Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche

Tesi di laurea

Sorveglianza della stabilità delle performance di processi in ambito industriale

Relatore Professoressa Giovanna Capizzi
Dipartimento di Scienze Statistiche

Laureando Alberto Moro
Matricola N 2018895

Anno Accademico 2023/2024

Indice

Introduzione

In questa relazione si vuole affrontare un problema specifico riscontrabile nel contesto dei dati funzionali applicati alle carte di controllo. Si propone una sintesi dell'informazione caratteristica di tale tipo di dato che possa essere utile per la formulazione di procedure più efficienti.

Le motivazioni che mi hanno spinto ad approfondire questo argomento hanno origine dall'interesse verso l'analisi e l'interpretazione di dati svolta principalmente per offrire risposte coerenti e soluzioni concrete alle diverse problematiche ed esigenze delle aziende.

La sorveglianza di processo e la diagnosi delle anomalie sono compiti importanti che determinano il successo del processo e la qualità del prodotto finale. Negli ultimi anni molte realtà produttive e del settore terziario hanno modificato i loro processi produttivi, rendendoli sempre più automatizzati e integrando nuove tecnologie al fine di migliorare la qualità dei servizi e l'efficienza della produzione. Una precoce rilevazione delle difformità è vantaggiosa per prendere misure correttive e decisionali opportune prima che il processo sia completato per impedire problemi in futuro. La raccolta avviene, sempre più spesso, tramite una molteplice quantità di sensori ad alta frequenza di campionamento, installati su macchinari per la produzione o su dispositivi di bordo. Per far fronte alla mole sempre maggiore di dati a disposizione, provenienti da diverse fonti, l'acquisizione dei dati deve essere condotta di conseguenza.

L'obiettivo della tesi è l'adattamento di opportuni metodi in presenza di dinamicità in dati complessi, come ad esempio i dati funzionali.

Le tecniche statistiche alla base dell'approccio sono l'Analisi delle Componenti Principali (PCA) e la Regressione Multivariata (PLS). Queste procedure si propongono di studiare l'andamento di serie di dati, definendo un modello statistico che cerchi di descrivere il comportamento di un fenomeno che debba essere tenuto sotto controllo. L'utilizzo di metodi a variabili latenti ha rivoluzionato il controllo di processo stesso, permettendo di rilevare precocemente guasti e fornire anche degli strumenti di diagnosi del problema. L'applicazione dei metodi

di analisi multivariata, infatti, è stata estesa anche per la sorveglianza online (si veda Kourti, 2005). In una prima fase si illustra questa nuova tipologia di dato, espresso sotto forma di più curve, e che viene definito dato funzionale. Approcci recenti adottano metodi di clustering per identificare gruppi omogenei di osservazioni che possono essere raggruppati per ottenere informazioni più accurate sulla distribuzione dei dati. Per gestire il problema della diversa durata dei fenomeni sono disponibili diverse tecniche che si occupano di sincronizzazione. Si possono, ad esempio, applicare delle procedure di *time-warping*, ovvero procedure di deformazione o registrazione, che permettono di ricondursi ad un dominio temporale comune (si veda Ramsay e Silverman, 2005). Viene illustrata l'applicazione dei metodi trattati ad alcuni insiemi di dati reali, attraverso l'utilizzo del software `R` e uno studio di simulazione per verificare l'accuratezza di alcuni tra i metodi presentati.

La trattazione si snoda in quattro capitoli che cercano di coprire ordinatamente gli argomenti proposti. Nel primo capitolo si offre una panoramica della tipologia di dato preso in analisi nell'ambito della sorveglianza statistica di processo e viene trattato il *clustering* funzionale. Nel secondo capitolo si presenta una descrizione della metodologia adeguata per il trattamento di dati funzionali attraverso le carte di controllo e l'analisi funzionale delle componenti principali. Con il terzo e quarto capitolo, di carattere operativo, si mettono in pratica i procedimenti teorici sopra trattati attraverso analisi su due dataset reali. Si effettuano degli studi di simulazione applicati anche a metodi tradizionali e si confrontano i risultati con i metodi precedentemente elaborati, al fine di comprendere quale tra questi sia il più idoneo a trattare una sintesi efficace dell'informazione.

Nell'Appendice A si presentano le librerie e i codici implementati per le procedure.

Di recente nella letteratura sulla sorveglianza statistica ha acquisito particolare attenzione l'applicazione del *clustering* funzionale e l'implementazione di carte di controllo funzionali per la sorveglianza della stabilità. Gli approcci affrontati in questa tesi dimostrano l'efficacia e la sensibilità delle carte di controllo funzionali illustrate.

Capitolo 1

Dati Funzionali

Il tema del controllo della qualità è oggetto di grande interesse in ambito aziendale. Il cambiamento delle richieste di mercato ha reso necessaria l'introduzione di metodi per lo *Statistical Process Monitoring* (SPM). Le caratteristiche che un prodotto deve garantire dipendono prevalentemente dal livello di qualità del processo di produzione, il quale deve assicurare prodotti finali conformi alle specifiche tecniche predefinite. Un'eventuale presenza di difetti è solitamente imputata alla variabilità del processo ed, in tal caso, l'azienda deve intervenire per migliorare le capacità e ridurre eventuali inefficienze (si veda Bottani et al., 2023).

In questo capitolo vengono presentati i dati funzionali e il loro impiego nel controllo statistico di processo. Se ne fornisce una esposizione di base, volta semplicemente ad introdurre il contesto e le notazioni utilizzate nel seguito. Per approfondimenti si veda Ramsay e Silverman (2005).

Per dato funzionale si intende un dato generato a partire da una funzione incognita, in presenza o meno di errore. Usualmente si cerca una funzione *smooth*, per cui esiste un numero sufficiente di derivate. Si formula un modello del tipo

$$y_i(t) = f(x_i(t)) + \varepsilon_i(t), \quad i = 1, \ldots, N, \ t \in \mathcal{T} \tag{1.1}$$

dove $y_i(t)$ rappresenta la variabile risposta funzionale, $x_i(t)$ la variabile esplicativa funzionale rilevata su N punti ed $\varepsilon_i(t)$ la componente di errore del modello.

Un modo per sintetizzare l'informazione contenuta nel dato funzionale è calcolarne media, varianza e funzione di covarianza come segue:

$$\bar{x}(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t) \tag{1.2}$$

$$\mathbf{var}_X(t) = \frac{1}{N-1}\sum_{i=1}^{N}[x_i(t) - \bar{x}(t)]^2 \tag{1.3}$$

$$\mathbf{cov}_X(t_1, t_2) = \frac{1}{N-1}\sum_{i=1}^{N}\{x_i(t_1) - \bar{x}(t_1)\}\{x_i(t_2) - \bar{x}(t_2)\} \tag{1.4}$$

Nel caso si osservino dati appaiati (x_i, z_i) è possibile quantificare la dipendenza tra le variabili della coppia tramite la funzione di cross-covarianza, definita come

$$\mathbf{cov}_{X,Z}(t_1, t_2) = \frac{1}{N-1}\sum_{i=1}^{N}\{x_i(t_1) - \bar{x}(t_1)\}\{z_i(t_2) - \bar{z}(t_2)\} \tag{1.5}$$

Per rappresentare il dato funzionale in modo diverso si può sfruttare una tecnica denominata Espansione tramite funzioni di base. Grazie a questo procedimento si può scrivere

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t), \tag{1.6}$$

dove $\phi_k(t)$ rappresentano le funzioni di base scelte e c_k i coefficienti associati a ciascuna funzione per $k = 1, \ldots, K$.

Particolari scelte per le funzioni di base possono essere compiute a seconda della natura del processo da cui il dato funzionale deriva. In caso di processo periodico con periodo T, ad esempio, si utilizza l'espansione in serie di Fourier. Tale espansione ha coefficienti del tipo

$$\phi_1(t) = 1, \phi_2(t) = \sin(\omega t), \phi_3(t) = \cos(\omega t), \phi_4(t) = \sin(2\omega t), \tag{1.7}$$

dove il parametro ω è legato al periodo dalla relazione $\omega = 2\pi/T$.

Se il processo non risulta avere periodicità possono essere scelte delle basi di tipo B-*spline*. Le *spline* sono delle funzioni polinomiali con grado specifico definite a tratti. Per la loro definizione devono essere specificati l'intervallo, l'ordine della *spline* e il numero di nodi. Il numero di funzioni di base K per le *spline* è dato dalla somma di ordine e numero di nodi interni. In ogni sottointervallo in cui la *spline* viene divisa, il grado del polinomio è fissato. In ogni punto in cui i polinomi confinano uno con l'altro le loro derivate fino ad un certo ordine devono coincidere.

1.1 Metodi di *clustering* funzionale

Di solito, i dati funzionali consistono in realizzazioni $X_1, \ldots, X_N$ indipendenti da una variabile casuale funzionale X che prende valori in uno spazio infinito-dimensionale, ad esempio $L^2(\mathcal{T})$, lo spazio di Hilbert delle funzioni a quadrato integrabile sul dominio compatto $\mathcal{T}$.

Queste realizzazioni non sono completamente disponibili, ma osservate solo in un insieme finito di punti. Pertanto si osserva solo $\{X_{ij}\}$ agli istanti temporali $\{t_{ij}, j = 1, \ldots, m_i\}$, dove m_i rappresenta il numero di punti discreti per l'osservazione i.

L'obiettivo dell'analisi dei *cluster* è definire M partizioni dei dati $X_1, \ldots, X_N$ in modo che osservazioni nello stesso *cluster* siano il più simili possibile, e ci sia molta diversità tra osservazioni in *cluster* diversi (Capezza et al., 2021).

1.2 Metodi di *clustering* dei dati grezzi

Il primo metodo di *clustering* consiste nel creare dei gruppi a partire dai dati grezzi discretizzati. Uno degli algoritmi più usati a questo scopo è il *k-means*, che partiziona le osservazioni in *cluster* minimizzando di volta in volta la somma dei quadrati all'interno dei singoli *cluster* fino a trovarne il numero e la composizione ottimale. In alternativa è possibile utilizzare altri algoritmi di *clustering*, come gerarchico o basato su un modello probabilistico (Hastie et al., 2009). In tutti i casi, il numero di *cluster* deve essere determinato basandosi su opportuni criteri.

Questa metodologia porta a diverse problematiche, poiché non tiene conto della possibile natura funzionale del dato a disposizione. Spesso il numero di punti su cui valutare la procedura è molto alto ed esiste una forte correlazione.

1.3 Metodi di *clustering* con filtro

I metodi di *clustering* con filtro si basano sulla ricostruzione delle osservazioni funzionali $\{X_i\}$ a partire dai dati discretizzati $\{X_{ij}\}$. L'approccio più comune consiste nell'assumere che le osservazioni funzionali siano parte di uno spazio funzionale di dimensione finita ed esprimibili tramite un insieme finito di funzioni di base. Ogni X_i può essere espresso come

$$X_i(t) = \mathbf{c}_i^T \phi(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \tag{1.8}$$

dove $\phi = (\phi_1, \ldots, \phi_K)^T$ rappresenta il vettore di funzioni di base sul sottoinsieme K-dimensionale di $L^2(\mathcal{T})$, e $\mathbf{c}_i$ il vettore dei coefficienti. Le funzioni di base possono essere specificate a priori, oppure basate sui dati come nel caso della *functional*-PCA (Hall e Hosseini-Nasab, 2006). Nel caso si utilizzino funzioni di base specificate a priori, se le variabili funzionali sono osservate con errore di misura il vettore di coefficienti viene stimato tramite minimi quadrati penalizzati con parametro di penalizzazione $\lambda > 0$. Il parametro di lisciamento λ è scelto come compromesso tra varianza e distorsione. Usualmente viene ottenuto come il valore che minimizza la funzione di convalida incrociata generalizzata.

Le covariate funzionali vengono, quindi, ricostruite come

$$X_i^{DA}(t) = \sum_{l=1}^{L} \xi_{il}\psi_l(t) = \xi_i^T \psi(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \tag{1.9}$$

dove $\psi = (\psi_1, \ldots, \psi_L)^T$ è il vettore delle L componenti principali e ξ_i il vettore degli *score*, definiti da $\xi_{il} = \int_{\mathcal{T}} \psi_l(t) X_i(t)\, dt$. Generalmente, la scelta di L viene fatta sulla base della percentuale di varianza spiegata dalle componenti principali.

La ricostruzione del dato funzionale permette, quindi, di ridurre la dimensionalità dei dati riassumendo ciascuna curva tramite un insieme finito di parametri. Infine si esegue un *clustering* multivariato standard su tale insieme di parametri.

1.4 Metodi di *clustering* adattivo

A differenza dei metodi spiegati nella sezione precedente, dove i coefficienti dell'espansione in basi vengono considerati parametri da stimare nella fase di lisciamento, gli stessi sono considerati variabili aleatorie con distribuzioni di probabilità specifiche per ogni *cluster*. La ricostruzione del dato funzionale in questo contesto avviene simultaneamente alla fase di *clustering*.

Come illustrato da James e Sugar (2003), se l'osservazione funzionale X_i appartiene all'm-esimo *cluster*, $m = 1, \ldots, M$, viene espressa tramite funzioni di base come

$$X_i(t) = \eta_{im}^T \phi(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \tag{1.10}$$

dove $\phi = (\phi_1, \ldots, \phi_K)^T$ sono *spline* cubiche naturali, e η_{ik} è un vettore di coefficienti casuali con distribuzione normale, definito da

$$\eta_{im} = \mu_m + \gamma_i, \tag{1.11}$$

dove μ_m è il vettore di coefficienti relativo alla media dell'm-esimo *cluster*, e

$\gamma_i \sim N(0, \Gamma)$ l'effetto casuale specifico per l'i-esima curva. Il vettore di valori discretizzati $\mathbf{X}_i = (X_{i1}, \ldots, X_{im_i})$ è modellato come

$$X_i = S_i(\mu_m + \gamma_i) + \varepsilon_i,$$

con $\mathbf{S}_i = (\phi(t_{i1}), \ldots, \phi(t_{im_i}))^T$ la matrice di realizzazioni associata al vettore ϕ, e $\varepsilon_i \sim N(0, \mathbf{R})$ il vettore di errori di misura. La matrice di covarianza $\mathbf{R}$ viene posta pari a $\sigma^2 I_{m_i}$. I parametri ignoti μ_m, Γ e σ vengono stimati tramite verosimiglianza mistura, dove il vettore rappresentante i *cluster* è modellato come una variabile aleatoria multinomiale con parametri $(\pi_1, \ldots, \pi_M)$, con π_m probabilità che un'osservazione appartenga al *cluster* m.

La verosimiglianza mistura è definita come prodotto di mistura delle densità condizionate di $(X_i|m) \sim N(S_i\mu_m, \Sigma_i)$, dove $\Sigma_i = \sigma^2 I_{m_i} + S_i \Gamma S_i^T$. La massimizzazione di solito è fatta tramite algoritmo EM (*Expectation-Maximization*, si veda Hastie et al., 2009). Dopo aver stimato i parametri ignoti, ogni curva X_i è assegnata al *cluster* con probabilità a posteriori $\pi_{m|i} = \hat{f}_m(X_i)\hat{\pi}_m / \sum_{j=1}^{M} \hat{f}_j(X_i)\hat{\pi}_j$ più alta. Vengono applicati criteri tradizionali come quelli AIC e BIC per la selezione del numero M di *cluster* e del numero di basi K.

L'uso di basi di tipo *spline* ha diversi svantaggi. Tali strumenti non sono adeguati se usati con funzioni molto irregolari, e richiedono un costo computazionale molto alto, pertanto non sono adeguati nemmeno se la dimensionalità dei dati è molto alta. Giacofci et al. (2013) propongono l'utilizzo di un metodo adattivo basato sulla decomposizione *wavelet* delle curve. Se l'osservazione funzionale X_i appartiene al *cluster* m, $m = 1, \ldots, M$, viene modellata come

$$X_i(t) = \mu_m(t) + U_i(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \tag{1.12}$$

dove μ_m rappresenta l'effetto fisso che caratterizza la m-esima media di *cluster* ed U_i una deviazione casuale da μ_m specifica per ogni curva. Applicando la trasformata wavelet discreta al modello (1.12) in cui è presente l'errore di misura $E_i(t), t \in \mathcal{T}$, il modello viene ridotto ad un modello lineare con effetti misti così formulato:

$$(c_i^T, d_i^T)^T = (\alpha_m^T, \beta_m^T)^T + (\nu_i^T, \theta_i^T)^T + (\varepsilon_{c_i}^T, \varepsilon_{d_i}^T)^T. \tag{1.13}$$

I vettori $(\alpha_m^T, \beta_m^T)^T$, $(\nu_i^T, \theta_i^T)^T$, $(\varepsilon_{c_i}^T, \varepsilon_{d_i}^T)^T$ e $(c_i^T, d_i^T)^T$ rappresentano, rispettivamente, i vettori dei coefficienti di scala e i coefficienti della *wavelet* associati a μ_m, U_i, E_i ed $X_i + E_i$; α_m e β_m sono parametri non casuali, mentre $(\nu_i^T, \theta_i^T)^T$ e

$(\varepsilon_{c_i}^T, \varepsilon_{d_i}^T)^T$ sono vettori casuali normali con media zero e matrici di covarianza $\mathbf{G}$ e $\sigma^2 I_{m_i}$, rispettivamente.

Una volta proiettato nel dominio della *wavelet*, il modello (1.13) torna ad essere un modello ad effetti casuali con varianza espressa in forma particolare. I parametri vengono stimati massimizzando la funzione di verosimiglianza sfruttando l'algoritmo iterativo EM. Infine si assegna ogni curva al proprio *cluster* massimizzando la probabilità a posteriori. Il numero ottimale di *cluster* viene scelto applicando metriche come il BIC.

L'ultimo metodo di *clustering* adattivo presentato utilizza per la variabile funzionale X_i appartenente al *cluster* m la seguente espansione in basi

$$X_i(t) = \gamma_{im}^T \mathbf{\Psi}(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \tag{1.14}$$

dove $\mathbf{\Psi} = (\Psi_1, \ldots, \Psi_K)^T$ è un vettore di funzioni di base assegnato, e γ_{im} è un vettore aleatorio K-dimensionale (si veda Bouveyron e Jacques, 2011). Le variabili X_i appartenenti allo stesso *cluster* sono descritte da un sottospazio funzionale latente a bassa dimensionalità $d_m < K$ con gruppo di funzioni di base $\{\varphi_{mj}\}$. I coefficienti latenti $\lambda_i = (\lambda_{i1}, \ldots, \lambda_{id_m})^T$ tra le funzioni di base specifiche per gruppo sono legate a γ_{im} nel modo seguente:

$$\gamma_{im} = \mathbf{U}_m \lambda_i + \varepsilon_i, \tag{1.15}$$

con $\varepsilon_i \in \mathbb{R}^K$ termine di errore, $\mathbf{U}_m$ matrice $K \times d_m$ formata dalle prime d_m colonne della matrice ortogonale $K \times K$ $\mathbf{Q}_m$, le cui entrate rappresentano i coefficienti che legano linearmente $\{\Psi_k\}$ e $\{\varphi_{mj}\}$. Se si assume $\lambda_i \sim N(\mu_m, \mathbf{S}_m)$, dove $\mathbf{S}_m = \text{diag}(a_{m1}, \ldots, a_{md_m})$, e $\varepsilon_i \sim N(0, \Xi_m)$ si ricava che

$$\gamma_{im} \sim N(\mathbf{U}_m \mu_m, \mathbf{Q}_m \mathbf{\Delta}_m \mathbf{Q}_m^T). \tag{1.16}$$

La matrice $K \times K$ $\mathbf{\Delta}_m$ è definita da $\mathbf{Q}_m(\mathbf{U}_m \mathbf{\Delta}_m \mathbf{U}_m^T + \mathbf{\Xi}_m)\mathbf{Q}_m^T$ e la matrice di covarianza del termine di errore $\mathbf{\Xi}_m$ è scelta in modo tale da rendere $\mathbf{\Delta}_m$ diagonale con primi d_m elementi pari alle componenti di S_m e restanti $K - d_m$ pari a b_m. La massimizzazione della verosimiglianza mistura avviene, come per il metodo precedente, tramite l'algoritmo EM. È possibile imporre vincoli sui parametri del modello. La dimensione del sottospazio latente d_m e il numero di $clusterM$ sono scelti tramite *scree-plot* e BIC.

1.5 Metodi di *clustering* basati sul concetto di distanza

Questa classe di metodi estende l'algoritmo di *clustering* geometrico classico al contesto funzionale. Si definisce una misura funzionale di prossimità tra le curve X_i e X_j tramite

$$d_l(X_i, X_j) = \left(\int_{\mathcal{T}} \left(X_i^{(l)} - X_j^{(l)} \right)^2 dt \right)^{1/2}, \tag{1.17}$$

dove $X^{(l)}$ rappresenta la derivata di ordine l di X. Il numero di *cluster* viene scelto calcolando l'indice di *silhouette*. Come per i metodi di *clustering* con filtro, prima si ricostruiscono le curve a partire dai dati discreti, poi si calcolano le misure di prossimità tra tutte le coppie di curve a disposizione. Per ulteriori dettagli si veda Capezza et al. (2021).

Capitolo 2

Sviluppo metodologia

Usualmente, nell'ambito dei modelli statistici per dati funzionali, si è interessati ad esaminare la relazione tra la variabile risposta e le variabili esplicative e per fare ciò si introduce un modello di regressione. In particolare, nel contesto funzionale che si va ad analizzare in questo e nei capitoli successivi, dove non specificato altrimenti, si ha a che fare con una variabile risposta scalare e covariate funzionali.

Ci si occupa di sorveglianza di dati profilo, si applica pertanto l'analisi delle componenti principali funzionali (MFPCA) alle variabili esplicative per cercare di modellare al meglio la relazione tra le esplicative e la risposta. Questa tecnica consente, inoltre, di estendere l'applicazione delle tecniche di sorveglianza di profili tramite carte di controllo basate su T^2 di Hotelling e *Squared Prediction Error* (SPE) alla sorveglianza congiunta delle diverse esplicative funzionali.

2.1 Strumenti di sorveglianza statistica

Di seguito se ne illustrano i passaggi, che vengono generalizzati anche per la sorveglianza in tempo reale delle covariate funzionali e della risposta scalare.

Fase I: si stima un modello di regressione con variabile risposta scalare e covariate funzionali basato su un insieme di dati di riferimento in controllo (IC), che dovrebbe contenere tutta l'informazione sugli scostamenti delle variabili dalle loro traiettorie medie in condizioni normali di funzionamento (si vedano Capezza et al., 2020 e Zhang et al., 2015);

Fase II: si sorvegliano le nuove osservazioni delle covariate funzionali tramite le carte di controllo funzionali T^2 e SPE, e le osservazioni relative alla risposta scalare tramite carta di controllo basata sui residui della regressione, *Response Prediction Error* (RPE). In questo modo si verifica se le nuove osservazioni manifestano un comportamento coerente con il campione di Fase I o paragonabile a

ciò che avviene in condizioni di processo fuori controllo (OC) (si vedano Kourti, 2005 e Nomikos e MacGregor, 1995);

Diagnostica: nell'ultima fase, quando si rileva una condizione di OC, si vogliono evidenziare tramite opportuni grafici le variabili più influenti (si veda Kourti e MacGregor, 1996);

2.2 Modello di regressione con risposta scalare e covariate funzionali

Si considerino $\tilde{X}_1, \ldots, \tilde{X}_P$ covariate funzionali con valori in $L^2(\mathcal{T})$.
Siano $\mu_p^X(t) = E[\tilde{X}_p(t)]$, e $\nu_p^X(t) = \mathbf{var}[\tilde{X}_p(t)]$ rispettivamente la funzione media e varianza della singola componente per ogni $p = 1, \ldots, P$ e funzione di correlazione $C = \{C_{p_1,p_2}\}_{p_1,p_2=1,\ldots,P}$, dove

$$\begin{aligned} C_{p_1,p_2}(t_1,t_2) &= Corr(\tilde{X}_{p_1}(t_1), \tilde{X}_{p_2}(t_2)) \\ &= Cov(\tilde{X}_{p_1}(t_1), \tilde{X}_{p_2}(t_2))\nu_{p_1}(t_1)^{-1/2}\nu_{p_2}(t_2)^{-1/2} \end{aligned} \tag{2.1}$$

Le variabili funzionali vengono, quindi, normalizzate tramite la trasformazione $\nu_p(t)^{-1/2}(\tilde{X}_p(t) - \mu_p^X(t))$.
Sia X_p la variabile funzionale normalizzata per $p = 1, \ldots, P$. Si denoti con y la variabile risposta scalare. Si consideri un campione casuale semplice di numerosità N da $(\tilde{X}, y)$.

La distribuzione condizionata di y_i data l'osservazione delle covariate funzionali normalizzate X_i è modellata tramite la seguente relazione di regressione:

$$y_i = \beta_0 + \sum_{p=1}^{P} \int_{\mathcal{T}} X_{ip}(t)\beta_p(t)\, dt \; + \varepsilon_i, \quad i = 1, \ldots, N \tag{2.2}$$

con β_0 numero reale e $\beta_1(t), \ldots, \beta_P(t)$ funzioni lisce da stimare. Le componenti ε_i rappresentano i termini di errore, indipendenti e identicamente distribuiti come una variabile casuale normale, con media 0 e varianza σ^2. Gli errori si assumono incorrelati con le covariate funzionali, ovvero $E(\varepsilon_i X_p(t)) = 0$ per ogni $i = 1, \ldots, N$, $p = 1, \ldots, P$ e $t \in \mathcal{T}$.

2.3 Fase I

In Fase I, come affermato precedentemente, si utilizza un dataset di riferimento. I coefficienti β_0 e $\beta = (\beta_1, \ldots, \beta_P)$ possono essere stimati risolvendo il problema

ai minimi quadrati. Poiché il problema non è risolvibile data l'infinita dimensionalità dello spazio funzionale, si riduce la dimensionalità del dataset applicando l'analisi funzionale delle componenti principali (f-PCA), conosciuta anche come espansione Karhunen-Loève, che permette di conservare l'informazione utile e scartare l'errore. Si ricorre a tale approccio sia per le covariate funzionali che per i parametri associati. Si può scrivere

$$X(t) = \sum_{l=1}^{\infty} \xi_l \psi_l(t) \tag{2.3}$$

dove le $\psi_l(t)$ rappresentano le L componenti principali funzionali associate ad $X(t)$ e formano una base ortonormale dello spazio infinito-dimensionale $\mathcal{H}$. Le componenti ξ_l sono variabili casuali con media nulla, incorrelate tra loro e con varianza pari all'autovalore corrispondente dell'operatore di covarianza.

Esprimendo i coefficienti $\beta(t)$ con le stesse componenti principali usate per $X(t)$, come

$$\beta(t) = \sum_{l=1}^{\infty} b_l \psi_l(t), \tag{2.4}$$

e sostituendo in (2.2) si ricava

$$y_i = \beta_0 + \sum_{l=1}^{\infty} \xi_{il} b_l \ + \varepsilon_i, \quad i = 1, \ldots, N, \tag{2.5}$$

dove ξ_{il} rappresentano gli *score* associati all'osservazione X_i. Grazie all'ortogonalità degli *score*, i coefficienti b_l possono essere stimati separatamente. Non potendo stimare infiniti parametri, si scelgono le componenti principali che massimizzano la varianza spiegata dalle stesse. Generalmente, si tengono le prime L e si considera un'approssimazione L-dimensionale per le covariate $X(t)$. L'insieme di componenti principali funzionali su cui concentrare l'attenzione è quindi $\mathcal{M} = \{1, \ldots, L\}$.

Una scelta parsimoniosa può essere fatta scartando tutte le componenti con varianza minore di un valore soglia per cui i risultati non risultano significativi per il problema di regressione.

Una statistica come la somma dei quadrati delle previsioni può essere calcolata tramite convalida incrociata *leave-one-out* come

$$PRESS = \sum_{i=1}^{N} (y_i - \hat{y}_{[i]})^2, \tag{2.6}$$

dove $\hat{y}_{[i]}$ rappresenta la previsione di y_i basata sul modello di regressione in cui non si considera l'osservazione i-esima. Si selezionano solo le componenti che fanno ridurre la quantità sopra specificata di una certa soglia.

Dopo aver stimato le funzioni media e varianza come indicato al Capitolo precedente (vedere formule 1.2 e 1.3), si procede alla stima dell'intercetta $\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} y_i$ e dei coefficienti

$$\hat{b}_l = \frac{\sum_{i=1}^{N} y_i \hat{\xi}_{il}}{\sum_{i=1}^{N} \hat{\xi}_{il}^2} \tag{2.7}$$

La stima di β può essere ottenuta tramite $\hat{\beta}(t) = \sum_{l\in\mathcal{M}} \hat{b}_l \hat{\psi}_l(t)$.

La previsione per y_i risulta $\hat{y}_i = \hat{\beta}_0 + \sum_{l\in\mathcal{M}} \hat{\xi}_{il}\hat{b}_l$.

Una stima della varianza dell'errore per il modello in Equazione (2.2) è

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N - L - 1}$$

2.4 Fase II

Si supponga che sia disponibile una nuova osservazione (X^{new}, y^{new}). I nuovi *score* $\{\hat{\xi}_l^{new}\}_{l\in\mathcal{M}}$ sono calcolati come

$$\hat{\xi}_l^{new} = \langle X^{new}, \hat{\psi}_l \rangle_{\mathcal{H}}, \quad l \in \mathcal{M} \tag{2.8}$$

Si assume che X^{new} sia completamente osservata sul dominio $\mathcal{T}$. Vengono definite due carte di controllo per la sorveglianza delle covariate funzionali, la T^2 di Hotelling e la SPE. La terza carta, denominata RPE, permette di controllare il comportamento della variabile risposta.

- **Carta di controllo $\mathbf{T^2}$ di Hotelling**: tiene sotto controllo la variabilità delle covariate funzionali nel modello di regressione. È definita nel modo seguente:

$$T^2 = \sum_{l\in\mathcal{M}} \frac{(\hat{\xi}_l^{new})^2}{\hat{\lambda}_l}, \tag{2.9}$$

  e la sua distribuzione dipende da quella degli *score* in $\mathcal{M}$, non nota. Un limite di controllo superiore può essere preso come il quantile empirico $(1 - \alpha_{T^2})$ dei valori della stessa ottenuti per il campione di Fase I.

- **Carta di controllo SPE**: tiene in considerazione la parte di variabilità non considerata dalla statistica T^2, approssimando X^{new} con

$$\hat{X}_L^{new}(t) = \sum_{l \in \mathcal{M}} \hat{\xi}_l^{new} \hat{\psi}_l(t),$$

ed è definita come

$$\begin{aligned} SPE &= ||X^{new} - \hat{X}_L^{new}||_{\mathcal{H}}^2 \\ &= \sum_{l \in \{1,\dots,N-1\} \setminus \mathcal{M}} (\hat{\xi}_l^{new})^2. \end{aligned} \tag{2.10}$$

Come per la statistica T^2, un limite di controllo superiore può essere preso come il quantile empirico $(1 - \alpha_{SPE})$ dei valori della stessa ottenuti per il campione di Fase I.

- **Carta di controllo RPE**: monitora la risposta scalare tramite i residui di previsione $y^{new} - \hat{y}^{new}$. Poiché la distribuzione assunta per gli errori sperimentali è normale i limiti di controllo inferiore e superiore L_{α_y} vengono ottenuti come quantile $(1 - \alpha_y/2)$ di una distribuzione t di Student con $N - L - 1$ gradi di libertà, calcolato come

$$L_{\alpha_y} = t_{N-L-1,1-\alpha_y/2} \left[\hat{\sigma}^2 \left(1 + \frac{T^2}{N-1} \right) \right]^{1/2}. \tag{2.11}$$

Si osserva come il limite dipenda dal valore della statistica T^2. Se la componente di errore non è gaussiana, i limiti della carta non possono essere uguali.

Controllare il *Familywise Error Rate (FWER)*

Usualmente, l'utilizzo simultaneo di tre carte di controllo richiede di selezionare i limiti di controllo in modo da controllare il FWER per un livello di significatività α. Siano α_{T^2}, α_{SPE} e α_y i livelli usati separatamente nelle carte T^2 di Hotelling, SPE e RPE rispettivamente. In caso le carte siano dipendenti tra loro viene utilizzata la correzione di Bonferroni, che garantisce una FWER pari o inferiore al livello di significatività specificato. Ciò si ottiene scegliendo α_{T^2}, α_{SPE} e α_y in modo che

$$\alpha_{T^2} + \alpha_{SPE} + \alpha_y = \alpha.$$

È possibile assegnare la stessa correzione a tutte e tre le carte, ovvero $\alpha_{T^2} = \alpha_{SPE} = \alpha_y = 1/3$; in alternativa, si separano i contributi delle carte che monito-

rano le covariate funzionali (T^2 ed SPE, $\alpha_{T^2} = \alpha_{SPE} = 1/4$) da quello riguardante la risposta (RPE, $\alpha_{RPE} = 1/2$).

2.5 Diagnostica guasti tramite diagrammi di contributo

Si può stabilire il comportamento di una nuova osservazione confrontando le statistiche T^2, SPE ed RPE con i rispettivi limiti di controllo ottenuti in Fase I. Si segnala un allarme se almeno una delle carte supera i limiti di controllo.

Si osserva che la statistica T^2 può essere riscritta come

$$\begin{aligned} T^2 &= \sum_{l \in \mathcal{M}} \frac{(\hat{\xi}_l^{new})}{\hat{\lambda}_l} \langle X^{new}, \hat{\psi}_l \rangle_{\mathcal{H}} \\ &= \sum_{p=1}^{P} \sum_{l \in \mathcal{M}} \frac{(\hat{\xi}_l^{new})}{\hat{\lambda}_l} \langle X_p^{new}, \hat{\psi}_{lp} \rangle. \end{aligned}$$

Per ogni carta di controllo si definiscono i contributi di ogni variabile funzionale, nel modo seguente:

$$CONT_p^{T^2} = \sum_{l \in \mathcal{M}} \frac{\hat{\xi}_l^{new}}{\hat{\lambda}_l} \langle X_p^{new}, \hat{\psi}_{lp} \rangle, \quad p = 1, \ldots, P, \tag{2.12}$$

$$CONT_p^{SPE} = ||X_p^{new} - \hat{X}_{Lp}^{new}||^2, \quad p = 1, \ldots, P. \tag{2.13}$$

Le statistiche T^2 e SPE sono sempre positive per definizione, il contributo $CONT_p^{T^2}$ definito poco sopra per qualche variabile può risultare negativo.

I contributi non hanno la stessa distribuzione per ogni variabile a disposizione, pertanto deve essere scelto un limite di controllo superiore appropriato per ogni contributo. Di solito lo si stima partendo dalla distribuzione empirica basata sul campione di Fase I.

Per quanto riguarda la carta RPE quando viene segnalato un allarme le possibili cause vanno cercate tra le covariate funzionali non incluse nel modello.

2.6 Carte di controllo basate sulla regressione con risposta e covariate funzionali

La carta di controllo per la regressione, presentata alla Sezione 2.2, viene vista come un caso particolare della carta di controllo per la regressione funzionale

(FRCC) proposta da Centofanti et al. (2021). Per sfruttare la FRCC è necessario definire il modello di regressione funzionale da adattare, il metodo di stima e la strategia di sorveglianza dei residui.

L'implementazione di un modello di regressione con risposta e covariate funzionali avviene tramite

$$y_i(t) = \beta_0(t) + \sum_{p=1}^{P} \int_{\mathcal{S}} X_{ip}(s)\beta_p(s,t)\, ds \; + \varepsilon_i(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \quad (2.14)$$

con $\beta_0 \in L^2(\mathcal{T})$ intercetta funzionale e $\beta = (\beta_1(t), \ldots, \beta_P(t))^T \in L^2(\mathcal{S} \times \mathcal{T})^P$ coefficienti funzionali. Le componenti ε_i rappresentano i termini di errore, indipendenti e identicamente distribuiti con media zero e funzione di covarianza $V_\varepsilon \in L^2(\mathcal{S} \times \mathcal{T})$. Si assuma che sia la risposta che le covariate siano state standardizzate. Successivamente si applica la f-PCA multivariata separatamente alle covariate e alla risposta funzionale y, per la quale si usano R componenti principali, ottenendo le autofunzioni $\psi_l^X = (\psi_{l1}^X, \ldots, \psi_{lP}^X)^T \in L^2(\mathcal{S})^P$ e $\psi_r^Y \in L^2(\mathcal{T})$, a cui corrispondono gli autovalori λ_l^X e λ_r^Y. Le covariate, la variabile risposta e i coefficienti funzionali possono essere rappresentati tramite versioni troncate o componenti principali. Il modello può, quindi, essere approssimato da

$$\xi_{ir}^Y = \sum_{l=1}^{L} \xi_{il}^X b_{lr} + \epsilon_{ir}, \quad r = 1, \ldots, R, \quad i = 1, \ldots, N. \quad (2.15)$$

Dopo aver ottenuto la stima per i coefficienti relativi alle basi tramite minimi quadrati e di conseguenza i valori previsti per la variabile risposta, si calcolano i residui funzionali come

$$e_i(t) = y_i(t) - \hat{y}_i(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, N,$$

rispetto alle prime K componenti principali, scelte in modo da soddisfare un vincolo sulla varianza spiegata.

Quando N non è particolarmente grande in presenza di cambiamenti nella media si preferisce utilizzare una versione studentizzata di tali residui in modo da pesare di più le covariate più estreme, nel modo seguente:

$$e_{i,stu}(t) = \frac{y_i(t) - \hat{y}_i(t)}{\left(\hat{\nu}_\varepsilon(t) + \Psi^Y(t)^T \hat{\Sigma}_\varepsilon \Psi^Y(t) \sum_{l=1}^{L} (\xi_{il}^X)^2 / \lambda_l^X\right)^{1/2}}, \quad t \in \mathcal{T}, \quad i = 1, \ldots, N, \quad (2.16)$$

dove $\Psi^Y = (\psi_1^Y, \ldots, \psi_R^Y)^T$ è il vettore delle componenti principali di y,
$\epsilon_i = (\epsilon_{i1}, \ldots, \epsilon_{iR})^T$ è il vettore degli errori del modello approssimato (2.15),

$$\hat{\nu}_\varepsilon^2(t) = \sum_{i=1}^{N} e_i(t)^2/(N-1), t \in \mathcal{T}$$

rappresenta lo stimatore della varianza dei residui funzionali e

$$\hat{\Sigma}_\varepsilon = \sum_{i=1}^{N} \epsilon_i \epsilon_i^T/N$$

lo stimatore della matrice di covarianza degli errori di Equazione (2.15).

La terza e ultima fase riguarda la sorveglianza dei residui basato sulle carte T^2 di Hotelling e SPE costruite sugli *score* ξ_k^e. Anche in questo contesto, un profilo fuori controllo è segnalato da una statistica che supera i limiti di controllo. Se si volessero sorvegliare anche le covariate funzionali si dovranno utilizzare le carte di controllo proposte al paragrafo 2.4.

Capitolo 3

Caso studio: valutazione della qualità delle giunzioni nelle saldature a punti

L'obiettivo di questo capitolo è illustrare l'applicazione di alcune metodologie trattate finora tramite l'utilizzo del software `R`. Vengono presentate alcune applicazioni delle procedure di *clustering* funzionale per affrontare il problema della ricerca di gruppi omogenei di curve.

Viene esposto uno studio di simulazione atto alla verifica empirica della bontà di adattamento dei diversi modelli, cercando di mostrare le potenzialità e l'applicabilità pratica dei metodi di *clustering* a dati rappresentati da curve.

Questo caso studio analizza dei dati provenienti da test effettuati in ambito industriale, più precisamente nel settore automobilistico. Per le analisi si considera la versione originale del *dataset* della libreria `curvclust` (Giacofci et al., 2012).

I codici utilizzati vengono riportati in Appendice A.1.

3.1 Descrizione del contesto

La saldatura a punti a resistenza (*resistance spot welding*, RSW) è la tecnica più comunemente usata per unire lamiere di spessori e materiali diversi durante l'assemblaggio di carrozzerie di automobili. La qualità dei giunti viene monitorata per garantire l'integrità e la solidità degli assemblaggi saldati in ogni veicolo. Il controllo di qualità si basa tipicamente su test eseguiti alla fine del processo RSW (off-line) su sottogruppi finiti attraverso la valutazione diretta e indiretta delle

caratteristiche dei giunti saldati. I giunti si formano applicando una pressione sull'area di saldatura tra due lati di lamiere in acciaio zincato per mezzo di due elettrodi di rame. La tensione applicata genera corrente tra il materiale perché la resistenza prodotta dai metalli provoca generazione di calore (effetto Joule) che fa aumentare la temperatura del metallo sulla superficie da saldare fino al punto di fusione. Grazie alla pressione meccanica degli elettrodi, il metallo fuso delle lamiere da unire si raffredda e si solidifica dando origine ad una saldatura solida.

Gli effetti che più influenzano questo processo sono la resistenza elettrica del metallo e l'area di contatto tra le lamiere. Questi effetti si sviluppano per mezzo del calore prodotto dal flusso di corrente e della pressione di chiusura generata dagli elettrodi di rame.

3.2 Struttura dei dati

I dati sono stati raccolti durante test di laboratorio RSW presso il Centro Ricerche Fiat e fanno parte del progetto ICOSAF (*Integrated COllaborative systems for SmArt Factory*). Il progetto si propone come obiettivo la riconfigurazione degli impianti al fine di migliorarne i processi produttivi.

Per migliorare l'analisi del processo RSW si studiano le osservazioni della curva di resistenza dinamica (DRC) che rappresenta la curva convenzionale ed è riconosciuta come la firma tecnologica della saldatura a punti (Figura 3.1). L'osservazione delle curve permette, infatti, di individuare saldature a punti dello stesso tipo e di identificarne le proprietà che contribuiscono ad ottimizzare il processo di saldatura.

I dati grezzi sono organizzati in colonne $(x, V1, V2, \ldots)$, dove $V1, V2, \ldots$ sono i valori di resistenza elettrica utilizzati per ottenere ogni DRC, ricavati non direttamente ma ottenuti secondo la prima Legge di Ohm come rapporto tra la tensione degli elettrodi e le misure di intensità di corrente. L'energia è stata fornita da un singolo impulso di corrente. Il periodo di saldatura è di 237 ms. Le misure di tensione e corrente sono raccolte in una griglia di 238 punti equispaziati distanti tra loro 1 ms.

In Figura 3.1 sono evidenziate le caratteristiche usate per studiare il comportamento tipico del DRC.

I Differenza di ampiezza tra valori minimi e massimi della resistenza

II Differenza di fase tra ascisse massime e minime

III Valore finale della resistenza

IV Intervallo temporale tra massimo locale e valore finale

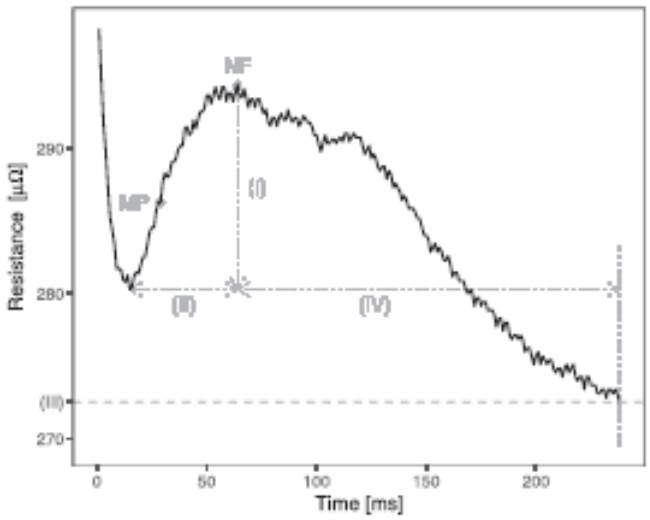


Figura 3.1: Comportamento tipico curve DRC

Il punto di flesso che si trova tra il minimo e il massimo locale del DRC rappresenta il punto di fusione della saldatura (MP). Il punto di massimo locale NF è l'inizio della formazione della saldatura solida (pepita).
I valori di DRC aumentano con la resistenza elettrica del metallo e la temperatura del materiale, diminuiscono con l'area di contatto.

I valori tipici diminuiscono all'inizio fino al minimo locale, per poi aumentare per effetto della resistenza, fino al punto di massimo locale. Successivamente, i valori diminuiscono fino al punto finale per effetto della fusione del metallo nell'area di contatto.

Il dataset `data.csv` è costituito da 538 DRC (Capezza et al., 2021). Ogni unità statistica (DRC) si riferisce a giunti di saldatura con la stessa posizione su diversi attacchi di 2 lamiere in acciaio zincato, con spessore pari a 0.7 e 1.3 mm, rispettivamente.

Le curve rappresentative dei dati grezzi sono riportate in Figura 3.2. L'asse delle ascisse riporta i vari tempi di registrazione delle misure, mentre in ordinata si hanno i corrispondenti valori di resistenza espressi in micro Ohm.

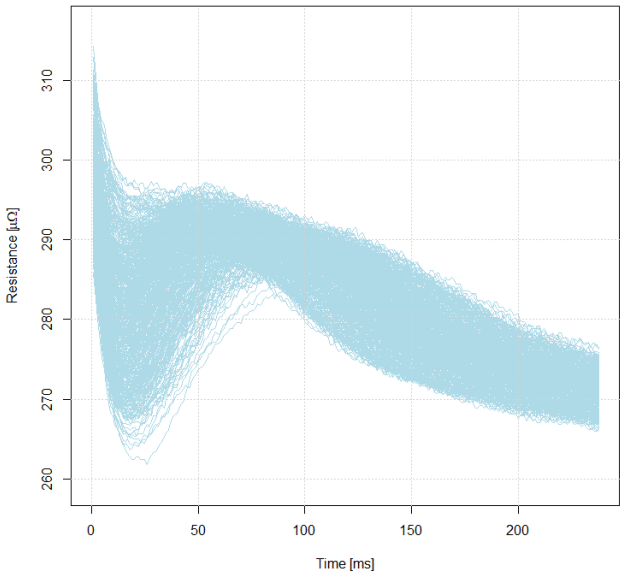


Figura 3.2: Grafico delle curve DRC originali

3.3 Analisi e interpretazione dei dati

Come si può notare dal grafico dei dati grezzi, i 538 DRC hanno un comportamento non sempre coerente con quello della Figura 3.1, probabilmente dovuto alla variabilità naturale tra i profili. Si prova, pertanto, ad identificare gruppi omogenei di DRC in modo che siano facilmente distinguibili. L'obiettivo è trovare saldature a punti con proprietà simili. Vengono proposti metodi di *clustering* funzionale che possono essere applicati in modo automatico.

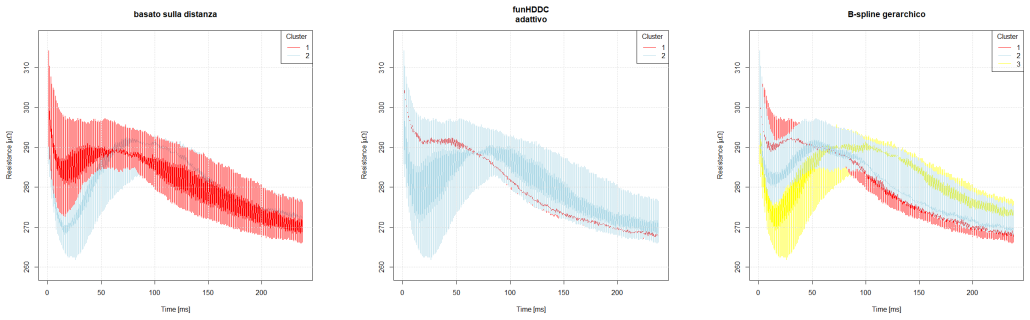
Adattando i diversi metodi di *clustering* ai dati originali od opportunamente trasformati, come spiegato nel Capitolo 1, si nota innanzitutto che i metodi di tipo adattivo sono computazionalmente molto più onerosi rispetto a tutti gli altri. Pur facendo uso di un processore Intel i7 da 3 GHz, i tempi di esecuzione per avere i risultati sono dell'ordine di diverse ore. Si tenga presente che per effettuare i calcoli nel caso dei dati grezzi originali, data la forte correlazione presente, si è scelto di lavorare con un sottoinsieme di istanti temporali, sempre equidistanti tra loro, estratti dal dataset iniziale. Nello specifico si seleziona un tempo ogni 13, per ottenere 19 istanti temporali per ogni curva; questo passaggio viene utilizzato solamente per questo tipo di modello, tutti gli altri modelli, che non risentono del problema della multicollinearità non richiedendo inversioni esplicite di matrici, non necessitano di questa operazione.

Si è provato a specificare un numero di *cluster* M variabile da 2 a 10 per l'adattamento delle diverse procedure.

Per procedere con le analisi di Fase I si deve esprimere la funzione $x(t)$ come combinazione lineare di funzioni di base e vettore di coefficienti ad essi associati (formula 1.8). Per la scelta del numero ottimale di basi si ricorre alla regola del "gomito", utile per stabilire una soglia minima di varianza durante l'applicazione della convalida incrociata (si veda Auer et al., 2008).

Si adatteranno sempre delle basi di tipo B-*spline* cubiche, con un singolo nodo per ogni punto in cui cambiano i polinomi che compongono la *spline*. Si terrà conto del fatto che, in ognuno di questi punti, le prime due derivate dei polinomi devono essere uguali tra loro.

In una prima fase di analisi tutte le curve originali vengono rappresentate tramite *cluster*. I grafici mostrano come i diversi metodi non portino ad ottenere sempre lo stesso numero di *cluster*. Alcuni metodi identificano come numero ottimale di gruppi il minimo stabilito. Il metodo basato sulla distanza (formula 1.17), il metodo gerarchico applicato ai dati grezzi e i metodi adattivi tramite funzione `funHDDC` (formula 1.14) e `curvclust` (formula 1.12) selezionano due *cluster*, altri basati sull'identificazione di un modello, sia tramite B-*spline* che componenti principali funzionali, tendono a raggruppare i DRC in molti più gruppi, che potrebbero portare a classificazioni informative meno affidabili.



(a) *Distanza con criterio silhouette* (b) *Adattivo tramite funzione funHDDC* (c) *B-spline con metodo gerarchico*

Figura 3.3: Grafici dei dati funzionali. Curve colorate in base al *cluster* di appartenenza

Il Grafico di Figura 3.3 mostra le DRC colorate in base all'assegnazione dei *cluster* ottenuta con i metodi specificati. L'interpretazione non risulta agevole dato l'elevato numero di curve, pertanto per le successive analisi si procede con la determinazione delle curve medie, ossia i centroidi, che verranno utilizzate per tutti i modelli elaborati.

Due dei principali algoritmi di *clustering* sono *k-means* e gerarchico. Nel contesto funzionale il primo metodo mira a dividere le osservazioni in M gruppi in modo da minimizzare la somma dei quadrati all'interno di ciascun *cluster*. Il metodo gerarchico, invece, fornisce una rappresentazione in cui i *cluster* ad ogni livello della gerarchia sono formati da tutti e solo dai *cluster* dei livelli più interni. Le diverse strategie per elaborare quest'ultimo metodo si basano su due approcci principali:

- divisivo, che inizia collocando tutte le osservazioni in un unico gruppo per poi dividerle in base ad un criterio stabilito;
- agglomerativo (usato nello studio tramite la funzione `hclust` del linguaggio `R`), che inizia con un *cluster* per ogni osservazione e unisce, ad ogni passo, due gruppi di osservazioni per formarne uno più grande in base alla loro somiglianza.

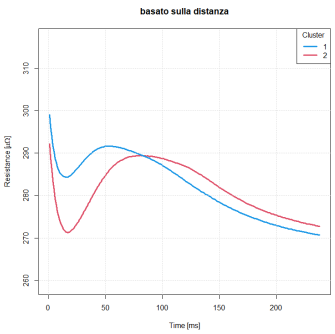
Per la scelta del numero di *cluster* in entrambi i metodi descritti viene usato l'indice di silhouette definito per ogni osservazione come

$$s(i) = \frac{\max(a(i), b(i))}{b(i) - a(i)},$$

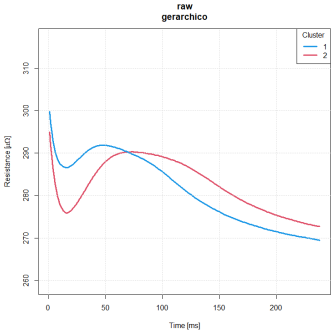
dove $a(i)$ rappresenta la dissimilarità media tra il punto i e gli altri punti dello stesso gruppo, $b(i)$ la minima distanza tra l'osservazione i e il suo gruppo più vicino. Le osservazioni con $s(i)$ grande (intorno ad 1) sono ben raggruppate, un indice pari a zero sta ad indicare che l'osservazione si trova tra due *cluster*. In caso di valore negativo, si ipotizza che il *clustering* sia errato e che l'osservazione dovrebbe fare parte di un altro gruppo.

I Grafici di Figura 3.4 mostrano le DRC singole colorate in base al *cluster* di appartenenza. Sono evidenziati i centroidi dei due *cluster* ottenuti. I centroidi dei diversi *cluster* hanno punti di minimo locale con ascissa simile, ma valori diversi di resistenza; punti di massimo locali simili con ascissa diversa; valori di resistenza diversa alla fine del processo.
Nel *cluster* 1, dove la differenza di fase è più bassa, si osserva che l'intervallo di tempo tra il punto di inizio saldatura (NF) e la fine del processo è più lungo.

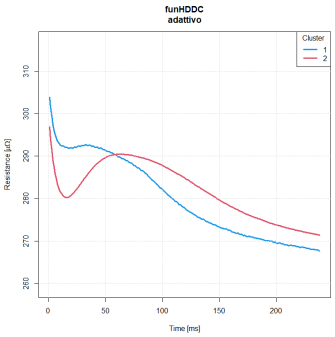
Più ampio è lo spazio tra il punto di massimo locale e la fine del processo stesso, più lunga è l'esposizione al calore, più grande è la saldatura solida (pepita) prodotta. Nel secondo *cluster* la differenza di fase, invece, è più alta, l'intervallo di tempo leggermente inferiore rispetto a quello evidenziato nel *cluster* 1. Si presuppone la formazione di una pepita più piccola, quindi una saldatura meno resistente.



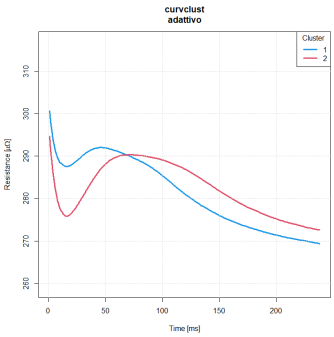
(a) *Distanza con criterio silhouette*



(b) *Metodo gerarchico applicato ai dati grezzi*



(c) *Adattivo tramite funzione funHDDC*

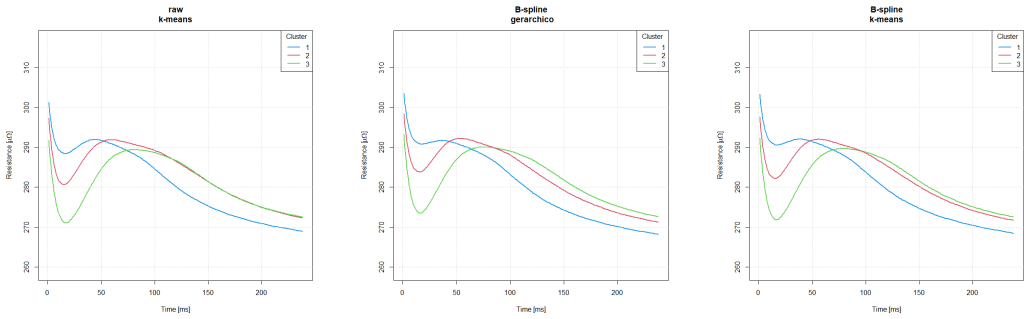


(d) *Adattivo tramite funzione curvclust*

Figura 3.4: Grafici dei centroidi ottenuti con i diversi metodi

In base alla regola di maggioranza, il numero di *cluster* ottimale risultante dai diversi modelli di Figura 3.5 è 3. Pochi *cluster*, infatti, forniscono gruppi di funzioni distinte che permettono di avere classificazioni più informative. Si focalizzano le analisi sull'interpretazione e la caratterizzazione dei DRC appartenenti agli stessi *cluster*.

I centroidi associati al *cluster* 1 hanno differenza di ampiezza tra valori minimi e massimi di resistenza, differenza di fase tra ascisse massime e minime e valore finale della resistenza più piccoli di quelli dei *cluster* 2 e 3. I valori di minimo locale nella porzione del primo dominio nel *cluster* 1 sono più grandi e diminuiscono più velocemente nell'ultima parte del dominio rispetto ai *cluster* 2 e 3. La differenza di ampiezza tra valori minimi e massimi della resistenza, la differenza di fase tra ascisse massime e minime e il valore finale della resistenza aumentano contemporaneamente passando dal *cluster* 1 al *cluster* 3. L'ascissa del punto di minimo è costante. Per il *cluster* 3 si nota una resistenza minima molto più bassa nella prima parte del dominio e un intervallo di tempo minore tra il punto NF e la fine del processo.



(a) *Metodo k-means applicato ai dati grezzi*

(b) *Metodo gerarchico applicato ai dati lisciati tramite B-spline*

(c) *Metodo k-means applicato ai dati lisciati tramite B-spline*

Figura 3.5: Grafici dei centroidi ottenuti con i diversi metodi

Si ipotizza, pertanto, una pepita più piccola con una saldatura meno resistente.

Le curve appartenenti al *cluster* 2 evidenziano una differenza di fase minore e un intervallo di tempo maggiore tra il punto di massimo locale NF e la fine del processo. Pertanto, data l'esposizione più alta all'energia termica, avranno una pepita più grande. Questo presuppone una maggior tenuta della saldatura stessa.

Parametri quali corrente, pressione e tempo sono collegati tra loro e influenzano il risultato finale del processo di saldatura, come ad esempio la larghezza del punto di saldatura e l'usura dell'elettrodo.

Le curve mostrano un andamento variabile con il tempo, indice di un possibile collegamento tra l'usura dell'elettrodo e le variazioni delle curve stesse.
L'usura dell'elettrodo si manifesta con l'aumento graduale della superficie di contatto tra l'elettrodo stesso e la zona di saldatura, una minor intensità di corrente di saldatura. Il risultato sarà, in questo caso, una minor resistenza meccanica della saldatura finale.

Per meglio comprendere l'impatto dell'usura dell'elettrodo, si prende come riferimento il metodo di filtraggio gerarchico tramite B-*spline* e vi si sovrappongono cinque DRC casuali tra le 538 originali colorate con lo stesso colore del *cluster* di appartenenza, ma con tre tipi di linea, continua, tratteggiata o punteggiata (si veda Figura 3.6). Mentre il comportamento del centroide del *cluster* 1 è in linea con il comportamento del DRC con elettrodo nuovo, quello del *cluster* 3 evidenzia un elettrodo usurato. Il risultato è confermato anche dal fatto che l'area di contatto tra l'elettrodo e la zona di saldatura aumenta con l'avanzamento dell'usura dell'elettrodo.

In conclusione, le curve appartenenti al *cluster* 2, sembrerebbero indicare una performance migliore del processo di saldatura a punti sia per quanto riguarda la larghezza del punto di saldatura che l'usura dell'elettrodo.

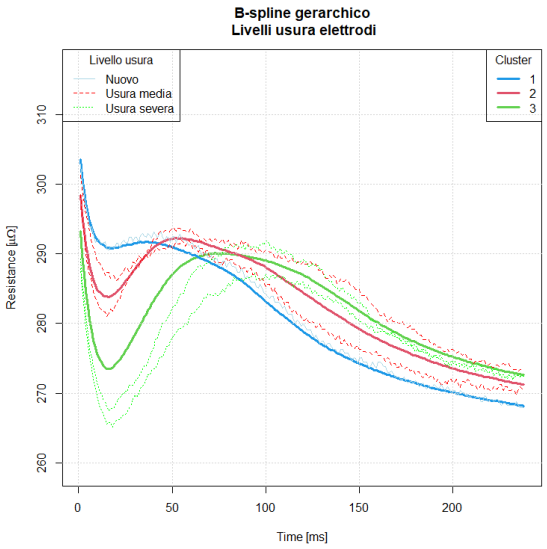


Figura 3.6: Grafico dello stato di usura degli elettrodi su cinque delle curve DRC originali

Osservazioni

Confrontando i diversi metodi di *clustering* adattati, si osserva come quelli che hanno portato all'identificazione di tre gruppi mostrino centroidi con comportamento simile. Un basso numero di *cluster* fornisce gruppi di funzioni distinte che hanno maggiori probabilità di portare a classificazioni informative. Una differenza nell'andamento delle curve si nota nei metodi che identificano solamente due gruppi, dove si osserva una maggior discordanza, specialmente nel primo periodo di osservazione.

I metodi di *clustering* funzionale potrebbero essere adottati per analisi SPM online. Identificati i gruppi associati a differenti caratteristiche di qualità del processo, si può procedere ad eseguire test di qualità solo sui gruppi più critici. Per quanto riguarda i *cluster* del caso reale trattato, se la distanza per un futuro DRC è troppo alta, la saldatura a punti viene segnalata dalla carta di controllo come anomala rispetto al campione di riferimento utilizzato per l'identificazione dei *cluster*.

3.4 Metodo aggiuntivo per la sorveglianza tramite carte di controllo

Nel caso specifico della saldatura a punti RSW l'obiettivo primario è sorvegliare i valori delle curve DRC rilevati al fine di ottimizzare il risultato della saldatura delle due lamiere in acciaio zincato. Un'analisi accurata delle curve permetterà all'azienda di intraprendere accurati programmi di manutenzione.

3.4.1 Carta di controllo per sorveglianza di Fase II

In caso di acquisizione di nuove osservazioni, il problema è raggruppare anche questi profili nei *cluster* identificati in Fase I.
Si può utilizzare un metodo di *clustering* funzionale per assegnare un'osservazione ad uno dei *cluster* precedentemente identificati. Si seleziona il *cluster* con il centroide che raggiunge la distanza minore dal futuro DRC, e si ipotizza una sua collocazione nello stesso.
Come misura di distanza viene usata L^2 dal centroide più vicino (formula 1.17), che può essere adottata come statistica di sorveglianza.

Prima si esegue l'operazione di *clustering*, successivamente per ogni *cluster* si calcolano le statistiche di sorveglianza. Infine si sceglie il limite di controllo basandosi sull'errore di prima specie desiderato e sulla distribuzione della statistica L^2 osservata per le curve di quel *cluster*.

Di seguito un esempio di possibile applicazione del *clustering* funzionale alle carte di controllo.

Il dataset `test_data.csv` utilizzato per l'approfondimento riporta i valori di 10 curve precedentemente lasciate fuori dal dataset di 538 profili usate per le analisi alla sezione 3, più altre 3 curve riferite a saldature affette da schizzi di saldatura, un problema causato dal fatto che il metallo liquido caldo finisce su altre parti della lamiera (Capezza et al., 2020).

Per l'implementazione della carta di controllo si calcola la distanza L^2 (formula 1.17) sui 13 profili selezionati. Il limite di controllo superiore (UCL) viene impostato per ogni *cluster* come il quantile empirico $(1 - \alpha)$ delle distanze L^2, dove α rappresenta il tasso di falso allarme (Capezza et al., 2021).

La matrice delle distanze tra le curve e i centroidi corrispondenti ha la seguente struttura:

Tabella 3.1: *Struttura della matrice distanze per 13 profili fuori controllo*

DRC	osservazioni	distanza	*cluster*	limiti
V121	1	5.07	2	19.38
V122	2	7.81	2	19.38
V126	3	1.76	2	19.38
V14	4	9.21	2	19.38
V205	5	3.53	3	32.01
V34	6	4.97	3	32.01
V411	7	101.23	3	32.01
V53	8	18.94	2	19.38
V574	9	96.56	3	32.01
V617	10	81.53	1	8.44
V89	11	3.54	2	19.38
V90	12	7.23	3	32.01
V95	13	5.14	2	19.38

Con i dati di Tabella (3.1) si definisce una carta di controllo basata sulla regressione con risposta scalare e covariate funzionali delle distanze di ogni curva aggiustate per l'effetto delle covariate funzionali adottate (Capezza et al.,2021).

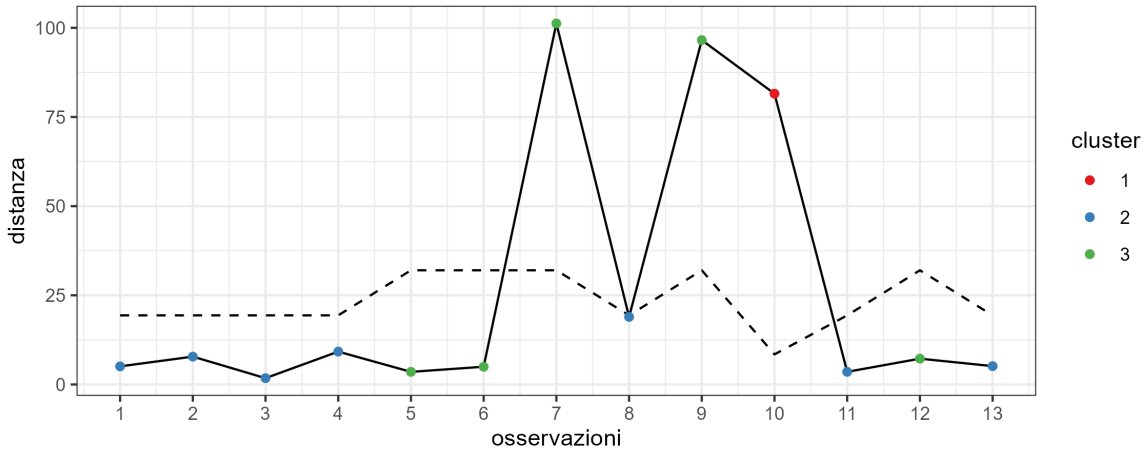


Figura 3.7: Carta di controllo basata sulle distanze di ogni curva dal centroide più vicino

La carta di controllo in Figura 3.7 evidenzia i punti corrispondenti ad un profilo, e riporta il valore della distanza. I punti sono colorati in base al *cluster* più vicino. La linea tratteggiata rappresenta i limiti superiori di controllo. I limiti di controllo sono variabili perché le carte di controllo sono costruite separatamente per ogni *cluster*. Ogni nuova curva viene prima assegnata ad uno dei tre *cluster* e poi monitorata con la corrispondente carta di controllo.

I punti che si trovano al di sopra dell'UCL sono considerati fuori controllo, e si ipotizzano parte del campione di riferimento utilizzato per identificare i *cluster*. Si convalida, pertanto, il fatto che le osservazioni inizialmente scartate siano effettivamente non in linea con il resto della produzione. La Figura 3.8 mette, infatti, in evidenza i tre profili fuori controllo, con resistenze minime piuttosto basse e valori di resistenza finale vicini al minimo globale.

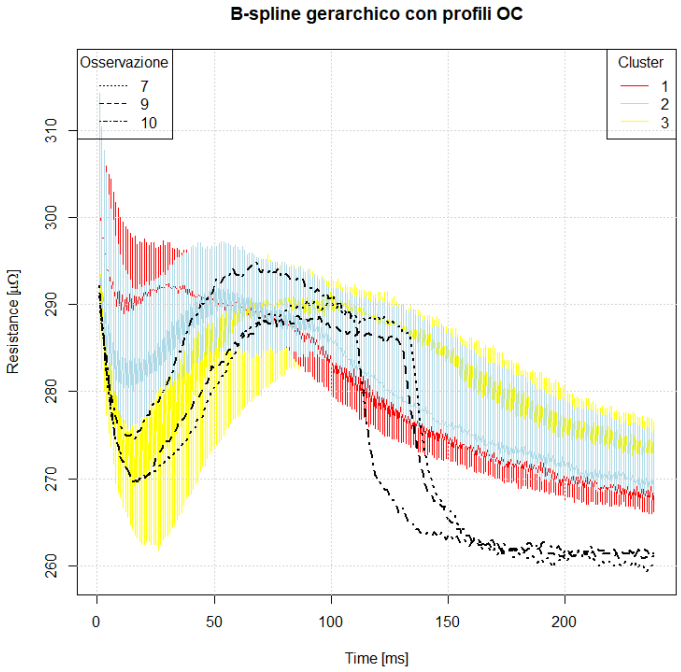


Figura 3.8: Grafico delle curve DRC originali filtrate tramite B-*spline* con metodo gerarchico, evidenziati i tre profili identificati dalla carta di controllo

3.5 Confronto tra *clustering* funzionale e metodi tradizionali

L'obiettivo di questo paragrafo è illustrare l'applicazione di metodologie alternative a quelle trattate nei capitoli precedenti. Si confrontano i risultati ottenuti dall'uso di alcuni metodi di *clustering* funzionale con due approcci multivariati basati su caratteristiche comunemente adottate per sintetizzare i processi RSW. Viene esposto uno studio di simulazione tramite *bootstrap* per confrontare le diverse procedure in base all'indice di *silhouette*, quando si vuole discriminare tra il modello multivariato e i metodi per dati funzionali, per stabilire se l'utilizzo di un metodo multivariato, non adatto in questo contesto, fornisce risultati comparabili con un metodo potenzialmente più efficace.

3.5.1 Metodo multivariato applicato ai dati originali

In questa sezione si presenta un approccio più semplice dei metodi funzionali visti finora, in cui si tengono in considerazione le caratteristiche tipicamente adottate per la sintesi dell'informazione contenuta nei DRC, ovvero la differenza di ampiezza tra valori massimi e minimi di resistenza, la differenza di fase tra ascisse massime e minime, il valore finale della resistenza e il tempo trascorso tra il massimo locale e il valore temporale finale (si veda Figura 3.1).

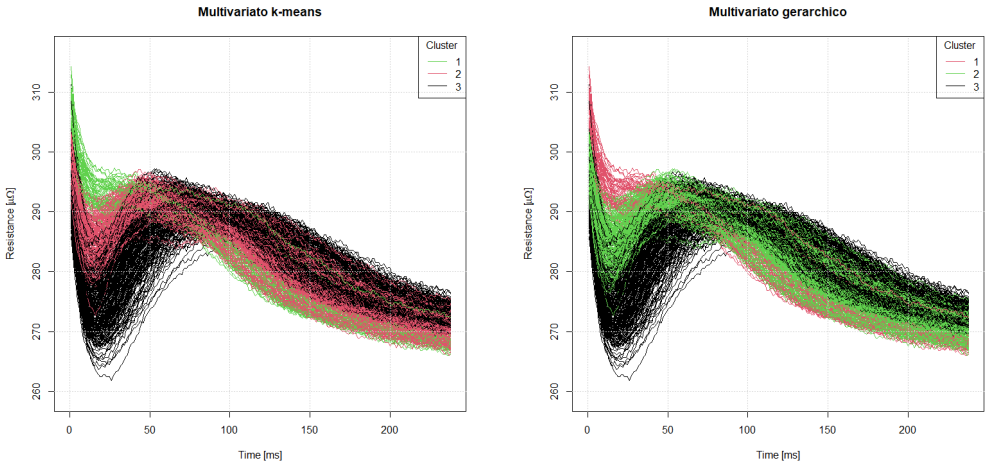
Per le analisi si considera la versione ridotta, presente anche nella libreria `funcharts` (Capezza et al., 2023).

Di seguito viene riportato un estratto della matrice ottenuta calcolando le quantità sopra descritte:

Tabella 3.2: *Struttura della matrice ottenuta con metodo multivariato*

DRC	Resistenza finale ($\mu\Omega$)	$\Delta t(ms)$	ΔRes ($\mu\Omega$)	Δt_{finale} (ms)
1	272.2656	84	21.389465	138
2	273.2309	45	11.074066	173
...	...	...	...	...
63	275.0314	36	6.399872	186
64	273.2309	68	7.206024	148
...	...	...	...	...
189	271.7929	39	14.750549	185
190	274.0938	41	19.557404	179
...	...	...	...	...
537	268.2448	38	9.939301	185
538	266.9035	19	4.177826	208

I grafici dei profili in Figura 3.9 mostrano le curve DRC colorate in base al *cluster* corrispondente, in entrambi i metodi il numero di *cluster* selezionati risulta essere 3. Per confrontare tra di loro metodi multivariati basati sulle caratteristiche e metodi funzionali, adattati tramite metodi gerarchico o *k-means*, si ricorre all'indice di *silhouette* basato sulla distanza L^2 tra profili. Ricorrendo ad un'approssimazione *bootstrap* si ottiene una valutazione dell'incertezza nel calcolo dell'indice stesso.



(a) *Metodo k-means* (b) *Metodo gerarchico*

Figura 3.9: Grafici dei profili ottenuti con il metodo multivariato

3.5.2 Approssimazioni via *bootstrap*

Nel seguito sarà applicata una tecnica di ricampionamento dei dati utilizzata nell'ambito delle simulazioni in inferenza statistica: il *bootstrap* parametrico. Questa procedura consiste nel simulare un elevato numero B di campioni dal modello statistico che ha come parametro la stima di massima verosimiglianza $\hat{\theta}^{\text{oss}}$ basata sui dati a disposizione.

La possibile strategia per applicare il *bootstrap* parametrico che porta a risultati distributivi asintotici è il *bootstrap* di massima verosimiglianza. Tale strategia assicura un risultato distributivo con accuratezza del primo ordine.

In alcune situazioni bisogna adottare un approccio diverso, ovvero il *bootstrap* non parametrico. A differenza del metodo parametrico, la distribuzione della statistica di interesse T viene ricavata tramite ricampionamento con reimmissione da un campione di osservazioni di cui non si conosce la distribuzione.

Si suppone di avere un insieme di N osservazioni e di voler determinare una statistica di interesse. Usualmente, per applicare questa procedura vengono generati B nuovi campioni, ciascuno di lunghezza N, dal campione osservato e si calcola il valore della statistica per ogni campione *bootstrap*. La distribuzione *bootstrap* fornisce una stima della vera distribuzione da cui provengono i dati.

In Figura 3.10 si riporta l'andamento dei primi 30 valori di *silhouette* ottenuti tramite *bootstrap* con $B = 500$ replicazioni per ciascuno dei quattro metodi considerati, mentre nei boxplot di Figura 3.11 si riassume la distribuzione delle statistiche *bootstrap*.

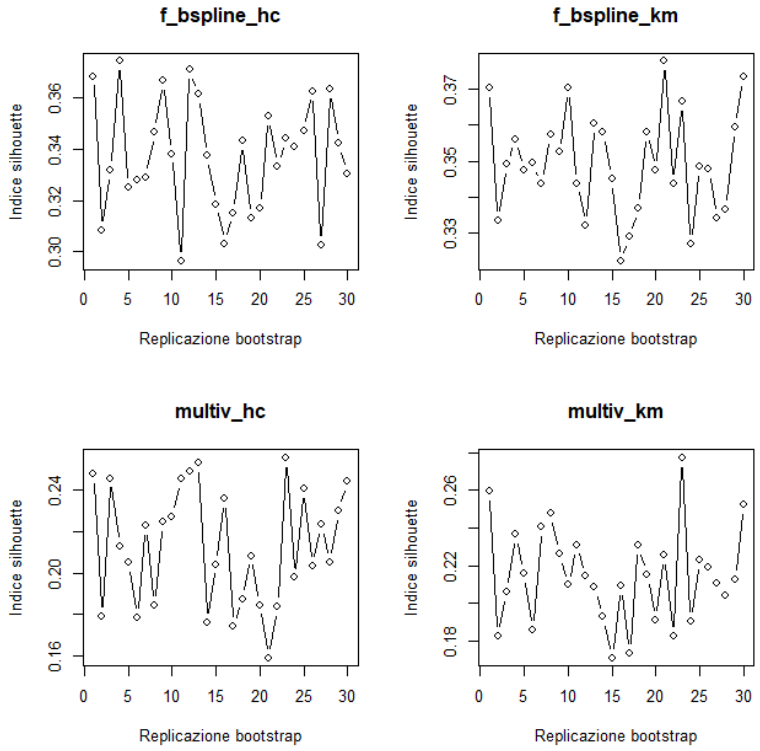


Figura 3.10: Indici di silhouette calcolati tramite *bootstrap* non parametrico con B=500

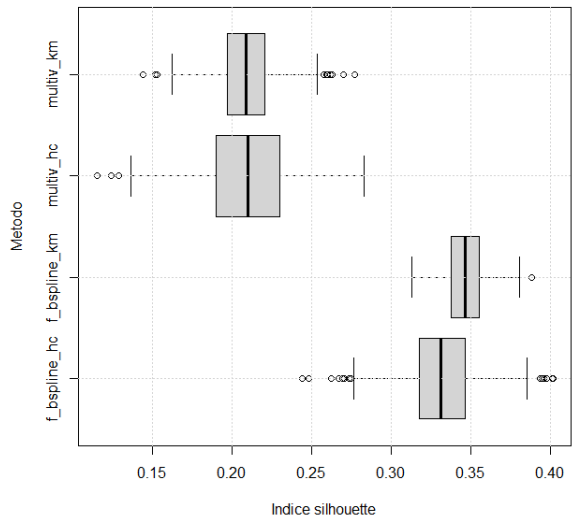


Figura 3.11: Boxplot degli indici di *silhouette* con $B = 500$ replicazioni per confrontare i metodi multivariati basati sulle caratteristiche e i metodi funzionali

Dal grafico emerge come i metodi per dati funzionali si comportino meglio degli analoghi metodi multivariati basati sulle caratteristiche in quanto la distribuzione dell'indice di *silhouette* è spostata verso valori più alti. L'ampiezza del boxplot indica la maggior variabilità del metodo multivariato. Il metodo *k-means* applicato alle B-*spline* mostra, invece, variabilità più contenuta ed indice di *silhouette* maggiore.

Si può affermare che tra i quattro metodi messi a confronto, quest'ultimo risulta essere il più valido e più adeguato per lo specifico studio sulle saldature a punti.

In letteratura si trovano pochi studi sui metodi di *clustering* funzionale applicati alle osservazioni DRC per acquisire informazioni tecniche sui processi RSW. L'efficacia dei metodi *k-means* e gerarchico è stata superiore quando applicata nel contesto funzionale rispetto all'alternativa multivariata poiché quest'ultima, basata su caratteristiche ben specifiche, tiene in considerazione solo una parte dell'informazione, mentre trattando i dati come funzionali si colgono tutti gli aspetti delle curve a disposizione. L'approssimazione via *bootstrap* ha permesso di comprendere meglio la differenza, in termini di prestazioni, tra i metodi adatti al contesto funzionale e quelli più semplici in cui non si tiene conto della natura funzionale del dato stesso.

Capitolo 4

Caso studio: sorveglianza delle condizioni operative delle navi e delle emissioni di CO_2

Scopo di questo capitolo è presentare delle sintesi legate alla sorveglianza delle carte di controllo tramite modellazione e analisi delle componenti principali. Le nuove tecnologie di acquisizione si basano su sensori multipli e ad alta frequenza, che permettono di rilevare grandi quantità di dati su molte variabili di processo e caratteristiche qualitative. Le tecniche proposte si concentrano sui residui di una regressione lineare con covariate funzionali usate come regressori e risposta scalare.

In questo caso studio, si applicano le carte di controllo ad un insieme di dati riguardanti le emissioni di CO_2 nel settore marittimo. Si analizzano le diverse caratteristiche di una nave che possono concorrere ad anomalie nelle emissioni di anidride carbonica durante i viaggi in mare. Per le analisi si considera la versione del *dataset* presente nella libreria `funcharts` (Capezza et al., 2023).

I codici utilizzati vengono riportati in Appendice A.2.

4.1 Descrizione del contesto

Le emissioni di CO_2 sono emissioni di anidride carbonica. Con l'alta produzione di gas, petrolio e carbone, entrano nella nostra aria pulita e creano uno strato invisibile intorno alla terra. Questo strato trattiene il calore all'interno della terra e questo processo, chiamato effetto serra, causa il riscaldamento globale. L'anidride carbonica è infatti un gas serra, che può rimanere nell'atmosfera per

migliaia di anni e nuocere gravemente alla natura e a tutti gli esseri viventi. Negli ultimi anni, il problema del sorveglianza di queste emissioni è diventato prioritario in vista dei cambiamenti climatici.

Nel settore del trasporto marittimo, il Comitato per la protezione dell'ambiente marino ha dato vita a programmi contro l'inquinamento atmosferico che richiedono la sorveglianza e la verifica delle emissione di CO_2. Le compagnie di navigazione stanno installando sulle loro navi dei sistemi multisensoriali che consentono di trasmettere e memorizzare enormi quantità di dati osservati.

La sorveglianza resta comunque una sfida in quanto le emissioni misurate dipendono anche da diversi altri fattori come ad esempio il tipo di nave, il pescaggio, la velocità, l'accelerazione, il vento, stato del mare, ecc.

L'obiettivo è la costruzione di modelli che riescano a prevedere le emissioni di CO_2 delle imbarcazioni sulla base dei dati osservati che descrivono le condizioni operative delle navi e ne permettono la sorveglianza per individuare anomalie e diagnosticare guasti.

4.2 Struttura dei dati

I dati `ShipNavigation.RData` presenti nel pacchetto `funcharts` (Capezza et al., 2023) si riferiscono a delle osservazioni raccolte da una nave da crociera, la Ro-Pax, di proprietà della compagnia di navigazione Gruppo Grimaldi. La nave è dotata di 2 set motori per la propulsione dotati di elica e un generatore ad albero per l'alimentazione elettrica. Per motivi di privacy nome della nave, itinerario e data del viaggio non vengono prese in considerazione in questo studio.

Il dataset originale, che comprendeva circa 260 mila osservazioni, è stato ridotto a 88271 perché si sono presi in considerazione solo i viaggi che partivano da Porto 3 verso Porto 1.
Le variabili presenti nel dataset erano 28, con la fase di *preprocessing* sono state ridotte sostituendo, ad esempio, quelle appartenenti alla stessa categoria con la loro media; altre variabili, reputate non necessarie, non sono state prese in esame. Le nove covariate funzionali ritenute utili per lo studio sono riportate di seguito con simbolo ed unità di misura:

- Velocità, $\mathbf{V}$, kn (nodi, miglia nautiche l'ora)
- Accelerazione, $\mathbf{A}$, NM/h^2

- Differenza di potenza tra l'albero dell'elica di sinistra e quello di dritta, **ΔP**, kW
- Distanza della rotta nominale, **Dist**, NM (Miglia nautiche)
- Componente longitudinale del vento, $\mathbf{W_L}$, kn
- Componente trasversale del vento, $\mathbf{W_T}$, kn
- Temperatura dell'aria media di quattro motori, **T**, ℃
- Tempo di navigazione cumulativo, **H**, h
- Assetto, **Trim**, m

Le componenti del vento sono calcolate in base a velocità del vento e direzione relativa alla nave, misurata in radianti ed indicata con Ψ. La componente W_L è ottenuta come $W \cos \Psi$, mentre W_T come $|W \sin \Psi|$.

Nel caso studio reale proposto, i dati funzionali sono ottenuti da profili raccolti durante la navigazione con una frequenza di 5 minuti dal sistema multisensore di bordo.

4.3 Analisi e interpretazione dei dati

Come anticipato nel Capitolo 2, il controllo statistico di processo multivariato ha come obiettivo il rilevamento, la localizzazione e la diagnosi di variazioni anomale del processo.

Il primo step consiste nella raccolta di dati storici e nella creazione degli insiemi di *training* e *test*, su cui si definiscono le statistiche T^2 e SPE. I dati di *test* sono proiettati sullo spazio definito dalle componenti principali del modello per autenticare la robustezza delle carte.

Nel secondo step si prevede la sorveglianza delle nuove osservazioni per mezzo delle carte di controllo calcolate allo step precedente.

Nell'ultimo step, per ogni osservazione non a norma, si calcola il contributo di ciascuna variabile del processo alle statistiche T^2 e SPE per determinare quali variabili e quali istanti temporali siano responsabili della variazione di entità maggiore del normale.

4.3.1 Preprocessing e registrazione

Il primo passo da compiere è ottenere osservazioni omogenee $\tilde{X}_i = (\tilde{X}_{i1}, \ldots, \tilde{X}_{iP})$ delle covariate funzionali $\tilde{X}$ ad ogni viaggio $i = 1, \ldots, N$. Per ogni $i = 1, \ldots, N$ e $p = 1, \ldots, P$, $\tilde{X}_{ip}$ può essere ottenuto dai dati discreti x_{ipN} per $N = 1, \ldots, m_i$, utilizzando una base B-*spline* cubica con nodi equidistanti

$$\tilde{X}_{ip}(t) = \sum_{q=1}^{Q} c_{iqp}\phi_q(t), \quad i = 1, \ldots, N, \quad p = 1, \ldots, P, \quad t \in \mathcal{T},$$

dove $\phi_1, \ldots, \phi_Q$ sono le funzioni base B-*spline* e c_{iqp} sono i coefficienti associati alle funzioni di base. I dati funzionali sono stati ottenuti lisciando i dati con la regolarizzazione, utilizzando il pacchetto `fda`. Poiché il numero di funzioni base deve essere abbastanza grande da garantire che la regolarizzazione sia controllata dalla scelta del parametro di lisciamento, si sono usate 100 basi con nodi equidistanti ed è stata scelta una penalità di rugosità sulla derivata seconda integrata al quadrato. Per ogni variabile funzionale e per ogni osservazione, i parametri di lisciamento vengono scelti separatamente minimizzando il criterio di convalida incrociata generalizzata.

Anche se il tempo è naturalmente incline a essere scelto come dominio funzionale, il tempo totale di viaggio potrebbe variare da viaggio a viaggio. Pertanto, una scelta ragionevole è quella di utilizzare la frazione di distanza percorsa nel corso del viaggio come dominio comune $\mathcal{T} = [0, 1]$ dei dati funzionali. Questa scelta può essere considerata come una registrazione di riferimento del set di dati funzionali dal dominio temporale specifico per ogni funzione al dominio comune $[0, 1]$ con il gruppo delle trasformazioni affini con pendenza positiva come gruppo di funzioni di deformazione e i punti di partenza e di arrivo del viaggio come punti di riferimento.

4.3.2 Stima del modello e sorveglianza prospettica

Il metodo più diffuso per la selezione delle componenti principali consiste nel tracciare gli autovalori in ordine decrescente di grandezza e successivamente cercare nel grafico quello che viene definito "gomito", ossia una brusca variazione di pendenza che permette di individuare qual è il valore da scegliere per il numero di componenti principali. La percentuale di varianza spiegata da ogni componente

principale è legata al corrispondente autovalore tramite la formula

$$\% \ var_i^{sp} = \frac{\lambda_i}{\sum_{j=1}^{L} \lambda_j},$$

dove λ_i corrisponde all'autovalore legato alla i-esima componente, mentre L indica il numero massimo di componenti.

La scelta dell'insieme di componenti principali da tenere nel modello viene attuata considerando sia la variabilità spiegata delle covariate che la statistica

$$PRESS = \sum_{i=1}^{N} (y_i - \hat{y}_{[i]})^2.$$

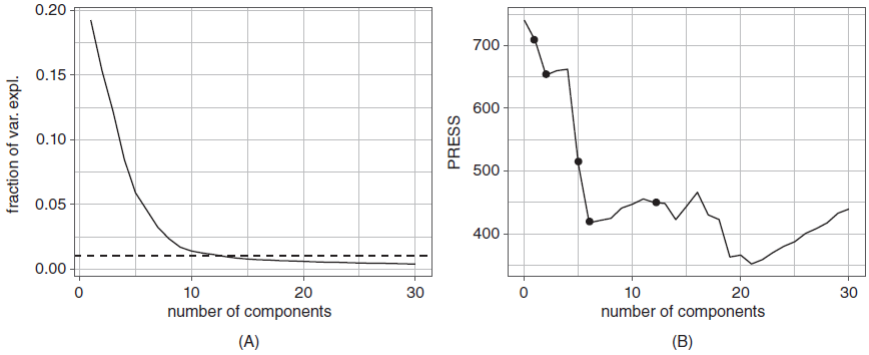


Figura 4.1: (A) Frazione di varianza delle variabili funzionali spiegata da componenti principali funzionali con soglia 0.01. (B) Statistica PRESS in funzione delle prime l componenti principali

L'insieme delle componenti principali che ottiene la riduzione di statistica PRESS più alta $\mathcal{M}$ risulta $\{1, 2, 5, 6, 12\}$. In Figura 4.1 sono riportate la frazione di varianza spiegata e il grafico delle prime l componenti principali. L'assunzione di normalità per la componente di errore è supportata dal test di Shapiro-Wilk.

Per controllare il FWER, come spiegato al paragrafo 2.4, si applica la correzione di Bonferroni con contributi delle carte separati in base alla categoria di variabile da sorvegliare.

Ora si procede con la Fase II, riguardante la sorveglianza di covariate funzionali e risposta scalare, ossia la CO_2 per miglio.

Nella prima parte delle analisi per stimare i limiti di controllo per la sorveglianza si utilizzano $n = 139$ osservazioni appartenenti a 30 viaggi consecutivi.

Di seguito vengono presentate le tre carte di controllo già trattate al paragrafo 2.4. I punti indicano i valori delle rispettive statistiche di sorveglianza ad ogni viaggio, mentre i limiti di controllo sono impostati come quantile di ordine $(1-\alpha)$

delle singole statistiche, con contributi associati alle carte come specificato alle ultime righe del paragrafo 2.4.

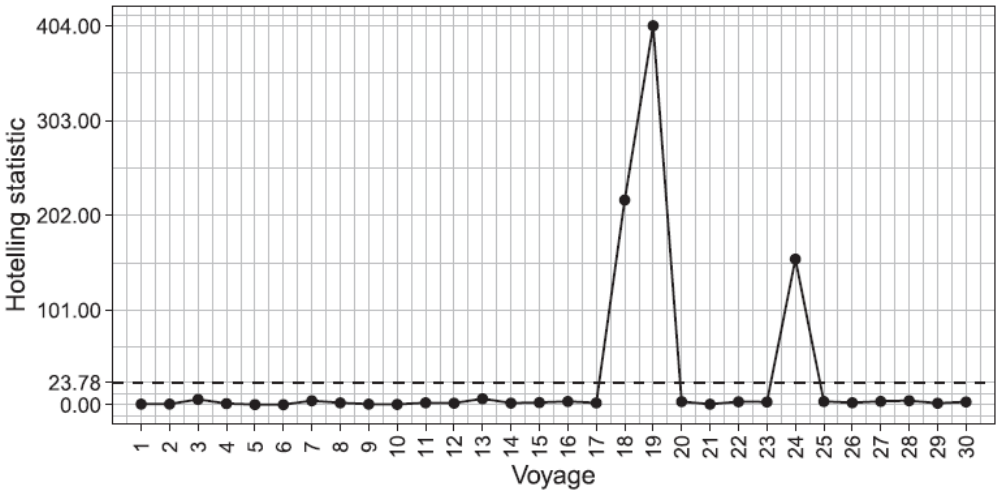


Figura 4.2: Carta T^2 di Hotelling per sorveglianza di Fase II

Come si può notare da Figura 4.2, la carta segnala tre viaggi fuori controllo, nello specifico il 19 con valore della statistica di controllo molto elevato, il 18 e il 24 con valori accettabili, sebbene OC.

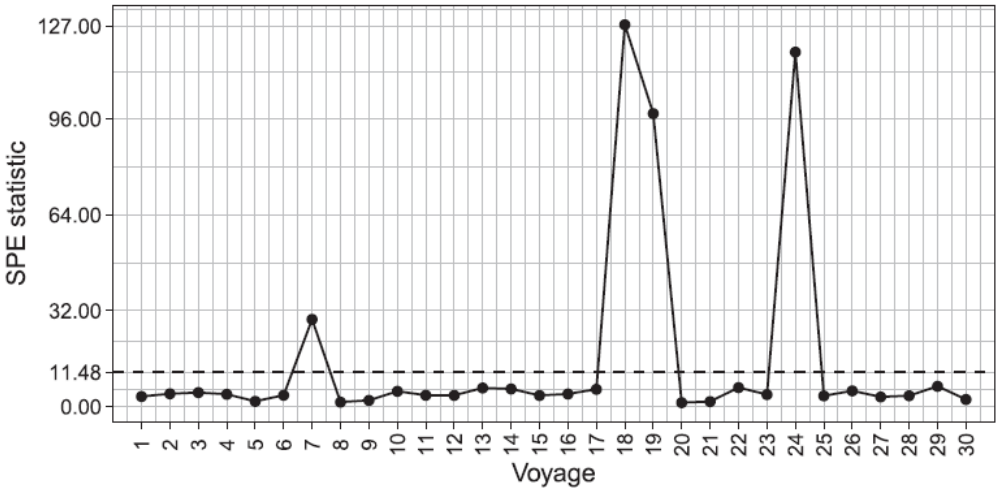


Figura 4.3: Carta SPE per sorveglianza di Fase II

La carta SPE segnala un allarme di lieve entità per il viaggio 7, mentre per i viaggi 18, 19 e 24 si osserva un valore molto più alto. Questi valori risultano comunque inferiori agli analoghi della carta T^2.

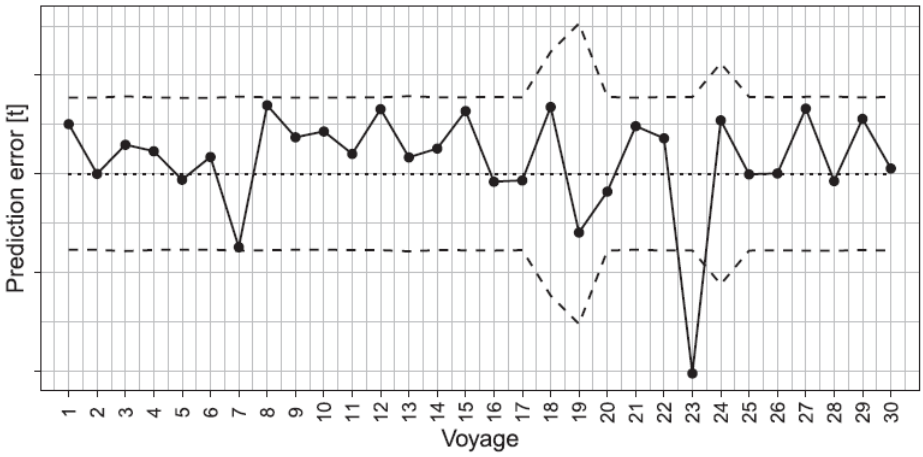


Figura 4.4: Carta RPE per sorveglianza di Fase II

La carta di controllo RPE identifica un solo viaggio con valore anomalo. Il viaggio 23, infatti, mostra covariate IC ma indica che le emissioni totali di CO_2 sono inferiori al valore previsto. Potrebbe dipendere da fattori esterni, come per esempio altre variabili non prese in considerazione per questa analisi.

Diagnosi dei guasti attraverso i diagrammi di contributo

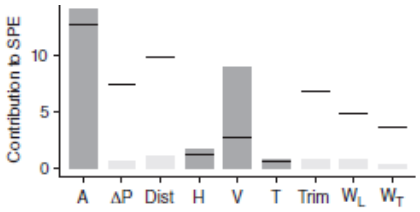
La diagnosi dei guasti, quando viene rilevata una situazione fuori controllo, ha lo scopo di mettere in evidenza le variabili più influenti, ovvero quelle la cui variazione contribuisce maggiormente all'aumento del valore osservato della statistica di controllo considerata. Il comportamento di una nuova osservazione viene, quindi, studiato ricorrendo all'analisi dei contributi di ogni variabile funzionale per ogni carta di controllo analizzata (formule 2.12 e 2.13).

Si osservi che per quanto riguarda la carta RPE, gli eventuali segnali fuori controllo che provengono da essa devono essere esplorati andando a cercare le covariate funzionali non incluse nel modello.

Poiché i contributi non hanno la stessa distribuzione per ogni variabile a disposizione, deve essere scelto un limite di controllo superiore appropriato per ogni contributo. Di solito lo si stima partendo dalla distribuzione empirica basata sul campione di Fase I.

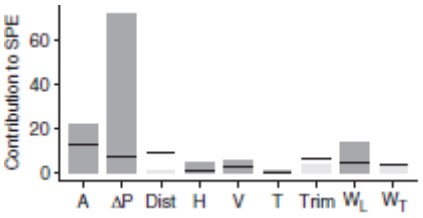
La Figura 4.5 mostra i contributi di ciascuna variabile funzionale alle statistiche di controllo. Le linee nere rappresentano i limiti calcolati sulla base dei viaggi di riferimento, le barre più scure i contributi superiori al limite.

Viaggio 7

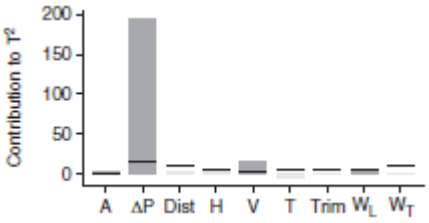


(a)

Viaggio 18

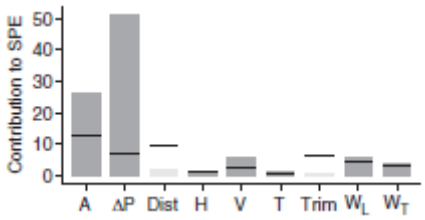


(b)

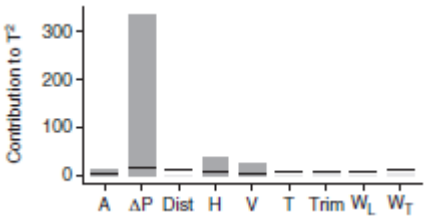


(c)

Viaggio 19

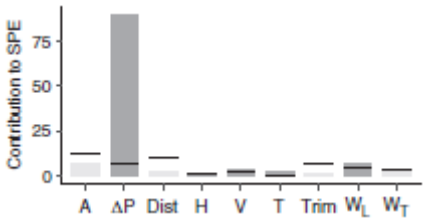


(d)

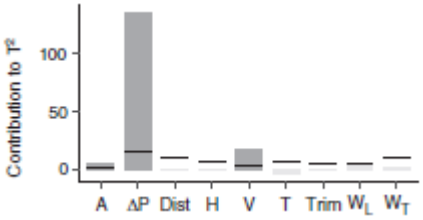


(e)

Viaggio 24



(f)



(g)

Figura 4.5: Contributi delle covariate funzionali alle statistiche T^2 di Hotelling e SPE per i viaggi risultati fuori controllo

Per il proseguimento delle analisi si scelgono come riferimento il viaggio 7, caratterizzato da un valore fuori controllo osservato solo per la statistica SPE, e i viaggi 18 e 19 segnalati fuori controllo da entrambe le carte T^2 e SPE.

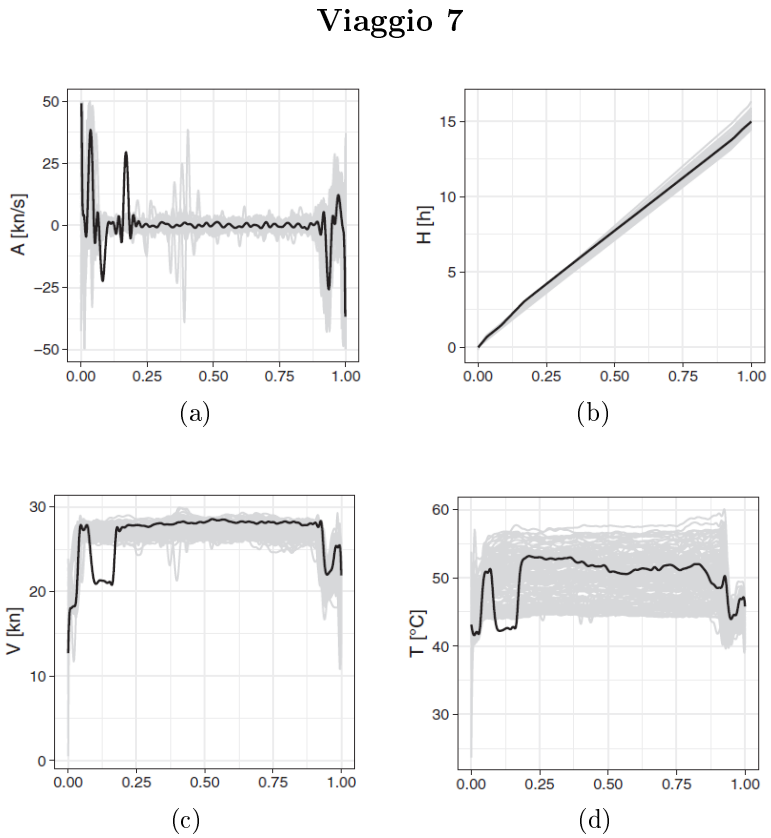


Figura 4.6: Grafici delle covariate funzionali indicate come responsabili di OC per il viaggio 7. Sull'asse delle ascisse sono riportati i valori di frazione di distanza percorsa. La zona grigia rappresenta i profili di riferimento, la linea nera l'osservazione corrente

Il **viaggio 7** è il primo ad essere segnalato fuori controllo. Dalla Figura 4.5(a) si può osservare che le variabili maggiormente responsabili della segnalazione sono l'accelerazione (A) e la velocità (V). I grafici delle traiettorie di Figura 4.6 mostrano come la nave stesse viaggiando a velocità bassa nella prima parte del tragitto per poi normalizzarsi nel tratto intermedio. L'accelerazione aumenta significativamente nel primo tratto per permettere alla nave di raggiungere una velocità di navigazione adeguata e non accumulare ritardi nel tempo di arrivo. La temperatura media dell'aria dei motori ha un comportamento molto simile a quello della velocità.

Viaggio 18

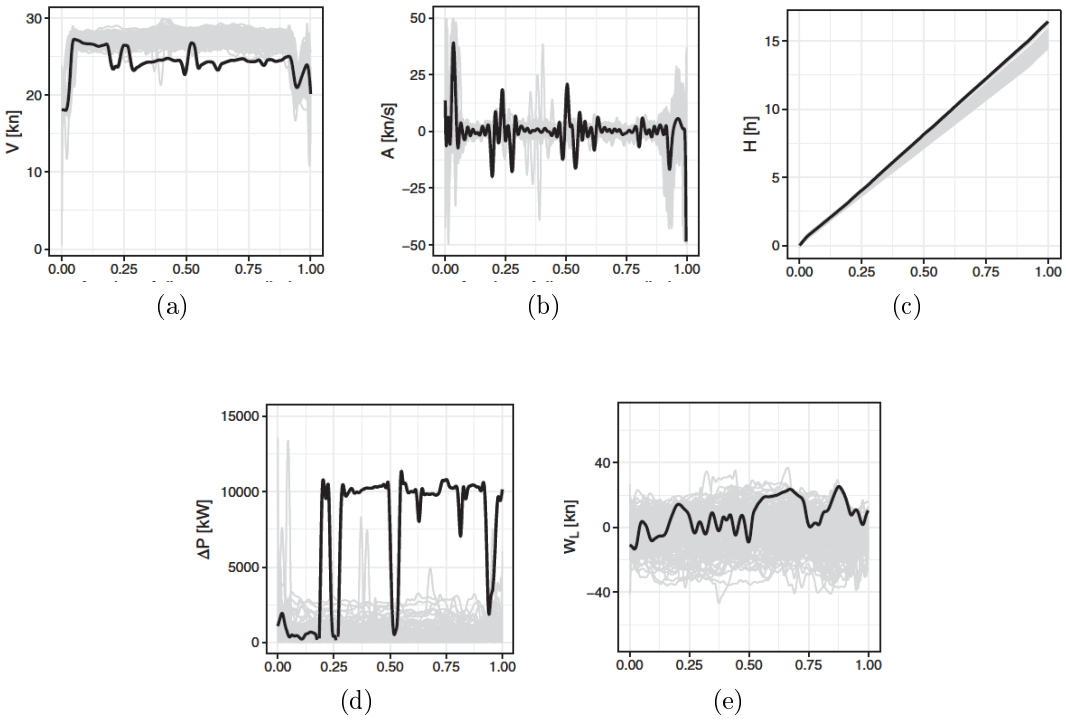


Figura 4.7: Grafici delle covariate funzionali indicate come responsabili di OC per il viaggio 18. Sull'asse delle ascisse sono riportati i valori di frazione di distanza percorsa. La zona grigia rappresenta i profili di riferimento, la linea nera l'osservazione corrente

Il **viaggio 18** è segnalato fuori controllo da entrambe le carte T^2 ed SPE. Dalla Figura 4.5(b) si può osservare che le variabili responsabili di contributi elevati alla carta T^2 sono la velocità (V) e la differenza di potenza tra gli assi dell'elica (ΔP). I contributi rilevanti alla statistica SPE sono gli stessi evidenziati per la statistica T^2, con l'aggiunta della componente longitudinale del vento (W_L), del tempo di navigazione cumulativo (H) e dell'accelerazione (A). Dal grafico dei profili in Figura 4.7 si può osservare un andamento altalenante della velocità e un alternarsi di accelerazioni e decelerazioni che influiscono molto sul tempo di navigazione, il quale mostra un forte ritardo al momento dell'entrata in porto. Osservando il profilo della differenza di potenza tra gli assi delle eliche (ΔP) si nota che la nave ha avuto problemi ad uno dei motori per la maggior parte della durata del viaggio, fenomeno aggravato dalla componente longitudinale del vento.

Viaggio 19

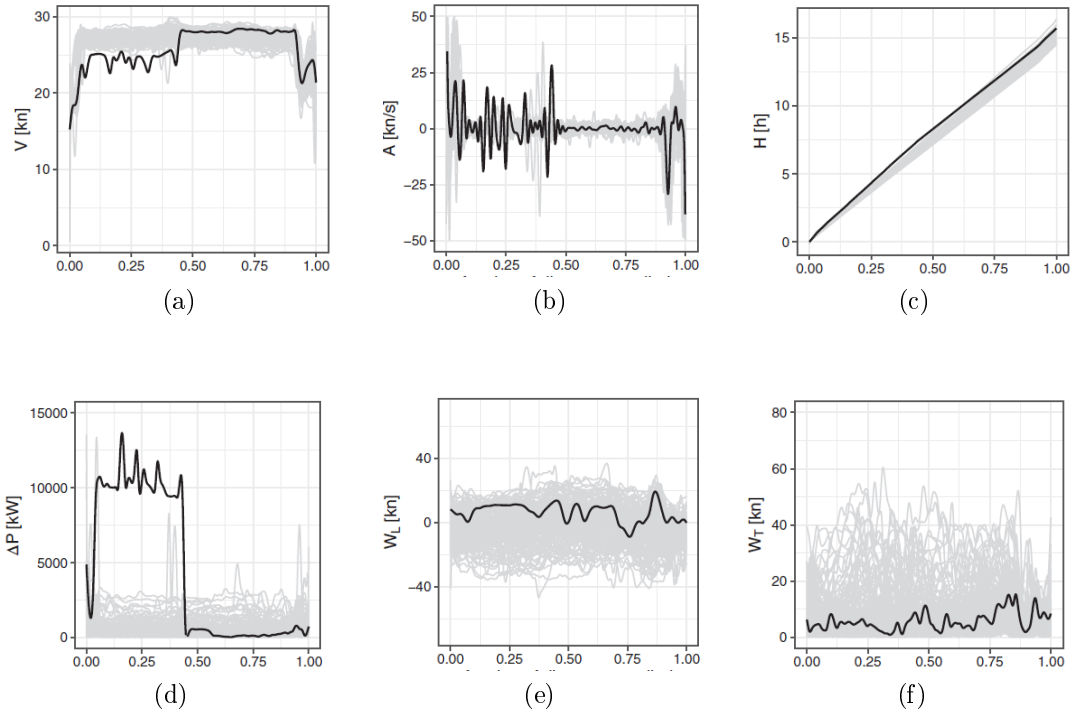


Figura 4.8: Grafici delle covariate funzionali indicate come responsabili di OC per il viaggio 19. Sull'asse delle ascisse sono riportati i valori di frazione di distanza percorsa. La zona grigia rappresenta i profili di riferimento, la linea nera l'osservazione corrente

Il **viaggio 19** è segnalato da entrambe le carte di controllo funzionali, proprio come il viaggio 18. Il valore alto per la statistica T^2 comporta limiti di controllo più larghi per la previsione sulla risposta. I diagrammi di contributo relativi alla statistica T^2 identificano l'accelerazione (A), la differenza di potenza tra le eliche (ΔP), il tempo cumulativo (H) e la velocità (V) come responsabili dell'allarme segnalato. La statistica SPE riporta in più le componenti del vento longitudinale (W_L) e trasversale (W_T), poco al di sopra del corrispondente limite di controllo. Dal grafico di Figura 4.8(a) si osserva un profilo della velocità con valori bassi durante la prima parte del viaggio, che si riflettono su un tempo di navigazione più alto del normale a metà tragitto. Questo ha portato ad un arrivo nei tempi previsti. La differenza di potenza tra le eliche riportata in Figura 4.8(d) ha permesso di scovare un malfunzionamento in uno dei due motori ad inizio viaggio.

4.4 Carte di controllo per dati di Fase II

Fino ad ora, la strategia di sorveglianza presentata al paragrafo 2.4 presupponeva che tutte le covariate funzionali fossero completamente osservate sul loro dominio compatto $\mathcal{T}$. Si vuole, ora, estendere la sorveglianza anche a punti interni non osservati. Scopo dell'analisi è dimostrare l'uso diretto e in tempo reale della strategia di sorveglianza proposta per supportare una diagnosi tempestiva dei guasti durante la navigazione. Si definisce, pertanto, una funzione di deformazione in tempo reale, una versione funzionale delle carte di controllo che sfrutta le informazioni accumulate fino al punto corrente di interesse. Si indichino con t^* l'istante al quale si vuole sorvegliare la procedura e $k^* \in \mathcal{T} = [0, 1]$ la frazione di distanza percorsa. La variabile risposta da sorvegliare y^* rappresenta il totale delle emissioni di CO_2 fino al tempo t^*. Alla fine del viaggio, y^* coincide con y^{new}.

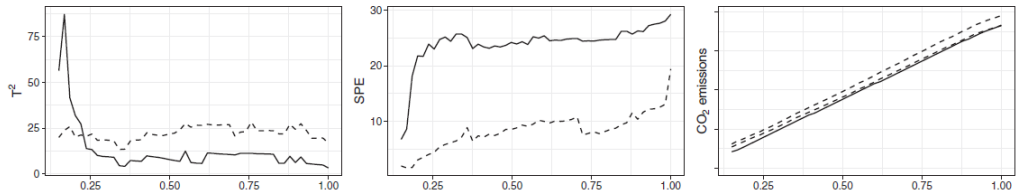
Ad ogni istante t^* si usa come funzione di deformazione in tempo reale la funzione $f : [0, t^*] \to [0, k^*]$ che associa ad ogni $t \in [0, t^*]$ la frazione di distanza percorsa

$$k(t) = d(t)/d,$$

dove $d(t)$ rappresenta la distanza percorsa fino al tempo $t \leq t^*$ e d la distanza totale alla fine del viaggio. Poiché d non è nota al tempo t^* sono necessari degli accorgimenti per determinare $k(t)$.
In primo luogo si considera la posizione corrente della nave P^*, ottenuta tramite latitudine e longitudine. Si identifica, quindi, il punto $\bar{P}^*$ sulla rotta percorsa dalla nave con minima distanza dalla posizione corrente all'istante di riferimento t^*. Infine si calcola la frazione di distanza percorsa a t^* come $k(t^*) = d^*/d$, dove d^* indica la lunghezza della tratta dal porto di partenza al punto P^*.

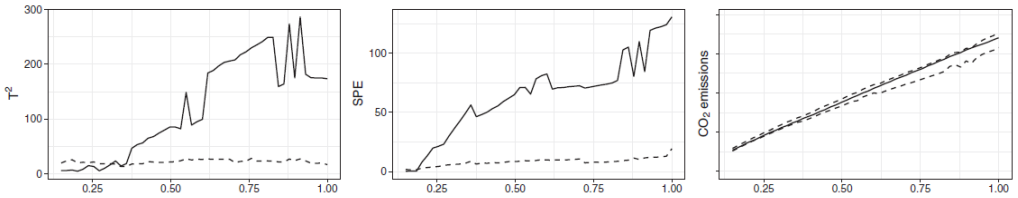
Il dataset di riferimento all'istante t^* può essere ricavato troncando le covariate funzionali a k^*, definendo il nuovo dominio $[0, k^*]$. I dati così ottenuti possono essere usati per ripetere le analisi di Fase I, descritte al paragrafo 2.3, per calcolare le statistiche di sorveglianza T^2 e SPE a k^* e i limiti delle carte di controllo corrispondenti, calcolati una sola volta per ogni valore di k^*. Questo perché in Fase II i limiti di queste due carte non cambiano. Al contrario, i limiti della carta RPE dipendono anche dai profili attuali delle covariate e devono essere determinati in tempo reale per ogni viaggio durante la Fase II.

Viaggio 7



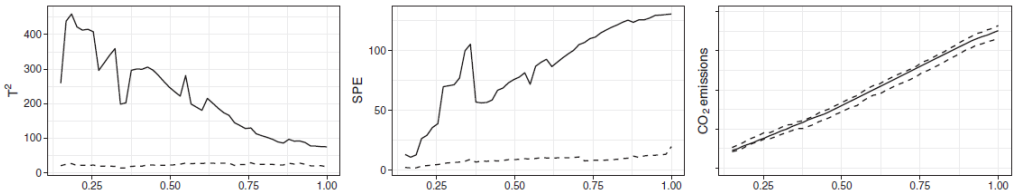
(a)

Viaggio 18



(b)

Viaggio 19



(c)

Figura 4.9: sorveglianza in tempo reale dei viaggi 7 (a), 18 (b) e 19 (c). Grafici delle carte di controllo T^2, SPE ed RPE. La linea solida indica i profili delle statistiche di sorveglianza, le linee tratteggiate i corrispondenti limiti di controllo. Sull'asse delle ascisse sono riportati i valori di frazione di distanza percorsa.

In Figura 4.9 si riportano le carte di controllo funzionali in tempo reale realizzate per i tre viaggi segnalati come OC, ossia i viaggi 7, 18 e 19. I comportamenti delle tre carte vengono successivamente confrontati con i profili delle covariate critiche riportate alla Sezione precedente.

Prendendo in considerazione il viaggio 7, si osserva come il profilo della statistica SPE sia fuori dal limite di controllo su tutto il dominio. Coerentemente con i grafici che riportano le covariate critiche (Figura 4.6), la carta di controllo funzionale T^2 identifica anche valori anomali ad inizio viaggio. Per il viaggio 18 entrambe le carte T^2 e SPE segnalano una situazione fuori controllo per la maggior parte della tratta.

La necessità di gestire dati complessi creati tramite l'implementazione di un moderno sistema di supervisione richiede delle procedure statistiche sempre più

sofisticate. A detta degli autori, le analisi affrontate finora sono basate su nuove tecniche di sorveglianza. Le tre fasi di sorveglianza hanno richiesto un dominio comune su cui operare, pertanto si è resa necessaria una registrazione dei punti di riferimento.

L'utilizzo simultaneo delle carte di controllo T^2 ed SPE ha permesso di evidenziare i comportamenti insoliti rispetto ai viaggi passati e compiere un'adeguata diagnosi dei guasti. Si è considerato anche uno scenario con dati ottenuti in tempo reale, che ha mostrato la validità delle procedure sopra esposte e la loro applicabilità anche in contesti dinamici, per facilitare i processi decisionali.

4.5 Carta di controllo per regressione con risposta e covariate funzionali

In questa sezione si vuole studiare l'efficienza delle carte di controllo in grado di tener conto di più covariate funzionali che possono influenzare la caratteristica di qualità di interesse, trattata come dato funzionale.
Lo scopo è ottenere carte di controllo in grado di emettere un allarme quando il processo è OC e dimostrare la loro efficacia.

Si prendono ad esempio i dati relativi ad osservazioni che si presume siano rappresentative delle prestazioni del processo IC.

Di seguito si provano ad analizzare le emissioni di CO_2 dopo l'Iniziativa di Efficienza Energetica (EEI) eseguita su una nave Ro-Pax. Si può pensare che le migliorie apportate possano produrre uno spostamento medio delle emissioni stesse. Si focalizza l'analisi sulla sorveglianza di Fase II per ogni viaggio della nave su una rotta specifica. Si sorvegliano le emissioni di CO_2, dopo l'efficientamento energetico, durante la navigazione tra Porto 3 e Porto 1. L'analisi viene condotta attraverso profili delle emissioni di CO_2 per miglio nautico (identificati dalla variabile `CO2pm`), aggiustati dall'influenza di quattro covariate funzionali, ossia la velocità della nave (V), l'assetto (Trim), il vento longitudinale (W_L) e il vento trasversale (W_T).

I dati vengono suddivisi in due gruppi: al primo gruppo vengono associati i dati di Fase I con 161 viaggi avvenuti nell'ultimo anno e antecedenti all'EEI. Nel secondo i dati di Fase II corrispondenti a 174 viaggi, avvenuti sempre sulla stessa tratta ma dopo l'avvio dell'efficientamento energetico.
La caratteristica di qualità `CO2pm` è una variabile funzionale. Per le analisi si applicano pertanto le carte di controllo basate sulla risposta funzionale in presenza di covariate multivariate presentate alla Sezione 2.6.

Il codice di *preprocessing* fornito in Appendice A.2 carica i dati e restituisce gli oggetti y_1, x_1, y_2, x_2 che contengono le osservazioni della risposta funzionale e delle covariate suddivise in Fase I e Fase II, rispettivamente. Successivamente rimuove eventuali *outlier* e divide le osservazioni IC in y_1 e x_1 per ottenere osservazioni di *training* e *tuning* della risposta e delle covariate. A questo punto si può adattare la carta di controllo per risposta e covariate funzionali. Si fissa un tasso di errore di tipo I pari a 0.01 e si procede con la creazione delle carte di controllo T^2 di Hotelling e SPE.

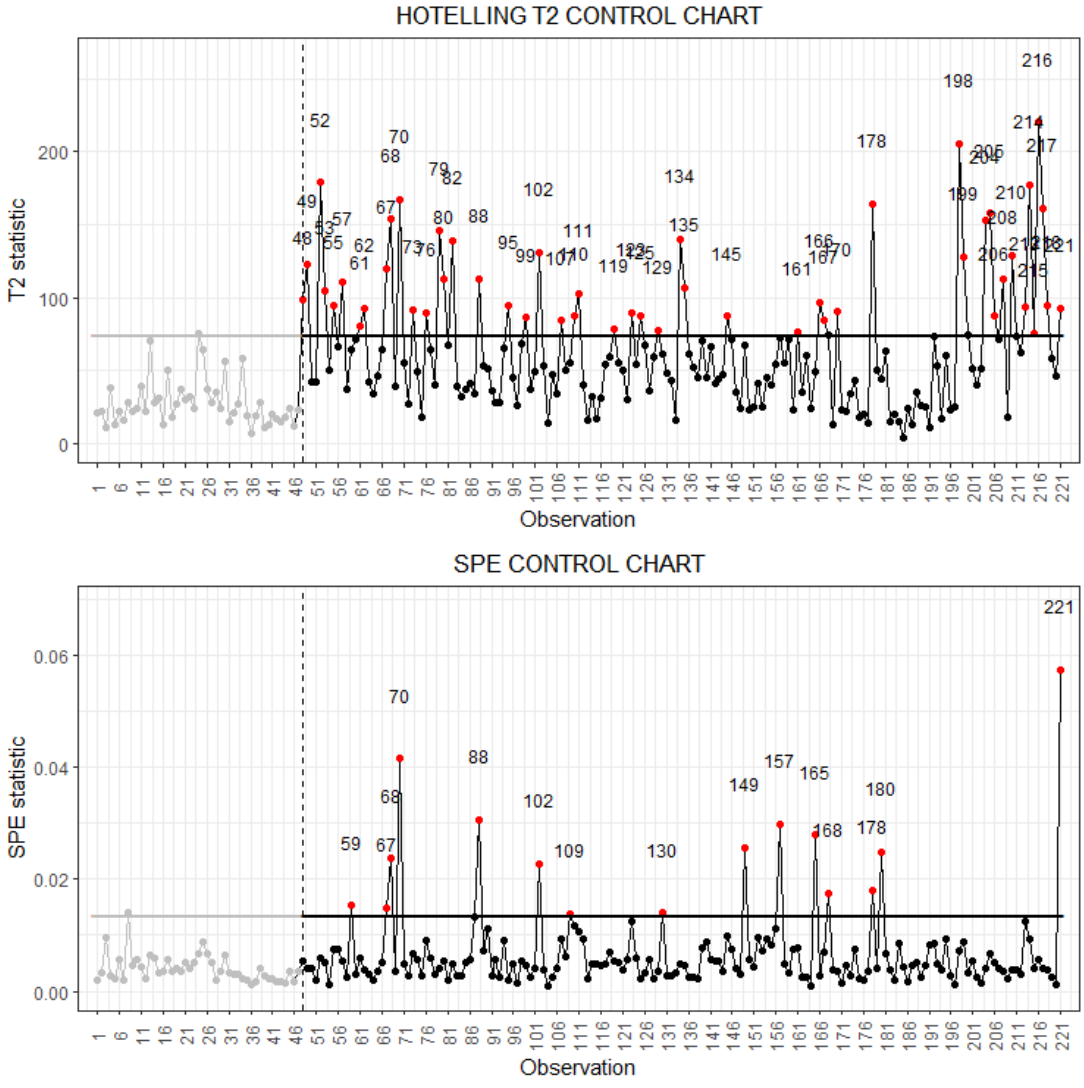
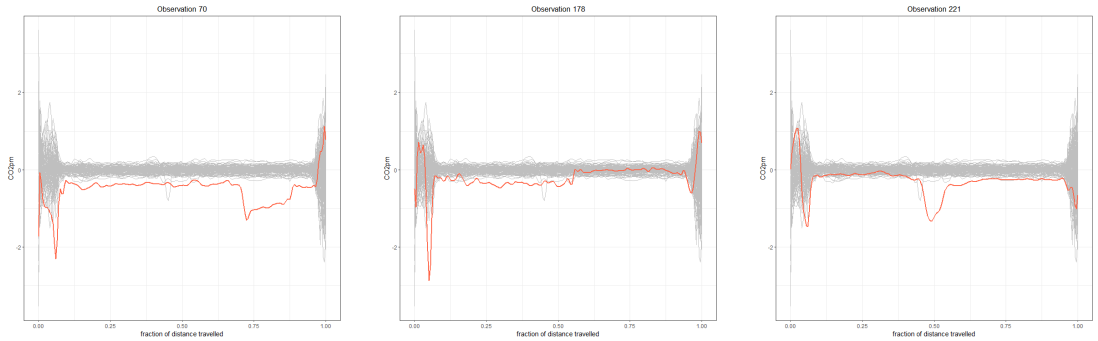


Figura 4.10: Carta di controllo funzionale. La linea verticale separa le osservazioni di tuning dai dati di Fase II

Nella Figura 4.10 le statistiche di sorveglianza che corrispondono alle osservazioni di Fase I prima dell'EEI sono riportate sul lato sinistro della linea verti-

cale tratteggiata, mentre a destra in nero si riportano le osservazioni di Fase II, ottenute dopo l'EEI. In rosso vengono segnalate le osservazioni fuori controllo. Poiché l'obiettivo è dimostrare l'efficacia e la sensibilità delle carte funzionali, si scelgono solo alcuni profili OC rappresentativi. Per le successive analisi si sono scelte nello specifico le osservazioni 70, 178 e 221.

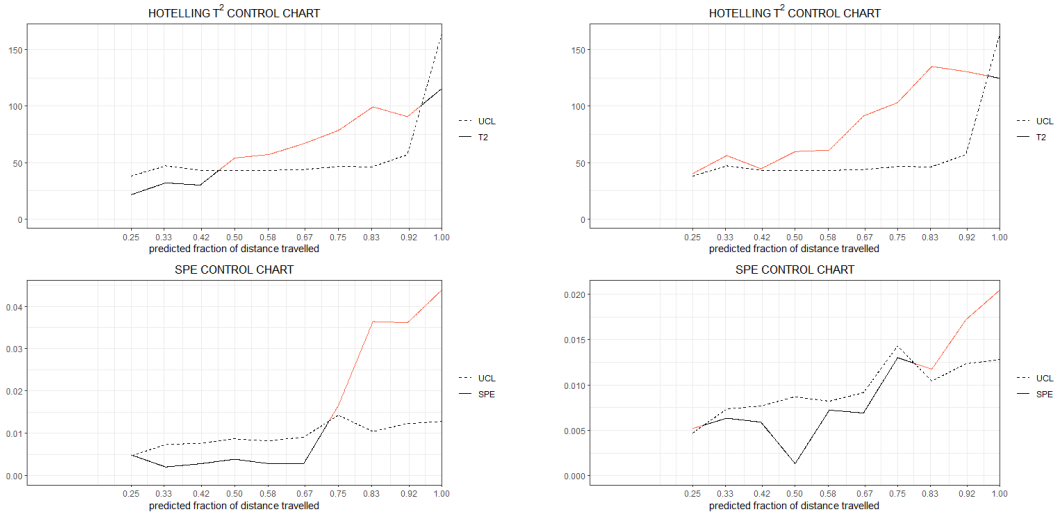


(a) (b) (c)

Figura 4.11: Profili OC dei residui studentizzati corrispondenti alle osservazioni 70 (a), 178 (b) e 221 (c). I grafici raffigurano in grigio i residui studentizzati calcolati sull'insieme di dati di training. La linea rossa indica l'andamento dei profili fuori controllo. Sull'asse delle ascisse sono riportati i valori di frazione di distanza percorsa.

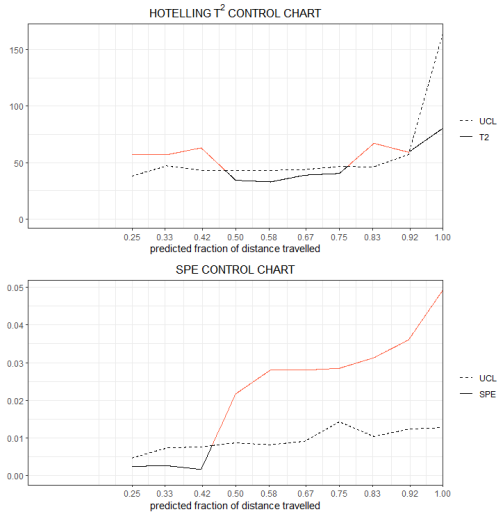
Il profilo dei residui corrispondente all'osservazione 70 risulta negativo per la maggior parte del tempo, con un residuo negativo rilevante per le emissioni di CO_2 nella prima parte del percorso e verso la fine, prima di diventare positivo. Nella prima parte del tragitto relativa all'osservazione 178 si osserva un residuo prima positivo, poi negativo nella prima parte del viaggio. Per quanto riguarda, invece, le emissioni legate all'osservazione 221 si osserva un residuo negativo nella parte centrale del viaggio.

Nella Figura 4.12(a) possiamo notare come le carte di controllo siano alte, la T^2 specialmente nella parte centrale del dominio, la SPE nella parte finale. Questo rispecchia il comportamento messo in evidenza nei grafici di Figura 4.11(a), dove si possono osservare dei residui grandi sia nella parte iniziale che finale della tratta. La Figura 4.12(b) mostra che la carta T^2 di Hotelling è sempre al di sopra del corrispondente limite di controllo. Per quanto riguarda l'osservazione 221, si osserva una statistica SPE molto grande nella parte centrale del dominio. Ciò evidenzia che in quella parte del viaggio, tenendo conto dei valori delle covariate funzionali, il profilo delle emissioni di CO_2 per miglio nautico è più basso di quanto ci si aspetti.



(a)

(b)



(c)

Figura 4.12: Carta di controllo in tempo reale per sorvegliare le osservazioni 70 (a), 178 (b) e 221 (c) di Fase II. Le linee solide mostrano i profili delle statistiche di sorveglianza (T^2 ed SPE), in rosso se sopra il corrispondente limite di controllo superiore, in nero se sotto

4.6 Considerazioni finali

Le carte proposte possono essere progettate, in modo relativamente semplice, e offrono prestazioni soddisfacenti, anche in presenza di un grande numero di osservazioni. Nel caso studio specifico si sceglie di sorvegliare il controllo delle emissioni di CO_2 durante la navigazione, in modo da risultare conformi alle normative UE per contrastare i cambiamenti climatici. L'analisi delle osservazioni in tempo reale ha permesso di mettere in evidenza i punti critici che potrebbero causare variazioni di emissione di anidride carbonica non a norma.

Quando la carta segnala un problema, gli operatori potranno intervenire tempestivamente per indagarne le cause e procedere alla loro risoluzione.

Conclusioni

Il tema della sorveglianza per la stabilità delle performance dei processi è stato oggetto, nel corso degli anni, di un crescente interesse in ambito industriale e non. Il cambiamento delle richieste di mercato ha reso necessaria l'introduzione di metodologie statistiche in grado di sorvegliare e analizzare la grande mole di dati provenienti dalle moderne tecnologie utilizzate nei diversi settori produttivi e dei servizi.

L'obiettivo principale di questa tesi è stato quello di approfondire alcuni metodi sviluppati per la sorveglianza in tempo reale delle dinamiche di dati funzionali. In particolare, l'elaborato ha presentato alcune procedure di sorveglianza di modelli di regressione con risposta scalare o funzionale e covariate funzionali. Alcune tecniche, attraverso algoritmi di regressione e *clustering*, possono essere utilizzate in questi contesti. Nello specifico, i metodi presi in considerazione sono stati quelli del *clustering* funzionale combinato all'applicazione di carte di controllo. Tali metodi sono stati applicati a due casi studio. L'elaborazione e l'analisi dei dati disponibili è stata complessa vista la struttura e la loro numerosità.

Nella prima parte del lavoro è stato introdotto e adattato il *clustering* funzionale; in particolare sono stati adattati i metodi *k-means*, gerarchico e gli approcci *filtering B-spline*, per i dati grezzi, basato sulla distanza, adattivo tramite `funHDDC` e adattivo tramite `curvclust`, per studiare le curve e interpretare le eventuali anomalie nel processo. Sono stati effettuati degli studi di simulazione applicati anche a metodi tradizionali basati su caratteristiche multivariate e sono stati confrontati i risultati tra i diversi metodi attraverso il calcolo dell'indice di *silhouette* e l'adattamento via *bootstrap* non parametrico. I risultati ottenuti mostrano l'efficacia dei metodi *k-means* e gerarchico quando applicati nel contesto funzionale rispetto all'alternativa multivariata poiché quest'ultima, basata su caratteristiche ben specifiche, tiene in considerazione solo una parte dell'informazione. Trattando, invece, i dati come funzionali si possono cogliere tutti gli aspetti delle curve a disposizione.

Si sarebbe potuto ridurre il notevole costo computazionale delle analisi im-

plementate ricorrendo a metodi di semplificazione dei calcoli per l'esecuzione di alcune di esse.

Una volta verificata l'efficacia e la potenza dei metodi illustrati durante la Fase I, è stata studiata la sensibilità di tali metodi in Fase II. Nel primo caso preso in esame, le analisi del *clustering* funzionale hanno permesso di individuare i gruppi di curve nella saldatura a punti RSW non in linea con le caratteristiche standard, che fanno supporre inefficienze dell'impianto o degli strumenti utilizzati.

Nel secondo caso, invece, come anticipato nell'ultima parte della tesi, sono state implementate delle carte di controllo di Fase II per modelli di regressione con risposta scalare o funzionale su un insieme di dati funzionali. Queste carte hanno permesso di mettere in evidenza le osservazioni anomale, le cui caratteristiche potrebbero causare aumento delle emissioni di CO_2.

I vantaggi delle procedure di sorveglianza descritte sembrano crescere in presenza di elevata dinamicità e confermano l'importanza e la potenzialità di queste tecniche statistiche in molti ambiti applicativi.

Questo studio si colloca in un contesto in cui le carte di controllo sono già usate regolarmente perché hanno un campo di applicazione piuttosto vasto. Per il futuro, ci si aspetta un ulteriore sviluppo di queste procedure e un più ampio utilizzo in molti altri settori.

Ringraziamenti

Il mio primo ringraziamento va alla mia relatrice Professoressa Giovanna Capizzi. La sua persona è stata fondamentale per la scelta dell'argomento da trattare e la realizzazione dell'elaborato.

Non meno importante è stato l'aiuto che ho ricevuto dai Professori Capezza, Lepore e Palumbo del Dipartimento di Ingegneria Industriale dell'Università Federico II di Napoli, autori degli articoli esaminati. La loro precisione nelle analisi e i loro suggerimenti hanno reso possibile lo sviluppo di una parte del mio progetto.

Ringrazio il Dott. Corrado Scalco, titolare della ditta Deltaline s.r.l. e mio zio, per avermi permesso di utilizzare computer con maggiori risorse computazionali grazie ai quali sono stati elaborati gli studi di simulazione.

Un ruolo importante lo hanno avuto i miei genitori, il cui supporto e fiducia non sono mai venuti meno durante tutto il mio percorso universitario.

Per ultimo, vorrei ringraziare mio nonno che da sempre crede in me, mi ha consolato e spronato anche nei momenti più difficili. Il suo sostegno, che ho sempre percepito, è stato prezioso.

Appendice A

Codici utilizzati

```
rm(list = ls())
library(fBasics)
library(curvclust)
library(funHDDC)
library(ggsci)
library(funcharts)
library(fda)
library(fda.usc)
library(fdacluster)
library(ggplot2)
library(dplyr)
library(mclust)
library(clValid)
library(NbClust)
library(tidyverse)
library(tidyfun)
library(tidyr)
library(qcc)
library(pbmcapply)
library(factoextra)
library(gridExtra)
library(cowplot)

source("fclust.R")
source("functions.R")

resample <- function(x, ...) x[sample.int(length(x), ...)]
load(file.choose())
```

A.1 Analisi e studi di simulazione dataset `DRC.csv`

```
dati <- read.csv(file = "data.csv")
View(dati)

num_cluster_seq <- 2:10
ncores <- 1


x11(width=8, height=8)
plot(1:238, dati[,2], type="l", ylim = c(floor(min(dati[,-1])) -2,
                                         ceiling(max(dati[,-1]))+2),
     col="light blue", ylab = expression(paste("Resistance [",mu, Omega, "]")
    ),
     xlab = "Time [ms]", xlim=c(0,238))
for(i in 3:539){
  lines(dati[,i], col="light blue")
}
grid()
```

Metodi di *Clustering* per dati grezzi

```
tic <- proc.time()
dat <- dati
names(dat) <- gsub("X", "", names(dat))
datList <- list(X = as.matrix(dat[, - 1]), grid = dat$x)

datList$X <- datList$X[seq(1, 238, by = 13), ]
mod0<-raw_ms_nbclust(data = datList, num_cluster_seq = num_cluster_seq)
mod0$toc <- proc.time() - tic

clusterRaw_km <- mod0$mod_opt$ind_km$cluster
clusterRaw_gerarc <- mod0$mod_opt$ind_hc$cluster
clusterRaw_model <- mod0$mod_opt$mod_opt$classification
```

Metodi di *Clustering* per dati filtrati con B-*splines*

```
tic <- proc.time()
# 12 funzioni di base gomito GCV
modFiltBSpline <- fil_bspline_ms_nclust(data = datList, num_cluster_seq
    = 2:6, nbasis = 12)
modFiltBSpline$toc <- proc.time() - tic
clusterBSpl_km <- modFiltBSpline$mod_opt$ind_km$cluster
clusterBSpl_gerarc <-modFiltBSpline$mod_opt$ind_hc$cluster
clusterBSpl_model <- modFiltBSpline$mod_opt$mod_opt$classification
```

Metodi di *Clustering* basati sulla distanza (fda.usc)

```
dat <- dati
names(dat) <- gsub("X", "", names(dat))
datList <- list(X = as.matrix(dat[, - 1]), grid = dat$x)

set.seed(0)
tic <- proc.time()
modDist<-distance_ms(data = datList, num_cluster_seq = num_cluster_seq,
                  met="other", ncores = 1)
modDist$toc <- proc.time() - tic
clusterDist_silh <- modDist$mod_opt_sil$clus
```

Metodi di *Clustering* basati su punteggi FPCA (funHDDC)

```
# Tempo elaborazione circa 7 ore
tic <- proc.time()
modFPCA<-fit_funHDDC_ms(data=datList,num_cluster_seq = num_cluster_seq,
                    model = c('AkjBkQkDk','AkjBQkDk', 'AkBkQkDk',
                              'ABkQkDk', 'AkBQkDk', 'ABQkDk'),
                    threshold_seq = c(.2, .5, .9),
                    nb.rep = 20)
modFPCA$toc <- proc.time() - tic
clusterFPCA_model <- modFPCA$mod_opt$all_results$AKJBKQKDK_2_0.5$class
```

Metodi di *Clustering* basati su algoritmo adattivo (curvclust)

```
tic <- proc.time()
# Algoritmo curvclust implementato in Giacofci (2012)
num_cluster_seq <- 2:6
parameters <- expand.grid(
  structures = c("constant", "group", "scale.location", "group.scale.location",
    "none"),
  mixed = c(TRUE, FALSE), reduction = c(TRUE, FALSE)) %>%
  filter(!(structures == "none" & mixed),
         !(structures != "none" & !mixed)) %>%
  mutate(structures = as.character(structures))

mod2 <- vector(mode = "list", length = nrow(parameters))
names(mod2) <-
  sapply(1:nrow(parameters), function(ii) parameters[ii, ] %>%
           mutate(
             structures = paste0("structure_", structures),
             mixed = paste0("mixed_", mixed),
             reduction = paste0("reduction_", reduction)
           ) %>%
           paste0(collapse = " "))

for (ii in 1:nrow(parameters)) {
  tic <- proc.time()
  mod2[[ii]] <- curvclust_ms(data=datList,
                             num_cluster_seq = c(1, num_cluster_seq),
                             structure = parameters$structures[ii],
                             mixed = parameters$mixed[ii],
                             reduction = parameters$reduction[ii])
  mod2[[ii]]$toc <- proc.time() - tic
}

bic <- mod2 %>% sapply(function(x) x$BIC)

# seleziono numero cluster
bic %>%
  as.data.frame %>%
  mutate(K = c(1, num_cluster_seq)) %>%
```

```
  pivot_longer(- K, values_to = "BIC", names_to = "model") %>%
  ggplot +
  geom_line(aes(K, BIC, col = model))

arrind <- which(bic == max(bic), arr.ind = TRUE)
modopt <- mod2[[arrind[2]]]
Kopt <- which.max(modopt$BIC)[1] + 1
clusterCurvClust <- modopt$class[[which.max(modopt$BIC)[1]]]
```

Metodi di *Clustering* basati su componenti principali

```
tic <- proc.time()
modPrinComp<-fil_fpca_ms(data=datList,num_cluster_seq = num_cluster_seq)
modPrinComp$toc <- proc.time() - tic
clusterPrinComp_km <- modPrinComp$mod_opt$mod_opt_sil[[1]]
```

Grafici dati funzionali

```
# dati funzionali basati su metodo adattivo funHDDC
clusterfunHDDC <- get_clustered_funs_df(clusterFPCA_model, datList, "
    funHDDC adattivo")
x11(width=8, height=8)
plot(clusterfunHDDC$time[as.numeric(clusterfunHDDC$cluster)==1],
    clusterfunHDDC$Resistance[as.numeric(clusterfunHDDC$cluster)==1],
    type = "l", col="red",
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "funHDDC\nadattivo", ylim = c(floor(min(dati[,-1])) -2,
                                           ceiling(max(dati[,-1]))+2))
lines(clusterfunHDDC$time[as.numeric(clusterfunHDDC$cluster)==2],
    clusterfunHDDC$Resistance[as.numeric(clusterfunHDDC$cluster)==2], col
    ="lightblue")
grid()
legend("topright", legend=c(1,2), title="Cluster", col=c("red","lightblue"),
    lty = 1)

# dati funzionali rappresentati tramite
# metodo basato su distanze crit. silhouette
```

```
clusterFunDist_silh <- get_clustered_funs_df(clusterDist_silh, datList,
    "basato sulla distanza")
x11(width=8, height=8)
plot(clusterFunDist_silh$time[as.numeric(clusterFunDist_silh$cluster)
    ==1],clusterFunDist_silh$Resistance[as.numeric(clusterFunDist_silh$
    cluster)==1], type = "l", col="lightblue",
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "basato sulla distanza", ylim = c(floor(min(dati[,-1])) -2,
                                              ceiling(max(dati[,-1]))+2)
    )
lines(clusterFunDist_silh$time[as.numeric(clusterFunDist_silh$cluster)
    ==2],clusterFunDist_silh$Resistance[as.numeric(clusterFunDist_silh$
    cluster)==2], col="red")
grid()
legend("topright", legend=c(1,2), title="Cluster", col=c("red","lightblue"),
    lty = 1)

# dati funzionali basati su B-spline con metodo gerarchico
clusterFunBSpline_gerarc <- get_clustered_funs_df(clusterBSpl_gerarc,
    datList, "B-spline con metodo gerarchico")
x11(width=8, height=8)
plot(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==1],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==1], type = "l", col="yellow",
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "B-spline gerarchico", ylim = c(floor(min(dati[,-1])) -2,
                                            ceiling(max(dati[,-1]))+2))
lines(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==2],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==2], type = "l", col="red")
lines(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==3],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==3], type = "l", col="lightblue")
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c("red","lightblue",
     "yellow"), lty = 1)
```

Calcolo centroidi

```
centroidiRawkmeans <- get_centroids_df(clusterRaw_km, datList, "k-means
    dati grezzi")
centroidiRawgerarc <- get_centroids_df(clusterRaw_gerarc, datList, "
    gerarchico dati grezzi")
centroidiRaw_model <- get_centroids_df(clusterRaw_model, datList, "dati
    grezzi basato su modello")

centroidiBSpline_km <- get_centroids_df(clusterBSpl_km, datList, "k-
    means B-spline")
centroidiBSpline_gerarc <- get_centroids_df(clusterBSpl_gerarc, datList,
     "gerarchico B-spline")
centroidiBSpline_model <- get_centroids_df(clusterBSpl_model, datList, "
    B-spline basato su modello")

centroidiFPCA_model <- get_centroids_df(clusterFPCA_model, datList, "
    Functional PCA con modello")
centroidiCurvClust <- get_centroids_df(clusterCurvClust, datList, "
    Algoritmo adattivo curvclust")
centroidiDist_silh <- get_centroids_df(clusterDist_silh, datList, "basato
     sulla distanza")

# centroidi dati grezzi con metodo k-means
x11(width=8, height=8)
plot(centroidiRawkmeans$time[as.numeric(centroidiRawkmeans$cluster)==1],
    centroidiRawkmeans$Resistance[as.numeric(centroidiRawkmeans$cluster)
    ==1], type = "l", col=4, lwd = 2,
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "raw\nk-means", ylim = c(floor(min(dati[,-1])) -2,
                                          ceiling(max(dati[,-1]))+2))
lines(centroidiRawkmeans$time[as.numeric(centroidiRawkmeans$cluster)
    ==2],centroidiRawkmeans$Resistance[as.numeric(centroidiRawkmeans$
    cluster)==2], col=2, lwd = 2)
lines(centroidiRawkmeans$time[as.numeric(centroidiRawkmeans$cluster)
    ==3],centroidiRawkmeans$Resistance[as.numeric(centroidiRawkmeans$
    cluster)==3], col=3, lwd = 2)
grid()
```

```
legend("topright", legend=c(1,2,3), title="Cluster", col=c(4,2,3), lty = 1,
     lwd = 2)


# centroidi dati grezzi con metodo gerarchico
x11(width=8, height=8)
plot(centroidiRawgerarc$time[as.numeric(centroidiRawgerarc$cluster)==1],
    centroidiRawgerarc$Resistance[as.numeric(centroidiRawgerarc$cluster)
    ==1], type = "l", col=2,
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "raw\ngerarchico", ylim = c(floor(min(dati[,-1])) -2,
                                         ceiling(max(dati[,-1]))+2), lwd
    = 3)
lines(centroidiRawgerarc$time[as.numeric(centroidiRawgerarc$cluster)
    ==2],centroidiRawgerarc$Resistance[as.numeric(centroidiRawgerarc$
    cluster)==2], col=4, lwd = 3)
grid()
legend("topright", legend=c(1,2), title="Cluster", col=c(4,2), lty = 1, lwd
     = 3)

# centroidi basati sulla ditanza con criterio silhouette
x11(width=8, height=8)
plot(centroidiDist_silh$time[as.numeric(centroidiDist_silh$cluster)==1],
    centroidiDist_silh$Resistance[as.numeric(centroidiDist_silh$cluster)
    ==1], type = "l", col=2,
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "basato sulla distanza", ylim = c(floor(min(dati[,-1])) -2,
                                         ceiling(max(dati[,-1]))+2), lwd
    = 3)
lines(centroidiDist_silh$time[as.numeric(centroidiDist_silh$cluster)
    ==2],centroidiDist_silh$Resistance[as.numeric(centroidiDist_silh$
    cluster)==2], col=4, lwd = 3)
grid()
legend("topright", legend=c(1,2), title="Cluster",col=c(4,2), lty = 1, lwd
    = 3)

# centroidi B-spline con metodo k-means
x11(width=8, height=8)
```

```
plot(centroidiBSpline_km$time[as.numeric(centroidiBSpline_km$cluster)
    ==1],centroidiBSpline_km$Resistance[as.numeric(centroidiBSpline_km$
    cluster)==1], type = "l", col=4, lwd = 2,
     xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
     main = "B-spline\nk-means", ylim = c(floor(min(dati[,-1])) -2,
                                          ceiling(max(dati[,-1]))+2))
lines(centroidiBSpline_km$time[as.numeric(centroidiBSpline_km$cluster)
    ==2],centroidiBSpline_km$Resistance[as.numeric(centroidiBSpline_km$
    cluster)==2], col=3, lwd = 2)
lines(centroidiBSpline_km$time[as.numeric(centroidiBSpline_km$cluster)
    ==3],centroidiBSpline_km$Resistance[as.numeric(centroidiBSpline_km$
    cluster)==3], col=2, lwd = 2)
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c(4,2,3), lty = 1,
     lwd = 2)

# centroidi B-spline con metodo gerarchico
x11(width=8, height=8)
plot(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==1],centroidiBSpline_gerarc$Resistance[as.numeric(
    centroidiBSpline_gerarc$cluster)==1], type = "l", col=3,
     xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
     main = "B-spline\n gerarchico", ylim = c(floor(min(dati[,-1])) -2,
                                              ceiling(max(dati[,-1]))+2)
    , lwd=2)
lines(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==2],centroidiBSpline_gerarc$Resistance[as.numeric(
    centroidiBSpline_gerarc$cluster)==2], col=4, lwd=2)
lines(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==3],centroidiBSpline_gerarc$Resistance[as.numeric(
    centroidiBSpline_gerarc$cluster)==3], col=2, lwd=2)
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c(4,2,3), lty = 1,
     lwd=2)

# centroidi B-spline basati sul modello
x11(width=8, height=8)
plot(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
```

```
    cluster)==1],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==1], type = "l", col=1,
     xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
     main = "B-spline\n con modello", ylim = c(floor(min(dati[,-1])) -2,
                                              ceiling(max(dati[,-1]))
    +2))
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==2],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==2], col=2)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==3],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==3], col=3)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==4],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==4], col=4)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==5],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==5], col=5)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==6],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==6], col=6)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==7],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==7], col=7)
lines(centroidiBSpline_model$time[as.numeric(centroidiBSpline_model$
    cluster)==8],centroidiBSpline_model$Resistance[as.numeric(
    centroidiBSpline_model$cluster)==8], col=8)
grid()
legend("topright", legend=c(1,2,3,4,5,6,7,8), title="Cluster", col=as.
    numeric(centroidiBSpline_model$cluster), lty = 1)

# centroidi metodo funHDDC adattivo
x11(width=8, height=8)
plot(centroidiFPCA_model$time[as.numeric(centroidiFPCA_model$cluster)
    ==1],centroidiFPCA_model$Resistance[as.numeric(centroidiFPCA_model$
    cluster)==1], type = "l", col=4,
     xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
     main = "funHDDC\nadattivo", ylim = c(floor(min(dati[,-1])) -2,
```

```
                                                ceiling(max(dati[,-1]))+2),
    lwd = 3)
lines(centroidiFPCA_model$time[as.numeric(centroidiFPCA_model$cluster)
    ==2],centroidiFPCA_model$Resistance[as.numeric(centroidiFPCA_model$
    cluster)==2], col=2, lwd = 3)
grid()
legend("topright", legend=c(1,2), title="Cluster", col=c(4,2), lty = 1, lwd
     = 3)

# centroidi metodo adattivo curvclust
x11(width=8, height=8)
plot(centroidiCurvClust$time[as.numeric(centroidiCurvClust$cluster)==1],
    centroidiCurvClust$Resistance[as.numeric(centroidiCurvClust$cluster)
    ==1], type = "l", col=4,
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "curvclust\nadattivo", ylim = c(floor(min(dati[,-1])) -2,
                                             ceiling(max(dati[,-1]))+2),
    lwd = 3)
lines(centroidiCurvClust$time[as.numeric(centroidiCurvClust$cluster)
    ==2],centroidiCurvClust$Resistance[as.numeric(centroidiCurvClust$
    cluster)==2], col=2, lwd = 3)
grid()
legend("topright", legend=c(1,2), title="Cluster", col=c(4,2), lty = 1, lwd
     = 3)


# B-spline con metodo gerarchico per livello usura elettrodi
x11(width=8, height=8)
plot(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==1],centroidiBSpline_gerarc$Resistance[as.numeric(
    centroidiBSpline_gerarc$cluster)==1], type = "l", col=3,
      xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
      main = "B-spline gerarchico\n Livelli usura elettrodi ", ylim = c(floor(
    min(dati[,-1])) -2,
                                                ceiling(max(dati[,-1]))+2)
    , lwd=3)
lines(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==2],centroidiBSpline_gerarc$Resistance[as.numeric(
```

```
    centroidiBSpline_gerarc$cluster)==2], col=4, lwd=3)
lines(centroidiBSpline_gerarc$time[as.numeric(centroidiBSpline_gerarc$
    cluster)==3],centroidiBSpline_gerarc$Resistance[as.numeric(
    centroidiBSpline_gerarc$cluster)==3], col=2, lwd=3)

lines(dat[,2], lty="dotted", col="green")
lines(dat[,25], lty="dashed", col="red")
lines(dat[,188], lty="dotted", col="green")
lines(dat[,238], lty="dashed", col="red")
lines(dat[,18], col="lightblue")
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c(4,2,3), lty = 1,
     lwd=3)
legend("topleft", legend=c("Nuovo","Usura media", "Usura severa"), title="
    Livello usura", col=c("lightblue", "red", "green"), lty = c(1,2,3))

# Adattamento procedure clustering a dataset con osservazioni OC
test_data <- read_csv("test_data.csv")
test_mat <- as.matrix(test_data)

matplot((datList$X), col = clusterBSpl_gerarc, type = "l")

clusterBSpl_gerarc_OC <- clusterBSpl_gerarc
clusterBSpl_gerarc_OC[clusterBSpl_gerarc == 1] <- 3
clusterBSpl_gerarc_OC[clusterBSpl_gerarc == 2] <- 1
clusterBSpl_gerarc_OC[clusterBSpl_gerarc == 3] <- 2

delta <- datList$grid[2]

cluster_assignment <- sapply(1:3, function(ii) {
  colSums(((test_mat -
              (get_centroids_df(clusterBSpl_gerarc_OC, datList, "filtering
    B-spline hc") %>%
                 filter(cluster == ii) %>%
                 pull(Resistance))) ^ 2) * delta)
}) %>%
  apply(1, which.min)

# calcolo distanze
distances <- lapply(1:3, function(ii) {
```

```
  colSums(((datList$X[, clusterBSpl_gerarc_OC == ii] -
              (get_centroids_df(clusterBSpl_gerarc_OC, datList, "filtering
    B−spline hc") %>%
                 filter(cluster == ii) %>%
                 pull(Resistance))) ^ 2) * delta)
}) %>%
  setNames(1:3)


distance_test <- sapply(1:3, function(ii) {
  colSums(((test_mat -
              (get_centroids_df(clusterBSpl_gerarc_OC, datList, "filtering
    B−spline hc") %>%
                 filter(cluster == ii) %>%
                 pull(Resistance))) ^ 2) * delta)
})

limits <- sapply(distances, function(x) quantile(x, .99))

control_chart_df <-  lapply(1:3, function(ii)
  data.frame(distanza = distance_test[cluster_assignment == ii, ii],
             cluster = ii,
             limiti = limits[ii])
) %>%
  bind_rows %>%
  mutate(osservazioni = rownames(.),
         cluster = factor(cluster)) %>%
  arrange(osservazioni) %>%
  mutate(osservazioni = 1:n())

control_chart <- control_chart_df %>%
  ggplot +
  geom_line(aes(osservazioni, distanza), col = "black") +
  geom_line(aes(osservazioni, limiti), col = "black", lty = 2) +
  geom_point(aes(osservazioni, distanza, col = cluster)) +
  theme_bw() +
  scale_x_continuous(breaks = 1:100) +
  scale_color_brewer(palette = "Set1")
control_chart
ggsave("cartaControllo.png", plot = control_chart)
```

```
# modello B-spline gerarchico con profili OC
x11(width=8, height=8)
plot(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==1],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==1], type = "l", col="yellow",
     xlab = "Time [ms]", ylab = expression(paste("Resistance [",mu, Omega,
    "]")),
     main = "B-spline gerarchico con profili OC", ylim = c(floor(min(dati
    [,-1])) -2,
                                                          ceiling(max(
    dati[,-1]))+2))
lines(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==2],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==2], type = "l", col="red")
lines(clusterFunBSpline_gerarc$time[as.numeric(clusterFunBSpline_gerarc$
    cluster)==3],clusterFunBSpline_gerarc$Resistance[as.numeric(
    clusterFunBSpline_gerarc$cluster)==3], type = "l", col="lightblue")
lines(1:NROW(test_data), test_data$V411, lty = "dotted", lwd = 2)
lines(1:NROW(test_data), test_data$V574, lty = "dashed", lwd = 2)
lines(1:NROW(test_data), test_data$V617, lty = "dotdash", lwd = 2)
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c("red","lightblue",
     "yellow"), lty = 1)
legend("topleft", legend=c("7","9", "10"), title="Osservazione", col="black"
    , lty = c("dotted", "dashed", "dotdash"))


# distanza tra cluster
distanzaCluster <- function(dati, centroide, pesi=1){
  result <- matrix(NA, NCOL(dati$X), 3)
  for(j in 1:NCOL(dati$X)){
    for(i in 1:3){
      d_ij <- dati$X[,j] - centroide$Resistance[as.numeric(centroide$
    cluster) == i]
      result[j,i] <- sqrt(pesi*t(d_ij)%*%(d_ij))
    }
  }
  result
}
```

```
matriceDistanze <- distanzaCluster(datList, centroidiBSpline_gerarc)
nearestCluster <- apply(matriceDistanze, 1, which.min)

gerarchico_grezzi <- hcut(matriceDistanze, k=3, stand = TRUE, plot =
    TRUE)

# Definizione variabili per metodo multivariato
resistenzaFinale <- datList$X[238,]
idx <- (1:NROW(datList$X) <= 50)
resistenzaMinima <- apply(datList$X[idx,], 2, min)
t_idx <- apply(datList$X[idx,], 2, which.min)
resistenzaMassima <- apply(datList$X[(min(t_idx)+1):70,], 2, max)
t_NF <- apply(datList$X[(min(t_idx)+1):NROW(datList$X),], 2, which.max)
    + min(t_idx)
Delta_T <- t_NF-t_idx
Delta_Res <- resistenzaMassima-resistenzaMinima
Delta_TFinale <- length(idx)-t_NF
caratteristicheDataset <- as.data.frame(cbind(resistenzaFinale, Delta_T,
     Delta_Res, Delta_TFinale))

# Metodo k-means applicato a metodo multivariato
multiv_km <- kmeans(caratteristicheDataset,centers = 3)
x11(width=8, height=8)
plot(1:238, dati[,2], type="l", ylim = c(floor(min(dati[,-1])) -2,
                                         ceiling(max(dati[,-1]))+2),
     col=(multiv_km$cluster[1]), ylab = expression(paste("Resistance [",mu
    , Omega, "]")),
     xlab = "Time [ms]", xlim=c(0,238), main = "Multivariato k-means")
for(i in 3:539){
  lines(dati[,i], col=(multiv_km$cluster[i-1]))
}
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c(3,2,1), lty=1)

# Metodo gerarchico applicato a metodo multivariato
multiv_gerarc <- hcut(caratteristicheDataset, k = 3)
x11(width=8, height=8)
plot(1:238, dati[,2], type="l", ylim = c(floor(min(dati[,-1])) -2,
                                         ceiling(max(dati[,-1]))+2),
     col=(multiv_gerarc$cluster[1]), ylab = expression(paste("Resistance [
```

```
    ",mu, Omega, "]")),
     xlab = "Time [ms]", xlim=c(0,238), main = "Multivariato gerarchico")
for(i in 3:539){
  lines(dati[,i], col=(multiv_gerarc$cluster[i-1]))
}
grid()
legend("topright", legend=c(1,2,3), title="Cluster", col=c(2,3,1), lty=1)

# indice BIC per clustering basato sul modello
BIC <- mclust::mclustBIC(dati[,-1], G=1:4)
multiv_model <- mclust::Mclust(dati[,-1], G=1:4, x=BIC)
summary(multiv_model, parameters = FALSE)

# rappresentazione B-spline per covariate funzionali
basis <- create.bspline.basis(c(0, 1), nbasis = 12)
loglam         = seq(-10, -4, 0.25)
Gcvsave        = numeric()
for(i in 1:length(loglam)){
  fdPari  = fdPar(basis, Lfdobj=2, 10^loglam[i])
  Sm.i    = smooth.basis(datList$grid, datList$X, fdPari)
  Gcvsave[i] = sum(Sm.i$gcv)
}
lambda_s=10^loglam[which.min(Gcvsave)]
fdPari  = fdPar(basis, Lfdobj=2,lambda_s)
X_fd<-smooth.basis(datList$grid,
                   datList$X,
                   fdPari)$fd
# calcolo distanza L2 per le curve a disposizione
dist_l2 <- semimetric.basis(X_fd)

# Creazione dataframe per modelli multivariati semplici
df_rev <- bind_cols(
  datList$X %>%
    as.data.frame %>%
    mutate(x = 1:238) %>%
    pivot_longer(-x) %>%
    filter(x < 50) %>%
    group_by(name) %>%
    summarise(min = min(value),
              which_min = x[which.min(value)]),
```

```
  datList$X %>%
    as.data.frame %>%
    mutate(x = 1:238) %>%
    pivot_longer(-x) %>%
    filter(x > 25) %>%
    group_by(name) %>%
    summarise(max = max(value),
              which_max = x[which.max(value)]) %>%
    select(-name),

  datList$X %>%
    as.data.frame %>%
    mutate(x = 1:238) %>%
    pivot_longer(-x) %>%
    group_by(name) %>%
    summarise(end = tail(value, 1)) %>%
    select(-name)
) %>%
  mutate(ampl_diff = max - min,
         phase_diff = which_max - which_min) %>%
  mutate(name = factor(name, levels = colnames(datList$X))) %>%
  arrange(name)

# Definisco matrice B-spline a partire dai dati funzionali
B_spline<-t(X_fd$coefs)
dimnames(B_spline) <- NULL

# Procedura Bootstrap per confronto silhouette tra i 4 modelli
set.seed(0)
bootstrap_sil <- mclapply(1:500, function(ii) {
    rows <- sample(1:538, replace = TRUE)
    B <- caratteristicheDataset %>%
      slice(rows) %>%
      select(resistenzaFinale, Delta_T, Delta_Res)

    mod_opt_hc <- hcut(B, k = 3)
    mod_opt_km <- kmeans(B, centers = 3)

    cl_hc <- factor(mod_opt_hc$cluster)
```

```
    cl_km <- factor(mod_opt_km$cluster)

    mod_opt_hc_bspline <- hcut(B_spline[rows, ], k = 3)
    mod_opt_km_bspline <- kmeans(B_spline[rows, ], centers = 3)

    data.frame(
      f_bspline_hc   = mean(silhouette(mod_opt_hc_bspline$cluster, dist_
    l2[rows, rows])[, 3]),
      f_bspline_km   = mean(silhouette(mod_opt_km_bspline$cluster, dist_
    l2[rows, rows])[, 3]),
      multiv_hc         = mean(silhouette(as.numeric(cl_hc), dist_l2[
    rows, rows])[, 3]),
      multiv_km         = mean(silhouette(as.numeric(cl_km), dist_l2[
    rows, rows])[, 3])
    )
  }, mc.cores = 1) %>%
    bind_rows()

# grafici Indice Silhouette bootstrap
x11(width = 8, height = 8)
par(mfrow = c(2, 2))
for(i in 1:NCOL(bootstrap_sil)){
  plot(bootstrap_sil[1:30,i], type="b", xlab = "Replicazione bootstrap",
    ylab = "Indice silhouette", main = names(bootstrap_sil)[i])
}

# sintesi distribuzione Indice Silhouette tramite boxplot
x11(width=10, height=10)
boxplot(bootstrap_sil, horizontal = TRUE, xlab = "Indice silhouette", ylab =
     "Metodo", names = names(bootstrap_sil))
grid()
```

A.2 Analisi e studi di simulazione dataset ShipNavigation

```
load("ShipNavigation.RData")
load(file.choose())
str(ShipNavigation)
ShipNavigation <- as.data.frame(ShipNavigation)
View(ShipNavigation)
```

```r
# Calcolo temperatura media motori
temp <- cbind(ShipNavigation$temp1, ShipNavigation$temp2, ShipNavigation
    $temp3, ShipNavigation$temp4)
temp <- rowMeans(temp)

ShipNavigation <- cbind(ShipNavigation[,1:11], temp, ShipNavigation[,16:
    NCOL(ShipNavigation)])

# seleziono solo le curve con le tratte da Porto 3 a Porto 1
# definisco anche la variabile che rappresenta il rapporto tra le
    emissioni
# e la distanza percorsa
dft <- ShipNavigation %>%
  group_by(VN) %>%
  mutate(CO2pm = CO2_emissions / nautic_miles) %>%
  ungroup() %>%
  filter(position == "PORT3 -> PORT1") %>%
  na.omit()

yvar <- "CO2pm"
xvar <- c("sog", "w_long", "w_trasv", "trim")

x <- get_mfd_df(dt = dft,
                arg = "fraction of distance travelled",
                domain = c(0, 1),
                id = "VN",
                variables = c(xvar, yvar),
                n_basis = 100)
obs <- unique(dft$VN)
obs_numeric <- as.numeric(substr(obs, 3, 7))
obs1 <- obs[obs_numeric >= 517 & obs_numeric <= 1248]
obs2 <- obs[obs_numeric > 1248 & obs_numeric <= 2054]

x1 <- x[obs1, xvar]
y1 <- x[obs1, yvar]

# Modello regressione con risposta funzionale
# e covariate funzionali
mod <- fof_pc(y1, x1, type_residuals = "studentized")
outliers <- get_outliers_mfd(mod$residuals)
```

```
obs1_ok <- obs1[-outliers]
obs1_train <- obs1_ok[1:110]
obs1_tun <- setdiff(obs1_ok, obs1_train)
x1train <- x1[obs1_train]
y1train <- y1[obs1_train]
x1tun <- x1[obs1_tun]
y1tun <- y1[obs1_tun]
x2 <- x[c(obs1_tun, obs2), xvar]
y2 <- x[c(obs1_tun, obs2), yvar]

modTrain <- fof_pc(y1train, x1train, type_residuals = "studentized")
plot_bifd(modTrain$beta_fd)

# Carta di controllo su dati di training
cc <- regr_cc_fof(modTrain,
                  mfdobj_y_new = y2,
                  mfdobj_x_new = x2,
                  mfdobj_y_tuning = y1tun,
                  mfdobj_x_tuning = x1tun,
                  alpha = 0.01)
x11(width = 8, height = 8)
plot_control_charts(cc, nobsI = length(obs1_tun))
dev.off()

# componenti principali funzionali
xPCA <- pca_mfd(x1train, nharm = 30)
x11(width = 8, height = 8)
plot(1:30, xPCA$varprop, type = "l", xlab = "Numero componenti", ylab = "
    Frazione varianza spiegata", main = "Componenti principali funzionali")
lines(1:30,y=rep(0.01, 30), lty = "dashed")
grid()
min(which(xPCA$varprop <= 0.01))

# Errore di previsione
yhat <- predict_fof_pc(object = modTrain,
                       mfdobj_y_new = y2,
                       mfdobj_x_new = x2)

x11(width=8, height=8)
plot_mon(cc, modTrain$residuals, yhat$pred_error[70], plot_title = TRUE)
```

```
plot_mon(cc, modTrain$residuals, yhat$pred_error[178], plot_title = TRUE
    )
plot_mon(cc, modTrain$residuals, yhat$pred_error[221], plot_title = TRUE
    )

# Carte di controllo di Fase II
xl <- get_mfd_df_real_time(dt = dft,
                           domain = c(0, 1),
                           arg = "predicted fraction of distance travelled",
                           id = "VN",
                           variables = c(xvar, yvar),
                           n_basis = 100)
y1l <- lapply(xl, function(x) x[obs1_train, yvar])
x1l <- lapply(xl, function(x) x[obs1_train, xvar])
y1ltun <- lapply(xl, function(x) x[obs1_tun, yvar])
x1ltun <- lapply(xl, function(x) x[obs1_tun, xvar])
y2l <- lapply(xl, function(x) x[c(obs1_tun, obs2), yvar])
x2l <- lapply(xl, function(x) x[c(obs1_tun, obs2), xvar])

modl <- fof_pc_real_time(mfdobj_y_list = y1l,
                         mfdobj_x_list = x1l,
                         type_residuals = "studentized")

ccl <- regr_cc_fof_real_time(mod_list = modl,
                             mfdobj_y_new_list = y2l,
                             mfdobj_x_new_list = x2l,
                             mfdobj_y_tuning_list = y1ltun,
                             mfdobj_x_tuning_list = x1ltun,
                             alpha = 0.01)

x11(width = 8, height = 8)
plot_control_charts_real_time(ccl, 70)
plot_control_charts_real_time(ccl, 178)
plot_control_charts_real_time(ccl, 221)
```

Bibliografia

Auer P. e Gervini D. (2008). «Choosing Principal Components: A New Graphical Method Based on Bayesian Model Selection». *Communications in Statistics - Simulation and Computation* 37.5, pp. 962–977. DOI: 10.1080/03610910701855005.

Bottani E., Montanari R., Volpi A. e Tebaldi L. (2023). «Statistical process control of assembly lines in manufacturing». *Journal of Industrial Information Integration* 32, p. 15. DOI: https://doi.org/10.1016/j.jii.2023.100435.

Bouveyron C. e Jacques J. (2011). «Model-based clustering of time series in group-specific functional subspaces». *Adv Data Anal Classif* 5, pp. 281–300. DOI: 10.1007/s11634-011-0095-6.

Capezza C., Centofanti F., Lepore A., Menafoglio A., Palumbo B. e Vantini S. (2023a). «funcharts: control charts for multivariate functional data in R». *Journal of Quality Technology*. DOI: 10.1080/00224065.2023.2219012.

Capezza C., Centofanti F., Lepore A., Menafoglio A., Palumbo B. e Vantini S. (2023b). *funcharts: Functional Control Charts*. R package version 1.4.1. URL: https://CRAN.R-project.org/package=funcharts.

Capezza C., Centofanti F., Lepore A. e Palumbo B. (2020). *funclustRSW repository on github*. URL: https://github.com/unina-sfere/funclustRSW.

Capezza C., Centofanti F., Lepore A. e Palumbo B. (2021). «Functional clustering methods for resistance spot welding process data in the automotive industry». *Applied Stochastic Models in Business and Industry* 37.5, pp. 908–925. DOI: 10.1002/asmb.2648.

Capezza C., Lepore A., Menafoglio A., Palumbo B. e Vantini S. (2020). «Control charts for monitoring ship operating conditions and CO_2 emissions based

on scalar-on-function regression». *Applied Stochastic Models in Business and Industry* 36.3, pp. 477–500. DOI: 10.1002/asmb.2507.

Centofanti F., Lepore A., Menafoglio A., Palumbo B. e Vantini S. (2021). «Functional Regression Control Chart». *Technometrics* 63.3, pp. 281–294. DOI: 10.1080/00401706.2020.1753581.

Giacofci M., Lambert-Lacroix S., Marot G. e Picard F. (2012). *curvclust: curve clustering*. R package version 0.0.1. URL: https://cran.r-project.org/src/contrib/Archive/curvclust/.

Giacofci M., Lambert-Lacroix S., Marot G. e Picard F. (2013). «Wavelet-Based Clustering for Mixed-Effects Functional Models in High Dimension». *Biometrics* 69, pp. 31–40. DOI: 10.1111/j.1541-0420.2012.01828.x.

Hall P. e Hosseini-Nasab M. (2006). «On properties of functional principal components analysis». *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 109–126. DOI: 10.1111/j.1467-9868.2005.00535.x.

Hastie T., Tibshirani R. e Friedman J. (2009). *The Elements of Statistical Learning*. Springer New York, NY. DOI: 10.1007/978-0-387-84858-7.

James Gareth M. e Sugar Catherine A. (2003). «Clustering for Sparsely Sampled Functional Data». *Journal of the American Statistical Association* 98.462, pp. 397–408. DOI: 10.1198/016214503000189.

Kourti T. (2005). «Application of latent variable methods to process control and multivariate statistical process control in industry». *International Journal of Adaptive Control and Signal Processing* 19.4, pp. 213–246. DOI: 10.1002/acs.859.

Kourti T. e MacGregor John F. (1996). «Multivariate SPC Methods for Process and Product Monitoring». *Journal of Quality Technology* 28.4, pp. 409–428. DOI: 10.1080/00224065.1996.11979699.

Nomikos P. e MacGregor John F. (1995). «Multivariate SPC Charts for Monitoring Batch Processes». *Technometrics* 37.1, pp. 41–59. DOI: 10.1080/00401706.1995.10485888.

Ramsay J. O. e Silverman B. W. (2005). *Functional Data Analysis*. Springer New York. DOI: 10.1007/b98888.

Zhang J., Ren H., Yao R., Zou C. e Wang Z. (2015). «Phase I analysis of multivariate profiles based on regression adjustment». *Computers & Industrial Engineering* 85, pp. 132–144. DOI: 10.1016/j.cie.2015.02.025.