

Integrando um simulador de manobras de navios, de alta fidelidade, com ROS: estudo de caso de um seguidor de caminho

Bruno Scaglione

Pedro Marzagão

Orientador: Prof. Dr. Eduardo Aoun Tannuri

Escola Politécnica da Universidade de São Paulo (POLI-USP)

TCC de Engenharia Mecatrônica, 26 de dezembro de 2021

Sumário

1 Introdução

2 Métodos

- Arquitetura
- API HTTP
- Modelo dinâmico
- Controle
- Guiança: *Line-of-Sight*
- Filtros

3 Resultados

4 Conclusão

Motivação

- Projeto seguro e eficiente de canais de navegação;
- fator principal: perfil de largura dos canais;
- análise da distribuição de trajetórias.



Figura: Animação do navio *EverGiven* encalhado no canal de suez [1]

Escopo

- Simulador de manobras de alta fidelidade do TPN (Tanque de Provas Numérico): *pydyna*;
- simulações *real-time* e *fast-time*;
- problema: infraestrutura legado das simulações *fast-time*.

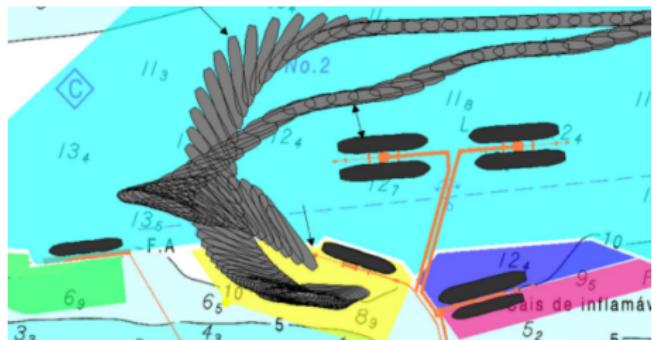


Figura: Exemplo de uma simulação *fast-time* [2].

Objetivos

- Ideia principal: mudar a infraestrutura das simulações *fast-time* para o ROS (Robot Operating System);
- focos:
 - ➊ Disponibilizar o *pydyna* no ambiente ROS;
 - ➋ Prova de conceito: aplicação representativa de uma simulação *fast-time*.

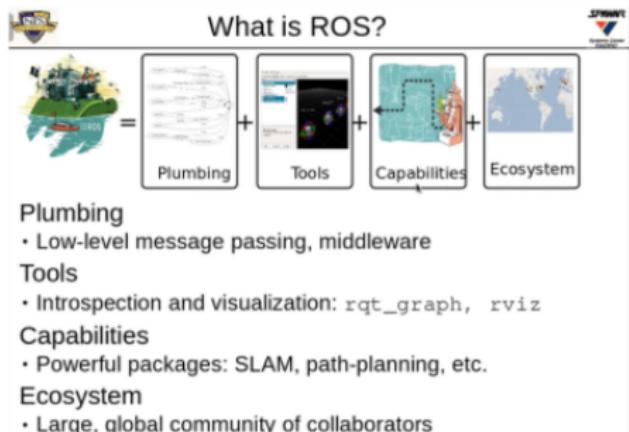


Figura: Resumo do ROS [3].

Tarefa

- Navio seguidor de caminho.
Caminho é definido por linhas que conectam os *waypoints*;
 - $\text{waypoint} = (x_{des}, y_{des}, u_{des})$ onde (x_{des}, y_{des}) é a localização desejada no plano 2D e u_{des} é a velocidade longitudinal desejada nessa localização;
 - objetivo é seguir as linhas com menor erro possível;
 - bônus: atingir as velocidades desejadas.

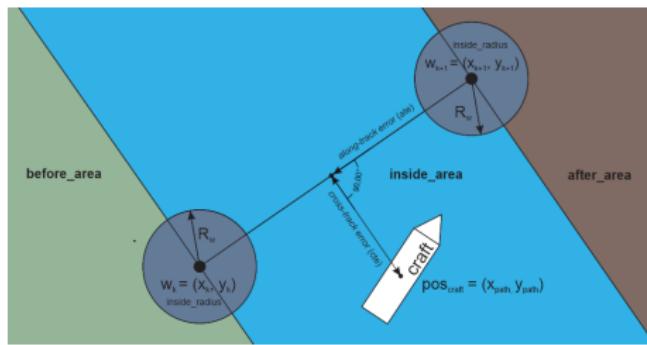


Figura: Diagrama representativo da tarefa.

Avaliação

Para que a tarefa seja bem sucedida, as duas condições abaixo precisam ser satisfeitas:

$$ate \leq R_{wFinal} \quad (1)$$

onde ate é o erro paralelo à linha atual do caminho, e R_{wFinal} é o raio de aceitação final.

$$cte_m < \frac{beam}{2} \quad (2)$$

Onde cte_m é o erro médio perpendicular à linha atual do caminho e $beam$ é a largura do navio no ponto mais largo.

Navio utilizado

O navio utilizado está representado abaixo:

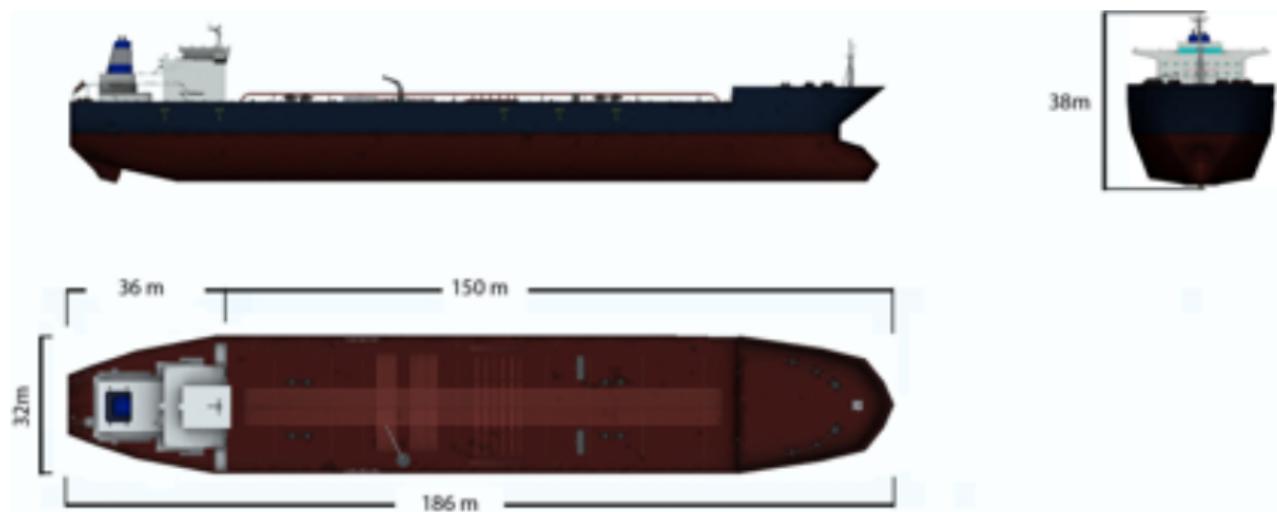


Figura: Vistas e parametrização do navio *Tanker55000DWT* [4].

Arquitetura

A arquitetura principal (de produção) é toda digital e está representada abaixo:

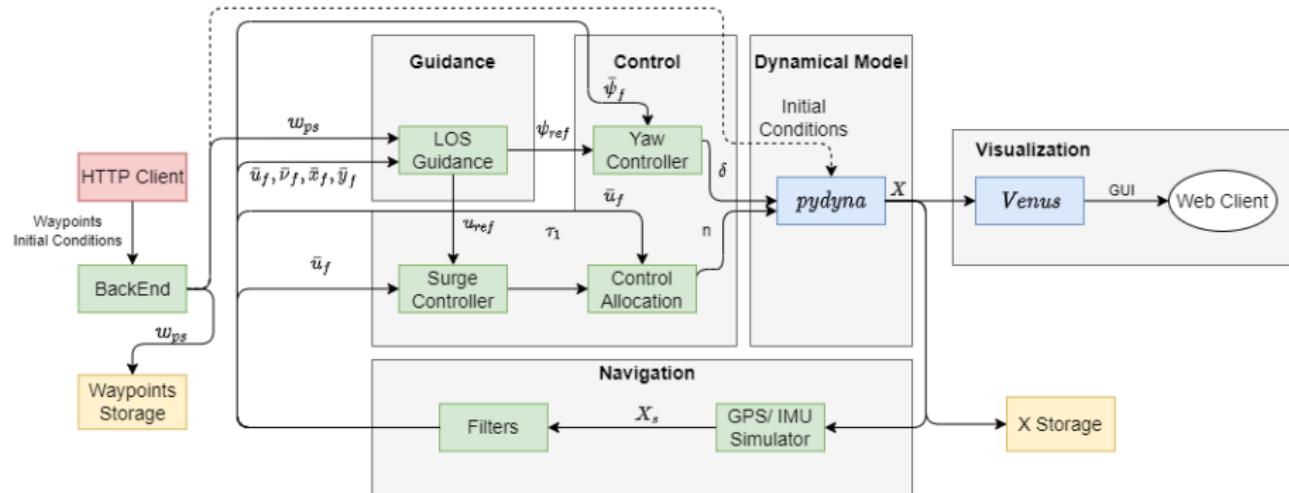
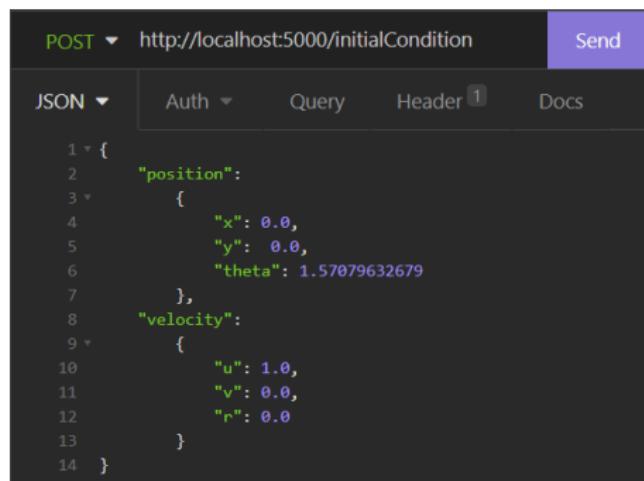


Figura: Arquitetura do ambiente de produção.

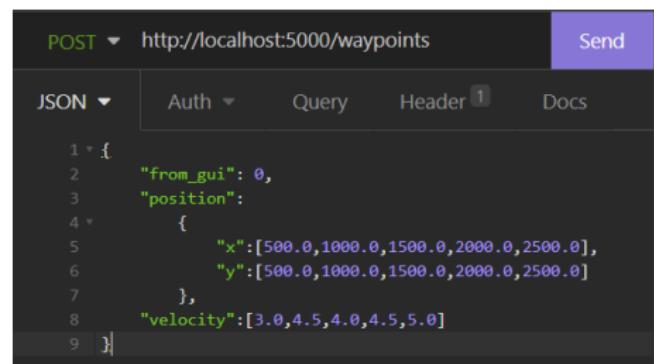
API HTTP

Figura: Exemplo de um *POST* request, passando a condição inicial do navio, através do cliente *Insomnia*



```
POST ▾ http://localhost:5000/initialCondition Send
JSON ▾ Auth ▾ Query Header 1 Docs
1 ▾ {
2   "position": {
3     "x": 0.0,
4     "y": 0.0,
5     "theta": 1.57079632679
6   },
7   "velocity": {
8     "u": 1.0,
9     "v": 0.0,
10    "r": 0.0
11  }
12 }
13 }
14 }
```

Figura: Exemplo de um *POST* request, passando os *waypoints*, através do cliente *Insomnia*



```
POST ▾ http://localhost:5000/waypoints Send
JSON ▾ Auth ▾ Query Header 1 Docs
1 ▾ {
2   "from_gui": 0,
3   "position": {
4     "x": [500.0,1000.0,1500.0,2000.0,2500.0],
5     "y": [500.0,1000.0,1500.0,2000.0,2500.0]
6   },
7   "velocity": [3.0,4.5,4.0,4.5,5.0]
8 }
9 }
```

Modelo dinâmico: simplificações adotadas

- Modelo do navio em 3-DOF: posições (x, y) e ângulo de guinada ψ ;
- navio com simetria longitudinal: desacopla o modelo de velocidade longitudinal do resto;
- corrente desprezível;
- forças hidrostáticas desprezíveis;
- forças de coriolis e centrípetas lineares em torno de u ;
- velocidade latitudinal v e angular r pequenas;
- navio operando com $|u| > 2 \text{ m/s}$.

Modelo dinâmico: 3-DOF

Um diagrama do modelo 3-DOF do navio:

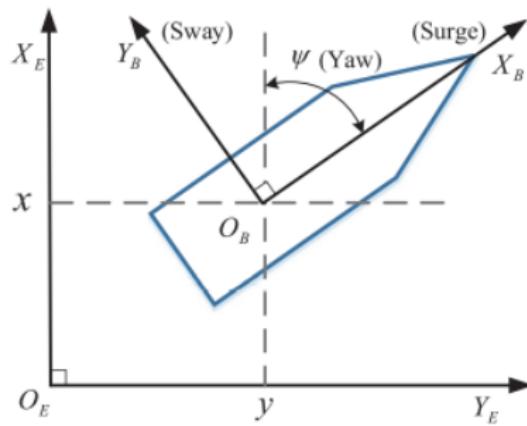


Figura: Diagrama do modelo 3-DOF do navio [5].

onde *sway* é a velocidade latitudinal e *surge* a velocidade longitudinal. As velocidades estão no referencial solidário ao navio, enquanto as posições no referencial fixo na terra.

Modelo dinâmico: ângulo de guinada

O modelo adotado é a modelo de primeira ordem de Nomoto, representado abaixo:

$$\frac{r}{\delta}(s) = \frac{K}{1 + Ts} \quad (3)$$

como uma função de transferência. Onde: δ é o ângulo de leme; r é a velocidade angular; K é o ganho; T a constante de tempo.

Os parâmetros K e T são identificados ao observar a resposta ao degrau unitário.

Usando ψ e representando no domínio do tempo:

$$T\ddot{\psi} + \dot{\psi} = K\delta \quad (4)$$

Modelo dinâmico: velocidade longitudinal

Para a velocidade longitudinal, é utilizado um modelo não linear:

$$(m - X_{\dot{u}})\dot{u} - X_{|u|u}u|u| = \tau_1 + w_{\tau_1} \quad (5)$$

onde: m é a massa do navio; $X_{\dot{u}}$ e $X_{|u|u}$ são os coeficientes de massa adicional e amortecimento; u é a velocidade longitudinal; τ_1 é a força do propulsor (ação de controle); e w_{τ_1} é o ruído de processo devido aos扰动 ambientais (ondas, vento e correntes) e dinâmica não modelada.

Os parâmetros m , $X_{\dot{u}}$ e $X_{|u|u}$ são retirados do arquivo de configuração do navio.

Controle: controle de guinada

Para o controle de guinada é utilizado um controlador PID digital, descrito abaixo:

$$\delta(\tilde{\theta}) = -K_p \cdot \tilde{\theta}_k - K_d \cdot \frac{\tilde{\theta}_k - \tilde{\theta}_{k-n}}{t_k - t_{k-n}} - \text{enable}(\tilde{\theta}_k) \cdot \left(K_i \cdot \sum_{i=0}^{k-1} \tilde{\theta}_i \cdot \Delta_t \right) \quad (6)$$

Onde: θ é o ângulo de guinada medido de leste no sentido anti-horário; $\tilde{\theta}_k$ é o erro no instante k ; K_p [adim], K_d [s] e K_i [1/s] são os ganhos do PID; Δ_t é o passo de tempo do simulador; $\text{enable}(\tilde{\theta})$ é uma estratégia de *anti-windup* que desativa a contribuição integral para erros grandes; e n é um parâmetro usado para mudar a aproximação da derivada do erro.

Os parâmetros K_p [adim], K_d [s] e K_i [1/s] são projetados, depois tunados para a planta de guinada, e por fim, tunados na arquitetura de produção.

Controle: controle de velocidade longitudinal

Para o controle da velocidade longitudinal, é utilizado o *Sliding Mode Control*. Aqui a notação está um pouco diferente: x refere-se a velocidade longitudinal e $u = \frac{\tau_1}{m - X_{\dot{x}}}$ é o input do modelo dinâmico da variável de escorregamento $s = \tilde{x}$.

O controlador fica definido como:

$$u = -\hat{f}_p - k \cdot \text{sat} \left(\frac{s(x)}{\phi} \right) \quad (7)$$

Onde: \hat{f}_p é a estimativa pontual de $f(x) = \frac{X_{|x|}x|x|}{(m - X_{\dot{x}})} + w_{\tau_1}$; k precisa ser o suficientemente grande para atingir os requisitos de aceleração impostos e compensar incertezas de $f(x)$, para garantir o escorregamento; e ϕ é um parâmetro que dita a espessura da camada limite.

Guiança

O método de guiança (prover os set-points para o controlador de guinada) utilizado é o *enclosure-based-LOS (Line-Of-Sight)*. A essência desse método é cruzar uma circunferência em torno do navio com a linha do caminho para obter ângulo de guinada desejado. As equações principais estão abaixo:

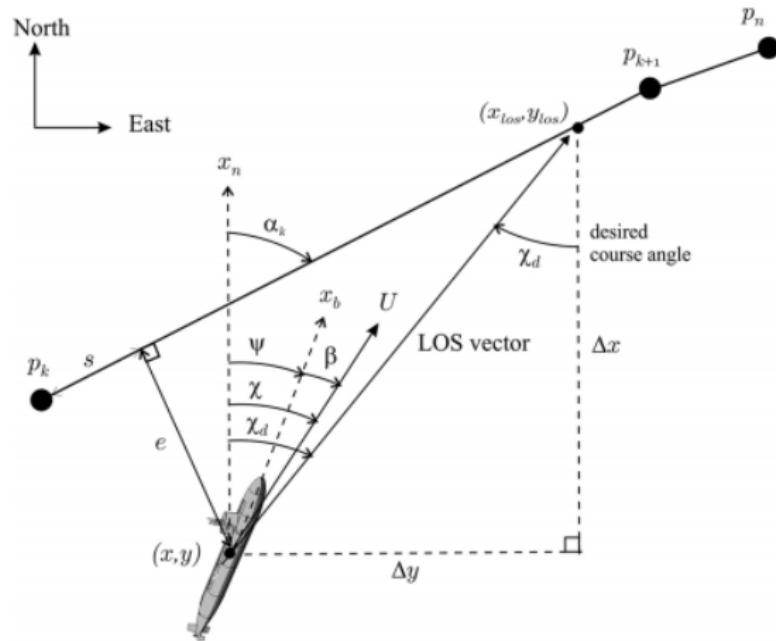
$$\psi_d(t) = \chi_d(t) + \beta \quad (8)$$

$$\chi_d(t) = \text{arctan}2(y_{los} - y(t), x_{los} - x(t)) \quad (9)$$

onde: $\psi_d(t)$ é o ângulo desejado de guinada, β é o *offset angular* entre a guinada e velocidade do navio (causado por velocidade latitudinal imposta pelo ambiente); (x_{los}, y_{los}) é o ponto onde a circunferência do navio intercepta a linha do caminho; e $(x(t), y(t))$ é a localização do navio no instante t .

Guiança

As variáveis envolvidas no algoritmo LOS podem ser conferidas no diagrama abaixo:



Filtros: filtro de onda

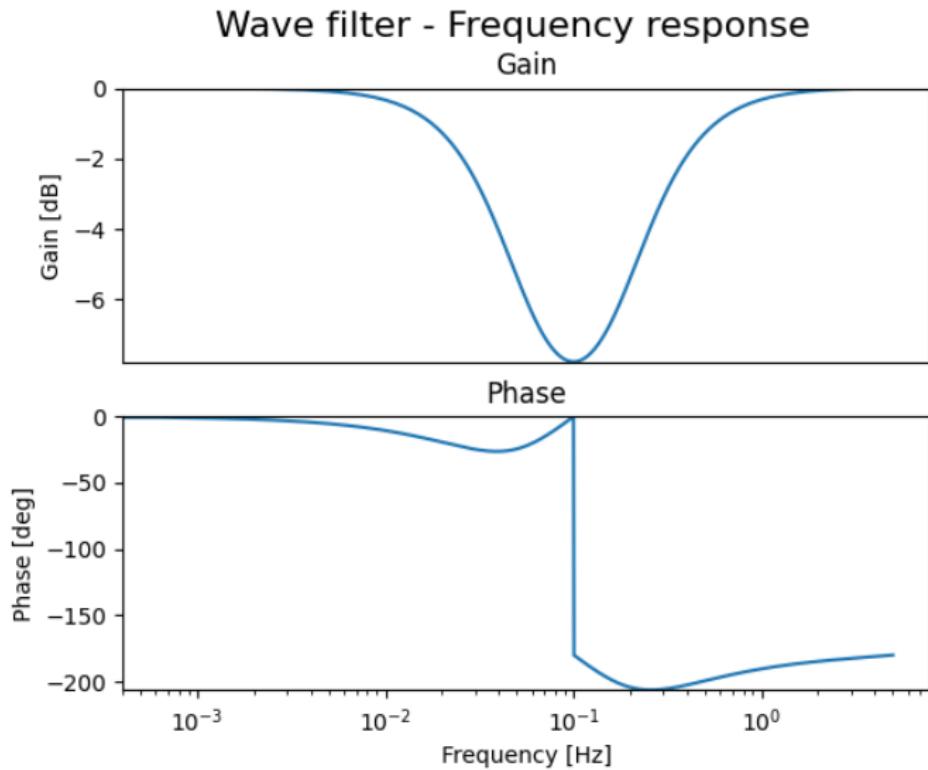
O filtro de onda implementado é um filtro *notch* digital de 6^a ordem, projetado no domínio contínuo e passado para o domínio discreto pela transformada bi-linear.

O filtro no domínio contínuo:

$$h_n(s) = \prod_{i=1}^3 \frac{s^2 + 2\zeta s + w_i^2}{(s + w_i)^2} \quad (10)$$

onde: ζ é o coeficiente de amortecimento; w_1 [rad/s]; w_2 [rad/s]; w_3 [rad/s] são frequências próximas. A frequência central w_2 [rad/s] é a frequência da onda (extraída do arquivo de configuração).

Filtros: filtro de onda. Diagrama de Bode.



Filtros: filtro de ruído dos sensores

Sensores são emulados dentro da arquitetura. Ruído Gaussiano é adicionado ao estado real, com intuito de aproximar a simulação de uma situação real. Para remover o ruído de alta frequência, um filtro passa-baixa digital de 6^a ordem é utilizado. O diagrama de Bode pode deste filtro pode ser visto a direita.

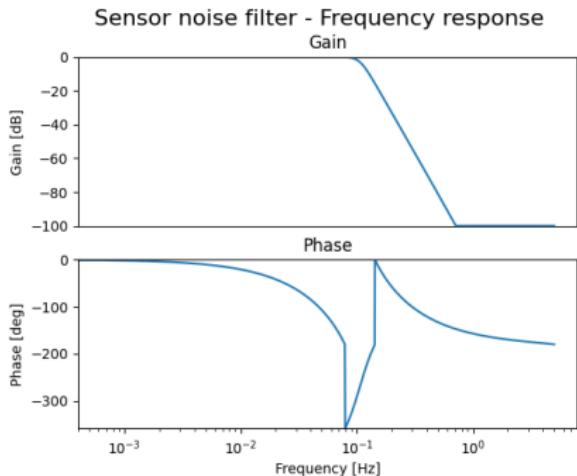


Figura: Diagrama de Bode do filtro de ruído dos sensores.

Caso 7: condições

- Zigzag *waypoints*.
- estado inicial:
 - $(x, y, \theta) = (0 \text{ m}, 0 \text{ m}, 1.2 \text{ rad})$;
 - $(u, v, r) = (2 \text{ m/s}, 0 \text{ m/s}, 0 \text{ rad/s})$;
- emulação de ruído dos sensores: ativada;
- $R_w = R_{los} = 2L_{pp}$;
- $R_{wFinal} = 50 \text{ m}$;
- coordenadas iniciais: $(-23.06255, -44.2772)$ ¹;
- condição ambiental: condição ambiental principal (descrita num slide posterior);
- chegou no *waypoint* final: sim.

¹Localização: Angra dos Reis, Rio de Janeiro, Brasil.

Caso 7: condições. *Waypoints*

Tabela: Case 7: *waypoints*

Waypoints	Posição x [m]	Posição y [m])	Velocidade u [m/s]
1	500	600	3
2	1000	900	3.5
3	1500	1600	4
4	2000	1900	4.5
5	2500	2600	5

Caso 7: condições. Condicão ambiental principal

- ondas: com $T_w = 12 \text{ s}$, $h = 2 \text{ m}$ e atuando a 225 deg (a partir de leste anti-horário). Onde T_w é o período da onda e h a altura dela;
- vento: com $v_w = 8.74 \text{ m/s}$ e atuando a 210 deg (a partir de leste anti-horário). Onde v_w é a velocidade do vento;
- corrente: com $v_c = 1 \text{ m/s}$ e atuando a 330 deg (a partir de leste anti-horário). Onde v_c é a velocidade da corrente.

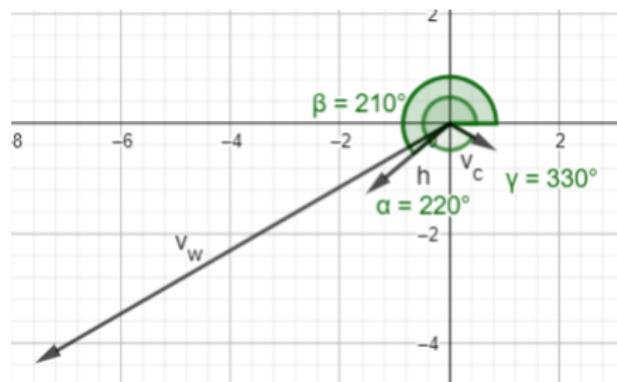


Figura: Vetores que descrevem a condição ambiental principal. Errata:
 $\alpha = 225 \text{ deg}$

Caso 7: GIF

Caso 7: métricas de erro obtidas

Tabela: Caso 7: métricas

Run	cte_m [m]	cte_{max} [m]	we_m [m]	we_{max} [m]
1	14.8	35.6	39.3	87.3
2	15.3	34.6	40.6	95.5
3	16.3	47.2	43.2	98.7

Onde: cte_m é o *cross-track error* médio; cte_{max} é o *cross-track error* máximo; we_m é o *width error* médio; e we_{max} o *width error* máximo.

Cross-track error é o erro perpendicular ao caminho, medido a partir do centro de massa do navio. Já o *width error* é medido em relação ao ponto do navio mais distante do caminho.

Caso 7: gráficos obtidos. Velocidade longitudinal.

Figura: Caso 7: velocidade longitudinal real e medida pelo sensor. A real está em laranja, e a medida pelo sensor em azul.

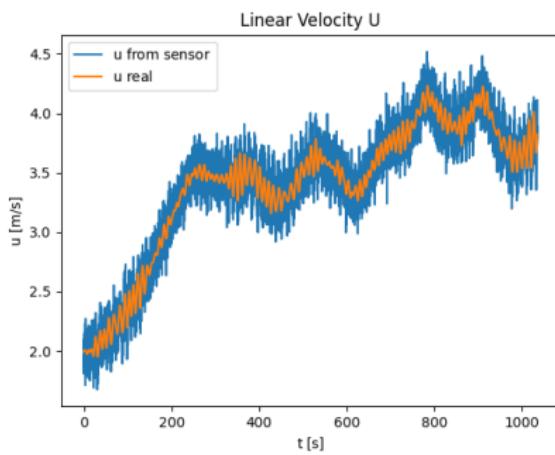
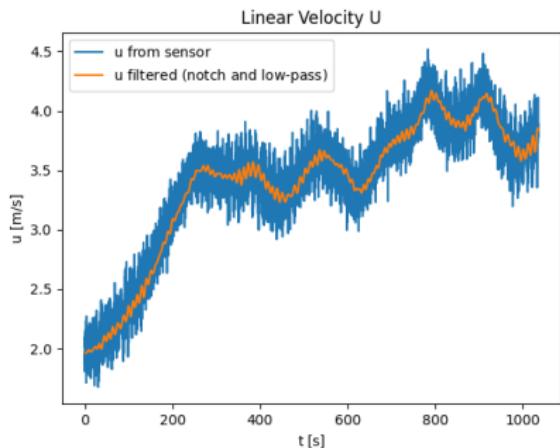


Figura: Caso 7: velocidade longitudinal filtrada e medida pelo sensor. A filtrada está em laranja e a medida pelo sensor em azul.



Caso 7: gráficos obtidos. Ângulo de guinada.

Figura: Caso 7: ângulo de guinada real e medido pelo sensor. O real está em laranja, e o medido pelo sensor em azul.

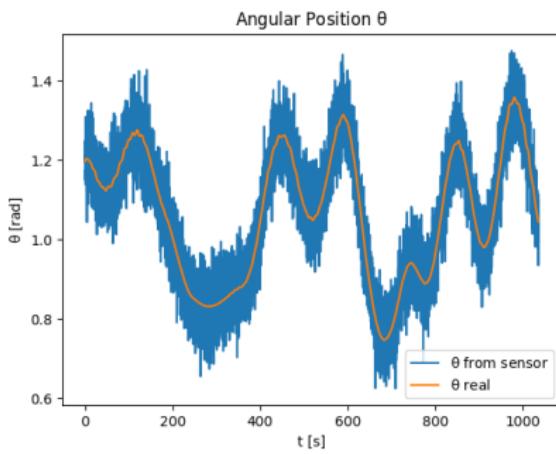
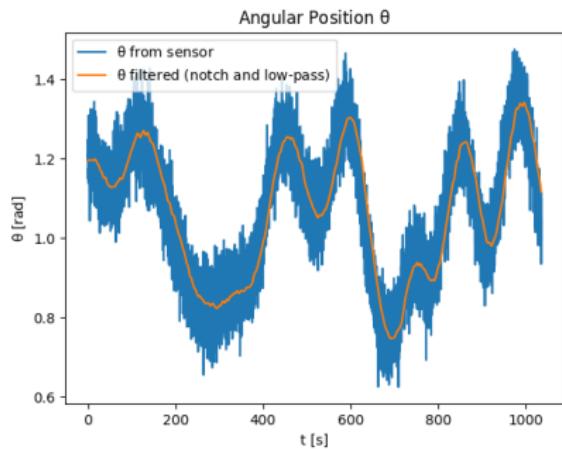


Figura: Caso 7: ângulo de guinada filtrado e medido pelo sensor. O filtrado está em laranja, e o medido pelo sensor em azul.



Caso 7: gráficos obtidos. Atuadores.

Figura: Caso 7: rotação do propulsor.

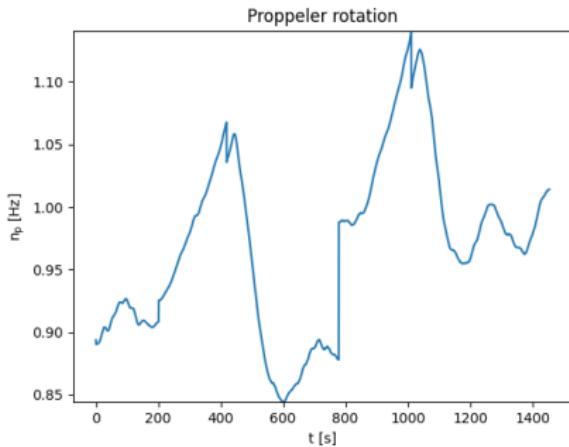
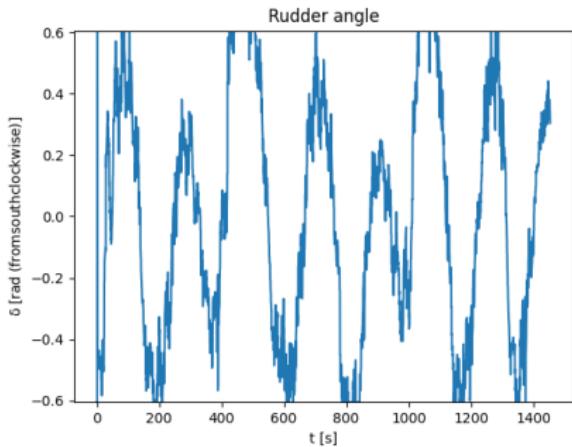
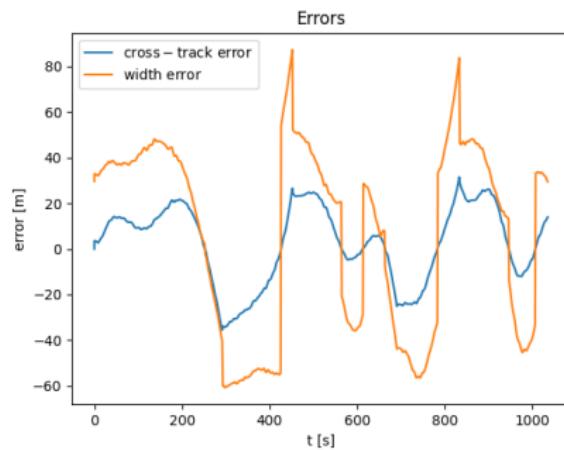


Figura: Caso 7: ângulo do leme.



Caso 7: gráficos obtidos. Erros

Figura: Caso 7: *cross-track error* e *width error*. *Cross-track error* está em azul, e *width error* em laranja.



Discussão

- Capacidade de seguir caminho: o erro médio de *cross-track* obtido foi de $cte_m = 15.47 \text{ m}$. Como $15.47 < \frac{beam}{2} = \frac{32.2}{2} = 16.1 \text{ m}$, logo o navio foi bem sucedido na tarefa de seguir o caminho.
- capacidade de seguir velocidade: a velocidade longitudinal real não seguiu exatamente as desejadas, mas teve um comportamento próximo. É preciso levar em conta o modelo desacoplado e ruído de baixa frequência presente.
- comportamento dos atuadores: o propulsor teve um comportamento suave em geral. O leme teve uma componente de alta frequência que pode levar à um desgaste mais rápido.
- observação: a tarefa de seguir as velocidades desejadas funciona muito bem para caminhos co-lineares.

Conclusão

Os pacotes *ROS2* abaixo foram desenvolvidos e contém documentação de uso geral.

- Pacote do simulador de manobras sozinho: *pydyna_simple*
- pacote do seguidor de caminho: *path_following*

Assim, uma nova infraestrutura para as aplicações do TPN foi apresentada. Através de uma prova de conceito, em forma de estudo de caso, para as simulações *fast-time*. O projeto de canais pode se tornar mais eficiente e robusto por consequência.

O time do TPN tem uma aplicação base que serve de ponto de partida para realizar projetos de grande impacto dentro do ambiente *ROS*.

Referências

-  L. G. Kofi. (2021, Apr.) The ever given – suez canal accident: The genesis to the current legal proceedings. Odomankoma. [Online]. Available: <https://odomankoma.com/2021/04/19/the-ever-given-suez-canal-accident-the-genesis-to-the-current-legal-proceedings/>
-  (2020, Jun.) Technical studies. Numerical Offshore Tank. [Online]. Available: <https://tpn.usp.br/simulador/TechnicalStudies.html>
-  N. P. School, "Robotics operating system short-course," Jul. 2020, figura retirada dos slides do curso.
-  (2020, Jun.) Homepage. Numerical Offshore Tank. [Online]. Available: <https://tpn.usp.br>
-  Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

Obrigado!