

# Controle em Tempo Real de uma Plataforma de Manobras através de um Dispositivo FPGA

<sup>1</sup>Dino P. Cassel, <sup>1</sup>Vitor I. Gervini, <sup>1</sup>Vagner S. da Rosa, <sup>2</sup>Sebastião C. P. Gomes

<sup>1</sup>Centro de Ciências Computacionais - FURG  
<sup>2</sup>Instituto de Matemática Estatística e Física - IMEF

## 1. Introdução

A evolução da indústria de microeletrônica possibilita a construção de circuitos integrados com centenas de milhões de transistores, o que possibilita a criação de circuitos digitais extremamente complexos. Esta evolução se aplica a dispositivos FPGA (*field-programmable gate array*), que podem ter até centenas de milhares de blocos de lógica programáveis. Estes dispositivos permitem a concepção de sistemas inteiros incluindo Hardware e Software com apenas um circuito integrado [1].

Com o objetivo de realizar ensaios em modelos utilizando um sistema com cinco graus de liberdade e seis atuadores [2], optou-se por fazer o controle deste usando uma FPGA. Tal sistema possui alguns algoritmos complexos, como recepção e interpretação de comandos de usuário, implementação de uma lei de controle, além do gerenciamento de memória.

## 2. Sistema de Hardware/Software Embarcado

Para implementar um sistema embarcado que controle em tempo real atuadores robóticos foi usada uma placa FPGA Virtex-II Pro [3]. Ela é um dispositivo de hardware programável com dois processadores PowerPC, um módulo de memória e vários dispositivos de I/O. A linguagem de programação usada para a implementação deste software é a C, com algumas diferenças da tradicional pelo fato de se tratar de um dispositivo embarcado [4].

## 3. Comunicação Computador/FPGA

A trajetória de referência que a lei de controle vai seguir é gerada numa interface gráfica instalada no computador. Uma vez gerada, ela será enviada para a FPGA conectada neste computador. Essa comunicação ocorre através da porta serial RS-232, através de um protocolo que envia um bloco de dados e gera um checksum do bloco. Quando a FPGA recebe o dado, ela calcula e verifica o valor do checksum gerado com o calculado.

## 4. Lei de Controle

Os atuadores, que são de corrente alternada, são controlados em malha fechada segundo uma lei de controle do tipo proporcional e integral (PI) em velocidade [4], com correção em posição utilizando lógica FUZZY. Existem algumas restrições para implementar o controle usando o processador PowerPC. Uma destas é a inexistência de tipos de dados em ponto flutuante, que se faz necessária em situações como na aquisição dos sinais lidos dos sensores e dos cálculos da lei de controle. A solução encontrada é converter estes dados para ponto fixo. Essa conversão é feita a partir do uso de variáveis com 32 bits de comprimento, onde 16 são usados para a parte inteira e outros 16 para a parte lógica.

## 5. Gerenciamento de Memória

A placa usada no projeto dispõe de um módulo de memória DDR SDRAM com 2GB de capacidade de armazenamento. Esta pode ser endereçada por dois tipos de barramento, o PLB e OPB. Dispositivos rápidos são conectados no PLB, que está conectado fisicamente com a FPGA, já dispositivos mais lentos são conectados no OPB, que é mapeado no PLB através do módulo plb2opb. O acesso a memória é feito através de um ponteiro do tipo inteiro. Assim, ela pode ser acessada como um vetor unidimensional, onde cada posição é um dado de 32 bits.

Nesse projeto, a memória é dividida em partes iguais, cada uma com tipos específicos de dados (velocidade, posição). O tamanho de cada parte é definido segundo parâmetros enviados juntamente com a trajetória para a placa.

## 6. Resultados

Diversos resultados experimentais foram obtidos na fase de testes do sistema, resultados estes em malha fechada e controlando todos os cinco movimentos simultaneamente. Um destes resultados é uma trajetória de referência chamada de “oito geométrico”, conforme visto na Fig. 1. Esta figura mostra a trajetória de referência e a trajetória de saída.

A Fig. 2 mostra a trajetória de referência e a de saída, no plano horizontal. Observar que a referência é idêntica à gerada na interface do usuário (comparar com a Fig. 1). Pode ser visto que a trajetória real dos atuadores das duas plataformas (em azul) é realmente muito próxima da referência (em vermelho), denotando assim um bom desempenho em malha fechada.

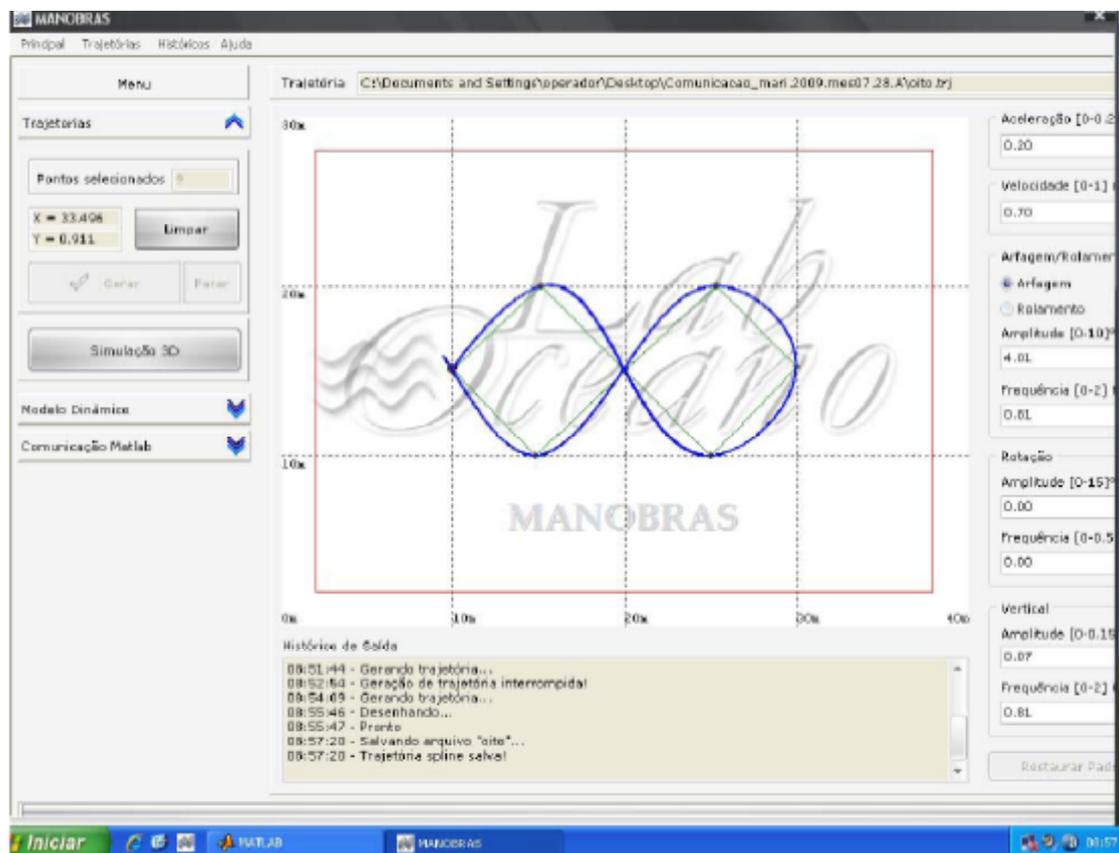


Figura 1. Geração de uma trajetória de referência na interface do usuário.

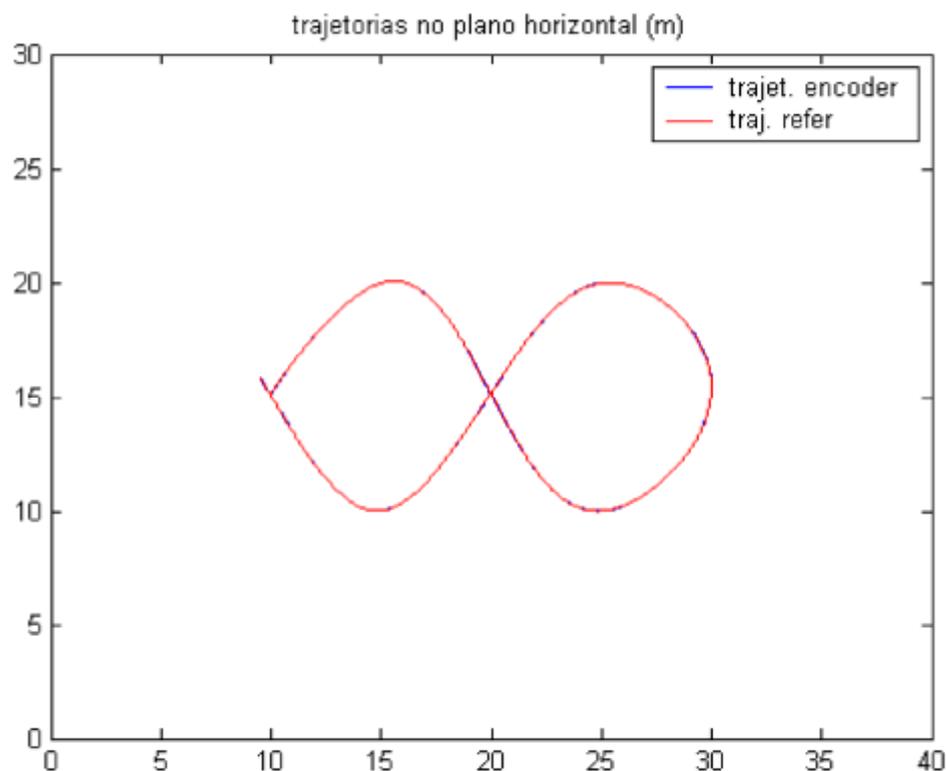


Figura 2. Trajetória de referência e trajetória de saída (sinais dos encoders) no plano horizontal.

## 7. Conclusões

O projeto visa mostrar a atuação de leis de controle em modelos reais, usando para isso uma placa FPGA, que é um dispositivo embarcado com um grande poder de processamento e que deve contribuir bastante para a eficiência do sistema. Em síntese, os resultados apresentaram boa robustez quanto à estabilidade (privilegiada pelo controle), sem, no entanto deixar de apresentar um bom desempenho de uma forma geral. Experimentos foram repetidos por diversas vezes, tendo-se atestado que o sistema de controle consegue manter um excelente padrão de repetição dos resultados.

## 8. Referências

- [1] Inc. Digilent, “Virtex II Pro Development System”, Mar. 2009;  
<http://www.digilentinc.com/Products/Detail.cfm?Nav1=Products&Nav2=Programmable&Prod=X UPV2P>.
- [2] Levi, C.A.; Pinto, W.T.; Gomes, S.C.P.; Sales Jr, J.S.; Costa, P.R.; Rosa, V.S.; and Gervini, V.I. A maneuvering test platform conceptual design. In: 26th International Conference on Offshore Mechanics and Artic Engineering, 2007, San Diego, CA, OMAE 2007.
- [3] X. Shao, and D. Sun, “A FPGA-based Motion Control IC Design”, *Proceedings of IEEE International Conference on Industrial Technology (ICIT 2005)*, Hong Kong, December, 2005.
- [4] C. Hackney, “FPGA Motor Control Reference Design”, Xilinx Application Note: Spartan and Virtex FPGA Families (XAP 808 v1.0), 2005.