

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Plataforma para Simulação em Tempo Real de Sistemas Dinâmicos e Hardware-in-the-loop em Sistemas Embarcados do tipo PSOC

Isaac D. T. de Souza¹, Sérgio N. Silva² e Marcelo A. C. Fernandes³

Departamento de Engenharia da Computação e Automação – DCA

Universidade Federal do Rio Grande do Norte – UFRN, Natal, RN, Brasil

Resumo. Testes em tempo real são uma etapa fundamental no desenvolvimento de aplicações embarcadas para automação e controle de processos industriais. Apesar disso, nem sempre esses testes são viáveis devido aos seus elevados custos. Para reduzir esses custos, um tipo de simulação conhecida como HIL (*Hardware-in-the-loop*) pode ser utilizada. Nesse tipo de simulação, a planta real é substituída por um modelo computacional para testes preliminares, de forma que os resultados desses testes possam ser obtidos mais rapidamente. Dessa forma, objetiva-se, com esse trabalho, a definição de uma plataforma embarcada em plataforma de hardware do tipo PSOC (*Programmable System on Chip*) para simulação de sistemas dinâmicos em tempo real. É possível aplicar o simulador em tempo real tanto no ambiente acadêmico, como na indústria.

Palavras-chave. Tempo Real, Simulação, Sistemas Dinâmicos, Sistemas Embarcados, PSOC.

1 Introdução

O processo que envolve a criação de novos algoritmos e técnicas de controle de sistemas envolve um alto custo de tempo e recursos econômicos. Por exemplo, para o teste em tempo real de um novo algoritmo de controle da suspensão de um carro, seria necessária a criação de um modelo real, as vezes em escala menor, do sistema de interesse do carro. Os resultados desses testes poderiam ser obtidos primordialmente utilizando-se um Simulador em Tempo Real (STR) do sistema, isto é, um dispositivo com entradas e saídas de sinais capaz de emular a reação do carro aos sinais de controle recebidos.

No *tunning* de um dispositivo de hardware controlador para uma determinada aplicação, os testes desses algoritmos nas plantas reais podem ter custos demasiadamente proibitivos. Além disso, testar dispositivos de controle exige uma série de testes com diversas configurações que podem ser difíceis atingir com a planta em um laboratório comum. Para

¹isaacdiego@gmail.com

²sergionatan@dca.ufrn.br

³mfernandes@dca.ufrn.br

driblar alguns desses problemas, foi criado um tipo de teste que não utiliza a planta real, mas um modelo computacional da mesma, a simulação (ou teste) HIL (*Hardware-in-the-loop*) [3, 4]. Utilizando as mesmas interfaces de hardware que o controlador usaria para interagir com a planta, este trabalha como se estivesse interagindo com a planta real, sem alterações muito bruscas nos seus parâmetros internos. Simulação HIL se caracteriza pela operação de componentes reais em conjunto com elementos simulado. Por exemplo, o desenvolvedor pode desejar fazer o *tunning* do dispositivo simulando somente a planta e os atuadores, mas mantendo os sensores reais interagindo com o modelo matemático.

A pesquisa envolvendo simulação em tempo real é um campo amplo que envolve não só o estudo de novas propostas de hardware e software, mas também novas técnicas de modelagem para sistemas comerciais existentes. Isso mostra que ainda existe espaço para novas pesquisas, especialmente na área de simulações HIL. Dessa forma, o propósito desse trabalho é apresentar uma plataforma de simulação em tempo real e HIL para otimização e teste de algoritmos de controle. Essa plataforma, chamada de Simulador em Tempo Real de Sistemas Dinâmicos embarcada em PSOC (*Programmable System on Chip*) (STRSD-PSOC), é baseada na proposta apresentada em [1] que define uma plataforma distribuída e de baixo custo para STR. Todavia, diferentemente das plataformas propostas em [1, 2, 5, 6], este trabalho apresenta um sistema de STR embarcado em um único elemento de hardware do tipo PSOC.

A principal meta da plataforma é prover uma ferramenta para desenvolvedores de sistemas de controle embarcados focada em reduzir os custos envolvidos no processo de teste em tempo real. No âmbito acadêmico, o STRSD-PSOC pode ser usado como uma planta didática programável em disciplinas de cursos de engenharia. Além disso, usando o STRSD-PSOC, um estudante de graduação ou pós-graduação poderá testar algoritmos de controle em diversos sistemas dinâmicos sem a necessidade de ter uma planta didática real. Outra importante característica do STRSD-PSOC é que ele é capaz de simular o funcionamento de *drivers* e sensores associados às plantas simuladas, o que é fundamental no desenvolvimento de algoritmos que precisem ser embarcados para fins práticos.

2 Arquitetura do STRSD-PSOC

A Figura 1 apresenta um resumo da arquitetura física proposta para a plataforma. Nela existem três tipos de componentes: o Módulo de Controle Principal (MCP), o Módulo de Aquisição de Dados (MAD) e o Módulo de Geração de Sinais (MGS). Um STRSD-PSOC pode ter M MADs e N MGSs, mas somente um MCP. Diferentemente do SRT apresentado em [1], o MCP não possui periféricos externos acoplados a ele pois o hardware PSOC permite integrar em um único chip vários dispositivos como ADC's, DAC's, geradores PWM e outros.

O MCP é o centro de processamento do simulador. Nele, os dados de entrada (variáveis de controle) são processados e, de acordo com modelo matemático do sistema a ser simulado contido nele, são calculadas as saídas do sistemas (variáveis de estado). Um MCP tem M MADs associados a ele e N MGSs. Um MAD é um componente responsável por traduzir os sinais de controle vindos de um ou mais dispositivos externos. Um MAD tem

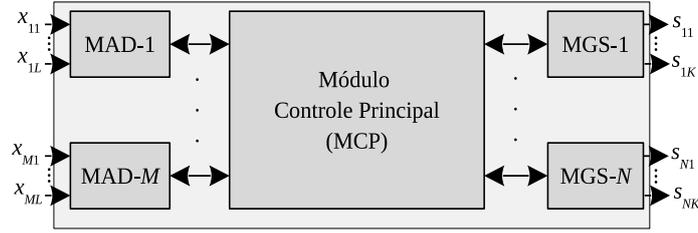


Figura 1: Arquitetura física da plataforma STRSD.

L sinais de controle associados a ele. O MGS é um componente responsável por traduzir as quantidade numéricas relacionadas às variáveis de processo em sinais elétricos capazes de serem entendidos por dispositivos controladores como níveis de tensão, PWM, ASCII via conexão serial e etc. De forma a garantir os requisitos de tempo real a comunicação entre o MCP e o MGS deve ser determinística. Um MGS tem K sinais saída associados a ele.

O tempo de amostragem t_a do simulador pode ser definido por:

$$t_a \geq t_{MAD} + t_1 + t_{AD} + t_s + t_2 + t_{MGS}, \quad (1)$$

em que t_{MAD} é o tempo de processamento mais lento de todos os MADs, t_1 é o tempo correspondente à menor taxa de transferência de informação entre os MADs e o MCP, t_{AD} é o tempo máximo em que todas os sinais de entrada são transferidos para a memória do MCP, t_s é o tempo máximo para a execução da resolução da equações diferenciais ordinárias (EDOs), t_2 é o tempo máximo de comunicação entre o MCP e todos os MGS, e t_{MGS} é o maior tempo necessário para que o processamento das saídas ocorra em todos os MGSs.

3 Protótipo Desenvolvido

O protótipo criado para validação da plataforma descrita nesse artigo foi construído utilizando um kit de desenvolvimento PSoC com um microcontrolador tipo 5 (ARM Cortex-M3) acoplado a ele. De forma a manter todos os componentes da arquitetura em um mesmo chip, os módulos de aquisição de dados e de geração de sinais foram implementados por módulos internos da plataforma PSoC. Esse componentes devem ser definidos uma vez no momento da compilação do sistema a ser gravado na placa. O sistema simulado foi o modelo longitudinal de um veículo, como ilustrado na Figura 2. Nesse modelo, o torque nas rodas e a inclinação longitudinal da pista são os parâmetros de entrada. O parâmetro de saída é a velocidade do veículo.

A dinâmica do veículo pode ser expressa como

$$M \frac{dx(t)}{dt} = f_t(t) - f_a(t), \quad (2)$$

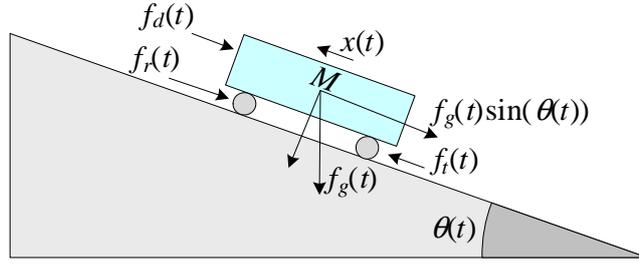


Figura 2: Esquemático do modelo longitudinal do veículo.

em que, M é a massa do veículo (em Kg), $x(t)$ é a velocidade linear do veículo (em m/s), $f_t(t)$ é a força de tração exercida pelo veículo (em N) e $f_a(t)$ é a força de atrito (em N). O torque pode ser expressa por

$$f_t(t) = \frac{\tau_m(t)}{r} \quad (3)$$

em que r é o raio da roda do veículo (em m) e $\tau_m(t)$ é o torque (em Nm) gerado pelo motor. A força de atrito, $f_a(t)$, pode ser expressa por

$$f_a(t) = f_d(t) + f_r(t) + f_g(t) \sin(\theta(t)), \quad (4)$$

onde $f_d(t)$ é a força de atrito aerodinâmico (em N), $f_r(t)$ é a força de resistência devido o rolamento (em N), $f_g(t)$ é a força gravitacional (em N), e $\theta(t)$ é o ângulo de inclinação do plano em que o veículo se encontra. O atrito aerodinâmico pode ser descrito como

$$f_d(t) = \frac{1}{2} \rho C_d A_{fr} x^2(t), \quad (5)$$

em que ρ é a densidade do ar, C_d é o coeficiente de arrasto aerodinâmico, e A_{fr} é a área frontal do veículo (em m^2). A força de resistência devido ao rolamento pode ser descrita por

$$f_r(t) = Mg (C_0 + C_1 x^2(t)), \quad (6)$$

em que C_0 e C_1 são os coeficientes de rolamento e g é a aceleração devido à gravidade (em m/s^2). Finalmente, a força gravitacional é dada por:

$$f_g(t) = Mg. \quad (7)$$

O sistema dinâmico foi simulado em tempo real, empregando o método de Euler para resolução de EDOs. A Figura 3 mostra os componentes utilizados na implementação, assim como a forma como eles foram interligados para formar o simulador.

Para o controle do acesso à informação no MCP, foi utilizado um componente temporizador (Timer). Esse foi configurado para gerar uma interrupção a cada 100 ms (t_a segundo a notação da Equação 1), com resolução de contagem de 8 bits. Esse tempo é mais do que suficiente para todo o processamento realizado na simulação nesse componente, dado que o tempo de processamento no pior caso (t_s na Equação 1) ficou em $330 \mu s$ (medido com o osciloscópio).

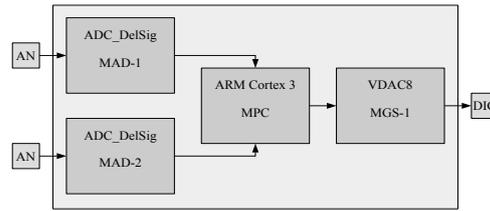


Figura 3: Conexões e componentes usados na implementação do STRSD na plataforma PSoC.

4 Resultados e Testes

Após o desenvolvimento do protótipo foram testados dois cenários de simulação em tempo real. No primeiro teste, o torque do veículo foi mantido em 3000 Nm e a inclinação da pista foi mantida em 0 graus. No segundo teste da série, foi mantida fixa em 0 a inclinação da pista. Já o torque começou em 500 Nm, mas foi alterado para 3000 Nm após 30 segundos.

Os resultados, apresentados nas Figuras 4 e 5, foram comparados com simulações em tempo não real realizadas no Matlab/Simulink. O erro médio, EM , entre a resposta do STRSD-PSOC e as respostas obtidas no Matlab/Simulink para ambos os cenários é apresentado na Tabela 1, que também apresenta os valores de variância desse erro. O valor do EM é definido por

$$EM = \frac{1}{N} \sum_{i=0}^{N-1} |X(i) - Y(i)| \quad (8)$$

onde, $X(i)$ é o valor da i -ésima amostra coletada do STRSD-PSOC e $Y(i)$ é o valor da simulação no Matlab/Simulink no mesmo i -ésimo instante. A variável N representa o número de amostras coletadas. Ambos referem-se ao valor da velocidade do veículo, $x(t)$.

O gráfico do comportamento do sistema para esse teste pode ser observado na Figura 4. Nesse gráfico, é possível notar que a subida se deu ao mesmo passo que a simulação em *Simulink*. Mais do que isso, é possível ver também que o sinal do simulador em hardware oscila em torno do implementado em software na subida, isso se dá pelas interferências e incertezas na captação do sinal, assim como pelo uso de conversor digital-analógico na saída, que gera um sinal analógico que muda de valor seguindo uma escala digital. Também por esse último motivo (e pelo fato de o sistema ter alcançado um valor próximo do limite interno definido para aumentar a resolução da saída), o sinal do simulador em hardware estagna em um valor um pouco inferior ao sinal do simulador em software.

O gráfico da Figura 5 traz os resultados no tempo dessa simulação. É possível notar concordância com um resultado real pois o tempo que sistema leva para alcançar a região de regime é maior para o primeiro caso em que o carro está menos acelerado. Também é possível notar que a diferença de valores entre as duas simulações é quase 0. Nesse caso,

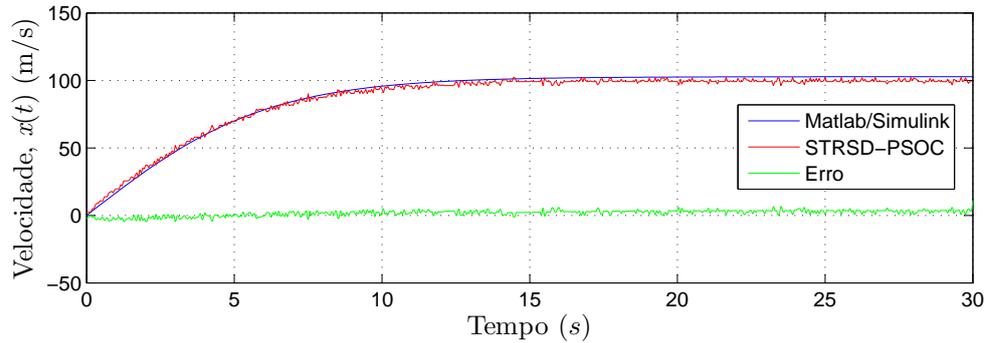


Figura 4: Resultado para a simulação para o torque e o ângulo de inclinação fixos (teste 1). Os resultados em vermelho foram obtidos com auxílio de um osciloscópio.

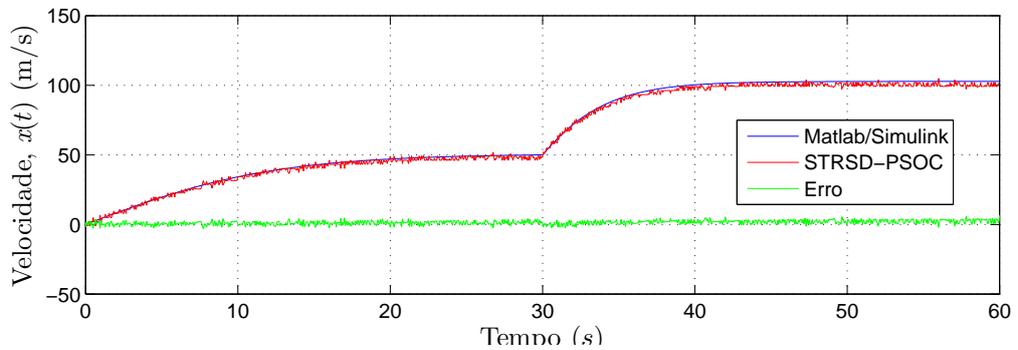


Figura 5: Resultado para a simulação para o torque variável e o ângulo de inclinação fixo (teste 2). Os resultados em vermelho foram obtidos com auxílio de um osciloscópio.

somente na segunda parte, é que se nota uma pequena diferença visível entre os dois. Isso se dá pelos motivos explicados anteriormente.

5 Conclusões

O trabalho apresentou uma implementação em hardware da plataforma de simulação em tempo real para aplicações em HIL, chamada de STRSD-PSOC. Para esta implementação, foi utilizada a plataforma de desenvolvimento de sistemas embarcados PSoC, com o microcontrolador tipo 5, um ARM Cortex M3. A implementação se mostrou muito prática pois a filosofia de hardware reconfigurável com componentes em formato de blocos do fabricante da plataforma PSoC foi benéfica na implementação, salvando bastante tempo de desenvolvimento que, em outras plataformas. Os resultados obtidos mostram, além de bastante concordância com a simulação do mesmo sistema em software, erros suportáveis

Cenários	EM	Variância
Cenário 1	1,7834	4,3031
Cenário 2	1,445	1,8727

Tabela 1: Medidas de Similaridade entre as Simulações para os cenários 1 e 2.

para a aplicação em questão. Futuros trabalhos visam um software para a programação automática da placa com o sistema e os componentes escolhidos interativamente pelo usuário.

Referências

- [1] I. D. T. de Souza, S. N. Silva, R. Teles, and M. A. C. Fernandes, “Platform for real-time simulation of dynamic systems and hardware-in-the-loop for control algorithms,” *Sensors*, vol. 14, no. 10, pp. 19 176–19 199, 2014.
- [2] V. Dixit, M. Patil, and M. Chandorkar, “Real time simulation of power electronic systems on multi-core processors,” in *Power Electronics and Drive Systems, 2009. PEDS 2009. International Conference on*, 2009, pp. 1524–1529.
- [3] M. Harakawa, H. Yamasaki, T. Nagano, A. Simon, C. Dufour, and J. Bélanger, “Real-time simulation of a complete pmsm drive at $10\mu s$ time step,” *Record of IPEC-Nigata*, 2005.
- [4] R. Isermann, J. Shaffnit, and S. Sinsel, “Hardware-in-the-loop simulation for the design and testing of engine-control systems,” *Control Engineering Practice*, 1999.
- [5] M. Keller and A. Kolb, “Real-time simulation of time-of-flight sensors,” *Simulation Modelling Practice and Theory*, vol. 17, no. 5, pp. 967 – 978, 2009.
- [6] M. Monga, M. Karkee, S. Sun, L. KiranTondehal, B. Steward, AtulKelkar, and J. Zambreno, “Real-time simulation of dynamic vehicle models using a high-performance reconfigurable platform,” *Procedia Computer Science*, vol. 9, no. 0, pp. 338 – 347, 2012.
- [7] H. B. Pacejka, “Tire and vehicle dynamics,” *Society of Automotive Engineers and Butterworth-Heinemann*, 2002.