

JAIC

Journal of Applied Instrumentation and Control

Ensinando Métodos de Discretização para Projeto de Sistemas de Controle: Comparação de Controladores PID em Sistemas Embarcados

Felipe C. Conceição, Wesley Beccaro, João F. Justo

Resumo— Diversas abordagens têm sido exploradas para ensinar os principais métodos de discretização de sistemas contínuos. Uma alternativa interessante para ensinar estes métodos pode ser realizada através do projeto de controladores Proporcional-Integral-Derivativos (PID). Esta investigação propõe uma análise comparativa de diferentes formas discretas do controlador PID. Inicialmente, são apresentados os aspectos teóricos dos métodos de discretização e das estruturas típicas do controlador PID na sua forma paralela e com o elemento derivativo na realimentação (PI-D). Posteriormente, foram realizadas simulações dos controladores digitais e suas diferentes implementações. Finalmente, os algoritmos discretos de PID foram embarcados em uma plataforma contendo um microcontrolador ARM® Cortex®-M4 de baixo custo. Os resultados indicam importantes ganhos educacionais demonstrando que o controlador em tempo real PI-D discretizado utilizando aproximação retangular para a frente para o termo integrativo e a regra retangular para trás para o termo derivado representa o melhor compromisso entre custo computacional e estabilidade.

Palavras Chaves— Teoria de controle, Controlador PID, Controle digital, Microcontroladores.

I. INTRODUÇÃO

SISTEMAS de controle digital compreendem um dos principais tópicos do currículo dos cursos de engenharia [1]-[4]. Uma das estratégias para o desenvolvimento de controladores digitais é a aproximação discreta de controladores contínuos. Nestes casos, o controlador projetado por meio da transformada de Laplace é transferido para o tempo discreto, obtendo um controlador digital equivalente ao

controlador original. Para realizar esta aproximação, são apresentadas aos alunos diferentes técnicas de aproximação de tempo discreto, entre elas a diferenciação numérica (diferenciação para frente ou método de Euler e a diferenciação para trás), a integração numérica (método trapezoidal denominado método de Tustin), dentre outras. Entretanto, usualmente, não se faz uma discussão crítica sobre quais são as vantagens e desvantagens de cada método. Além disso, não são exploradas as condições de aplicação destes métodos em sistemas em tempo real.

Dentro dos possíveis controladores apresentados aos alunos, os controladores PID (Proporcional-Derivativo-Integrativo) são amplamente empregados [5]-[8]. Existem diferentes formas de implementação do algoritmo PID em controladores digitais, que variam de acordo com o método de discretização utilizado, tanto em relação à parte integrativa como à parte derivativa e seu filtro passa-baixas [8]-[13]. Este artigo apresenta um estudo comparativo entre as diferentes discretizações do controlador PID. Para tanto, são analisadas as possibilidades de discretizações do controlador PID em termos de estabilidade, influência da presença de ruídos e custo computacional [11], [12].

Inicialmente, são apresentadas e discutidas com os alunos as diferentes técnicas de discretização. Posteriormente, são detalhadas as discretizações do controlador PID, considerando as topologias com e sem filtro no elemento derivativo, e também com o elemento derivativo na saída do controlador (controlador PI-D). Para realizar as simulações e caracterizações, foram projetados controladores aplicados a um processo de primeira ordem com retardo, função de transferência tipicamente encontrado em plantas industriais. Em seguida, os controladores foram implementados em um microprocessador ARM® Cortex®-M4 para avaliação do seu tempo de execução através da análise do número de ciclos de *clock* necessários para executar o algoritmo de controle.

O artigo está estruturado como se segue: a Seção II detalha

F. C. Conceição, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil.

W. Beccaro, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, wesley@lme.usp.br.

J. F. Justo, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil.

os métodos de discretização de sistemas contínuos. A Seção III discute os materiais e métodos utilizados para realizar os testes comparativos das diferentes implementações do PID discreto. A Seção IV apresenta os testes realizados e uma análise comparativa e as conclusões são detalhadas na Seção V.

II. MÉTODOS DE DISCRETIZAÇÃO DE SISTEMAS CONTÍNUOS E CONTROLADORES PID

A. Discretização de Sistemas Contínuos: Mapeamento do plano-s no plano-z

O projeto de controladores discretos pode ser realizado diretamente no domínio z, ou através de emulação, que consiste no projeto de um controlador no domínio s e a posterior conversão para o domínio z, para ser implementado digitalmente em sistemas embarcados, sejam eles microcontroladores, PLCs (Programmable Logic Controllers) ou FPGAs (Field Programmable Gate Arrays).

Para obter um equivalente discreto de um controlador contínuo, existem diferentes métodos de aproximação, sendo os mais comuns: a diferenciação para frente, a diferenciação para trás e o método de Tustin. Também podem ser considerados os métodos de mapeamento de polos e zeros, a equivalência do segurador, etc.

O método da diferença em avanço retangular para frente (forward rectangular) é uma técnica simples que substitui a derivada de uma função pela aproximação (1).

$$\frac{de(t)}{dt} \approx \frac{e(t_{k+1}) - e(t_k)}{T_s}, \quad (1)$$

onde T_s é o período de amostragem e $t_k = kT_s$ os instantes de amostragem.

Para $e(t) = 0$ em $0 \leq t < T_s$ tem-se no domínio s (2).

$$sE(s) = \frac{e^{sT_s} E(s) - E(s)}{T_s} = E(s) \left[\frac{e^{sT_s} - 1}{T_s} \right], \quad (2)$$

Com $z = e^{sT_s}$ obtém-se a eq. (3) que permite, através da troca da variável s, a discretização de uma função de transferência de um sistema contínuo:

$$s = \frac{z - 1}{T_s} \quad (3)$$

Analogamente, a diferença em atraso, retangular para trás (backward rectangular) apresenta a equivalência dada pela eq. (4).

$$s = \frac{1 - z^{-1}}{T_s} = \frac{z - 1}{T_s z} \quad (4)$$

Já o método de Tustin é uma técnica que aproxima o problema através de uma integração numérica. A aproximação

trapezoidal permite a equivalência discreta através da eq. (5).

$$s = \frac{2}{T_s} \frac{(1 - z^{-1})}{(1 + z^{-1})} = \frac{2}{T_s} \frac{(z - 1)}{(z + 1)} \quad (5)$$

A Fig. 1 apresenta o mapeamento do semiplano esquerdo de s no plano z para as aproximações: retangular para frente, retangular para trás e Tustin, respectivamente.

Para a aproximação retangular para frente, verifica-se que sistemas estáveis, alocados no semiplano esquerdo de s, podem se tornar instáveis quando mapeados para o plano z. Dessa forma, seu uso deve considerar possíveis questões de estabilidade.

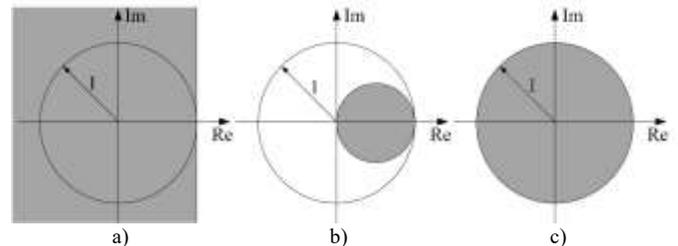


Fig. 1. Mapeamento do semiplano da esquerda do plano s no plano z para as aproximações: a) retangular para frente, b) retangular para trás e c) trapezoidal ou método de Tustin.

A aproximação retangular para trás mapeia sistemas estáveis em s em sistemas estáveis no plano z. Contudo, nem toda a região de estabilidade no plano z é utilizada no mapeamento, o que restringe o espaço de projeto, tornando a solução mais difícil caso os requisitos de projeto demandem desempenho específicos.

Por último, o método Tustin mapeia todo o semiplano esquerdo de s no interior do círculo unitário do plano z. Para que haja congruência das regiões de estabilidade há uma distorção em frequência, que pode ser compensada.

B. Discretização do controlador PID

Para o controlador PID em tempo contínuo, sua forma paralela, apresentada na Fig. 2, e sua transformada de Laplace são descritos pelas eqs. (7) e (8), respectivamente:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (7)$$

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (8)$$

onde K_p é o ganho proporcional, $K_i = K_p/T_i$ representa o ganho da integral (reset gain), T_i é o tempo integrativo (reset time), $K_d = K_p T_d$ representa o ganho da derivada (rate gain) e T_d é o tempo derivativo.

No modelo do controlador PID paralelo, os parâmetros têm significado físico. T_i corresponde ao tempo para compensar a soma de todos os erros anteriores fazendo com que a saída do integrador atinja o valor K_p para uma entrada unitária. Desta forma, para valores pequenos de K_i , o sistema pode levar muito

tempo para atingir o valor de referência. Por outro lado, se K_i possuir valor elevado, o sistema pode se tornar instável. Já a ação derivativa fornece uma correção antecipada do erro, em um tempo T_d , diminuindo o tempo de resposta e melhorando a estabilidade do sistema.

O controlador PID na sua forma mais convencional apresenta alguns inconvenientes associados à possibilidade de saturação de componentes da malha de controle. Em especial, o termo associado à derivada no controlador PID produz um ganho que tende ao infinito em altas frequências [5]-[7]. Tal questão pode ser vista através da resposta ao degrau de um controlador PID que fornece uma função delta de Dirac na saída do elemento derivativo [8].

Como a presença de ruídos é inevitável nas aplicações reais, é usual ocorrer saturação dos elementos finais de controle, como, por exemplo, amplificadores e atuadores, devido à derivação de um sinal com ruído, bem como os altos valores de ganho dos elementos derivativos [6], [7].

Usualmente, para contornar esse problema, pode-se variar a função de transferência do controlador PID, acrescentando-se um filtro passa-baixas de primeira ordem no termo associado à derivada. Outra estratégia que pode ser adotada é o uso do controlador PID sem o bloco derivativo. Existem outras técnicas para restringir grandes variações nas entradas dos blocos derivativos e integrativos [6]-[8].

Numa implementação prática e realista, para que o efeito do ruído seja atenuado, é comum inserir ao termo derivativo um polo fora da origem, que representa um filtro passa-baixas de primeira ordem (9):

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{NK_d s}{s + N}, \quad (9)$$

onde N corresponde ao coeficiente do filtro derivativo.

Além da forma paralela do controlador PID, uma outra forma de implementar o controlador consiste na alocação do elemento derivativo diretamente na realimentação da saída. Com essa alteração, as variações abruptas no sinal de referência que podem levar a ação de controle a valores de amplitude muito altos, efeito conhecido como “kick derivativo”, são atenuadas. A estrutura que implementa o termo derivativo com filtro atuando na realimentação da saída é apresentada na Fig. 3. No domínio do plano complexo, considera-se: $R(s)$ o sinal de referência, $E(s)$ o sinal de erro, $U(s)$ o sinal de controle e $Y(s)$ corresponde ao sinal de saída do processo.

Como descrito anteriormente, em aplicações digitais é necessário discretizar os elementos do controlador PID. Como há mais de uma maneira de realizar essa implementação, os sistemas discretos nem sempre possuem o mesmo comportamento dos sistemas em tempo contínuo originalmente representados. Durante a discretização, cada tipo de aproximação possui suas vantagens e desvantagens.

Uma forma de discretizar o controlador PID é converter separadamente os termos integral e derivativo para sua forma de tempo discreto correspondente. As eqs. (10), (11) e (12) representam essas variações para o termo integrativo do

controlador PID. Dado o período de amostragem T_s , o termo integral K_i/s pode ser representado na forma discreta por:

$$\text{Retangular para frente: } \frac{K_i T_s}{z-1}, \quad (10)$$

$$\text{Retangular para trás: } \frac{K_i T_s z}{z-1}, \quad (11)$$

$$\text{Tustin: } \frac{K_i T_s}{2} \frac{z+1}{z-1}. \quad (12)$$

De forma similar, o termo derivativo com filtro (9) pode ser discretizado por (13), (14) e (15):

$$\text{Retangular para Frente: } \frac{K_d N}{1 + \frac{NT_s}{z-1}} = \frac{NK_d(z-1)}{z-1+NT_s}, \quad (13)$$

$$\text{Retangular para Trás: } \frac{NK_d(z-1)}{(1+NT_s)z-1}, \quad (14)$$

$$\text{Tustin: } \frac{NK_d(z-1)}{\left(1 + \frac{NT_s}{2}\right)z + \frac{NT_s}{2} - 1}. \quad (15)$$

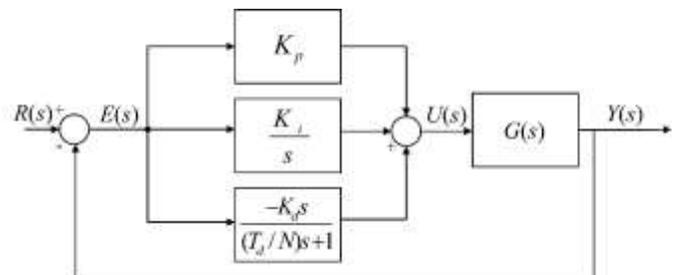


Fig. 2. Topologia do controlador PID paralelo e com filtro no elemento derivativo.

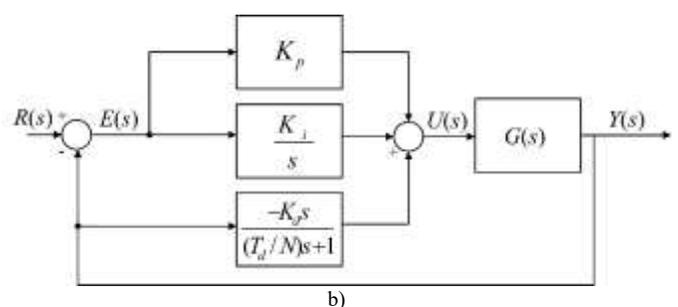


Fig. 3. Topologia do controlador PI-D com o elemento derivativo realimentado na saída.

O controlador PID de tempo discreto pode ser obtido através do uso dos mesmos processos de discretização para ambos os termos, integral e derivativo. Além disso, é possível realizar a implementação através de uma combinação dos métodos de discretização.

A título de exemplo, as eqs. (16), (17) e (18) apresentam as discretizações utilizando, respectivamente, os métodos retangular para frente, retangular para trás e Tustin para ambos os termos integrativos e derivativos.

$$G_c(z) = K_p + \frac{K_i T_s}{z-1} + \frac{N(z-1)}{z-1+NT_s} \quad (16)$$

$$G_c(z) = K_p + \frac{K_i T_s z}{z-1} + \frac{NK_d(z-1)}{(1+NT_s)z-1} \quad (17)$$

$$G_c(z) = K_p + \frac{K_i T_s}{2} \frac{z+1}{z-1} + \frac{N(z-1)}{\left(1 + \frac{NT_s}{2}\right)z + \frac{NT_s}{2} - 1} \quad (18)$$

Na eq. (19) é utilizado o método retangular para frente para o termo integral e o método retangular para trás no termo derivativo. A eq. (20) apresenta a realização utilizando o método Tustin para o termo integral e o método retangular para trás para o termo derivativo.

$$G_c(z) = K_p + \frac{K_i T_s}{z-1} + \frac{NK_d(z-1)}{(1+NT_s)z-1} \quad (19)$$

$$G_c(z) = K_p + \frac{K_i T_s}{2} \frac{z+1}{z-1} + \frac{NK_d(z-1)}{(1+NT_s)z-1} \quad (20)$$

C. Implementação do Controlador PID Discreto com Filtro Digital no Termo Derivativo

Para que a implementação seja feita em um microprocessador, o controlador discreto em z deve ser representado através de equações de diferenças.

Sejam P , D e I respectivamente as saídas das parcelas do controlador referentes aos elementos proporcional, derivativo com o filtro de primeira ordem e integrativo, suas equações a diferenças na forma geral são dadas pelas eqs. (21), (22) e (23) [8]:

$$P(t_k) = K_p (b \cdot r(t_k) - y(t_k)), \quad (21)$$

$$D(t_k) = a_d D(t_{k-1}) - b_d [y(t_k) - y(t_{k-1})], \quad (22)$$

$$I(t_{k+1}) = I(t_k) + b_{i_1} e(t_{k+1}) + b_{i_2} e(t_k), \quad (23)$$

sendo b o parâmetro de ajuste do erro, $0 \leq b \leq 1$. Por vezes, é possível utilizar essa mesma estratégia aplicando este peso ao erro. Já os parâmetros a_d , b_d , b_{i_1} e b_{i_2} , têm seus correspondentes valores atribuídos para cada tipo de discretização, conforme a Tabela I, adaptada de [8], que sintetiza a análise comparativa das possíveis discretizações obtidas através dos coeficientes da equação a diferenças. O sinal de saída do controlador corresponde à soma das partes proporcional, derivativa e integral, conforme (24) [8], [9]:

$$v(t_k) = P(t_k) + D(t_k) + I(t_k), \quad (24)$$

sendo $v(t_k)$ a soma das partes dos termos do controlador PID que deve ser mantido dentro de limites de saturação para se obter o sinal de controle $u(t_k)$.

Algumas observações podem ser extraídas das diferentes

implementações apresentadas. Para todos os termos discretizados, o período de amostragem afeta tanto o elemento integral como o derivativo. Observa-se que para frequências de amostragem elevadas não há variação significativa entre os tipos de discretização, desde que seja mantida a estabilidade. Isso ocorre porque valores pequenos de T_s anulam os coeficientes da parte integral, bem como o efeito do filtro passa-baixas. Em outras palavras, quanto maior a frequência de amostragem, maior é a equivalência entre os métodos e maior é a aproximação com o sistema em tempo contínuo original.

De acordo com a eq. (22), o controlador é estável quando $|a_d| < 1$. Para o método retangular para frente isso impõe a restrição ao tempo derivativo, $T_d > NT_s/2$, desta forma valores pequenos de T_d causam instabilidade [8].

Todos os outros métodos são estáveis para qualquer valor de T_d . Por esse motivo, o método retangular para frente é raramente utilizado para a discretização do elemento derivativo. Além disso, para aplicações práticas de controle, devido à causalidade, é impossível utilizar o método retangular para frente, já que há necessidade de uma amostra que ainda não está disponível.

TABELA I
COEFICIENTES DAS EQUAÇÕES DE DIFERENÇAS OBTIDAS ATRAVÉS DE DIFERENTES APROXIMAÇÕES

COEFICIENTES DA EQUAÇÃO A DIFERENÇAS	RETANGULAR PARA FRENTE	RETANGULAR PARA TRÁS	TUSTIN
b_{i_1}	0	$\frac{KT_s}{T_i}$	$\frac{KT_s}{2T_i}$
b_{i_2}	$\frac{KT_s}{T_i}$	0	$\frac{KT_s}{2T_i}$
a_d	$1 - \frac{NT_s}{T_d}$	$\frac{T_d}{T_d + NT_s}$	$\frac{2T_d - NT_s}{2T_d + NT_s}$
b_d	KN	$\frac{KT_d N}{T_d + NT_s}$	$\frac{2KT_d N}{2T_d + NT_s}$

O método de Tustin gera boas aproximações para a função de transferência contínua, porém para valores pequenos de T_d , o valor de a_d torna-se negativo, gerando comportamento oscilatório indesejável.

Desta maneira, o método retangular para trás é preferencialmente utilizado para aproximações do termo derivativo, pois funciona bem com valores pequenos de T_d , sendo capaz de produzir bons resultados até mesmo para a condição em que T_d seja igual a 0, o que é desejável como opção de desligar a parte derivativa em um controlador de propósito geral [8].

A implementação das equações pode ser descrita no pseudocódigo descrito no algoritmo contido no Quadro 1 [8], [9].

Este algoritmo descreve a implementação utilizando as aproximações retangular para trás para a parte derivativa e retangular para frente na parte integral. Um aspecto de implementação deve ser destacado para este pseudocódigo que

consiste na independência do coeficiente b_i para a aproximação retangular para frente (vide Tabela 1), o que possibilita o cálculo da parcela integral com a antecipação da chegada de novas amostras. Assim, o algoritmo possui um menor número de operações dentro da rotina de interrupção.

Quadro 1 - Algoritmo do controlador PID paralelo com aproximação retangular para trás no elemento derivativo e método retangular para frente no elemento integral

```

1 // Coeficientes do controlador
2 ad=Td/(Td+N*Ts)
3 bd=K*Td*N/(Td+N*Ts)
4 bi1=0
5 bi2=K*Ts/(Ti)
6 // Interrupção
7 PID_function_ISR ()
8 r=adc_dma_chan_1() // ADC referência
9 y=adc_dma_chan_2() // ADC variável de processo
10 e=r-y // Erro
11 P=K*(b*r-y) // Parte proporcional, b ≠ 0
12 D=ad*D-bd*(y-y_old) // Parte derivativa
13 v=P+I+D // PID
14 u=sat(v,ulow,uhigh) // Saturação da saída DAC
15 dac_dma_chan_1(u) // DAC sinal de controle
16 I=I+bi2*(e) // Parte integrativa
17 y_old=y // Atualização da variável de processo y
    
```

III. MATERIAIS E MÉTODOS

A partir dos conceitos apresentados, são propostas duas atividades práticas. A primeira consiste na simulação utilizando a ferramenta matemática MATLAB®/Simulink e a segunda através da implementação dos controladores PID discretos em um sistema embarcado em tempo real. Através desses exercícios, os alunos aprendem conceitos de simulação e programação, bem como verificam na prática o comportamento dos métodos de discretização e das estruturas PID paralela e PI-D.

A. Simulação das Estruturas PID e PI-D Utilizando Simulink

Para avaliação da resposta ao degrau do sistema de controle realimentado foi proposta a simulação conforme a Fig. 4. Optou-se em demonstrar aos alunos a representação mais próxima da implementação real, que consiste no cálculo da equação a diferenças diretamente em um bloco de função. Realizações estruturais distintas do elemento derivativo foram obtidas mediante ajuste do código do bloco. Considerou-se ainda a presença de ruído de medição na saída da planta. A Fig. 4 representa o diagrama utilizado, que descreve os seguradores de ordem zero, o bloco da equação de diferenças do algoritmo PID, a função de transferência do processo e seu respectivo atraso de transporte.

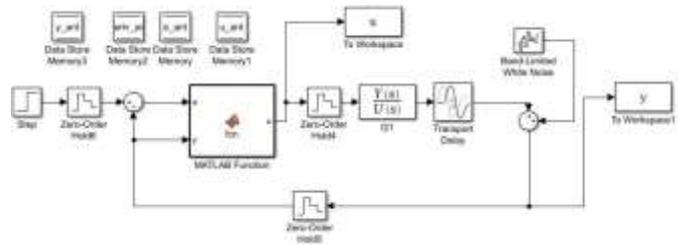


Fig. 4. Diagrama em blocos do sistema de controle implementado através da discretização de diferentes aproximações do controlador PID.

B. Caracterização do Tempo de Processamento Utilizando o Microprocessador ARM® Cortex®-M4

Para análise do tempo de processamento das implementações do controlador PID foi utilizado o microcontrolador ARM® Cortex®-M4 (STM32F407VGT6) com 32 bits e 168 MHz/210 DMIPS, 1 MB de memória flash e 192+4 KB de memória RAM (Random Access Memory). O microcontrolador possui como periféricos 3 conversores analógico-digital (ADC) e 2 conversores digital-analógico (DAC) de 12 bits, ambos realizando os processos de conversão via aproximações sucessivas.

Foram utilizados todos esses recursos disponibilizados no kit de desenvolvimento STM32F407 Discovery. O software embarcado disponibilizado foi concebido em bare metal (sem sistema operacional), baseado em rotina de interrupção.

O microcontrolador dispõe de dois controladores de acesso direto à memória (Direct Memory Access - DMA) que possibilitam o recurso de buffer ping-pong. A utilização do DMA é fundamental, ao passo que diminui a carga do processador, deixando-o livre para executar o processamento e só sendo interrompido quando o DMA finaliza os processos de transferência do ADC. Com a utilização do recurso de buffers ping pong é possível manter o bloco de amostras que já foi movido e está sendo processado (bloco ativo), enquanto o controlador DMA copia o bloco subsequente para o segundo buffer (inativo). Assim, logo que o processamento é concluído, outro bloco já está disponível para ser processado [15].

Um temporizador (timer) é responsável por garantir cada período de amostragem, garantindo maior precisão na amostragem e assim diminuindo a relação sinal-ruído associada ao método de conversão. Ao término da conversão, o controlador DMA transfere o resultado da amostragem, localizado no registrador de dados do ADC, para uma posição de memória.

Uma rotina de interrupção associada ao DMA sinaliza, através de uma flag, que os dados estão prontos para o processamento. O loop principal testa essa flag continuamente, e, quando a mesma é acionada, o programa executa uma rotina responsável por selecionar o buffer ativo e chamar a função que implementa o algoritmo de controle.

Na arquitetura ARM® Cortex®-M4, quando uma interrupção é disparada, há mudança de contexto do modo thread para o modo de interrupção [16], [17] o que demanda alguns ciclos de clock do microprocessador, aumentando assim a latência. Da forma que o software foi projetado e disponibilizado aos alunos, não há interrupções associadas a cada amostra e sim a cada

bloco mínimo de amostras, o que torna mais eficiente o código em termos de latência para que seja mantida a execução em tempo real. Optou-se, também, por utilizar o DMA com blocos de tamanho mínimo, que consistem em duas amostras: a saída da planta realimentada e o valor de referência (*set point*).

IV. RESULTADOS E DISCUSSÕES

A. Análise da Estabilidade das Estruturas PID e PI-D

Os alunos avaliam três estruturas: o controlador PID com o elemento derivativo sem filtro, o controlador com o filtro de primeira ordem na forma ideal do controlador PID, e o controlador PI-D com bloco derivativo separado da malha direta, sendo inserido com o filtro de primeira ordem como realimentação da saída da planta. Utilizou-se para a discretização, nas três situações, o método de Tustin para a parte integral, garantindo uma aproximação sempre estável e o método retangular para trás para a parte derivativa, que funciona bem para diferentes valores de T_d e, conforme a discussão da Seção II, demonstrou ser o mais adequado para a parcela derivativa.

A planta de teste consiste em um sistema de primeira ordem com atraso de transporte, cuja função de transferência é dada por $G(s) = \frac{e^{-0,2s}}{s+1}$ [18].

O controlador PID foi ajustado com coeficientes $K_p = 4$, $T_i = 1$, $T_d = 0,1$ e constante $N = 10$ [18], garantindo valor de tempo de acomodação menor que 1,98 s e sobressinal máximo menor do que 27,46% para todos os casos analisados.

A Fig. 5 apresenta a resposta ao degrau obtida para as três configurações do controlador PID. Pode-se notar um menor sobressinal para o controlador PI-D. Para complementar a análise, a Fig. 6 representa, para cada caso, a ação de controle do algoritmo PID implementado.

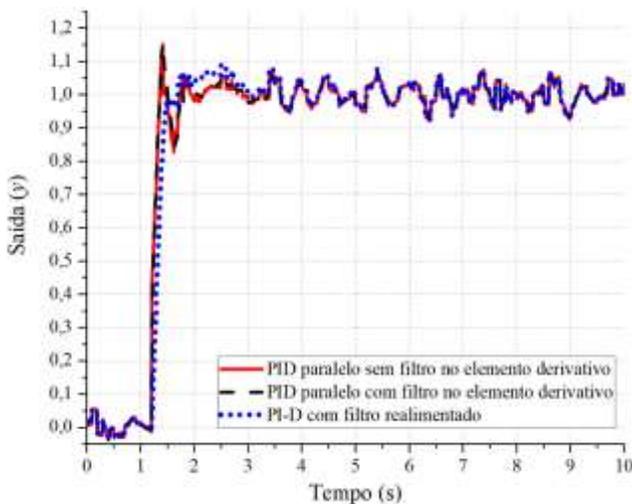


Fig. 5. Resposta ao degrau do sistema de controle implementado para as três configurações do controlador PID.

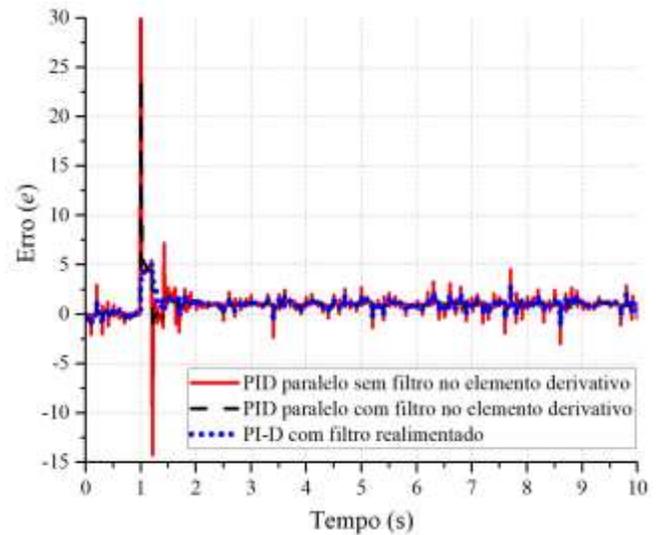


Fig. 6. Ação de controle para as três implementações do controlador PID.

Como a resposta ao degrau do elemento derivativo fornece um delta de Dirac em sua saída, é necessário um maior esforço de controle para manter um sobressinal dentro das especificações desejadas. Esta análise permite, de fato, demonstrar que o uso do elemento derivativo na realimentação (controlador PI-D) é capaz de atenuar o efeito do “kick derivativo”. Conseqüentemente, o controlador PI-D possui menor sobressinal na resposta ao degrau do sistema evitando a saturação de elementos finais de controle, como, por exemplo, amplificadores e atuadores.

B. Análise do Tempo de Execução dos Controladores PID

O tempo de execução dos blocos de controle é um aspecto importante a ser considerado em sistemas digitais com processamento em tempo real. Com a implementação das equações a diferenças no microcontrolador STM32F4, no ambiente de desenvolvimento Keil MDK, foi possível adquirir o número de ciclos de *clock* gasto para a execução da rotina que realiza o controle PID discreto.

A arquitetura ARM disponibiliza uma unidade *Data Watchpoint and Trace Unit* (DWT) com contador dedicado de ciclos de *clock* a nível de *hardware* [16], [17]. De posse da quantidade de ciclos gastos é possível estimar o tempo de execução da rotina considerando a frequência de operação do microprocessador de 168 MHz. Tais valores estão representados na Tabela II.

A frequência de *clock* do microprocessador é muito maior do que a frequência de amostragem, no caso 10 Amostras/s, o que garante a execução em tempo real sem restrições de tempo.

A metodologia adotada teve como objetivo analisar comparativamente o método de discretização do elemento integrativo, mantendo-se o método retangular para trás para a parcela derivativa para todas as variações de algoritmos implementados.

A partir da Tabela II, levantada pelos alunos, verifica-se que o método de Tustin aplicado ao elemento integrativo apresenta maior tempo de execução, sendo a maior diferença notada no

caso de sua implementação sem o filtro.

Para os métodos retangular para a frente e Tustin, a simples adição do filtro de primeira ordem aumenta em aproximadamente 13% e 12% o tempo de execução do algoritmo de controle, respectivamente. A diferença entre os tempos das discretizações retangular para frente e Tustin ocorre devido a um coeficiente a menos no termo integrativo da aproximação retangular para frente, tal como descrito na Tabela I, o que gera tempos absolutos menores.

TABELA II
ANÁLISE COMPARATIVA DOS FILTROS DIGITAIS IMPLEMENTADOS COM DISCRETIZAÇÃO RETANGULAR PARA TRÁS PARA O TERMO DERIVATIVO

MÉTODO DE DISCRETIZAÇÃO DO ELEMENTO INTEGRATIVO	FILTRO IMPLEMENTADO	CICLOS DE CLOCK	TEMPO DE EXECUÇÃO (EM NS)
	Sem Filtro	126	750
Retangular para Frente	Com filtro de primeira ordem e forma ideal	143	851
	Com o filtro realimentado na saída	134	797
	Sem Filtro	131	779
Tustin	Com filtro de primeira ordem e forma ideal	147	875
	Com o filtro realimentado na saída	142	845

Comparando as discretizações com o controlador PI-D com filtro realimentado na saída, verifica-se que o método de Tustin tem um acréscimo de, aproximadamente, 6% em comparação ao método retangular para frente.

V. CONCLUSÕES

Esse trabalho apresenta uma análise dos diferentes métodos de discretização aplicados ao projeto de controladores PID discretos. Foi possível verificar por meio de testes de simulação o comportamento do sistema através da sua resposta ao degrau, realizados de forma a caracterizar a implementação mais próxima da realidade (equação a diferenças e adição de ruído de medição). Verificou-se que a o controlador PID com filtro realimentado na saída representa a implementação mais robusta, com menor sobressinal, sendo o mais adequado para situações genéricas a fim de evitar saturação dos demais elementos do sistema.

Em termos de tempo de execução, foi observado que com o método retangular para trás no elemento derivativo e o método retangular para frente no termo integrativo é possível obter o menor tempo de execução dentre as situações avaliadas. Ressalta-se que para o controle de sistemas físicos lentos, a ordem de grandeza dos valores de tempo de execução obtidos (nano segundos), pode ser considerada irrelevante. Contudo para sistemas que trabalham com elevada frequência de amostragem, a diferença de dezenas de nano segundos pode ser crucial. Dessa forma, a escolha dentre as possíveis implementações do controlador PID discreto deve ser baseada nas necessidades e requisitos do projeto, que ditam a relevância

de se utilizar uma estrutura com maior ou menor custo computacional.

A arquitetura ARM® Cortex®-M4 mostrou-se capaz de executar o processamento demandado e sugere assim que a aplicação possa ser ampliada em complexidade e ser estendida a uma rede distribuída de controladores.

Este trabalho apresenta aspectos que tanto os alunos de engenharia como engenheiros devem considerar no desenvolvimento de controladores PID discretos, contribuindo para os projetistas na escolha das aproximações que atendam aos requisitos de projeto estabelecidos. Tal fato se insere dentro de um contexto de aumento da demanda por sistemas embarcados capazes de realizar controle digital em tempo real, tendo em vista uma maior presença da automação no cotidiano das pessoas.

REFERÊNCIAS

- [1] O. Jaramillo Chamorro, L. D. Guerrero Otoyá and J. F. Flórez Marulanda, "Rapid Control Prototyping with Scilab/Scicos, RTAI-Lab for the Mic955 Training Module of the Feedback Instruments," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 13(3), pp. 85-92, Aug. 2018.
- [2] P. N. Baldini, G. L. Calandrini, P. D. Doñate and H. R. Bambill, "Early Introduction of Robust Control Concepts: Hands-On Experience and CAD," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 11(2), pp. 56-62, May 2016.
- [3] J. A. N. Cocota, T. D'Angelo and P. M. de Barros Monteiro, "A Project-Based Learning Experience in the Teaching of Robotics," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 10(4), pp. 302-309, Nov. 2015.
- [4] J. I. Sosa et al., "Industrial Plant at Academic Level for Teaching Industrial Informatics in an Electronic Engineering Undergraduate Degree," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 12(1), pp. 1-9, Feb. 2017.
- [5] P. L. Castrucci, A. Bittar, R. M. Sales, *Controle Automático*. 1 ed. Rio de Janeiro: GEN/LTC, 2011.
- [6] C. A. Smith, A. B. Corripio, *Principles and practice of automatic process control*. 3. ed. New York: John Wiley, 2005.
- [7] W. L. Luyben, *Process modeling, simulation and control for chemical engineers*. 2. ed. New York: McGraw-Hill, 1990.
- [8] K. J. Aström, K.J. and Hägglund, T. *Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society*, Research Triangle Park, NC 27709, 2005.
- [9] W. S. Levine, *The Control Handbook*, Boca Raton: CRC Press, 1996.
- [10] O. Garpinger, *Design of robust PID controllers with constrained control signal activity*, Licentiate Thesis LUTFD2/TFRT-3245 -SE, Department of Automatic Control, Lund University, Sweden, 2009.
- [11] A. Isaksson, S. Graebe, "Derivative filter is an integral part of PID design," *Control Theory and Applications, IEE Proceedings*, vol. 149, no. 1, pp. 41-45, 2002.
- [12] P. O. Larsson, T. Hägglund, "Control signal constraints and filter order selection for PI and PID controllers," *Proceeding of American Control Conference*, 2011.
- [13] T. Sekara, M. Matausek, "Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise," *IEEE Transactions of Automatic Control*, vol. 54(1), pp. 184, 2009.
- [14] P. Pivoňka, M. Schmidt, "Comparative Analysis of Discrete Derivative Implementations in PID Controllers," *Proceeding of the 11th WSEAS International Conference on Systems*, Greece, 2007.
- [15] T. Martin, *The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach*, 2 ed. Newnes, 2016.
- [16] J. Yiu, *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors*, 3 ed. Amsterdam: Newnes, 2013.
- [17] D. S. Reay, *Digital Signal Processing Using the ARM Cortex M4*, 1 ed. Hoboken, New Jersey: Wiley, 2015.
- [18] A. Visioli, *Practical PID Control*, 2006 edition. London: Springer, 2006.

Received: 05 June 2019
Accepted: 25 September 2020;
Published: 15 October 2020



© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative

Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Teaching Discretization Methods for Control Systems Design: Comparative Analysis of Embedded Discrete-Time PID Controllers

Abstract— Several techniques have been used to teach the conversion of continuous systems into discrete ones. An interesting alternative to teaching the discretization method can be performed presenting the design of Proportional–Integral–Derivative (PID) controllers. Besides, comparing the implementations of PID in terms of stability and execution time helps the students to understand the limitations of PID structures and algorithms. This investigation proposes a comparative analysis of different discrete form of the PID controller. A theoretical discussion was described initially, presenting the discretization methods and the typical PID in parallel form and with derivative action in the feedback path (PI-D controller). Subsequently, simulations were performed. Finally, the discrete PID algorithms were implemented in an embedded platform containing a low-cost ARM® Cortex®-M4 microcontroller. The results indicate important educational gains by demonstrating that the real-time PI-D controller with forward rectangular rule for the integrative term and the backward rectangular rule for the derivative term represents the best compromise between computational cost and robust implementation.

Key Words— Control theory, PID control, Digital control, Microcontrollers, Control engineering computing