

LEONARDO ISHIDA ABE

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE MÓDULOS DE  
SIMULAÇÃO EM TEMPO REAL PARA O SIMULADOR TPN**

Trabalho de Formatura 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

Abe, Leonardo Ishida

Desenvolvimento e Implementação de Módulos de Simulação em  
Tempo Real para o Simulador TPN/ Leonardo Ishida Abe. – São Paulo, 2010.

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

1. Simulação 2.Tempo Real 3.Sistema de Posicionamento DinâmicoI.  
Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia  
Mecatrônica e Sistemas Mecânicos II.t

*Dedico este trabalho ao meu  
avô Eng. Valter Ishida,  
ilustre Politécnico da turma  
de 57 e à minha avó Yoko  
Habe Ishida, pelos anos de  
dedicação e carinho.*

## AGRADECIMENTOS

Ao Prof. Dr. Eduardo Aoun Tannuri pelos conselhos e recomendações ao longo do desenvolvimento deste trabalho, em especial com relação a temas relativos a controle e Engenharia Naval.

Aos pesquisadores do Tanque de Provas Numérico Denis Taniguchi, Asdrúbal do Nascimento Queirós Filho, Edgar Szilagy, Fernando Henrique de Paula Luz, Silvio Luis Stanzani, Alexandre Massao Miyagi, Henrique Magalhães Soares, e em especial a Humberto Shigueki Makiyama, pela idéia inicial e apoio ao longo de todo trabalho.

Aos professores do Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da Escola Politécnica pelos conhecimentos passados ao longo da graduação, que proporcionaram minha base sólida em Engenharia.

À Escola Politécnica da Universidade de São Paulo, pelas oportunidades únicas disponibilizadas, permitindo ao aluno crescer tanto no mercado de trabalho quanto na área acadêmica.

Aos meus pais, Cecília e Eduardo, e à minha irmã Gabriela, pela dedicação incondicional, sempre muito presentes em todos os momentos da minha vida.

Ao meu pai, Mário Akito Abe, Politécnico da Turma de 75, que sempre acreditou no meu potencial como Engenheiro.

À Louise Nagashima Omi, pelo carinho e paciência comigo em todo esse período.

Aos meus amigos e mentores, Arthur Anthony Hill e Antônio Carlos Martos, pelas orientações e conselhos valiosos que direcionaram com sucesso este meu início de carreira profissional.

Aos meus amigos, grandes companheiros de todas as horas.

## RESUMO

O Tanque de Provas Numérico é o resultado de uma convenio de pesquisa entre a Universidade de São Paulo e a Petrobrás, cujo objetivo é auxiliar na análise de sistemas flutuantes, utilizando ferramentas de cálculos de esforços hidrodinâmico em computação paralela em *cluster* e sistemas de visualização tridimensional em realidade virtual. Projetos atualmente em desenvolvimento estão sistemas de Posicionamento Dinâmico, utilizado em estabilização de embarcações.

Este projeto consiste em desenvolver um ambiente de simulação em tempo real em imersão total para os aplicativos do laboratório, com maior interatividade entre os dados calculados e o usuário, com possibilidade de realizar interferências nas simulações.

Palavras-chave: Simulação em tempo real; Joystick; cálculo de esforços hidrodinâmicos; socket UDP; Plataforma Stewart.

## Abstract

The Tanque de Provas Numérico is the result of a research agreement between the São Paulo University and the Petrobras, whose objective aid project analysis of floating systems, using powerful hydrodynamic analysis tools in *cluster* parallel computation and virtual reality tridimensional programs for visualization. Current work in progress are Dynamic Positions systems, used in FPSOs (*Floating Production Storage and Offloading*) as a replacement for oil rigs.

This project consist on the development of an interactive total immersion real time simulation environment for the softwares used on the laboratory, allowing interferences on the scenario being simulated.

Keywords: Real time simulation; Joystick; hydrodynamics stress analysis; socket UDP, Stewart Platform.

## SUMÁRIO

<b>Lista de Figuras.....</b>	<b>VI</b>
<b>Lista de Tabelas.....</b>	<b>VII</b>
<b>GLOSSARIO.....</b>	<b>VIII</b>
<b>1. Introdução.....</b>	<b>1</b>
1.1. Contextualização do trabalho .....	1
1.1.1. O Tanque de Provas Numérico .....	1
1.1.2. TPN.....	2
1.1.3. TPNView .....	2
1.1.4. O TPN em modo <i>offline</i> .....	3
1.2. Objetivos .....	4
1.2.1. Módulo de entrada de dados.....	4
1.2.2. Módulo de comunicação.....	5
1.2.3. Módulo de <i>feedback</i> .....	5
1.3. Justificativa.....	6
1.4. Trabalhos semelhantes.....	6
1.5.  Conteúdo dos capítulos.....	7
<b>2. Desenvolvimento teórico .....</b>	<b>7</b>
2.1. Protocolo UDP .....	7
2.2. Sistema de Posicionamento Dinâmico .....	8
2.3. Cálculo de altura de ondas.....	9
2.4. Cálculo de espectro .....	10
2.4.1. Regular .....	10
2.4.2. Gaussiano .....	11
2.4.3. Pierson-Moskowitz.....	11
2.4.4. JONSWAP .....	12
2.4.5. Cálculo das alturas para onda Unidirecional sem espalhamento .....	13
<b>3. Metodologia .....</b>	<b>13</b>
3.1. Estratégia de trabalho .....	14
<b>4. Implementação .....</b>	<b>15</b>
4.1. Módulo de entrada de dados.....	15
4.1.1. Joystick .....	16
4.1.2. Simulador de Sistemas de Posicionamento Dinâmico acoplado à entrada de dados.....	17
4.2. Módulo de comunicação.....	22
4.2.1. TPNClient.....	23
4.2.2. Transmissão de Corpos Flutuantes .....	23
4.2.3. Transmissão de posições de linha .....	24
4.2.4. Transmissão de forças de linha.....	25
4.2.5. Mapa de Ondas .....	26
4.3. Módulo de <i>Feedback</i> .....	28
4.3.1. Plataforma Móvel .....	28
<b>5. Conclusões e proposta de pesquisas futuras.....</b>	<b>39</b>
5.1. Conclusões .....	39
5.2. Proposta de pesquisas futuras .....	40
<b>6. Bibliografia .....</b>	<b>41</b>

## LISTA DE FIGURAS

Figura 1: Logo TPN antigo .....	1
Figura 2: diagrama de blocos TPN antigo .....	4
Figura 3: Diagrama de blocos em simulação em tempo real .....	6
Figura 4: demonstração dos movimentos, adaptado de Tannuri, 2009 .....	8
Figura 5: Registro de ondas tipico com os valores de H amostrados .....	9
Figura 6: Função distribuição de Probabilidade dos H amostrados .....	10
Figura 7: Tela do aplicativo Joystick Virtual .....	17
Figura 8: Diagrama de blocos do sistema de Posicionamento Dinâmico com Joystick .....	19
Figura 9: Diagrama de bloco do sistema do módulo de comunicação .....	22
Figura 10: Esquema do controle de tempo para uma simulação de 0,5 segundos de passo. ....	23
Figura 11: Esquema ilustrando a extrusão do perfil de onda, nota-se a rotação do eixo da onda de ângulo $\alpha$ de incidência. ....	27
Figura 12: Perfil de ondas típico calculado .....	27
Figura 14: Série temporal utilizada para testes .....	30
Figura 15: Série temporal com redimensionamento simples .....	31
Figura 16: Série temporal com filtro de acelerações .....	32
Figura 17: Detalhe da mesma série, evidenciando as descontinuidades de velocidade .....	33
Figura 18: Série com filtro de acelerações e passando FFT, com filtro de acelerações apenas (azul) e com FFT(verde) .....	34
Figura 19 :Série temporal com Filtro de velocidades .....	35
Figura 20: Detalhe da série temporal com o filtro .....	36
Figura 21: Filtro com logaritmo .....	37
Figura 22: Detalhe do arredondamento das pontas para o filtro de logaritmo (vermelho) e redimensionamento simples (azul) .....	37
Figura 23: Comparação de pequenos movimentos entre redimensionamento simples (azul) e logaritmo(vermelho) .....	38

**LISTA DE TABELAS**

Tabela 1: Tabela de dados padrão TPN .....	16
Tabela 2: Modos do Joystick ativo .....	20
Tabela 3: Parâmetros do módulo de controle .....	21
Tabela 4 - Pacote para corpos flutuantes .....	23
Tabela 5 - Detalhamento da carga de posições .....	24
Tabela 6 - Detalhamento da carga de velocidades .....	24
Tabela 7 - Detalhamento da carga de acelerações .....	24
Tabela 8 - Pacote para posições dos nós da linha .....	25
Tabela 9 - Pacote da carga para posição de cada nó .....	25
Tabela 10 - Pacote para posições das forças da linha .....	26
Tabela 11: Pacote da carga para forças de cada elemento .....	26
Tabela 12: Pacote de Ondas do mar .....	28
Tabela 13: Tabela com os limites da plataforma .....	29
Tabela 14: Pacote enviado à plataforma .....	38
Tabela 15: Palavras de comando disponíveis para à plataforma .....	39

## GLOSSARIO

Movimento de <i>Roll</i>	Rotação no plano vertical em torno do eixo transversal (caturro)
Movimento de <i>Pitch</i>	Rotação no plano vertical em torno do eixo longitudinal (balanço ou jogo)
Movimento de <i>Yaw</i>	Rotação no plano horizontal (guinada)
Movimento de <i>Surge</i>	Translação longitudinal (avanço)
Movimento de <i>Sway</i>	Translação lateral (deriva)
Movimento de <i>Heave</i>	Translação vertical (arfagem)



# 1. Introdução

Este trabalho de formatura do curso de Engenharia Mecatrônica da Escola Politécnica trata do desenvolvimento de adaptações que permitem execução de simulações em tempo real de sistemas flutuantes no aplicativo de cálculo hidrodinâmico TPN, desenvolvido no laboratório Tanque de Provas Numérico da Engenharia Naval da Escola Politécnica da Universidade de São Paulo.

Como resultado deste projeto espera-se obter um novo ambiente de simulação, mais ágil e com maior interação entre o usuário e a simulação em execução, melhorando o processo de análise e abrindo novas possibilidades de ensaios.

## 1.1. Contextualização do trabalho

### 1.1.1. O Tanque de Provas Numérico

O Tanque de Provas Numérico é o resultado de um convênio de pesquisa iniciado em 2002 entre a Universidade de São Paulo e a Petrobrás. Desde então o laboratório



Figura 1: Logo TPN  
antigo

vem atuando na análise e no projeto de sistemas flutuantes utilizando como ferramenta principal um simulador de processamento paralelo desenvolvido por equipe de pesquisadores. As simulações dinâmicas caracterizam com precisão os fenômenos hidrodinâmicos e estruturais atuantes no sistema de produção *offshore* incluindo particularidades do projeto como, por exemplo, a existência de sistemas de posicionamento ou os efeitos decorrentes de *sloshing* em tanques com superfície livre.

Para prover a capacidade de analise integrada de sistemas foi necessária a instalação de um “cluster” de computadores. Um *cluster* é um agregado de processadores dedicados, trabalhando uma única tarefa de forma cooperativa e integrada, isto é, as operações de cálculo são subdivididas e distribuídas, via rede, pelos processadores que o constituem. O *cluster* utilizado no laboratório foi providenciado pela SUN Microsystems e é dividido em duas unidades: A primeira conta com 192 blades X6175, racks C48 refrigerados à água, 15 TFlops de processamento, 1536 cores Intel Nehalem de 2.8Ghz, 4.5 TB memória RAM total e 150TB de espaço para

armazenamento. A segunda unidade conta com 16 servers X4440, 256 cores AMD Shanghai de 2.66GHz, 1TB de memória RAM e 8 unidades de processamento vetorial nVidia Tesla S1070, totalizando 2 TFlops de processamento genérico e 30 TFlops de processamento vetorial.

Para ampliar a capacidade interpretativa dos resultados, foi construída uma sala de visualização 3-D que utiliza equipamentos de visualização e sonorização estereoscópica de ultima geração, inclusive com instalação de uma plataforma móvel de simulação dos movimentos, cujas funcionalidades foram implementadas neste projeto.

### **1.1.2. TPN**

O simulador numérico hidrodinâmico TPN, trabalha com discretização temporal do comportamento dinâmico do sistema flutuante sob a ação de esforços externos, sejam estes decorrentes das condições ambientais (corrente, vento e ondas), da dinâmica das linhas de amarração e *risers*) ou da interação de corpos múltiplos (efeito sombra) (Fucatu, 1998, Fucatu, Nishimoto e Masseti, 2001, Nishimoto, Donato, Maseti, Jacob, Martins, Meneses e Hirata, 2003). O TPN é capaz de realizar uma analise acoplada de todos estes parâmetros provendo agilidade e economia no processo de análise e desenvolvimento de sistemas flutuantes.

As principais funcionalidades do TPN são: cálculos de esforços em corpos flutuantes, cálculo de atuação de forças externas de ondas, ventos e corrente, cálculo de esforços em linhas e desenvolvimento de controles para sistemas de Posicionamento Dinâmico, utilizados no controle de embarcações. Dentre seus projetos estão plataforma monocolumna MonoBR, o FPSOBR, a plataforma P52, o SSB – boião de sub-superfície, os projetos em parceria com a marinha da operação de *load-out* do submarino Timbira, a docagem do porta-aviões NAeSP e a montagem do submarino Tikuna.

### **1.1.3. TPNView**

Os resultados das simulações realizadas no TPN geram um volume imenso de dados, o que torna a análise e interpretação uma tarefa complexa. Com o intuito de facilitar esta tarefa a equipe do Tanque de Provas Numérico desenvolveu o programa de visualização TPNView: uma ferramenta de visualização científica, com especial cuidado na coerência geométrica e realismo gráfico e, principalmente, na facilidade de

detecção de propriedades relativas ao comportamento da embarcação frente às condições ambientais.

O TPNView está baseado em um ambiente tridimensional capaz de criar imersão *in-situ* do analista no projeto, incluindo interfaces que permitem acesso a todos os dados, assim como uma extensa gama de ferramentas de análise de dados (gráficos, histogramas e estatísticas).

#### **1.1.4. O TPN em modo *offline***

Em modo “*offline*” TPN recebe as condições ambientais da simulação de um arquivo .p3d, realiza os cálculos e gera uma planilha com todos os resultados obtidos. Esta planilha é transferida para o TPNView, que se encarrega de transformar os dados em uma maquete eletrônica. Esta maquete se comporta como um vídeo, permitindo pausar, avançar e retroceder a simulação. A interface entre usuário e a simulação executada é feita por uma sequência de comandos em um arquivo de texto.

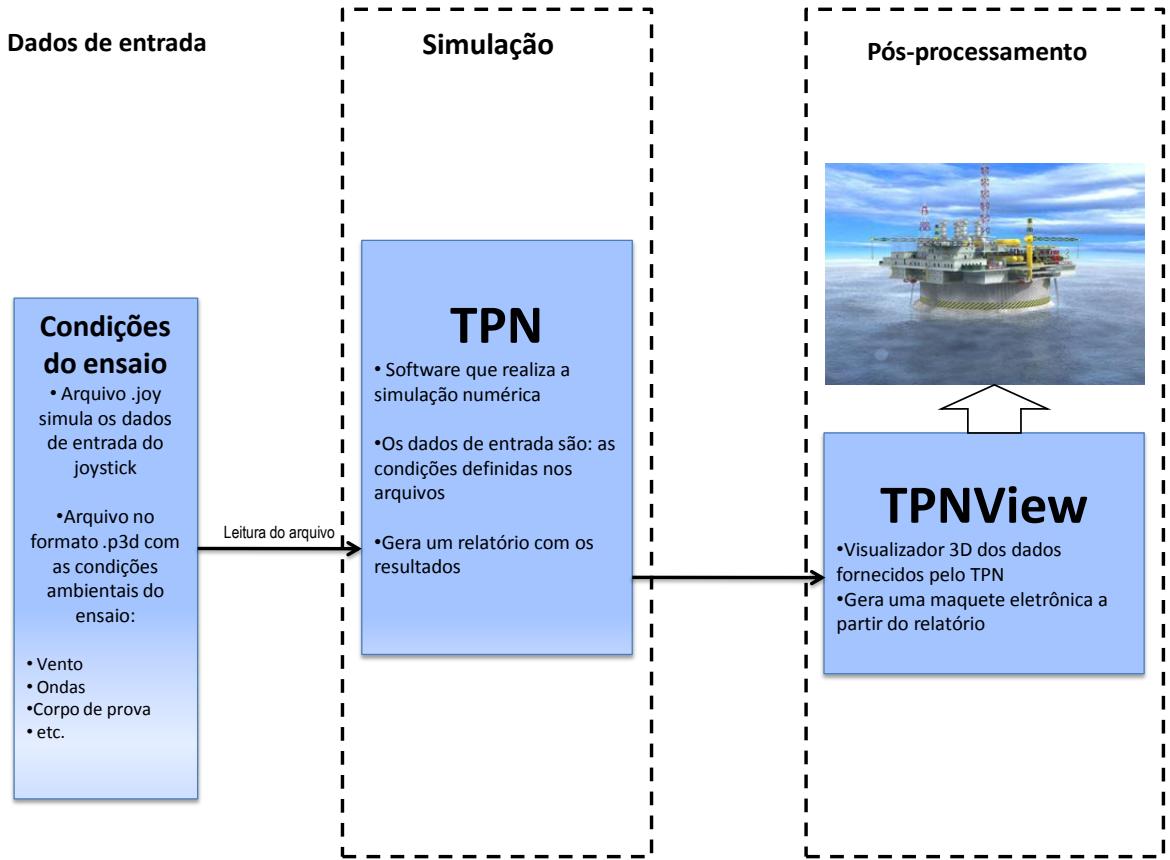


Figura 2: diagrama de blocos TPN antigo

## 1.2. Objetivos

O objetivo deste trabalho é criar módulos que permitam executar simulações hidrodinâmicas em tempo real. Este projeto é composto por três frentes:

1. Módulo de comunicação.
2. Módulo de entrada de dados.
3. Módulo de *feedback*.

### 1.2.1. Módulo de entrada de dados

Um módulo responsável por criar um canal de comunicação entre o usuário e a simulação sendo executada. Esta frente de trabalho deverá permitir que o usuário gere interferências no que está sendo calculado, permitindo avaliar os seus resultados instantaneamente.

### **1.2.2. Módulo de comunicação**

Este módulo é responsável pelo estabelecimento e manutenção de fluxo de dados entre o simulador e o software de visualização

Os dados enviados são divididos em três categorias:

- Corpos Flutuantes (*Vessels*): Englobam embarcações, bóias, etc. É composto por posições, velocidades e acelerações
- Linhas (*Lines*): Consiste nas linhas de amarração, *risers* e *hausers*, discretizados ao longo do comprimento. Cada linha é composta por dois vetores: um composto por **n** nós, armazenando as posições no espaço, e outro composto por **n-1** elementos entre cada nó, armazenando as forças incidentes.
- Ondas (*Wave*): Um mapa de alturas calculado utilizando métodos estatísticos de cálculo de ondas irregulares, sincronizado com a movimentação dos corpos flutuantes.

### **1.2.3. Módulo de *feedback***

Criar uma nova maneira de interação entre o operador e os dados exibidos no TPNView, na forma de movimentos físicos, aumentando grau de imersão da simulação com o usuário.

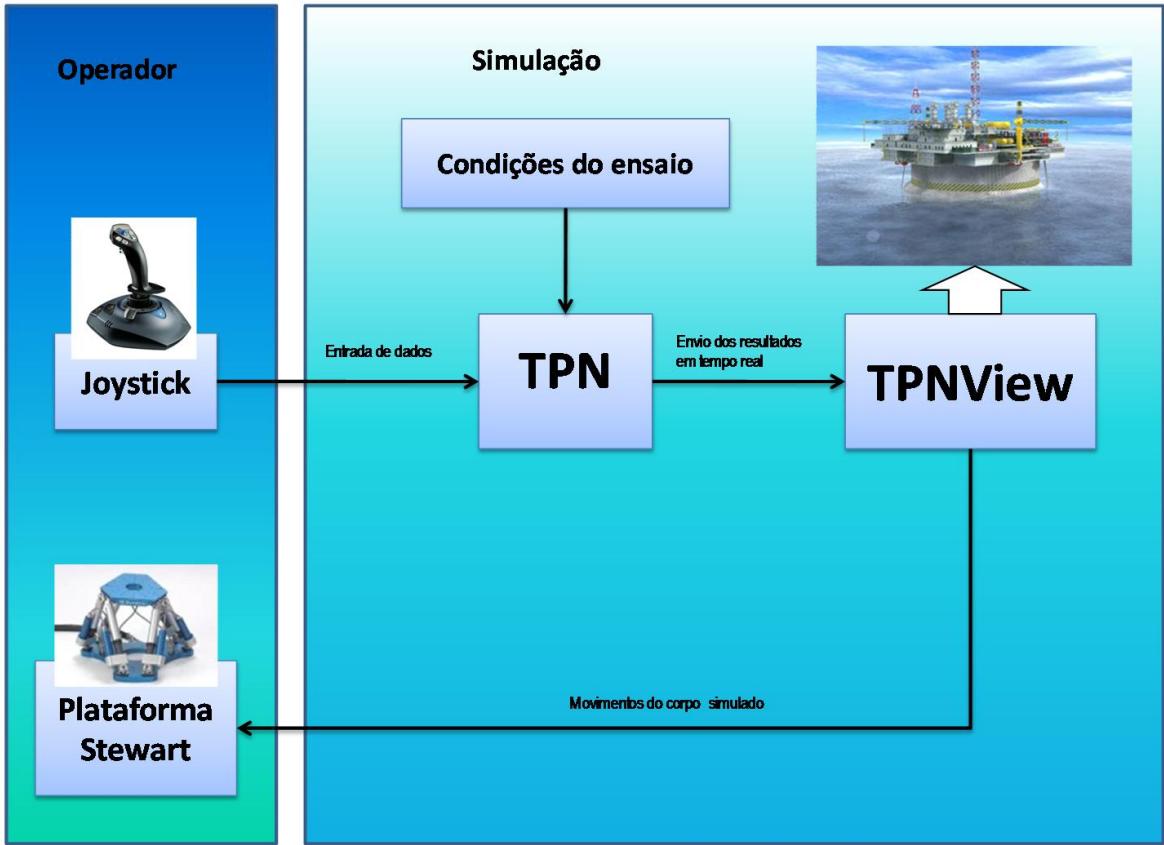


Figura 3: Diagrama de blocos em simulação em tempo real

### 1.3. Justificativa

Em modo *offline* a maquete é criada realizando pós-processamento nos dados pelo TPN. Com a simulação em tempo real os dados podem ser visualizados imediatamente assim que são calculados, agilizando o processo de análise da simulação. Com os novos módulos de entrada de dados e *feedback*, a interatividade entre o usuário e a simulação recebe grandes melhorias, tornando a análise mais atraente ao operador.

### 1.4. Trabalhos semelhantes

Simulações em tempo real em imersão têm sido utilizadas largamente para treinamento de marinheiros. No VTC (*Vocational Training Concil*) de Hong Kong o departamento de treinamento de serviços marinhos (MSTI) conta com três simuladores desta categoria. O *Applied Research International* na India apresenta também um

simulador neste formato, com aplicações similares.

A *Fag-skoler* da Noruega disponibiliza um vídeo de uma simulação de embarcação de grande porte em mar agitado, com um impressionante realismo gráfico.

## 1.5. |Conteúdo dos capítulos

O capítulo 2 expõe tópicos teóricos utilizados durante o projeto, explicitando a sua aplicação para o desenvolvimento dos módulos.

O capítulo 3 mostra a metodologia utilizada para o desenvolvimento do projeto, com relação ao aparato utilizado e estratégia de trabalho.

O capítulo 4 exibe as especificações do projeto, as soluções propostas e os resultados das implementações.

O capítulo 5 traz as conclusões retiradas da análise dos programas executados, avaliando sobre as possibilidades e restrições dos módulos implementados, apontando para as possíveis continuidades deste projeto.

# 2. Desenvolvimento teórico

## 2.1. Protocolo UDP

O UDP é um dos principais protocolos de rede utilizados para Internet. Suas principais características são: baixa confiabilidade de transmissão, possibilidade de perda de pacote, ausência de *hand-shake*, alta velocidade e possibilidade de recebimento de múltiplos clientes simultâneos. É utilizado em aplicações que necessitem transmissão em tempo real sem preocupações com a integridade de todos os pacotes (Ex: VoIP e jogos online). Seu funcionamento é bastante simples, necessitando apenas dos dados, da porta utilizada e o endereço IP do servidor para envio dos dados.

## 2.2. Sistema de Posicionamento Dinâmico

Define-se posicionamento dinâmico como um sistema que controla automaticamente a posição e apropamento de uma embarcação por meio de propulsão ativa (Bray, 1998 e Fossen, 1994). Estes movimentos são conhecidos como *surge*, *sway* e *yaw*. O SPD constitui em um complexo sistema de controle, composto por sensores, atuadores e um núcleo que executa o algoritmo de controles com interface com o operador.

A medição de posição e apropamento provenientes de sensores são filtradas por um filtro de onda. Em alto-mar as forças provocadas pela correnteza onda e vento geram movimentos de alta freqüência e baixa freqüência. O objetivo do SPD é controlar exclusivamente os movimentos de baixa freqüência horizontais, pois os movimentos de alta freqüência exigiram uma potencia muito elevada, o que pode levar a desgastes dos propulsores.

O algoritmo de controle é responsável por calcular os valores das forças resultantes e momento necessário para manter a posição da embarcação, comparando os valores obtidos pelo sensoriamento com a posição requerida (*set-point*). Um controlador do tipo PD (proporcional-derivativo) é aplicado a cada um dos três graus de liberdade. Os valores das forças calculadas são distribuídas entre os atuadores por meio de um algoritmo de otimização não-linear com restrições. A influência do vento é compensada por uma malha de *feedforward*.

O TPN apresenta um módulo de controle de corpos flutuantes por SPD, testado e validado, com entrada de dados de posição e modos de controle pré-definidos por um arquivo de texto.

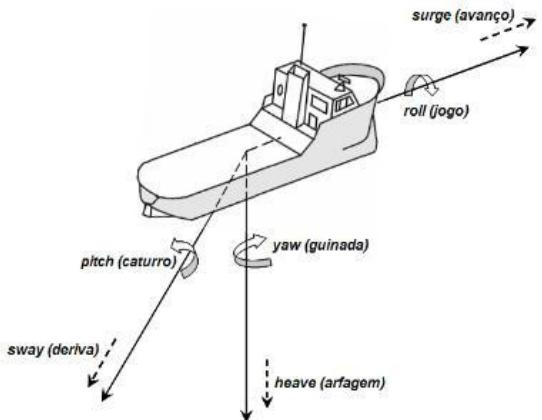


Figura 4: demonstração dos movimentos, adaptado de Tannuri, 2009

### 2.3. Cálculo de altura de ondas

Dentre os diferentes fenômenos físicos responsáveis por induzir efeitos ondulatórios no ambiente marítimo, encontramos a geração de ondas causada pela ação do vento sobre a superfície do mar. As ondas de superfície originadas por esta ação são aquelas que apresentam maior interesse no contexto da engenharia naval e oceânica, uma vez que apresentam períodos e amplitudes típicas capazes de excitar de forma significativa a dinâmica de navios ou sistemas oceânicos usuais. Estas ondas freqüentemente apresentam caráter bastante aleatório, com períodos e alturas de ondas variando continuamente com o tempo e, muitas vezes, com propagação em diversas direções. Essa aleatoriedade nos obriga, então, a uma modelagem estatística das ondas do mar, com o intuito de extraímos informações importantes sobre os efeitos causados por diferentes “estados de mar” sobre navios ou sistemas oceânicos (Muckle, 1975).

Ondas irregulares do mar podem ser representadas por uma sobreposição de ondas regulares de diferentes amplitudes e freqüências. Supondo um registro de típico de ondas  $\zeta(t)$ :

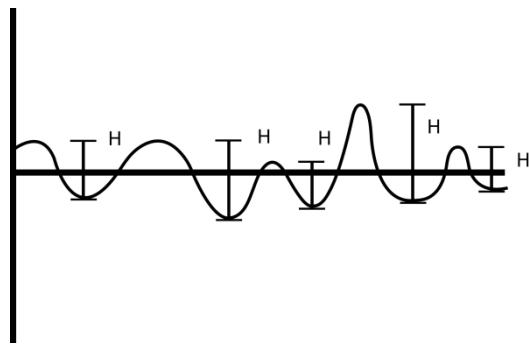


Figura 5: Registro de ondas tipico com os valores de H amostrados

Temos a amostragem de alturas de onda  $H$ , definida por:

$$H = \zeta_{crista} - \zeta_{proxima\ depressão} \quad (1)$$

Gerando uma função de distribuição de Probabilidade:

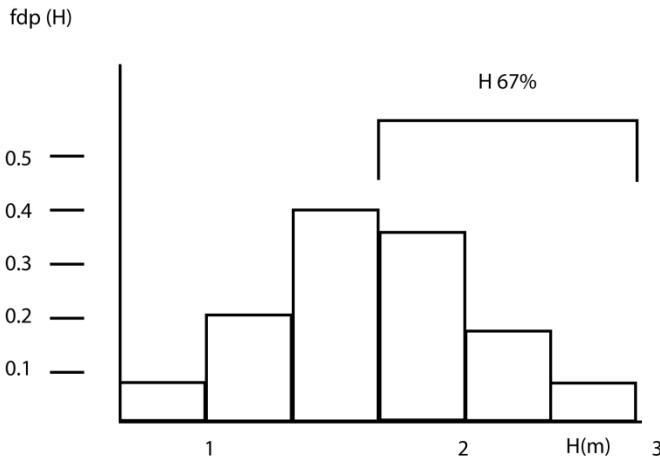


Figura 6: Função distribuição de Probabilidade dos H amostrados

Para o cálculo do espectro de ondas, é utilizada a Altura significativa do registro ( $H_s$ ), definida pela média das alturas dos 33% maiores registros, logo:

$$H_s = \int_{H_{67\%}}^{H_{100\%}} H * f dp(H) dH \quad (2)$$

## 2.4. Cálculo de espectro

Para calculo dos espectros são utilizados os seguintes métodos:

- Regular
- Gaussiano
- Pierson-Moskowitz
- JONSWAP

O primeiro aplicado para uma onda senoidal simples, o segundo para uma distribuição de freqüências gaussiana e os dois últimos para ondas irregulares qualquer.

### 2.4.1. Regular

Para ondas regulares (sem superposição de ondas) há apenas é utilizada apenas uma freqüência:

$$\omega_0 = \frac{2\pi}{T} \quad (3)$$

Resultando em um espectro dado por apenas um valor:

$$S = \frac{H_s^2}{8.0} \quad (4)$$

Onde:

$S$  - Energia do espectro.

$T$  - Período médio.

$H_s$  – Altura significativa.

## 2.4.2. Gaussiano

Para espectro que segue uma distribuição de freqüências Gaussiana, o espectro é calculado como:

$$S_\zeta(\omega) = \frac{\exp\left(-50(\omega-1)^2(0.25H_s)^2\right)}{0.250662\omega_0} \quad (5)$$

Onde:

$S(\omega)$  – função densidade espectral.

$\omega$  – freqüência circular da onda.

$\omega_0$  – freqüência inicial.

$H_s$  - altura significativa.

$T$  – Período médio.

## 2.4.3. Pierson-Moskowitz

O espectro proposto por Pierson & Moskowitz foi obtido de forma semi-empírica com base na análise de um grande numero de registros de ondas do Atlântico Norte. Por ter sido obtido em região de mar aberta, supostamente se referem a mares plenamente desenvolvidos. O espectro resultante é dado por:

$$S(\omega) = (H_s^2 T) \frac{0.11}{2\pi} \left(\frac{\omega T}{2\pi}\right)^{-5} \exp\left[-0.44\left(\frac{\omega T}{2\pi}\right)^{-4}\right] \quad (6)$$

Onde:

$S(\omega)$  – função densidade espectral.

$\omega$  – freqüência circular da onda.

$H_s$  - altura significativa.

$T$  – Período médio.

#### 2.4.4. JONSWAP

Entre 1968 e 1969 um extenso projeto de monitoração de ondas, conhecido como *Joint North Sea Wave Project* (JONSWAP), foi conduzido no Mar do Norte ao longo de uma linha de 100 milhas com origem na ilha Sylt. A análise dos dados resultou na proposta de um formato de espectro para mares gerados em pistas limitadas (*fetch-limited*) ou costeiros. O espectro é definido por:

$$S(\omega) = \alpha g^2 \omega^{-5} \exp\left[-125\left(\frac{\omega}{\omega_p}\right)^{-4}\right] \gamma^{\exp\left[\frac{\left(\frac{\omega}{\omega_p}-1\right)^2}{2\sigma^2}\right]} \quad (7)$$

$S(\omega)$  – função densidade espectral.

$\omega$  – freqüência circular da onda.

$H_s$  - altura significativa.

$T$  – Período médio.

$\alpha, \gamma, \sigma$  - parâmetros de forma do espectro.

$\omega_p$  - freqüência de pico.

Sendo  $\sigma$ :

$$\begin{aligned} \sigma &= 0.07 \text{ se } \omega \leq \omega_p \\ \sigma &= 0.09 \text{ se } \omega > \omega_p \end{aligned} \quad (8)$$

Dado que:

$$H_s^2 = 16 \int_0^\infty S(\omega) d\omega \quad (9)$$

Então o parâmetro  $\alpha$ , função da altura significativa da onda, é calculado pela formula:

$$Hs^2 = 16\alpha g^2 \int_0^\infty \omega^{-5} \exp \left[ -125 \left( \frac{\omega}{\omega_p} \right)^{-4} \right] \gamma \exp \left[ \frac{\left( \frac{\omega}{\omega_p} - 1 \right)^2}{2\sigma^2} \right] d\omega \quad (10)$$

#### 2.4.5. Cálculo das alturas para onda Unidirecional sem espalhamento

Com os valores de espectro calculados, podemos obter a distribuição de ondas para um sistema unidirecional por uma soma de cossenos:

$$z = \sum_{i=1}^n \sqrt{A(i)} \cos(-K(i)x + \omega(i)t + F(i)) \quad (11)$$

Sendo

$$\begin{aligned} K(i) &= \frac{\omega(i)^2}{G} \\ A(i) &= \frac{2S(i)}{\Delta\omega(i)} \\ F(i) &= 2\pi * rnd \end{aligned}$$

Onde:

$\omega(i)$  – Freqüência de onda.

$S(i)$  – Espectro correspondente à  $\omega(i)$ .

$A$  – Amplitude da onda correspondente à  $\omega(i)$ .

$K(i)$  – Número da onda correspondente à  $\omega(i)$ .

$F(i)$  – Fase da onda correspondente à  $\omega(i)$ .

$\Delta\omega(i)$  – Diferença entre o valor atual de  $\omega$  e o anterior, caso não haja valor anterior  $\Delta\omega(i) = \frac{\omega(i)}{2}$ .

$rnd$  - valor aleatório gerado a partir de uma semente, garantindo aleatoriedade da fase.

### 3. Metodologia

Para o desenvolvimento dos trabalhos foi utilizado a seguinte metodologia:

Computadores Quad-core Intel em plataforma Linux Ubuntu em compilador gcc

versão 4:4.4.1. Para compilação dos códigos foram utilizados scripts de SConstruct (linguagem Python) para o TPN e Makefile para o módulo do joystick, gerador de onda e programas testes dos sockets. Para o desenvolvimento dos aplicativos desenvolvidos em plataforma Windows foi utilizado o ambiente de desenvolvimento (API) Qt Creator da Nokia.

A linguagem utilizada é C++, biblioteca STL com extensão TR1 (Meyers, 2005, Holzner 2001), bibliotecas Boost, uma coleção de bibliotecas que funcionam como uma extensão da linguagem, com especial atenção à *boost::asio* aplicada para configuração de rede em Linux, biblioteca Qt inclusa no ambiente de desenvolvimento *Qt Creator* para aplicativos em windows, e SDL, *framework* voltado para desenvolvimento de jogos utilizados para leitura e implementação do joystick.

### **3.1. Estratégia de trabalho**

Os módulos desenvolvidos neste projeto seguem o seguinte protocolo:

1. Definição das funcionalidades, especificações e restrições do módulo.
2. Desenvolvimento de classes para cálculo de dados, caso necessário. Neste projeto o módulo gerador de onda e o módulo de *feedback* apenas passam por este processo.
3. Teste e validação dos cálculos implementados anteriormente. Neste projeto o módulo gerador de onda e o módulo de *feedback* apenas passam por este processo.
4. Definição dos pacotes quanto a: conteúdo, tamanho, seqüencia e formato do pacote.
5. Definição da estrutura do cliente e servidor, atentando sempre a possíveis incongruências em caso de diferenças entre os computadores comunicantes.
6. Criação de dois grupos cliente-servidor que se comunicam entre si – ou seja, o cliente de um dos grupos deverá se comunicar com o servidor do outro grupo e vice-versa. Desta forma é possível o desenvolvimento das partes de forma independente e com garantia de transferência.

7. Testes e validação da comunicação de rede.
8. Definição da estratégia de incorporação dos novos módulos aos programas atendendo às especificações de coleta de dados e operação dos módulos implementados.
9. Incorporação dos módulos aos programas (TPN e TPNView).
10. Testes e validação dos programas com os módulos implementados.

## 4. Implementação

Neste capítulo será exposto a implementação dos módulos propostos, desde a definição das especificações e restrições às soluções propostas e suas limitações.

### 4.1. Módulo de entrada de dados

As especificações deste módulo são:

- Criar uma forma de entrada de dados pelo usuário de forma interativa e imediata durante a simulação.
- Os dados transmitidos devem ser confiáveis e bem comportados para a aplicação escolhida no TPN.
- A comunicação entre o módulo e o TPN deve ser feita remotamente, tornando desnecessária a conexão física, caso seja usado algum periférico.

Atendendo às exigências foi proposto um aplicativo cliente-servidor de joystick, periférico utilizado em jogos eletrônicos, que executa a leitura do dispositivo remotamente (cliente) e envia os dados via protocolo UDP ao destino executando a simulação (servidor).

A primeira aplicação deste módulo é acoplar os dados recebidos a embarcações com Sistemas de Posicionamento Dinâmico, permitindo realizar testes de controladores em tempo real.

#### 4.1.1. Joystick

Os dispositivos de entrada do tipo joystick são divididos em eixos e botões. Tais elementos são interpretados e digitalizados para o computador. No caso de eixos, cada controle tem a sua posição discretizada em forma de inteiros negativos ou positivos. Neste caso, o zero representa a posição central. Já os botões possuem apenas dois estados: pressionado e não pressionado.

A leitura do dispositivo é realizada em software através do framework SDL (*Simple Direct Layer*), bastante comum em desenvolvimento de jogos para computador. Este framework facilita o processo de leitura do dispositivo. As posições do joystick são lidas e escalonadas segundo o seguinte padrão:

Tabela 1: Tabela de dados padrão TPN

Eixos	-256 à 256
Botões	1 ou 0

Devido a variações oscilações decorrentes do mau condicionamento do periférico, foi criada uma interface gráfica, batizada de Joystick Virtual, que permite definir as variáveis do joystick de forma precisa. O software permite também a configuração imediata do endereço de rede IP e porta do servidor executando o TPN, permitindo mudanças de servidor de forma ágil.

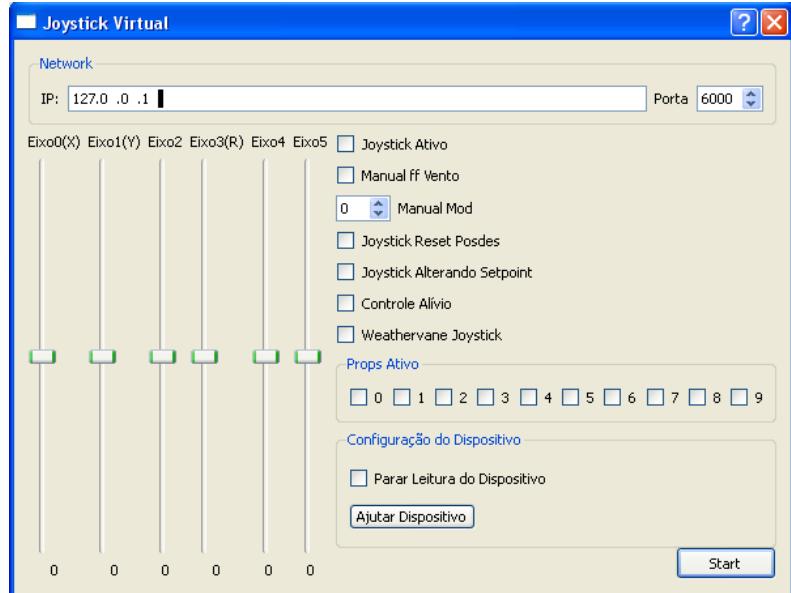


Figura 7: Tela do aplicativo Joystick Virtual

#### **4.1.2. Simulador de Sistemas de Posicionamento Dinâmico acoplado à entrada de dados**

Os dados enviados pelo cliente do joystick são inseridos no módulo de controle do TPN, realizando modificações somente as partes relevantes à leitura do dispositivo sem alterar o equacionamento utilizado. Assim, o código que realizava a leitura do antigo arquivo foi substituído por um pequeno módulo de servidor integrado, que recebe os dados do joystick e repassa aos métodos, alocando-os de acordo com o algoritmo definido.

O módulo de controle apresenta dois modos de operação: Manual e semi-automático.

##### **4.1.2.1. Modo manual**

No controle manual o operador comanda diretamente as forças que agem no sistema, sendo, responsável por determinar as forças que contrabalançarão as atuantes ambientais e controlarão o sistema na posição ou trajetória desejada. Nesta situação o operador utiliza apenas os módulos de alocação e atuação de empuxos, não necessitando, portanto, de algoritmos de controle ou filtragem.

Toda interface operador/ controlador é realizada pelo módulo de comando. A lógica básica de conversão dos comandos do operador em forças e momentos incorporados pelo SPD segue as equações descritas abaixo:

$$F_{SPD} = \begin{cases} P_{Joystick} \cdot F_{max_{SPD}}^+ & \text{se } P_{Joystick} \geq 0 \\ |P_{Joystick}| \cdot F_{max_{SPD}}^- & \text{se } P_{Joystick} \leq 0 \end{cases} \quad (12)$$

Onde  $F_{SPD}$  são as forças e momentos totais considerados pelo sistema de alocação e atuação do SPD,  $P_{Joystick}$  é a configuração ou posicionamento instantâneo do joystick principal, variando entre -1 e 1 para os três graus de liberdade controlados, e  $F^{+/-}_{max_{SPD}}$  são as forças e momentos máximos que o SPD pode fornecer na configuração simulada, nos dois sentidos.

Vale lembrar que esta equação pode envolver valores não simétricos para um mesmo grau de liberdade, acarretando em magnitudes de forças positivas diferentes de forças negativas.

Durante a execução de um controle manual, o controlador autônomo é interrompido, sendo retomado ao fim da intervenção do operador. Ao fim da manobra manual o operador tem a opção de, ao religar o controle autônomo, retornar à posição de referência inicial ou redefinir o ponto de referência como sendo qualquer uma das posições instantâneas ocupadas durante a manobra.

#### **4.1.2.2. Modo Semi-Automático**

O controle semi-automático se caracteriza pela atuação conjunta do operador e do controle autônomo. Este último fica responsável pela pré-estabilização do navio, contrabalançando as forças ambientais e garantindo uma intervenção mais segura e confortável para o operador. Este tem a liberdade de atuar no sistema, utilizando a potência remanescente disponível do SPD, realizando mudanças de posição ou alterações na trajetória base pré-definida. Neste tipo de controle o operador utiliza todas as potencialidades do módulo de comandos desenvolvido, ampliando as opções de intervenção e/ou adaptação do controle autônomo à situação exigida pelo operador. A lógica básica da interface de forças e momentos do operador/SPD é mostrada a seguir, na forma de diagrama de blocos:

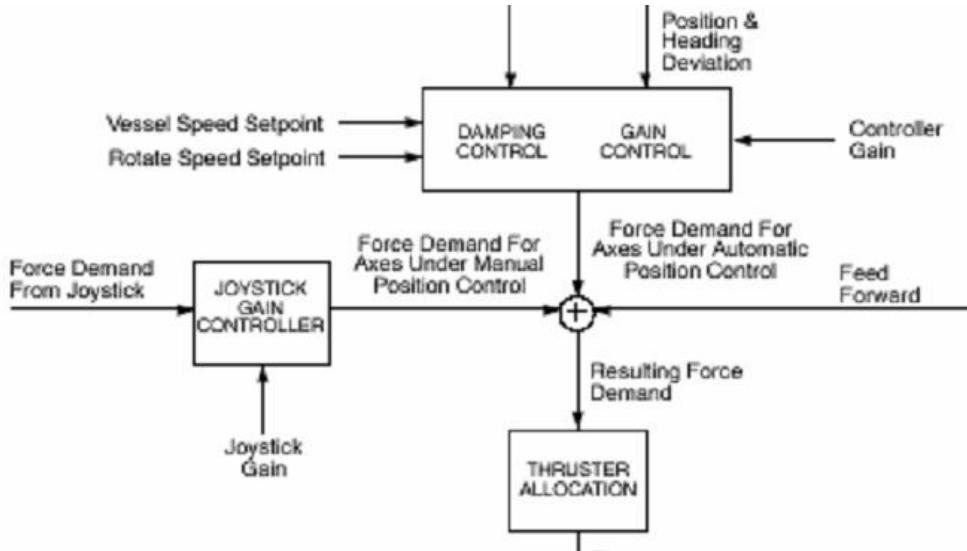


Figura 8: Diagrama de blocos do sistema de Posicionamento Dinâmico com Joystick

As equações que definem as forças e momentos produzidos pelo SPD em modo automático são dadas por:

$$F_{SPD} = \begin{cases} F_{SPD\,automático} + P_{Joystick} \cdot (F_{max_{SPD}}^+ - F_{SPD\,automático}) & \text{se } P_{Joystick} \geq 0 \\ F_{SPD\,automático} - |P_{Joystick}| \cdot (F_{max_{SPD}}^- - F_{SPD\,automático}) & \text{se } P_{Joystick} \leq 0 \end{cases} \quad (13)$$

Onde  $F_{SPD}$  são as forças e momentos totais considerados pelo sistema de alocação e atuação do SPD,  $F_{SPD\,automático}$  são as forças e momentos calculados pelo controle automático e já saturados pela potência máxima disponível,  $P_{Joystick}$  é a configuração ou posicionamento instantâneo do joystick principal, variando entre -1 e 1 para os três graus de liberdade controlados, e  $F^{+/-}_{max_{SPD}}$  são as forças e momentos máximos que o SPD pode fornecer na simulação, nos dois sentidos.

Assim, se a força calculada pelo controle automático está abaixo da potência total disponível, a potência remanescente pode ser redirecionada para a atuação do operador. Em contrapartida, caso as condições ambientais exijam a potência máxima do SPD, as mudanças solicitadas pelo operador não poderão ser atendidas pelo SPD já saturado. Os cinco modos de operação disponíveis para o usuário são combinações destes dois modos de controle.

Tabela 2: Modos do Joystick ativo

Modo	Descrição
0	Neste modo os movimentos de surge, sway e yaw estão em controle de malha aberta; ou seja, o operador do sistema está controlando manualmente todos os eixos, independente das condições ambientais. Os valores inseridos pelo joystick são multiplicados pelo ganho do sistema e convertido em forças de atuação direta no sistema.
1	Neste modo o movimento yaw está em modo automático; O controle sobre este grau de liberdade é passado ao controle autônomo do navio, manipulando automaticamente os propulsores com a finalidade de manter um determinado aprofundamento. Os outros eixos do navio estão em malha aberta, sendo controlados diretamente pelo operador.
2	Neste modo os movimentos de <i>surge</i> e <i>yaw</i> estão em modo automático; O controle sobre estes graus de liberdade é passado ao controle autônomo do navio, manipulando automaticamente os propulsores com a finalidade de manter estes graus de liberdade dentro do set point estabelecido. O movimento de <i>sway</i> do navio está em malha aberta, sendo controlado diretamente pelo operador.
3	Neste modo os movimentos de <i>sway</i> e <i>yaw</i> estão em modo automático; O controle sobre estes graus de liberdade é passado ao controle autônomo do navio, manipulando automaticamente os propulsores com a finalidade de manter estes graus de liberdade dentro do set point estabelecido. O movimento de <i>surge</i> do navio está em malha aberta, sendo controlados diretamente pelo operador.
4	Neste modo os movimentos de <i>surge</i> e <i>sway</i> estão em modo automático; O controle sobre estes graus de liberdade é passado ao controle autônomo do navio, manipulando automaticamente os propulsores com a finalidade de manter estes graus de liberdade dentro do set point estabelecido. O movimento de <i>yaw</i> do navio está em malha aberta, sendo controlados diretamente pelo operador.

#### 4.1.2.3. Parâmetros do controle

A Tabela 3 descreve cada parâmetro disponível ao usuário pelo dispositivo de joystick:

Tabela 4: Parâmetros do módulo de controle

manual_ff_vento	Com este modo ligado, o <i>feedforward</i> do vento é aplicado nos modos manuais, auxiliando na estabilização da embarcação quando o vento está presente.
reset_posdes	Reset Posição de referência Esta chave, presente no joystick, atualiza a posição de referência da embarcação para a posição atual.
joystick_alterando_setpoint	Modo atualização de <i>setpoint</i> Neste modo a embarcação está inteiramente sob controle de malha fechada: os valores inseridos pro joystick são utilizados como valor de referência da malha fechada. A diferença entre o valor de atual e o valor de referência é convertida em forças nos propulsores do SPD, movimentando o barco de forma segura e confortável para o controlador.
controle_alivio	Modo controle do aliviador Neste modo de operação o sistema posiciona o navio controlado com SPD perpendicularmente ao corpo de referência, a uma distância de 20 metros, independente da posição atual do aliviador. Vide <b>Erro! Fonte e referência não encontrada..</b>
weathervane_joystick	Modo <i>Weathervane</i> Neste modo o aliviador se aproa de forma a minimizar as forças incidentes do vento
Propulsores ativo	Este comando determina o número de propulsores disponíveis para o controle

## 4.2. Módulo de comunicação

O módulo de comunicação apresenta as seguintes especificações:

- Providenciar um canal de comunicação entre o TPN e o TPNView, de forma a suprir o visualizador com todos os dados necessários para a criação da maquete 3D.
- O fluxo de dados deve ser constante e com perdas mínimas, atualizando a maquete sem quebras de imagem.
- O volume de dados enviados não deve saturar a rede.
- O intervalo entre cada passo da simulação deve ser próximo ao tempo equivalente real. Em outras palavras deve haver um controle de tempo entre cada passo.

Para atender às especificações foi adicionada uma classe, batizada de TPNClient, encarregada do gerenciamento de pacotes, serialização, controle de tempo e envio via rede.

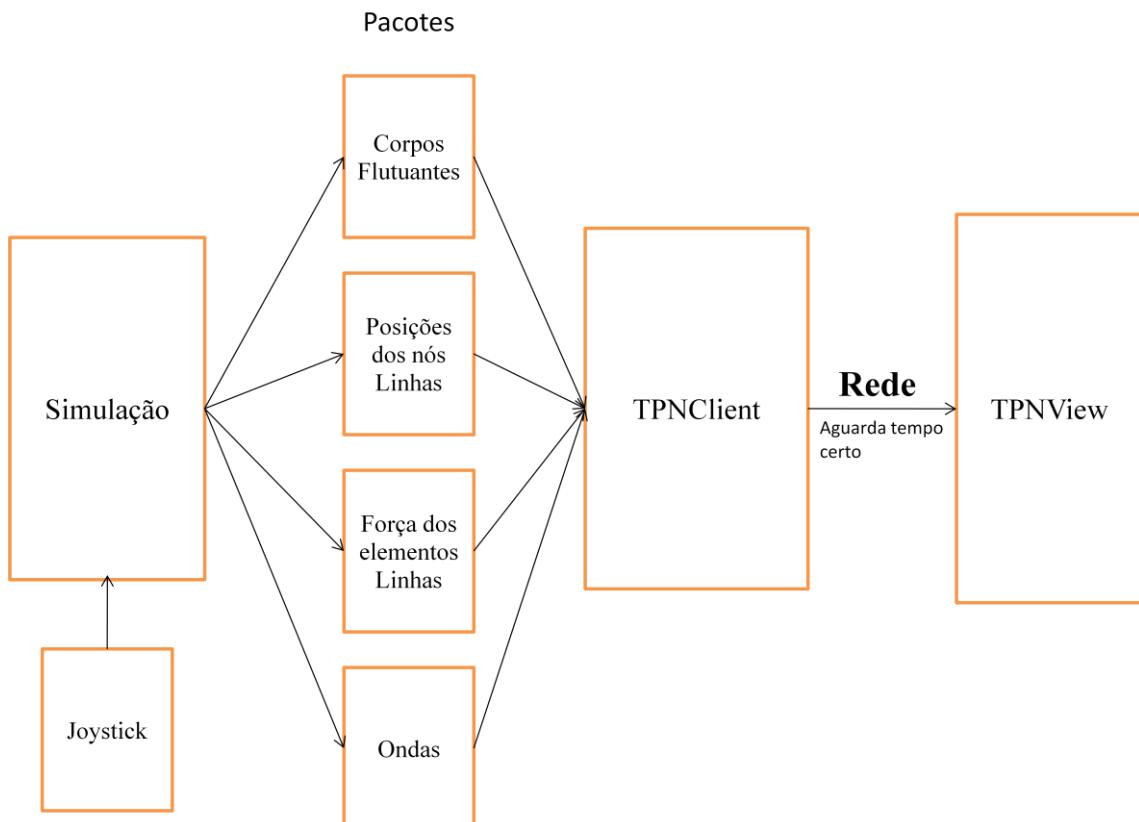


Figura 9: Diagrama de bloco do sistema do módulo de comunicação

#### 4.2.1. TPNClient

Para o gerenciamento dos dados a serem enviados foram criados filas separadas de acordo com tipo de pacote. Depois de efetuados todos os cálculos do instante atual, o TPNClient entra em modo de espera até que o tempo real seja igual ao passo de simulação, envia os dados e reiniciando o ciclo. Caso o tempo de cálculo seja superior ao passo de simulação o modo de simulação em tempo real se torna inviável

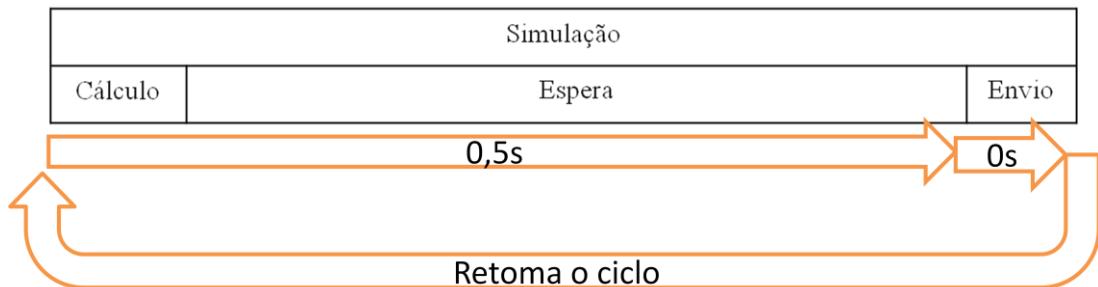


Figura 10: Esquema do controle de tempo para uma simulação de 0,5 segundos de passo.

#### 4.2.2. Transmissão de Corpos Flutuantes

Os valores de posição, velocidade e aceleração dos corpos flutuantes (definidos por três variáveis lineares e três variáveis angulares) de cada corpo são copiados das classes onde estão sendo calculados e enviados ao gerenciador de pacotes.

O pacote de corpos flutuantes é de tamanho fixo, composto por um cabeçalho de 8 bytes e um carga de 72 bytes.

Tabela 5 - Pacote para corpos flutuantes

Cabeçalho		Carga		
Tipo	Id	Posição	Velocidade	Aceleração

Sendo cada componente do carga representado por:

Tabela 6 - Detalhamento da carga de posições

Posição					
pos_x	pos_y	pos_z	pos_xx	pos_yy	pos_zz

Tabela 7 - Detalhamento da carga de velocidades

Velocidade					
vel_x	vel_y	vel_z	vel_xx	vel_yy	vel_zz

Tabela 8 - Detalhamento da carga de acelerações

Aceleração					
acel_x	acel_y	acel_z	acel_xx	acel_yy	acel_zz

Com x, y e z as posições nos eixos e xx, yy e zz as rotações nos eixos x, y e z respectivamente.

#### 4.2.3. Transmissão de posições de linha

Como as linhas são calculadas por métodos de elementos finitos, o volume de pontos calculados é grande, o que poderia saturar a rede e causar perda outros dados. Como solução foi proposto que o TPN enviasse apenas 1/5 das posições dos nós de cada linha, cabendo ao TPNView interpolar entre os pontos. O resultado desta simplificação foi visualmente satisfatório, provando-se aplicável ao caso.

Cada pacote contém um header de 12 bytes (tipo, id e tamanho) e cargas de 12 bytes cada, totalizando 12 vezes tamanho em bytes. Cada carga representa a coordenada de um nó da linha no espaço.

Tabela 9 - Pacote para posições dos nós da linha

Cabeçalho			Carga			
Tipo	id	tamanho	Posição	Posição	....	Posição

Sendo cada pacote da carga:

Tabela 10 - Pacote da carga para posição de cada nó

Posição		
pos_x	pos_y	pos_z

Sendo x, y e z a posição de cada nó.

#### 4.2.4. Transmissão de forças de linha

As forças de linha recebem um tratamento similar às posições; cada linha separa 1/5 dos elementos calculados e os envia ao TPNView realizar a interpolação. Novamente esta simplificação gerou bons resultados.

Cada pacote contém um header de 16 bytes (tipo, id, tamanho) e cargas de 12 bytes cada, totalizando 12 vezes tamanho bytes. Cada carga representa a coordenada de um nó da linha no espaço. Cada carga representa as forças nos eixos x y e z em cada elemento enviado - por elemento **n** entende-se como o elemento entre os nós **n** e **n+1**.

Tabela 11 - Pacote para posições das forças da linha

Cabeçalho			Carga			
tipo	id	tamanho	Forças	Forças	....	Forças

Sendo cada pacote do cargo:

Tabela 12: Pacote da carga para forças de cada elemento

Forças		
for_x	for_y	for_z

Com for\_x, for\_y e for\_z as forças nos eixos e m\_x, m\_y e m\_z os momentos em torno dos respectivos eixos.

#### 4.2.5. Mapa de Ondas

Um dos principais problemas do gerador de ondas existente no TPNView é a falta de sincronia entre os corpos simulados e as ondas incidentes, gerando situações contraditórias. Como solução foi definida que o TPN passaria a ser responsável pela geração e envio via rede do mapa, cabendo ao TPNView apenas o desenho das ondas.

O projeto inicial previa que o gerador de ondas calculasse as alturas de um mapa de 512 por 512 pontos. Este grande volume de pontos, multiplicado pelo alto custo computacional do método para o cálculo de apenas uma altura, demandou um tempo excessivo de cálculo por passo – aproximadamente 2 segundos, superior ao passo padrão de simulação de 0.5 segundo. Como alternativa foi realizada uma redução do numero de pontos calculados, passando a enviar um perfil de ondas unidimensional (512 pontos) mais com o ângulo de incidência, cabendo ao TPNView realizar a extrusão do perfil para a renderização do mapa.

Embora o mapa gerado não seja tão próximo do mar real, a solução simplificada permite a sincronização do mapa com os corpos, que não ocorria no gerador de ondas

existente no TPNView, e a possibilidade de execução em tempo real, tornando esta simplificação válida e adequada para os objetivos deste projeto.

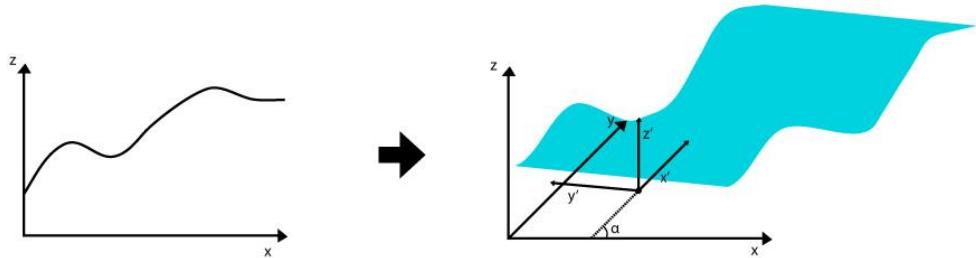


Figura 11: Esquema ilustrando a extrusão do perfil de onda, nota-se a rotação do eixo da onda de ângulo  $\alpha$  de incidência.

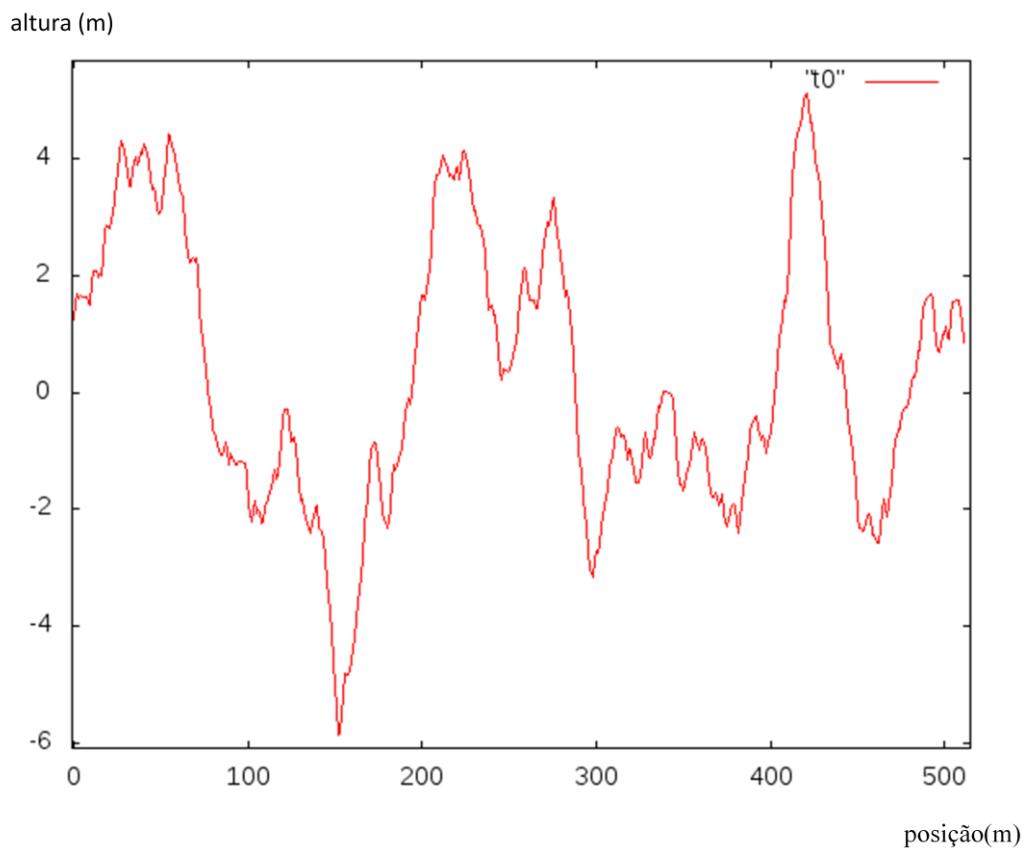


Figura 12: Perfil de ondas típico calculado

O pacote de envio de ondas do mar consiste em um cabeçalho fixo e uma carga de tamanho variável, contendo a distribuição de alturas do perfil no espaço. O tamanho do pacote é definido de acordo com o numero de pontos calculados pelo simulador, sendo *height* a altura da onda para cada ponto e *Angle* o ângulo de incidência da onda em relação ao eixo *z* do visualizador. A ordem de montagem da carga define a sua posição

no espaço, sendo o primeiro referente à origem do sistema global.

Tabela 13: Pacote de Ondas do mar

Cabeçalho				Carga			
tipo	Id	ângulo	tamanho	altura	altura	....	altura

### 4.3. Módulo de *Feedback*

As especificações deste módulo são:

- Transmitir as sensações de movimento da embarcação para o usuário.
- Adequar os dados aos limites mecânicos.
- Enviar os dados com rede de acordo com o protocolo do driver da plataforma.

Para este módulo foram realizadas adaptações no TPNView para estabelecer comunicação com a plataforma Stewart, localizada na sala de visualização do laboratório, de forma que os dados utilizados para o processamento do corpo flutuante sejam reproduzidos na plataforma, transmitindo as sensações de movimento.

Aliando o modo de projeção estereoscópica dos projetores da sala de visualização, um sistema de câmeras preso à embarcação e os movimentos da plataforma, obtém-se uma ambiente de testes em imersão altamente atraente.

#### 4.3.1. Plataforma Móvel

A plataforma instalada na sala de visualização é do tipo Hexapod com 6 graus de liberdade modelo 171-131 da MOOG FCS, com 6 atuadores lineares acionados por servomotores de alta performance. Os limites mecânicos seguem como a tabela abaixo:

Tabela 14: Tabela com os limites da plataforma

Eixo	Posição	Velocidade	Aceleração
<i>Roll</i>	$\pm 21^\circ$	$\pm 30^\circ/\text{s}$	$500^\circ/\text{s}^2$
<i>Pitch</i>	$\pm 22^\circ$	$\pm 30^\circ/\text{s}$	$500^\circ/\text{s}^2$
<i>Yaw</i>	$\pm 22,5^\circ$	$\pm 40^\circ/\text{s}$	$400^\circ/\text{s}^2$
<i>Surge</i>	- 0,24/ +0,25 metros	$\pm 0,5 \text{ m/s}$	$\pm 0,6 \text{ G}$
<i>Sway</i>	$\pm 0,25 \text{ metros}$	$\pm 0,5 \text{ m/s}$	$\pm 0,6 \text{ G}$
<i>Heave*</i>	- 0,35 metros	$\pm 0,3 \text{ m/s}$	- 0,5/ +0,7 G

\*O zero deste grau de liberdade tem o zero definido no ponto de maior altura, decrescendo para baixo.

O sistema eletrônico é composto por duas partes: o primeiro pelo controle de alimentação e drivers de controle e o segundo para computação e lógica (distribuição do movimento, controle dos modos). Um monitor conectado ao sistema eletrônico exibe o status dos atuadores e possíveis mensagens de aviso. Toda a comunicação com a plataforma é realizada por serial RS-485.

Como os dados recebidos pelo TPNView estão em unidades reais (metros para graus de liberdade lineares, radianos para angulares), torna-se necessário realizar uma filtragem dos valores antes do envio, particularmente dos lineares. Este filtro deve transformar os dados transmitidos em movimentos da plataforma, reproduzindo as sensações aos passageiros na embarcação simulada. Estes movimentos deverão respeitar os limites físicos da plataforma, sem esbarrar nos marcadores de fim de curso da plataforma.

No escopo deste projeto somente os graus de liberdade *roll*, *pitch* e *heave* serão passados. Os dois primeiros apresentam pequenos valores, podendo ser passados diretamente. O *heave* deverá passar por uma filtragem adequada.

#### 4.3.1.1. Filtros

Em uma embarcação os movimentos mais sensíveis ao passageiro são *roll*, *pitch* e *heave*. Os movimentos de rotação (*roll* e *pitch*) apresentam pequenas variações e, portanto, podem ser passados diretamente à plataforma. Já o movimento de *heave* apresenta grandes amplitudes e velocidades variáveis, sendo necessária aplicação de um filtro nos dados antes do envio à plataforma.

Inicialmente foram propostas quatro soluções de filtros:

1. Redimensionamento simples
2. Filtro de acelerações com Transformada de Fourier Inversa
3. Filtro limitador de velocidade
4. Redimensionamento com logaritmo

Para execução de testes foi utilizado uma série temporal como referência, calculada utilizando dados reais:

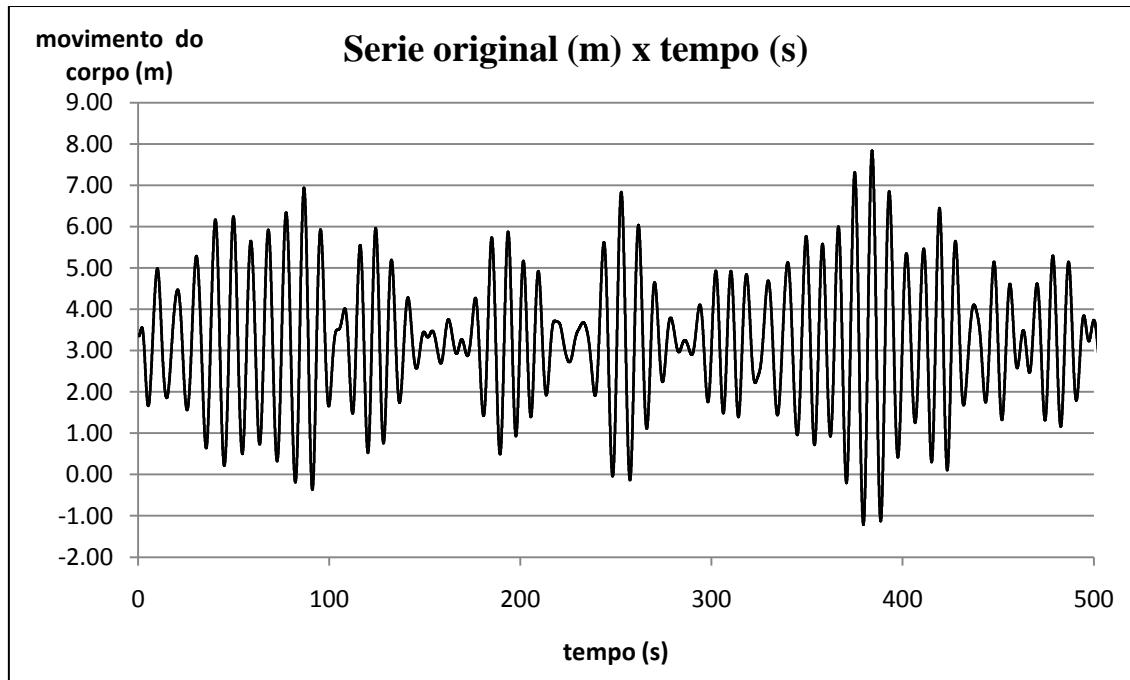


Figura 13: Série temporal utilizada para testes

##### 3.4.1.1.1 Redimensionamento simples

Neste método de filtragem os dados são multiplicados por um valor. O

multiplicador utilizado é definido de acordo com os valores extremos da série, de forma que em nenhum momento algum ponto ultrapasse o limite físico da plataforma. A série temporal resultante é idêntica a serie original, porém redimensionada para a plataforma.

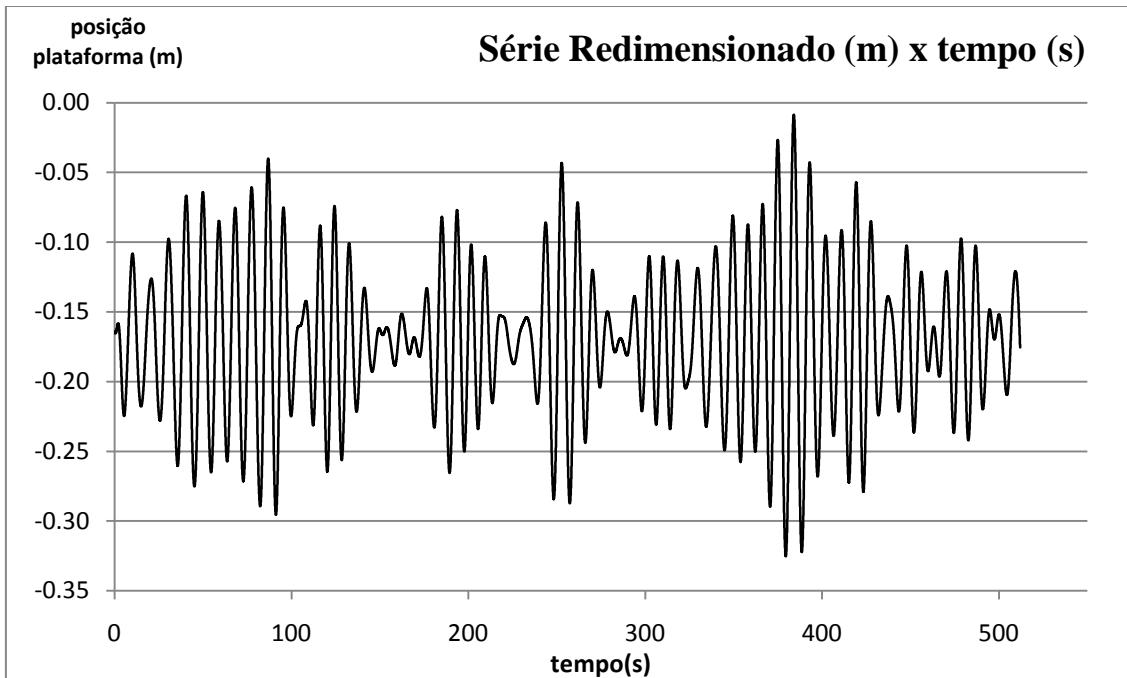


Figura 14: Série temporal com redimensionamento simples

Para séries com grandes variações de altura, a restrição dos limites da plataforma obriga um multiplicador mais atuante, prejudicando os movimentos de baixa amplitude e podendo torná-los até imperceptíveis.

#### 3.4.1.1.2 Filtro de acelerações

O ser humano apresenta um sistema não visual que permite saber a orientação e movimentos do corpo, denominado aparelho vestibular. Este sistema é um conjunto de órgãos do ouvido responsáveis pela sensação de movimento linear e angular.

Foi estabelecido que o limite de percepção de aceleração é definido entre  $0.002g$  e  $0.027g$  (equivalente a  $0.02m/s^2$  e  $0.27m/s^2$  ao nível do mar). Abaixo deste valor a movimentação é considerada imperceptível para o corpo (Benson, 1990).

Baseado neste princípio foi proposto um filtro, dividido em duas etapas:

1. Na primeira etapa, onde a aceleração é mais acentuada, o movimento deve

reproduzir os movimentos da série temporal.

2. Após a embarcação reduzir a aceleração a um valor limite (definido pelo usuário), a embarcação cessa o movimento, retornando à posição neutra lentamente sem que o usuário sinta, pronto para os próximos movimentos.

Desta forma os pontos de maior aceleração – mais sensíveis - são preservados sem ultrapassar os limites mecânicos, dado que o movimento não é inteiramente reproduzido pela plataforma.

O equacionamento do filtro segue o seguinte algoritmo:

$$\begin{aligned} z_i &= z_{i-1}, & \text{se } |\ddot{z}| < \ddot{z}_{min} \\ z_i &= z_i + k(\dot{z}_i \Delta t + \frac{\ddot{z}_i (\Delta t)^2}{2}), & \text{se } |\ddot{z}| > \ddot{z}_{min} \end{aligned} \quad (14)$$

Com  $z_i$  o passo atual,  $z_{i-1}$  o passo anterior,  $\dot{z}_i$  velocidade no passo,  $\ddot{z}_i$  aceleração do passo,  $k$  fator multiplicador para adaptação do movimento à plataforma e  $\Delta t$  intervalo de tempo entre cada passo. Aplicando o filtro na série temporal obteve-se o seguinte resultado:

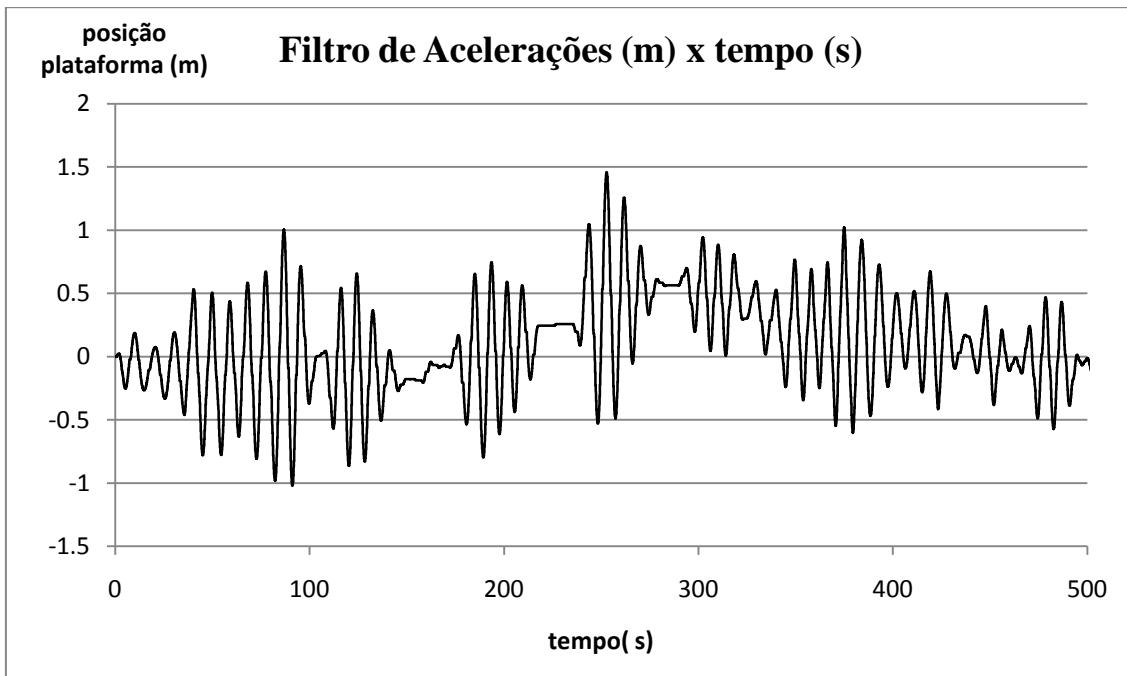


Figura 15: Série temporal com filtro de acelerações

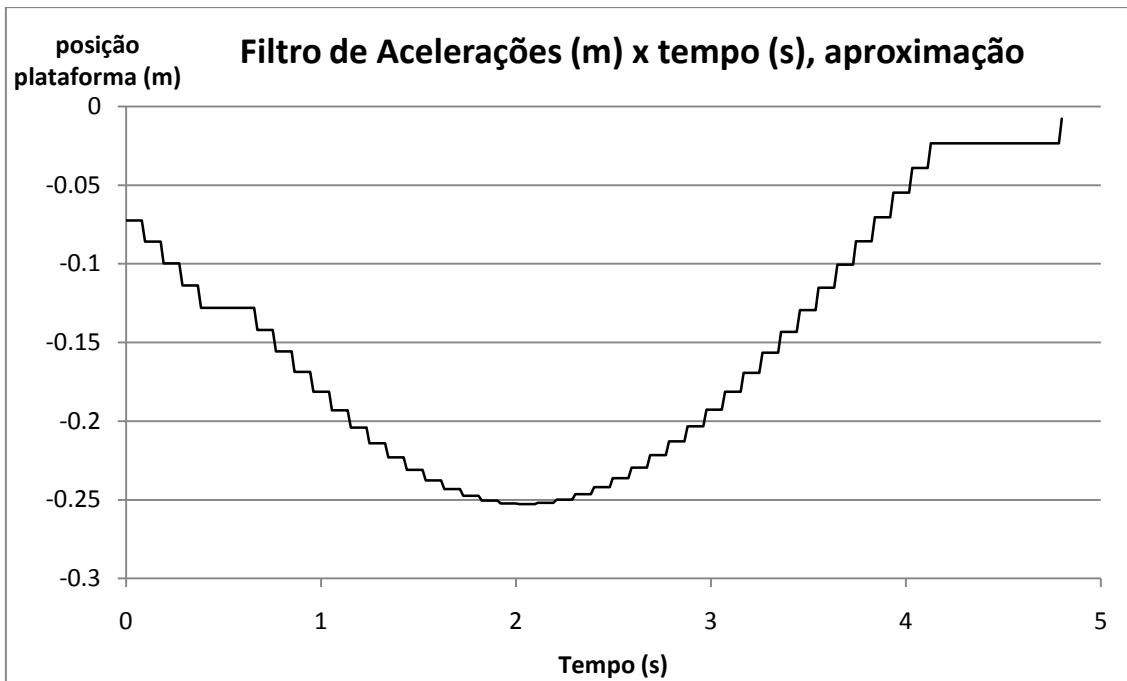


Figura 16: Detalhe da mesma série, evidenciando as descontinuidades de velocidade

Nota-se que há certos pontos de descontinuidade de velocidades, decorrentes do próprio algoritmo, gerando movimentos bruscos de parada indesejáveis.

Para amenizar estas descontinuidades foi proposto filtrar parte das freqüências do espectro de movimento, utilizando transformação discreta direta e inversa de Fourier:

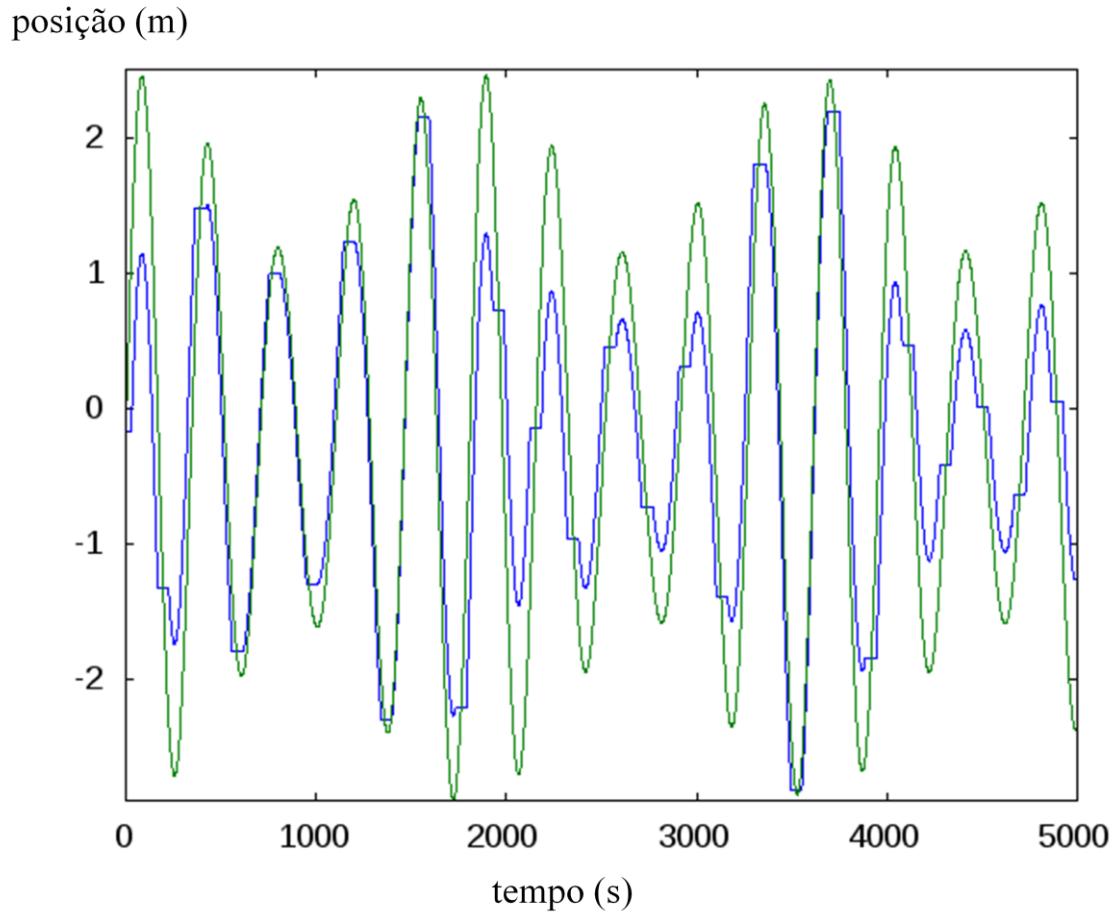


Figura 17: Série com filtro de acelerações e passando FFT, com filtro de acelerações apenas (azul) e com FFT(verde)

Ao final, conclui-se que este filtro é ineficiente para a aplicação, pois:

- Ocorrência de distorções severas em certos pontos, gerando uma série temporal inadequada à plataforma.
- Impróprio para grandes variações de aceleração, pois não é possível encontrar um valor limite que propicie uma sensação fiel aos dados de entrada e dentro das restrições da plataforma.
- Não pode ser aplicado em tempo real. Para cálculo do espectro é necessário o conhecimento de toda a série temporal.

#### 3.4.1.1.3 Filtro de Velocidades

Este filtro tem como principal objetivo reduzir a diferença entre os picos e vales

da série temporal. Dado que o movimento é composto por uma série harmônica, é possível predizer, a partir da velocidade, a altura do próximo ponto extremo (máximo ou mínimo). Ou seja, aplicando à velocidade durante a subida/descida pode-se reduzir o módulo da altura do próximo ponto extremo.

O filtro de velocidades segue o seguinte algoritmo:

$$\begin{aligned} z_i &= z_{i-1} + k(v\Delta t + \frac{\ddot{z}_i(\Delta t)^2}{2}), \quad \text{se } |v| < \dot{z}_l \\ z_i &= z_{i-1} + k(\dot{z}_l\Delta t + \frac{\ddot{z}_i(\Delta t)^2}{2}), \quad \text{se } |v| > \dot{z}_l \end{aligned} \quad (15)$$

Com  $z_i$  o passo atual,  $z_{i-1}$  o passo anterior,  $\dot{z}_l$  velocidade no passo,  $\ddot{z}_i$  aceleração do passo,  $v$  a velocidade limite e  $\Delta t$  intervalo de tempo entre cada passo. Caso a velocidade seja maior que o valor limite, indicando possível pico de altura á frente, a restrição é aplicada.

No entanto, esta abordagem se mostrou pouco prática, pois:

Como o movimento é irregular, as velocidades de subida e descida entre um ponto extremo não são exatamente iguais, gerando pequenos erros de posicionamento. Estes erros vão se sobrepondo no decorrer da série temporal, fazendo-a oscilar fora do seu zero. Isto pode ser notado claramente no gráfico abaixo, com a aplicação do filtro na série de referência:

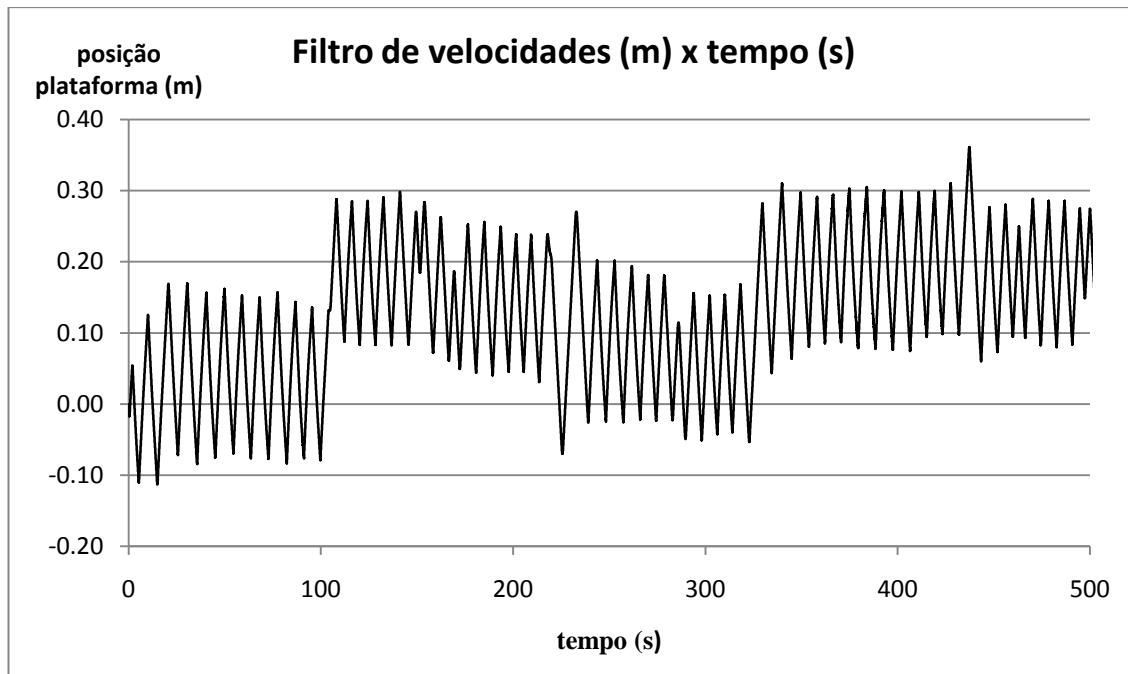


Figura 18 :Série temporal com Filtro de velocidades

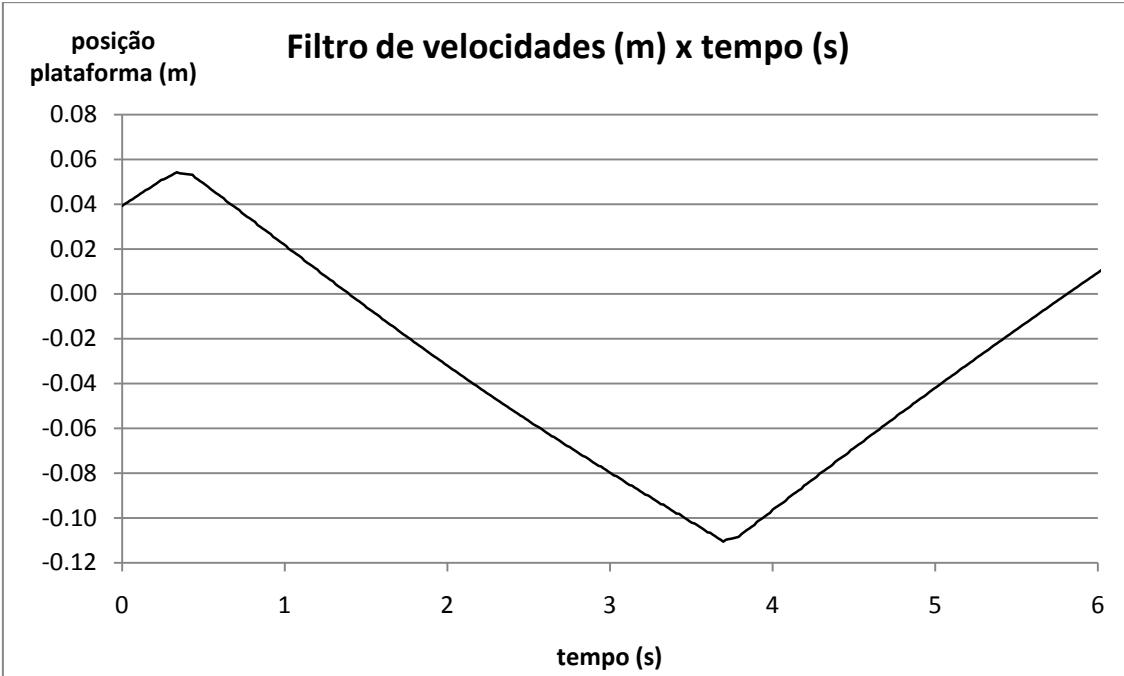


Figura 19: Detalhe da série temporal com o filtro

#### 3.4.1.1.4 Redimensionamento com logaritmo

Este filtro é uma versão modificada do escalonamento simples. O operador logaritmo tem como propriedade de reduzir a diferença entre valores altos e baixos. Utilizando este operador dos valores podemos “achatar” os pontos de altura mais acentuada sem comprometer os pontos de baixas alturas. No entanto para valores tendendo a zero, o resultado tende ao infinito, resultando em descontinuidades indesejáveis.

Como solução do problema foi proposto o seguinte algoritmo:

$$\begin{aligned}
 z_i &= z_{i-1} + v\Delta t + \frac{\ddot{z}_i(\Delta t)^2}{2}, \quad \text{se } |z_i| < 1 \\
 z_i &= 1 + \ln \left( z_{i-1} + \dot{z}_i \Delta t + \frac{\ddot{z}_i(\Delta t)^2}{2} \right), \quad \text{se } z_i > 1 \\
 z_i &= -1 - \ln \left( \left| z_{i-1} + \dot{z}_i \Delta t + \frac{\ddot{z}_i(\Delta t)^2}{2} \right| \right), \quad \text{se } z_i < -1
 \end{aligned} \tag{16}$$

Após a filtragem o valor é escalonado para dentro do intervalo da plataforma. Como a diferença entre os valores extremos se torna menor devido à aplicação de logaritmo, o multiplicador é menos requisitado, prejudicando menos os movimentos de pequenas amplitudes.

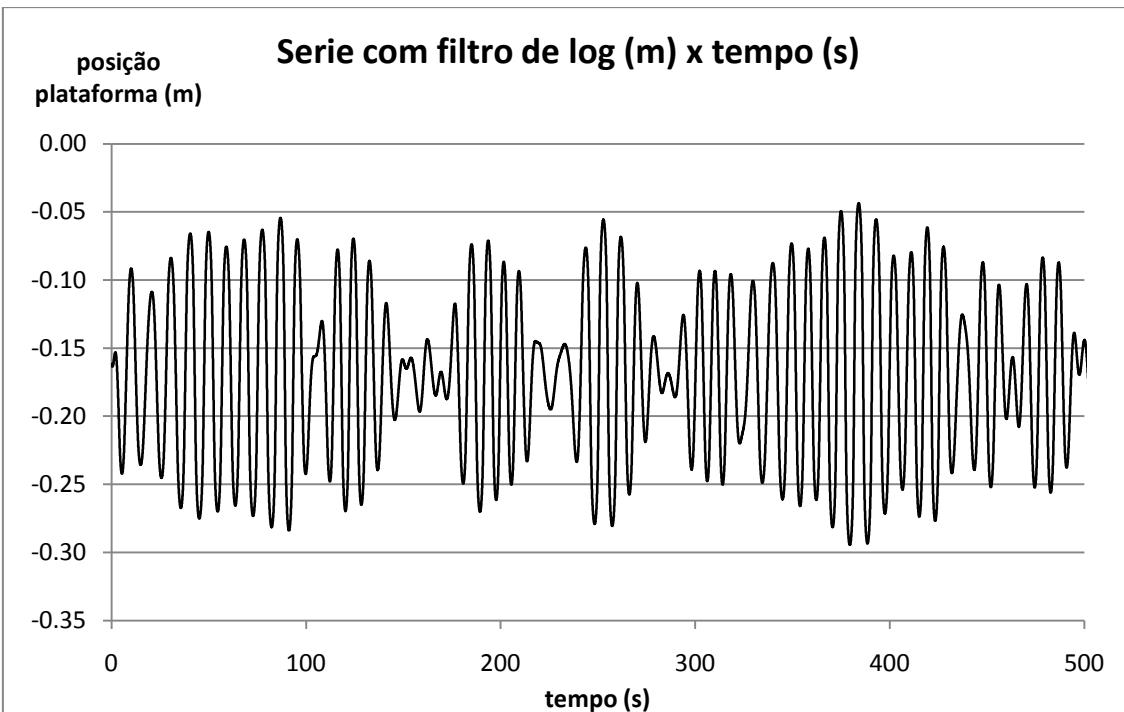


Figura 20: Filtro com logaritmo

Este algoritmo gerou bons resultados, mantendo as pequenas oscilações e diminuindo. No entanto o movimento fica mais suave nas pontas, diminuindo a sensação de movimento. Este método pode ser aplicado em simulação em tempo real, porém, tal como o método de redimensionamento simples, é necessário o conhecimento prévio dos movimentos possíveis para definição do multiplicador.

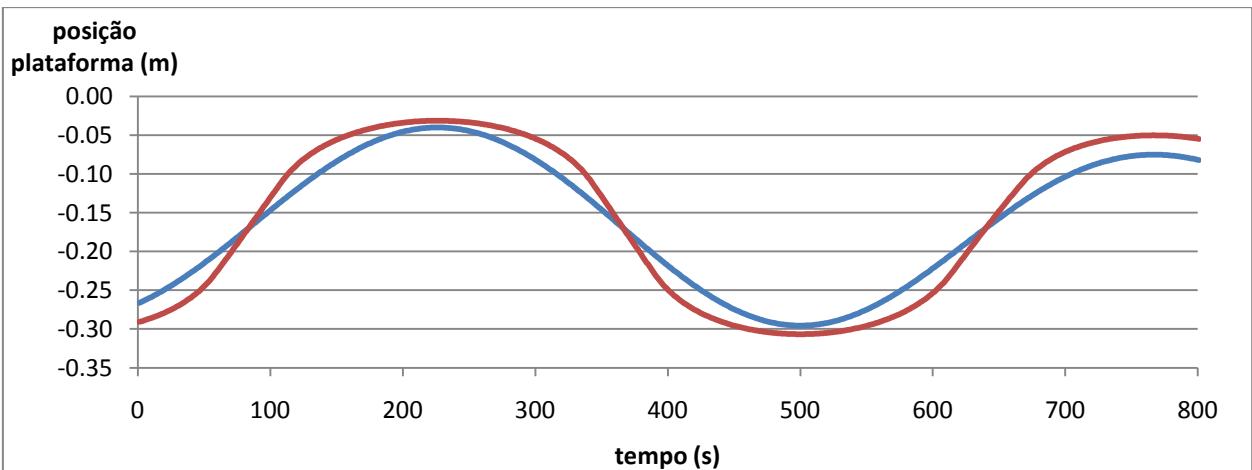


Figura 21: Detalhe do arredondamento das pontas para o filtro de logaritmo (vermelho) e redimensionamento simples (azul)

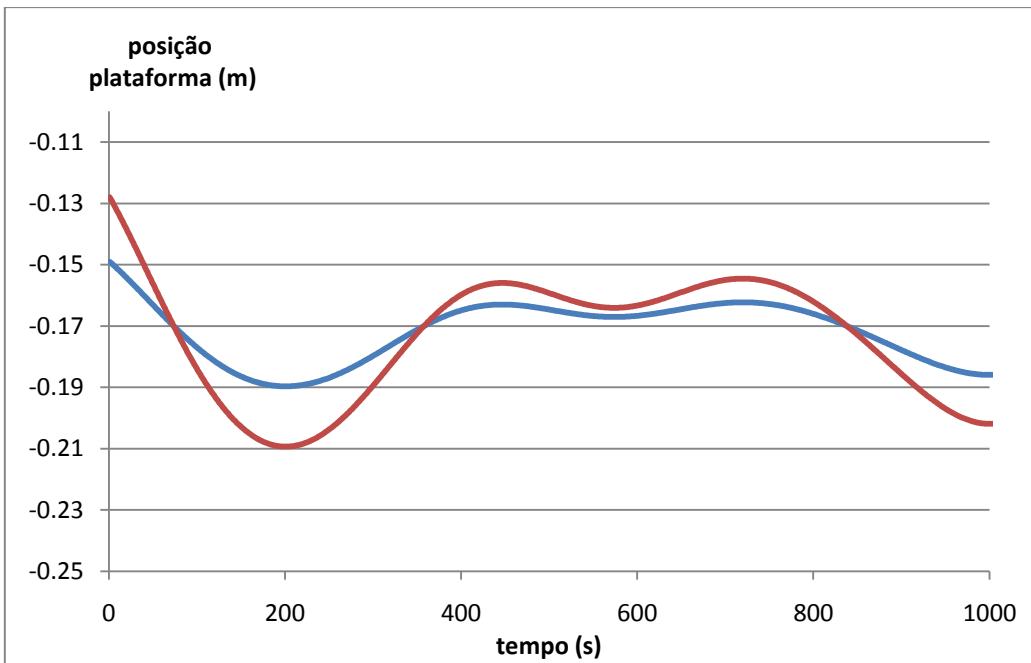


Figura 22: Comparaçāo de pequenos movimentos entre redimensionamento simples (azul) e logaritmo(vermelho)

#### 4.3.1.2. Transmissão de dados

Os pacotes da plataforma são fixos como uma seqüência de oito palavras de 32 bits. No modo controle dos graus de liberdade, o pacote segue o seguinte padrão:

Tabela 15: Pacote enviado à plataforma

Palavra	Dado	Descrição	Unidade
0	MCW	Palavra de comando	-
1	roll	Roll da plataforma	radianos
2	pitch	Pitch da plataforma	radianos
3	position_z	Heave da plataforma	metros
4	position_x	Surge da plataforma	metros
5	yaw	Yaw da plataforma	radianos
6	position_y	Sway da plataforma	metros
7	-	Vazio	-

Com palavras de comando:

Tabela 16: Palavras de comando disponíveis para à plataforma

Comando	Hexadecimal	Palavra			
Desabilitar plataforma ( <i>Disable</i> )	DC	00000000	0000000	00000000	11011000
Ir para posição inicial ( <i>Park</i> )	D2	00000000	0000000	00000000	11010010
Iniciar plataforma ( <i>Engage</i> )	B4	00000000	0000000	00000000	10110100
Iniciar simulação ( <i>Start</i> )	AF	00000000	0000000	00000000	00000000
Modo de controle de graus de liberdade ( <i>DOF</i> )	AA	00000000	0000000	00000000	10101010

Para o envio dos comandos de configuração o pacote deverá conter o comando na primeira palavra e tamanho de 256 bits, equivalente a 8 vezes 32 bits. Por definição do software da plataforma, cada palavra do pacote deve estar em formato *little-endian* (ultimo algarismo significante primeiro).

A frequência de trabalho de envio de dados para a plataforma deve ser exatamente 60 Hz (envio a cada 0,0166 segundos), caso contrário a plataforma começa a trepidar podendo até perder conexão por *timeout*. Para que a série temporal seja enviada no tempo certo, foi criado um timer que utiliza o próprio clock do processador para cálculo do tempo. A comunicação entre o TPNView e a unidade de computação da plataforma é feita por socket UDP.

## 5. Conclusões e proposta de pesquisas futuras

### 5.1. Conclusões

Os módulo propostos neste trabalho foram, de maneira geral, implementados com sucesso.

O módulo de entrada de dados operou dentro das expectativas, os sistemas de comunicação em rede atenderam as especificações previstas no projeto e a solução proposta para filtro da plataforma do módulo de *feedback* mostrou-se ideal para a especificação exigida.

Como o acoplamento do TPNView com o módulo de *feedback* exige um estudo

extenso sobre seu funcionamento, foi dada preferência ao desenvolvimento do filtro. Por hora os dados enviados à plataforma são provenientes de um arquivo com uma série temporal calculada a partir de dados reais.

Este trabalho foi o primeiro passo no desenvolvimento de um ambiente de simulação em tempo real para o TPN. Simulações com os módulos em operação foram bem recebidas, sinalizando interesse em dar continuidade a este trabalho.

## 5.2. Proposta de pesquisas futuras

Como continuidade deste trabalho espera-se o desenvolvimento dos seguintes tópicos:

- Otimização do tamanho dos pacotes enviados, em especial para o caso das forças e posições de linhas, aplicando técnicas de compactação de dados.
- Expansão do módulo de entrada de dados para outras aplicações além do controle de SPD.
- Implementação de um método de cálculo mapa de ondas com espalhamento para aplicação em tempo real, na forma de paralelização dos cálculos ou algum método similar que permita calcular toda a malha em tempo real.
- Tratamento dos dados recebidos no TPNView para geração de movimento da plataforma, pois os valores enviados pelo TPN não estão discretizados o suficiente para utilização direta na plataforma, necessitando de um método único para esta aplicação.
- Incorporação do módulo de *feedback* no TPNView, na forma de um processo independente para garantir o envio de pacotes no tempo certo. Caso contrário a plataforma pode trepidar e até encerrar conexão com o TPNView. Esta implementação deve considerar o item anterior, incluindo em suas especificações uma estratégia de comunicação entre este processo e o fluxo de dados do TPNView.

## **6. Bibliografia**

BRAY, D. **Dynamic Positioning**, The Oilfield Seamanship Series, Volume 9, Oilfield Publications Ltd. (OPL), 1998.

FOSSEN, T. I.: **Guidance and Control of Ocean Vehicles**, John Wiley and Sons, Ltd., 1994.

FUCATU, H. C.: **Desenvolvimento de um Simulador para Análise de Navios Amarrados**, São Paulo, 1998. Lv. Tese (mestrado) – Escola Politécnica, Universidade de São Paulo.

FUCATU, C. H.; NISHIMOTO, K.; MASETTI, I.: **Dynasim – a time domain Simulator of anchored FPSO**, 20<sup>th</sup> Int. Conf. on Offshore Mechanics and Arctic Engineering (OMAE), 2001.

MUCKLE, W: **Rev. ed. Of Naval architecture for marine engineers**, Butterworths, 2<sup>nd</sup> Edition, 1975.

HOLZNER, S.: **C++ Black Book**, 1<sup>a</sup> Edição, MAKRON BOOKS, 2001.

MEYERS, S.: **Effective C++: 55 Specific Ways to Improve Your Programs and Designs**, Addison-Wesley Professional, 3rd Edition, 2005.

MEYERS, S.: **Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library**, Addison-Wesley Professional, 2001.

NISHIMOTO, K.: **Development of numerical offshore tank for ultra deep water oil production systems with multi-bodies**, Int. Symp. On Technology of Ultra Deep Ocean Engineering, Tokyo, 2005.

NISHIMOTO, K.; DONATO, M.; MASETI, I. Q.; JACOB, B. P.; MARTINS, M.; MENEZES, I.; HIRATA, K.: **Development of numerical offshore tank for ultra deep water oil production systems**, 21<sup>st</sup> Int. Conf. on Offshore Mechanics and Arctic Engineering (OMAE), 2003.

TANNURI, E. T.: **Sistemas de Posicionamento Dinâmico: Projeto, Análise e Novos Desenvolvimentos**. São Paulo, 2009. Lv. Tese (livre docência) – Escola Politécnica, Universidade de São Paulo.

BENSON, A.J.: **Sensory Functions and Limitations of the Vestibular Systems**.

Retirado de R. WARREN and A.H. WERTHEIM, *Perception and Control of Self Motion*. Lawrence Erlbaum Assoc.: Hillsdale, NJ, 1990.

**PETERS, R.A.: Dynamics of the Vestibular System and Their Relation to Motion Perception, Spatial Disorientation, and Illusions.** NASA CR-1309, NASA, 1969