



UNIVERSIDADE FEDERAL DO PARANÁ

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA – PPGEE**

ESTIMAÇÃO PARAMÉTRICA UTILIZANDO ALGORITMO RLS

Rodrigo Urban de Moraes

CURITIBA, JUNHO DE 2007.

Sumário

1. Introdução	3
1.1. Filtragem Adaptativa	3
1.2. Sistema Adaptativo	3
1.3. Filtro FIR	5
1.4. Solução ótima de Wiener	6
2. Algoritmos Adaptativos	7
3. Algoritmo Recursivo de Mínimos Quadráticos (RLS - <i>Recursive Least Square</i>)	9
4. Exemplo de Aplicação: Estimação de Potência em circuitos CMOS	12
4.1 Estimação de potência	13
4.2. Resultados	15
4.5. Conclusões	18
5. Referências Bibliográficas	19

1. Introdução

1.1. Filtragem Adaptativa

A filtragem adaptativa constitui uma ferramenta fundamental no processamento de sinais digitais. Ela é aplicada atualmente em um grande número de problemas de engenharia. Esta técnica tem sido explorada com sucesso em problemas de Economia, Engenharia Biomédica, Equalização de Canais, Sistemas de Controle e em Telecomunicações.

O estudo da filtragem adaptativa ganhou impulso com o desenvolvimento do algoritmo LMS por Widrow e Hopf em 1959, e tem recebido considerável atenção de muitos pesquisadores nas últimas décadas. Este interesse deve-se ao fato de muitos problemas práticos não poderem ser resolvidos de maneira satisfatória através da utilização de filtros digitais.

Assim, o uso de filtros adaptativos permite a expansão das capacidades de processamento que não seriam possíveis de outra forma. Observando a figura 1.1 percebemos que o processo de um sistema de filtragem adaptativa consiste na utilização de um filtro adaptativo para processar sinais resultantes de ambientes com estatísticas estocásticas, fornecendo na saída um sinal tratado que atende aos critérios do filtro adaptativo.



Figura 1.1: Sistema de Filtragem Adaptativa

1.2. Sistema Adaptativo

Um sistema adaptativo é aquele cuja estrutura é alterável (através do ajuste dos seus coeficientes) de tal forma que seu comportamento melhore de acordo com algum critério de desempenho através da exposição ao ambiente no qual será inserido.

O ajuste dos coeficientes do filtro adaptativo é realizado através da implementação de um algoritmo, devidamente escolhido, cujo objetivo é atender a requisitos do sistema. Estes algoritmos são definidos como algoritmos adaptativos. Segundo Widrow e Stearns os sistemas adaptativos possuem todas ou algumas destas características abaixo listadas:

1. Adaptação automática à medida que ocorre a modificação do ambiente e/ou mudança das necessidades do sistema (auto-otimização);
2. Possibilidade de serem treinados para desenvolver uma tarefa específica de filtragem ou decisão, ou seja, podem ser programados através de um processo de treinamento (auto-programáveis);
3. Em virtude disso, não precisam de procedimentos elaborados de síntese (são basicamente auto-programáveis);
4. Podem extrapolar o espaço de conhecimento e lidar com novas situações após o treinamento com um pequeno conjunto de padrões de entrada (auto-aprendizado);
5. Até certo ponto podem reparar a si mesmos, ou seja, podem adaptar-se em regiões próximas da ótima mesmo quando sujeitos a certos tipos de defeitos ou limitações;
6. Geralmente são mais complexos e difíceis de analisar que sistemas não adaptativos, mas oferecem a possibilidade de um desempenho substancialmente melhor quando as características do ambiente são desconhecidas ou variantes no tempo.

A operação de um sistema de filtragem adaptativa envolve dois processos básicos: o processo de filtragem com o objetivo de produzir uma saída em resposta a uma seqüência de entrada, e o processo adaptativo. O propósito é fornecer um mecanismo para ajuste de um conjunto de parâmetros utilizados no processo de filtragem.

A estrutura de um sistema adaptativo é representada pela figura 1.2. Neste sistema sua saída é modificada de tal forma que seu comportamento sofre alterações de acordo com o critério desejado. Neste exemplo observa-se um sistema em diagrama de blocos representando um sistema adaptativo, com uma seqüência de entrada $x(n)$, o filtro *FIR* e a saída $y(n)$: um sinal $d(n)$ que representa a resposta desejada, e que subtraindo do sinal $y(n)$ gera o valor de erro e , que é o valor da diferença entre o sinal desejado e a estimativa do filtro. A equação 1.1 apresenta a expressão que representa o valor de erro, dada por:

$$e(n) = d(n) - y(n) \quad (1.1)$$

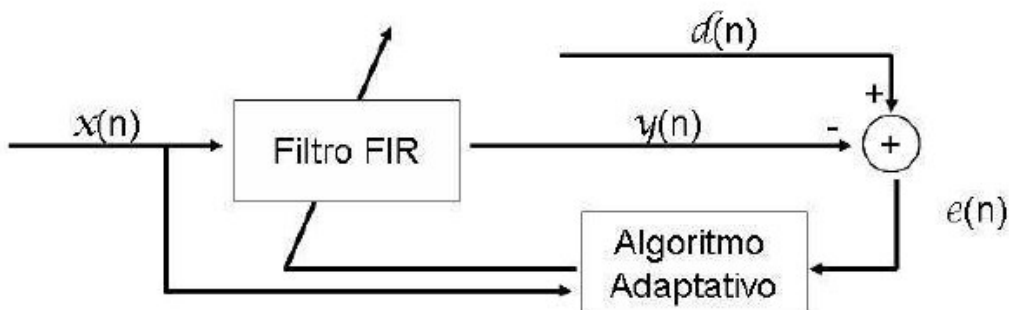


Figura 1.2: Sistema Adaptativo

A partir desta percepção verifica-se que o objetivo final do filtro é estimar seu valor de saída, aproximando ao máximo este resultado com o valor desejado, até atingir o valor de erro nulo ou próximo de zero. A realimentação do filtro com o sinal de erro procura forçar o sistema a esta adaptação.

1.3. Filtro FIR

Filtros FIR são estruturas responsáveis pelas características do processo de filtragem, e respondem pela duração da resposta impulsional deste filtro. Sua construção obedece o critério desejado, e é o responsável pelo comportamento do sistema adaptativo.

Na figura 1.3 é apresentada a ilustração da estrutura de um filtro transversal, onde o número de elementos de atraso determina a duração finita de sua resposta ao impulso, representados por Z^{-1} .

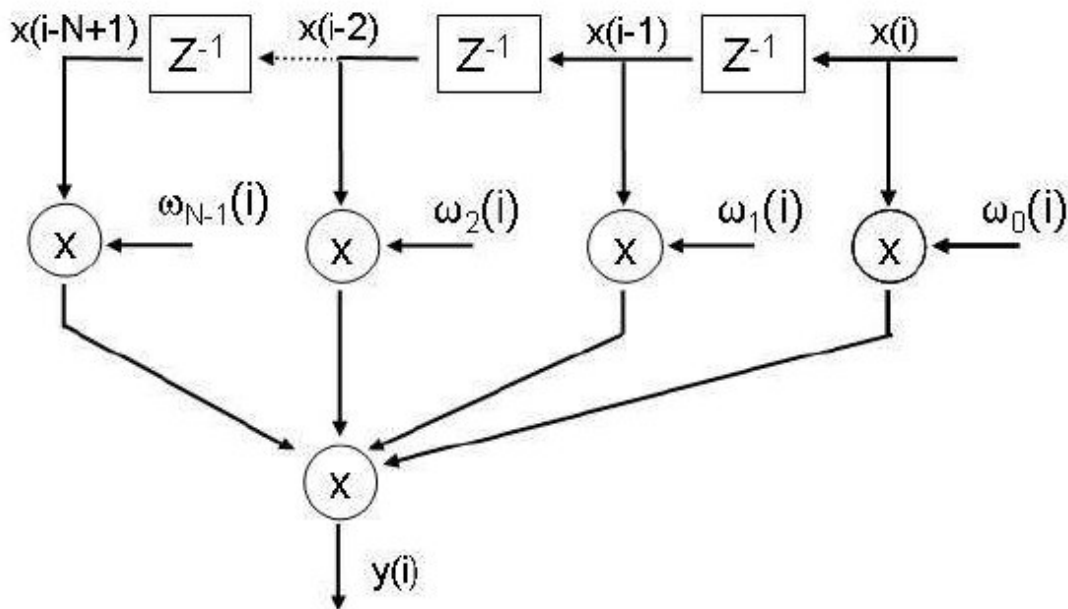


Figura 1.3: Filtro FIR

Os filtros FIR são utilizados no processo de filtragem no domínio de uma determinada frequência. Eles podem diferir na forma de atuação sobre o sinal. São filtros estáveis, e de fácil implementação. Este filtro, chamado filtro de resposta ao impulso de duração finita, é utilizado para filtragem adaptativa em tempo real. Sua ordem é baseada no número de elementos de atraso, e sua saída y é determinada pela equação 1.2:

$$y = \sum_{i=0}^{N-1} \omega_i x(n - i) \quad (1.2)$$

É importante salientar que o desempenho do sistema adaptativo é diretamente dependente do algoritmo adaptativo utilizado no filtro, sendo este tópico abordado mais adiante.

1.4. Solução ótima de Wiener

A solução de Wiener é proveniente de uma base estatística, onde os coeficientes ótimos de um filtro linear são obtidos através da minimização do erro quadrático. Este filtro requer informações adicionais a respeito do sinal original e do ruído; e além disso ele considera a estacionariedade do ambiente. O objetivo da solução de Wiener é fornecer o vetor ω , também chamado ponto ótimo. Porém o ponto ótimo para uma solução em tempo real passa a ser utópico, visto que para a utilização da solução de Wiener é necessário que conheçamos antecipadamente a matriz de correlação dos dados de entrada R e vetor de correlação cruzada do sinal desejado p . Assim, diante do impasse em utilizar a solução de Wiener, se faz necessário utilizar um algoritmo adaptativo que procure aproximar-se do valor desejado de resposta.

A finalidade do algoritmo adaptativo é ajustar o filtro para minimizar o valor médio quadrático do erro $e(n)$. O sistema de adaptação irá continuamente ajustar os coeficientes do filtro, e a cada iteração tentar minimizar o erro. O comportamento do filtro, sua complexidade computacional e tempo de resposta dependem diretamente do algoritmo adaptativo implementado no filtro. O filtro de Wiener utiliza um filtro transversal como base de sua estrutura, que apresenta baixa complexidade computacional e pequeno desajuste.

2. Algoritmos Adaptativos

Entre os diversos algoritmos existentes na literatura, pode-se citar o LMS, o *Normalized-LMS* (NLMS), o *Recursive Least Square* (RLS) e o *Affine Projection* (AP), entre outros. Cada um destes algoritmos apresenta características bem peculiares, o que faz com que a escolha de um deles seja baseada com o tipo de problema a que ele será aplicado.

Em particular, uma das aplicações onde a filtragem adaptativa tem permitido excelentes resultados é no cancelamento de eco acústico. O eco acústico é um fenômeno causado pela reflexão do som. Ele ocorre quando uma fonte emissora emite e recebe de volta, além do retorno direto, uma outra versão do som, refletido (e distorcido) por uma superfície, após um determinado intervalo de tempo, resultando em repetição do som original.

Assim, quando ocorre o evento, surge uma dificuldade em estabelecer comunicação, em virtude deste retorno. Como exemplo clássico, temos as redes de telecomunicações, onde o eco vem a ser a principal fonte de deterioração.

No processo de cancelamento de eco acústico, diversas estruturas de hardware são estudadas. Tais estruturas são baseadas em processadores digitais de sinais (DSPs). Entretanto, conforme já mencionado, o melhor desempenho de um sistema adaptativo está diretamente relacionado ao algoritmo adaptativo implementado.

Assim, além da implementação de variados algoritmos adaptativos, também é necessário ter o conhecimento das técnicas de redução de potência e de aumento de desempenho, a fim de alcançar métricas necessárias para o funcionamento. O desempenho de um algoritmo adaptativo pode ser avaliado por um conjunto de características. Segundo as principais são:

- 1. Taxa de convergência:** define o número de iterações necessárias para o algoritmo levar os coeficientes do filtro para um valor em torno da solução ótima de Wiener, no sentido médio quadrático;
- 2. Desajuste:** é a medida quantitativa da diferença entre o erro médio quadrático residual devido ao algoritmo e o erro médio quadrático mínimo, que é produzido pelo filtro de Wiener;
- 3. Tracking:** capacidade do algoritmo de acompanhar as variações das características estatísticas dos sinais em um meio não-estacionário. Seu desempenho é influenciado pela taxa de convergência e pelas flutuações em regime permanente;
- 4. Robustez:** capacidade do algoritmo operar satisfatoriamente com sinal de entrada mal condicionado;
- 5. Complexidade Computacional:** número de operações por iteração do algoritmo. Este fator pode determinar a viabilidade de sua implementação em tempo real;

6. Estrutura: o filtro adaptativo pode ser implementado utilizando-se diversas estruturas, tais como treliças, forma transversal e outras;

7. Estabilidade numérica: filtros adaptativos freqüentemente precisam ser implementados em processadores digitais de sinais (DSP's), os quais operam com aritmética de precisão finita. Um algoritmo é considerado numericamente estável se o vetor de erro nos coeficientes do filtro permanece limitado ao longo do processamento com precisão finita.

3. Algoritmo Recursivo de Mínimos Quadráticos (RLS - *Recursive Least Square*)

O algoritmo **RLS (recursive least squares)** é um algoritmo de estimação paramétrica utilizado em filtragem adaptativa, para encontrar os coeficientes do filtro que relacionam à produção recursiva de mínimos quadrados do sinal de erro (diferença entre o sinal desejado e o verdadeiro).

Na técnica de estimação seqüencial (SLS) o estimador médio é atualizado seqüencialmente com base no valor anterior. Na técnica RLS, o estimador médio é atualizado com base em um conjunto de valores previamente simulados em vez de ser atualizado com um único valor como na técnica SLS.

As vantagens do algoritmo RLS são que há rápida convergência, há um excelente desempenho, é necessária pouca informação sobre os dados e o algoritmo é simples. Esta técnica usa o conceito de ponto de estimação em vez de um intervalo de base para a estimação e assim o estimador converge para o valor real mais rapidamente.

Considerando o modelo de séries temporais linear:

$$y(n + 1) = wx(n) + e(n) \quad (3.1)$$

onde $e(n)$ é o ruído branco. Desejamos estimar o parâmetro w através de mínimos quadrados, e a cada tempo N nos referimos à nova estimação de mínimos quadrados dada por \hat{w}_N . À medida em que o tempo evolui, é desejável evitar que todo esse algoritmo de mínimos quadrados seja re-executado para encontrar a nova estimativa para \hat{w}_{N+1} , em termos de \hat{w}_N , e atualiza os valores utilizando diversas técnicas.

O maior benefício de usar o algoritmo RLS é que não há a necessidade de inverter matrizes extremamente grandes, e, conseqüentemente, economiza tempo de processamento.

Na técnica de estimação RLS há um desvio nos dados $x[k]$ fornecidos, em relação aos dados desejados $d[k]$ devido a características dos dados de entrada onde o dado desejado $d[k]$ estão próximos do resultado desejado. Os dados que são medidos através da experimentação (valores observados) têm erro inerente quando comparados aos valores de reais (desejados). Os dados observados são conhecidos como dados ruidosos, uma combinação ponderada dos dados observados é usada para estimar os valores reais. Pesos são atribuídos aos valores dos dados observados e o algoritmo RLS computa os pesos

ótimos com base na estimação prévia dos valores dos erros. A função objetivo para o sistema adaptativo RLS é obtido deterministicamente como:

$$\xi^d(k) = \sum_{n=0}^k \lambda^{k-n} e^2(n) \quad (3.2)$$

$$\xi^d(k) = \sum_{n=0}^k \lambda^{k-n} [d(n) - \bar{x}(n)\bar{w}(n)]^2 \quad (3.3)$$

onde $e(i)$ é o erro de saída no instante i e $\bar{w}(i) = [w_0(i) w_1(i) \dots w_N(i)]$ é o vetor de pesos no instante i . O parâmetro λ é uma função de ponderação exponencial, a qual precisa ser escolhida na faixa $0 << \lambda \leq 1$ (normalmente está na faixa de 0,95 a 0,99).

Diferenciando a equação (3.3) e resolvendo para $w(k)$ temos,

$$w(k) = R_D^{-1}(k) p_D(k) \quad (3.4)$$

onde $R_D(k)$ e $p_D(k)$ são a matriz de correlação dos dados observados e o vetor de correlação cruzada entre os dados de entrada e os dados desejados, respectivamente. Para computar a matriz de correlação e sua inversa, aplica-se o lema da inversão de matrizes, para simplificar o cálculo:

$$S_D(k) = R_D^{-1}(k) = \frac{1}{\lambda} \left[S_D(k-1) - \frac{S_D(k-1)x(k)x^T(k)S_D(k-1)}{\lambda + x^T(k)S_D(k-1)x(k)} \right] \quad (3.5)$$

Para (3.5) assume-se que $R_D(k)$ é não singular. Porém, se $R_D(k)$ é singular, uma inversa generalizada pode ser usada para obter a solução para $\bar{w}(k)$. A matriz de correlação cruzada pode ser calculada como:

$$p_D(k) = \lambda p_D(k-1) + d(k)x(k) \quad (3.6)$$

O estimador pode então ser calculado pela equação:

$$y(k) = \bar{w}^T(k) \bar{x}(k) \quad (3.7)$$

Em uma técnica de estimação linear, o estimador é a combinação linear dos dados observados, isto é:

$$\xi^d(k) = \sum_{n=0}^k \lambda w(n) x(n) \quad (3.8)$$

onde λ é o fator “de esquecimento”, $w(k)$ é o vetor de coeficientes / pesos.

4.1 Estimação de potência

A simplicidade na implementação do algoritmo RLS é explorada na estimação da potência média dissipada em circuitos lógicos. A potência dissipada em um circuito lógico depende dos padrões de entrada. O sistema SIS (que é um sistema para síntese de circuitos sequenciais) é usado para realizar simulações com o objetivo de obter a potência dissipada para um conjunto particular de padrões de entrada. SIS usa a seguinte equação para computar a potência dissipada por uma porta:

$$P_{med} = \frac{1}{2} C_{carga} \frac{V_{dd}^2}{T_{ciclo}} E(transições) \quad (4.1)$$

onde P_{med} é a potência média, C_{carga} é a capacitância de carga da porta, V_{dd} é a tensão da fonte de alimentação, T_{ciclo} é o período do relógio global e $E(transições)$ é o valor esperado do número de transições de portas de saída por ciclo de relógio.

As simulações são realizadas no circuito usando SIS, com os padrões de entrada gerados randomicamente por meio de um gerador de padrões automático, e os valores correspondentes das potências são coletados.

O estimador, a matriz de correlação e o vetor de correlação cruzada são calculados em cada iteração pelas equações (3.7), (3.6) e (3.5), respectivamente, com os dados obtidos da simulação. A diferença absoluta entre os valores consecutivos do estimador é computado após cada iteração. Se este valor é menor que o ε (>0), a estimação é interrompida. O valor resultante da potência é computado como o valor médio dissipado pelo circuito.

A ordem do algoritmo é definida como o número de pontos de dados que são usados em cada iteração, para obter-se o objetivo. O fator de esquecimento λ junto com a ordem do algoritmo contribui para acelerar o algoritmo e ajustar a exatidão. A função de erro acumulativo (3.1) pode ser escrita como:

$$\xi^d(k) = \lambda \xi^d(k-1) + e^2(k) (1 - x^T(k) S_D(k) x(k)) \quad (4.2)$$

Seja a escolha de λ em (4.2) ser tal que $\xi^d(k) = \xi^d(k-1)$. Com esta escolha do fator de esquecimento compensará a cada passo k o novo erro de informação oriundo da última medida. Assim, de (4.2) fazendo-se $\lambda = \lambda(k)$

$$\lambda(k) = 1 - \frac{e^2(k)(1 - x^T(k)S_D(k)x(k))}{\xi^d(k-1)} \quad (4.3)$$

A relação entre o comprimento da memória e o fator de esquecimento é mostrado a seguir.

Considerando $e^2(i) = 1$ em (3.1) temos

$$\xi^d(k) = \sum_{n=0}^k \lambda^{k-1} = \frac{(1 - \lambda^{k+1})}{(1 - \lambda)} \quad (4.4)$$

O comprimento de memória N é selecionado de forma que a razão $\xi^d(N)/\xi^d(\infty) \approx 0,9$. Agora, se $1 - \lambda^N = 0,9$ temos

$$N = \frac{\log 0,1}{\log \lambda} \quad (4.5)$$

Quando λ é muito próximo à unidade, $\log \lambda \approx \lambda - 1$. Assim, a memória efetiva do algoritmo de (4.5) e $\lambda = \lambda(k)$ pode ser definido como

$$N = \frac{1}{(1 - \lambda(k))} \quad (4.6)$$

O λ_{\min} tal que não ocorra perda de exatidão é dado em termos da memória como

$$\lambda_{\min} = 1 - \frac{1}{N_a} \quad (4.7)$$

onde $N_a = p + q$, sendo p o comprimento do vetor de entrada e q o comprimento do vetor de erro. No problema de estimação da potencia média o comprimento do vetor de entrada é igual ao comprimento do vetor de erros de forma que a equação (4.7):

$$\lambda_{\min} = 1 - \frac{1}{2N} \quad (4.8)$$

onde N é a ordem do algoritmo. Neste trabalho, a ordem do algoritmo foi escolhida como 16 e assim, usando a mínimo λ recomendado ($\lambda = 0,969$) da equação (4.8).

4.2. Resultados

Os resultados obtidos são apresentados nas Tabelas 4.1 a 4.6, onde:

MC – Resultados para o método Monte Carlo;

SLS – Resultados para o método SLS;

RLS – Resultados para o método RLS.

Table 1. Results for MCNC'91 circuits for an accuracy of $\epsilon = 0.01$

Circuit Name	# of Iterations			Power (μw)			Cpu Time (ms)		
	MC	SLS	RLS	MC	SLS	RLS	MC	SLS	RLS
C432	192	50	16	888	891	893	60	10	5
C499	89	82	46	1217	1216	1223	20	20	14
C880	87	89	26	1369	1367	1366	20	20	8
C1355	41	337	29	3022	3010	3011	10	80	8
C1908	126	205	22	5198	5193	5193	40	50	7
C2670	59	131	48	5650	5647	5650	20	30	15
C3540	102	101	21	8611	8589	8594	30	20	7
C5315	144	643	18	15793	15659	15676	50	160	6
C6288	386	671	50	31941	32049	32057	120	170	10
C7552	311	754	37	23156	23358	23352	100	190	10

Tabela 4.1 – Resultados para circuitos MCNC'91, com precisão de 0,01.

Table 2. Results for MCNC'91 circuits for an accuracy of $\epsilon = 0.001$

Circuit Name	# of Iterations		Power (μw)		Cpu Time (ms)	
	SLS	RLS	SLS	RLS	SLS	RLS
C432	611	87	886	899	150	27
C499	270	218	1215	1223	70	68
C880	235	209	1362	1361	60	118
C1355	682	124	3013	3010	170	39
C1908	798	216	5194	5191	200	68
C2670	323	235	5640	5658	80	74
C3540	260	129	8600	8601	60	40
C5315	1777	118	15670	15676	450	30
C6288	1422	227	32047	32050	360	58
C7552	1034	87	23365	23353	260	30

Tabela 4.2 – Resultados para circuitos MCNC'91, com precisão de 0,001.

Table 3. Results for MCNC'91 circuits for an accuracy of $\epsilon = 0.0001$

Circuit Name	# of Iterations		Power (μw)		Cpu Time (ms)	
	SLS	RLS	SLS	RLS	SLS	RLS
C432	2416	312	883	883	610	100
C499	2103	430	1212	1223	530	135
C880	586	386	1369	1361	150	121
C1355	2310	287	3017	3011	590	90
C1908	798	998	5194	5194	200	313
C2670	1774	235	5621	5658	450	73
C3540	1565	243	8577	8601	400	76
C5315	2149	247	15681	15676	540	77
C6288	2891	402	32078	32050	730	125
C7552	3013	374	23360	23358	760	117

Tabela 4.3 – Resultados para circuitos MCNC'91, com precisão de 0,0001.

Table 4. Results for ISCAS'89 circuits for an accuracy of $\epsilon = 0.01$

Circuit Name	# of Iterations			Power (μw)			Cpu Time (ms)		
	MC	SLS	RLS	MC	SLS	RLS	MC	SLS	RLS
S27	289	57	24	124	125	124	90	10	8
S208	143	63	126	469	472	462	40	20	40
S298	97	108	82	826	826	829	30	30	26
S344	118	128	73	1032	1023	1031	40	30	23
S349	111	173	54	1040	1034	1041	40	40	17
S382	188	153	149	1124	1127	1125	60	40	47
S386	168	55	35	1142	1143	1145	50	10	11
S1494	274	97	24	3954	4001	3973	90	30	8
S5378	366	259	112	11901	11940	11894	110	70	36
S13207	1027	687	848	37508	37508	37512	320	190	269
S15850	6103	4337	2076	40026	39992	40035	1900	1200	659
S35932	4649	4621	1157	123041	123049	123046	1450	1280	367
S38584	4905	5053	1998	112605	112903	112623	1530	1400	630

Tabela 4.4 – Resultados para circuitos ISCAS'89, com precisão de 0,01.

Table 5. Results for ISCAS'89 circuits for an accuracy of $\epsilon = 0.001$

Circuit Name	# of Iterations		Power (μw)		Cpu Time (ms)	
	SLS	RLS	SLS	RLS	SLS	RLS
S27	180	113	124	124	50	35
S208	229	176	471	462	60	55
S298	230	204	826	828	60	64
S344	327	200	1030	1031	80	63
S349	173	129	1034	1040	40	40
S382	153	245	1127	1125	40	77
S386	424	220	1148	1145	100	69
S1494	247	103	4022	3981	70	33
S5378	303	191	11943	11927	80	61
S13207	997	1096	37579	37520	280	348
S15850	5111	3398	39999	40019	1420	1079
S35932	4621	2662	123049	123092	1280	845
S38584	6823	2001	112876	112675	1890	635

Tabela 4.5 – Resultados para circuitos ISCAS'89, com precisão de 0,001.

Table 6. Results for ISCAS'89 circuits for an accuracy of $\epsilon = 0.0001$

Circuit Name	# of Iterations		Power (μw)		Cpu Time (ms)	
	SLS	RLS	SLS	RLS	SLS	RLS
S27	917	420	125	123	230	132
S208	315	400	470	462	80	125
S298	720	386	828	828	180	121
S344	759	407	1032	1032	190	128
S349	333	392	1033	1040	80	123
S382	603	436	1128	1125	130	137
S386	1139	248	1147	1145	270	78
S1494	841	459	4023	3986	230	146
S5378	1164	437	11947	11926	320	139
S13207	2050	2444	37578	37520	570	776
S15850	10457	3398	39990	40019	2900	1079
S35932	12932	4471	123045	123071	3590	1420
S38584	17253	2029	112876	112675	4790	644

Tabela 4.4 – Resultados para circuitos ISCAS'89, com precisão de 0,0001.

4.5. Conclusões

Os resultados experimentais indicam que para altas exatidões o método de estimação de mínimos quadrados converge rapidamente. Além disso, o nível de exatidão pode ser definido pelo usuário. Para alguns níveis de exatidão a abordagem de mínimos quadrados recursiva converge até 12 vezes mais rápido.

5. Referências Bibliográficas

- [1] MURUGAVEL, A. K.; RANGANATHAN, N.; CHANDRAMOULI, R.; CHAVALI, S. **Least-Square Estimation of Average Power in Digital CMOS Circuits**. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v. 10, n.1, pp. 55-58, 2002.
- [2] WIKIPEDIA – The Free Encyclopedia – **Recursive Least Squares Filter** - http://en.wikipedia.org/wiki/Recursive_Least_Squares
- [3] RUCH, Conrado Jr. **Implementação de Filtros Adaptativos em FPGA**. Universidade Católica de Pelotas, Dezembro de 2006.
- [4] BRANDÃO, João Célio. **Utilização de Antenas Adaptativas em Sistemas CDMA**. Pontifícia Universidade Católica do Rio de Janeiro. Setembro de 2001.
- [5] RACHID, Euler Mendes. **Cancelamento de Eco em Telefonia IP**. Universidade Estadual de Campinas – UNICAMP. Outubro de 2004.