

Simulação em Simulink da sincronização em comunicações digitais

© Sílvio A. Abrantes

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

Maio de 2008

Conteúdo

1. Introdução	1
2. Fundamentos teóricos	1
2.1. Recuperação da fase da portadora	1
2.2. Recuperação da temporização de símbolos	2
3. Simulações	4
3.1. Recuperação da fase da portadora	4
3.2. Recuperação da temporização de símbolos	7
3.2.1. Receptor sincronizado	8
3.2.2. Receptor não sincronizado	9
3.2.3. Recuperação de relógio com dispositivo quadrático	10
3.2.4. Recuperação de relógio com o sincronizador "Early-Late"	10
3.2.5. Recuperação de relógio com o método de Gardner	11
4. Referências	12

1. Introdução

A sincronização em comunicações digitais inclui a recuperação da fase de portadoras em modulações digitais e a recuperação da temporização de símbolos. A ferramenta de simulação e cálculo científico Simulink dispõe de blocos próprios de sincronização no seu "Communications Blockset". O objectivo deste texto é exemplificar como esses blocos podem ser usados em modulações lineares como MPSK e QAM.

Antes da descrição das experiências de recuperação de sincronismo é feita uma pequena abordagem teórica para enquadramento do assunto.

2. Fundamentos teóricos

As considerações teóricas seguintes seguem o exposto no texto do autor "Introdução à sincronização em modulações digitais", de Março de 2007 [1].

2.1. Recuperação da fase da portadora

A detecção coerente de sinais modulados exige que a referência local no receptor esteja em fase com o sinal recebido. A estimativa dessa fase pode ser realizada de diversos modos sendo os mais conhecidos os seguintes:

- Recuperação de fase com “Costas Loop”
- Recuperação de fase com dispositivo não-linear
- Estimação de fase de máxima verosimilhança

Simulink recorre a um método híbrido que combina o dispositivo não-linear com a estimação de máxima verosimilhança [2]. Na recuperação de fase com dispositivo não-linear, usada nas modulações MPSK e M-QAM, o sinal recebido é elevado à potência de ordem M e a fase da portadora é estimada por filtragem e realimentação em malha fechada com um PLL. Assim, por exemplo, em BPSK é usado um dispositivo quadrático e em QPSK é usado um “elevador” à quarta potência. A estimação de fase de máxima verosimilhança, ao invés, é um método em malha aberta, tal como a variante usada em Simulink. Nesta o sinal recebido é também elevado à M -ésima potência em MPSK e à quarta potência em QAM, sendo a fase da portadora uma média estimada de acordo com

$$\text{MPSK: } \hat{\theta} = \frac{1}{M} \arg \left[\sum_{k=1}^L z^M(k) \right] \qquad \text{QAM: } \hat{\theta} = \frac{1}{4} \arg \left[-\sum_{k=1}^L z^4(k) \right]$$

em que $z(1), z(2), \dots, z(L)$ são os valores (complexos) dos símbolos recebidos durante um intervalo de observação de L símbolos, intervalo durante o qual se supõe que a fase permanece constante¹, e

$\arg(\mathbf{x}) = \arctg \left[\frac{\text{Im}(\mathbf{x})}{\text{Re}(\mathbf{x})} \right]$ é a fase (ângulo) do complexo \mathbf{x} na sua representação polar. Note-se que ao

contrário de MPSK, na expressão da recuperação da fase em QAM a potência e o denominador são sempre 4, independentemente do número de pontos da constelação². Como o efeito da diferença de fase entre as fases do sinal recebido e da portadora local é uma rotação dos pontos da constelação no espaço de sinal, a fase estimada $\hat{\theta}$ é aproveitada para rodar esses pontos em sentido contrário.

2.2. Recuperação da temporização de símbolos

A recuperação da temporização de símbolos, ou recuperação de relógio, tem como objectivo estabelecer os melhores instantes de amostragem no receptor, determinando com precisão o início e o fim de bits, símbolos, palavras de código, tramas ou outras entidades semelhantes. Vamos debruçar-nos apenas sobre a simulação em Simulink da recuperação da temporização de bits e símbolos.

Admitamos que $R=1/T$ é a taxa de símbolos (número de símbolos por segundo). Deseja-se estimar o mais adequado instante de amostragem do k -ésimo símbolo, $t_k = kT + \hat{t}_k$. A recuperação da temporização do símbolo, \hat{t}_k (e, por consequência, t_k), vai ser simulada usando três métodos:

- com dispositivo quadrático
- com sincronizador “Early-Late”
- com sincronizador de Gardner

Em todos os métodos se vai trabalhar em banda-base. No primeiro o diagrama de blocos para MPSK e QAM é, simplifadamente, o que se apresenta na Fig. 1. Tratando-se de sinais BPSK bastará considerar um dos ramos da figura.

O diagrama de blocos do sincronizador “Early-Late” adequado a um sinal BPSK é apresentado na Fig. 2 na versão com filtro adaptado na entrada. O bloco TED designa o Detector de Erro de Temporização (“Timing Error Detector”) e apresenta na saída o erro de temporização, e_k , para o k -ésimo símbolo. Este último é estimado pelo decisor com base na amostra $z(t_k)$ que lhe é aplicada à entrada.

¹ Recordar-se que numa constelação bidimensional cada símbolo é um complexo com a parte real em fase e a parte imaginária em quadratura.

² Em Simulink é usada a expressão de MPSK para QAM, com $M=4$. É um erro.

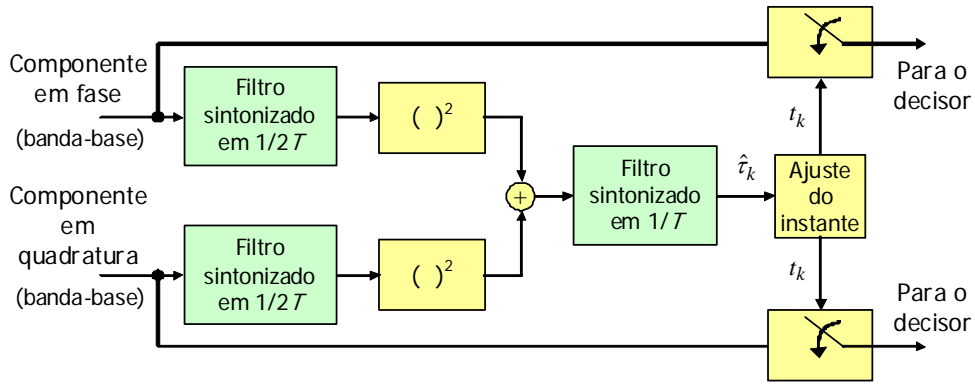


Fig. 1 Sincronizador de símbolos com dispositivo quadrático para sinais MPSK e QAM.

O erro de temporização pode ser calculado de duas maneiras, pelo menos, usando numa a entrada do decisor e na outra a saída. No segundo caso temos

$$e_k = \hat{b}_k [z(t_k - \delta) - z(t_k + \delta)],$$

em que $\hat{b}_k = \pm 1$ é a estimativa do bit b_k , e no primeiro temos

$$e_k = z(t_k) [z(t_k - \delta) - z(t_k + \delta)],$$

como se indica na Fig. 2. É esta a abordagem feita em Simulink, como vamos ver em breve.³

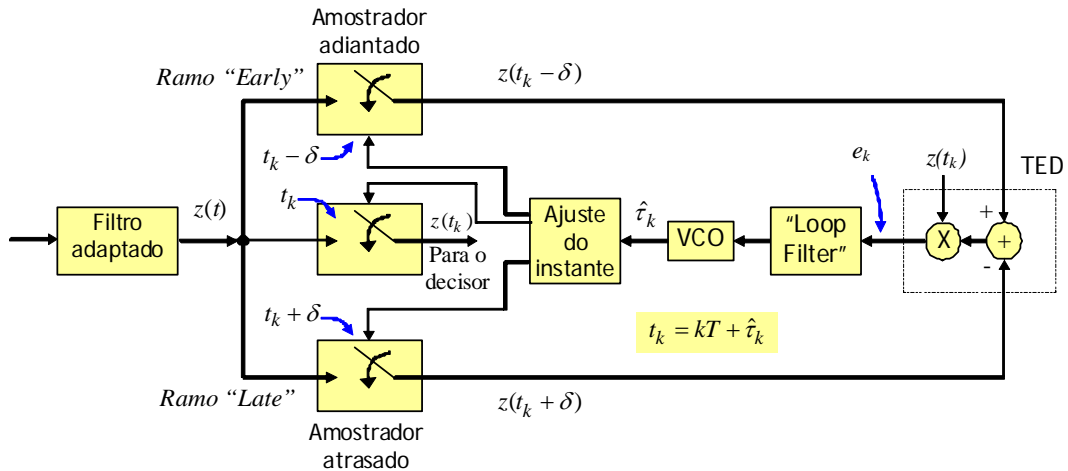


Fig. 2 O sincronizador "Early-Late" com filtro adaptado.

O instante de temporização \hat{t}_{k+1} estimado para o bit de ordem $k + 1$ é obtido actualizando o valor calculado para o bit anterior, \hat{t}_k , de acordo com a expressão

$$\hat{t}_{k+1} = \hat{t}_k - \mu e_k,$$

onde $\mu > 0$ é o *passo de adaptação*.

Tratando-se de sinais MPSK ou QAM, cujo espaço de sinal é bidimensional, temos de duplicar o diagrama da Fig. 2 de maneira a incluir os ramos em fase e em quadratura. O bloco Simulink do sincronizador "Early-Late" usa as seguintes equações de actualização do instante de temporização, com $\delta = T/2$ e sendo $z_I(\cdot)$ e $z_Q(\cdot)$ as entradas em fase e em quadratura, respectivamente:

³ Também é vulgar calcular e_k de acordo com $e_k = |z(t_k - \delta)| - |z(t_k + \delta)|$.

$$\begin{aligned}
a_I(k) &= z_I(kT + \hat{\tau}_k) [z_I(kT - T/2 + \hat{\tau}_{k-1}) - z_I(kT + T/2 + \hat{\tau}_k)] && \text{(ramo em fase)} \\
a_Q(k) &= z_Q(kT + \hat{\tau}_k) [z_Q(kT - T/2 + \hat{\tau}_{k-1}) - z_Q(kT + T/2 + \hat{\tau}_k)] && \text{(ramo em quadratura)} \\
e_k &= a_I(k) + a_Q(k) \\
\hat{\tau}_{k+1} &= \hat{\tau}_k - \mu e_k
\end{aligned}$$

O diagrama de blocos do sincronizador de Gardner é apresentado na Fig. 3, onde os amostradores da entrada trabalham a uma cadência superior a $1/T$ (tomando, por exemplo, 4 amostras por símbolo).

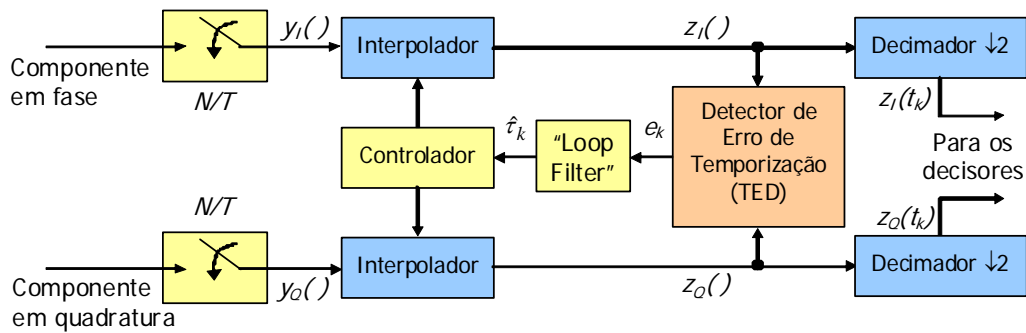


Fig. 3 Diagrama de blocos do método de sincronização de símbolo de Gardner.

As equações de actualização da temporização usadas pelo Simulink neste método são as seguintes:

$$\begin{aligned}
a_I(k) &= \{z_I(kT + \hat{\tau}_k) - z_I[(k-1)T + \hat{\tau}_{k-1}]\} z_I(kT - T/2 + \hat{\tau}_{k-1}) \\
a_Q(k) &= \{z_Q(kT + \hat{\tau}_k) - z_Q[(k-1)T + \hat{\tau}_{k-1}]\} z_Q(kT - T/2 + \hat{\tau}_{k-1}) \\
e_k &= a_I(k) + a_Q(k) \\
\hat{\tau}_{k+1} &= \hat{\tau}_k - \mu e_k
\end{aligned}$$

O passo de adaptação é normalmente inferior a $1/N$, em que N é o número de amostras por símbolo recolhidas pelos amostradores de entrada.

3. Simulações

As experiências que se seguem pressupõem a utilização da versão Matlab 7.3.0/Simulink 6.5 (R2006b) e tiveram como ponto de partida os exemplos de sincronização apresentados no *Help* de Simulink. Os blocos Simulink a usar irão sendo sucessivamente identificados e localizados nas suas bibliotecas as quais, excepto se mencionado, pertencem ao *Communications Blockset*. Nos parâmetros de configuração das simulações deve escolher-se *Solver ode45 (Dormand Prince)* do tipo *Variable Step*.

3.1. Recuperação da fase da portadora

Vamos proceder à recuperação da fase da portadora de um sinal 8-PSK. Para isso usaremos o bloco *M-PSK Phase Recovery*, que combina um dispositivo não linear e a estimação de máxima verosimilhança. Sendo $M=8$, o sinal recebido deve, portanto, ser elevado à oitava potência. Consideremos então o modelo da Fig. 4 onde, no emissor, temos um modulador 8-PSK em banda-base e um filtro de formatação de raiz de cosseno elevado. O sinal atravessa um canal AWGN de ruído desprezável e no receptor passa por um idêntico filtro de raiz de cosseno elevado (de modo que, no conjunto, os dois filtros são equivalentes a um único filtro de cosseno elevado). Ainda no emissor, é introduzido um esfasamento deliberado de 10° através do bloco *Phase/Frequency Offset*, valor que vai ser recuperado e corrigido pelo bloco *M-PSK Phase Recovery*.

MPSK com erro de fase na portadora

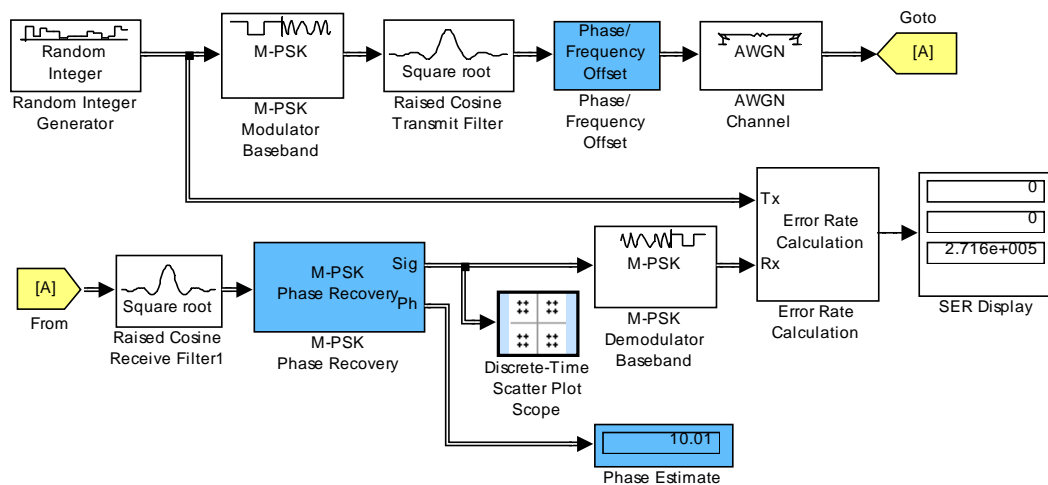


Fig. 4 Modelo Simulink de recuperação da fase da portadora de MPSK com dispositivo não-linear.

Especificações do sistema:

- Duração de cada símbolo: $T = 1/100$ s ($R = 100$ símbolos/s)
- Modulação: 8-PSK com mapeamento binário
- Factor de *roll-off* dos filtros de cosseno elevado: $\alpha = 0,2$
- Atraso introduzido por cada filtro: 4 símbolos
- Ruído AWGN: SNR = 40 dB com potência do sinal de entrada = 1 W. Com este valor elevado da relação sinal-ruído aproximamo-nos da situação em que o ruído é desprezável.
- Número de amostras por símbolo nos filtros e nas formas de onda: $N = 8$
- Número de símbolos por trama: 100.
- Diferença de fase a estimar no recuperador: 10° .

Localização e parâmetros dos blocos Simulink necessários à simulação:

- Random Integer Generator: *Comm Sources* → *Random Data Sources*
M-ary number = 8.
Sample time = 1/100.
Frame-based outputs.
Samples per frame = 100.
- M-PSK Modulator Baseband e M-PSK Demodulator Baseband: *Modulation* → *Digital Baseband* → *PM*
M-ary Number = 8.
Phase offset (rad) = 0.
- Raised Cosine Transmit Filter: *Comm Filters*
Group delay = 4.
Rolloff factor = 0.2.
Upsampling factor = 8
- Raised Cosine Receive Filter: *Comm Filters*
Input samples per symbol = 8.
Group delay = 4.
Rolloff factor = 0.2.
Output mode = Downsampling.
Downsampling factor = 8
- Phase/Frequency Offset: *RF Impairments*
Frequency offset (Hz) = 0.
Phase offset (deg) = 10.
- AWGN Channel: *Channels*
Mode = Signal to noise ratio (SNR).

- SNR (dB) = 40.
- Discrete-Time Scatter Plot Scope: *Comm Sinks*
Samples per symbol = 1.
- Goto e From: *Signal Routing (Simulink)*
Usar os parâmetros por omissão.
- MPSK Phase Recovery: *Synchronization* → *Carrier Phase Recovery*
M-ary Number = 8.
Observation Interval (symbols) = 300.
- Error Rate Calculation: *Comm Sinks*
Receive delay = 8. É o atraso do par de filtros de raiz quadrada de cosseno elevado.
Computation delay = 300. Corresponde ao intervalo de observação estabelecido em MPSK Phase Recovery.
Output data = Port.
- Display: *Sinks (Simulink)*.
Incluem-se aqui os blocos redenominados SER Display e Phase Estimate.

Encontrámos atrás a equação de estimação da fase $\hat{\theta} = \frac{1}{M} \arg \left\{ \sum_{k=1}^L [z(k)]^M \right\}$. Vemos agora que o nosso intervalo de observação vale $L=300$ símbolos pelo que, nesta simulação concreta, a expressão anterior se reescreve como

$$\hat{\theta} = \frac{1}{8} \arg \left\{ \sum_{k=1}^{300} [z(k)]^8 \right\}.$$

De um modo geral vamos lidar com símbolos excepto quando usarmos formas de onda, como nos filtros; nestes casos teremos de considerar várias amostras por símbolo. No modelo de simulação proposto o filtro do emissor faz a sobreamostragem do sinal tomando 8 amostras/símbolo e o filtro do receptor faz a correspondente subamostragem por 8, repondo a cadência inicial.

O recuperador de fase da portadora vai “apontar” para o esfasamento total que lhe chega, resultante da soma de “Phase offset” (do modulador) com “Phase offset” (do bloco *Phase/Frequency Offset*). Como o que se deseja é estimar o último valor, para evitar erro de estimação é necessário que no modulador se escolha “Phase offset” = 0.

Só após o intervalo de observação de 300 símbolos é que o bloco de recuperação da fase roda a constelação e a entrega devidamente posicionada ao desmodulador na saída *Sig*. A Fig. 5 mostra duas situações, captadas no Discrete-Time Scatter Plot Scope antes e depois da rotação.

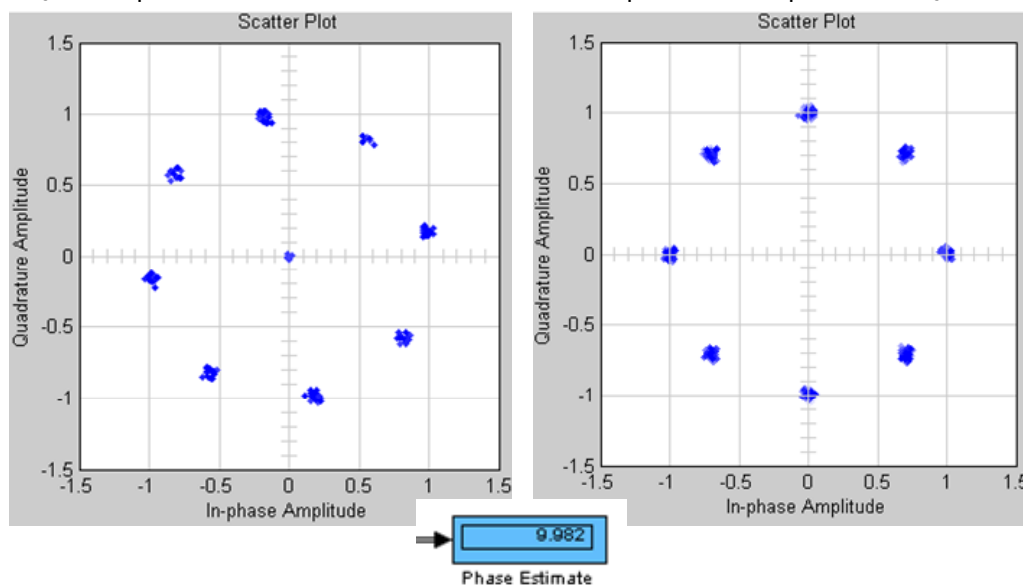


Fig. 5 Símbolos recebidos pelo desmodulador de 8-PSK antes (esquerda) e depois da correcção da fase da portadora.

A estimativa de fase apresentada na figura, em baixo, foi obtida num certo intervalo de observação no momento, já em regime permanente, em que se suspendeu a simulação e está bastante próxima dos 10° impostos no início.

3.2. Recuperação da temporização de símbolos

Desta vez vamos considerar um sinal QAM. Tal como atrás, o sinal é formatado no emissor por um filtro de raiz de cosseno elevado, atravessa um canal AWGN de baixo ruído e passa por idêntico filtro de raiz de cosseno elevado no receptor. Supondo sempre que estamos a trabalhar em banda-base iremos observar situações com e sem recuperação de temporização.

Especificações do sistema:

- Duração de cada símbolo: $T = 1/100$ s ($R = 100$ símbolos/s)
- Modulação: 16-QAM com mapeamento binário
- Factor de *roll-off* dos filtros de cosseno elevado: $\alpha = 0,2$
- Atraso introduzido por cada filtro: 4 símbolos
- Ruído AWGN: SNR = 40 dB com potência do sinal de entrada = 1 W
- Número de amostras por símbolo nos filtros e nas formas de onda: $N = 8$
- Número de símbolos por trama: 100.

O modelo Simulink que nos vai servir de base é o seguinte:

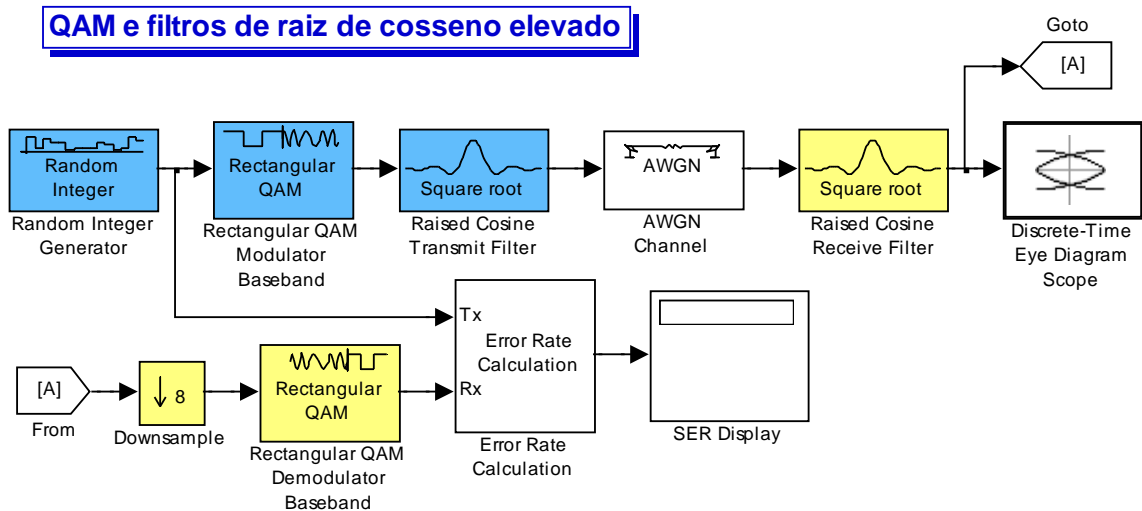


Fig. 6 Diagrama-base do sistema QAM para recuperação de relógio.

Alguns dos blocos da figura já foram usados anteriormente. Indicam-se em seguida os parâmetros e a localização dos novos blocos, bem como os novos parâmetros de blocos antigos.

- Random Integer Generator: *Comm Sources* → *Random Data Sources*
M-ary number = 16.
Sample time = 1/100.
Frame-based outputs.
Samples per frame = 100.
- Rectangular QAM Modulator Baseband e Rectangular QAM Demodulator Baseband: *Modulation* → *Digital Baseband* → *AM*
M-ary number = 16.
Normalization method = Peak Power.
Peak power = 1.
- Raised Cosine Receive Filter: *Comm Filters*
Input samples per symbol = 8.
Group delay = 4.
Rolloff factor = 0.2.
Output mode = None.

- Discrete-Time Eye Diagram Scope: *Comm Sinks*
 Symbols per trace = 2.
 Traces displayed = 100.
 Samples per symbol = 8.
- Downsample: *Signal Operations (Signal Processing Blockset)*
 Downsample factor = 8.
- Error Rate Calculation: *Comm Sinks*
 Receive delay = 8. É o atraso do par de filtros de raiz quadrada de cosseno elevado.
 Computation delay = 0.
 Output data = Port.

Ao abordarmos a questão da recuperação de fase fez-se a sobreamostragem do sinal no filtro do emissor tomando 8 amostras/símbolo e a subamostragem por 8 no filtro do receptor, repondo os símbolos. Se, contudo, o filtro do receptor for seguido de um recuperador de temporização é recomendado que essa subamostragem não seja feita (pois disso se encarregará o recuperador de temporização). Daí que no bloco *Raised Cosine Receive Filter* se tenha escolhido agora *Output mode = None*. Como no modelo da Fig. 6 (e da Fig. 7, mais adiante) ainda não estamos a fazer a recuperação de relógio, é necessário colocar um bloco *Downsample* (com redução da amostragem a 1/8) à saída do filtro do receptor. Quando passarmos a usar blocos de recuperação de relógio retiraremos o bloco *Downsample*.

Em qualquer dos métodos de recuperação de relógio que abordaremos o bloco recuperador usa os 100 símbolos de cada trama para estimar o atraso e escolher o instante de amostragem desses 100 símbolos. A estimativa do atraso varia de trama para trama e com ela o instante de amostragem.

3.2.1. Receptor sincronizado

No modelo da Fig. 6 o canal não introduz atraso e o receptor está sincronizado com o emissor. Além disso, com SNR = 40dB no canal a probabilidade de símbolo errado é praticamente nula. Obtêm-se os diagramas de olho da Fig. 7, onde é bem visível que não há interferência intersimbólica e que os instantes de amostragem óptimos se situam em 5 ms, 15 ms, etc. Assim deveria ser pois o tempo de símbolo é 10 ms e a amostragem deverá fazer-se a meio do símbolo (detecção no ponto central).

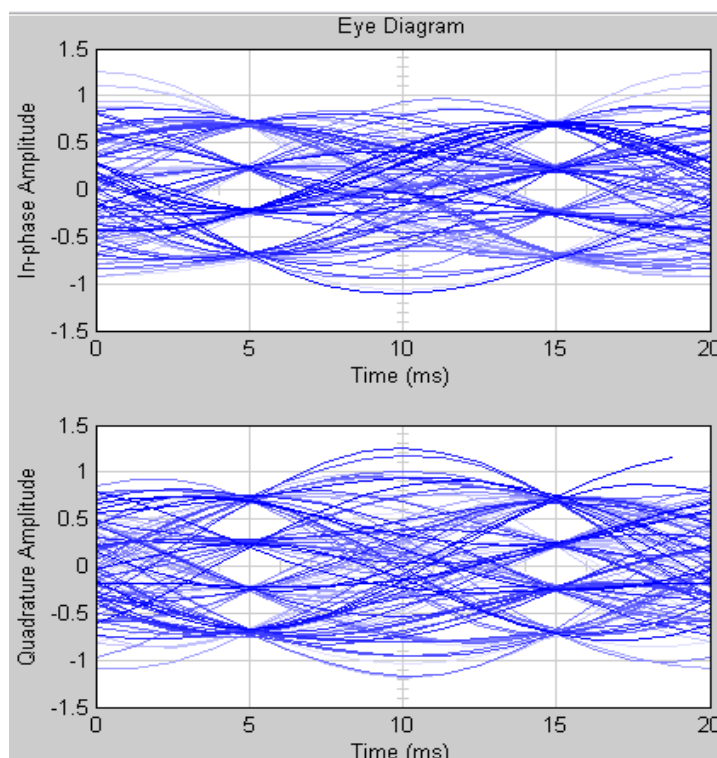


Fig. 7 Diagrama de olho do receptor sincronizado.

3.2.2. Receptor não sincronizado

No modelo seguinte, na Fig. 8, vamos introduzir um certo atraso τ , medido em amostras, para o qual se obtêm os diagramas de olho apresentados na Fig. 9. Precisaremos de dois novos blocos:

- Variable Fractional Delay: *Signal Operations (Signal Processing Blockset)*
Usar os parâmetros por omissão.
- Constant: *Sources (Simulink)*
Constant = 2.5. É o número de amostras de atraso introduzido pelo sistema

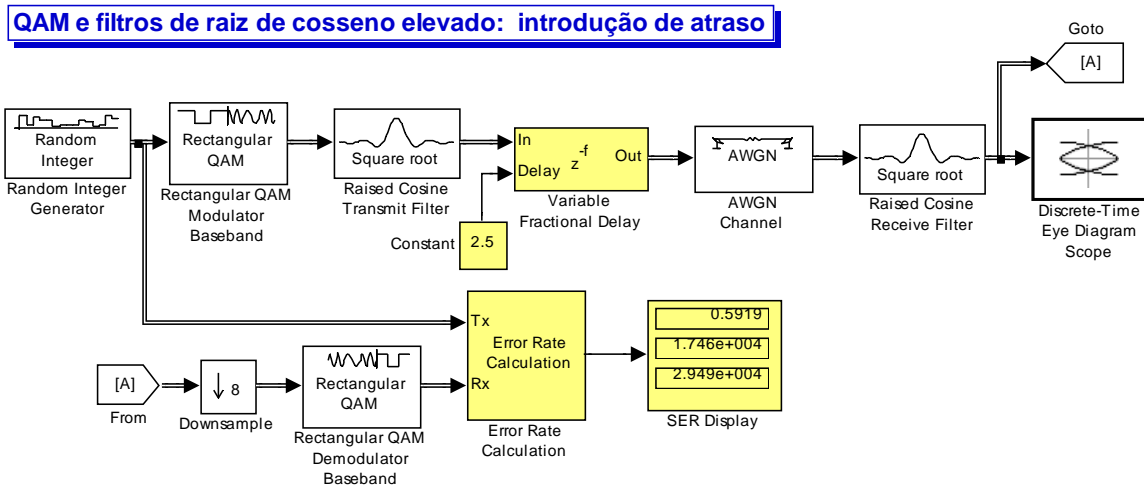


Fig. 8 Modelo Simulink com introdução de atraso na propagação.

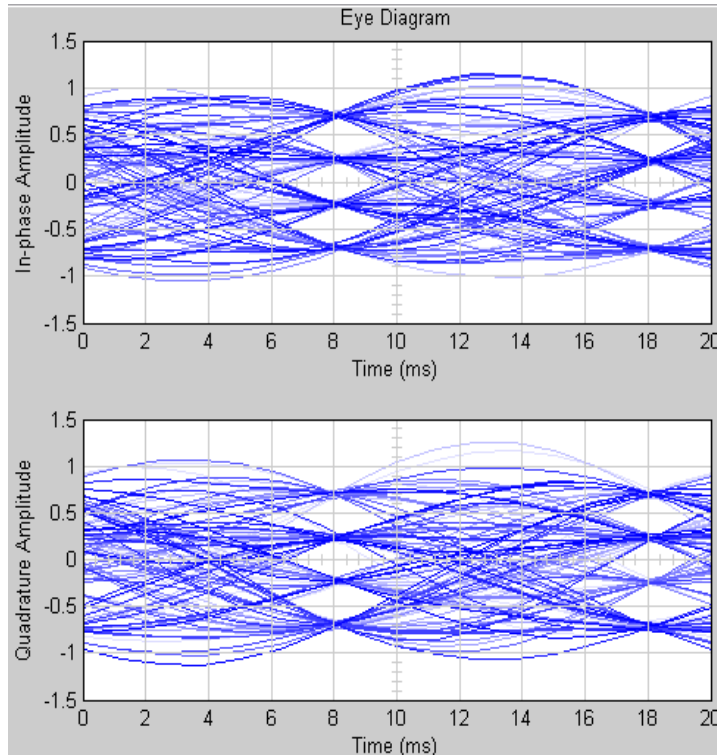


Fig. 9 Diagrama de olho do receptor não-sincronizado.

Se cada símbolo dura 10 ms e é representado por 8 amostras, um atraso de 2,5 amostras corresponderá nos tempos a $\frac{2,5}{8} \times 10\text{ms} = 3,125\text{ ms}$. Genericamente falando, um atraso em amostras,

τ , traduz-se num atraso temporal de $\frac{\tau}{N}T$ segundos, em que N é o número de amostras/símbolo e T é a duração de cada símbolo.

Claro que agora os instantes de amostragem óptimos já não são os mesmos de há pouco e, por causa do atraso imposto, estão situados em $5 + 3,125 = 8,125$ ms, $15 + 3,125 = 18,125$ ms, etc., como os diagramas de olho denunciam. No entanto, o decisor do desmodulador toma as suas decisões nos instantes $t = 5$ ms, 15 ms, 25 ms, ... , como antes, e por isso vão ocorrer muitíssimos erros de decisão ($P_e = 0,59$ ao fim de 3 milhões de símbolos). Esta grave situação era previsível pois o erro de temporização é percentualmente muito elevado: $3,125/10 = 31,25\%$

3.2.3. Recuperação de relógio com dispositivo quadrático

O modelo Simulink da Fig. 10 efectua a recuperação da temporização de símbolo através de um dispositivo quadrático. Note-se a ausência anunciada do bloco *Downsample*.

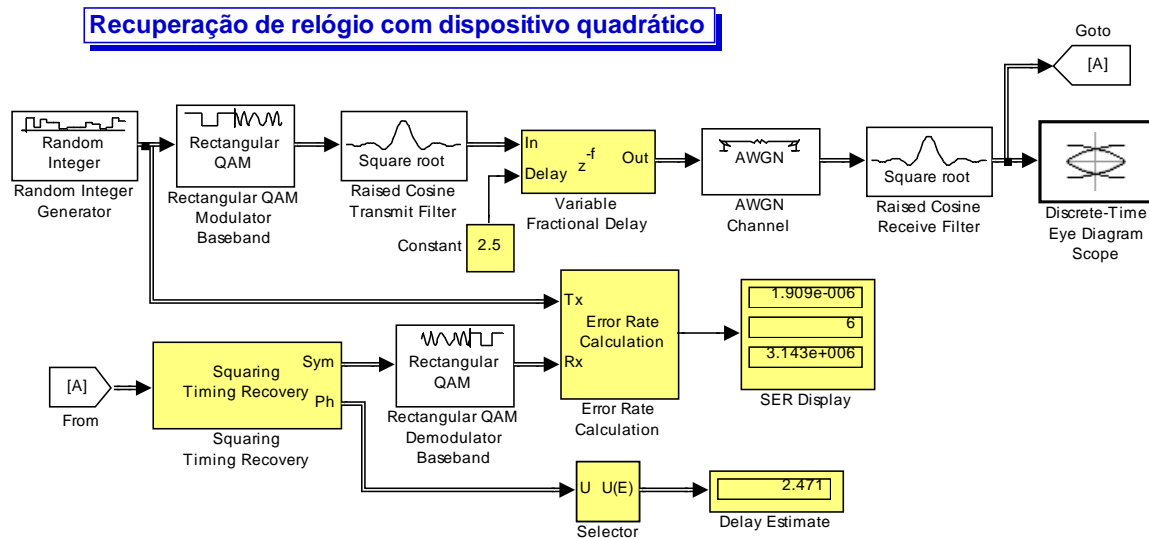


Fig. 10 Modelo Simulink com recuperação de relógio por dispositivo quadrático.

Além do bloco *Delay Estimate* (um vulgar bloco *Display* a que se deu outro nome) os novos blocos são os seguintes:

- Squaring Timing Recovery: *Synchronization* → *Timing Phase Recovery*
Symbols per frame = 100.
Samples per symbol = 8.
- Selector: *Signal Routing (Simulink)*
Elements = 1. Faz com que o bloco seleccione o primeiro elemento da trama (cujos elementos, aliás, são todos iguais)
Input port width = 100.

A probabilidade de símbolo errado muito pequena mostra a eficácia do método de sincronização, também evidenciada na Fig. 10 pela estimativa do atraso.

3.2.4. Recuperação de relógio com o sincronizador "Early-Late"

No modelo anterior vamos substituir o bloco *Squaring Timing Recovery* pelo bloco *Early-Late Gate Timing Recovery* (ver Fig. 11):

- Early-Late Gate Timing Recovery: *Synchronization* → *Timing Phase Recovery*
Samples per symbol = 8.

Este bloco introduz um atraso de 3 símbolos quando a sua entrada se baseia em amostras ("sample-based input") e de 2 símbolos quando a entrada se baseia em tramas ("frame-based

input”), como é o caso aqui. Este atraso adicional deve ser tido em conta no bloco de contagem de erros: em vez do atraso de 8 símbolos até aqui usado nesse bloco, agora o atraso é de 10 símbolos. Neste mesmo bloco de contagem de erros poderá estabelecer-se um valor superior a zero para o parâmetro *Computation delay* (100, por exemplo) para evitar contar os erros excessivos que sempre ocorrem no período de aquisição inicial.

Recuperação de relógio com sincronizador "Early-Late"

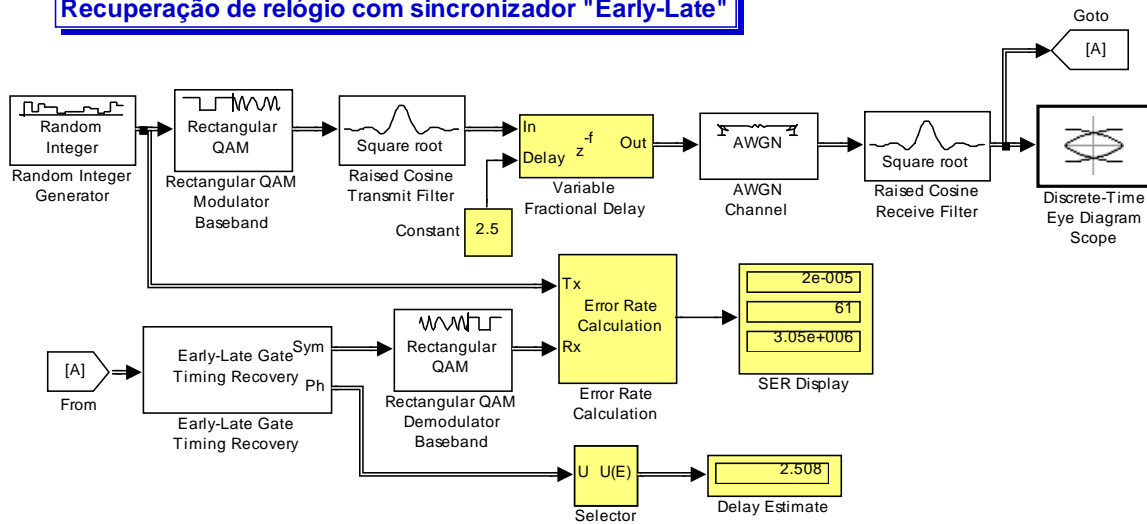


Fig. 11 Modelo Simulink com recuperação de relógio através do sincronizador "Early-Late".

Também com este método a probabilidade de erro é baixa, como se observa na figura, comprovando de novo a sua eficácia. A própria figura indica também a boa estimativa do atraso.

3.2.5. Recuperação de relógio com o método de Gardner

Usemos agora o bloco *Gardner Timing Recovery* (Fig. 12):

- Gardner Timing Recovery: *Synchronization* → *Timing Phase Recovery*
Samples per symbol = 8.

Este bloco introduz um atraso de 2 símbolos com entradas "frame-based", tal como o bloco *Early-Late Gate Timing Recovery*.

Recuperação de relógio com o método de Gardner

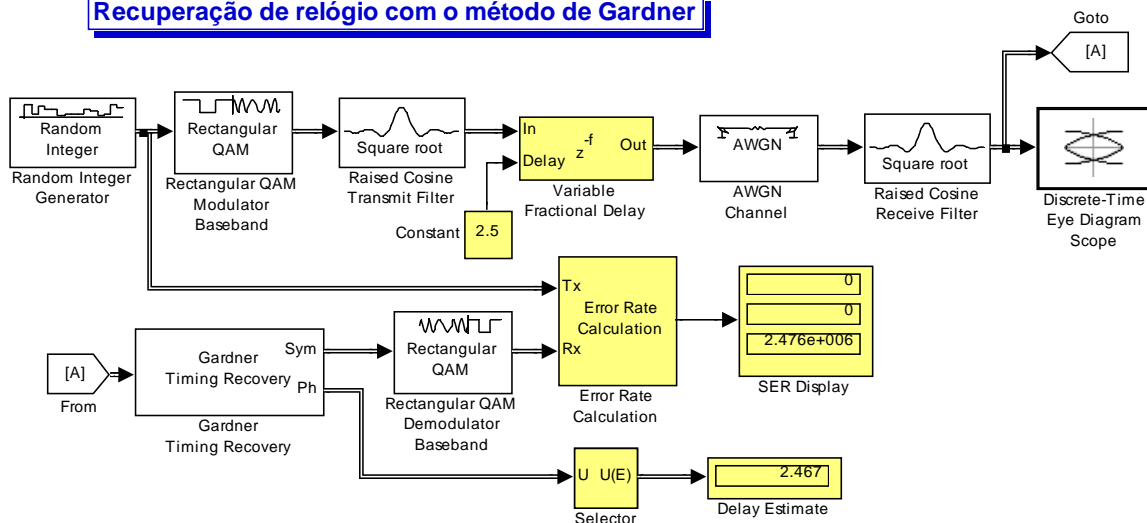


Fig. 12 Modelo Simulink com recuperação de relógio através do método de Gardner.

Chegamos às mesmas conclusões que anteriormente.

4. Referências

- [1] Sílvio A. Abrantes, “Introdução à sincronização em modulações digitais”, Março de 2007, disponível *online* em http://www.fe.up.pt/si/PUBLS_PESQUISA.FORMVIEW?P_ID=13765.
- [2] Marc Moeneclaey and Geert de Jonghe, “ML-Oriented NDA Carrier Synchronization for General Rotationally Symmetric Signal Constellations”, *IEEE Transactions on Communications*, vol. 42, nº 8, Agosto de 1994, pp. 2531-2533.