UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI SCIENZE STATISTICHE

CORSO DI LAUREA IN STATISTICA E TECNOLOGIE INFORMATICHE

MIGLIORAMENTI DELLA CARTA DI CONTROLLO AEWMA

Relatore: Chiar.mo Prof. Guido Masarotto

Candidato: Marco Scutari

ANNO ACCADEMICO 2003–2004

Ai miei genitori
ed a tutti coloro che
mi hanno sostenuto

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Le carte per controllo della qualità, pur appartenendo ad una classe di metodi statistici piuttosto semplici dal punto di vista teorico, vengono ampiamente applicate per monitorare i procssi produttivi e garantire la qualità del prodotto finale.

Le carte di tipo EWMA (*Exponentially Weighted Moving Average*) in particolare hanno avuto una diffusione piuttosto ampia grazie alla loro semplicità di interpretazione, alla velocità di implementazione ed alla robustezza dimostrati.

Successivamente la loro estensione nella classe AEWMA (*Adaptive Exponentially Weighted Moving Average*), che include l'EWMA come caso particolare, ha in parte risolto il problema di inerzia che le caratterizza ridefinendo il modo in cui le osservazioni vengono pesate nel calcolo di questa statistica. Tuttavia, nonostante questi miglioramenti, permangono ancora dei difetti di funzionamento nella fase di avviamento della carta di controllo.

L'oggetto di studio di questa tesi è proprio l'applicazione di alcune modifiche note in letteratura alle carte di tipo EWMA ed AEWMA, con il preciso scopo di migliorare la loro responsività nella fase di avviamento senza compromettere le loro prestazioni complessive. In particolare ho preso in analisi la tecnica nota come FIR (*Fast Initial Response*), che sfutta l'uso contemporaneo di due carte di controllo, ed una possibile ridefinizione delle soglie d'allarme sulla base della vera varianza delle statistiche alla base di EWMA ed AEWMA.

Capitolo 1

Carte per il controllo della qualità

Le carte di controllo sono una classe di tecniche statistiche pensate per l'analisi obiettiva delle caratteristiche di un processo stocastico e per l'individuazione di variazioni sistematiche nel suo andamento (Wetherill e Brown [8], 1991).

In generale si tratta di metodi statistici semplici, sia nell'applicazione che nell'interpretazione, utilizzati in ambito industriale per tenere sotto controllo il processo produttivo ed assicurare la qualità del prodotto finale (Lucas e Saccucci [5], 1990). Per questo motivo i due parametri che vengono maggiormente monitorati sono il livello medio del processo e la sua variabilità, che sono spesso soggette a vincoli severi derivanti da esigenze pratiche o norme di legge.

La qualità delle carte di controllo viene valutata in base alla rapidità con cui vengono segnalate eventuali anomalie nel processo in esame ed al numero di falsi allarmi. La statistica con cui questi parametri vengono stimati è la *run length*, ovvero il numero di osservazioni che intercorrono tra la segnalazione di una anomalia e l'anomalia stessa, ed il suo valore atteso, l'*average run length*.

In una situazione ideale fintanto che il processo rimane in controllo, l'*average run length* della carta dovrebbe essere estremamente grande, mentre al minimo cambiamento la segnalazione dovrebbe avvenire istantaneamente. In pratica eventuali ano-

malie, soprattutto se di modesta entità, vengono segnalate con un certo ritardo o non vengono rilevate affatto, mentre valori assolutamente accettabili causano a volte falsi allarmi. Una taratura ottimale può solo limitare i falsi allarmi diminuendo la sensibilità della carta oppure cercare di ottenere una segnalazione molto rapida per certi tipi di variazioni rischiando falsi allarmi; una ottimizzazione globale semplicemente non è possibile.

L'AEWMA (*Adaptive Exponentially Weighted Moving Average*) è una estensione di carte di controllo esistenti che unisce la stabilità tipica dell'EWMA (*Exponentially Weighted Moving Average*) alla rapidità di reazione tipica delle carte di tipo Shewhart. In questo modo si propone di ottenere un comportamento globalmente buono, o quanto meno ottimo per alcuni tipi di anomalie, pur rimanendo di facile interpretazione.

1.1 Carte di controllo Shewhart

Le carte *Shewhart* (così chiamate dal nome del loro inventore) costituiscono il punto di partenza di tutta la teoria delle carte per il controllo della qualità. Il processo viene monitorato utilizzando direttamente le osservazioni (singolarmente oppure raggruppate in *batch*) e le soglie di allarme (*action limits*) vengono stabilite in base alla varianza stimata di queste ultime.

Sono presenti inoltre delle soglie di allerta (*warning limits*), calcolate in modo simile ma più vicine al livello medio del processo.

Ad esempio, in una delle sue tante versioni, la carta segnala un errore quando:

- una osservazione (o un *batch*) superano le soglie di allarme.
- due osservazioni successive si trovano oltre la medesima soglia di allerta.

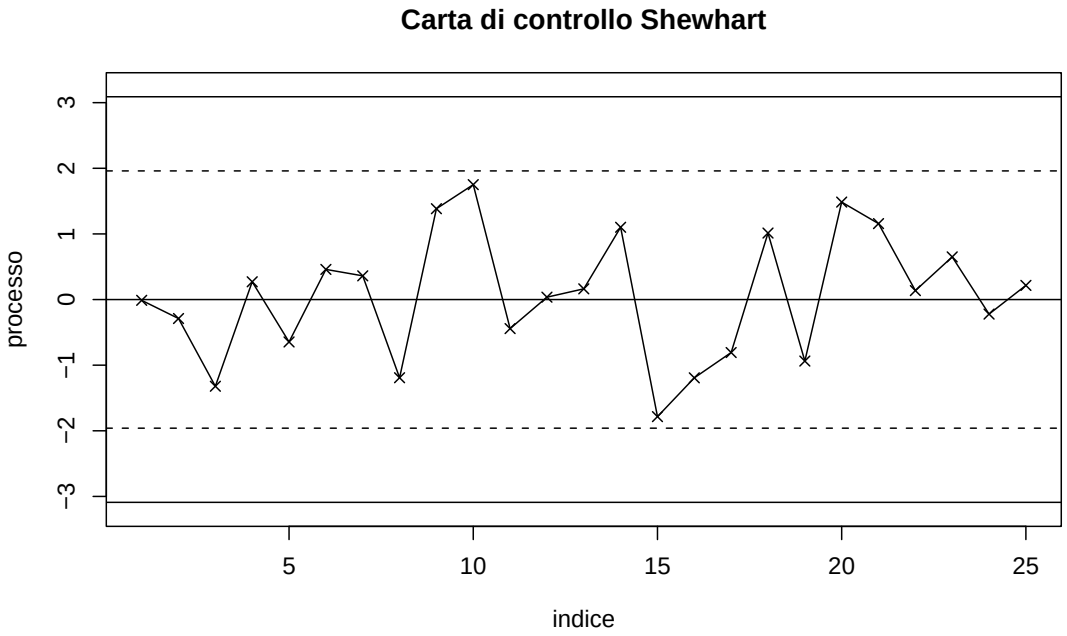

Figura 1.1: Carta di controllo Shewhart.

1.2 Exponentially Weighted Moving Average (EWMA)

Questo tipo di carta di controllo si basa sulla statistica

$$x_t = (1-\lambda)x_{t-1} + \lambda y_t \qquad 0 < \lambda \leq 1$$

dove y_t rappresenta il processo sotto osservazione, t è l'indice temporale e λ è una costante positiva che determina il peso della nuova osservazione y_t nel calcolo di x_t. Il valore iniziale della carta, x_0 , solitamente coincide con η_0, il livello medio fissato per il processo. Quest'ultimo si considera fuori controllo quando $| x_t - \eta_0 |$ supera una certa soglia h, che viene scelta in modo da limitare i falsi allarmi.

Il suo nome deriva dalla possibilità di riscrivere la formula precedente come una sommatoria in cui il peso delle osservazioni passate decade in modo geometrico :

$$(1-\lambda)^t x_0 + \lambda \sum_{i=0}^{t-1} (1-\lambda)^i y_{t-i}$$

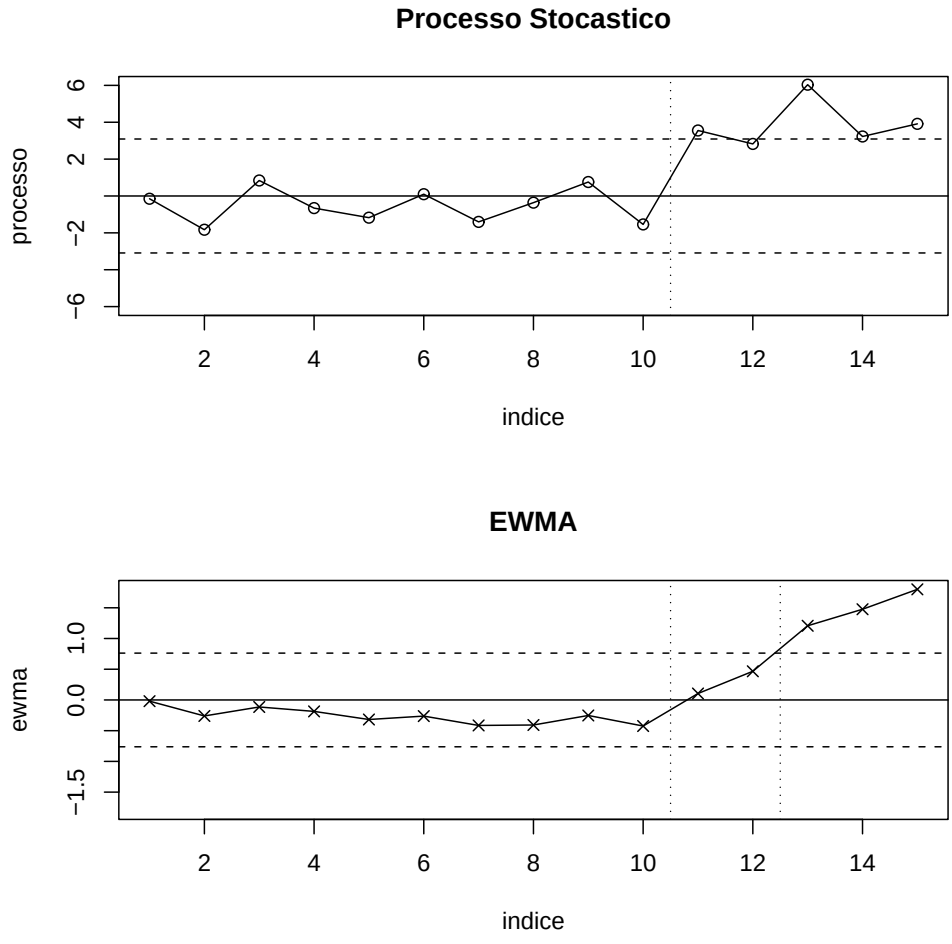

Figura 1.2: Carta di controllo EWMA.

Il valore della costante λ determina molte delle proprietà dell'EWMA, tra cui l'inerzia iniziale. Infatti, se le osservazioni sono indipendenti, per costruzione questa carta ha varianza

$$\sigma^2(x_t) = [(1-(1-\lambda)^{2i})\lambda/(2-\lambda)]\sigma_y^2$$

che converge asintoticamente a

$$\sigma^2(x) = [\lambda/(2-\lambda)]\sigma_y^2$$

dove σ_y^2 è la varianza del processo stocastico. Per valori piccoli di λ questa convergenza è più lenta, ed è quindi piú difficile che valori anomali riscontrati nella fase di avviamento vengano segnalati (Wetherill e Brown [8], 1991).

Un certo grado di inerzia tuttavia rimane anche quando la carta è entrata a regime. Per questo motivo la valutazione di una carta EWMA deve tenere conto dell'*in-control ARL* e del *worst case ARL*, con cui si valutano rispettivamente la *run length* di una carta già a regime e la velocità di reazione ad uno scostamento del livello del processo quando la carta si trova nelle vicinanze della soglia opposta.

Inoltre, dato che valori piccoli di λ sono adatti a rilevare piccoli cambiamenti nel processo e valori più elevati sono necessari per segnalare rapidamente scostamenti di grande entità, non è possibile ottenere un singolo EWMA ottimale in entrambi i casi (Lucas e Saccucci [5], 1990).

1.3 Adaptive Exponentially Weighted Moving Average (AEWMA)

Questa carta di controllo è una estensione della carta esaminata nella sezione precedente. La statistica su cui si basa è

$$x_t = x_{t-1} + \phi(e_t) \qquad x_0 = \eta_0$$

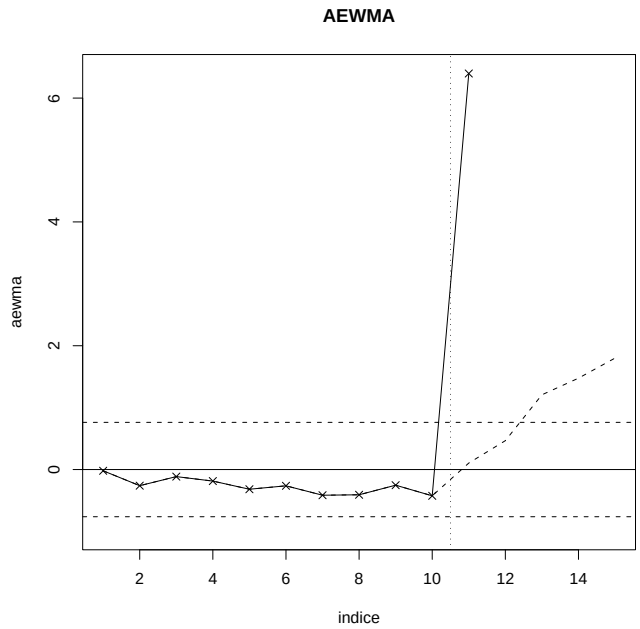

Figura 1.3: Andamento dell'AEWMA rispetto ad una carta di controllo EWMA.

dove $e_t = y_t - x_{t-1}$ e $\phi(e)$ è una qualsiasi funzione con le seguenti caratteristiche:

- sia monotona crescente rispetto ad e.
- sia una funzione dispari ($\phi(e) = -\phi(-e)$ per $\forall e \in \mathbb{R}$).

- $\phi(e) \approx \lambda e$ quando $\mid e \mid$ è piccolo.
- $(\phi(e)/e) \approx 1$ quando $\mid e \mid$ è grande.

In questo modo l'AEWMA si comporta come una carta EWMA quando $\mid e \mid$ è piccolo, e come una carta Shewhart quando $\mid e \mid$ è grande, limitando il problema dell'inerzia nella segnalazione di comportamenti anomali(Capizzi e Masarotto [3], 2003). Infatti riscrivendola come

$$\begin{cases} x_t = (1 - w(e_t))x_{t-1} + w(e_t)y_t & \\ w(e) = \phi(e)/e & \text{se } e \neq 0 \\ w(e) = 0 & \text{se } e = 0 \end{cases}$$

si ottiene una carta di controllo Shewhart se $w(e) \approx 1$ (ovvero quando $\mid e \mid \to \infty$) ed un'EWMA se $w(e)$ è costante e compreso nell'intervallo [0,1).

Per quanto riguarda la funzione dei pesi $\phi(e)$ alcuni esempi presenti in (Capizzi e Masarotto [3], 2003) sono:

$$\phi_{hu}(e) = \begin{cases} e + (1-\lambda)k & \text{per } e < -k \\ \lambda e & \text{per } \mid e \mid \leq k \\ e + (1-\lambda)k & \text{per } e > k \end{cases}$$

$$\phi_{bs}(e) = \begin{cases} e\Big(1 - (1-\lambda)\big(1 - (e/k)^2\big)^2\Big) & \text{per } \mid e \mid \leq k \\ e & \text{altrimenti} \end{cases}$$

$$\phi_{cb}(e) = \begin{cases} e & \text{per } e \leq p_1 \\ -\tilde{\phi}_{cb}(-e) & \text{per } -p_1 < e < -p_0 \\ \lambda e & \text{per } \mid e \mid \leq p_0 \\ \tilde{\phi}_{cb}(-e) & \text{per } p_0 < e < p_1 \\ e & \text{per } e \geq p_1 \end{cases}$$

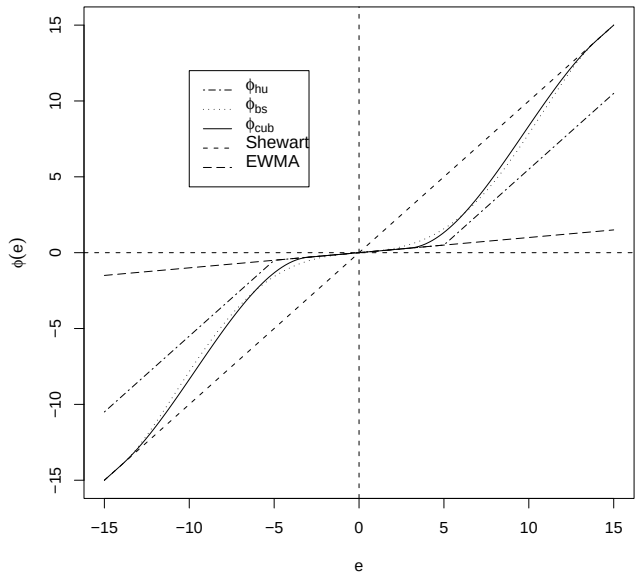

Figura 1.4: Funzioni dei pesi $\phi(e)$.

dove $0 < \lambda \leq 1$, $0 \leq p_o \leq p_1$ sono delle costanti opportune e

$$\tilde{\phi}_{cb}(e) = \lambda e + (1-\lambda)\left(\frac{e-p_0}{p_1-p_0}\right)^2 \left(2p_1 + p_0 - (p_0+p_1)\left(\frac{e-p_0}{p_1-p_0}\right)\right)$$

è un polinomio di terzo grado tale che ϕ_{cb} e la sua derivata prima sono continue.

1.4 Fast Initial Response

Accade spesso che le carte di controllo diano problemi in corrispondenza delle prime osservazioni a causa delle loro proprietà intrinseche. L'eventuale inefficacia dei controlli eseguiti dopo l'ultima segnalazione di errore rischia quindi di non essere rilevata tempestivamente (Lucas e Crosier [4], 1982). Il FIR (*Fast Initial Response*) è

una variazione che si può applicare alle carte di controllo di tipo EWMA o CUSUM per migliorare la loro reattività nella fase di avviamento, e risulta quindi molto utile in situazioni in cui si hanno problemi di inerzia.

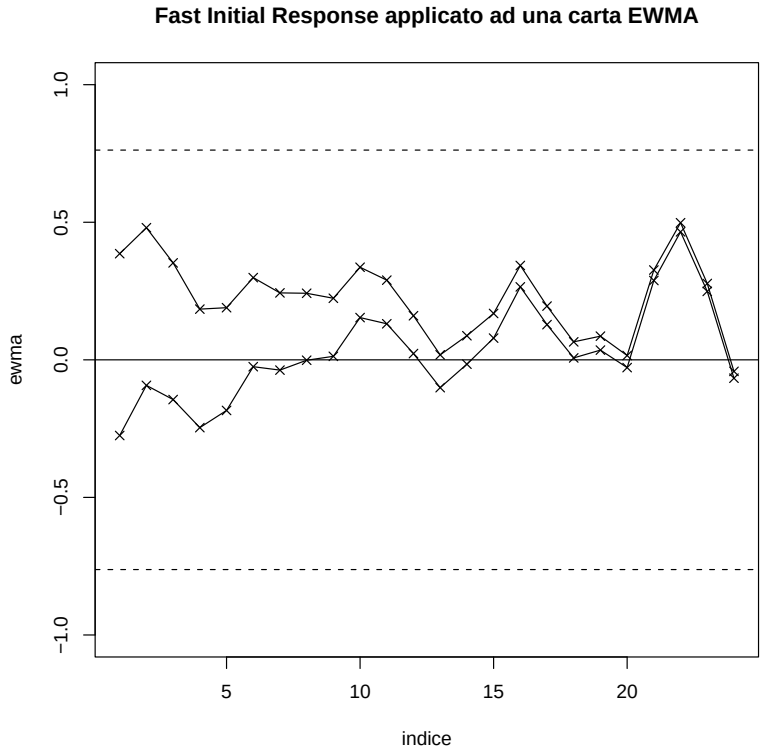

Figura 1.5: Fast Initial Response applicato ad una carta EWMA.

In pratica si tratta di implementare contemporaneamente due carte di controllo fissando come livelli di partenza dei punti simmetrici rispetto al livello medio del processo:

$$\eta_0 \pm \alpha h \qquad\qquad 0 < \alpha < 1$$

Se il processo parte fuori controllo, una delle due carte si trova in buona posizione

per segnalare un allarme; in caso contrario queste tendono a convergere verso il vero livello del processo quando questo entra a regime, rendendo inutile il mantenimento del FIR (Lucas e Saccucci [5], 1990).

Purtroppo però questa tecnica ha numerosi svantaggi: oltre ad essere computazionalmente più impegantiva, rende relativamente più difficile l'interpretazione e non sembra facilmente generalizzabile al caso multivariato. Rimodellare le soglie di allarme tenendo conto della vera varianza della carta può essere un metodo più semplice ed efficiente per ottenere lo stesso effetto senza rinunciare alla semplicità tipica dell'EWMA (MacGregor e Harris [6], 1990).

1.5 Frontiere mobili

Sia le carte EWMA che quelle AEWMA utilizzano soglie di allarme basate sulla varianza asintotica

$$\sigma^2(x) = \frac{\lambda}{2-\lambda}\sigma_y^2$$

mentre la varianza reale della statistica è, nel caso dell'EWMA,

$$\sigma^2(x_i) = \left(1-(1-\lambda)^{2i}\right)\frac{\lambda}{2-\lambda}\sigma_y^2$$

Chiaramente per l'AEWMA questa è una stima approssimata, dato che λ non è costante, ma risulta comunque abbastanza accurata in virtù dell'andamento della funzione dei pesi $\phi(e)$.

Per quanto la convergenza sia veloce, questa impostazione causa comunque problemi nella fase di avviamento della carta. La soluzione più semplice, sia dal punto di vista dell'implementazione che da quello teorico, consiste nel sostituire le so-

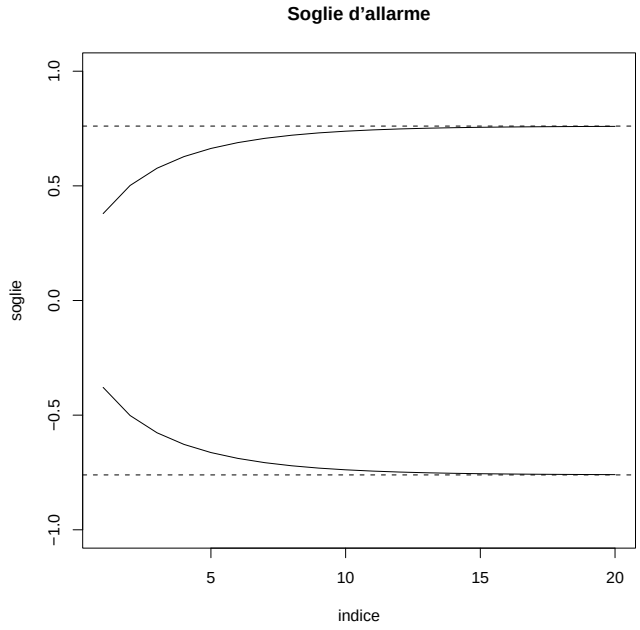

Figura 1.6: Soglie di allarme basate sulla vera varianza dell'EWMA.

glie tradizionali con delle soglie basate sulla vera varianza della carta. Una volta che quest'ultima è entrata a regime, ovvero quando

$$\frac{\sigma^2(x_i)}{\sigma^2(x)} = 1 - (1-\lambda)^{2i} \simeq 1$$

si può tornare ad utilizzare le soglie tradizionali.

Capitolo 2

Metodi di calcolo

L'analisi condotta in questa tesi ha come obiettivo principale il confronto tra EWMA ed AEWMA come carte di controllo per il livello di un processo ed il loro comportamento rispetto alle tecniche di *fast initial response*. Pertanto tutti i metodi discussi in questo capitolo sono visti in questa ottica, e finalizzati all'analisi delle reazioni a processi la cui media differisce da quella ipotizzata nella carta con cui sono monitorati.

2.1 Tecniche per la valutazione di una carta

L'analisi di una carta può essere svolta principalmente in 3 modi:

1. **Simulazione:** consiste semplicemente nello studio del comportamento della carta attraverso l'uso di processi simulati. È il metodo più diretto.

2. **Catena Markoviana:** le proprietà di una carta possono essere valutate anche discretizzando lo spazio in cui essa opera, per poi approssimarla con una catena di Markov (Brooks ed Evans [1], 1972). In pratica si suddivide l'intervallo tra le soglie di allarme in un certo numero di sottointervalli di dimesione omogenea. Ognuno di questi rappresenta uno stato della catena markoviana. Il resto dello

spazio reale ricade in un ulteriore stato, detto *absorbing space*, da cui non è più possibile spostarsi. Questo arrangiamento può essere riassunto in una matrice di transizione del tipo

$$P = \begin{pmatrix} \mathbf{R} & (\mathbf{I}-\mathbf{R})\mathbf{1} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

dove R è una sottomatrice che contiene le probabilità di transizione tra i vari stati (Lucas e Saccucci [5], 1990).

3. **Approssimazione numerica:** in alternativa è possibile ricavare una approssimazione numerica della distribuzione della statistica utilizzata dalla carta di controllo, e studiarne le proprietà.

Per questa tesi ho scelto di utilizzare il primo metodo, per la semplicità di implementazione e di interpretazione. Inoltre non risulta troppo pesante dal punto di vista computazionale, e consente di ottenere una grande mole di risultati in poco tempo. Nel mio caso lo studio di una carta di controllo per 11 diversi scarti (tra il vero livello del processo e quello ipotizzato dalla carta), calcolando di volta in volta l'ARL sulla base di 1000000 di *run length* simulate, ha richiesto circa 30 minuti sulla seguente macchina:

Processore:	Dual Athlon MP 1200
RAM:	768 Mb
Sistema Operativo:	Debian GNU/Linux, kernel 2.4.25

2.2 ARL e valutazione di una carta

L'uso dell'*average run length* come statistica di riferimento nella valutazione delle carte per il controllo della qualità è motivato da ragioni pratiche. Tuttavia essendo calcolato come

$$ARL = E(RL) = \lim_{n\to\infty} \sum_{i=1}^{n} \frac{RL_i}{n}$$

è molto sensibile alla presenza di valori anomali. Le *run length* hanno però per loro stessa natura una distribuzione geometrica; quindi nei casi in cui l'ARL è elevato la coda della distribuzione distorce considerevolmente la stima.

Sembrerebbe quindi sensato considerare anche la mediana, che al contrario della media non è sensibile ai valori anomali, pur continuando a condurre l'analisi in base l'*average run length*. Può accadere infatti che una differenza rilevante delle mediane (a parità di ARL) indichi una effettiva variazione di prestazioni, che verrebbe altrimenti nascosta dalla coda della distribuzione: viceversa una differenza apparentemente significativa in media può essere messa in dubbio se le mediane coincidono.

scarto	0.00	0.25	0.50	0.75	1.00	1.50	2.00
media	465.43	116.13	33.34	16.00	10.06	5.71	4.03
mediana	325	83	26	13	9	5	4

Tabella 2.1: Media e mediana delle *run length* simulate da una carta EWMA.

scarto	0.00	0.25	0.50	0.75	1.00	1.50	2.00
media	501.40	131.14	36.35	16.93	10.44	5.78	3.95
mediana	349	93	28	14	9	5	4

Tabella 2.2: Media e mediana delle *run length* simulate da una carta AEWMA.

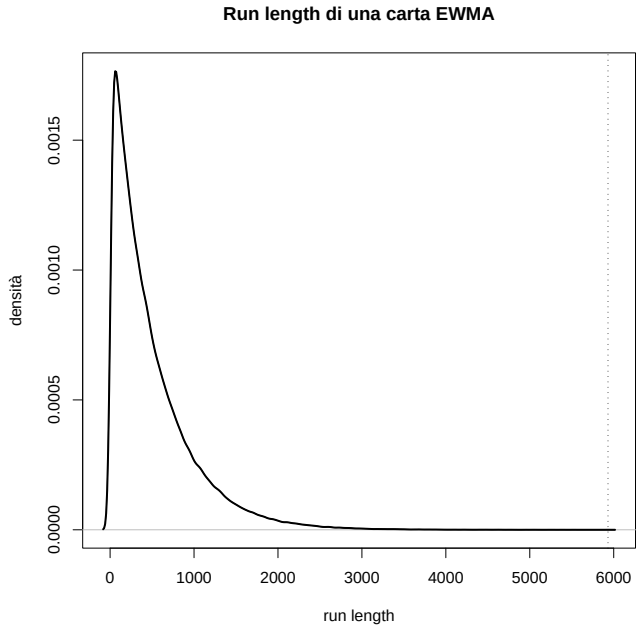

Figura 2.1: Distribuzione della run length in controllo di una carta EWMA.

2.3 Taratura dei parametri di EWMA ed AEWMA

Si osservi innanzitutto che per confrontare tra loro carte di tipo diverso è necessario che l'*average run length* del processo in controllo sia uguale, o quanto meno simile, nei vari casi. In questo modo le differenze di prestazioni che si riscontrano nei vari esperimenti possono essere attribuite interamente alle caratteristiche peculiari della singola carta.

Le tecniche utilizzate in letteratura per individuare i valori ottimali per i parametri dell'EWMA e dell'AEWMA, dato l'ARL del processo in controllo, sono diverse per le due carte.

2.3.1 EWMA

Stabilito l'ARL in controllo della carta, il valore ottimale di λ cresce al crescere dello scarto che si vuole individuare nel processo sotto esame (Lucas e Saccucci [5], 1990), anche se molto dipende dalla struttura della variabilità delle osservazioni.

In generale non esiste una procedura analitica, anche se è comunque necessario

1. stabilire l'ARL in controllo.
2. individuare lo scarto che si desidera individuare con maggior rapidità.
3. calcolare l'*average run length* per gli altri possibili scarti, e verificare che siano segnalati in modo sufficientemente veloce.

shift	$\lambda = 0.75$	$\lambda = 0.50$	$\lambda = 0.25$	$\lambda = 0.10$	$\lambda = 0.05$
0.00	500	500	500	500	500
0.25	321	255	170	106	84.1
0.50	140	88.8	48.2	31.3	28.8
0.75	62.5	35.9	20.1	15.9	16.4
1.00	30.6	17.5	11.1	10.3	11.4
1.50	9.90	6.53	5.46	6.09	7.12
2.00	4.54	3.63	3.61	4.36	5.23
2.50	2.69	2.50	2.74	3.44	4.17
3.00	1.88	1.93	2.26	2.87	3.50
4.00	1.22	1.34	1.73	2.19	2.69
5.00	1.04	1.07	1.32	1.94	2.16

Tabella 2.3: Andamento dell'*average run length* rispetto al parametro λ.

Tutttavia, sapendo che l'andamento dell' *average run length* rispetto a λ presenta un solo minimo per ogni singolo scarto, è possibile risalire al valore ottimale utilizzando un qualsiasi algoritmo di minimizzazione numerica.

2.3.2 AEWMA

Detto θ l'insieme dei parametri utilizzati in una carta AEWMA, il metodo utlizzato in (Capizzi e Masarotto [3], 2003) si articola in tre punti:

1. per prima cosa si sceglie l'*average run length B* del processo in controllo e due diversi scarti μ_1 e μ_2, rispetto ai quali si opera il processo di ottimizzazione. In particolare μ_1 dovrebbe essere lo scarto di piccola entità che accade più frequentemente, e μ_2 il suo equivalente tra gli scarti più grandi.

2. si trova la combinazione di parametri θ^* che ottimizza l'ARL rispetto a μ_2 come soluzione del sistema

$$\begin{cases} min_\theta ARL(\mu_2, \theta) \\ ARL(0, \theta) = B \end{cases}$$

   dove $ARL(\mu, \theta)$ indica l'*average run length* di una carta di parametri θ con scarto μ tra il vero livello del processo e quello ipotizzato.

3. infine, scelta una costante positiva α, si trova il θ ottimale come soluzione numerica approssimata di

$$\begin{cases} min_\theta ARL(\mu_1, \theta) \\ ARL(0, \theta) = B \\ ARL(\mu_2, \theta) \leqslant (1+\alpha) ARL(\mu_2, \theta^*) \end{cases}$$

   Si ottiene quindi la carta con ARL minimo per scarti di tipo μ_1 tra quelle per cui l'ARL rispetto a μ_2 è ottimale o quasi ottimale.

Questo approccio riesce a produrre delle configurazioni ragionevoli per variazioni di piccola e grande entità nel livello del processo. Le sue prestazioni per altre carte non sono altrettanto buone, probabilmente a causa della scarsa flessibilità di queste ultime (Capizzi e Masarotto [3], 2003).

2.3.3 Calibrazione mediante un processo di Robbins-Monro

Per determinare il valore ottimale della soglia esiste un metodo basato su una relazione ricorsiva suggerita da Robbins e Monro (1951) ed analizzata in (Capizzi e Masarotto [2], 1998):

$$h_n = h_{n-1} - \frac{A}{n} y_n$$

dove nel nostro caso h_n è il valore della soglia (o del quantile che andrà moltiplicato per la varianza della statistica alla base della carta), A è una costante positiva, ed y_n è una *run length* riscalata rispetto all'ARL che si vuole ottenere per il processo in controllo

$$y_n = \frac{rl_n - B}{B}$$

La convergenza al vero valore di h è assicurata nella maggior parte dei casi (Ruppert, 1991), e può essere accelerata da una buona scelta rispettivamente del valore di partenza h_0 e della costante A.

Tutti gli altri parametri della carta si considerano dati, e devono quindi essere calcolati con una delle tecniche precedenti.

Per quanto riguarda la regola di arresto, un possibile approccio è stato suggerito in (Stroup e Braun [7], 1982). Sostituendo y_n con $l_n = (y_{n1} + y_{n2})/2$ nella formula della relazione, e quindi utilizzando due *run length* per ogni passaggio, si può definire

$$N = inf\{n > k : u_n = (\sum_{i=n-k+1}^{n} l_i^2/k)(\sum_{i=1}^{n} e_i^2/n) < w\}$$

dove $e_n^2 = (y_{n1} - y_{n2})/2$ e k è un intero appropriato. Sapendo che $2ku_n$ converge in distribuzione a

$$\lim_{n\to\infty} \sum_{i=n-k+1}^{n} \big(l_i - E(l_i)\big)^2$$

inoltre è possibile fissare il limite di arresto: è sufficiente porre $2kw$ uguale ad un percentile della distribuzione χ^2. Dato che non è necessario preoccuparsi degli errori di primo tipo, che causano solo una ulteriore iterazione del processo, quest'ultimo può avere una probabilità molto alta.

Capitolo 3

Risultati e conclusioni

3.1 Condizioni dell'analisi

In letteratura quasi sempre l'analisi delle caratteristiche delle carte EWMA ed AEWMA è basata su processi simulati in cui le osservazioni sono normali indipendenti ed identicamente distribuite. Anche se queste condizioni ideali si presentano di rado in casi reali, si usa operare in questo modo poiché tutte le statistiche necessarie sono facilmente calcolabili e le proprietà teoriche diventano immediatamente verificabili.

Pertanto ho eseguito le varie simulazioni seguendo questo schema, in modo da ottenere risultati confrontabili con quelli contenuti in (Lucas e Saccucci [5], 1990) e (Capizzi e Masarotto [3], 2003). Ulteriori utilizzi delle carte in questione possono essere studiati sulla base dei risultati presenti in questo capitolo, che chiaramente non sono esaustivi e non coprono la vasta gamma di situazioni reali in cui si può operare.

Per quanto riguarda la scelta della soglia di allarme tramite la procedura di Robbins-Monroe, avendo scelto questa impostazione non mi è sembrato necessario implementare la regola di arresto. Computazionalmente risulta più conveniente bloccare il processo dopo un numero predefinito di iterazioni ed utilizzare come stime delle sogle le medie delle osservazioni successive al *burn-in*.

3.2 Risultati

Alla base del differente comportamento delle due carte vi sono le differenti tecniche con cui vengono ottimizzate. Mentre l'AEWMA è impostato come compromesso tra due diverse esigenze (proteggere contemporaneamente da scarti piccoli e grandi), l'EWMA tradizionale è tarato rispetto ad un solo tipo di variazione.

	EWMA		AEWMA	
	$\lambda = .133$		$\lambda = .1354 \quad h = .7928267$	
	$h = .7688665$		$k = 3.2587$	
scarto	ARL	mediana	ARL	mediana
0.00	498.6897	348	500.1558	348
0.25	121.3911	87	130.9033	93
0.50	34.2721	26	36.31515	28
0.75	16.30784	14	16.91417	14
1.00	10.20712	9	10.44298	10
1.50	5.776916	5	5.779801	5
2.00	4.074102	4	3.949896	4
2.50	3.187964	3	2.936611	3
3.00	2.643968	3	2.256669	2
4.00	2.065173	2	1.417542	1
5.00	1.786132	2	1.084595	1

Tabella 3.1: Risultati di (Lucas e Saccucci [5]) e di (Capizzi e Masarotto [3]).

Come si vede chiaramente dalla tabella 3.1, che contiene il profilo dell'*average run length* per un'EWMA ottimizzato per 1σ ed un'AEWMA ottimizzato per 1σ e 5σ, quest'ultima ha un comportamento complessivamente migliore (soprattutto per scarti elevati). Chiaramente in questo caso l'EWMA rimane preferibile solo per scarti di piccola entità, che vengono privilegiati a scapito di quelli superiori a 2σ.

Il parametro *k*, attraverso il quale è possibile stabilire il grado di reattività dell'AEWMA, regola la differenza di prestazioni di queste due carte. Valori molto elevati di questo parametro infatti non permettono alla funzione dei pesi $\phi(e)$ di spostare signi-

ficativamente la statistica e rallentano la segnalazione di eventuali anomalie, rendendo questa carta equivalente all'EWMA tradizionale (come si vede dalla tabella 3.2).

scarto	EWMA	AEWMA $k = 4.2648$	AEWMA $k = 3.7190$	AEWMA $k = 3.2587$	AEWMA $k = 3.0902$
0.00	498.6897	499.4197	499.546	500.1558	499.5282
0.25	121.3911	121.6801	122.9936	130.9033	136.6959
0.50	34.2721	34.24438	34.53009	36.31515	37.62583
0.75	16.30784	16.29746	16.38704	16.91417	17.34831
1.00	10.20712	10.21100	10.24527	10.44298	10.64591
1.50	5.776916	5.774381	5.770319	5.779801	5.837528
2.00	4.074102	4.061307	4.025305	3.949896	3.941216
2.50	3.187964	3.154405	3.079348	2.936611	2.881027
3.00	2.643968	2.565783	2.443706	2.256669	2.177782
4.00	2.065173	1.778664	1.587136	1.417542	1.360377
5.00	1.786132	1.300393	1.160556	1.084595	1.065744

Tabella 3.2: Comportamento dell'AEWMA al variare del parametro k.

L'applicazione del *fast initial response* sembra migliorare significativamente il comportamento complessivo dell'EWMA, migliorando le prestazioni in particolare per scarti elevati. Al contrario l'effetto sull'AEWMA è piuttosto ridotto, come se la naturale reattività di questa carta garantisse una rapidità ottimale nella segnalazione degli errori anche senza bisogno di ulteriori miglioramenti.

Come si vede dalla tabella 3.3, che contiene i profili dell'*average run length* delle carte precedenti modificate con il FIR, le prestazioni dell'EWMA diventano paragonabili o addirittura migliori di quelle dell'AEWMA. Tuttavia si nota una drastica diminuzione nella mediana delle *run length* per il processo in controllo, che potrebbe indicare una tendenza a segnali d'allarme fasulli durante la fase di avviamento della carta.

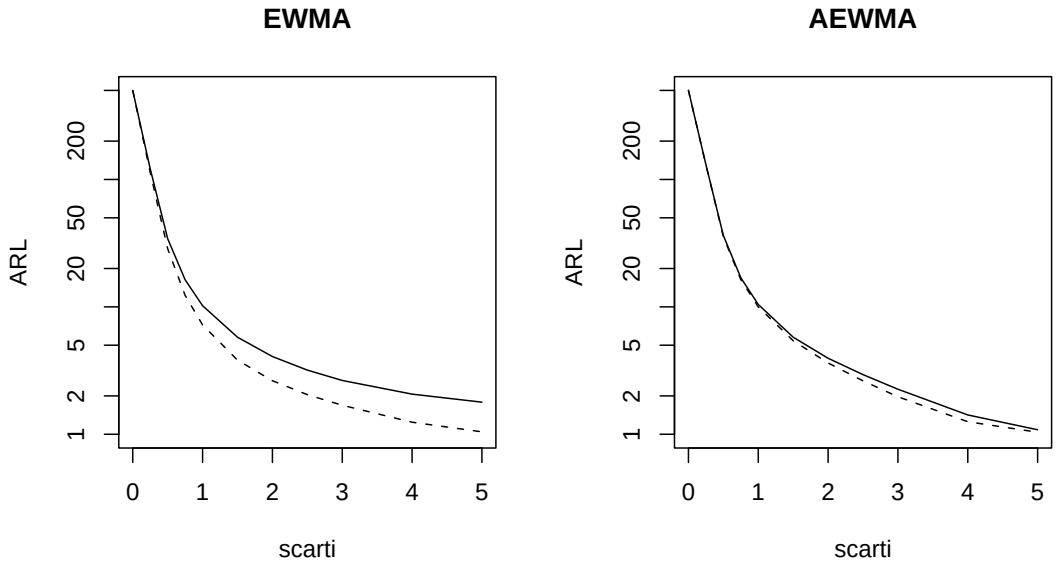

Figura 3.1: Miglioramenti apportati dal FIR.

	EWMA		AEWMA	
	$\lambda = .133$		$\lambda = .1354 \quad h = .7928267$	
	$h = .7688665$		$k = 3.2587$	
scarto	ARL	mediana	ARL	mediana
0.00	499.0436	338	500.5593	348
0.25	113.6423	77	130.1564	92
0.50	28.59582	20	35.60371	27
0.75	12.31396	9	16.36354	14
1.00	7.208975	6	9.990572	9
1.50	3.820696	3	5.422492	5
2.00	2.631152	2	3.625984	4
2.50	2.045524	2	2.62524	3
3.00	1.689084	2	1.969824	2
4.00	1.243036	1	1.253896	1
5.00	1.044738	1	1.037998	1

Tabella 3.3: Fast Initial Response (FIR) applicato ad EWMA ed AEWMA.

L'utilizzo dei confini mobili, pur garantendo risultati complessivamente migliori del *fast initial response*, non sembra causare alcun calo nella mediana delle *run length* del processo in controllo.

	EWMA		AEWMA	
	$\lambda = .133$		$\lambda = .1354 \quad h = 2.948888$	
	$h = 2.888284$		$k = 3.2587$	
scarto	ARL	mediana	ARL	mediana
0.00	500.3413	346	500.9855	347
0.25	119.4476	84	129.1030	91
0.50	32.45038	25	34.61926	26
0.75	14.68341	12	15.39986	13
1.00	8.660691	8	9.005223	8
1.50	4.335843	4	4.473143	4
2.00	2.745243	3	2.824083	3
2.50	1.977702	2	2.027465	2
3.00	1.547216	1	1.581156	1
4.00	1.135063	1	1.149601	1
5.00	1.017444	1	1.020146	1

Tabella 3.4: EWMA ed AEWMA con confini mobili.

Chiaramente in questo caso la "soglia ottimale" in quanto tale non esiste, quindi è necessario utilizzare un quantile delle osservazioni standardizzate; quest'ultimo verrà poi opportunamente moltiplicato di volta in volta per ottenere la soglia effettiva.

La differenza di prestazioni tra questi due approcci, al di là di altre considerazioni, dipende comunque dagli scarti rispetto a cui le due carte sono ottimizzati; l'AEWMA, per esempio, mostra margini di miglioramento molto superiori se impostata a 0.25σ e 1σ (anche se la mediana per il processo in controllo indica che probabilmente vi sono più falsi allarmi precoci). Questo probabilmente è dovuto al fatto che in questo caso l'*averge run length* per scarti grandi non è vincolato,e non ha quindi in origine un andamento ottimale; pertanto può essere notevolmente migliorato con l'introduzione del FIR o dele soglie mobili.

	FIR-AEWMA		AEWMA (c.m.)	
	$\lambda = .0137 \quad h = 0.1833171$		$\lambda = .0137 \quad h = 2.317535$	
	$k = 3.4473$		$k = 3.4773$	
scarto	ARL	mediana	ARL	mediana
0.00	500.4117	356	501.3085	316
0.25	77.59155	84	59.01833	45
0.50	33.15105	31	19.05973	15
0.75	20.77502	20	9.633566	8
1.00	15.02162	15	5.969093	5
1.50	9.403889	10	3.134404	3
2.00	6.403645	7	2.064704	2
2.50	4.357591	5	1.549897	1
3.00	2.899518	2	1.273388	1
4.00	1.445035	1	1.046636	1
5.00	1.070946	1	1.003671	1

Tabella 3.5: FIR-AEWMA ed AEWMA con confini mobili.

3.3 Conclusioni

L'*adaptive exponentially weighted moving average*, costruita come combinazione di carte Shewhart ed EWMA tradizionale, risulta più flessibile ed adattabile di quest'ultima. In particolare sembra adatta a proteggere contro scarti diversi, pur rimanendo semplice da implementare ed interpretare, ed è facilmente estendibile ad altre distribuzioni univariate e multivariate. La scelta dei parametri, in congiunzione la regolazione della soglia d'allarme tramite il processo di Robbins-Monroe, permette inoltre una configurazione relativamente rapida ed automatizzata.

L'uso di soglie mobili contribuisce ulteriormente a migliorare le prestazioni di questa carta in modo più immediato ed efficace del *fast initial response*. Lo stesso sembra accadere anche per l'EWMA tradizionale. L'alto grado di ottimizzazione dell'AEWMA, combinato con l'applicazione di questa tecnica, permette di ottenere prestazioni ottimali anche per scarti rispetto a cui la carta non è stata ottimizzata, ampliando

ulteriormente lo spettro di possibili anomalie segnalate tempestivamente.

Nel confronto tra EWMA e AEWMA si osservi infine che l'effetto del FIR e delle soglie mobili si esaurisce nella fase di avviamento della carta senza influenzarne l'andamento una volta raggiunto lo *steady state*. La capacità delle differenti carte di segnalare un allarme quando il cambiamento si manifesta in avanti nel tempo non risulta quindi modificata rispetto alle versioni senza FIR delle carte e quindi, visti i risultati in (Capizzi e Masarotto [3], 2003) l'AEWMA rimane consigliabile.

Appendice A

Codice sorgente di programmi utlizzati

A.1 Ambiente di lavoro

Per la compilazione e l'utilizzo dei programmi presenti in questo capitolo ho utilizzato la seguente macchina

Processore:	Dual Athlon MP 1200
RAM:	768 Mb
Sistema Operativo:	Debian GNU/Linux, kernel 2.4.25
Compilatore:	gcc versione 4:3.3.4-1
Libreria C:	2.3.2.ds1-13
R:	1.9.0

Il comando con cui ho compilato i sorgenti è

gcc -I/usr/lib/R/include -L/usr/lib/R/bin -lRmath -lm -O2 -Wall *file* -o *binario*

in modo da accedere agli header ed alle librerie presenti in R in aggiunta a quelle standard. Il motivo alla base di questa scelta è la disponibilità della funzione rnorm per

la simulazione dalla distribuzione normale, non presente nelle comuni librerie C. La disponibilità di altre librerie che replicano questa funzione può rendere inutile questo accorgimento.

Ho deciso di aggiungere anche le le opzioni -Wall per facilitare l'eventuale debug delle applicazioni e -O2 per ottenere un eseguibile ottimizzato.

A.2 EWMA

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

#define L 2.880695*sqrt(alpha/(2-alpha))
#define MEAN 0.00
#define VAR 1
//#define DEBUG

int main (int argc, char *argv[]) {

int i = 0;
int time = 1;                           // time
double alpha = .133;                    // weight
double serie = 0;                       // process

    for (i = 0; i < atoi(argv[1]); i++){
```

```
        serie = alpha * rnorm(MEAN, VAR);

        while ((serie > -L) & (serie < L)) {

            #ifdef DEBUG
            printf("%lf ", serie);
            #endif
            serie = (1 - alpha) * serie + alpha *
                                    rnorm(MEAN, VAR);
            time++;

        }//WHILE

        printf("%d\n", time);
        time = 1;

    }//FOR

    return 0;

}//MAIN
```

A.3 EWMA con Fast Initial Response

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

#define L 2.901284*sqrt(alpha/(2-alpha))
#define MEAN 5.00
#define VAR 1
//#define DEBUG

int main (int argc, char *argv[]) {

int i = 0;
int time = 1;                          // time
double temp = 0;
double alpha = .133;                   // weight
double upper_serie = 0.5 * L;          // process (upper)
double lower_serie = -0.5 * L ;        // process (lower)

    set_seed(563707652, 1250716184);
    for (i = 0; i < atoi(argv[1]); i++){

        temp = rnorm(MEAN, VAR);
        lower_serie = (1 - alpha) * (-0.5 * L)
                                        + alpha * temp;
        upper_serie = (1 - alpha) * ( 0.5 * L)
```

```
                                        + alpha * temp;

        while ((upper_serie > -L) & (upper_serie < L) &
               (lower_serie > -L) & (lower_serie < L)) {

            #ifdef DEBUG
            printf("%lf-%lf ", upper_serie, lower_serie);
            #endif
            temp = rnorm(MEAN, VAR);
            lower_serie = (1 - alpha) * lower_serie
                                        + alpha * temp;
            upper_serie = (1 - alpha) * upper_serie
                                        + alpha * temp;
            time++;

        }//WHILE

        printf("%d\n", time);
        time = 1;

    }//FOR

    return 0;

}//MAIN
```

A.4 EWMA con frontiere mobili

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

//#define L 2.880695*sqrt(alpha/(2-alpha))
#define L 2.888284*sqrt((1-pow(1-alpha,2*time)) \
                                    *alpha/(2-alpha))
#define MEAN 5.00
#define VAR 1
//#define DEBUG

int main (int argc, char *argv[]) {

int i = 0;
int time = 1;                           // time
double alpha = .133;                    // weight
double serie = 0;                       // process

    for (i = 0; i < atoi(argv[1]); i++){

        serie = alpha * rnorm(MEAN, VAR);

        while ((serie > -L) & (serie < L)) {

            #ifdef DEBUG
```

```
            printf("%lf ", serie);
            #endif
            serie = (1 - alpha) * serie +
                            alpha * rnorm(MEAN, VAR);
            time++;

        }//WHILE

        printf("%d\n", time);
        time = 1;

    }//FOR

    return 0;

}//MAIN
```

A.5 AEWMA

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

#define L 0.1759769
#define MEAN 5.00
#define VAR 1
#define KAPPA 4.2176
//#define DEBUG

double phi(double e, double k, double lambda) {

    if (e < -k) return e + (1-lambda) * k;
    else if (e > k) return e - (1-lambda) * k;
    else return lambda * e;

}//ALPHA

int main (int argc, char *argv[]) {

int i = 0;
double y;
int time = 1;                          // time
double alpha = .0137;                  // weight
double serie = 0;                      // process
```

```
set_seed(769704650, 1159710186);
for (i = 0; i < atoi(argv[1]); i++){

    y = rnorm(MEAN, VAR);
    serie = phi(y - 0, KAPPA, alpha);

    while ((serie > -L) & (serie < L)) {

        y = rnorm(MEAN, VAR);
        #ifdef DEBUG
        printf("k:%lf\ty:%lf\te:%lf\tphi(e):%lf
            \tw(e):%lf\tl:+/-%lf\n",
            serie, y, y-serie,
            phi(y-serie,KAPPA,alpha),
            phi(y-serie,KAPPA,alpha)/(y-serie),  L);
        #endif
        serie += phi(y - serie, KAPPA, alpha);
        time++;

    }//WHILE

    printf("%d\n", time);
    time = 1;

}//FOR
```

```
    return 0;

}//MAIN
```

A.6 AEWMA con Fast Initial Response

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

#define L 0.1833171  //2.856*sqrt(alpha/(2-alpha))
#define MEAN 5.00
#define VAR 1
#define KAPPA 3.4473
//#define DEBUG

double phi(double e, double k, double lambda) {

    if (e < -k) return e + (1-lambda) * k;
    else if (e > k) return e - (1-lambda) * k;
    else return lambda * e;

}//ALPHA

int main (int argc, char *argv[]) {

int i = 0;
double y;
int time = 1;                          // time
double alpha = .0137;                  // weight
double upper_serie = 0.5 * L;          // process (upper)
```

```
double lower_serie = -0.5 * L;        // process (lower)

    set_seed(769704650, 1159710186);
    for (i = 0; i < atoi(argv[1]); i++){

        y = rnorm(MEAN, VAR);
        lower_serie = phi(y - 0.5 * L, KAPPA, alpha);
        upper_serie = phi(y + 0.5 * L, KAPPA, alpha);

        while ((upper_serie > -L) & (upper_serie < L) &
               (lower_serie > -L) & (lower_serie < L)) {

            y = rnorm(MEAN, VAR);
            #ifdef DEBUG
            printf("lo:%lf\tup:%lf\tphi(lo):%lf
                \tphi(up):%lf\n",
                lower_serie, upper_serie,
                phi(y-lower_serie,KAPPA,alpha),
                phi(y-upper_serie,KAPPA,alpha));
            #endif
            lower_serie += phi(y - lower_serie,
                                        KAPPA, alpha);
            upper_serie += phi(y - upper_serie,
                                        KAPPA, alpha);
            time++;

        }//WHILE
```

```
        printf("%d\n", time);
        time = 1;

    }//FOR

    return 0;

}//MAIN
```

A.7 AEWMA con frontiere mobili

```
#include <stdio.h>
#define MATHLIB_STANDALONE
#include <Rmath.h>

#define L 2.317535 * sqrt((1 - pow(1 - alpha, 2*time)) \
                                    * alpha/(2 - alpha))
#define MEAN 5.00
#define VAR 1
#define KAPPA 3.4473
//#define DEBUG

double phi(double e, double k, double lambda) {

    if (e < -k) return e + (1-lambda) * k;
    else if (e > k) return e - (1-lambda) * k;
    else return lambda * e;

}//ALPHA

int main (int argc, char *argv[]) {

int i = 0;
double y;
int time = 1;                          // time
double alpha = .0137;                  // weight
```

```
double serie = 0;                       // process

    set_seed(769704650, 1159710186);
    for (i = 0; i < atoi(argv[1]); i++){

        y = rnorm(MEAN, VAR);
        serie = phi(y - 0, KAPPA, alpha);

        while ((serie > -L) & (serie < L)) {

            y = rnorm(MEAN, VAR);
            #ifdef DEBUG
            printf("k:%lf\ty:%lf\te:%lf\tphi(e):%lf
                \tw(e):%lf\tl:+/-%lf\n",
                serie, y, y-serie,
                phi(y-serie,KAPPA,alpha),
                phi(y-serie,KAPPA,alpha)/(y-serie),  L);
            #endif
            serie += phi(y - serie, KAPPA, alpha);
            time++;

        }//WHILE

        printf("%d\n", time);
        time = 1;

    }//FOR
```

```
    return 0;

}//MAIN
```

Bibliografia

[1] Brooks, D. ed Evans, D. A. (1972): *"An Approach to the Probability Distribution of CUSUM Run Length"*, Biometrika, 59, 539-549.

[2] Capizzi, G. e Masarotto, G. (1998): *"Calibrazione di una Carta di Controllo mediante Approssimazione Stocastica"*, Atti della XXXIX Riunione Scientifica della Società Italiana di Statistica.

[3] Capizzi, G. e Masarotto, G. (2003): *"An Adaptive Exponentially Weighted Moving Average Control Chart"*, Technometrics, 45, 199-207.

[4] Lucas, J. M. e Crosier, R. B. (1982): *"Fast Initial Response for CUSUM Quality-Control Schemes: Give Your CUSUM a Head Start"*, Technometrics, 24, 199-205.

[5] Lucas, J. M. e Saccucci, M. S. (1990): *"Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements"*, (con relative discussioni) Technometrics, 32, 1-29.

[6] MacGregor, J. F. e Harris, T. J. (1990): discussione relativa a *"Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements"* di J. M. Lucas e M. S. Saccucci, Technmetrics, 32, 23-26.

[7] Stroup, D. F. e Braun, H. I. (1982): *"On a New Stopping Rule for Stochastic Approximation"*, Z. Warsch. Verw. Gebeite, 60, 535-554.

[8] Wetherill, G. B. e Brown, D. W. (1991): *"Statistical Process Control"*, Chapman & Hall.

Finito di stampare il 25 giugno 2004 utilizzando LaTeX 2ε