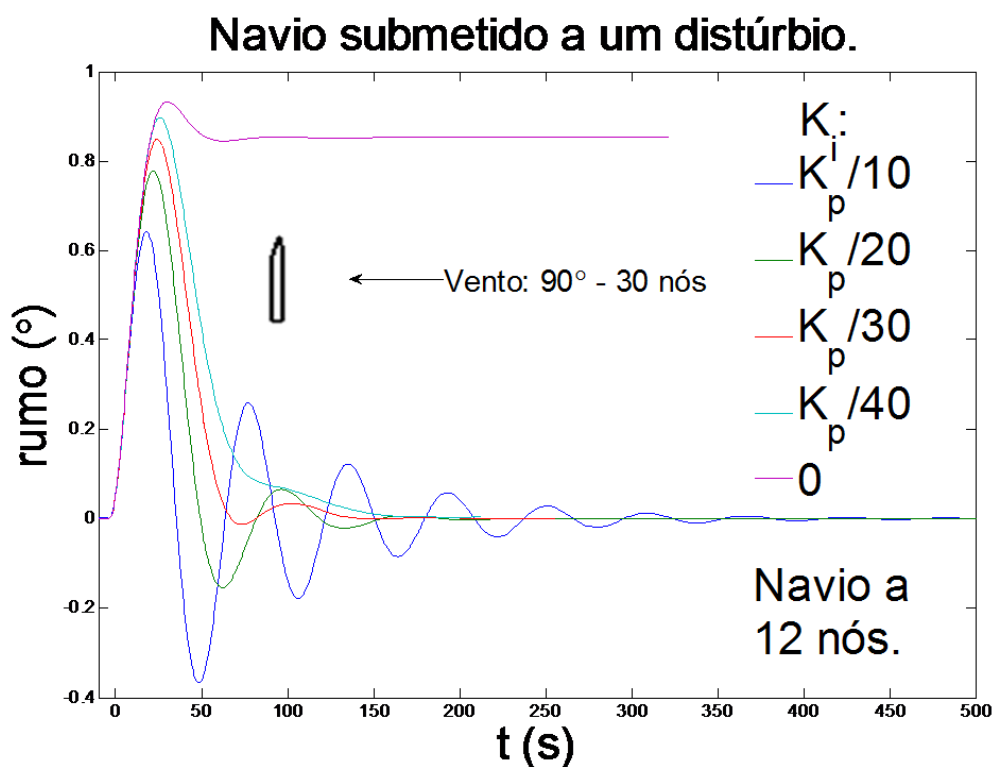


Figura 7.7 –Inclusão de K_i .

A Figura 7.8 mostra a implementação do piloto automático no modelo. Note que o ganho integrador só atuará quando o erro for menor do que 10° .

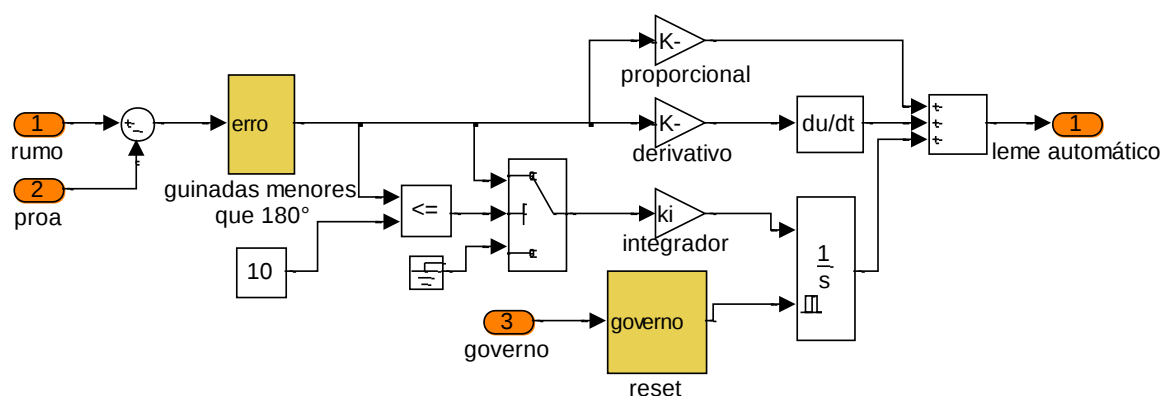


Figura 7.8 – Piloto automático.

7.4 Arquivo NPa_data.m

Esse arquivo, presente no Anexo N, carrega os valores de todas as variáveis usadas no modelo, exceto os coeficientes das derivadas hidrodinâmicas que dizem respeito à resistência ao arrasto, que é usado em “calc_derivadas.m”, além de carregar o arquivo “sim_NPa.mdl”, responsável pela simulação do modelo.

8 Resultados

8.1 Piloto automático

A Figura 8.1 mostra o navio seguindo para o rumo 90° de forma estável.

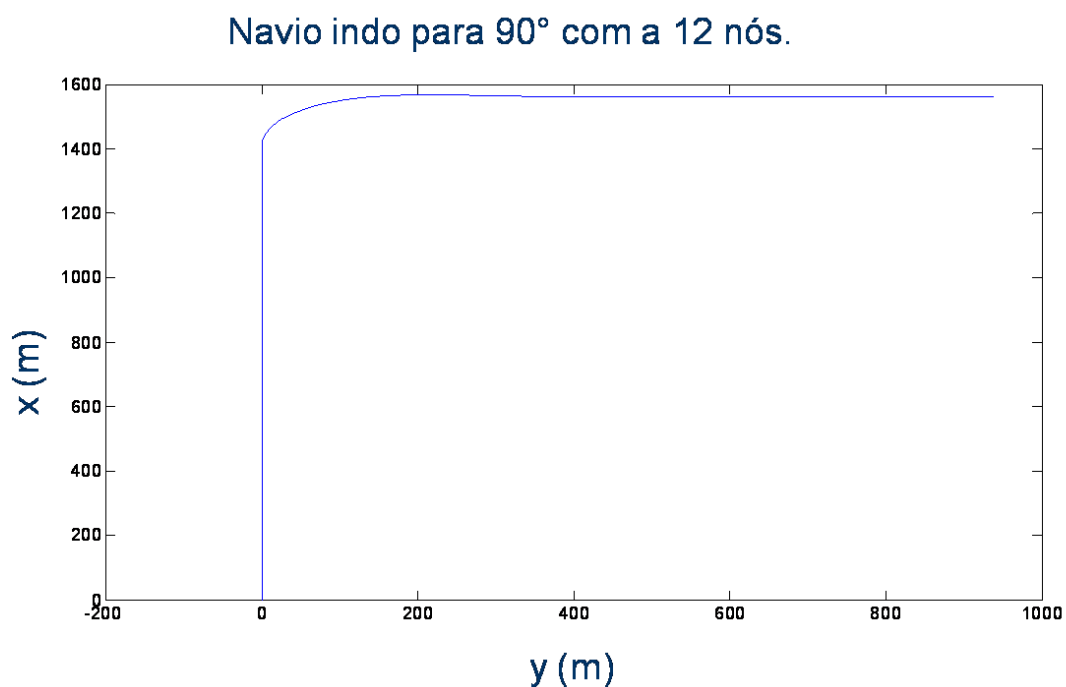


Figura 8.1 – Deslocamento para 90° .

Na Figura 8.2, observa-se o comportamento dinâmico do navio com um rumo de referência de 90° . É possível verificar que o controle de trajetória é estável e também o sobressinal foi pequeno, pelo fato de o termo integrador começar a atuar quando o erro for menor do que 10° . A diferença entre a resposta dinâmica do modelo completo e da aproximação linear se dá por:

- O navio possui um ângulo de leme máximo de 40° ;

- O leme possui uma velocidade máxima de $15^\circ/\text{s}$;
- A velocidade do navio não se mantém constante;
- Na aproximação linear, o controlador é PID e atua todo instante;

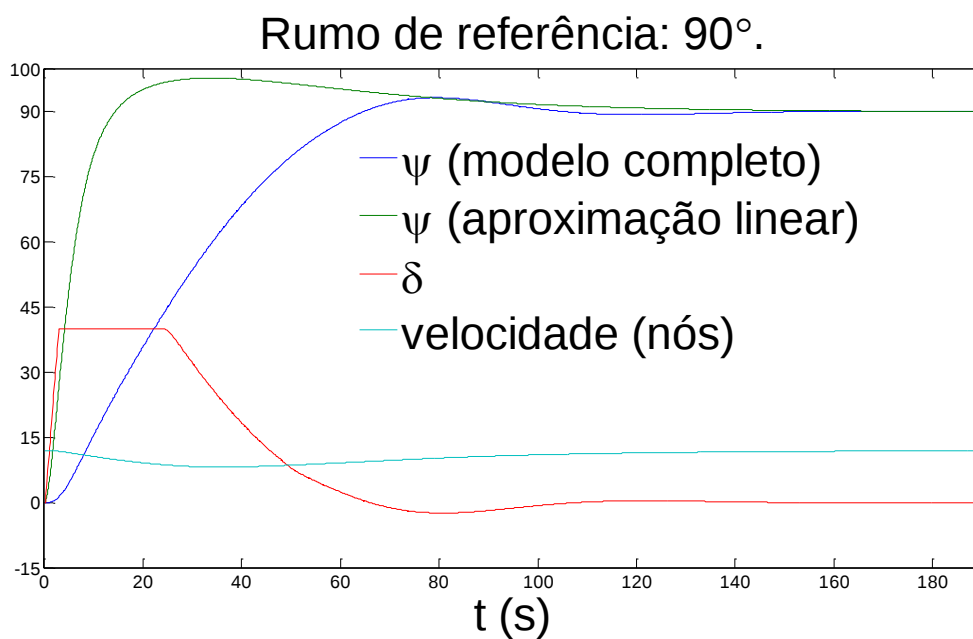


Figura 8.2 - Variação da proa para o rumo 90° .

A Figura 8.3 evidencia a velocidade máxima do leme de $15^\circ/\text{s}$ (reta com coeficiente angular constante) e ângulo máximo de 40° .

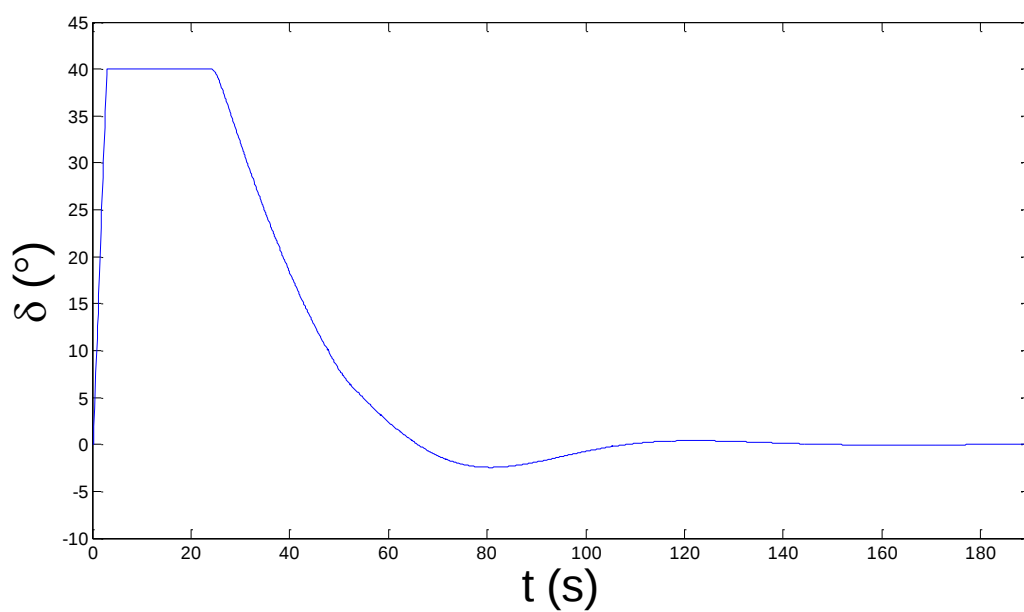


Figura 8.3 – Ângulo do leme.

A Figura 8.4 mostra o comportamento do roll durante a estabilização. Nota-se que quando o navio estabiliza no rumo pretendido, o roll estabiliza cerca de 0°.

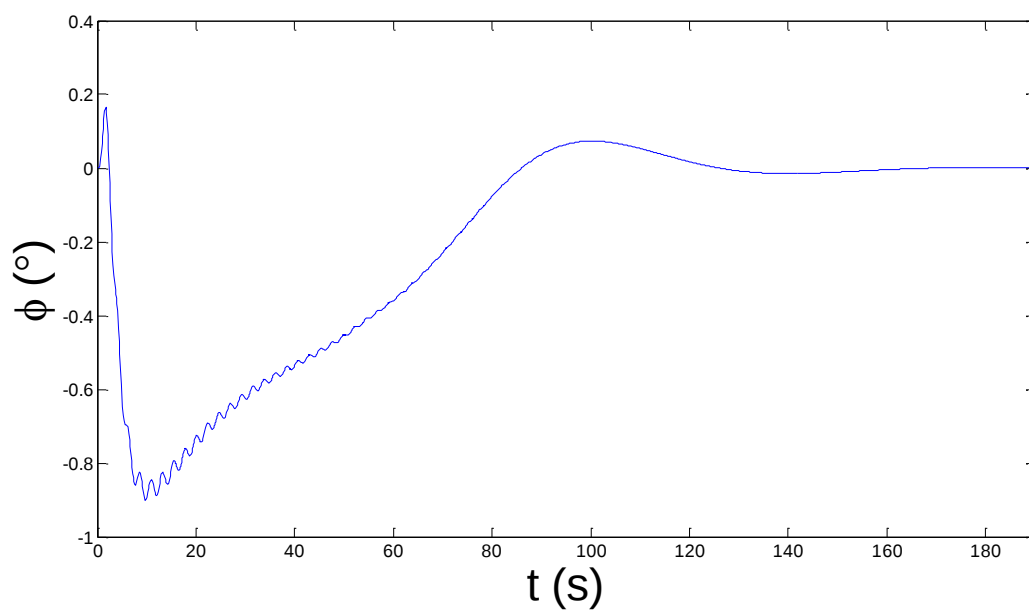


Figura 8.4 - Roll durante estabilização

8.1.1 Piloto automático com distúrbio

Na Figura 8.5, simula-se vento de intensidade de 30 nós (15,43 m/s) vindo do norte (0°). É possível notar que o leme estabiliza em um ângulo diferente de 0° , com a finalidade de compensar o distúrbio.

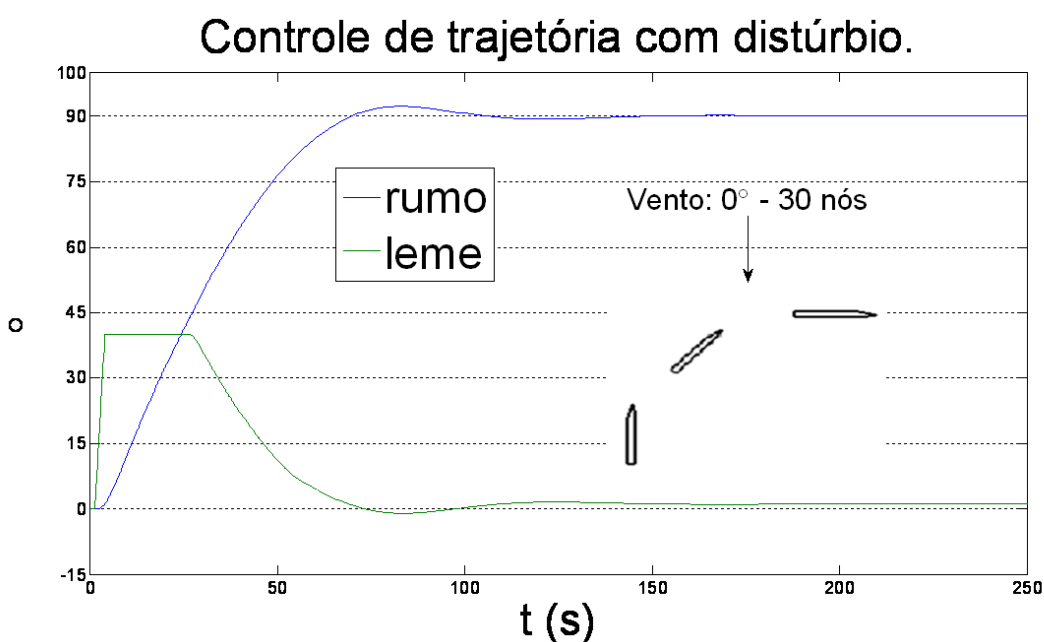


Figura 8.5 – Controle com distúrbio (Vento 0° - 30 nós).

8.1.2 Navio funcionando com um único eixo

Esse teste apenas visou verificar a eficiência do controlador. Foi imposto aos dois eixos com 100% de demanda, sendo simulada uma falta de óleo no acoplamento fluido (AF) de boreste (BE).

A Figura 8.6 mostra, a partir do instante que o AF de BE esvazia, o navio se deslocando para a direita por estar com um eixo só.

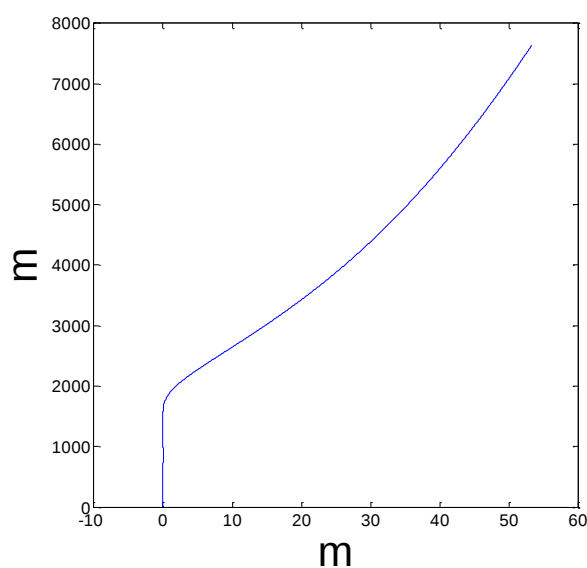


Figura 8.6 – Deslocamento para o norte (0°).

A Figura 8.7 mostra a queda que a velocidade de *surge* sofre quando ocorre a pane no AF.

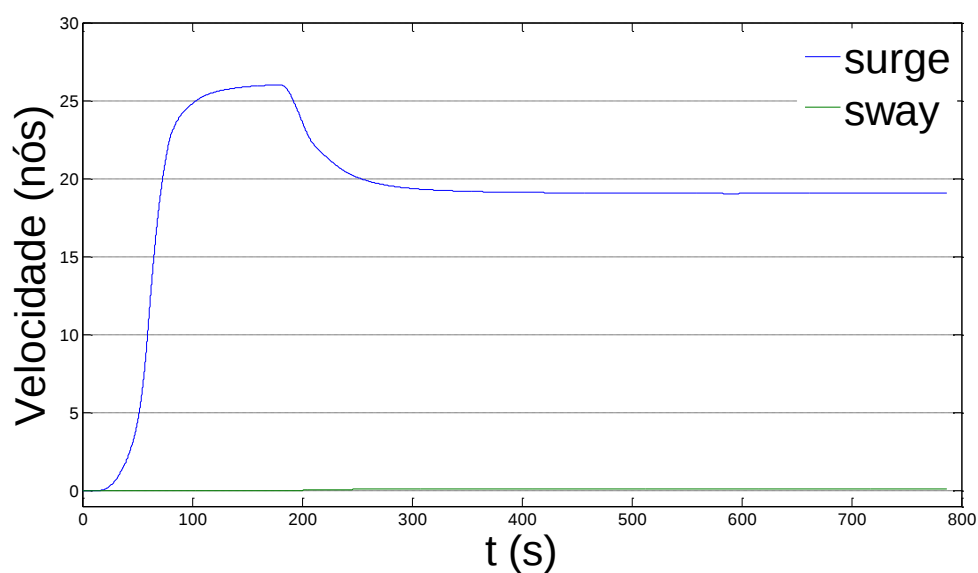


Figura 8.7 – Velocidades de *surge* e *sway*.

A Figura 8.8 mostra o pequeno aumento da velocidade de *sway*. Essa aceleração explica o deslocamento horizontal visto na Figura 8.6.

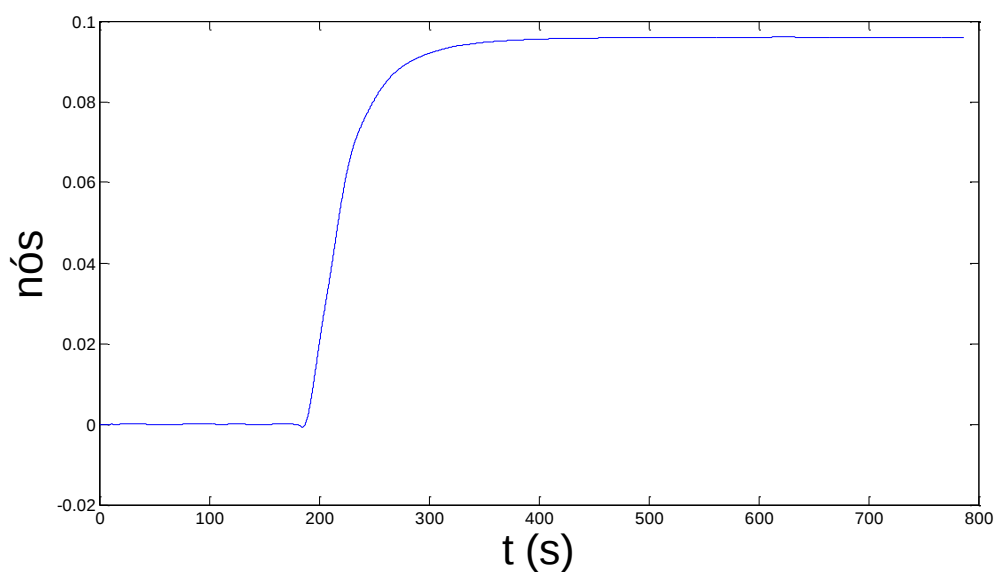


Figura 8.8 – Velocidade de sway.

A Figura 8.9 mostra que no instante que o eixo de BE pára, o navio começa girar para BE, contudo o piloto automático passa a atuar, não permitindo que a proa varie mais de 0,6°.

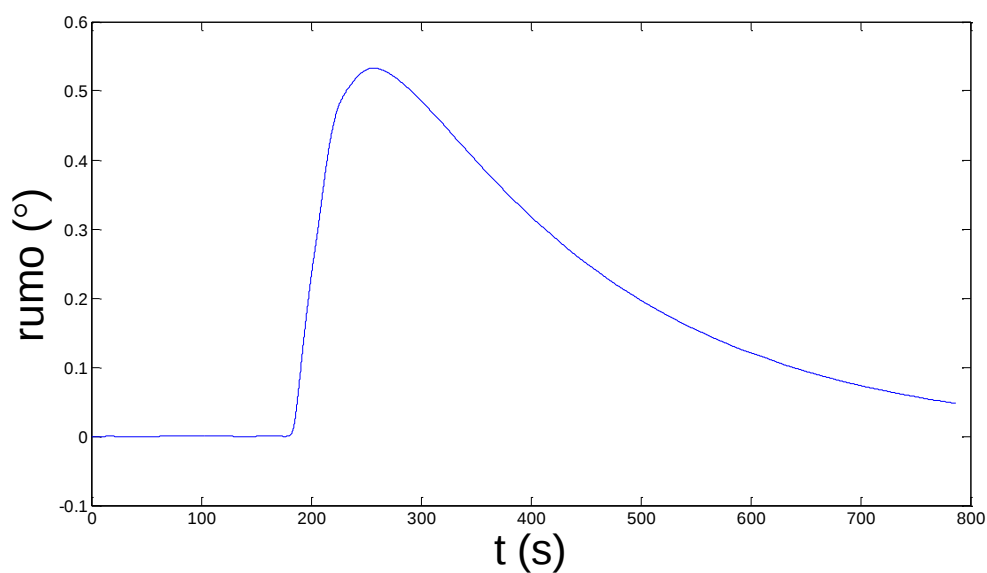


Figura 8.9 – Variação do rumo (pane em um eixo).

A Figura 8.10 mostra que, quando o eixo de BE pára de funcionar, o leme vai para bombordo (BB) para compensar o momento que se forma na popa do navio.

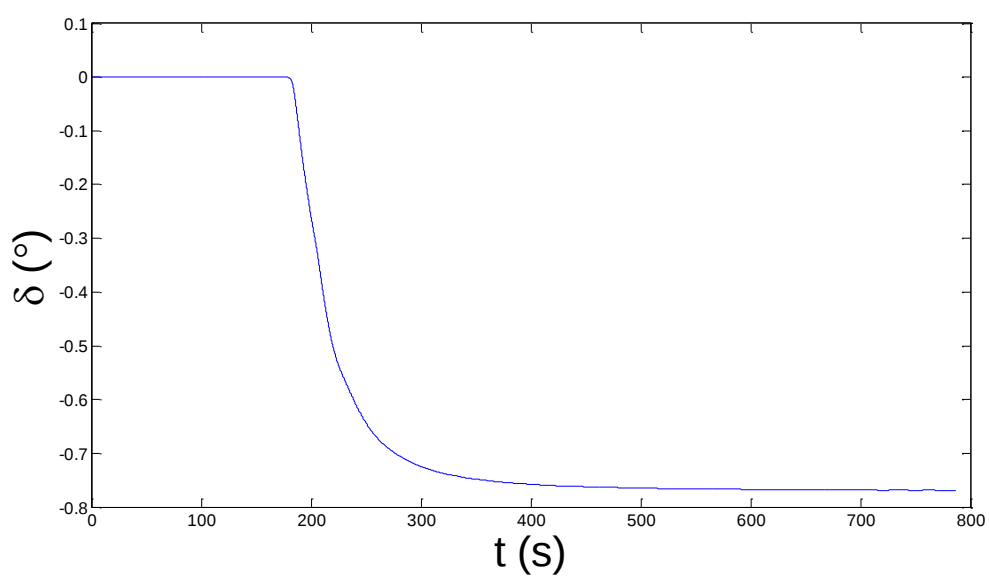


Figura 8.10 – Compensação do leme.

Desse modo, mesmo com um eixo só, o controlador é capaz de manter o rumo pretendido.

9 Conclusões

Os controles dos motores e do passo controlado mostraram-se capazes de fornecer o torque necessário para que o navio pudesse desenvolver sua velocidade máxima projetada: 25 nós (12,86 m/s);

O piloto automático mostra-se robusto, mesmo em situações em que o navio está sob influência de distúrbios, ou seja, o navio é capaz de manter sua trajetória mesmo sob um vento de velocidade maior do que a sua própria velocidade, e também funcionando com um eixo apenas.

Uma próxima etapa desse projeto seria construção de protótipo para ensaio em tanque de prova e dessa forma, obter as derivadas hidrodinâmicas de uma maneira mais precisa. Depois disso, realizar um novo ajuste do controlador. Por fim, implementar o controlador projetado no protótipo através de um microcontrolador embarcado.

10 Bibliografia

[1] Modelagem e Simulação do Sistema de Propulsão – Corveta Barroso. Laboratório de Automação e Controle da Escola Politécnica da USP.

[2] Teoria de voo – Fundamentos, disponível em <http://www.fpam.pt/Formacao/TecNiv1/tn1002NocoasAerodinamica.pdf>. Acessado em 16 de junho de 2010.

[3] FOSSEN, T. I.: “**Guindance and Control of Ocean Vehicles**”, University of Trondheim, Norway, 1994.

[4] BERTRAM, Volker: “**Practical Ship Hydrodynamics**”, University of Oxford, 2000.

[5] OCIMF (1977). “**Prediction of Wind and Current Loads on VLCCs**”, Oil Companies International Marine Forum, London, pp 1-77.

[6] OGATA, Katsuhiko. “**Engenharia de Controle Moderno**”, Editora ABDR, 4^o edição.

[7] Tradições do mar – uso costumes e linguagens, disponível em http://www.mar.mil.br/menu_v/tradicoes_do_mar/caracteristicas_navio.htm. Acessado em 12 de abril de 2010.

Anexo A - calc_derivadas.m

```

function [sys,x0,str,ts,simStateCompliance] =
calc_derivadas(t,x,u,flag,p1,p2)

switch flag,

    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Derivatives %
    %%%%%%%%%%%%%%%
    case 1,
        sys=mdlDerivatives(t,x,u);

    case 2,
        % sys=mdlUpdate(t,x,u);

    %%%%%%%%%%%%%%%
    % Update %
    %%%%%%%%%%%%%%%

    case 3,
        sys=mdlOutputs(t,x,u,p1,p2);

    case 4,
        %sys=mdlGetTimeOfNextVarHit(t,x,u);

    %%%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%%
    case 9,
        sys=mdlTerminate(t,x,u);

    %%%%%%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%%%%%
    otherwise
        DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl

%
%=====
===
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-
function.

```

```

%=====
===
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 6;
sizes.NumInputs = 15;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [-1 0];

% Specify the block simStateCompliance. The allowed values are:
% 'UnknownSimState', < The default setting; warn and assume
DefaultSimState
% 'DefaultSimState', < Same sim state as a built-in block
% 'HasNoSimState', < No sim state
% 'DisallowSimState' < Error out when saving or restoring the model sim
state
simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=====
===
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
===
%
function sys=mdlDerivatives(t,x,u)

```

```

sys = [];

% end mdlDerivatives

%
%=====
===
% mdlOutputs
% Return the block outputs.
%=====
===
%
function sys=mdlOutputs(t,x,u,p1,p2)

% derivadas hidrodinâmicas
%X          Y          N          K
Xvr = -.00311;   Yv = -8.435e-3; Nv = -3.0534e-3; Kv = .0003026;
Xvv = -.00386;   Yr = 2.3886e-3; Nr = -1.9879e-3; Kr = -.000063;
Xrr = .0002;     Yp = 0;         Np = .000213;   Kf = -.000021;
Xff = -.0002;    Yf = -.000063; Nf = -.0001424; Kp = -.0000075;
                Yvvv = -.109;   Nvvv = .001492; Kvvv = .002843;
                Yrrr = .00177; Nrrr = -.00229; Krrr = -.0000462;
                Yvvr = .0214;   Nvvr = -.0424; Kvvv = -.000588;
                Yvrr = -.0405;  Nvrr = .00156;  Kvrr = .0010565;
                Yvvf = .04605;  Nvvf = -.019058; Kvvf = -.0012012;
                Yvff = .00304;  Nvff = -.0053766; Kvff = -.0000793;
                Yrrf = .009325; Nrrf = -.0038592; Krrf = -.000243;
                Yrff = -.001368;Nrff = .0024195; Krff = .00003569;

% outros parâmetros
m = 5.2881e-3;
mx = 1.1221e-4;
my = 5.839e-3;
lx = .0313;
xG = 1.9947e-2;

M = p1;
L = p2;
AR = 1.82; % área do leme (m^2)
D = 1.7; % diâmetro do hélice (m)
g = 9.81; % aceleração da gravidade (m/s^2)
nabla = 476.42; % deslocamento (m^3)
rho=1025; % densidade da água do mar (kg/m^3)

U = sqrt(u(1)^2 + u(2)^2);
if U==0, U=.04; end
vx = u(1)/U;   epsilon = .97; cRr = -.156;   xR = -.5;
v = u(2)/U;   kk = .631;   cRrrr = -.275;   xH = -.48;
p = u(3)*L/U; cRrrv = 1.96;
r = u(4)*L/U; tau = 1.09;   cRX = .71;   Delta = 0.8; % razão de
aspecto do leme
n1 = (u(5)/60);
n2 = (u(6)/60); xp = -.526;   aH = .237;   d = 2/L;
fi = u(7);   cpv = 0;   zR = .033;
delta = u(8); cpr = 0;   GM = 1.81/L;

% resistência ao avanço

```

```

unos = u(1)*3600/1852; % velocidade em nós
if unos<15
    if u(1)<0, Xu=(1/2)*.56*rho*21.255*(u(1)^2)*2/(rho*(U*L)^2); %
21,255m^2 é a área do espelho de popa
    else Xu = -(.10100*unos^6 - 4.70999*unos^5 + 78.52611*unos^4 -
555.26441*unos^3 + 1828.04625*unos^2 - 1276.27612*unos)*2/(rho*(U*L)^2);
    end
else
    Xu=-(6.74916*unos^4 - 607.44525*unos^3 + 19111.93724*unos^2 -
233790.30655*unos + 981788.27925)*2/(rho*(U*L)^2);
end

Frvol=0.165*unos/(488^(1/6));
w=0.00343+0.11152*Frvol-0.27571*(Frvol^2)+0.16330*(Frvol^3)...
-0.02828*(Frvol^4);
t=0.43803-0.92242*Frvol+0.81950*(Frvol^2)-0.32145*(Frvol^3)...
+0.04659*(Frvol^4);

%distúrbios
dirw = u(9)*pi/180;Vw = u(10)*1852/3600;
psi = u(11);
[X,Y,N] = vento(dirw,Vw,vx*U,v*U,psi);
Xw = X^2/(rho*(U*L)^2);
Yw = Y^2/(rho*(U*L)^2);
Nw = N^2/(rho*U^2*L^3);

if u(5)==0, Q1=0;
    Kt1=0;
    uR1 = 0.0001;
else uP1=cos(v)*((1 - w) + tau*((v + xp*r)^2 + cpv*v + cpr*r));
    J1=uP1*U/(n1*D);
    Kt1 = u(12);
    Kq1 = u(14);
    Q1 = Kq1*rho*n1^2*D^5;
    uR1 = uP1*epsilon*sqrt(1 + 8*kk*abs(Kt1)/(pi*J1^2));
end

if u(6)==0, Q2=0;
    Kt2=0;
    uR2 = 0.0001;
else uP2=cos(v)*((1 - w) + tau*((v + xp*r)^2 + cpv*v + cpr*r));
    J2=uP2*U/(n2*D);
    Kt2 = u(13);
    Kq2 = u(15);
    Q2 = Kq2*rho*n2^2*D^5;
    uR2 = uP2*epsilon*sqrt(1 + 8*kk*abs(Kt2)/(pi*J2^2));
end

gama=.088;

vR = gama*v + cRr*r + cRrrr*r^3 + cRrrv*r^2*v;
alfaR1 = delta + atan(vR/uR1);alfaR2 = delta + atan(vR/uR2);
Fn1 = -((6.13*Delta)/(Delta + 2.25))*(AR/L^2)*(uR1^2 + vR^2)*sin(alfaR1);
Fn2 = -((6.13*Delta)/(Delta + 2.25))*(AR/L^2)*(uR2^2 + vR^2)*sin(alfaR2);
T1 = 2*rho*D^4/(U^2*L^2*rho)*Kt1*n1*abs(n1);
T2 = 2*rho*D^4/(U^2*L^2*rho)*Kt2*n2*abs(n2);
W = rho*g*nabla/(rho*L^2*U^2/2);

DX = Xu + (1-t)*T1 + (1-t)*T2 + Xvr*v*r + Xvv*v^2 + Xrr*r^2 + Xff*fi^2 +
cRX*Fn1*sin(delta) + cRX*Fn2*sin(delta) + (m+my)*v*r + Xw;

```

```

DY = Yv*v + Yr*r + Yp*p + Yf*fi + Yvvv*v^3 + Yrrr*r^3 + Yvvr*v^2*r +
Yvrr*v*r^2 + Yvvf*v^2*fi + Yvff*v*fi^2 + Yrrf*r^2*fi + Yrff*r*fi^2 + (1
+aH)*Fn1*cos(delta) + (1 +aH)*Fn2*cos(delta) - (m+mx)*vx*r + Yw;
DK = Kv*v + Kr*r + Kp*p + Kf*fi + Kvvv*v^3 + Krrr*r^3 + Kvvr*v^2*r +
Kvrr*v*r^2 + Kvvf*v^2*fi + Kvff*v*fi^2 + Krrf*r^2*fi + Krff*r*fi^2 - (1
+aH)*zR*Fn1*cos(delta) - (1 +aH)*zR*Fn2*cos(delta) + mx*lx*vx*r - W*GM*fi;
DN = Nv*v + Nr*r + Np*p + Nf*fi + Nvvv*v^3 + Nrrr*r^3 + Nvvr*v^2*r +
Nvrr*v*r^2 + Nvvf*v^2*fi + Nvff*v*fi^2 + Nrrf*r^2*fi + Nrff*r*fi^2 + (xR
+aH*xH)*Fn1*cos(delta) + (xR +aH*xH)*Fn2*cos(delta) + (T2-T1)*d - DY*xG +
Nw;
Acel = M\[DX;DY;DK;DN];
sys =
[Acel(1)*(U^2)/L;Acel(2)*(U^2)/L;Acel(3)*(U/L)^2;Acel(4)*(U/L)^2;Q1;Q2];

% end mdlOutputs

%
%=====
===
% mdlTerminate
% Perform any end of simulation tasks.
%=====
===
%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

```

Anexo B - calc_erro.m

```
function erro = calc_erro(x,yref)

ytemp = step(yref(4000)*[x(1) 1],[x(2)*x(3) x(2)+x(3) 1],0:.05:200);

erro = sum ((ytemp-yref).^2);
end
```

otimiza.m:

```
options = optimset('TolX',1e-8,'TolFun',1e-10);

[x, fval]=fminunc(@(x)calc_erro(x,yref),[10 6 6],options)

step(yref(4000)*[x(1) 1],[x(2)*x(3) x(2)+x(3) 1],0:.05:200);
hold all
p=plot(t.signals.values,r.signals.values);
set(p,'Color','red','LineWidth',1);
```

Anexo C - convunit.m

```

% CONVUNIT  Constantes multiplicadoras para conversão de unidades

fprintf('Conversão de unidades ... ');

bar2Pa      = 100000;          % [bar] -> [Pa]
ft2m        = 0.3048;          % [ft] -> [m]
ftps2mps    = 0.3048;          % [ft/s] -> [m/s]
hp2W        = 745.670;         % [hp] -> [W]
kgf2N        = 9.80665;         % [Kgf] -> [N]
knots2mps    = 0.514435;        % [knots] -> [m/s]
lb2kg        = 0.4535924;        % [lb] -> [kg]
lbf2N        = 4.44830;         % [lbf] -> [N] (g=9.80665 m/s^2)
lbfft2Nm     = 1.35584;         % [lbf.ft] -> [N.m] (g=9.80665 m/s^2)
                                % lbfft2kgfm      = 0.138257; %
[lbf.ft] -> [kgf.m] (g=9.80665 m/s^2)
                                % lbft22kgm2      = 0.0421408; %
[lb.ft^2] -> [kg.m^2]
                                % lbffts22kgfms2= 0.138257; %
[lbf.ft.s^2] -> [kgf.m.s^2]
lbffts22Nms2 = 1.35584;         % [lbf.ft.s^2] -> [N.m.s^2]
lbph2kgps    = 0.4535924/3600; % [lb/h] -> [kg/s]
lbs2kgps     = 0.4535924;        % [lb/s] -> [kg/s]
mmHg2Pa      = 133.3;           % [mmHg] -> [Pa]
rpm2radps    = 0.104720;        % [rpm] -> [rad/s]
ton2kg        = 1016.0469;       % [ton (UK)] -> [kg] (toneladas
inglesas)
                                % ton2kg = 907.2; % [ton] -> [kg]

deg2rad       = pi/180;          % [deg] -> [rad]
psia2bar      = 0.06894757;      % [psia] -> [bar]
psi2bar       = psia2bar;        % [psi] -> [bar]
R2K           = 0.5555556;       % [R] -> [K] (Rankine para
Kelvin)
W2kW          = 1/1000;          % [W] -> [kW]

fprintf('OK\n');

```

Anexo D - ctr_proa

```

function [sys,x0,str,ts,simStateCompliance] = ctr_proa(t,x,u,flag)
switch flag,

    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Derivatives %
    %%%%%%%%%%%%%%%
    case 1,
        sys=mdlDerivatives(t,x,u);

    case 2,
        % sys=mdlUpdate(t,x,u);

    %%%%%%%%%%%%%%%
    % Update %
    %%%%%%%%%%%%%%%

    case 3,
        sys=mdlOutputs(t,x,u);

    case 4,
        %sys=mdlGetTimeOfNextVarHit(t,x,u);

    %%%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%%
    case 9,
        sys=mdlTerminate(t,x,u);

    %%%%%%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%%%%%
    otherwise
        DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl

%
%=====
%
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-
% function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

```

```

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [-1 0];

% Specify the block simStateCompliance. The allowed values are:
% 'UnknownSimState', < The default setting; warn and assume
DefaultSimState
% 'DefaultSimState', < Same sim state as a built-in block
% 'HasNoSimState', < No sim state
% 'DisallowSimState' < Error out when saving or restoring the model sim
state
simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=====
===
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
===
%
function sys=mdlDerivatives(t,x,u)
sys = [];

% end mdlDerivatives

```

```

%
%=====
===
% mdlOutputs
% Return the block outputs.
%=====
===
%
function sys=mdlOutputs(t,x,u)
psi=u;
proa=psi;
if psi<0
    n=1;
    while proa<0
        proa=psi+(2*n)*pi;
        n=n+1;
    end
end
if psi>=2*pi
    n=1;
    while proa>=2*pi
        proa=psi-(2*n)*pi;
        n=n+1;
    end
end
sys = proa;

% end mdlOutputs


%
%=====
===
% mdlTerminate
% Perform any end of simulation tasks.
%=====
===

%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

```

Anexo E - ctr_data

```

fprintf('Controle ... ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Controle MCP                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Rampa adaptativa
% seleção
MCPrate = 0.9;           % seleção de taxas de rampa
% subida
MCPchgrateup = 2*rpm2radps;      % taxa de variação da rampa de rotação
                                   % para subida (default 4 rpm/s^2)

MCPrateupfast = 30*rpm2radps;    % taxa máxima da rampa de rotação
                                   % para subida (default 60 rpm/s)

MCPrateupslow = 2*rpm2radps;     % taxa mínima da rampa de rotação
                                   % para subida (default 4 rpm/s)
% descida
MCPchgratedown = 50*rpm2radps;   % taxa de variação da rampa de rotação
                                   % para descida (default 100 rpm/s^2)

MCPratedownfast = 200*rpm2radps; % taxa máxima da rampa de rotação
                                   % para descida (default 400 rpm/s)

MCPratedownslow = 50*rpm2radps;  % taxa mínima da rampa de rotação
                                   % para descida (default 100 rpm/s)

% Suavização
MCPepr = 10*rpm2radps; % suavização da rampa de rotação (default 20 rpm)

% Limites
MCPidle = 350*rpm2radps; % marcha lenta (default 350 rpm)
MCPmin = MCPidle;        % vel. mínima admissível (default "idle")
MCPmax = 1230*rpm2radps; % vel. máxima admissível (default 1230 rpm)

% Regulador MCP
% Parâmetros internos do regulador MCP
maxFR = 80; % cremalheira máxima admissível (default 80 mm)

GainBkpt = 0.2; % Inflexão para redução do ganho (default 20% crem)
GainMin = 0.2; % Ganho mínimo para carga elevada (default 20% crem)

zm = 1*rpm2radps; % janela de desligamento da atuação do regulador
                  % (default 1 rpm-motor)

Kg = 2/(100*rpm2radps); % Taxa de aumento de ganho para erro elevado
                        % (default 2/(100 rpm-motor))
zme = 100*rpm2radps; % Inflexão para aumento de ganho (default 100 rpm)

% Parâmetros de sintonia do regulador MCP
Kp_MCP = 1;
Ti_MCP = 1;
Td_MCP = 0;

```

```

N_MCP = 10;

MCPready = 0.85;    % MCP a X% da rotação do eixo (pronto para encher o
                    % acoplamento fluido) (default 85% rot-eixo)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Controle HPC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Taxas de rampeamento

HPCerate = 3;    % parcela básica da taxa de variação do passo
                % referente à bomba elétrica de atuação
                % (default 3 graus/s, ligeiramente inferior ao max.atuador)

HPChrate = 0.01; % parcela adicional de variação do passo com rot. eixo
                % referente à bomba hidráulica de atuação
                % (default 0.01 graus/rpm(eixo), lig. inf. ao max.atuador)

HPCrotmin = 85*rpm2radps;    % rotação de ativação da bomba hidráulica
                             % (default 85 rpm)

% Suavização
HPCeps = 1;    % suavização da rampa de passo (default 1 grau)

% Limites
HPCmin = -40;    % passo mínimo admissível (default -40 graus)
HPCmax = 40;    % passo máximo admissível (default 40 graus)

% Ajuste fino de passo
HPCajuste = 1;    % faixa de ativação do ajuste fino (default 1 grau)
Kp_HPC = 0.1;    % ganho proporcional do controlador (default 0.1)
Ti_HPC = 20;    % tempo de repetição do controlador (default 20)
aut_HPC = [-1 1];    % limite de autoridade (faixa de saída do
                    % controlador) (default [-1 1] graus)

% Geração de sinais lógicos
HPCfinal = 2;    % tolerância para estado estacionário (default 2 graus)
HPCant = 5;    % passo para disparar antecipação (default 5 graus)

fprintf('OK\n');

```

Anexo F - desenho_NPa

```

function [sys,x0,str,ts]=desenho_NPa(t,x,u,flag)

switch flag
    %%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%
    case 0
        [sys,x0,str,ts] = mdlInitializeSizes();

    %%%%%%%%%%%
    % Update %
    %%%%%%%%%%%
    case 2,
        sys=mdlUpdate(t,x,u);

    %%%%%%%%%%%
    % Unused flags %
    %%%%%%%%%%%
    case { 1, 3, 4, 9 },
        sys = [];

    otherwise
        error(['unhandled flag = ',num2str(flag)]);
end

% end limintm

%=====
%
% mdlInitializeSizes
%=====
%
function [sys,x0,str,ts] = mdlInitializeSizes()
global l1 l2 xmax xmin ymax ymin governo zig_zag par_emergencia
centralizar...
    ax1 ax2 tU tleme tt tpsi tMCP_BE tdados_MCP_BE tMCP_BB tdados_MCP_BB
...
    tHPC_BE tHPC_BB h h_bot x_d y_d psi...
    psi_ref tgoverno tR;

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 0;
sizes.NumInputs = 17;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0 = [];
ts = [-1 0]; % sample time: [period, offset]
close all;
f=figure;

```

```

xmini = -500;
xmaxi = 500;
ymini = -500;
ymaxi = 500;

xmin = xmini;
xmax = xmaxi;
ymin = ymini;
ymax = ymaxi;

ax1 = axes;
set(ax1, 'Position', [0.05 0.05 0.7 0.85]);

axis([xmini xmaxi ymini ymaxi]);

%Desenho do navio=====
l1=line(0,0);
set(l1, 'Color', 'black', 'linewidth', [1.5], 'EraseMode', 'xor');

l2=line(0,-58.3);
set(l2, 'Color', 'white', 'linewidth', [1.5], 'EraseMode', 'xor');

sy=1;

x_d = 0;
y_d = 0;
psi= 0;

Loa=58.3;
B = 8.6;
xG=25.14;

popa = [y_d-xG*sin(psi) ; sy*(x_d-xG*cos(psi)) ];
aBE = popa + B/2 * [cos(psi);sy*(-sin(psi))] ;
aBB = popa - B/2 * [cos(psi);sy*(-sin(psi))] ;
P = [y_d+(.8*Loa-xG)*sin(psi) ;sy*(x_d+(.8*Loa-xG)*cos(psi)) ];
bBE = P + B/2 * [cos(psi);sy*(-sin(psi))] ;
bBB = P - B/2 * [cos(psi);sy*(-sin(psi))] ;
proa = [y_d+(Loa-xG)*sin(psi) ; sy*(x_d+(Loa-xG)*cos(psi)) ];
set(l1, 'XData', [aBB(1) aBE(1) bBE(1) proa(1) bBB(1) aBB(1)]);
set(l1, 'YData', [aBB(2) aBE(2) bBE(2) proa(2) bBB(2) aBB(2)]);
%=====

set_param('sim_NPa/PCL_BE', 'Value', num2str(42));
set_param('sim_NPa/PCL_BB', 'Value', num2str(42));
set_param('sim_NPa/leme', 'Value', num2str(0));
set_param('sim_NPa/rumo', 'Value', num2str(0));
set_param('sim_NPa/governo', 'Value', num2str(0));
set_param('sim_NPa/Int.', 'Value', num2str(0));
set_param('sim_NPa/Dir.', 'Value', num2str(0));

governo=0;
zig_zag = 0;
par_emergencia=0;
centralizar = 0;
psi_ref=0;

```

```

ax2=axes;
set(ax2, 'Position', [0.75 0.5 0.25 0.85], 'Visible', 'off')
axis([0 1 0 1]);

tU = text(0.1,0.55,['Velocidade = ',num2str(0),'
nós'], 'FontSize',14, 'EraseMode', 'xor');
tleme = text(0.6,0.45,['leme = ',num2str(0),'
°'], 'FontSize',12, 'EraseMode', 'xor');
tt = text(0.1,0.45,['t = ',num2str(0),'
s'], 'FontSize',12, 'EraseMode', 'xor');
tpsi = text(0.1,0.5,['rumo = ',num2str(0),'
°'], 'FontSize',14, 'EraseMode', 'xor');
tMCP_BE = text(0.1,0.4,['MCP_B_E: ', ' Potência = ',num2str(0),'
KW'], 'FontSize',12, 'EraseMode', 'xor');
tdados_MCP_BE = text(0.1,0.35,['Rot = ',num2str(0),' rpm;', ' Torque =
',num2str(0),' N.m'], 'FontSize',12, 'EraseMode', 'xor');
tMCP_BB = text(0.1,0.30,['MCP_B_B: ', ' Potência = ',num2str(0),'
KW'], 'FontSize',12, 'EraseMode', 'xor');
tdados_MCP_BB = text(0.1,0.25,['Rot = ',num2str(0),' rpm;', ' Torque =
',num2str(0),' N.m;'], 'FontSize',12, 'EraseMode', 'xor');
tHPC_BE = text(0.1,0.20,['HPC_B_E: ', 'Rot = ',num2str(0),' rpm;', ' passo =
',num2str(0),' °'], 'FontSize',12, 'EraseMode', 'xor');
tHPC_BB = text(0.1,0.15,['HPC_B_B: ', 'Rot = ',num2str(0),' rpm;', ' passo =
',num2str(0),' °'], 'FontSize',12, 'EraseMode', 'xor');
tR = text(0.1,0.05,['', 'FontSize',12, 'EraseMode', 'xor');

% Botões dos gráficos

% Ângulo do leme
h(1) = uicontrol('Style', 'edit', 'units','normalized','String', '0',...
'Position', [.8 .3 .1 .05], 'backgroundcolor', [1 1 1]);
h_bot(1) = uicontrol('Style','pushbutton','units','normalized','String',
'leme_enter',...
'Position', [.8 .25 .1 .05], 'callback','leme_enter');

% Rumo
h(2) = uicontrol('Style', 'edit', 'units','normalized','String', '0',...
'Position', [.9 .3 .1 .05], 'backgroundcolor', [1 1 1]);
h_bot(2) = uicontrol('Style','pushbutton','units','normalized','String',
'rumo_enter',...
'Position', [.9 .25 .1 .05], 'callback','rumo_enter');

% PCL
h(3) = uicontrol('Style', 'edit', 'units','normalized','String', '42',...
'Position', [.9 .45 .1 .05], 'backgroundcolor', [0 1 0]);
h(4) = uicontrol('Style', 'edit', 'units','normalized','String', '42',...
'Position', [.8 .45 .1 .05], 'backgroundcolor', [1 0 0]);
h_bot(3) = uicontrol('Style','pushbutton','units','normalized','String',
'PCL_enter',...
'Position', [.8 .4 .2 .05], 'callback','PCL_enter');

% Vento
h(5) = uicontrol('Style', 'edit', 'units','normalized','String', '0',...
'Position', [.8 .15 .1 .05], 'backgroundcolor', [1 1 1]);
h(6) = uicontrol('Style', 'edit', 'units','normalized','String', '0',...
'Position', [.9 .15 .1 .05], 'backgroundcolor', [1 1 1]);

```

```
h_bot(8) = uicontrol('Style','pushbutton','units','normalized','String',
'Vento (dir - int)',...
'Position', [.8 .1 .2 .05],'callback','Vento_enter');
```

```
% Botões de Manobras
```

```
h_bot(4) = uicontrol('Style','pushbutton','units','normalized','String',
'Zig-zag',...
'Position', [.8 .05 .1 .05],'callback','manobra_zigzag');
h_bot(5) = uicontrol('Style','pushbutton','units','normalized','String',
'Parada em emergência',...
'Position', [.9 .05 .1 .05],'callback','parada_emergencia');
h_bot(6) = uicontrol('Style','pushbutton','units','normalized','String',
'Governo',...
'Position', [.8 0 .1 .05],'callback','manobra_governo');
tgvoverno = text(.1,.1,'Controle manual','FontSize',14,'EraseMode','xor');
h_bot(7) = uicontrol('Style','pushbutton','units','normalized','String',
'centralizar XY',...
'Position', [.9 0 .1 .05],'callback','manobra_centralizar');
```

```
% end mdlInitializeSizes
```

```
%=====
===
```

```
% mdlUpdate
```

```
%=====
===
```

```
function sys=mdlUpdate(t,x,u)
global zig_zag par_emergencia governo h centralizar ax1 tR tU tleme tt
tpsi...
tmCP_BE tdados_MCP_BE tMCP_BB tdados_MCP_BB x_d y_d psi...
tHPC_BE tHPC_BB psi_ref tgvoverno l1 l2 xmax xmin ymax ymin;
```

```
% u(1)          % u
% u(2)          % v
U=sqrt(u(1)^2+u(2)^2);
% u(3)          % leme
% u(4)          % tempo
% u(5)          % posição em x
% u(6)          % posição em y
% u(7)          % rumo
% u(8)          % r
% u(9)          % rotação MCP BE
% u(10)         % rotação MCP BB
% u(11)         % rotação HPC BE
% u(12)         % rotação HPC BB
% u(13)         % torque HPC BE
% u(14)         % torque HPC BB
% u(15)         % passo BE
% u(16)         % passo BB
% u(17)         % psi
```

```
% Manobra de zig-zag
```

```
%*****
```

```
if (zig_zag == 1)
n=round(u(17)/(2*pi));
```

```

    aux=psi_ref+2*n*pi;
    ang_max=aux+pi/18;
    ang_min=aux-pi/18;
    set(tgoverno,'String', 'Zig-zag');
    if (u(17) < ang_min)
        set_param('sim_NPa/leme','Value',num2str(20));
    end;
    if (u(17) > ang_max)
        set_param('sim_NPa/leme','Value',num2str(-20));
    end;
end;

% Parada em emergência
%*****
if (par_emergencia==1)
    if (sign(u(1))*U>3)
        set_param('sim_NPa/PCL_BE','Value',num2str(-100));
        set_param('sim_NPa/PCL_BB','Value',num2str(-100));
    else
        set_param('sim_NPa/PCL_BE','Value',num2str(0));
        set_param('sim_NPa/PCL_BB','Value',num2str(0));
        par_emergencia=0;
    end
end

% Botão governo
%*****
if (governo==1)
    zig_zag=0;
    psi_ref=str2num(get(h(2),'string'))*pi/180;
    set(tgoverno,'String', 'Piloto automático');
    set(tR,'String','');
else
    if (zig_zag == 1)
        set(tR,'String','');
    else
        set(tgoverno,'String', 'Controle manual');
        leme=u(3);
        if (leme~=0)
            Umps=U*1852/3600;
            R=abs(Umps/u(8));
            if (R>500)
                set(tR,'String','');
            else
                set(tR,'String', ['Raio de giro=
',num2str(round(10*R)/10),' m']);
            end
        else
            set(tR,'String','');
        end
    end
end

% Manobra de centralizar
%*****

if (centralizar ~= 0)
    limx = get(ax1,'Xlim');
    limy = get(ax1,'Ylim');

```

```

dx = limx(2)-limx(1);
dy = limy(2)-limy(1);

xmin = u(6)-dx/2;
xmax = u(6)+dx/2;

ymin = u(5)-dy/2;
ymax = u(5)+dy/2;

set(ax1,'Xlim',[xmin xmax]);
set(ax1,'Ylim',[ymin ymax]);

centralizar = 0;
end

sy=1;

x_d = u(5);
y_d = u(6);
psi= u(17);

Loa=58.3;
B = 8.6;
xG=25.14;

%Desenho do navio
popa = [y_d-xG*sin(psi) ; sy*(x_d-xG*cos(psi)) ];
aBE = popa + B/2 * [cos(psi);sy*(-sin(psi))] ;
aBB = popa - B/2 * [cos(psi);sy*(-sin(psi))] ;
P = [y_d+(.8*Loa-xG)*sin(psi) ;sy*(x_d+(.8*Loa-xG)*cos(psi)) ];
bBE = P + B/2 * [cos(psi);sy*(-sin(psi))] ;
bBB = P - B/2 * [cos(psi);sy*(-sin(psi))] ;
proa = [y_d+(Loa-xG)*sin(psi) ; sy*(x_d+(Loa-xG)*cos(psi)) ];
set(l1,'XData',[aBB(1) aBE(1) bBE(1) proa(1) bBB(1) aBB(1)]);
set(l1,'YData',[aBB(2) aBE(2) bBE(2) proa(2) bBB(2) aBB(2)]);

popa2 = [y_d-(xG+Loa)*sin(psi) ; sy*(y_d-(xG+Loa)*cos(psi)) ];
aBE2 = popa2 + B/2 * [cos(psi);sy*(-sin(psi))] ;
aBB2 = popa2 - B/2 * [cos(psi);sy*(-sin(psi))] ;
P2 = [y_d-xG*sin(psi) ;sy*(x_d-xG*cos(psi)) ];
bBE2 = P2 + B/2 * [cos(psi);sy*(-sin(psi))] ;
bBB2 = P2 - B/2 * [cos(psi);sy*(-sin(psi))] ;
set(l2,'XData',[aBB2(1) aBE2(1) bBE2(1) bBB2(1) aBB2(1)]);
set(l2,'YData',[aBB2(2) aBE2(2) bBE2(2) bBB2(2) aBB2(2)]);

if y_d>xmax
    limx = get(ax1,'Xlim');
    dx = limx(2)-limx(1);
    xmin = u(6);
    xmax = u(6)+dx;
    set(ax1,'Xlim',[xmin xmax]);
end;
if y_d<xmin
    limx = get(ax1,'Xlim');
    dx = limx(2)-limx(1);
    xmin = u(6)-dx;
    xmax = u(6);
    set(ax1,'Xlim',[xmin xmax]);

```

```

end;
if x_d>ymax
    limy = get(ax1,'Ylim');
    dy = limy(2)-limy(1);
    ymin = u(5);
    ymax = u(5)+dy;
    set(ax1,'Ylim',[ymin ymax]);
end;
if x_d<ymin
    limy = get(ax1,'Ylim');
    dy = limy(2)-limy(1);
    ymin = u(5)-dy;
    ymax = u(5);
    set(ax1,'Ylim',[ymin ymax]);
end;

set(tU,'String', ['Velocidade = ',num2str(round(10*U)/10),' nós']);

delta=u(3);
if (delta<-.05)
    set(tleme,'String', ['leme = ',num2str(round(10*abs(delta))/10),' °
BB']);
elseif (delta>.05)
    set(tleme,'String', ['leme = ',num2str(round(10*abs(delta))/10),' °
BE']);
else
    set(tleme,'String', ['leme = ',num2str(0),' °']);
end

set(tt,'String', ['t = ',num2str(round(u(4))),' s']);
set(tpsi,'String', ['rumo = ',num2str(round(10*u(7))/10),' °']);
set(tMCP_BE,'String', ['MCP_B_E:','Potência = ',num2str(round(u(11)*u(13)*2*pi/6000)/10),' KW']);
set(tdados_MCP_BE,'String', ['Rot = ',num2str(round(u(9))),' rpm;', 'Torque = ',num2str(round(u(13)/1.48),' N.m;']);
set(tMCP_BB,'String', ['MCP_B_B:','Potência = ',num2str(round(u(12)*u(14)*2*pi/6000)/10),' KW']);
set(tdados_MCP_BB,'String', ['Rot = ',num2str(round(u(10))),' rpm;', 'Torque = ',num2str(round(u(14)/1.48),' N.m;']);
set(tHPC_BE,'String', ['HPC_B_E:','Rot = ',num2str(round(u(11))),' rpm;', 'passo = ',num2str(round(10*u(15))/10),' °']);
set(tHPC_BB,'String', ['HPC_B_B:','Rot = ',num2str(round(u(12))),' rpm;', 'passo = ',num2str(round(10*u(16))/10),' °']);

sys = [];

% end mdlUpdate

```

Anexo G - histerese_embreagens.m

```
% velocidades entre os runners e pinhão  
  
VSUP=0.5; %1;  
VINP=-0.5; %-1;  
  
% dados da embreagem  
  
% SSS 124t  
  
logica_sss_124t= [ 0; 1; 0; 0; 0; 1; 0; 0];
```

Anexo H - hpc.m

```

% HPC
% Parâmetros do atuador de passo
fprintf('Atuador de passo ... ');

% acomodação da rampa
hpc_tol = 0.5*deg2rad;      % [rad]

% atraso do sistema hidráulico do HPC
hpc_delay = 2;              % [s]

% rotação mínima para ativação das bombas
hpc_rotmin = 85*rpm2radps; % [rad/s]

% taxas de variação do passo
% bomba elétrica
hpc_slew_e = 3.28*deg2rad; % [rad/s]
% bomba hidráulica
hpc_slew_h = (3.28*deg2rad)/(269*rpm2radps); % [-]

% relação entre ângulo e P/D
hpc_deg=[-40 -10.8 0 10.8 17.7 19.3 20.9 22.5 40]*deg2rad; % [rad]
hpc_pd=[-4.5 -.6 0 .6 1 1.1 1.2 1.3 4.5]; % [-]

fprintf('OK\n');

```

Anexo I - Integracao_casco_helice.m

```

function [V, PD, KTtab, KQtab, efftab, Jtab, RPMtab, BHPTab] =
Integracao_casco_helice
fprintf('Integração casco-hélice ...');
[KTAR, KQAR, VAR, effAR, JAR, RPMAR, BHPAR]=Integracao_casco_heliceAR;
[KTAV, KQAV, VAV, effAV, JAV, RPMAV, BHPAV]=Integracao_casco_heliceAV;
V=[-VAR VAV];
Z = zeros(1, length(V));
KT=[KTAR KTAV]; KTtab=[Z; KT];
KQ=[KQAR KQAV]; KQtab=[Z; KQ];
RPM=[RPMAR RPMAV]; RPMtab=[Z; RPM];
BHP=[BHPAR BHPAV]; BHPTab=[Z; BHP];
eff=[effAR effAV]; efftab=[Z; eff];
J=[JAR JAV]; Jtab=[Z; J];
PD=[0; 0.6; 0.7; 0.8; 0.9; 1; 1.1; 1.2; 1.3; 1.4; 1.5; 1.6; 1.7; 1.8; 1.9;
2];
fprintf('OK\n');
end

```

Anexo J - Integracao_casco_heliceAR.m

```

function [KTcav, KQcav, Vnos, effcav, Jcav, RPMcav, BHPcav] =
Integracao_casco_heliceAR
%-----DADOS DO NAVIO-----
Desl=488; % [ton]
L=56.5; % [m] comprimento da linha de água
calado=2.28; % [m] calado do navio
%-----DADOS DO HÉLICE-----
D=[1.7];
DAR=0.8;
%-----DADOS PARA O CÁLCULO DA CAVITAÇÃO-----
temperatura=15; % em Celsius
pv=pressao(temperatura); % sub rotina para calcular a pressão de vapor
alteixo=0.35; % altura do eixo apartir da linha de base
gravidade=9.81; % [m/s2]
rho=1025; % [Kg/m3]
atm=101300; % pressão atmosférica [N/m2]
po=atm+rho*gravidade*(calado-alteixo+0.0075*L); % pressão na altura do
hélice
%----- VALORES DE RESISTÊNCIA POR VELOCIDADE-----
%A cada velocidade o navio apresenta uma resistência ao avanço diferente.
%Vetor resistência----Res=[Rv1 Rv2 Rv3 Rv4 Rv5 Rv6 ... Rvn]
%Vetor velocidade----Vnos=[V1 V2 V3 ... Vn]
%-----
Res=[272840 161440 79107 25831]; % [N]
Vnos=[13 10 7 4]; % [nós]
V=Vnos*0.51444; % converte a velocidade em m/s
Pcasco=V.*Res;
%-----COEFICIENTES-----
for a=1:length(Vnos)
    Fnvola=0.165*Vnos(a)/((Desl)^(1/6));
    w(a)=0.00343+0.11152*Fnvola-0.27571*(Fnvola^2)+0.16330*(Fnvola^3)...
        -0.02828*(Fnvola^4);
    t(a)=0.43803-0.92242*Fnvola+0.81950*(Fnvola^2)-0.32145*(Fnvola^3)...
        +0.04659*(Fnvola^4);
    effrr(a)=0.81524+0.43985*Fnvola-0.47333*(Fnvola^2)+0.19918*(Fnvola^3)...
        -0.02351*(Fnvola^4)-0.00201*(Fnvola^5);
    Fnvola=0;
end
effs=0.98; % eficiência do eixo
efft=0.97; % eficiência de transmissão
%-----DADOS PARA A CURVA Kt(casco)-----
R=Res./2;
MR=0.25;
J=0:0.01:2;
KTcasco=zeros(length(Vnos), length(J)); % Gera a matriz de zeros

for b=1:length(Vnos)
    num=((1+MR)*R(b)*(J.^2));
    dem=((1-t(b)).*(V(b)*(1-w(b)).*D).^2)*1025;
    KTcasco(b,:)=num./dem; % Preenche linha a linha de KTcasco
end
%-----GERA O SPLINE DE KTcasco-----
s=0:0.0001:J(end); % gera vetor de discretização de J
KT_cascoS=zeros(length(Vnos), length(s)); % Gera a matriz de zeros

for d=1:length(Vnos)
    KT_cascoS(d,:)=spline(J,KTcasco(d,:),s); % cria uma spline KTcasco_S

```

end

%-----

%-----CONSTRUÇÃO DA SÉRIE GAWN-----

PD=[0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2];

%PD=0.6:0.05:2;

```
A=[0.1193852    0    0    0    0;
-0.6574682    0    0    0    1;
0.3493294    0    0    1    0;
0.4119366    0    0    1    1;
-0.1991927    0    0    2    1;
5.8630510    -2    0    2    2;
-1.1077350    -2    0    2    3;
-0.1341679    0    1    0    0;
0.2628839    0    1    0    1;
-0.5217023    0    1    1    1;
0.2970728    0    1    2    0;
6.1525800    -2    2    1    3;
-2.4708400    -2    2    2    3;
-4.0801660    -3    1    6    0;
4.1542010    -3    1    6    1;
-1.1364520    -3    2    6    0];
```

```
B=[1.5411660    -3    0    0    0;
0.1091688    0    0    0    1;
-0.3102420    0    0    0    2;
0.1547428    0    0    0    3;
-4.3706150    -2    0    1    0;
0.2490295    0    0    1    2;
-0.1594602    0    0    1    3;
8.5367470    -2    0    2    0;
-9.5121630    -2    0    2    1;
-9.3203070    -3    0    2    2;
3.2878050    -2    0    2    3;
5.4960340    -2    1    0    1;
-4.8650630    -2    1    1    0;
-0.1062500    0    1    1    1;
8.5299550    -2    1    2    0;
1.1010230    -2    2    0    3;
-3.1517560    -3    2    2    2];
```

for l=1:length(PD)

%

%Gerando Kt

kt=A(:,1).*(10.^A(:,2)).*(DAR.^A(:,3)).*(PD(l).^A(:,4));

for k=1:length(J)

auxJ=J(k).^A(:,5);

auxKt=kt.*auxJ;

Kt(:,k)=auxKt;

end

KT=sum(Kt);

%

%Gerando Kq

kq=B(:,1).*(10.^B(:,2)).*(DAR.^B(:,3)).*(PD(l).^B(:,4));

for k=1:length(J)

auxJ=J(k).^B(:,5);

```

        auxKq=kq.*auxJ;
        Kq(:,k)=auxKq;
    end
    KQ=sum(Kq);
%-----FILTRO DE VALORES DE Eff, KT E KQ-----
    eff=(KT./KQ).*(J./(2*pi));

    for z=2:length(J)
        if KT(z-1)<=0
            KT(z-1)=0;
            KT(z)=0;
        else
            KT(z)=KT(z);
        end
        if KQ(z-1)<=0
            KQ(z-1)=0;
            KQ(z)=0;
        else
            KQ(z)=KQ(z);
        end
    end

    for z=3:length(J)
        if eff(z-1)<=0
            eff(z)=0;
        else
            eff(z)=eff(z);
        end
        if eff(z)<0
            eff(z)=0;
        end
    end

    K_T(1,:)=KT;
    K_Q(1,:)=KQ;
    Eff(1,:)=eff;
%-----CONSTRUÇÃO DE SPLINES PARA A INTERSEÇÃO-----

    KT_S=spline(J,KT,s); % cria uma spline K_T
    KQ_S=spline(J,KQ,s); % cria uma spline K_Q

    for v=1:length(Vnos)
        y=find(KT_S-KT_cascoS(v,:)>=0); % gera vetor com índices das dif>=0

        c=y(end); % último elemento do vetor y

        Jint(1,v)=s(c); % preenche a matriz Jint

        KTint(1,v)=KT_S(c); % preenche a matriz KTint

        KTint_effrr(1,v)=KT_S(c)*effrr(v); % cria KTint*effrr seria o Kt
                                           % existente levando em
                                           % consideração que não está
                                           % em água aberta e sim na
                                           % popa do navio
        KQint(1,v)=KQ_S(c); % preenche a matriz KQint

        KQint_effrr(1,v)=KQ_S(c)/effrr(v); % cria KQint/effrr para o BHP

```

```

    rot(l,v)=V(v)*(1-w(v))/(s(c)*D); % rotações em rps

    tauc(l,v)=(KTint_effrr(l,v)*(rot(l,v)*D)^2)/(0.125*pi*(1.067-...
        0.229*PD(l))*((V(v)*(1-w(v)))^2+(0.7*pi*D*rot(l,v))^...
        2)); % cálculo da matriz de tau
    % o cálculo de tau está levando em consideração que o hélice está
    % posicionado na popa do navio

    sigma(l,v)=(po-pv)/(0.5*rho*((V(v)*(1-w(v)))^2+(0.7*pi*D*...
        rot(l,v))^2)); % cálculo da matriz de sigma
end

end
RPM=60*rot;
BHP=((2*pi*1025*D^5)*KQint_effrr.*rot.^3)./(efft))/10^3; % cria a matriz
% BHP em KW

effint=(KTint./KQint).*(Jint./(2*pi));

% %-----gráficos-----
--
% figure(1)
% hold on
% grid on
% title(['Integração casco-hélice'],'FontSize',14)
% ylabel(['Potência(BHP) em KW'],'FontSize',12);
% xlabel(['Rotação em RPM'],'FontSize',12);
% plot(RPM,BHP)
for v=1:length(V)
%
text(RPM(length(l),v),BHP(length(l),v),['\leftarrow',num2str(Vnos(v))...
% ],'HorizontalAlignment',...
% 'left','FontSize',8)
end
for l=1:length(PD)
% plot(RPM(l,:),BHP(l,:))
% text(RPM(l,length(V)),BHP(l,length(V)),['',num2str(PD(l))...
% ],'HorizontalAlignment',...
% 'right','FontSize',10)
end

% %-----
--
% %Os índices tau e sigma para a cavitação estão sendo calculados nas
linhas
% %166 e 169. os dados necessarios estao sendo inseridos no início da
rotina.
% %-----
--
%
% figure(2)
% grid on
% hold on
%
% title(['Cavitação'],'FontSize',14)
% ylabel(['\tau'],'FontSize',14);
% xlabel(['\sigma'],'FontSize',14);
%
% %-----
--
% %As regressões y10 e y25 foram obtidas apartir de pontos plotados do

```

```

% %gráfico de burril apresentado no marine propeller e calculado com o
apoio
% %do EXCEL. Os índices R^2 das regressões são próximos de um, logo
apresentam
% %proximidade do real, os índices são respectivamente 0.9677 e 0.9765.
% %-----
--
%

SIG=min(min(sigma)):0.001:max(max(sigma));          % obtém o max e o min sigma
y10=0.3777*SIG.^0.7294;
y25=0.2421*SIG.^0.6526;
% plot(SIG,y10,SIG,y25)
%
for l=1:length(PD)
    for v=1:length(V)
        % hold on
        if (0.2421*sigma(l,v)^0.6526)>=tauc(l,v)
            % plot(sigma(l,v),tauc(l,v),...
            % 'bo','LineWidth',1,...
            % 'MarkerEdgeColor','k',...
            % 'MarkerFaceColor','g',...
            % 'MarkerSize',3)
            % text(sigma(l,v),tauc(l,v), ['PD,V=',num2str(PD(l)),' ',...
            % num2str(Vnos(v)),'\rightarrow'],'HorizontalAlignment',...
            % 'right','FontSize',9)
            cav(l,v)=1;
        elseif (0.3777*sigma(l,v)^0.7294)>=tauc(l,v)
            % plot(sigma(l,v),tauc(l,v),...
            % 'bo','LineWidth',1,...
            % 'MarkerEdgeColor','k',...
            % 'MarkerFaceColor','b',...
            % 'MarkerSize',3)
            cav(l,v)=1;
        else
            % plot(sigma(l,v),tauc(l,v),...
            % 'bo','LineWidth',1,...
            % 'MarkerEdgeColor','k',...
            % 'MarkerFaceColor','w',...
            % 'MarkerSize',3)
            cav(l,v)=0;
        end
    end
end

% text(SIG(round(length(SIG)/2-5)), (0.3777*SIG(round(length(SIG)/2-
5))^0.7294),...
% '10% de cavitação no dorso \rightarrow',...
% 'HorizontalAlignment','right','FontSize',11)
%
text(SIG(round(length(SIG)/2+4)), (0.2421*SIG(round(length(SIG)/2+4))^0.6526
),...
% ' 2.5% de cavitação no dorso \rightarrow ',...
% 'HorizontalAlignment','right','FontSize',11)
%
% figure(3)
% hold on
% grid on
% title(['Integração casco-hélice com cavitação'],'FontSize',14)
% ylabel(['Potência(BHP) em KW'],'FontSize',12);

```

```

% xlabel(['Rotação em RPM'],'FontSize',12);
% plot(RPM,BHP)
%
for l=1:length(PD)
%     plot(RPM(l,:),BHP(l,:))
end
for l=1:length(PD)
    for v=1:length(V)
        if cav(l,v)==0
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','w',...
%                 'MarkerSize',3)
        elseif cav(l,v)==1
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','g',...
%                 'MarkerSize',3)
        else
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','b',...
%                 'MarkerSize',3)
        end
    end
end
end
effcav=effint.*cav;
Jcav=Jint.*cav;
KQcav=KQint.*cav;
KTcav=KTint.*cav;
BHPcav=BHP.*cav;
RPMcav=RPM.*cav;
end

```

Anexo K - Integração_casco_heliceAV.m

```

function [KTcav, KQcav, Vnos, effcav, Jcav, RPMcav, BHPcav] =
Integracao_casco_heliceAV
%-----DADOS DO NAVIO-----
Desl=488; % [ton]
L=56.5; % [m] comprimento da linha de água
calado=2.28; % [m] calado do navio
%-----DADOS DO HÉLICE-----
D=[1.7];
DAR=0.8;
%-----DADOS PARA O CÁLCULO DA CAVITAÇÃO-----
temperatura=15; % em Celsius
pv=pressao(temperatura); % sub rotina para calcular a pressão de vapor
alteixo=0.35; % altura do eixo a partir da linha de base
gravidade=9.81; % [m/s2]
rho=1025; % [Kg/m3]
atm=101300; % pressão atmosférica [N/m2]
po=atm+rho*gravidade*(calado-alteixo+0.0075*L); % pressão na altura do
hélice
%----- VALORES DE RESISTÊNCIA POR VELOCIDADE-----
%A cada velocidade o navio apresenta uma resistência ao avanço diferente.
%Vetor resistência----Res=[Rv1 Rv2 Rv3 Rv4 Rv5 Rv6 ... Rvn]
%Vetor velocidade----Vnos=[V1 V2 V3 ... Vn]
%-----
Res=[578 1389 2507 4004 5982 8559 11558 15992 21986 30215 37843 46587 55065
59539 67764 86269 109067 132401 154131 172497 186008 199697 212208 222257
228536 229648 232490]; % [N]
Vnos=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27]; % [nós]
V=Vnos*0.51444; % converte a velocidade em m/s
Pcasco=V.*Res;
%-----COEFICIENTES-----
for a=1:length(Vnos)
    Fnvol=0.165*Vnos(a)/((Desl)^(1/6));
    w(a)=0.00343+0.11152*Fnvol-0.27571*(Fnvol^2)+0.16330*(Fnvol^3)...
        -0.02828*(Fnvol^4);
    t(a)=0.43803-0.92242*Fnvol+0.81950*(Fnvol^2)-0.32145*(Fnvol^3)...
        +0.04659*(Fnvol^4);
    effrr(a)=0.81524+0.43985*Fnvol-0.47333*(Fnvol^2)+0.19918*(Fnvol^3)...
        -0.02351*(Fnvol^4)-0.00201*(Fnvol^5);
    Fnvol=0;
end
effs=0.98; % eficiência do eixo
efft=0.97; % eficiência de transmissão
%-----DADOS PARA A CURVA Kt(casco)-----
R=Res./2;
MR=0.25;
J=0:0.01:2;
KTcasco=zeros(length(Vnos), length(J)); % Gera a matriz de zeros

for b=1:length(Vnos)
    num=((1+MR)*R(b)*(J.^2));
    dem=((1-t(b)).*(V(b)*(1-w(b)).*D).^2)*1025;
    KTcasco(b,:)=num./dem; % Preenche linha a linha de KTcasco
end
%-----GERA O SPLINE DE KTcasco-----
s=0:0.0001:J(end); % gera vetor de discretização de J
KT_cascoS=zeros(length(Vnos), length(s)); % Gera a matriz de zeros

```

```

for d=1:length(Vnos)
    KT_cascoS(d,:)=spline(J,KTcasco(d,:),s);    % cria uma spline KTcasco_S
end
%-----

%-----CONSTRUÇÃO DA SÉRIE GAWN-----
PD=[0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2];
%PD=0.6:0.05:2;

A=[0.1193852    0    0    0    0;
   -0.6574682    0    0    0    1;
    0.3493294    0    0    1    0;
    0.4119366    0    0    1    1;
   -0.1991927    0    0    2    1;
    5.8630510   -2    0    2    2;
   -1.1077350   -2    0    2    3;
   -0.1341679    0    1    0    0;
    0.2628839    0    1    0    1;
   -0.5217023    0    1    1    1;
    0.2970728    0    1    2    0;
    6.1525800   -2    2    1    3;
   -2.4708400   -2    2    2    3;
   -4.0801660   -3    1    6    0;
    4.1542010   -3    1    6    1;
   -1.1364520   -3    2    6    0];

B=[1.5411660   -3    0    0    0;
   0.1091688    0    0    0    1;
  -0.3102420    0    0    0    2;
   0.1547428    0    0    0    3;
  -4.3706150   -2    0    1    0;
   0.2490295    0    0    1    2;
  -0.1594602    0    0    1    3;
   8.5367470   -2    0    2    0;
  -9.5121630   -2    0    2    1;
  -9.3203070   -3    0    2    2;
   3.2878050   -2    0    2    3;
   5.4960340   -2    1    0    1;
  -4.8650630   -2    1    1    0;
  -0.1062500    0    1    1    1;
   8.5299550   -2    1    2    0;
   1.1010230   -2    2    0    3;
  -3.1517560   -3    2    2    2];

for l=1:length(PD)
%
%Gerando Kt
    kt=A(:,1).*(10.^A(:,2)).*(DAR.^A(:,3)).*(PD(l).^A(:,4));
    for k=1:length(J)
        auxJ=J(k).^A(:,5);
        auxKt=kt.*auxJ;
        Kt(:,k)=auxKt;
    end
    KT=sum(Kt);
%
%Gerando Kq

```



```

KQint(l,v)=KQ_S(c); % preenche a matriz KQint

KQint_effrr(l,v)=KQ_S(c)/effrr(v); % cria KQint/effrr para o BHP

rot(l,v)=V(v)*(1-w(v))/(s(c)*D); % rotações em rps

tauc(l,v)=(KTint_effrr(l,v)*(rot(l,v)*D)^2)/(0.125*pi*(1.067-...
0.229*PD(l))*(V(v)*(1-w(v)))^2+(0.7*pi*D*rot(l,v))^...
2)); % cálculo da matriz de tau
% o cálculo de tau está levando em consideração que o hélice está
% posicionado na popa do navio

sigma(l,v)=(po-pv)/(0.5*rho*((V(v)*(1-w(v)))^2+(0.7*pi*D*...
rot(l,v))^2)); % cálculo da matriz de sigma
end

end
RPM=60*rot;
BHP=((2*pi*1025*D^5)*KQint_effrr.*rot.^3)./(efft))/10^3; % cria a matriz
% BHP em KW

effint=(KTint./KQint).*(Jint./(2*pi));

% %-----gráficos-----
--
% figure(1)
% hold on
% grid on
% title(['Integração casco-hélice'],'FontSize',14)
% ylabel(['Potência(BHP) em KW'],'FontSize',12);
% xlabel(['Rotação em RPM'],'FontSize',12);
% plot(RPM,BHP)
for v=1:length(V)
%
text(RPM(length(l),v),BHP(length(l),v),['\leftarrow',num2str(Vnos(v))...
], 'HorizontalAlignment',...
'left', 'FontSize',8)
end
for l=1:length(PD)
% plot(RPM(l,:),BHP(l,:))
% text(RPM(l,length(V)),BHP(l,length(V)),['',num2str(PD(l))...
], 'HorizontalAlignment',...
'right', 'FontSize',10)
end
end

% %-----
--
% %Os índices tau e sigma para a cavitação estão sendo calculados nas
linhas
% 166 e 169. os dados necessarios estao sendo inseridos no início da
rotina.
% %-----
--
%
% figure(2)
% grid on
% hold on
%
% title(['Cavitação'],'FontSize',14)
% ylabel(['\tau'],'FontSize',14);
% xlabel(['\sigma'],'FontSize',14);

```

```

%
% -----
--
% %As regressões y10 e y25 foram obtidas apartir de pontos plotados do
% %gráfico de burril apresentado no marine propeller e calculado com o
% %apoio
% %do EXCEL. Os índices R^2 das regressões são próximos de um, logo
% %apresentam
% %proximidade do real, os índices são respectivamente 0.9677 e 0.9765.
% -----
--
%

SIG=min(min(sigma)):10:max(max(sigma));          % obtém o max e o min sigma
y10=0.3777*SIG.^0.7294;
y25=0.2421*SIG.^0.6526;
% plot(SIG,y10,SIG,y25)
%
for l=1:length(PD)
    for v=1:length(V)
        hold on
        if (0.2421*sigma(l,v)^0.6526)>=tauc(l,v)
            plot(sigma(l,v),tauc(l,v),...
                'bo','LineWidth',1,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',3)
            text(sigma(l,v),tauc(l,v), ['PD,V=',num2str(PD(l))',' ',...
                num2str(Vnos(v)),'\rightarrow'],'HorizontalAlignment',...
                'right','FontSize',9)
            cav(l,v)=1;
        elseif (0.3777*sigma(l,v)^0.7294)>=tauc(l,v)
            plot(sigma(l,v),tauc(l,v),...
                'bo','LineWidth',1,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','b',...
                'MarkerSize',3)
            cav(l,v)=1;
        else
            plot(sigma(l,v),tauc(l,v),...
                'bo','LineWidth',1,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','w',...
                'MarkerSize',3)
            cav(l,v)=0;
        end
    end
end
% text(SIG(round(length(SIG)/2-5)), (0.3777*SIG(round(length(SIG)/2-
% 5))^0.7294),...
%     '10% de cavitação no dorso \rightarrow',...
%     'HorizontalAlignment','right','FontSize',11)
%
text(SIG(round(length(SIG)/2+4)), (0.2421*SIG(round(length(SIG)/2+4))^0.6526
),...
%     ' 2.5% de cavitação no dorso \rightarrow ',...
%     'HorizontalAlignment','right','FontSize',11)
%
% figure(3)

```

```

% hold on
% grid on
% title(['Integração casco-hélice com cavitação'],'FontSize',14)
% ylabel(['Potência(BHP) em KW'],'FontSize',12);
% xlabel(['Rotação em RPM'],'FontSize',12);
% plot(RPM,BHP)
%
for l=1:length(PD)
%     plot(RPM(l,:),BHP(l,:))
end
for l=1:length(PD)
    for v=1:length(V)
        if cav(l,v)==0
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','w',...
%                 'MarkerSize',3)
        elseif cav(l,v)==1
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','g',...
%                 'MarkerSize',3)
        else
%             plot(RPM(l,v),BHP(l,v),...
%                 'bo','LineWidth',1,...
%                 'MarkerEdgeColor','k',...
%                 'MarkerFaceColor','b',...
%                 'MarkerSize',3)
        end
    end
end
end
effcav=effint.*cav;
Jcav=Jint.*cav;
KQcav=KQint.*cav;
KTcav=KTint.*cav;
BHPcav=BHP.*cav;
RPMcav=RPM.*cav;
end

```

Anexo L – Arquivos referentes aos botões do modelo

Leme_ente.m:

```
global h

v1 =(get(h(1),'string'));
set_param('sim_NPa/leme','Value',v1);
```

PCL_enter:

```
global h

v3 =(get(h(3),'string'));
set_param('sim_NPa/PCL_BE','Value',v3);

v4 =(get(h(4),'string'));
set_param('sim_NPa/PCL_BB','Value',v4);
```

rumo_enter.m:

```
global h

v2 =(get(h(2),'string'));
set_param('sim_NPa/rumo','Value',v2);
```

Vento_enter.m:

```
global h

v5 =(get(h(5),'string'));
set_param('sim_NPa/Dir.','Value',v5);

v6 =(get(h(6),'string'));
set_param('sim_NPa/Int.','Value',v6);
```

Anexo M – Arquivos referentes às manobras do modelo

Manobra_centralizar.m:

```
% Manobra de centralização

global centralizar

centralizar = 1;
```

manobra_governo.m:

```
% Manobra governo

global governo

if (governo == 0)
    governo = 1;
    set_param('sim_NPa/governo', 'Value', num2str(1));
else
    governo = 0;
    set_param('sim_NPa/governo', 'Value', num2str(0));
    set_param('sim_NPa/rumo', 'Value', num2str(0));
end;
```

manobra_zigzag.m:

```
% Manobra de zig-zag

global zig_zag

if (zig_zag == 0)
    zig_zag = 1;
    set_param('sim_NPa/leme', 'Value', num2str(20));
else
    zig_zag = 0;
    set_param('sim_NPa/leme', 'Value', num2str(0));
end;
```

parada_emergencia.m:

```
% Parada emergência  
  
global par_emergencia  
  
par_emergencia = 1;
```

Anexo N - NPa_data.m

O arquivo, a seguir, é aquele que carrega todos arquivos e dados usados pelo modelo.

```
% Modelo dinâmico do NPa de 500 ton com seus coeficientes hidrodinâmicos
% baseado no modelo do navio portacontainer do anexo D do livro do Fossen

%Autor: Vitor de Oliveira Vargas - 18/3/2010

clear

% Coeficientes adimensionais do modelo

m = 5.2881e-3; % massa
mx = 1.1221e-4; % massa adicional em surge
my = 5.839e-3; % massa adicional em sway
Ix = 5.8913e-6; % momento de inércia em roll
Jx = 1.1381e-6; % momento de inércia adicional
Iz = 2.4166e-4; % momento de inércia em yaw
Jz = 3.1041e-4; % momento de inércia adicional
alfay = 4.8217e-2; % coordenada x de my
lx = .0313; % coordenada z de mx
ly = .0313; % coordenada z de my

% características do navio

L = 56.5; % comprimento entre perpendiculares em metros

% matriz da dinâmica do navio

M = [m+mx, 0, 0, 0;
     0, m+my, -my*ly, my*alfay;
     0, -my*ly, Ix+Jx, 0;
     0, my*alfay, 0, Iz+Jz];

% dados do piloto automático

T1=10.3792;
T2=3.6777;
T3=T2;

zeta = .9;
wn = .138;
k = .2503;

A=[T1*k, (1-2*zeta*wn*T1)*k;
   k, -wn^2*T1*k];
B=[2*zeta*wn*(T2+T3)-(4*zeta^2*wn^2-wn^2)*T2*T3;
   wn^2*(T2+T3)-2*zeta*wn^3*T2*T3];
K=A\B; kp=K(1); kd=K(2); ki=kp/40;
```

```

T4=T2*T3;
T5=T1*k*kd+T2+T3-2*zeta*wn*T2*T3;

p1=-zeta*wn;
p2=-T5/T4;

% carrega os dados
fprintf('-----\n');

convunit;
mtu_moddin;
reduitora;

% carrega dados adicionais

ctrl_data;
histerese_embreagens;
programas_demandados_aproximados;
[V, PD, KTtab, KQtab] = Integracao_casco_helice;
hpc;

fprintf('-----\n');

sim_NPa;

```

Anexo O - pressao.m

```
function [pres]=pressao(temp)

ss=temp;
T=[0 10 15 20 25];
P=[700 1250 1700 2300 3200];
Ts=0:0.01:T(end);
Y=spline(T,P,Ts);
TX=find(Ts==ss);
pres=Y(TX);
% plot(Ts,Y)
return
```

Anexo P - programas_demandados_aproximados.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Perfis Demandados Aproximados Por Segmentos de Retas %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Aproximações MOTOR DIESEL %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Aproximações de Rotação %%%%
PCL_Diesel_ROT=          [-100 -77 -54 -31 0 48 52 56 60 64 68 72 76 80 84
88 92 96 100];
ROT_Diesel=          [ 1099 852 607 360 350 357 388 411 440 482 526 604 645 681
712 743 826 854 870];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Aproximações de Rotação Hélice %%%%
PCL_Eixo_ROT=          [-100 -77 -54 -31 0 48 52 56 60 64 68 72 76 80 84 88
92 96 100];
ROT_Eixo=          [743 576 410 243 236 241 261 278 297 326 356 408 436 460 481
502 558 577 594];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Aproximações de passo %%%%
PCL_PASSO=          [-100 -1 0 48 68 72 88 92 100]; %PCL para PASSO
PASSO_PASSO=          [-10.8 -10.8 0 22.5 22.5 20.9 20.9 19.3 19.3]; % PASSO

```

Anexo Q - redutora.m

```

% Redutora
% Parâmetros da engrenagem redutora e acoplamento fluido

fprintf('Redutora e acoplamento fluido ... ');

% Momento de inércia do mecanismo antes do primário do AF
J_pr_AF=147.58; % [kgm^2]

% Momento de inércia do mecanismo depois do secundário do AF com SSS
% engatada
J_se_AF=9686.13; % [kgm^2]

% Momento de inércia do mecanismo depois do secundário do AF com SSS
% desengatada
J_se_AF_deseng=45.3;

% Taxa de amplificação do torque
RRF=1.48;

% Taxa de redução da rotação entre pinhão e eixo
RRP=1/1.48;

% Acoplamento fluido: constante de ajuste.

c=0.0285;

% Acoplamento fluido: enchimento

% constante de tempo.

tt=10; % equivale a tempo de 10 segundo para sair de 0 a 1

% curva sigmóide

sig_x=[0 0.15 0.5 0.85 1];
sig_y=[0 0.05 0.5 0.95 1];
ssig_x=0:0.005:1;
ssig_y= interp1(sig_x,sig_y,ssig_x,'spline');

% Rotação mínima

n1_min=0.1*515.515;

% Perdas Diesel

Cperdas=0;

```

```
% potência de acionamento (porcentual da potência máxima)
p_acio=[10 12.5 15 17.5 20 22.5 25 27.5 30 35 40 50 60 70 80 90
100]*(1/100);

% Potência máxima do MCP

P_max_MCP=5200e3; % [W]

% rendimento correspondente à potência de acionamento
eta=[89.7 91.6 92.9 93.85 94.5 95.05 95.5 95.85 96.15 96.55 96.85 97.35
97.7 97.95 98.15 98.3 98.4]*(1/100);

fprintf('OK\n');
```

Anexo R - Vento.m

```

function [X,Y,N] = vento(a,b,c,d,e)

dirw = a;Vw = b;
vx = c;
v = d;
psi = e;

if dirw < pi, psiw = dirw + pi;
else psiw = dirw - pi;
end

gamaR = psiw-psi;

uR = Vw*cos(gamaR) - vx;
vR = Vw*sin(gamaR) - v;
if Vw>0, VR = sqrt(uR^2 + vR^2);
else VR=0;
end

CX = .9*cos(gamaR);
CY = .85*sin(gamaR);
CN = .125*sin(2*gamaR);

% dimensões do navio
AT = 60.2; %área transversal projetada
AL = 215; % área lateral projetada
L = 58.329; % comprimento total
ro = 1.2; % densidade do ar (kg/m^3)

X = (1/7.6)*CX*ro*(VR^2)*AT;
Y = (1/7.6)*CY*ro*(VR^2)*AL;
N = (1/7.6)*CN*ro*(VR^2)*AL*L; end

```