

# Recuperação digital da temporização com amostragem assíncrona - Parte 1: transmissão em banda-base

© Sílvio A. Abrantes

Departamento de Engenharia Electrotécnica e de Computadores  
Faculdade de Engenharia, Universidade do Porto  
Porto, Portugal  
sam@fe.up.pt

Março de 2010

## 1. Introdução

Este texto ocupa-se da recuperação digital da temporização de símbolos, ou recuperação de relógio, assumindo que a transmissão se realiza em banda-base. Um texto subsequente tratará o mesmo tema aplicado à transmissão com modulações digitais como MPSK e QAM. A base de ambos é [1], que introduziu a sincronização digital em tempo contínuo e em tempo discreto no que se refere à recuperação da fase de portadoras e à recuperação de relógio. A presente contribuição descreve com algum pormenor técnicas de recuperação da temporização exclusivamente em tempo discreto e nesse aspecto é um complemento de [1]; aqui e ali segue também o exposto em [2], cap. 7.

A recuperação da temporização de símbolos é uma operação crítica em qualquer receptor digital e tem como objectivo estimar com a máxima precisão possível os instantes de início e de fim dos símbolos recebidos. Admitamos que os dados são transmitidos ao ritmo de  $R = 1/T$  símbolos por segundo e que a decisão sobre o  $k$ -ésimo símbolo deve ser tomada no instante  $kT + \tau$ . O objectivo dos procedimentos que se seguem é estimar  $\tau$  com o rigor possível. Sendo  $\hat{\tau}$  essa estimativa o decisor recebe amostras nos instantes  $t_k = kT + \hat{\tau}_k$ .

As técnicas de recuperação da temporização podem dividir-se em dois grupos consoante o tipo de amostragem que usam: síncrona ou assíncrona. No primeiro grupo a amostragem está ligada ao sinal enviado e é comandada por um oscilador de controlo numérico (NCO); no segundo a amostragem não está solidária com os impulsos recebidos – é independente deles – e o relógio local está fixo. A amostragem assíncrona é a única que permite uma realização, ou implementação, completamente digital e pode ser usada em esquemas de recuperação de malha aberta, sem realimentação (*feedforward*), ou de malha fechada, com realimentação (*feedback*).

Neste texto só vão ser abordadas técnicas de amostragem assíncrona. Como só vamos considerar transmissão em banda-base todos os sinais e amostras que iremos encontrar são reais.

A Fig. 1 apresenta um esquema geral de recuperação de relógio com amostragem assíncrona e realimentação e a Fig. 2 apresenta um esquema sem realimentação. Começemos pela Fig. 1. O sinal em banda-base  $r(t)$  recebido foi previamente filtrado por um filtro que lhe limita a largura de banda a  $B$  Hz e depois amostrado com uma frequência de amostragem  $f_s = 1/T_s$ . As amostras  $r(mT_s)$  resultantes são em seguida aplicadas a um filtro adaptado, que produz as amostras  $z(mT_s)$  à mesma cadência temporal de  $T_s$  segundos e as entrega ao bloco *Interpolador*. Este, por sua vez, produz amostras  $y(t_n)$  em instantes  $t_n$  diferentes de  $mT_s$ , amostras que são entregues ao decisor e ao bloco TED<sup>1</sup> (*Detector de Erro de Temporização*). O erro  $e(k)$  na saída deste é produzido de  $T$  em  $T$  segundos e vai servir ao controlador para indicar ao interpolador os instantes de interpolação  $\{t_n\}$ . A sequência que o decisor recebe é previamente decimada por  $N$  dado que o espaçamento entre amostras  $y(t_n)$  é  $T/N$  e o espaçamento entre símbolos é  $T$ .

---

<sup>1</sup> TED: *Timing Error Detector*.

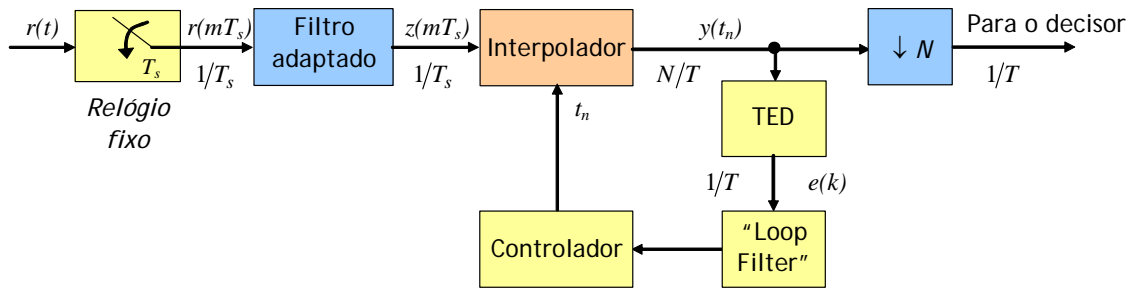


Fig. 1. Recuperação de relógio com amostragem assíncrona e realimentação.

De acordo com o teorema da amostragem a frequência de amostragem do amostrador tem de ser tal que  $f_s = 1/T_s \geq 2B$ . Normalmente vai do dobro ao quádruplo da taxa de símbolos  $R$ , isto é, o intervalo de amostragem  $T_s$  é cerca de metade a um quarto do tempo de símbolo  $T$  [2]. Em princípio  $T_s$  e  $T$  são *incomensuráveis*, isto é,  $T/T_s$  é um número irracional. E porquê? Porque, como o relógio do amostrador provém de uma fonte local independente do relógio do emissor, seria muita coincidência que houvesse qualquer relação entre  $T_s$  e  $T$ .

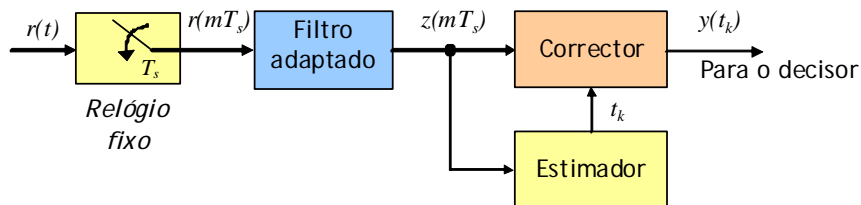


Fig. 2. Recuperação de relógio com amostragem assíncrona sem realimentação.

No método sem realimentação (Fig. 2) o bloco *Corrector*, a que poderíamos também chamar *Interpolador* como na Fig. 1, calcula as amostras  $y(t_k)$  nos instantes estimados de temporização  $t_k$  que recebe do *Estimador*. Note-se que não se calcula nenhum erro de temporização.

As Figs. 1 e 2 indicam-nos que, de um modo geral, a recuperação da temporização consiste em duas operações distintas:

1. *Medição da temporização*, quando se obtém a estimativa  $\hat{\tau}$  da “fase” de temporização  $\tau$ .  
Onde? No TED/Controlador e no Estimador.
2. *Ajuste da temporização*, quando se aplica  $\hat{\tau}$  ao processo de obtenção das amostras para o decisor.  
Onde? No Interpolador e no Corrector.

Vamos começar pelos métodos realimentados para depois terminar com os métodos de malha aberta.

## 2. Esquemas com realimentação

Tenhamos em conta a Fig. 1. Aos novos valores  $y(t_n)$  obtidos por interpolação vamos chamar *interpolantes*. Os intervalos de tempo entre interpolantes consecutivos não são rigorosamente constantes: andam (“flutuam”) à volta de um certo valor médio  $T/N$  submúltiplo de  $T$  chamado *intervalo de interpolação*, em que  $N$  é tipicamente 1 ou 2. É necessário um interpolante por símbolo para a detecção dos dados, naturalmente; veremos que, consoante o método, basta um ou dois interpolantes por símbolo para a sincronização.

O erro de temporização calculado pelo TED depende do algoritmo escolhido. Por exemplo, no detector de Gardner, a apresentar mais adiante, o erro é calculado com base em três amostras  $y(t_n)$  espaçadas de cerca de  $T/2$  segundos.

Convém atentar nos três índices de tempo que aparecem na Fig. 1:

- $m$  – índice das amostras do sinal de entrada: conta as amostras do sinal original que vão “entrando” no filtro adaptado. Intervalo de tempo entre amostras:  $T_s$  segundos.
- $n$  – índice das interpolações: conta os interpolantes  $y(t_n)$ . Intervalo de tempo: cerca de  $T/N$  segundos.
- $k$  – índice dos símbolos: conta os símbolos de dados recebidos. Intervalo de tempo:  $T$  segundos.

Em consequência, a cada índice está associado um instante: de amostragem ( $m$ ), de interpolação ( $n$ ) e de temporização ( $k$ ).

Não há uma relação simples entre  $m$  e  $n$  (pois, como se disse,  $T/T_s$  é um número irracional). Pelo contrário,  $n$  e  $k$  estão relacionados simplesmente por  $k = \lfloor n/N \rfloor$ , em que  $\lfloor x \rfloor$  representa o “chão” de  $x$ , isto é, o maior inteiro igual ou inferior a  $x$ . Sendo  $N$  normalmente 1 ou 2 então  $k = n$  ou  $k = \lfloor n/2 \rfloor$ .

Tendo disponíveis as amostras  $z(mT_s)$  é possível recuperar a forma de onda  $z(t)$  subjacente através da conhecida fórmula de amostragem

$$z(t) = \sum_{m=-\infty}^{\infty} \text{sinc}[(t - mT_s)/T_s] z(mT_s) = \sum_{m=-\infty}^{\infty} h_I(t - mT_s) z(mT_s), \quad (1)$$

com  $h_I(t) = \text{sinc}(t/T_s)$ . A eq. (1) significa que  $z(t)$  pode ser obtida fazendo passar as amostras  $z(mT_s)$  por um filtro de resposta impulsional  $h_I(t)$ . Contudo, a utilização de (1) é difícil e pouco prática porque o somatório é composto de um número infinito de parcelas. Com um número finito a recuperação do sinal é imperfeita, claro, mas se em vez do seno cardinal forem usadas determinadas *funções interpoladoras* bem escolhidas o erro cometido é reduzido.

Em resumo queremos: 1) encontrar adequadas funções interpoladoras  $h_I(t)$  de duração finita; 2) determinar os instantes de interpolação  $t_n$  apropriados a uma correcta recuperação da temporização; e 3) calcular o erro de temporização  $e(k)$ , tudo para atingirmos o nosso objectivo final, que é obter boas amostras  $y(t_n) = z(t_n)$  para serem entregues ao decisor<sup>2</sup>.

Comecemos pela interpolação.

## 2.1 O interpolador e a interpolação polinomial

Na Fig. 3 seja  $l_n T_s$  o instante de amostragem imediatamente anterior ao instante  $t_n$  que procuramos. Assim, podemos escrever

$$l_n = \lfloor t_n/T_s \rfloor$$

e

$$t_n = l_n T_s + \mu_n T_s = (l_n + \mu_n) T_s, \quad (2)$$

onde  $l_n$  é inteiro e  $\mu_n$  é fraccionário,  $0 \leq \mu_n < 1$ . O parâmetro  $l_n$  é o *índice do ponto-base* e  $\mu_n$  é o *intervalo fraccionário*, dado por

$$\mu_n \triangleq t_n/T_s - \lfloor t_n/T_s \rfloor.$$

<sup>2</sup> O uso da letra  $y$  em vez da letra  $z$  serve para distinguir as amostras provenientes do amostrador,  $z(mT_s)$ , das produzidas pelo interpolador,  $z(t_n)$ . De facto é  $y(t_n) = z(t_n)$  mas com letras diferentes sabemos imediatamente que  $z$  se refere ao amostrador e  $y$  ao interpolador.

Por exemplo, se  $t_n = 4,3T_s$ , então  $l_n = 4$  e  $\mu_n = 0,3$ . Logo, nos instantes  $t_n$  temos os interpolantes

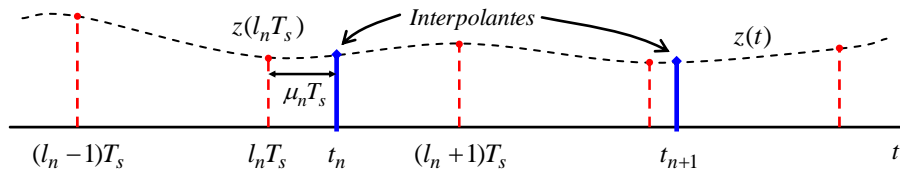


Fig. 3. Os instantes de amostragem e de interpolação.

$$y(t_n) = \sum_{m=-\infty}^{\infty} h_I[(l_n - m)T_s + \mu_n T_s] z(mT_s). \quad (3)$$

Desejamos fazer a interpolação com um número finito de parcelas tão baixo quanto possível. Vamos supor então que  $h_I(t)$  não é um seno cardinal mas sim uma função com um pequeno número de valores no intervalo de tempo  $-I_1 T_s \leq t \leq (I_2 + 1)T_s$  de duração  $(I_1 + I_2 + 1)T_s$ . O intervalo anterior corresponde a  $-I_1 T_s \leq (l_n - m)T_s + \mu_n T_s \leq (I_2 + 1)T_s$  ou  $l_n - I_2 \leq m \leq l_n + I_1$ , o que indica que apenas as amostras  $z(mT_s)$  com índices  $m$  entre  $l_n - I_2$  e  $l_n + I_1$  contribuem para o somatório. Nesses pressupostos (3) simplifica-se em

$$y(t_n) = \sum_{m=l_n-I_2}^{l_n+I_1} h_I[(l_n - m)T_s + \mu_n T_s] z(mT_s).$$

Fazendo  $i = l_n - m$  ainda simplificamos mais:  $y(t_n) = \sum_{i=-I_1}^{I_2} h_I(iT_s + \mu_n T_s) z[(l_n - i)T_s]$ . Definindo  $c_i(\mu_n) \triangleq h_I(iT_s + \mu_n T_s)$  obtemos a *fórmula fundamental do interpolador*:

$$y(t_n) = \sum_{i=-I_1}^{I_2} c_i(\mu_n) z[(l_n - i)T_s]. \quad (4)$$

Esta equação revela que para calcular uma amostra interpolante  $y(t_n)$  o interpolador (ver Fig. 4) precisa de:

- $I_1 + I_2 + 1$  pontos-base  $\{z(mT_s)\}$ ,  $l_n - I_2 \leq m \leq l_n + I_1$ ;
- $I_1 + I_2 + 1$  coeficientes  $\{c_i(\mu_n)\}$ ,  $-I_1 \leq i \leq I_2$ .

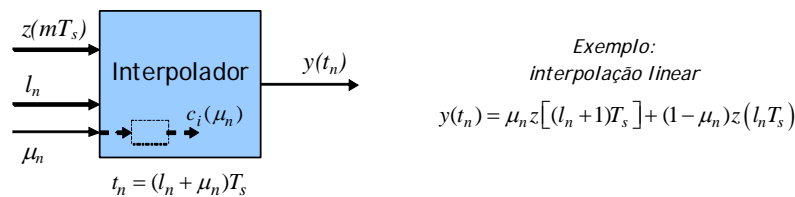


Fig. 4. Variáveis de entrada e saída do interpolador.

Em [3] e [4] chegou-se à conclusão de que se obtêm excelentes interpoladores práticos se os coeficientes  $\{c_i(\mu_n)\}$  forem certas funções polinomiais de curta duração (dois ou quatro intervalos de amostragem<sup>3</sup>). Ou seja, de acordo com a eq. (4) é como se a interpolação fosse realizada com um filtro de resposta impulsional de duração finita (FIR) cujos coeficientes são polinômios. Estes coeficientes só dependem dos intervalos fraccionários  $\mu_n$  e cada um deles é usado em cada um dos  $I_1 + I_2 + 1$  intervalos de amostragem.

<sup>3</sup> O número de pontos (e o número de intervalos de amostragem) deve ser par para que o filtro tenha resposta em fase linear. As durações indicadas,  $2T_s$  e  $4T_s$ , são as mais curtas possíveis.

Interessam-nos apenas interpolações lineares, quadráticas e cúbicas. A interpolação linear requer dois pontos-base (um de cada lado de  $t_n$ ) e as interpolações quadrática e cúbica requerem quatro (dois de cada lado de  $t_n$ ), como se mostra nas respectivas expressões seguintes e na Fig. 5:

$$y(t_n) = c_{-1}(\mu_n)z[(l_n + 1)T_s] + c_0(\mu_n)z(l_n T_s)$$

$$y(t_n) = c_{-2}(\mu_n)z[(l_n + 2)T_s] + c_{-1}(\mu_n)z[(l_n + 1)T_s] + c_0(\mu_n)z(l_n T_s) + c_1(\mu_n)z[(l_n - 1)T_s]$$

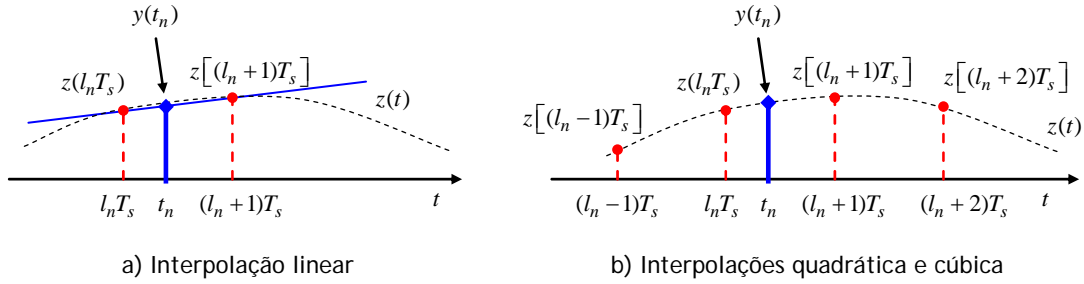


Fig. 5. Pontos-base para a interpolação: a) interpolação linear, dois pontos-base; b) interpolações quadrática e cúbica, quatro pontos-base.

Os polinómios seguintes são os mais adequados:

<b>Interpolador linear</b> (dois pontos-base e polinómios de 1º grau)	<b>Interpolador quadrático</b> (quatro pontos-base e polinómios de 2º grau)	<b>Interpolador cúbico</b> (quatro pontos-base e polinómios de 3º grau)
	$c_{-2}(\mu) = \alpha\mu^2 - \alpha\mu$	$c_{-2}(\mu) = \frac{\mu^3}{6} - \frac{\mu}{6}$
$c_{-1}(\mu) = \mu$	$c_{-1}(\mu) = -\alpha\mu^2 + (1 + \alpha)\mu$	$c_{-1}(\mu) = -\frac{\mu^3}{2} + \frac{\mu^2}{2} + \mu$
$c_0(\mu) = 1 - \mu$	$c_0(\mu) = -\alpha\mu^2 - (1 - \alpha)\mu + 1$	$c_0(\mu) = \frac{\mu^3}{2} - \mu^2 - \frac{\mu}{2} + 1$
	$c_1(\mu) = \alpha\mu^2 - \alpha\mu$	$c_1(\mu) = -\frac{\mu^3}{6} + \frac{\mu^2}{2} - \frac{\mu}{3}$

No interpolador quadrático  $\alpha$  é um parâmetro de projecto de que depende a precisão da interpolação [2]. Deve ser-lhe atribuído um valor próximo de 0,5. Se  $\alpha = 0$  a interpolação torna-se linear.

A soma dos coeficientes é 1 nos três casos, independentemente de  $\mu_n$ . O interpolador linear é o mais simples dos três, naturalmente, e o cúbico o mais complicado. Em muitas situações basta o primeiro para se obter um desempenho adequado; em situações mais críticas consegue-se excelente desempenho com interpolador quadrático e  $\alpha = 0,5$  [4].

### 2.1.1 Cálculo de interpolantes

Em face dos coeficientes indicados os interpolantes são calculados de acordo com as seguintes equações:

- Interpolação linear:

$$y(t_n) = \mu_n z((l_n + 1)T_s) + (1 - \mu_n) z(l_n T_s) \quad (5)$$

- Interpolação quadrática ( $\alpha = 0,5$ ):

$$y(t_n) = 0,5(\mu_n^2 - \mu_n)z[(l_n + 2)T_s] - (0,5\mu_n^2 - 1,5\mu_n)z[(l_n + 1)T_s] + (0,5\mu_n^2 + 0,5\mu_n - 1)z[l_n T_s] + 0,5(\mu_n^2 - \mu_n)z[(l_n - 1)T_s] \quad (6)$$

- Interpolação cúbica:

$$y(t_n) = \left(\frac{\mu_n^3}{6} - \frac{\mu_n}{6}\right)z[(l_n + 2)T_s] - \left(\frac{\mu_n^3}{2} - \frac{\mu_n^2}{2} - \mu_n\right)z[(l_n + 1)T_s] + \left(\frac{\mu_n^3}{2} - \mu_n^2 - \frac{\mu_n}{2} + 1\right)z[l_n T_s] - \left(\frac{\mu_n^3}{6} - \frac{\mu_n^2}{2} + \frac{\mu_n}{3}\right)z[(l_n - 1)T_s] \quad (7)$$

### Exemplo 1

#### Índices de pontos-base e intervalos fraccionários

As amostras  $z(mT_s)$  da tabela seguinte estão espaçadas de  $T_s$  segundos. Queremos usar interpoladores para determinar interpolantes espaçados de  $T$  segundos, em que  $T/T_s = \pi$ .

$m$	0	1	2	3	4	5	6	7
$z(mT_s)$	2,4339	2,4878	2,5364	2,5793	2,6163	2,6474	2,6722	2,6908

Os novos instantes de interpolação ocorrem em  $t_n = T, 2T, 3T, \dots$ , etc. – o primeiro, por exemplo, situa-se entre  $3T_s$  e  $4T_s$ . Os índices de pontos-base e os intervalos fraccionários correspondentes tomam os primeiros valores seguintes:

$n$	1	2	3	4	5	6	7	8
$l_n$	3	6	9	12	15	18	21	25
$\mu_n$	0,14159	0,28319	0,42478	0,56637	0,70796	0,84956	0,99115	0,13274

Como  $T/T_s = \pi$  é um número irracional os intervalos fraccionários nunca se vão repetir. Os valores dos interpolantes que podemos obter até  $7T_s$  são, de acordo com (5), (6) e (7),

Interpolador linear:	$y(T) = 2,5425$	$y(2T) = 2,6544$
Interpolador quadrático ( $\alpha = 0,5$ ):	$y(T) = 2,5432$	$y(2T) = 2,6557$
Interpolador cúbico:	$y(T) = 2,5428$	$y(2T) = 2,6551$

#### 2.1.2 A estrutura de Farrow

A fórmula fundamental do interpolador, eq. (4), apresenta um inconveniente de implementação: dado que  $\mu_n$  varia de instante para instante de interpolação o filtro FIR tem coeficientes  $c_i(\mu_n)$  variáveis (ver Fig. 7a). Se, contudo, rearranjarmos a equação como se mostra a seguir conseguimos determinar os interpolantes usando filtros de coeficientes fixos. Assim, definindo  $c_i(\mu_n) \triangleq \sum_{l=0}^M b_l(i)\mu_n^l$ , em que  $M$  é o grau do polinómio – 1, 2 ou 3 consoante o tipo de interpolação – a eq. (4) passa a escrever-se

$$y(t_n) = \sum_{i=-I_1}^{I_2} \sum_{l=0}^M b_l(i)\mu_n^l z[(l_n - i)T_s] = \sum_{l=0}^M \mu_n^l \sum_{i=-I_1}^{I_2} b_l(i)z[(l_n - i)T_s] = \sum_{l=0}^M \mu_n^l v(l) \quad (8)$$

em que  $v(l) = \sum_{i=-I_1}^{I_2} b_l(i)z[(l_n - i)T_s]$  pode ser interpretado como a saída de um filtro de coeficientes fixos  $b_l(i)$ .

Ao todo passamos a ter  $M+1$  filtros, cada um com  $I_1 + I_2 + 1$  coeficientes, no máximo. Logo, a interpolação linear requer 2 filtros com coeficientes fixos

$$\text{filtro 1:} \quad b_0(-1) = 0 \text{ e } b_1(-1) = 1 \quad (\text{pois } c_{-1}(\mu_n) = b_0(-1) + b_1(-1)\mu_n = \mu_n)$$

$$\text{filtro 2:} \quad b_0(0) = 1 \text{ e } b_1(0) = -1 \quad (\text{pois } c_0(\mu) = 1 - \mu)$$

e as interpolações quadrática e cúbica requerem três e quatro filtros, respectivamente, cada um com quatro coeficientes. Os correspondentes interpolantes são dados respectivamente por

$$y(t_n) = \sum_{l=0}^1 \mu_n^l v(l) = v(0) + \mu_n v(1). \quad (9)$$

$$y(t_n) = \sum_{l=0}^2 \mu_n^l v(l) = v(0) + \mu_n [v(1) + \mu_n v(2)] \quad (10)$$

$$y(t_n) = \sum_{l=0}^3 \mu_n^l v(l) = v(0) + \mu_n \{v(1) + \mu_n [v(2) + \mu_n v(3)]\}. \quad (11)$$

Note-se como diversas expressões estão encaixadas dentro de outras parecidas. Diz-se que as eqs. (9), (10) e (11) representam filtragens com *estrutura de Farrow* [5]. A Fig. 6 ilustra o caso do interpolador cúbico e os exemplos seguintes clarificam o assunto.

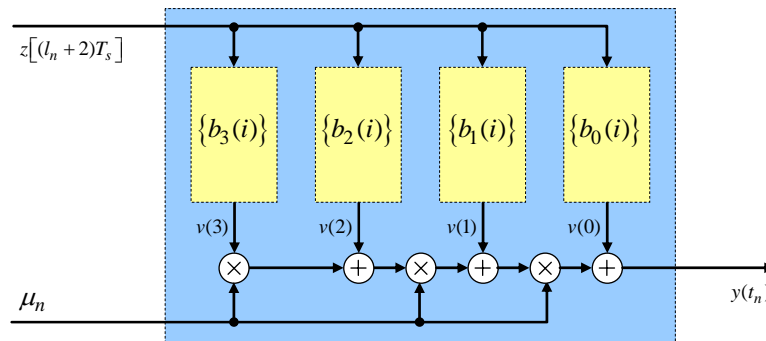


Fig. 6. Estrutura de Farrow para a interpolação cúbica.

## Exemplo 2 Interpolação linear

Reorganizemos a eq. (5) da interpolação linear de modo a obter a estrutura de Farrow da eq. (9):

$$\begin{aligned} y(t_n) &= \underbrace{z(l_n T_s)}_{v(0)} + \mu_n \underbrace{[z((l_n + 1)T_s) - z(l_n T_s)]}_{v(1)} = \\ &= v(0) + \mu_n v(1) \end{aligned}$$

A Fig. 7a) corresponde à eq. (5) e a Fig. 7b) corresponde à eq. (9).

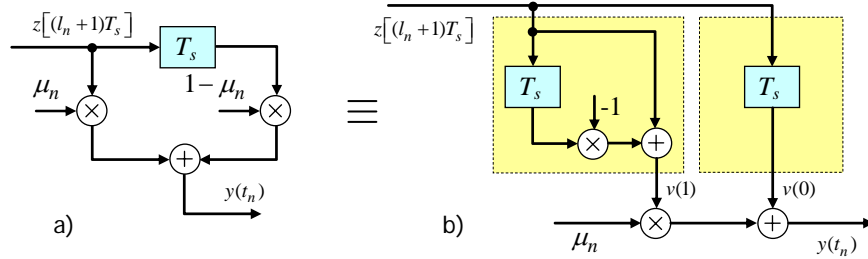


Fig. 7. Filtro para interpolação linear: a) coeficientes variáveis; b) coeficientes fixos (estrutura de Farrow).

### Exemplo 3

#### Interpolação quadrática e cúbica

Com interpolação quadrática e  $\alpha = 0,5$  as saídas dos filtros de Farrow são dadas por

$$v(0) = z(l_n T_s)$$

$$v(1) = -0,5z[(l_n + 2)T_s] + 1,5z[(l_n + 1)T_s] - 0,5z(l_n T_s) - 0,5z[(l_n - 1)T_s]$$

$$v(2) = 0,5z[(l_n + 2)T_s] - 0,5z[(l_n + 1)T_s] - 0,5z(l_n T_s) + 0,5z[(l_n - 1)T_s]$$

e com interpolação cúbica

$$v(0) = z(l_n T_s)$$

$$v(1) = -\frac{1}{6}z[(l_n + 2)T_s] + z[(l_n + 1)T_s] - \frac{1}{2}z(l_n T_s) - \frac{1}{3}z[(l_n - 1)T_s]$$

$$v(2) = \frac{1}{2}z[(l_n + 1)T_s] - z(l_n T_s) + \frac{1}{2}z[(l_n - 1)T_s]$$

$$v(3) = \frac{1}{6}z[(l_n + 2)T_s] - \frac{1}{2}z[(l_n + 1)T_s] + \frac{1}{2}z(l_n T_s) - \frac{1}{6}z[(l_n - 1)T_s]$$

## 2.2 O controlador e a correcção da temporização

Já sabemos calcular interpolantes mas ainda não sabemos como determinar os instantes de interpolação  $t_n$ , que desejamos estejam tão próximos quanto possível dos instantes ideais  $nT/N + \tau$ , isto é, tal como a Fig. 8 sugere  $\hat{t}'_n$  deve ir-se aproximando progressivamente de  $\tau$ . É esta a tarefa do bloco *Controlador* da Fig. 1: em troca da informação que recebe do Detector de Erro de Temporização entrega  $l_n$  e  $\mu_n$  ao interpolador.

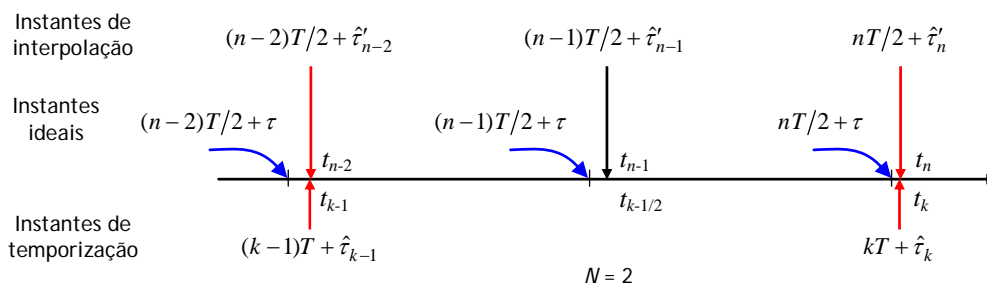


Fig. 8. Instantes de interpolação e de temporização ( $N = 2$ ).



É conveniente introduzir o sinal auxiliar de erro  $e'(n) \triangleq e(k) \Big|_{k=\lfloor n/N \rfloor}$ , obtido do erro de temporização  $e(k)$ , repetindo este  $N$  vezes, de modo que em cada  $T$  segundos temos um só valor de  $e(k)$  e  $N$  valores iguais de  $e'(n)$ , como na Fig. 9.

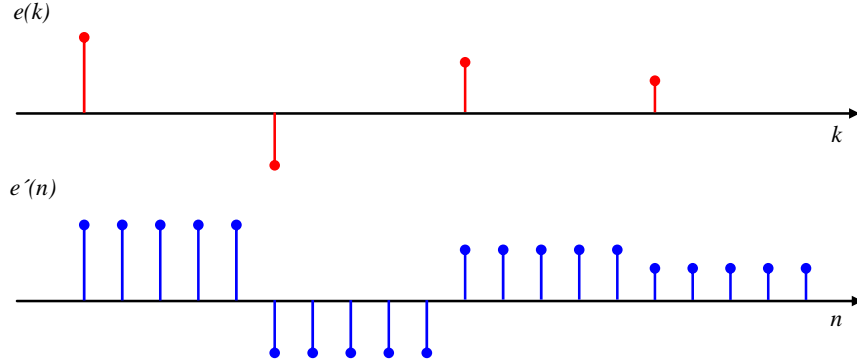


Fig. 9. Relação entre  $e(k)$  e  $e'(n)$  quando  $N = 5$ .

Os instantes de interpolação são actualizados iterativamente de  $T/N$  em  $T/N$  segundos através de

$$t_{n+1} = t_n + \frac{T}{N} + \frac{\gamma}{N} e'(n) \quad (12)$$

em que  $\gamma > 0$  é o passo de adaptação. Com  $\gamma$  pequeno o termo de ajuste  $\frac{\gamma}{N} e'(n)$  é pequeno e o intervalo  $t_{n+1} - t_n$  anda próximo de  $T/N$ . De igual modo, os instantes de temporização de símbolos, dados por  $t_k = kT + \hat{\tau}_k$ , com  $k = \lfloor n/N \rfloor$  e  $0 \leq \hat{\tau}_k < T$ , vão sendo actualizados recursivamente segundo

$$t_{k+1} = t_k + T + \gamma e(k)$$

(logo,  $t_{k+1} - t_k \approx T$  se  $\gamma e(k) \ll 1$ ) e a estimativa  $\hat{\tau}_k$  vai-se aproximando de  $\tau$  de  $T$  em  $T$  segundos de acordo com

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \gamma e(k). \quad (13)$$

Tal como está escrita, a eq. (12) não é a mais conveniente para o cálculo dos interpolantes  $y(t_n)$  pois precisamos de conhecer explicitamente os intervalos fraccionários  $\mu_n$ . O que devemos fazer é utilizar a eq. (2) e substituir  $l_n$  e  $\mu_n$  em (12) obtendo sucessivamente

$$(l_{n+1} + \mu_{n+1})T_s = (l_n + \mu_n)T_s + \frac{T}{N} + \frac{\gamma}{N} e'(n)$$

$$l_{n+1} + \mu_{n+1} = l_n + \mu_n + \frac{T}{NT_s} + \gamma \frac{e'(n)}{NT_s}$$

Tomando as partes inteiras e as partes fraccionárias de ambos os membros chegamos a

$$l_{n+1} = l_n + \left\lfloor \mu_n + \frac{T}{NT_s} + \gamma \frac{e'(n)}{NT_s} \right\rfloor \quad (14)$$

$$\mu_{n+1} = \mu_n + \frac{T}{NT_s} + \gamma \frac{e'(n)}{NT_s} - \left\lfloor \mu_n + \frac{T}{NT_s} + \gamma \frac{e'(n)}{NT_s} \right\rfloor, \quad (15)$$

expressões que o controlador usa para fornecer valores actualizados de  $l_n$  e  $\mu_n$  ao interpolador. Note-se o seguinte:

- Realmente não é necessário actualizar a estimativa  $\hat{\tau}_k$  (eq. (13)) para se obter  $t_k$  ou  $t_n$ , basta actualizar recursivamente  $l_n$  e  $\mu_n$  de acordo com (14) e (15);
- A diferença  $l_{n+1} - l_n = \left\lfloor \mu_n + \frac{T}{NT_s} + \gamma \frac{e'(n)}{NT_s} \right\rfloor$  indica o número, variável, de amostras de entrada<sup>4</sup> deslocadas para dentro do interpolador até o próximo interpolante ser obtido.

#### Exemplo 4

##### Índices de pontos-base, intervalos fraccionários e instantes de interpolação

Deseja-se estimar os instantes de interpolação adequados a uma sequência digital produzida ao ritmo de 50 kbits/s. O amostrador assíncrono do recuperador produz amostras intervaladas de 7  $\mu$ s e o bloco TED produziu a sequência de valores de erro

...0,0603 0,0465 -0,0214 0,0398 0,0377 ...

nas iterações ... 10, 11, 12, 13, 14, ... Sabendo que  $N = 2$ ,  $\gamma = 2 \cdot 10^{-5}$  e  $t_{20} = 195 \mu$ s quais são os valores de  $l_{21}$ ,  $l_{22}$ ,  $\mu_{21}$  e  $\mu_{22}$ ? Quando ocorre o instante  $t_{22}$ ?

R.: A  $k = 10$  e  $11$  correspondem  $n = 20$  e  $22$ , respectivamente. De  $t_{20}$  vemos que  $l_{20} = \lfloor t_{20}/T_s \rfloor = \lfloor 195/7 \rfloor = 27$  e  $\mu_{20} = t_{20}/T_s - \lfloor t_{20}/T_s \rfloor = 0,8571$ . Notando que  $T = 1/50 \cdot 10^3 = 20 \mu$ s,  $e'(20) = e(10) = 0,0603$  e  $e'(21) = e(10) = 0,0603$  calculamos sucessivamente

$$l_{21} = l_{20} + \left\lfloor \mu_{20} + \frac{T}{NT_s} + \gamma \frac{e'(20)}{NT_s} \right\rfloor = 27 + \left\lfloor 0,8571 + \frac{20}{2 \times 7} + 2 \cdot 10^{-5} \times \frac{0,0603}{2 \times 7 \cdot 10^{-6}} \right\rfloor = 27 + \lfloor 2,3719 \rfloor = 29$$

$$\mu_{21} = \mu_{20} + \frac{T}{NT_s} + \gamma \frac{e'(20)}{NT_s} - \left\lfloor \mu_{20} + \frac{T}{NT_s} + \gamma \frac{e'(20)}{NT_s} \right\rfloor = 2,3719 - \lfloor 2,3719 \rfloor = 0,3719$$

$$l_{22} = 29 + \lfloor 1,8866 \rfloor = 30$$

$$\mu_{22} = 1,8866 - \lfloor 1,8866 \rfloor = 0,8866$$

$$t_{22} = (l_{22} + \mu_{22})T_s = 30,8866 \times 7 \cdot 10^{-6} = 216,21 \mu$$

Só falta calcular o erro de temporização, trabalho a cargo do bloco TED, o nosso assunto seguinte.

### 2.3 O Detector de Erro de Temporização (TED)

O que se segue é uma selecção dos mais significativos algoritmos de cálculo do erro  $e(k)$  que são aplicados a esquemas de amostragem assíncrona com e sem realimentação e com e sem ajuda de dados; outros algoritmos podem ser encontrados em [2] e [6]. Suponhamos que  $N = 2$  e que à entrada do bloco TED se apresentam sucessivos interpolantes

$$\dots y(t_{n-2}), y(t_{n-1}), y(t_n), y(t_{n+1}), y(t_{n+2}), \dots$$

colhidos de  $T/2$  em  $T/2$  segundos, aproximadamente<sup>5</sup>, como na Fig. 10.

<sup>4</sup> Amostras colhidas de  $T_s$  em  $T_s$  segundos, não esquecer.

<sup>5</sup> Há uma excepção: no detector de Mueller & Müller, a apresentar, basta recolher amostras de  $T$  em  $T$  segundos.

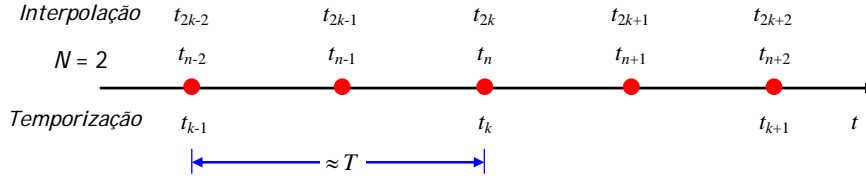


Fig. 10. Instantes de interesse para o TED ( $N = 2$ ).

Nos detectores seguintes os instantes de interpolação vão usar o índice  $n$  ( $t_{n-1}$ ,  $t_n$ , etc.). A Fig. 10 pode servir para reescrever as equações com índices  $k$ , se se preferir<sup>6</sup>.

### 2.3.1 Com realimentação e ajuda de decisões (DD)

Seja  $\hat{c}_k$  a decisão sobre o símbolo  $c_k$  e tome-se em conta a Fig. 11. Vão ser apresentados três detectores: Detector “Early-Late” (ELD), Detector “Zero Crossing” (ZCD) e Detector de Mueller & Müller (MMD).

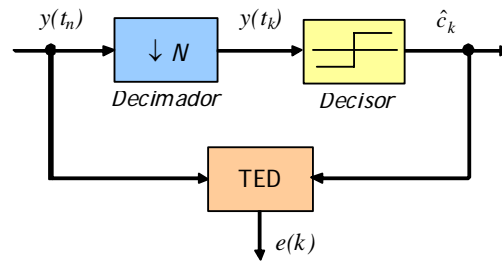


Fig. 11. Esquema genérico com ajuda das decisões.

#### 2.3.1.1 Detector “Early-Late” (ELD)

$$e(k) = \hat{c}_k [y(t_{n+1}) - y(t_{n-1})]$$

São precisas duas amostras por símbolo recolhidas de  $T/2$  em  $T/2$  segundos [2].

#### Exemplo de como mudar de índices

A equação anterior pode escrever-se  $e(k) = \hat{c}_k [y(t_{2k+1}) - y(t_{2k-1})]$ , em que  $t_{2k-1}$  e  $t_{2k+1}$  representam instantes de interpolação, ou  $e(k) = \hat{c}_k [y(t_{k+1/2}) - y(t_{k-1/2})]$ , em que  $t_{k-1/2}$  e  $t_{k+1/2}$  são instantes  $T/2$  segundos antes e depois do instante de temporização  $t_k$ . Uma terceira alternativa, com estimativas  $\hat{t}_{k-1}$  e  $\hat{t}_k$ , é escrever  $e(k) = \hat{c}_k [y(kT + T/2 + \hat{t}_k) - y(kT - T/2 + \hat{t}_{k-1})]$ . Não esquecer, contudo, que as amostras são fornecidas sempre pelo interpolador, independentemente do índice usado.

#### 2.3.1.2 Detector “Zero-Crossing” (ZCD)

$$e(k) = (\hat{c}_{k-1} - \hat{c}_k)y(t_{n-1})$$

São precisas duas amostras por símbolo recolhidas de  $T/2$  em  $T/2$  segundos. Detector proposto por Gardner em 1988 [7].

#### 2.3.1.3 Detector de Mueller & Müller (MMD)

$$e(k) = \hat{c}_{k-1}y(t_n) - \hat{c}_k y(t_{n-2})$$

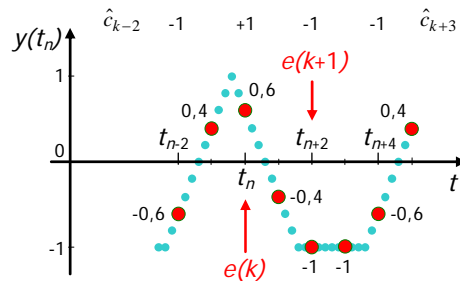
Ao contrário dos algoritmos ELD e ZCD, neste basta uma amostra por símbolo (recolhida de  $T$  em  $T$  segundos). Detector proposto em 1976 [8].

<sup>6</sup> Há que ter cuidado:  $y(t_n)$ ,  $y(t_{2k})$  e  $y(t_k)$ , por exemplo, representam todos o mesmo instante de tempo. Os dois primeiros são um instante de *interpolação* enquanto que o terceiro é um instante de *temporização*.

## Exemplo 5

### Cálculo do erro de temporização com vários TEDs

Na figura seguinte encontram-se valores de interpolantes  $y(t_n)$  e de estimativas de símbolos  $\hat{c}_k$ . Usemos os detectores anteriores para calcularmos o erro de temporização nos instantes de interpolação  $t_n$  e  $t_{n+2}$  – isto é, nos instantes de temporização  $t_k$  e  $t_{k+1}$ .



$$\begin{aligned} \text{ELD: } e(k) &= \hat{c}_k [y(t_{n+1}) - y(t_{n-1})] = (+1) \times (-0,4 - 0,4) = -0,8 \\ e(k+1) &= (-1) \times [-1 - (-0,4)] = 0,6 \end{aligned}$$

$$\begin{aligned} \text{ZCD: } e(k) &= (\hat{c}_{k-1} - \hat{c}_k) y(t_{n-1}) = (-1 - 1) \times 0,4 = -0,8 \\ e(k+1) &= [1 - (-1)] \times (-0,4) = -0,8 \end{aligned}$$

$$\begin{aligned} \text{MMD: } e(k) &= \hat{c}_{k-1} y(t_n) - \hat{c}_k y(t_{n-2}) = (-1) \times 0,6 - (+1) \times (-0,6) = 0 \\ e(k+1) &= (+1) \times (-1) - (-1) \times 0,6 = -0,4 \end{aligned}$$

### 2.3.2 Com realimentação e sem ajuda dos dados (NDA)

Vai ser apresentado apenas um método NDA que, como diz o nome, não requer o conhecimento de dados ou decisões. Trata-se do muito popular detector de Gardner, de 1986.

#### 2.3.2.1 Detector de Gardner [9]

$$e(k) = [y(t_{n-2}) - y(t_n)] y(t_{n-1})$$

Compare-se com o detector ZCD, do mesmo autor: substituíram-se as decisões  $\hat{c}_{k-1}$  e  $\hat{c}_k$  por amostras  $y(t_{k-1})$  e  $y(t_k)$ .

## Exemplo 6

### Interpolador linear e detector de Gardner em Matlab

Seguidamente apresenta-se um excerto de ficheiro Matlab que implementa um interpolador linear e um detector de Gardner. Alguns parâmetros ou variáveis são evidentes ( $z$ ,  $T$ ,  $T_s$ ,  $\text{gama}$ , etc.); outros que o são menos têm os seguintes significados:

```
base(n):      l_n
frac(n):      mu_n
elinha(n):    e'(n)
t_interp(n):  t_n
tau_estim(k): tau_hat_k
```

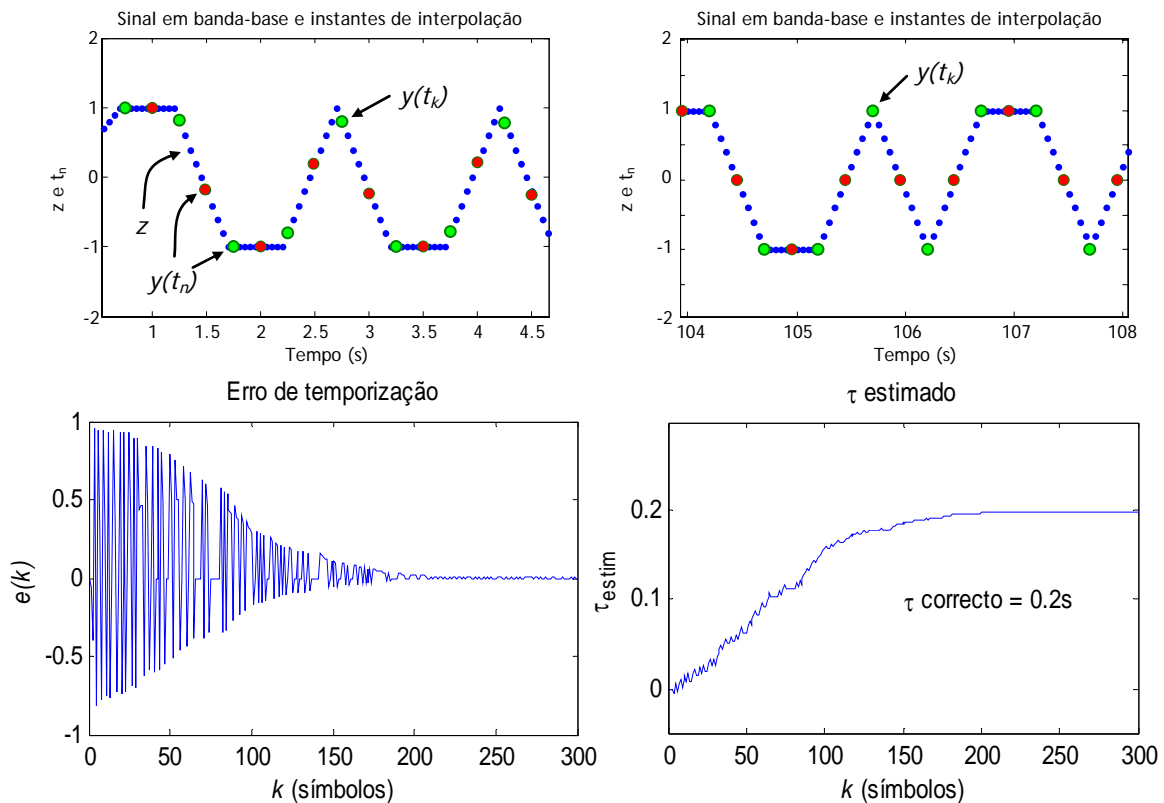
Note-se o uso de  $k = \text{ceil}(n/N)$  em vez de  $k = \text{floor}(n/N)$ . Tal deve-se ao facto de em Matlab os índices começarem em 1 e não em 0.

```

k=1;
for n=2:nmax
    % Determinação do instante de interpolação
    aux=frac(n-1)+T/(N*Ts)+gama*elinha(n-1)/(N*Ts);
    base(n)=base(n-1)+floor(aux);
    frac(n)=aux-floor(aux);
    t_interp(n)=(base(n)+frac(n))*Ts;
    % Cálculo dos interpolantes (interpolação linear)
    y(n)=frac(n)*z((base(n)+1))+(1-frac(n))*z(base(n));
    % Erro de temporização (TED)
    if mod(n,N)~=0
        k=ceil(n/N);
        e(k)= y(n-1)*(y(n-2)-y(n)); % Gardner
        tau_estim(k)=tau_estim(k-1)+gama*e(k-1);
    end;
    elinha(n)=e(k);
end;
yk=y(1:N:nmax); % Valores de T em T segundos a entregar ao decisor

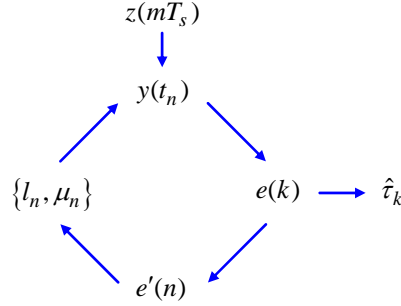
```

As figuras seguintes foram obtidas recorrendo ao "script" Matlab de que o excerto anterior faz parte. Na primeira fila são apresentadas amostras de um sinal NRZ antipodal à saída do filtro adaptado (com os triângulos e trapézios característicos) notando-se bem que no início do processo de recuperação da temporização (gráfico da esquerda) os instantes de interpolação  $t_n$  e os instantes de temporização  $t_k$  estão mal estimados, o que já não acontece na figura da direita, obtida na fase de seguimento. Os gráficos da segunda fila comprovam a convergência e o sucesso da aplicação do método de recuperação com realimentação, em particular do detector de Gardner, que fez evoluir a estimativa  $\hat{\tau}_k$  de 0 para o valor previamente imposto  $\tau = 0,2$  s.



## 2.4 Sequência de operações

Em face do exposto as operações de cálculo regem-se de acordo com o ciclo seguinte:



### 3. Esquema sem realimentação nem ajuda dos dados (NDA)

Até aqui foram apresentados métodos de recuperação de relógio com realimentação. Vamos agora tratar do método de malha aberta, um procedimento não recursivo e mais simples com diagrama de blocos genérico na Fig. 2. Nesta encontramos um *Corrector* e um *Estimador*. Vimos na Sec. 2.1 como funciona o primeiro (mas algo mais será dito) e falta-nos conhecer o segundo.

Dos vários métodos de estimação em malha aberta só vamos tratar de um, o Estimador de Oerder & Meyr.

#### 3.1 Estimador de Oerder & Meyr

Este estimador foi proposto em 1988 e é, nas palavras dos autores, “o equivalente digital da bem conhecida recuperação de temporização em tempo contínuo com lei quadrática” [10]. O método é aplicado a um bloco de símbolos, que origina um valor estimado de  $\tau$ . Em primeiro lugar suponhamos que o sinal recebido atravessa um filtro passa-baixo (que poderá estar adaptado aos impulsos) e é amostrado a uma taxa de pelo menos quatro vezes a taxa de símbolos,  $f_s = 1/T_s \geq 4/T$  (o que quer dizer que  $N_s \geq 4$ ). Se o nosso intervalo de observação contiver  $L$  símbolos – ou seja, se incluir  $N_s L$  amostras e durar  $LT$  segundos – a estimativa de  $\tau$  é calculada através de

$$\hat{\tau} = -\frac{T}{2\pi} \arg \left\{ \sum_{m=0}^{N_s L - 1} z^2(mT_s) e^{-j2\pi m/N_s} \right\}, \quad N_s \geq 4,$$

(o que significa que  $\hat{\tau}$  toma valores entre  $-T/2$  e  $T/2$ ), correspondendo-lhe o diagrama de blocos da Fig. 12. O somatório é o coeficiente da série de Fourier de  $z^2(mT_s)$  à frequência  $1/T$  Hz, aquela que nos interessa.

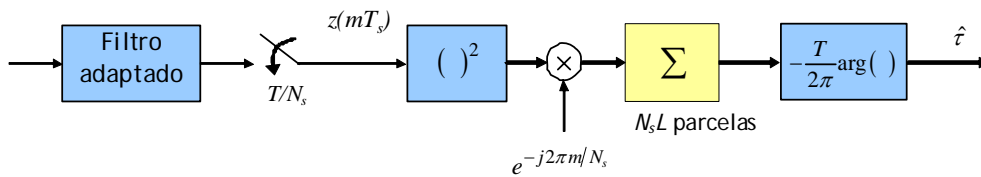


Fig. 12. Diagrama de blocos do estimador de Oerder & Meyr.

#### 3.2 Correção da temporização

Os instantes de temporização dos vários símbolos englobados no intervalo de observação são calculados através de  $t_k = kT + T/2 + \hat{\tau}$  (um aparte: a parcela  $T/2$  aparece porque  $-T/2 \leq \hat{\tau} \leq T/2$ ). Ora, por diversos motivos  $\tau$  pode não ser constante ao longo do tempo e nesse caso é necessário acompanhar-lhe o valor. Assim, em vez de um são usados vários blocos consecutivos de  $N_s L$  amostras (ou outro número): com o bloco de ordem  $i$  obtém-se a estimativa  $\hat{\tau}_i$  e aos símbolos que lhe pertencem correspondem os instantes  $t_k = kT + T/2 + \hat{\tau}_i$ , cada um com seu  $l_k$  e  $\mu_k$  (cf. Fig. 13). Ao *Corrector* da Fig. 2 o *Estimador* entrega então os índices dos pontos-base e os intervalos fraccionários obtidos das expressões iterativas seguintes:

$$l_{k+1} = l_k + \lfloor \mu_k + T/T_s \rfloor \quad (16)$$

$$\mu_{k+1} = \mu_k + T/T_s - \lfloor \mu_k + T/T_s \rfloor \quad (17)$$

Depois é só determinar os interpolantes  $y(t_k)$  como se fez na Sec. 2.1.

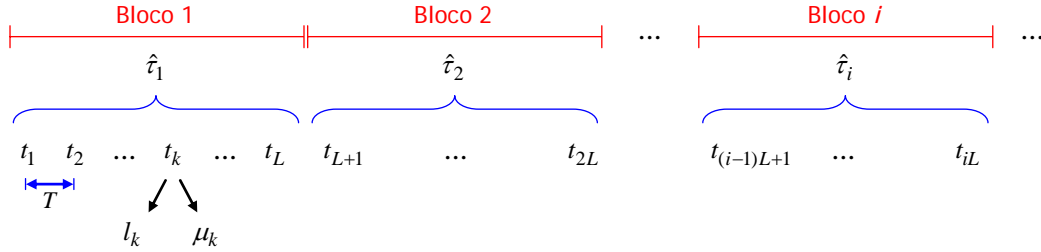


Fig. 13. Blocos de símbolos e instantes de temporização no esquema sem realimentação.

### Exemplo 7 Equações (16) e (17)

Como é que se terá chegado às eqs. (16) e (17)? Facilmente: tendo em conta que  $t_k = kT + T/2 + \hat{\tau} = (l_k + \mu_k)T_s$  então

$$l_k + \mu_k = k \frac{T}{T_s} + \frac{T}{2T_s} + \frac{\hat{\tau}}{T_s} \Rightarrow l_{k+1} + \mu_{k+1} = l_k + \mu_k + \frac{T}{T_s}$$

$$l_{k+1} + \mu_{k+1} = (k+1) \frac{T}{T_s} + \frac{T}{2T_s} + \frac{\hat{\tau}}{T_s}$$

e em seguida, tomando partes inteiras e fraccionárias como em (14) e (15),

$$l_{k+1} = l_k + \lfloor \mu_k + T/T_s \rfloor$$

$$\mu_{k+1} = \mu_k + T/T_s - \lfloor \mu_k + T/T_s \rfloor.$$

### 3.2.1 Transição entre blocos e o “unwrapping problem”

Suponhamos que do bloco  $i$  de símbolos para o seguinte o valor médio de  $\tau$  passou de  $0,48T$  para  $0,51T$ . As estimativas em cada um dos intervalos de observação serão  $\hat{\tau}_i \approx 0,48T$  e...  $\hat{\tau}_{i+1} \approx -0,49T$  (não?! Mas se  $-T/2 \leq \hat{\tau} \leq T/2$  ...). Isto não devia acontecer! Esse salto súbito de cerca de  $T$  segundos de um bloco para o seguinte, que leva a perder ou a duplicar instantes de temporização, resolve-se com o seguinte procedimento, dito de *unwrapping* [11]:

1. A partir de  $\hat{\tau}_i$  obtém-se uma nova estimativa  $\hat{\tau}_i^{(u)}$  de acordo com

$$\hat{\tau}_i^{(u)} = \hat{\tau}_{i-1}^{(u)} + \alpha \text{SAW}(\hat{\tau}_i - \hat{\tau}_{i-1}^{(u)}),$$

em que  $\alpha$  é um parâmetro de projecto e  $\text{SAW}(x)$  é a função dente-de-serra definida por

$$\text{SAW}(x) \triangleq (x - T/2) \bmod T - T/2$$

$$\text{ou, preferindo}^7, \text{SAW}(x) \triangleq x - T \left\lfloor \frac{x}{T} + \frac{1}{2} \right\rfloor.$$

<sup>7</sup> Se o período do dente-de-serra for  $T$  e a amplitude for  $\pm A/2$  então  $\text{SAW}(x) \triangleq A(x/T - \lfloor x/T + 1/2 \rfloor)$ .

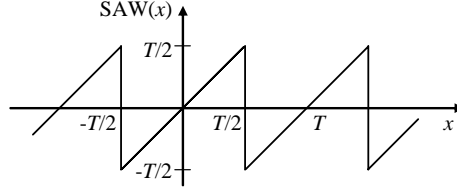


Fig. 14. A função dente-de-serra  $SAW(x)$ .

2. Ao mudar do bloco  $i$  para o bloco  $i+1$  (do instante  $t_{iL}$  para o instante  $t_{iL+1}$ , ver Fig. 13) o índice de ponto-base  $l_{iL+1}$  e o intervalo fraccionário  $\mu_{iL+1}$  calculam-se através de

$$l_{iL+1} = l_{iL} + \left\lfloor \mu_{iL} + \frac{T}{T_s} + \frac{\hat{\tau}_{i+1}^{(u)} - \hat{\tau}_i^{(u)}}{T_s} \right\rfloor$$

$$\mu_{iL+1} = \mu_{iL} + \frac{T}{T_s} + \frac{\hat{\tau}_{i+1}^{(u)} - \hat{\tau}_i^{(u)}}{T_s} - \left\lfloor \mu_{iL} + \frac{T}{T_s} + \frac{\hat{\tau}_{i+1}^{(u)} - \hat{\tau}_i^{(u)}}{T_s} \right\rfloor.$$

Este procedimento só é necessário na transição de blocos (no interior de cada bloco é inócua pois  $\hat{\tau}_i^{(u)} = \hat{\tau}_i$ ).

### Exemplo 8 Unwrapping

Retomando os valores numéricos de trás, se antes tínhamos  $\hat{\tau}_i \approx 0,48T$  e  $\hat{\tau}_{i+1} \approx -0,49T$  agora passamos a ter, com  $\alpha = 1$  e  $\hat{\tau}_i^{(u)} \approx 0,48T$ ,

$$\begin{aligned} \hat{\tau}_{i+1}^{(u)} &= 0,48T + \underbrace{SAW(-0,49T - 0,48T)}_{0,03T} = \\ &= 0,51T \end{aligned}$$

Agora já está certo.

## 3.3 Sequência de operações

Em resumo, na configuração sem realimentação a sequência de operações de cálculo é a seguinte:

$$z(mT_s) \longrightarrow \left( \hat{\tau}_i \rightarrow \hat{\tau}_i^{(u)} \right) \longrightarrow \{t_k\} \longrightarrow \{l_k, \mu_k\} \longrightarrow \{y(t_k)\}$$

## 4. Conclusão

Foram descritas as técnicas de recuperação da temporização de símbolos em tempo discreto e com amostragem assíncrona adequadas à transmissão digital em banda-base. Em primeiro lugar foi abordada a configuração com realimentação e mais tarde a configuração sem realimentação. De permeio foi descrito como, a partir de um sinal amostrado, se determinam amostras do sinal em instantes diferentes dos originais recorrendo a interpolação com filtros de coeficientes polinomiais ou filtros com estrutura de Farrow. Viu-se também como resolver uma questão – o chamado “unwrapping problem” – que surge nas configurações de malha aberta.

No esquema de malha fechada foram apresentados diversos Detectores de Erro de Temporização (TED) – em que uns precisam de conhecer decisões sobre símbolos (DD), como o detector de Mueller & Müller, e outros actuam de forma cega, sem ajuda de dados, como o detector de Gardner. No esquema de malha aberta, onde, pelo contrário, não existe nenhum TED e não é produzido qualquer erro de temporização, foi apresentado o estimador de Oerder & Meyr.



## 5. Referências

- [1] Abrantes, Sílvio A., “Introdução à sincronização em modulações digitais”, FEUP, Março de 2007 (revisto em Abril de 2009). Disponível online em [http://www.fe.up.pt/si/PUBLS\\_PESQUISA.FORMVIEW?P\\_ID=13765](http://www.fe.up.pt/si/PUBLS_PESQUISA.FORMVIEW?P_ID=13765).
- [2] Mengali, U. e D’Andrea, A. N., *Synchronization Techniques for Digital Receivers*, Plenum Press, 1997.
- [3] Gardner, Floyd M., “Interpolation in Digital Modems – Part I: Fundamentals”, *IEEE Transactions on Communications*, Vol. 41, nº 3, pp. 501-507, Março de 1993.
- [4] Erup, L., Gardner, F. M. e Harris, R. A., “Interpolation in Digital Modems – Part II: Implementation and Performance”, *IEEE Transactions on Communications*, Vol. 41, nº 6, pp. 998-1008, Junho de 1993.
- [5] Farrow, Cecil W., “A Continuously Variable Digital Delay Element”, *IEEE International Symposium on Circuits and Systems (ISCAS '88)*, vol.3, pp. 2641-2645, Junho de 1988.
- [6] Abrantes, Sílvio A., “Resumindo e concluindo... Em busca do tempo perdido”, FEUP, Fevereiro de 2009. Disponível online em [http://www.fe.up.pt/si/publs\\_pesquisa.FormView?P\\_ID=17221](http://www.fe.up.pt/si/publs_pesquisa.FormView?P_ID=17221).
- [7] Gardner, Floyd M., “Demodulator Reference Recovery Techniques suited for Digital Implementation“, *European Space Agency*, Final Report, ESTEC Contract No. 6847/86/NL/DG, Agosto de 1988.
- [8] Mueller, K. e Müller, M., “Timing Recovery in Digital Synchronous Data Receivers”, *IEEE Transactions on Communications*, Vol. 24, nº 5, pp. 516-531, Maio de 1976.
- [9] Gardner, Floyd M., “A BPSK/QPSK Timing-Error Detector for Sampled Receivers”, pp. 423-429, Maio de 1986.
- [10] Oerder, M. e Meyr, H., “Digital filter and square timing recovery”, *IEEE Transactions on Communications*, Vol. 36, nº 5, pp. 605-612, Maio de 1988.