

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277954245>

PID digital – Método de cálculo numérico

Article · June 2014

CITATIONS

0

READS

1,684

1 author:



[Henrique Frank Werner Puhlmann](#)

Instituto de Pesquisas Tecnológicas

50 PUBLICATIONS 7 CITATIONS

SEE PROFILE

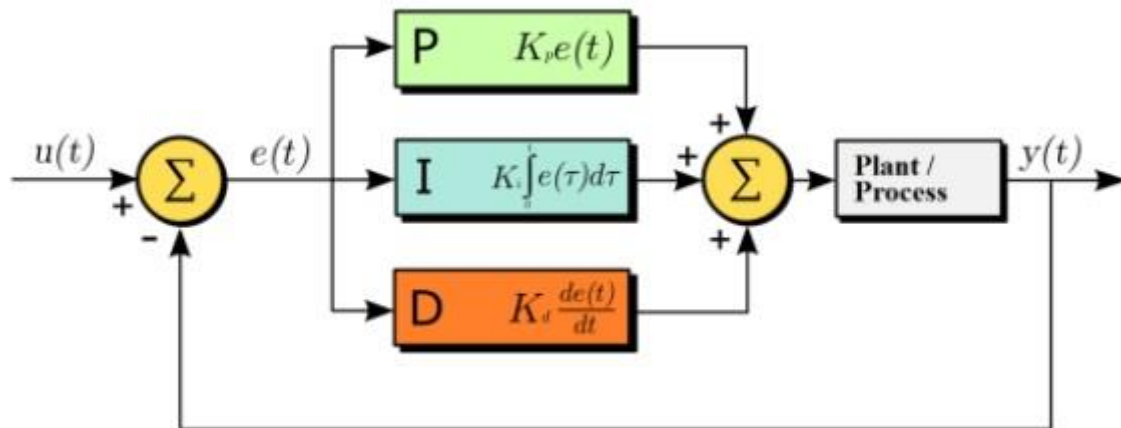
Some of the authors of this publication are also working on these related projects:



FAPESP/IPT [View project](#)

PID digital – Método de cálculo numérico

por Henrique Frank W. Puhlmann



Introdução

No artigo técnico Controlador PID digital: Uma modelagem prática para microcontroladores – Parte 1, é apresentado com detalhes o controlador PID, sua arquitetura típica, o desenvolvimento das equações para o controle e alguns exemplos de como se pode desenvolver um controle PID. Nesse artigo serão retomados alguns conceitos sobre esse tipo de controle e serão apresentadas formas de cálculo numérico para implementar a parte diferencial e a integral do controlador. Essa forma alternativa de cálculo casa muito bem com a aplicação em microcontroladores ou processadores de 8 bits ou outros relativamente lentos. Os resultados que serão apresentados a seguir, foram largamente aplicados nos primeiros controladores PID digitais e certamente ainda são úteis nos dias de hoje, por sua simplicidade e velocidade de processamento.

Arquitetura típica de um controlador PID

Um sistema de controle típico de PID em malha fechada tem o aspecto mostrado na **Figura 1**. Nessa figura pode-se observar uma entrada, referente a um sinal de referência ou *setpoint*, um bloco onde essa referência é subtraída de um sinal proveniente do processo, em geral medido de alguma forma por um sensor, transdutor etc. Essa subtração produz um sinal de erro que entra no controlador PID propriamente dito, e a saída do controlador, que atua sobre o processo que está sendo controlado, é aplicada para conduzir o processo ao estado desejado.

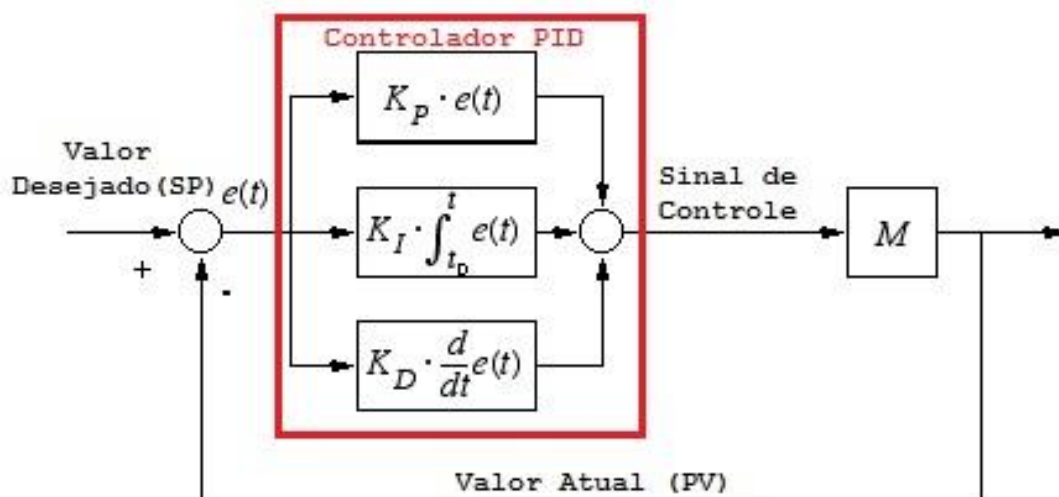


Figura 1: Controle PID típico

Na prática, o controlador PID digital destacado na **Figura 1** em vermelho, pode ser representado como mostrado na **Figura 2**. O erro é amostrado, produzindo assim um conjunto de amostras discretizadas do sinal de erro, que é conduzido a um conversor analógico / digital (A/D), depois são calculados os valores da derivada, da integral e do valor proporcional, em seguida esses valores calculados são somados e conduzidos a um conversor digital / analógico (D/A).

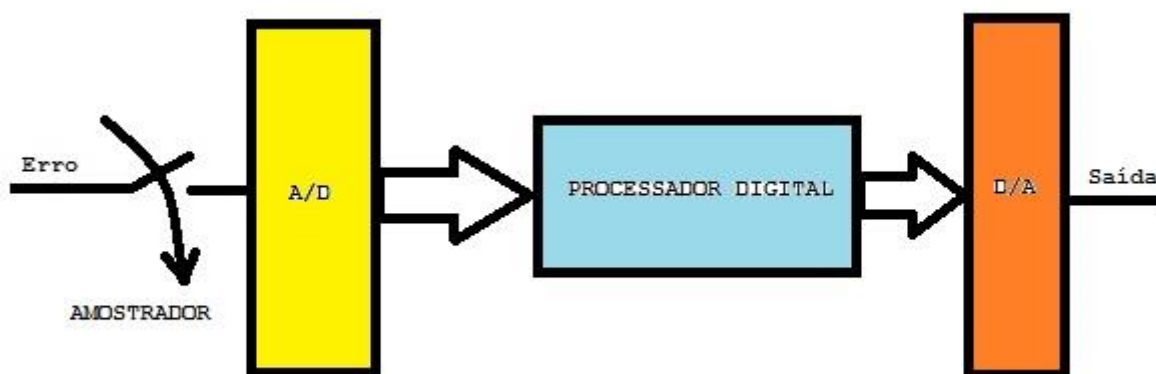


Figura 2: Controlador PID em blocos

O desenho da **Figura 2** representa o fluxo do sinal de erro para o controlador PID de forma a simplificar o desenvolvimento dos cálculos que serão apresentados a seguir. É evidente que num controlador real, os sinais de SP (*setpoint*) e PV (*present value*) também são digitalizados e subtraídos para que se obtenha o erro. O importante é fixar o conceito de que o erro deve ser entregue ao controlador numa frequência ou taxa fixa. Isso é requisito para o desenvolvimento que será realizado a seguir. Para que o sistema funcione bem, também é necessário que a atualização do sinal de saída pelo conversor D/A seja feito na mesma frequência do amostrador de entrada. Na **Figura 3** é mostrado um sinal analógico

como exemplo de sinal de erro e são destacadas as amostras realizadas num período fixo (Δ).

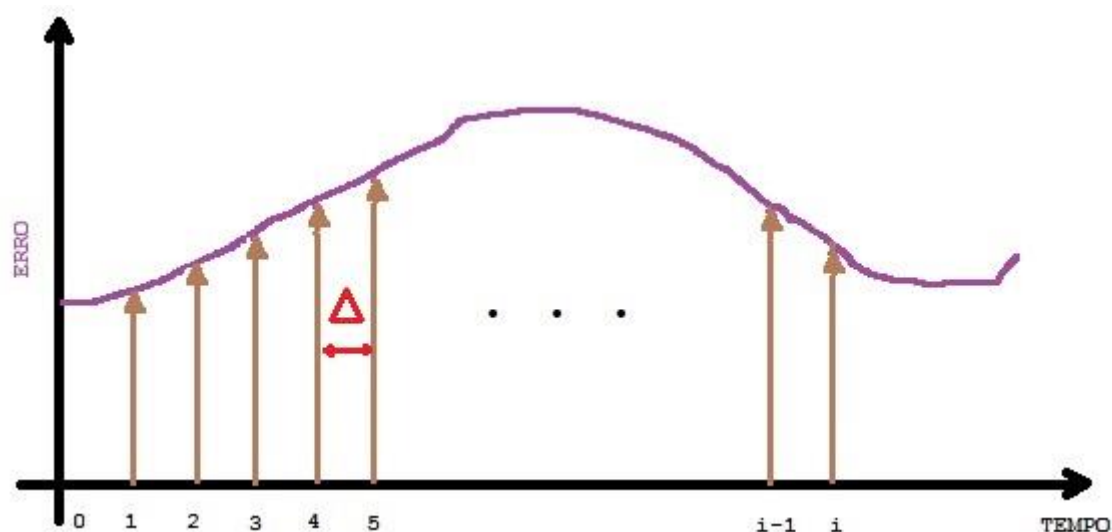


Figura 3: Exemplo de sinal de erro e de sua amostragem

A seguir será desenvolvido um pouco de teoria, o suficiente para que se possa saber de onde foram tiradas as equações que resultarão desse desenvolvimento. Se você preferir, pode consultar direto os quadros resumo que apresentam as equações resultantes desses cálculos.

Séries de Taylor

Brook Taylor elaborou uma teoria a respeito de funções matemáticas em 1715, conhecida como teorema de Taylor ou séries de Taylor. Esse teorema estabelece que: sendo f uma função infinitamente diferenciável, ela pode ser representada pela seguinte fórmula:

$$f(x) = \sum_{k=0}^{\infty} f^k(x_0) \frac{(x-x_0)^k}{k!} \quad (1)$$

onde:

- $f^k(x_0)$ representa a k-ésima derivada da função avaliada no ponto x_0 ;
- x_0 é um ponto arbitrário de referência;
- $k!$ é o fatorial de k.

Se a somatória acima convergir, a função pode ser representada por uma série finita de termos. Como por exemplo a expansão da série (1):

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + f''(x_0) \frac{(x-x_0)^2}{2!} + \dots + f^n(x_0) \frac{(x-x_0)^n}{n!} \quad (2)$$

Obs: Quando $x_0 = 0$, a série (2) é conhecida como série de McLaurin.

A seguir serão utilizadas as séries de Taylor para deduzir formas simples para o cálculo da primeira derivada de uma função e a integral dessa função.

Derivação Numérica

Seja definida uma função $f(x)$, representada por uma série de Taylor, tal como mostrada em (1), onde $x = i$ é uma amostra arbitrária de índice i tal como mostrado na **Figura 3**. O valor de $f(x)$ será batizado de f_i . Seja $f_{(i-1)}$ o valor dessa mesma função calculado no índice $(i-1)$. A distância entre essas amostras é de Δ , correspondente a uma unidade de tempo relacionada com a frequência ou taxa de amostragem. Daí obtemos:

$$f_i = \sum_{k=0}^{\infty} f^k(x_0) \frac{(x-x_0)^k}{k!} \Big|_{x=i} \quad (3)$$

Para avaliar a função em $(i-1)$, vamos substituir i por $(i-\Delta)$. Realizando a substituição e rearranjando a equação obtida, chega-se na equação (4).

$$f_{i-1} = \sum_{k=0}^{\infty} f_i^k \frac{(-\Delta)^k}{k!} \quad (4)$$

Escrevendo essa somatória na forma expandida, obtém-se:

$$f_{i-1} = f_i - f'_i \Delta + f''_i \frac{\Delta^2}{2!} - f'''_i \frac{\Delta^3}{3!} + \dots \quad (5)$$

Se desprezarmos os termos de segunda ordem e superiores e rearranjando, a equação fica assim:

$$f'_i = \frac{f_i - f_{i-1}}{\Delta} \quad (6), \text{ e o erro} = \frac{\Delta}{2} f''_i$$

A expressão (6) é conhecida como a **derivada com 2 pontos**.

É evidente que nesse cálculo, o erro é relativamente grande. Pode-se estender o raciocínio desenvolvido acima para um cálculo mais preciso, que pode ser obtido com 3, 4 ou mais amostras do sinal.

Se substituirmos i por $(i - 2\Delta)$ e for desenvolvido o mesmo raciocínio que foi utilizado para $i-1$ poderemos levantar a equação de f_{i-2} em função de f_i .

$$f_{i-2} = f_i - 2f'_i\Delta + f''_i\frac{4\Delta^2}{2!} - f'''_i\frac{8\Delta^3}{3!} + \dots \quad (7)$$

Usando a equação (5) e rearranjando, obtém-se:

$$f'_i = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2\Delta} \quad (8) \quad \text{onde o erro} = -\frac{2\Delta^2}{3}f'''_i$$

Essa é a equação de cálculo de **derivada com 3 pontos**. De maneira análoga pode-se desenvolver **derivada de 4 pontos**, mostrada na equação (9).

$$f'_i = \frac{11f_i - 18f_{i-1} + 9f_{i-2} - 2f_{i-3}}{6\Delta} \quad (9) \quad \text{onde o erro} = \frac{\Delta^3}{4}f'''_i$$

A seguir será apresentado um quadro resumo com as fórmulas desenvolvidas até aqui.

Quadro resumo da diferenciação digital

Aqui será adotada uma nova notação, mais frequentemente utilizada em processamento digital de sinais (DSP) e em controle digital. Isso facilitará o entendimento de quem está habituado a essa notação e introduz a notação para quem ainda não a conhece. A notação é assim:

- y = saída (nesse caso a derivada da função);
- x = amostra atual da função;
- z^{-n} = n -ésima amostra anterior;
- Δ = intervalo de tempo entre amostras.

Resumo das fórmulas referentes à diferenciação digital:

=====

- Derivada calculada com **2 pontos**:

$$y = \frac{x - z^{-1}}{\Delta} \quad (10)$$

- Derivada calculada com **3 pontos**:

$$y = \frac{3x - 4z^{-1} + z^{-2}}{2\Delta} \quad (11)$$

- Derivada calculada com **4 pontos**:

$$y = \frac{11x - 18z^{-1} + 9z^{-2} - 2z^{-3}}{6\Delta} \quad (12)$$

=====

Os resultados obtidos nas equações acima são facilmente implementáveis em microprocessadores.

.

Integração Numérica

Para desenvolver as equações referentes à integração numérica, o raciocínio é o mesmo já desenvolvido para a diferenciação. Voltando à **Figura 3**, onde temos representada uma forma de onda arbitrária amostrada em intervalos regulares de Δ segundos, pode-se definir que:

$$I(t) = \int_0^t f(t) dt = I(i-1) + \int_{i-1}^i f(t) dt \quad (13)$$

Seja o segundo termo da equação (13):

$$I = \int_{i-1}^i f(t) dt \quad (14)$$

Desenvolvendo a série para I_{i-1} se obtém uma equação semelhante à equação (5):

$$I_{i-1} = I_i - I'_i \Delta + I''_i \frac{\Delta^2}{2!} - \dots \quad (15)$$

Rearranjando, obtém-se:

$$I_i - I_{i-1} = f_i \Delta - f'_i \frac{\Delta^2}{2!} + f''_i \frac{\Delta^3}{3!} - \dots \quad (16)$$

Desenvolvendo:

$$f_{i-1} = f_i - f'_i \Delta + f''_i \frac{\Delta^2}{2!} - \dots \quad (17)$$

Substituindo:

$$I = \frac{\Delta}{2} (f_i + f_{i-1} + f''_i (\frac{\Delta^2}{6}) + \dots) \quad (18)$$

desprezando os termos de segunda ordem ou superiores chega-se à equação para o cálculo da integral com **2 pontos**.

.

$$I = \frac{\Delta}{2} (f_i + f_{i-1}) \quad (19) \quad \text{onde o erro} = f''_i \left(\frac{\Delta^3}{12} \right)$$

A fórmula (19) é conhecida como “regra do trapézio”.

O cálculo da integral pode ser estendido para mais pontos, para aumentar a precisão do resultado. A seguir, a equação para o cálculo com **3 pontos** (20) e **4 pontos** (21).

$$I = \frac{\Delta}{12} (5f_i + 8f_{i-1} - f_{i-2}) \quad (20) \quad \text{onde o erro} = \frac{\Delta^4}{12} f_i''''$$

$$I = \frac{\Delta}{24} (9f_i + 19f_{i-1} - 5f_{i-2} + f_{i-3}) \quad (21) \quad \text{onde o erro} = \frac{19\Delta^5}{720} f_i'''''$$

Quadro resumo da integração digital

Aqui também será adotada a nova notação, mais frequentemente utilizada em processamento digital de sinais (DSP) e em controle digital. A notação é assim:

- y = saída (nesse caso a integral da função);
- x = amostra atual da função;
- z^{-n} = n -ésima amostra anterior;
- Δ = intervalo de tempo entre amostras.

Resumo das fórmulas referentes à integração digital:

=====

- Integral calculada com **2 pontos**:

$$y = (x + z^{-1}) \frac{\Delta}{2} \quad (22)$$

- Integral calculada com **3 pontos**:

$$y = (5x + 8z^{-1} - z^{-2}) \frac{\Delta}{12} \quad (23)$$

- Integral calculada com **4 pontos**:

$$y = (9x + 19z^{-1} - 5z^{-2} + z^{-3}) \frac{\Delta}{24} \quad (24)$$

=====

Observe que as fórmulas acima calculam a integral incremental entre uma amostra e a anterior. Para que seja realizado um cálculo de integral de fato é necessário acumular os resultados.

Obtidos os resultados acima, é interessante verificar se eles realmente funcionam. Para isso eles serão simulados num programa para cálculos matemáticos conhecido por Scilab. Serão mostrados os respectivos códigos e os resultados. Para que você possa testar e explorar na sua própria máquina será necessário instalar o Scilab nela. Baixe o programa aqui: **SCILAB vers. 5.5.0**. O programa é gratuito e tem versões para Windows, GNU/Linux e Mac OS X.

Inicie o programa e digite o código a seguir. Se você preferir, você pode marcar o código, copiá-lo (Ctrl-C) e colá-lo no Scilab (Ctrl-V). Para iniciar as operações basta teclar o “Enter”.

Cálculo de Integrais numéricas utilizando o Scilab

```

1 // Inicializar as variáveis e a forma de onda para os testes
2
3 nCiclos = 10; // Define o número de ciclos da forma de onda
4 nAmostras = 40; // Define o número de amostras por ciclo
5
6 // Gera o vetor de referência para a produção da forma de onda
7
8 t = (1:2*%pi/nAmostras:nCiclos * 2 * %pi + (1-2*%pi/nAmostras));
9 nvErro = squarewave(t); // Gera uma onda quadrada com 400 amostras
10
11 // Gera o gráfico para essa forma de onda
12
13 a0 = figure(0);
14 a0.background = -2;
15
16 plot(nvErro);
17
18 // Gera o grid e as legendas
19
20 xtitle('Forma de onda do Erro para testes', 'Índice das amostras', 'Amplitude');
21 set(gca(), "data_bounds", [0,400,-1.2,1.2]);
22 set(gca(), "grid", [1,1]);

```

Os comandos acima geram um gráfico parecido com o apresentado na **Figura 4**.

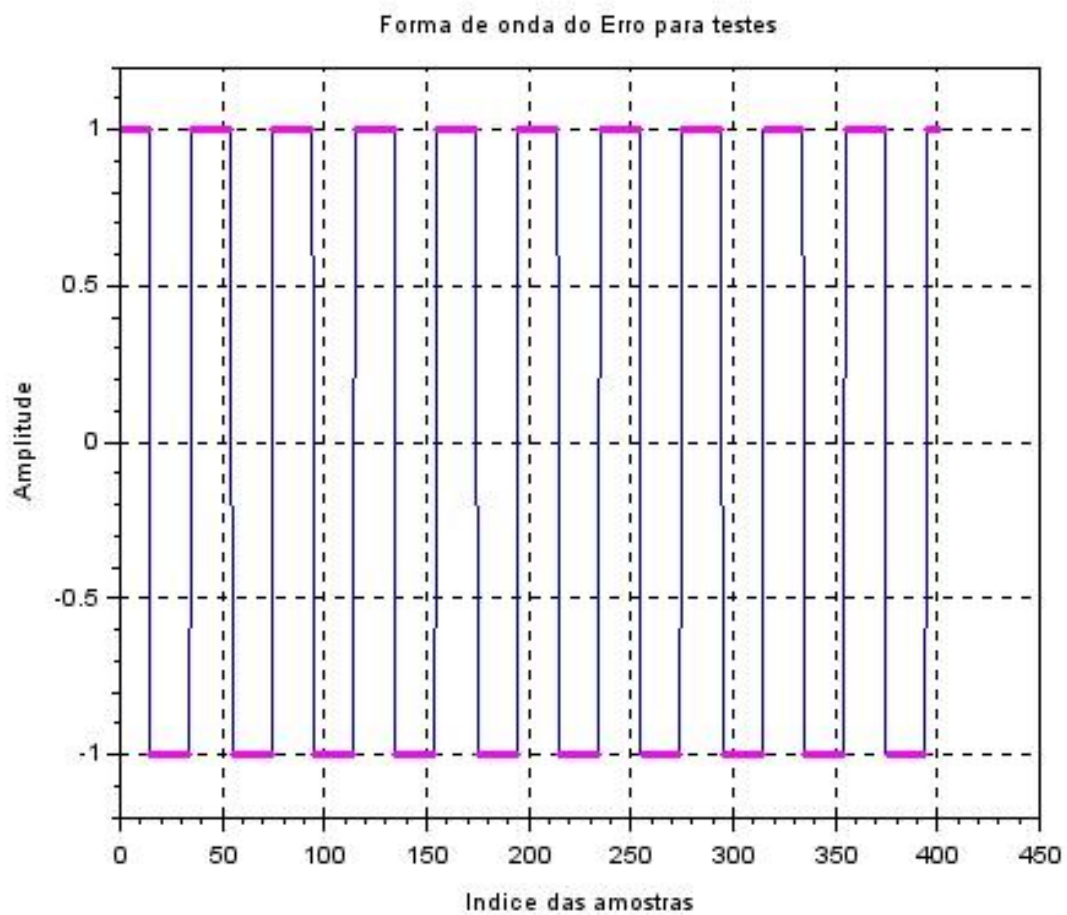


Figura 4: Gráfico da forma de onda quadrada gerada no Scilab

Como não há formas de onda triangulares prontas no Scilab, será utilizada a forma de onda quadrada, calculada sua integral para formar a onda triangular e calculada a diferencial da onda triangular para formar a onda quadrada novamente. Digite o seguinte código no Scilab (ou copie e cole):

```

1 // Inicializa as variáveis para o cálculo da integral com 2 pontos
2
3 Int2p = zeros(1, size(nvErro,2));
4 Int2pAcum = 0;
5 Z1 = 0;
6 Delta = 1;
7
8 // Calcula a Integral com 2 pontos
9
10
11 for nIndex = 1:size(nvErro,2)
12     Int2p(nIndex) = ((nvErro(nIndex) + Z1)* Delta/2) + Int2pAcum;
13     Int2pAcum = Int2p(nIndex);
14     Z1 = nvErro(nIndex);
15 end;
16
17 // Apresenta a integral no gráfico
18
19 a1 = figure(1);
20 a1.background = -2;
21
22 plot(Int2p, 'm');
23
24 // Gera o grid e as legendas
25
26 set(gca(), "background", [-2]);
27 xtitle('Integral da forma de onda calculada com 2 pontos', 'Indice das amostras', 'Amplitude');
28 set(gca(), "data_bounds", [0,400,-10,20]);
29 set(gca(), "grid", [1,1]);
30
31 =====
32
33 // Inicializa as variáveis para o cálculo da integral com 3 pontos
34
35 Int3p = zeros(1, size(nvErro,2));
36 Int3pAcum = 0;
37 Z1 = 0;
38 Z2 = 0;
39 Delta = 1;
40
41 // Calcula a Integral com 3 pontos
42
43 for nIndex = 1:size(nvErro,2)
44     Int3p(nIndex) = ((5*nvErro(nIndex) + 8*Z1- Z2)* Delta/12) + Int3pAcum;
45     Int3pAcum = Int3p(nIndex);
46     Z2 = Z1;
47     Z1 = nvErro(nIndex);
48 end;
49
50 // Gera gráfico
51
52 a2=figure(2);
53 a2.background = -2;
54
55 plot(Int3p, 'b');

```

```

56
57 // Gera o grid e as legendas
58
59 set(gca(), "background", [-2]);
60 xtitle('Integral da forma de onda calculada com 3 pontos', 'Indice das amostras', 'Amplitude');
61 set(gca(), "data_bounds", [0,400,-10,20]);
62 set(gca(), "grid", [1,1]);
63
64 //=====
65 integral com 4 ponto
66
67 Int4p = zeros(1, size(nvErro,2));
68 Int4pAcum = 0;
69 Z1 = 0;
70 Z2 = 0;
71 Z3 = 0;
72 Delta = 1;
73
74 // Calcula a Integral com 4 pontos
75
76 for nIndex = 1:size(nvErro,2)
77     Int4p(nIndex) = ((9*nvErro(nIndex) + 19*Z1 - 5*Z2 + Z3)* Delta/24) + Int4pAcum;
78     Int4pAcum = Int4p(nIndex);
79     Z3 = Z2;
80     Z2 = Z1;
81     Z1 = nvErro(nIndex);
82 end;
83
84 // Gera gráfico figure(3);
85
86 a3 = figure(3);
87 a3.background = -2;
88
89 plot(Int4p, 'r');
90
91 // Gera o grid e as legendas
92
93 set(gca(), "background", [-2]);
94 xtitle('Integral da forma de onda calculada com 4 pontos', 'Indice das amostras', 'Amplitude');
95 set(gca(), "data_bounds", [0,400,-10,20]);
    set(gca(), "grid", [1,1]);

```

Os gráficos obtidos são como o mostrado na **Figura 5**.

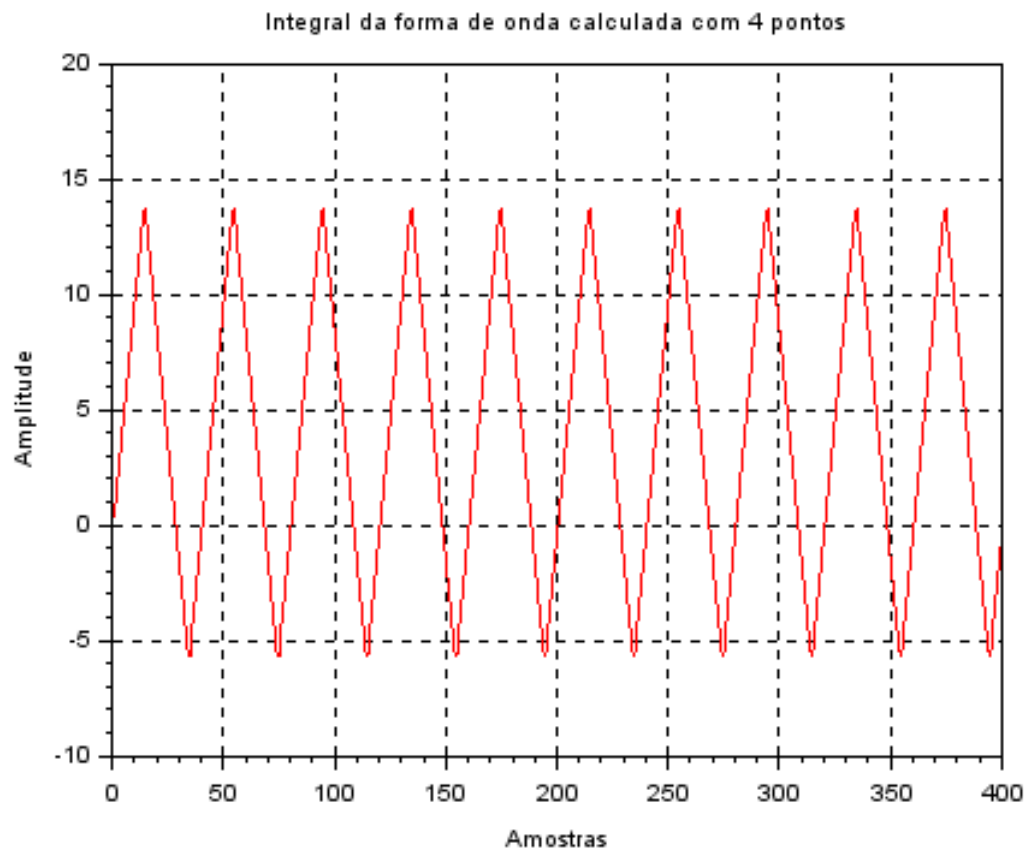


Figura 5: Gráfico da integral calculada com 3 pontos

Os três gráficos obtidos parecem ser iguais, por motivo da forma de onda quadrada ser muito simples. Se olharmos com detalhes os “bicos” das formas de onda ampliadas, poderemos observar as diferenças (**Figura 6**). Quanto maior o número de pontos usados para o cálculo, menor a diferença se comparada com a onda triangular ideal.

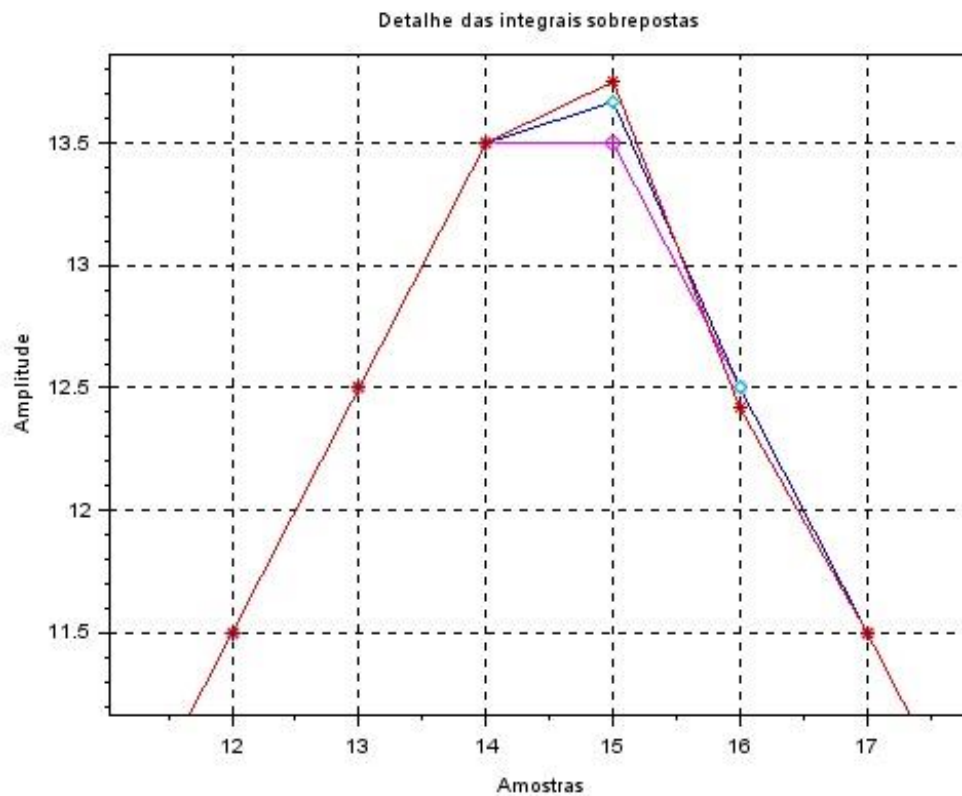


Figura 6: Gráfico das integrais calculadas sobrepostas

Cálculo de Diferencias numéricas utilizando o Scilab

Em seguida serão mostradas a utilização das diferenciais calculadas com 2, 3 ou 4 pontos. A onda utilizada para essas operações é a onda triangular resultante da integral calculada com 4 pontos.

```

1 // Calculo das diferencias numéricas utilizando-se 2 pontos
2
3 // Fecha os gráficos
4
5 close(a0);
6 close(a1);
7 close(a2);
8 close(a3);
9
10
11 Diff2p = zeros(1, size(Int4p,2));
12 Z1 = 0;
13 Delta = 1;
14

```

```

15 // Calcula a Diferencial com 2 pontos
16
17 for nIndex = 1:size(nvErro,2)
18     Diff2p(nIndex) = ((Int4p(nIndex) - Z1)* 1/Delta);
19     Z1 = Int4p(nIndex);
20 end;
21
22 // Apresenta a diferencial no gráfico
23
24 a0 = figure(0);
25 a0.background = -2;
26
27 plot(Diff2p, 'm');
28
29 // Gera o grid e as legendas
30
31 set(gca(), "background", [-2]);
32 xtitle('Diferencial da forma de onda calculada com 2 pontos', 'Indice das amostras', 'Amplitude');
33 set(gca(), "data_bounds", [0,400,-1.5,1.5]);
34 set(gca(), "grid", [1,1]);
35
36 //=====
37
38 // Calculo das diferencias numéricas utilizando-se 3 pontos
39
40 Diff3p = zeros(1, size(Int4p,2));
41 Z1 = 0;
42 Z2 = 0;
43 Delta = 1;
44
45 // Calcula a Diferencial com 3 pontos
46
47 for nIndex = 1:size(nvErro,2)
48     Diff3p(nIndex) = ((3*Int4p(nIndex) - 4*Z1 + Z2)* 1/(2*Delta));
49     Z2 = Z1;
50     Z1 = Int4p(nIndex);
51 end;
52
53 // Apresenta a deferencial no gráfico
54
55 a1 = figure(1);
56 a1.background = -2;
57
58 plot(Diff3p, 'b');
59
60 // Gera o grid e as legendas
61
62 set(gca(), "background", [-2]);
63 xtitle('Diferencial da forma de onda calculada com 3 pontos', 'Indice das amostras', 'Amplitude');
64 set(gca(), "data_bounds", [0,400,-2.5,2.5]);
65 set(gca(), "grid", [1,1]);
66

```



```

67 //=====
68
69 // Calculo das diferencias numéricas utilizando-se 4 pontos
70
71 Diff4p = zeros(1, size(Int4p,2));
72 Z1 = 0;
73 Z2 = 0;
74 Z3 = 0;
75 Delta = 1;
76
77 // Calcula a Diferencial com 4 pontos
78
79 for nIndex = 1:size(nvErro,2)
80     Diff4p(nIndex) = ((11*Int4p(nIndex) - 18*Z1 + 9*Z2 - 2*Z3)* 1/(6*Delta));
81     Z3 = Z2;
82     Z2 = Z1;
83     Z1 = Int4p(nIndex);
84 end;
85
86 // Apresenta a deferencial no gráfico
87
88 a2 = figure(2);
89 a2.background = -2;
90
91 plot(Diff4p, 'r');
92
93 // Gera o grid e as legendas
94
95 set(gca(), "background", [-2]);
96 xtitle('Diferencial da forma de onda calculada com 4 pontos', 'Indice das amostras', 'Amplitude');
97 set(gca(), "data_bounds", [0,400,-2.5,2.5]);
98 set(gca(), "grid", [1,1]);

```

Os gráficos obtidos são semelhantes ao da **Figura 7**.

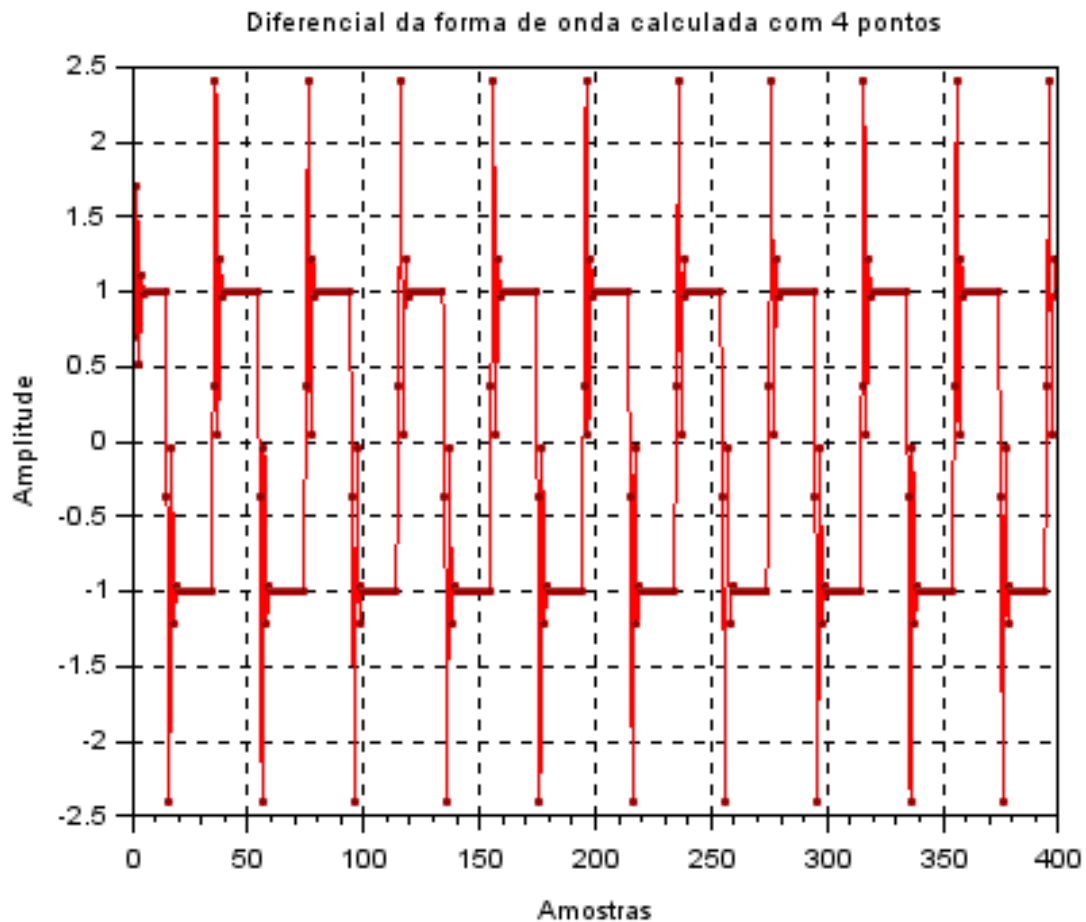


Figura 7: Diferencial da onda triangular calculada com 4 pontos

Na **Figura 8** pode-se observar as três diferenciais sobrepostas. Observe que a diferencial com mais pontos tem um transitório maior do que as outras, o que está coerente. Aqui encerram-se as simulações para demonstrar o funcionamento dos métodos numéricos.

É recomendável que o leitor faça experimentos com o Scilab, crie outras formas de onda, utilize formas de onda reais, pois o Scilab permite que se carregue dados de arquivos, e se familiarize com os recursos do Scilab, se acaso ainda não os conhece. O Scilab é uma ferramenta poderosa, tem recursos para cálculos de problemas de controle, processamento digital de sinais entre muitos outros.

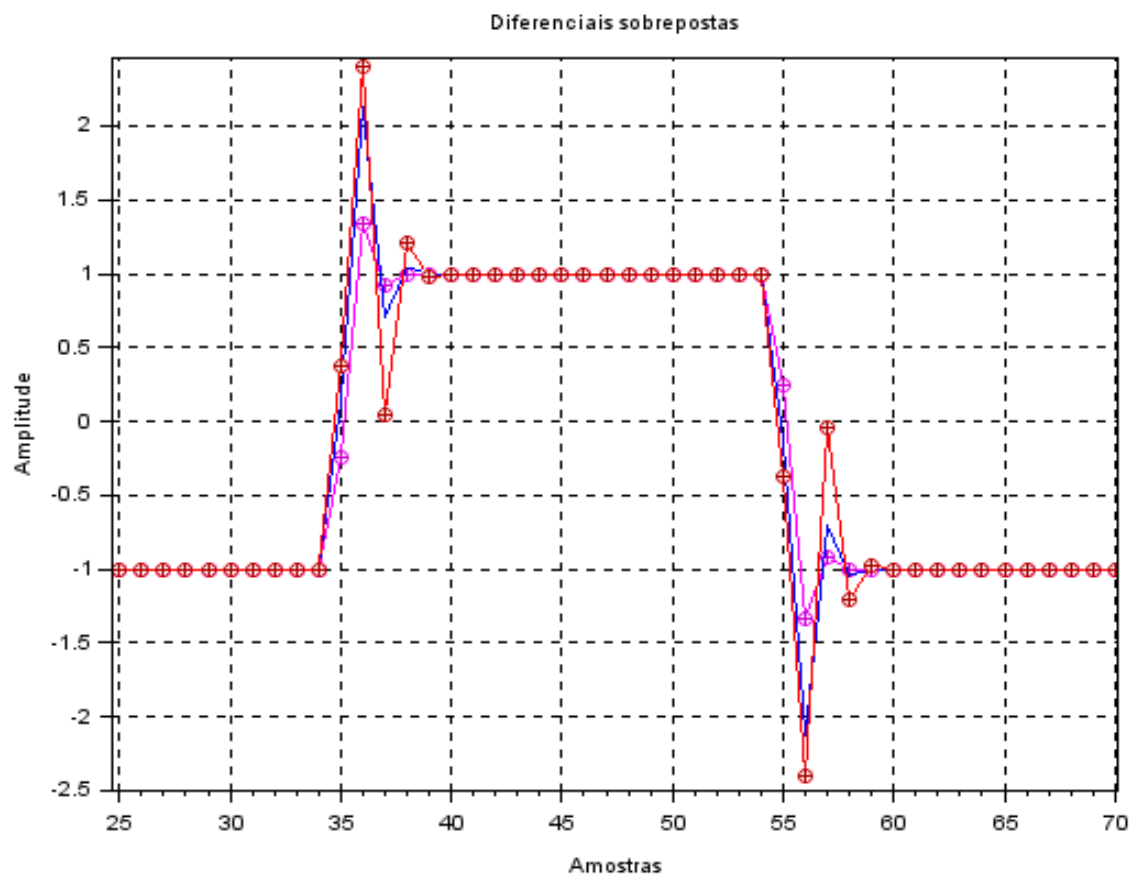


Figura 8: Detalhe da sobreposição das diferenciais calculadas

Nesse artigo técnico foram desenvolvidos métodos numéricos para o cálculo de integrais e diferenciais, típicas para uso em controle digital por PID. Esses métodos foram simulados no Scilab e foi mostrado o funcionamento correto das diversas aproximações e uma melhor precisão à medida que se utiliza mais pontos para o cálculo dos resultados. As equações desenvolvidas são muito adequadas para a utilização em microprocessadores pela sua simplicidade. Experimente usá-las.