

DESENVOLVIMENTO DE INTERFACES PARA A OPERAÇÃO DE UMA PLATAFORMA DE ENSAIOS DE EMBARCAÇÕES

MARIANE MENDES MEDEIROS¹; VITOR IRIGON GERVINI²; VAGNER SANTOS DA ROSA³;
SEBASTIÃO CÍCERO PINHEIRO GOMES⁴;

Abstract — This article gives an overview of MANOBRAS, an interface developed to perform the computer operation of a oceanic maneuvering platform, at COPPE-RJ. It starts with a brief description of the platform itself, followed by a description of all the software system: the main interface (including trajectory's generation aspects), the 3D simulator and finally the communication protocol between computer and onboard system.

I. INTRODUÇÃO

Todo sistema controlado por computadores, seja ele de natureza mecânica, elétrica, ou mesmo puramente computacional, que necessite de intervenção humana, exige obrigatoriamente algum tipo de interface homem/máquina. Segundo [1], a interface pode ser definida como um “software que dá forma à interação entre usuário e computador”, e neste sentido, possui importância fundamental para um bom aproveitamento dos recursos disponibilizados pelo sistema.

Neste contexto, uma interface necessita agregar de forma simples e objetiva todos os recursos e funcionalidades disponíveis, abstraindo do usuário a complexidade natural dos sistemas computacionais e deixando-o apto a utilizar o sistema com eficiência e segurança, sem a necessidade de conhecimentos técnicos aprofundados.

¹ Centro de Ciências Computacionais – FURG Universidade Federal do Rio Grande – mariane.mm@gmail.com

² Centro de Ciências Computacionais – FURG Universidade Federal do Rio Grande – gervini@gmail.com

³ Centro de Ciências Computacionais – FURG Universidade Federal do Rio Grande – vsrosa@inf.ufrgs.br

⁴ Instituto de Matemática, Estatística e Física – FURG Universidade Federal do Rio Grande – sebastiao.gomes@furg.br

Este trabalho apresenta, então, a interface MANOBRAS, desenvolvida para a operação da plataforma de ensaios oceânicos da COPPE – UFRJ. Esta plataforma destina-se a realizar ensaios dinâmicos com modelos reduzidos de embarcações, possibilitando até cinco movimentos simultâneos do modelo reduzido. Em [2] existem maiores detalhes sobre o projeto que originará a plataforma de ensaios, com previsão de entrada em operação durante o ano de 2010. O modelo reduzido da embarcação descreverá uma trajetória no interior de um tanque com importantes dimensões, trajetória essa gerada a partir da combinação de até cinco movimentos independentes. Estes movimentos do modelo reduzido permitirão a coleta de dados sobre a embarcação a ser construída, sendo, portanto, de grande importância para a indústria da construção naval.

Conforme adiantado anteriormente, este artigo apresenta a interface para a futura operação da plataforma, interface esta aqui designada MANOBRAS. Dentre as principais tarefas que podem ser realizadas a partir desta interface encontram-se: a geração das trajetórias a serem executadas, ou seja, trajetórias de referência a serem seguidas pelo modelo reduzido; a configuração de uma série de parâmetros necessários ao sistema de controle; a comunicação com o sistema de tempo real embarcado que de fato controla os atuadores da plataforma, a partir da qual são enviados ao sistema embarcado todos os parâmetros pré-configurados pelo usuário via interface.

II. VISÃO GERAL DO SISTEMA DE CONTROLE

Seis atuadores são responsáveis pelos movimentos do modelo reduzido, todos de corrente alternada. Sensores do tipo encoder incremental fornecem os dados de posição angular, os quais estão acoplados aos rotores dos atuadores. Existem ainda quatro sensores do tipo encoder absoluto, utilizados principalmente para o referenciamento inercial. Dois atuadores movimentarão a plataforma principal, responsável pelo movimento longitudinal, conforme pode ser visto no desenho da Figura 1. Portanto, existe um atuador em cada extremidade da plataforma principal e estes têm de funcionar em sincronia. Um outro atuador será o responsável pelo movimento transversal, conforme indicado na Figura 1. Um carro sobre trilhos se deslocará movido por este último atuador, ao longo da plataforma principal, gerando

assim o movimento transversal. Em resumo, estes dois movimentos são os responsáveis pelo deslocamento do centro de massa do modelo reduzido no plano horizontal.

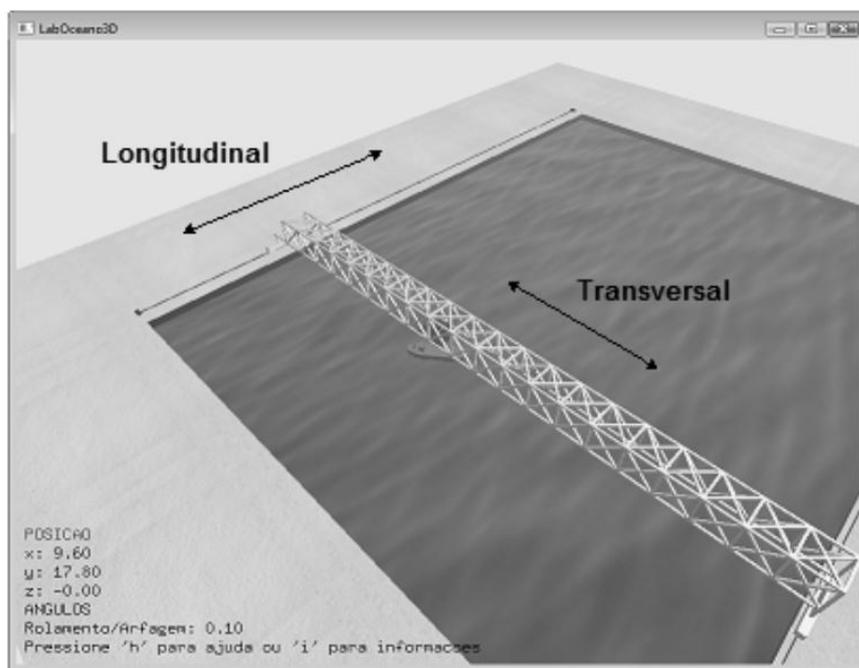


FIGURA 1: Imagem mostrando os dois principais movimentos de translação do modelo reduzido.

A Figura 2 mostra mais quatro possíveis movimentos do modelo reduzido: linear vertical, rotação em torno do eixo vertical, arfagem e rolamento. Estes dois últimos não ocorrerão ao mesmo tempo, ou seja, mecanicamente, deve-se, no momento da montagem do modelo reduzido na plataforma, escolher sobre qual dos dois movimentos pretende-se que ocorra.

O sistema de controle para a plataforma de manobras foi desenvolvido objetivando o controle dos seus cinco graus de liberdade (movimentos vistos nas Figuras 1 e 2).

A opção de projeto foi por um controle Proporcional e Integral (PI) em velocidade ([4], [5]), utilizado em conjunto com uma correção de posição baseada em um algoritmo fuzzy [6]. Considerou-se que todos os graus de liberdade são desacoplados do ponto de vista dinâmico, de forma que se projetou um controle específico para cada grau de liberdade. O controle é subdividido em duas malhas: uma mais interna, na qual existe o PI em velocidade, e outra mais externa, na qual se utiliza o ajuste da posição. O controlador PI gera o torque motor (T_m), sendo este

enviado para a planta física (atuador). O sinal de posição x (saída do sistema, equivalente ao sinal lido do encoder) é então comparado com a posição de referência x_r para gerar o sinal de erro em posição, o qual corresponde à entrada no algoritmo de correção de posição. Este algoritmo devolve um sinal que é acrescido ao controle, de forma a minimizar o erro em posição.

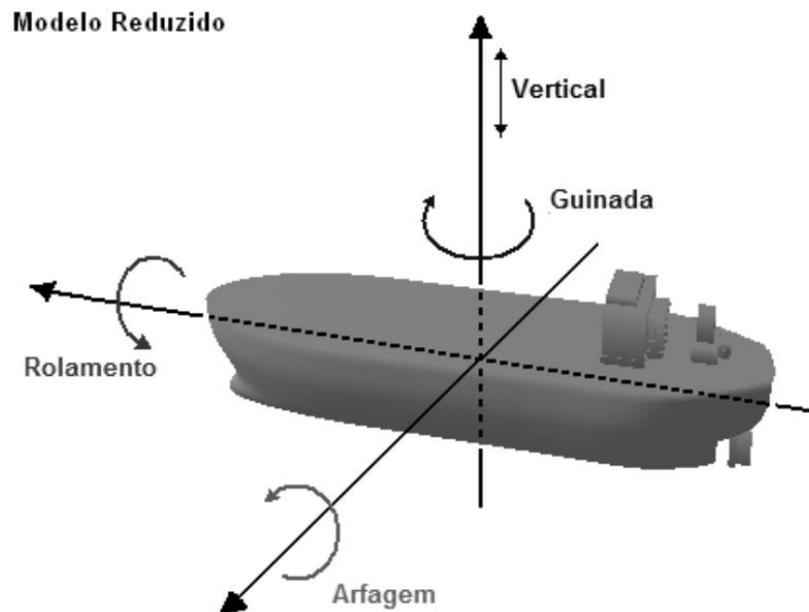


FIGURA 2: Imagem mostrando os três movimentos de rotação do modelo reduzido.

Os dois atuadores responsáveis pelo movimento da plataforma principal recebem os nomes de atuadores mestre e escravo. O controle em posição é aplicado de fato ao atuador mestre. Uma malha de controle interno (entre os atuadores mestre e escravo) faz com que o escravo siga o mestre com erro mínimo, garantindo assim o sincronismo do movimento.

A Figura 3 mostra um resultado experimental com o controle do atuador mestre, no seguimento de uma trajetória de referência. Tanto o sinal do encoder (posição real do atuador (m)) quanto o sinal de referência são mostrados no gráfico. O experimento teve duração total de 90 segundos e foi realizado com o atuador sem carga, uma vez que a plataforma de manobras ainda não foi montada sobre o tanque (montagem prevista para 2010). Percebe-se que o erro no seguimento da trajetória de referência foi quase nulo, denotando assim que o controle apresentou um bom desempenho.

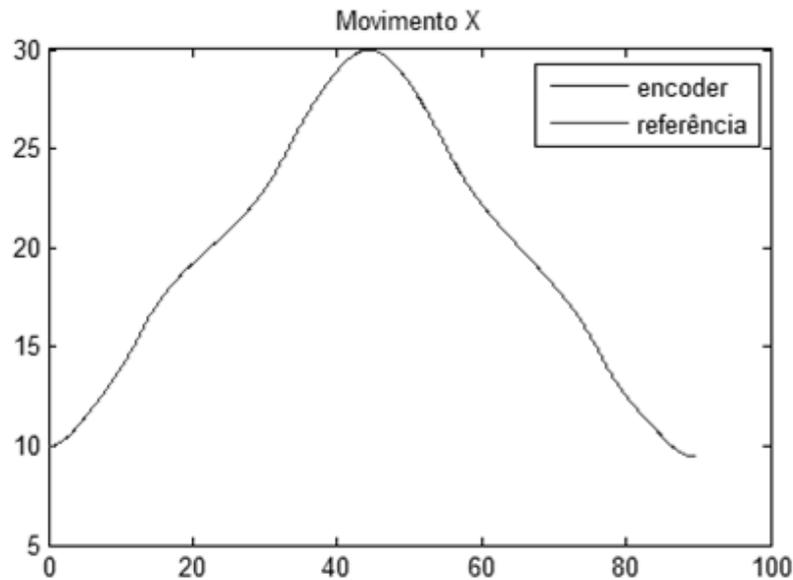


FIGURA 3: Resultado experimental relativo ao controle do atuador mestre da plataforma principal.

III. SISTEMA DE INTERFACE MANOBRAS

A interface MANOBRAS foi desenvolvida para a interação entre o usuário e o sistema físico da plataforma. A partir desta, estão disponíveis ferramentas essenciais, como o gerador de trajetórias e a comunicação com o sistema embarcado, bem como ferramentas de simulação de trajetórias e modelos dinâmicos em 3D.

De uma forma geral, a interface é composta de três módulos visíveis ao usuário, com alguns módulos auxiliares sendo processados em *background*. Estes últimos são responsáveis pela geração matemática de trajetórias e validação das mesmas, e pela realização de simulações com o modelo dinâmico a partir de trajetórias de referência. A responsabilidade da visualização dos resultados gerados por estes módulos fica a cargo dos módulos essenciais: a interface principal MANOBRAS, o sistema para visualização 3D de trajetórias LabOceano 3D e a comunicação, todas detalhadas na seqüência.

A. Interface Principal

Desenvolvida em C++ Builder, para uso em ambiente Windows, a interface MANOBRAS apresenta ao usuário todos os recursos necessários para a geração de trajetórias a serem executadas pela plataforma, possibilitando a configuração dos

seus parâmetros, como velocidade, aceleração, opções de movimentos desejados (rolamento ou arfagem) com suas respectivas freqüências e amplitudes, entre outros.

A geração é feita em duas etapas, sendo a primeira a seleção de coordenadas do tanque que criarão a trajetória de pontos base, por onde deseja-se que a trajetória final passe. Esses pontos são informados pelo usuário com o mouse, diretamente na tela principal da interface. Um exemplo de geração pode ser visto na Figura 4. Os pontos em destaque correspondem aos cliques com o mouse, sendo o ponto mais espesso o indicador do início da trajetória. Estas coordenadas são então submetidas a um módulo que calcula a curva final via splines de interpolação. Este tipo de interpolação faz sucessivos ajustes de curvas para que a curva resultante passe por um determinado conjunto de pontos chamados nós (no caso, os pontos selecionados pelo usuário como base para a geração). A Figura 4 mostra o resultado obtido para uma trajetória de pontos composta de nove coordenadas selecionadas. A curva interpolada, gerada via splines, será a trajetória de referência.



FIGURA 4: Geração de uma trajetória de referência a partir de interpolação utilizando splines.

Cada trajetória gerada via splines possui três informações para cada um dos seus milhares de pontos: informação de posição longitudinal (X), de posição transversal (Y) e de giro (θ) para o modelo reduzido em um determinado instante t de tempo. O tempo de amostragem utilizado como passo para a discretização das trajetórias geradas é fixo, sempre igual a $2.6ms$.

Além da geração de trajetórias, a interface MANOBRAS gerencia também as trajetórias armazenadas pelo usuário, bem como perfis de parâmetros padrões, históricos de uso e funcionamento do software em geral e arquivos de ajuda. Sendo esta a interface principal, ela dá acesso aos demais módulos, abstraindo do usuário o processo de troca de informações entre os mesmos. Conta, por fim, com diversas verificações de segurança importantes para garantir a integridade do sistema,

processando ainda no computador rotinas que deixam de ser averiguadas no sistema embarcado, diminuindo assim o tempo de comunicação com a plataforma e, conseqüentemente, a possibilidade de ocorrência de erros.

Quando o operador solicita uma trajetória de referência para ser executada na plataforma, a posição atual do modelo reduzido pode não estar coincidente com a posição inicial da trajetória solicitada pelo operador, a qual recebe o nome de trajetória principal. O software da interface gera então uma trajetória de referência (aqui chamada de trajetória primária), a fim de posicionar corretamente o modelo reduzido no início da principal. A trajetória primária é formada a partir de três movimentos distintos. Um primeiro de rotação sobre o eixo vertical, a fim de orientar corretamente o modelo reduzido na direção e sentido do início da principal. O segundo movimento é ao longo de uma reta, até que seja atingido o ponto do início da principal. O terceiro e último movimento é uma nova rotação em torno do eixo vertical, de forma a orientar o modelo reduzido na direção tangente à trajetória principal, no ponto correspondente ao início desta. A Figura 5 tenta ilustrar este processo, sendo a trajetória curvilínea a trajetória principal.

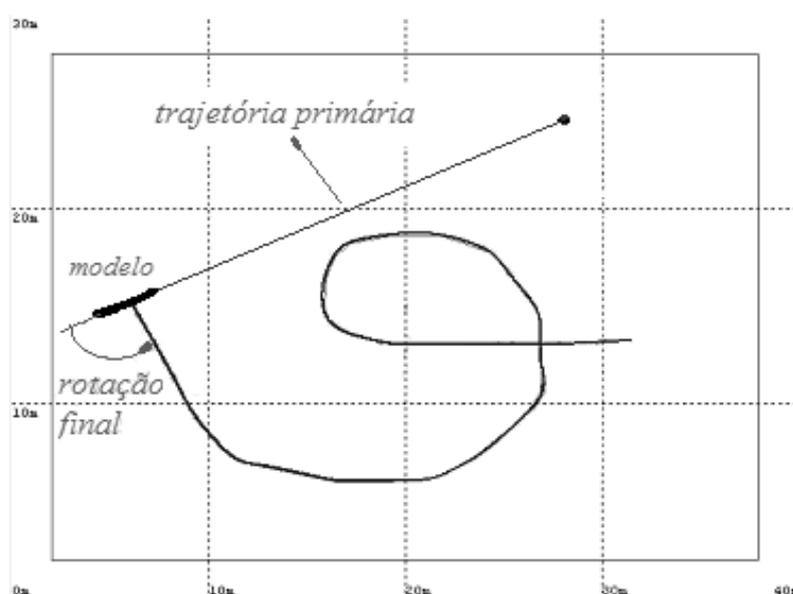


FIGURA 5: As trajetórias primária e principal.

A Figura 6 mostra resultados experimentais relativos à realização de uma trajetória primária. No gráfico superior encontra-se o movimento longitudinal da plataforma principal. No gráfico intermediário vê-se o movimento transversal da plataforma secundária, enquanto que no último vê-se o movimento de rotação em

torno do eixo vertical. Tanto as posições de referência quanto os sinais lidos dos encoders (respostas experimentais) são mostradas nos gráficos. Percebe-se, nas três respostas, que os erros nos seguimentos das trajetórias são quase nulos, ressaltando assim o bom desempenho do controle. Pode ser percebido ainda que, durante a primeira e a última rotação, as posições longitudinal e transversal permanecem constantes. Durante o movimento intermediário (reta), quem fica constante é a posição angular (velocidade angular nula em trono da vertical).

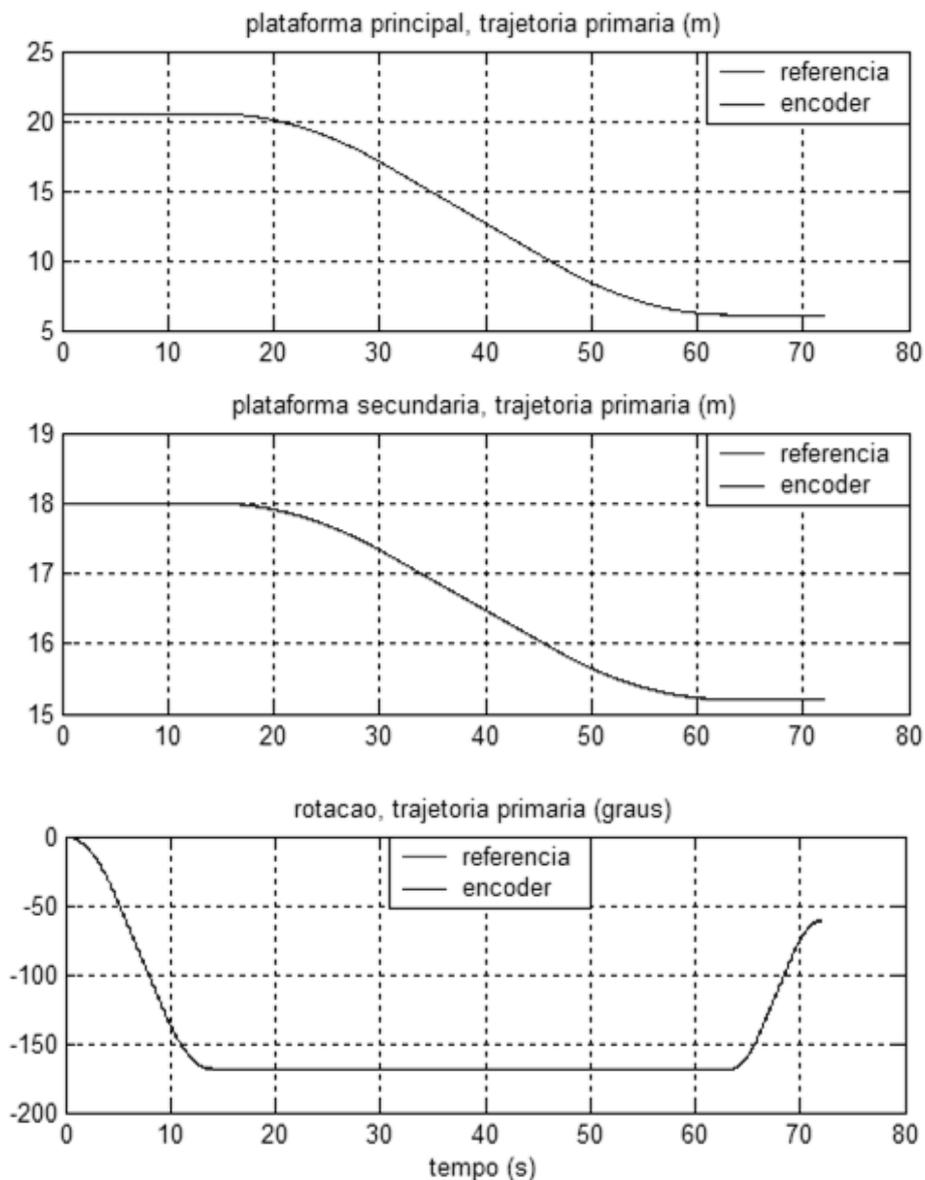


FIGURA 6: Resultados experimentais mostrando a realização de uma trajetória primária.

B. LabOceano 3D

Desenvolvido em Dev-C++ e utilizando-se da biblioteca gráfica OpenGL, o software para a animação 3D, intitulado LabOceano3D [7], proporciona ao usuário a possibilidade de executar trajetórias em ambiente simulado 3D muito semelhante ao ambiente real da plataforma.

Dentre as facilidades por ele disponibilizadas ao usuário está o acompanhamento da execução da trajetória (ou de trajetórias oriundas de simulações com o modelo dinâmico da plataforma) a partir da visualização de 9 ângulos diferentes da plataforma e do modelo reduzido. A Figura 7 mostra um ângulo de visão que ressalta em detalhes os movimentos dos mecanismos próximos ao modelo reduzido. A Figura 8 mostra mais dois diferentes ângulos de visão (câmeras) que o sistema de animação 3D disponibiliza ao usuário. Observa-se que a primeira imagem refere-se a uma câmera com visão inferior, ou seja, como se o observador estivesse dentro d'água, no fundo do tanque. O usuário pode ainda visualizar as informações de posição de todos os movimentos da plataforma a cada instante de tempo, conforme indicado na parte inferior das figuras 7 e 8.

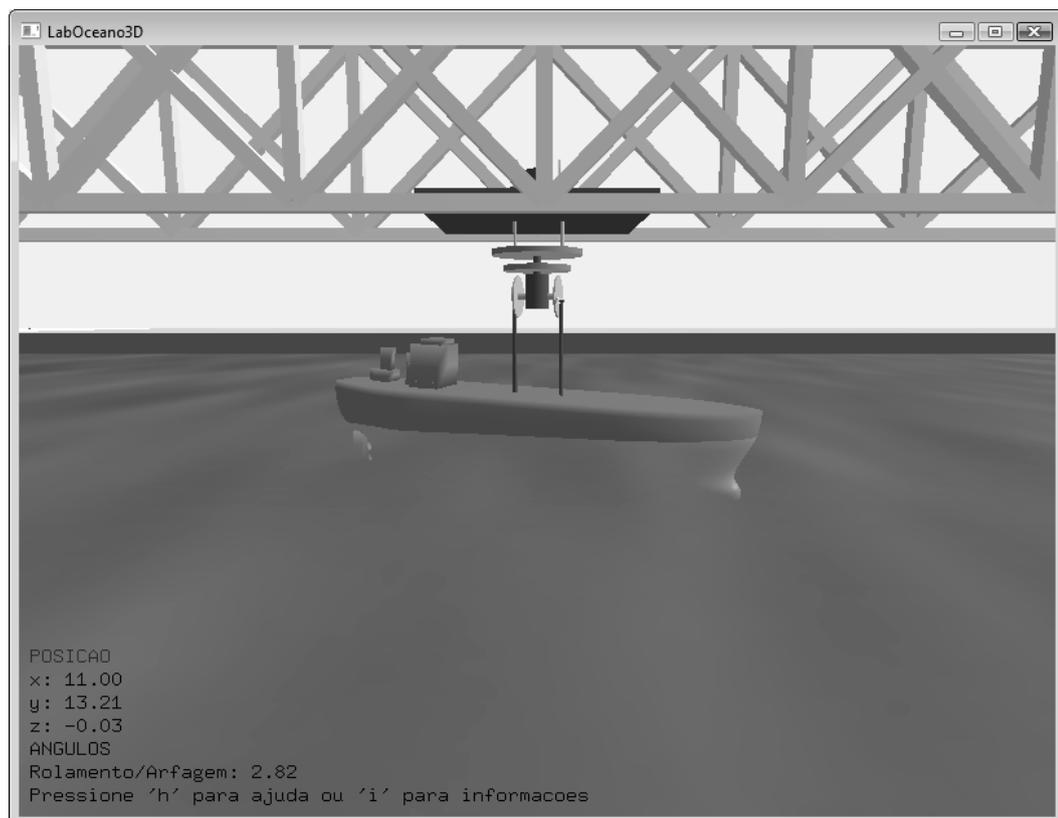


FIGURA 7: Visualização da trajetória a partir de um dos ângulos de visão do LabOceano 3D.

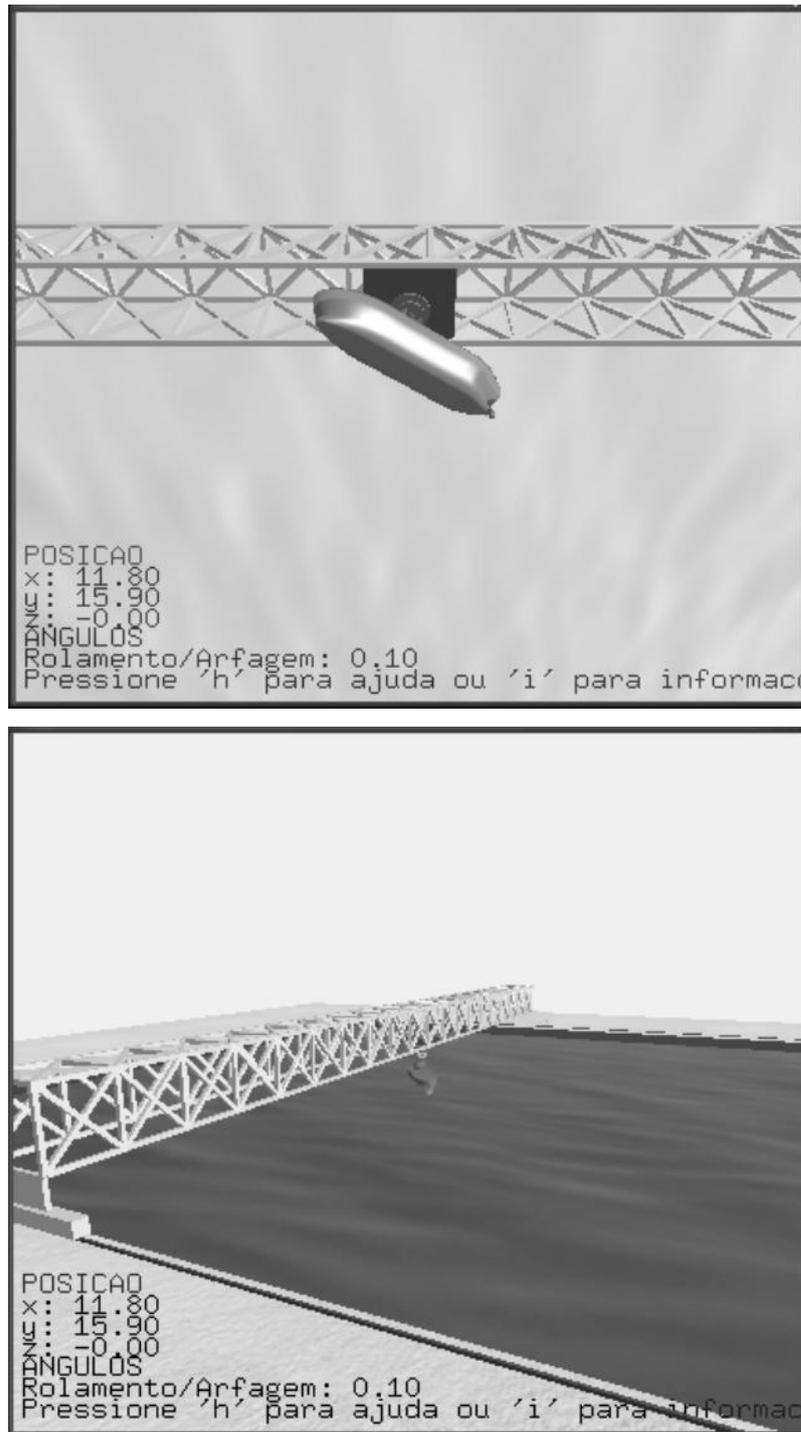


FIGURA 8: Dois dos nove ângulos de visão (câmeras) do sistema de animação 3D.

A visualização da animação 3D permite possibilidades de aproximação/afastamento do ponto de observação da simulação, aumento/diminuição do modelo reduzido e, por fim, imersão/emersão do mesmo. Pode ser dito que a sua maior utilidade reside no fato de que a execução da trajetória pode ser validada primeiramente no âmbito computacional, para só então

ser repassada à plataforma para execução real, evitando a movimentação desnecessária do sistema físico envolvido no processo.

C. Comunicação entre PC e Sistema Embarcado

A comunicação possibilita que todos os dados coletados pela interface e a trajetória gerada a ser executada sejam enviados a um dispositivo embarcado responsável pelo controle da plataforma. Este é um dispositivo do tipo FPGA (*Field-Programmable Gate Array*, modelo Virtex-II Pro, ver [3]), responsável pelo controle de todo o sistema da plataforma bem como pela verificação de sua integridade, em tempo real. Sendo assim, o FPGA permanece sempre disponível às solicitações advindas do computador do usuário para execução de tarefas, bem como está apto a coletar e disponibilizar informações quando necessário como resposta às solicitações do mesmo.

Por tratar-se de comunicação entre dois dispositivos de execução relativamente independentes, foi necessário a criação de um protocolo de comunicação que organizasse o processo. Ou seja, assim como o computador deve permanecer à disposição do usuário, o dispositivo FPGA deve fazer todo o controle em tempo real já mencionado, de maneira que um dispositivo não interfira no outro a não ser quando a demanda exigir. Sendo a comunicação feita de forma serial (utilizando-se do padrão RS232), optou-se então por uma comunicação onde apenas o computador possa requisitar ações à FPGA, nunca o oposto. As ações permitidas são identificadas por um código, seguidas de um comando de sincronização de comunicação que garanta que ambos os lados estão respondendo corretamente ao processo.

Devido a este modelo de protocolo estabelecido, o FPGA deve de alguma forma armazenar todo tipo de informação que possa ser útil ao usuário, uma vez que não as pode entregar ao computador senão via algum tipo de solicitação do mesmo. Estas informações podem ser tanto dados referentes à execução de trajetórias, a serem analisados pelo usuário, quanto podem ser informações de status do funcionamento do sistema, reportando neste caso, situações de sucesso ou erro em operações e possíveis falhas em componentes físicos da plataforma.

Tendo sido desenvolvida em Matlab, a comunicação apresenta-se em forma de console, com menus que abrangem todas as funcionalidades necessárias para a

realização do experimento sobre a trajetória escolhida e todas as verificações necessárias de funcionamento dos componentes físicos e de segurança. A Figura 9 mostra uma imagem oriunda de uma captura da tela do computador do usuário, durante a fase de comunicação com o FPGA. O sistema de comunicação permite ainda a coleta de dados de cada um dos movimentos após a realização de um experimento (ensaio com o modelo reduzido na plataforma), com possibilidade de visualização de gráficos 2D com os resultados.

```
selecione opcao para a trajetoria "testefinal.trj":
1 - Criar/Enviar Trajetoria Primaria
2 - Enviar Trajetoria Principal
3 - Controle
4 - Recepcao de log
5 - Plotagem dos dados de log
6 - Verificacao geral do sistema
7 - Reposicionamento manual da plataforma
8 - Modo manual (keyboard)
9 - Sair

opcao:
Iniciando...
solicitando posicoes atuais da plataforma...
- Encoder absoluto X ..... ok
- Encoder absoluto Y ..... ok
- Encoder absoluto Giro .... ok

Posicao absoluta da plataforma : longitudinal (X) ..... 18.035 m
                               transversal (Y) ..... 19.9347 m
                               angular ..... 50.002°

Posicao inicial da trajetoria : longitudinal (X) ..... 8.6 m
                              transversal (Y) ..... 19.5 m
                              angular ..... 34.7035°

Enviando parametros...
Envio de parametros confirmado!

solicitando envio de dados da trajetoria...
solicitacao aceita!

Enviando dados da trajetoria: 8 parte(s)!

  Enviando bloco ... 1/8
  OK!
```

FIGURA 9: Console da Comunicação

De todas as ações disponibilizadas pela interface de comunicação, o envio de trajetórias é a opção que demanda maior tempo e atenção do usuário. Isso se deve ao fato de que, devido às especificações técnicas do padrão serial RS-232 e do modelo FPGA utilizado, a transmissão de dados é feita a uma taxa de apenas 57600 bits por segundo, comparado a uma taxa de 10 megabits por segundo da tecnologia de interconexão 10Base-T Ethernet (IEEE 802.3), por exemplo. Da mesma forma, o

envio serial apresenta maior suscetibilidade a erros, uma vez que ruídos externos podem interferir na confiabilidade da transmissão dos dados. Sendo assim, o protocolo foi desenvolvido baseando todo o envio de dados em pacotes de no máximo 10000 bytes e verificações de integridade via códigos de checksum. Para tal, o computador calcula o checksum do pacote a ser enviado e recebe do FPGA o código por ele calculado. No caso de ambos serem diferentes, sabe-se que houve algum tipo de interferência e o bloco é reenviado, permitindo a recuperação da comunicação sem perda de informações e nem de tempo.

De forma experimental, dados de envio de trajetórias foram levantados. O tempo de envio de cada pacote varia em torno de 7.4 segundos, e todo o processo de verificação de integridade de dados provou a robustez da comunicação no envio de trajetórias grandes, com 60 pacotes, por exemplo (aproximadamente 191.000 pontos gerados via spline). Isso prova a eficiência do protocolo desenvolvido sobre a interface RS232 no ambiente industrial no qual a plataforma está inserida, independente de ruídos e conversores utilizados pra a propagação do sinal através dos cabos seriais.

Uma vez enviada a trajetória ao FPGA, uma verificação geral do sistema físico é realizada. No caso de todos os componentes necessários estarem funcionais, o usuário pode então liberar a plataforma para a execução. Durante este processo, há uma monitoração constante das variáveis de status controladas pelo sistema embarcado, de maneira que assim que o processo de execução seja encerrado, seja com sucesso ou por ocorrência de falhas graves, o usuário seja imediatamente notificado. Em caso de erro, ele poderá inclusive ter ciência de onde exatamente este ocorreu, providenciando as devidas medidas para o reparo.

Por fim, executada a trajetória, todos os dados de posição de referência e dos encoders incrementais de posição de todos os movimentos poderão ser acessados pelo usuário, bem como informações de velocidade. Estes podem ser usados para análise do experimento, a partir da visualização de gráficos 2D, como mostra a Figura 6, ou manipulação dos mesmos via Matlab.

IV. CONCLUSÃO

Partindo-se do princípio da necessidade de um software para a interação

homem/máquina em um sistema de alta complexidade, como o apresentado neste artigo, acredita-se que a proposta de agregar os recursos e funcionalidades em uma interface direta e de simples utilização foi plenamente cumprida com o desenvolvimento da interface MANOBRAS. Os resultados demonstram que a operação do sistema está sendo realizada com sucesso a partir do software desenvolvido, e que estão sendo executadas de maneira adequada desde as mais simples tarefas de gerenciamento de parâmetros até as mais complexas, como a geração de trajetórias e a comunicação com o dispositivo embarcado.

Contando ainda com diversas verificações de segurança, tanto em termos de operação do sistema físico da plataforma quanto em termos de funcionamento de software em si, a interface desenvolvida oferece a garantia de que somente operações seguras sejam executadas pela plataforma. Sendo assim, ela entrega ao usuário um grau de confiabilidade que o libera da decisão de quais tarefas devem ou não ser de fato executadas.

A interface MANOBRAS ainda está em fase de aperfeiçoamento e foi desenvolvida para a utilização na plataforma de ensaios oceânicos da COPPE – UFRJ. Desenvolvida tendo como objetivos a facilidade, a robustez e a confiabilidade de operação, tal solução em software teve sua primeira versão funcional já liberada para uso, embora continue sob manutenção e constante melhoramentos para futuras versões e adaptações às necessidades de evolução do sistema de controle da plataforma de ensaios.

REFERÊNCIAS

- [1] JOHNSON, S. **Interface Culture**. [S.l.]: Jorge Zahar Editor, 1997.
- [2] LEVI, C. A.; PINTO, W. T.; GOMES, S. C. P.; SALES Jr, J. S.; COSTA, P. R.; ROSA, V. S.; GERVINI, V. I. **A maneuvering test platform conceptual design**. 26th International Conference on Offshore Mechanics and Arctic Engineering. San Diego. 2007.
- [3] DIGILENT INC. **Virtex II Pro Development System**. Disponível em: <<http://www.digilentinc.com/Products/Detail.cfm?Nav1=Products&Nav2=Programmable&Prod=X UPV2P>>. Acesso em: Março 2009.

- [4] FRANKLIN, F. G.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback control of dynamic systems**. Addison-Wesley, 1994.
- [5] GERVINI, V. I.; GOMES, S. C. P.; ROSA, V. S. **A new robotic drive joint friction compensation mechanism using neural networks**. Journal of the Brazilian Society of Mechanical Science & Engineering, ABCM, April-June, Vol. XXV, Nº. 2, 2003.
- [6] MACHADO, C. C.; GOMES, S. C. P.; BORTOLI, A. L.; GERVINI, V. I.; ROSA, V. S. **Adaptive Neuro-Fuzzy Friction Compensation Mechanism to Robotic Actuators**. Seventh International Conference on Intelligent System Design and Application, 2007, Rio de Janeiro. ISDA2007, 2007. 6pgs.
- [7] HARTMANN, E.; SHULTZ, T. P.; CASTELLUBER, W.; GERVINI, V. IRIGON; GOMES, S. C. P. **A new simulator to flexible manipulators**. Congresso Nacional de Engenharia Mecânica, 2008, Salvador, BA. CONEM 2008, 8 pgs, 2008.