

CONTROLADORES PREDITIVOS GPC PARA UM SISTEMA TURBINA-GERADOR: TESTES EM COSIMULAÇÃO E FPGA-IN-THE-LOOP

MÁRCIO A. S. GRASSI, EDSON A. BATISTA, CRISTIANO Q. ANDREA, RAYMUNDO C. GARCÍA, EDNILSO S. DO NASCIMENTO

*Laboratório de Sistemas Embarcados LABSEM, Departamento de Engenharia Elétrica, Universidade Federal de Mato Grosso do Sul UFMS
Caixa Postal 79070-900, Campo Grande, MS*

*E-mails: marciograssil4@gmail.com, edson.batista@ufms.br,
quevedo_unesp@yahoo.com.br, rcorderog@gmail.com, ednilso@hotmail.com*

Abstract— This paper presents results about the hardware implementation of a Model-Based Predictive Controller (MPC), with and without state observer, for a generator turbine system, using VHDL language and FPGA. The objective of this study are the implementation of predictive control in FPGA, and the development of tests using cosimulation (the process is simulated in the Simulink environment, while the controller implemented is in the Modelsim), and FPGA-in-the-loop (similar to cosimulation, but the controller is implemented in FPGA). The digital implementation of MPC is a difficult task due to the samples of the state and output variables of the plant are delayed respect to the estimation of the control law and the mathematical operations required in MPC. In order to solve this delay problem, an event counter was added to the MPC. Thus, it is possible to provide to the control algorithm the adequate values of the current and past state variables required for the calculation of the control law. In relation to matrix operations, Altera's specific library adaptations were developed. The results show that the MPC control law, calculated in VHDL, is similar to the obtained one using MATLAB.

Keywords— Generalized MPC, MPC with state observer, VHDL, FPGA-in-the-loop, cosimulation

Resumo— Este artigo apresenta resultados da implementação em hardware de um Controlador Preditivo baseado em Modelo (MPC) com e sem observador de estado, para um sistema turbina gerador, usando linguagem VHDL sintetizados em FPGA. O objetivo deste trabalho é a implementação de controle preditivo em FPGA, e realizar testes usando cosimulação (o processo é simulado no ambiente Simulink, enquanto que o controlador é implementado no programa Modelsim), e FPGA-in-the-loop (semelhante à cosimulação, mas o controlador é embarcado em FPGA). A implementação digital do MPC é uma tarefa difícil, pois as amostras das variáveis de estado e de saída da planta estão atrasadas em relação a estimação da lei de controle e os cálculos matemáticos necessários no MPC. Para solucionar o problema de atraso, adicionou-se um contador de eventos ao MPC. Assim, é possível fornecer no tempo correto os valores das variáveis de estados, atuais e passadas, necessários para o cálculo da lei de controle. Em relação às operações matriciais desenvolveram-se adequações de bibliotecas específicas da Altera. Os resultados mostram que a lei de controle do MPC, calculada em VHDL, é semelhante ao obtido em MATLAB.

Palavras-chave— MPC generalizado, MPC com observador de estado, VHDL, FPGA-in-the-loop, cosimulação

1 Introdução

O Controle Preditivo baseado em Modelo (MPC-Model Predictive Control) é uma técnica de controle que tem a atenção da indústria, não só pela habilidade de entregar um controlador confiável, mas também por permitir a utilização de restrições em relação aos parâmetros a serem controlados. De acordo com (Wang, 2009) os engenheiros industriais preferem utilizar controladores robustos, transparentes e que sejam de fácil implementação. Segundo (Rossiter, 2004), o MPC possui características que o tornam uma abordagem específicas, dentre as quais pode-se citar: conceitos intuitivos, fáceis de entender e implementar para uma variedade de sistemas; capacidade de lidar com sistemas MIMO (Multiple Input Multiple Output); e por fim, praticidade de lidar com mudanças dinâmicas.

Inicialmente o MPC foi adotado em processos lentos como sistemas petroquímicos, mas recentemente vem sendo desenvolvido para outras áreas como: controle de turbina eólica (Morsi, Abbas and Mohamed, 2017), aplicação em veículos elétricos e *smart grid* (Elsisi et al, 2017), controle de motores e

inversores (Wang et al, 2017), robótica (He et al, 2017) e controle de reator nuclear (Na et al, 2006).

O objetivo deste trabalho é desenvolver um algoritmo de controle preditivo por meio da linguagem VHDL e aplicá-lo em um estudo de caso de uma planta que será simulada pela ferramenta SIMULINK do programa MATLAB da Mathworks. Durante a cosimulação, o controlador foi simulado no programa MODELSIM PE da empresa Mentor-Graphics, e em seguida foi embarcado em uma plataforma FPGA do kit de desenvolvimento da Altera DE2-115 e testado pelo método FPGA-in-the-loop. Além disso, outro ponto explorado neste trabalho foi a implementação de diferentes algoritmos de MPC, sendo o modelo tradicional e com observador de estados.

Este artigo está dividido da seguinte forma: na Seção 2 mostra-se as equações referente a abordagem utilizada para o projeto do MPC com ou sem observador de estado. Na Seção 3 é feita uma descrição da metodologia de implementação do MPC em VHDL. Na Seção 4 apresenta os resultados e discussões das simulações e a Seção 5 trata das conclusões e dos possíveis trabalhos futuros.

2 Controle Preditivo baseado em Modelo - MPC

É importante destacar que o controle preditivo possui diversas técnicas de projeto e cabe ao projetista determinar a melhor abordagem para obter o resultado desejado. Neste artigo a forma do MPC projetado segue o modelo proposto em (Wang, 2009).

De acordo com (Wang, 2009) foi utilizado a modelagem em espaço de estado, no qual ocorre a adição de um integrador. As variáveis de estado são determinadas pela variação dos estados e a saída é adicionada como um estado. Na equação 1 mostra-se a representação do modelo matemático para MPC com os parâmetros da planta.

Sendo que: A, B e C são chamadas de matrizes aumentadas do sistema, $x(k+1)$ são os próximos estados, $x(k)$ são os estados atuais, $\Delta u(k)$ a variável de controle, $o_m = [0 \ 0 \dots 0]$ sendo n_1 a dimensão do vetor de espaço de estado e $y(k)$ a saída do sistema.

$$\begin{bmatrix} \Delta x_m(k+1) \\ y_m(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (1)$$

$$y(k) = \begin{bmatrix} c \\ o_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

A saída do algoritmo de controle preditivo é dado pela equação 2 (Wang, 2009).

$$Y = Fx(ki) + \phi \Delta U \quad (2)$$

As matrizes ϕ e F são calculadas com as matrizes aumentadas do sistema e a dimensão delas depende do horizonte de predição (NP) e do horizonte de controle (NC). As definições de F e ϕ são apresentadas nas equações 3 e 4, respectivamente.

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{NP} \end{bmatrix} \quad (3)$$

$$\phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{NP-1}B & CA^{NP-2}B & \dots & CA^{NP-NC}B \end{bmatrix} \quad (4)$$

2.1 Função de Otimização

O MPC se baseia na minimização de uma função de custo. A função utilizada neste projeto é apresentada na equação 5:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (5)$$

Sendo R_s um vetor que contém a informação do set-point $R_s^T = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} r(k_i)$, $r(k_i)$ é o set-point. O parâmetro \bar{R} é uma matriz diagonal com a

informação do peso r_w : $\bar{R} = r_w I_{NC \times NC}$. O r_w penaliza a variação na lei de controle (ΔU), para que a mesma não seja divergente.

A otimização da função de custo é simplesmente derivar (2) e igualá-la a zero.

Após a otimização da função custo, define-se os ganhos do MPC, sendo, um ganho para a realimentação dos estados (K_x) e outro para a realimentação do erro (K_y). Ambos os ganhos resultam da equação 7 que calcula o K_{mpc} geral do MPC. As equações 6 e 7 mostram o cálculo dos ganhos supracitados.

$$K_{mpc} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} (\phi^T \phi + r_w I_{NC})^{-1} (\phi^T F) \quad (6)$$

$$K_{mpc} = \begin{bmatrix} K_x & K_y \end{bmatrix} \quad (7)$$

Sendo I_{NC} a matriz identidade quadrada com a dimensão dependente do horizonte de controle (NC).

2.2 Lei de controle do MPC generalizada

Com os valores dos ganhos a lei de controle é então calculada, a cada amostra, adicionando-se o valor de $\Delta u(k_i)$, visto na equação 8, ao valor anterior de $u(k_i)$ como mostra o diagrama de blocos da Figura 1.

$$\Delta u(k_i) = K_y (r(k_i) - y(k_i)) - K_x \Delta x(k_i) \quad (8)$$

A inclusão do observador de estados no modelo preditivo é definido pela equação 9.

$$\hat{x}(k_i + 1) = A\hat{x}(k_i) + B\Delta u(k_i) + K_{ob} (y(k_i) - C\hat{x}(k_i)) \quad (9)$$

A definição da lei de controle do MPC com observadores de estado é apresentada na equação 10.

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} \hat{x}(k_i) \quad (10)$$

A utilização de observador de estados é atrativa, pois em algumas aplicações industriais não tem acesso as variáveis de estados, seja por motivo da complexidade do processo ou por uma redução de custo.

Nas Figuras 1 e 2 mostram um diagrama de blocos para o MPC com e sem observador.

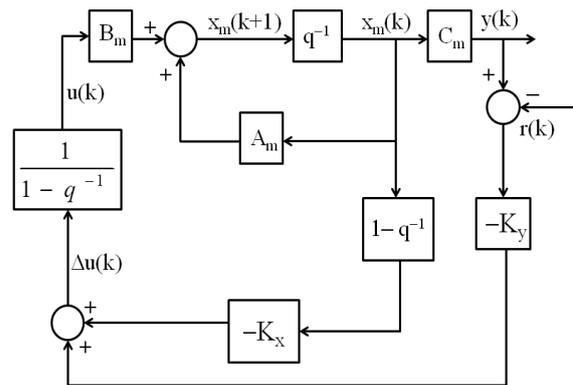


Figura 1 - Bloco diagrama do MPC, adaptado de (Wang, 2009).

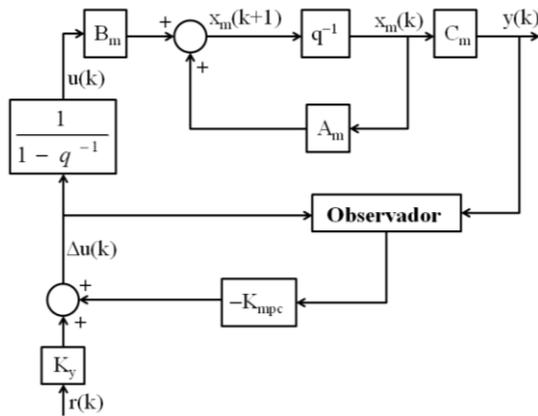


Figura 2 - Bloco diagrama do MPC com observador, adaptado de (Wang, 2009).

3 Metodologia

A implementação do MPC utilizou a aritmética de ponto fixo para sintetizar o circuito digital e, todas as variáveis utilizaram 15 bits para a parte inteira, 16 bits para a parte fracionária e 1 bit para representação do sinal, pois estes limites são padronizados na biblioteca (s-fixed) específica da Altera. As matrizes e fórmulas vistas neste artigo foram incorporadas ao código VHDL para obtenção do MPC. O algoritmo de controle preditivo implementado em VHDL neste trabalho é dinâmico, ou seja, o projetista pode alterar os parâmetros de NP, NC e do peso r_w e o controlador é respectivamente atualizado. Este dinamismo do algoritmo tornam flexíveis os testes em hardware do controlador MPC generalizado.

Na subseção 3.1 é mostrado como o controlador foi projetado e na 3.2 a planta a ser controlada.

3.1 Implementação do controlador

Um dos desafios da implementação do MPC em meio digital é o atraso de dados dos estados e da saída que ocorrem durante a borda de descida do clock, o que prejudica no projeto do MPC com circuitos digitais. Na Figura 3 mostra-se a simulação dos estados sendo amostrados com o atraso. A Figura 3 foi desenvolvida no programa ModelSim PE sendo: o sinal *clk* representa o clock do circuito, *rst* o sinal de reset que está em nível lógico alto para o controle funcionar, os estados são representados pelos sinais *xm1*, *xm2* e *xm3*, a saída está no sinal *Yout* (neste artigo será a velocidade do sistema), e a referência é vista no sinal *rki*.

clk					
rst					
Yout	0		3,92364		
rki	10				
xm1	0		0,0385138		
xm2	0		0,033826		
xm3	0		0,00784728		

Figura 3 - Atualização dos estados com atraso

Neste artigo implementou-se diferentes projetos de MPC, sendo que na Figura 4 mostra-se o controlador com o algoritmo generalizado. Basicamente, ele é composto por 3 componentes, em que cada um deles realiza uma tarefa específica no algoritmo do MPC. O primeiro bloco de nome *MPC_gain* tem como entrada as matrizes do modelo da planta a ser controlada. Este bloco é sintetizado apenas com circuitos combinacionais, e como saída fornece os ganhos da realimentação e do integrador, conforme as equações 3 e 4.

O segundo bloco de nome *Clock_cont* é responsável por calcular o tempo e emitir um sinal (*U_enable*) de um bit que permita a lei de controle a ser calculada pelo terceiro bloco. Além disso, este bloco também tem a função de gravar o sinal de controle e dos estados que serão necessários para calcular a próxima lei de controle.

Por fim, o terceiro bloco de nome *Control_Law* é o bloco responsável por calcular a lei de controle, em que os ganhos (K_x e K_y), a referência $r(k_i)$, a saída (*Yout*) e outros sinais necessários para calcular a equação 8 no MPC tradicional e 10 no MPC com observador.

O MPC com observador de estados possui os mesmos blocos, porém como os estados são estimados, não haverá mais os sinais $X(k)$. Também será necessário mandar as matrizes aumentadas para cálculo da lei de controle, já que esta depende dos estados estimados. Além disso, o bloco *Clock_cont* passa a ter a função de fornecer a lei de controle e os estados observados do passo anterior. O esquemático do VHDL com observador de estado é apresentado na Figura 5.

3.2 Planta controlada

O sistema selecionado para ser controlado é uma planta descrita em (Nise, 2002), como mostra-se na Figura 6. No sistema de controle apresentado, propõe-se o controle de frequência de saída de energia elétrica de um sistema turbina-gerador. Regulando a velocidade, o sistema de controle assegura que a frequência gerada permaneça dentro da tolerância. Os desvios a partir da velocidade desejada são medidos e uma válvula de vapor é alterada para compensar o erro de velocidade.

Uma simulação em malha aberta mostra que esta planta se estabiliza em 4,42 segundos, o que nos indica que um possível período de amostragem de 0,442 segundos, pois precisa-se de 10 pontos do transitório. Para este artigo o período de amostragem foi ajustado em 0,442 segundos. Na Figura 7 observa-se o resultado da simulação em malha aberta.

O tutorial do MPC com MATLAB disponível em (Wang, 2009) foi utilizado para base de comparação com o resultado do MPC em VHDL. O valor do peso (r_w) será fixo em zero e o NP será variado. Os ganhos do observador também foram obtidos de um tutorial de (Wang, 2009) cabendo ao projetista a alocação dos pólos do observador. Neste artigo os pólos foram 0,001 0,00105 0,0011 0,00108.

4 Resultados

Na primeira análise, comparamos a quantidade de elementos lógicos (ELs) demandada pelo MPC e a diferença de usar ou não o observador de estados. Com NC fixo em 1 e o peso em zero, a Tabela 1 mostra os dados de tempo de compilação e elementos lógicos utilizados pelo código em VHDL do MPC. Na Tabela 2 informa-se os mesmos dados para o MPC com observador.

Com os dados das tabelas 1 e 2 percebe-se que o observador de estados ocupa mais ELs, pois é necessário armazenar as matrizes aumentadas para calcular a lei de controle e os estados estimados. Os cenários vistos nas tabelas 1 e 2 foram testados e comparados com o tutorial, com exceção do NP = 10. Na Figura 8 mostra-se a resposta do MPC sem observador para NP = 2 gerado por meio do *script* do MATLAB (Wang, 2009). Na Figura 9 apresenta-se a resposta do MPC sem observador em VHDL para NP = 2 testado em cosimulação.

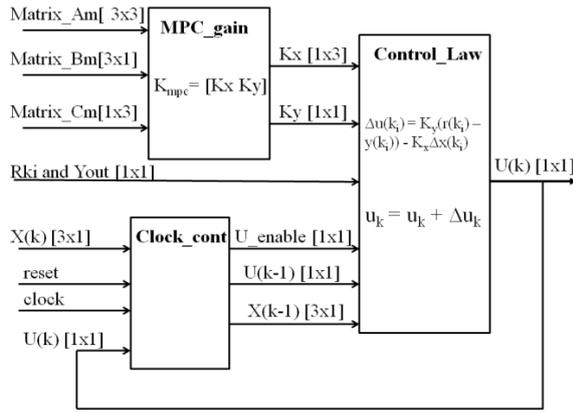


Figura 4 - Esquemático do MPC em VHDL

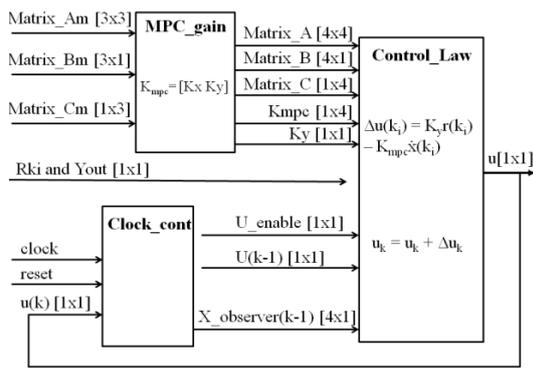


Figura 5 - Esquemático do MPC com observador.

Tabela 1. Elementos lógicos utilizados pelo MPC.

	NP = 2	NP = 5	NP = 10
ELs	3103(3%)	3772(3%)	4,619(4%)
Tempo	3'51"	6'07"	9'19"

Tabela 2. Elementos lógicos utilizados pelo MPC com observador.

	NP = 2	NP=5	NP=10
ELs	7325(6%)	7997(7%)	8910(8%)
Tempo	5'50"	7'52"	10'39"

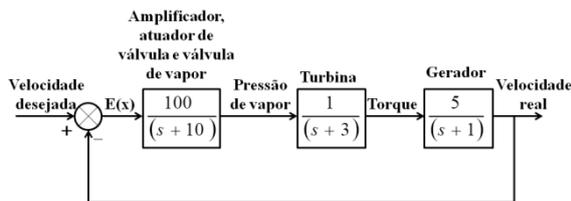


Figura 6 - Processo a ser controlado pelo MPC, adaptado de (Nise, 2002)

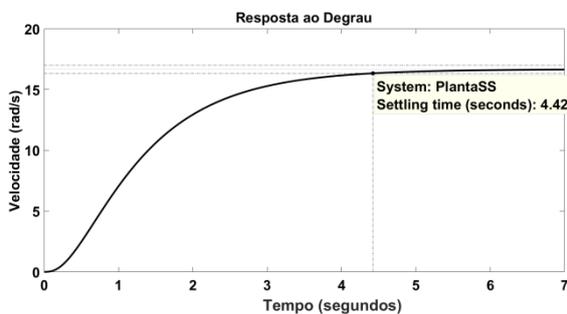


Figura 7 - Simulação em malha aberta da planta

O problema de atraso da aplicação da lei de controle foi relatado em (Sandre-Hernandez et al, 2015). Neste projeto almejamos validar o algoritmo de controle preditivo em VHDL para diferentes NP.

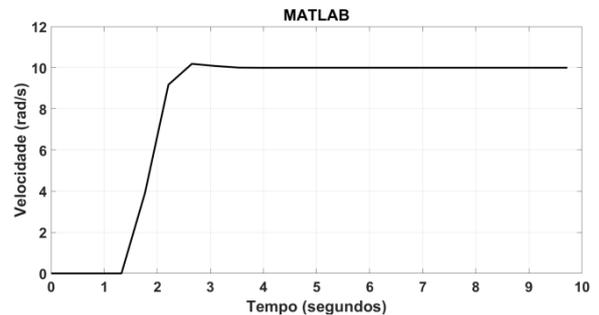


Figura 8 - Resposta do MPC nos diferentes testes para NP = 2.

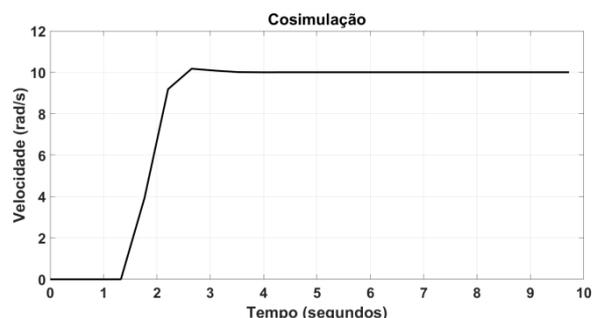


Figura 9 - Resposta do MPC nos diferentes testes para NP = 2.

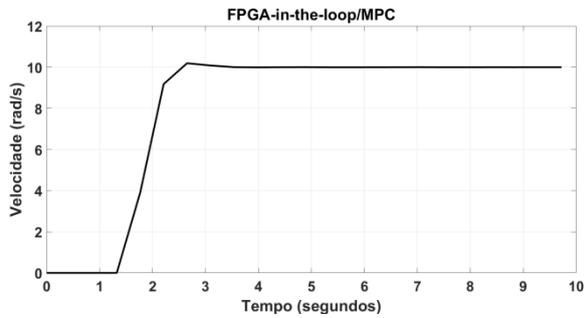


Figura 10 - Resposta do MPC nos diferentes testes para NP = 2.

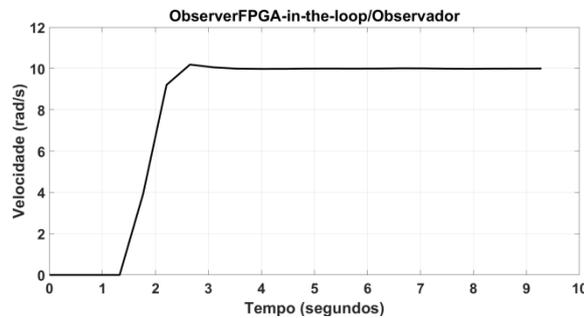


Figura 11 - Resposta do MPC nos diferentes testes para NP = 2.

Na Figura 10 mostra-se a resposta do MPC sem observador para NP = 2 e testado em FPGA-in-the-loop. Na Figura 11 demonstra-se a resposta do MPC com observador de estados para NP = 2 e testado em FPGA-in-the-loop.

Apesar de haver alguma diferença numérica entre os quatro ensaios, o overshoot ficou em torno de 1,9% e o tempo de assentamento menor que 2 segundos. Esta diferença ocorre porque o ensaio com o *script* do MATLAB e a cosimulação utilizam números reais e os ensaios com FPGA-in-the-loop utiliza aritmética de ponto fixo. A aritmética de ponto fixo os dados são representados com 16 bits para a parte inteira sendo 1 bit para o sinal e 16 bits para a parte fracionária, o que ocorre uma limitação nos dados. As formas de onda ficaram bem semelhantes o que comprova a validação do projeto. Na Figura 12 mostra-se a resposta do MPC sem observador gerado por meio do *script* do MATLAB (Wang, 2009) para NP = 5. Na Figura 13 apresenta-se a resposta do MPC em VHDL sem observador de estados para NP = 5 testado em cosimulação.

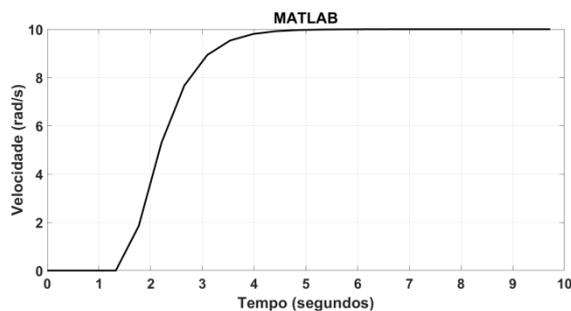


Figura 12 - Resposta do MPC nos diferentes testes para NP = 5.

Na Figura 14 apresenta-se a resposta do MPC para NP = 5 testado em FPGA-in-the-loop. Na Figura

15 apresenta-se a resposta do MPC com observador de estados para NP = 5 testado em FPGA-in-the-loop.

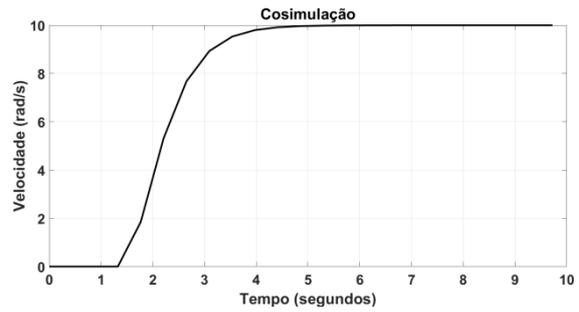


Figura 13 - Resposta do MPC nos diferentes testes para NP = 5.

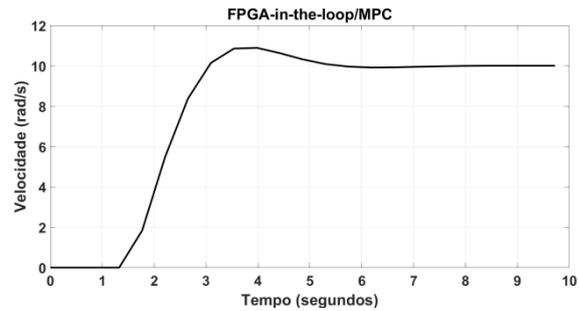


Figura 14 - Resposta do MPC nos diferentes testes para NP = 5.

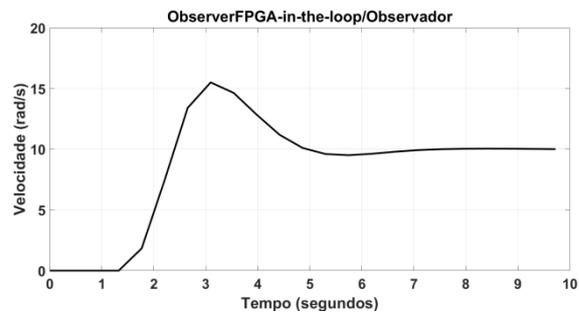


Figura 15 - Resposta do MPC nos diferentes testes para NP = 5.

A simulação em hardware do MPC tradicional e com observador obteve uma resposta diferente da esperada. Enquanto na cosimulação a resposta foi de segunda ordem sem overshoot, no ensaio com FPGA-in-the-loop do MPC generalizado ocorreu um overshoot de 8,94%, e com o observador de 55%, com tempo de estabilização pouco maior que 3 segundos. Isto ocorreu porque com um valor maior de NP, números muito grandes aparecem nas matrizes F e ϕ . Quando o MPC é embarcado na FPGA os dados são representados com limite de 16 bits para a parte inteira e fracionária, e por isto, nem sempre esses valores grandes são representados com precisão. Para estes ensaios os ganhos calculados estiveram próximos dos valores esperados com exceção do ganho do terceiro estado onde o esperado era 96.7572 e o embarcado em hardware foi igual a 71. Isto prejudicou o desempenho do controle, mas ainda assim o MPC conseguiu fazer o rastreamento.

Na Figura 16 mostra-se o resultado do teste utilizando FPGA-in-the-loop para MPC generalizado.

Sendo que em *a* é a conexão USB para carregar o hardware na placa e realizar a comunicação com o ambiente SIMULINK, *b* é a placa Altera DE2-115, *c* é a planta com o respectivo controlador preditivo, *d* é a resposta da simulação.

Na Figura 17 mostra-se o resultado do teste utilizando FPGA-in-the-loop para MPC com observador de estados. Sendo que em *a* é a planta e o controlador preditivo com o observador e *b* é a resposta da simulação. A comunicação e a placa são a mesma do ensaio anterior.

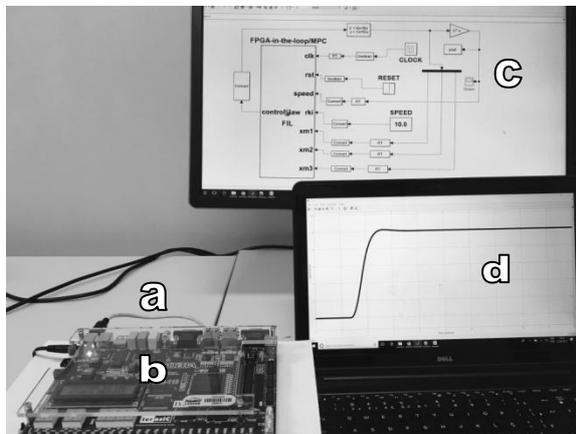


Figura 16 - Teste MPC generalizado FPGA-in-the-loop

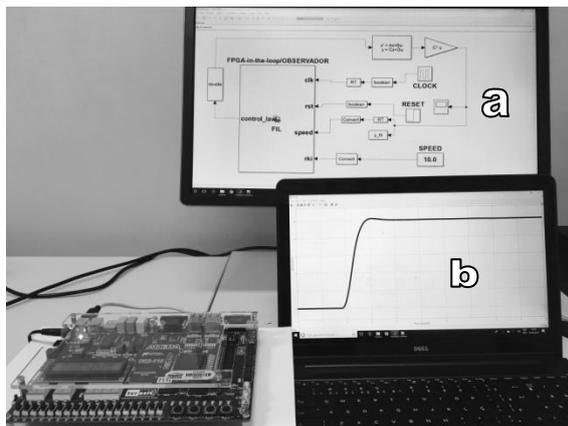


Figura 17 - Teste MPC com observador FPGA-in-the-loop

5 Conclusão

Com base nos resultados apresentados conclui-se que o controle preditivo implementado em linguagem VHDL foi validado com o algoritmo generalizado e com observador de estado. No projeto destes controladores desenvolveu-se uma plataforma que permite ao projetista modificar os valores de NP e NC e automaticamente o hardware se reconfigura. Outra contribuição deste trabalho está na exploração de testes utilizando os recursos de FPGA-in-the-loop. Como trabalho futuro almeja-se testar esta abordagem em plantas não lineares com múltiplas entradas e saídas e também outros algoritmos mais avançados de modelo preditivo.

Agradecimentos

Os autores agradecem a CAPES pelo apoio para realização deste trabalho.

Referências Bibliográficas

- Elsisi, M., Soliman, M., Aboeela, M. A. S., Mansour, W. (2017). Model Predictive Control of Plug-in Hybrid Electric Vehicles For Frequency Regulation in a Smart Grid. *IET Generation, Transmission & Distribution*, 11(16), pp. 3974-3983.
- He, F., Bi, L., Lu, Y., Li, H., and Wang, L. (2017). Model Predictive Control for a Brain-controlled Mobile Robot. In: *IEEE International Conference on Systems, Man, and Cybernetics(SMC)*. Banff: pp. 3184 - 3188.
- Lucia, S., Navarro, D., Lucia, O., Zometa, P., Findeisen, R. (2018). Optimized FPGA Implementation of Model Predictive Control for Embedded Systems Using High Level Synthesis Tool. *IEEE Transactions on Industrial Informatics*, 14(1), pp. 137-145.
- Morsi, A., Abbas, H. S., Mohamed, A. M. (2017). Wind turbine control based on a modified model predictive control scheme for linear parameter-varying systems. *IET Control Theory & Applications*, 11(17), pp. 3056-3068.
- Na, M. G., Upadhyaya B. R., Xu, X., Hwang, I. J. (2006). Design of a Model Predictive Power Controller for an SP-100Space Reactor. *Nuclear Science and Engineering*, 154(3), pp. 353-366.
- Nise, N. S. (2002). *Engenharia de Sistemas de Controle*. 3ª edição. Rio de Janeiro: LTC.
- Rossiter, J. A. (2004). *Model-based Predictive Control: a practical approach*. Boca Raton: CRC Press.
- Sandre-Hernandez, O., Rangel-Magdaleno, J. J., Tlelo-Cuatle, E., Morales-Caporal, R. (2015). FPGA-based Delay Compensation on Model Predictive Control for a PM Synchronous Machine. *IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*.
- Wang, F., Mei, X., Rodríguez, J., Kennel, R. (2017). Model Predictive Control for Electrical Drive Systems-An Overview. *CES Transactions on Electrical Machines and Systems*, 1(3), pp. 219-230.
- Wang, L. (2009). *Model Predictive Control System Design and Implementation Using Matlab®*. London: Springer.