



UNIVERSITA' DEGLI STUDI DI PADOVA

Facoltà di ingegneria

Corso di Laurea Specialistica in

Ingegneria Informatica

**ANALISI DELL'APPLICAZIONE DELLA
TRASFORMATA DI KARHUNEN-LOÈVE A
SEGNALI DA PULSAR OTTICHE**

Relatore: *Prof. Giampiero Naletto*

Correlatore: *Dott. Enrico Verroi*

Laureando: *Giacomo Paolo Codogno*

Anno Accademico 2009/2010

Indice

1 Introduzione

1.1 Abstract	1
1.2 L'informazione della luce	2
1.3 QuantEye.....	3
1.4 Da QuantEye a IquEye	4

2 IquEye

2.1 IquEYE : Italian Quantum Eye.....	7
2.2 Il telescopio NTT di La Silla	8
2.3 IquEye : caratteristiche ottiche	9
2.4 IquEye : caratteristiche progettuali.....	12
2.5 IquEye : sistema di acquisizione dati.....	14

3 Pulsar

3.1 Pulsating stars	19
3.2 Curva di luce.....	22
3.3 Oggetto dell'osservazione.....	23

4 Analisi dati

4.1 Il segnale ed il suo preprocessing.....	25
4.2 Analisi di Fourier	27
4.3 Folding dei dati per Crab pulsar	30
4.4 Folding dei dati per PSR B0540-69	36
4.5 Ricerca del periodo di rotazione di PSR B0540-69	38
4.6 L'idea della KLT per Vela pulsar.....	41

5 Principal Component Analysis (PCA)

5.1 Introduzione.....	43
5.2 Il fondamento della Principal Component Analysis	44
5.3 Base ottimale del data set.....	47
5.4 Matrice di covarianza.....	48
5.5 Costruzione di P	50
5.6 La scelta delle componenti principali.....	52

5.7 L'algoritmo per il calcolo della PCA.....	54
6 Karhunen-Loève Transform (KLT)	
6.1 Un approccio intuitivo alla KLT per signal processing.....	57
6.2 La teoria per il caso continuo.....	58
6.3 Il calcolo della KLT nel caso discreto.....	63
6.4 L'eigenspectrum.....	66
7 Autovalori ed autovettori	
7.1 Introduzione al problema.....	71
7.2 Soluzione del problema.....	72
7.3 Metodo delle potenze.....	72
7.4 Il metodo di Jacobi.....	73
7.5 Il metodo di Lanczos.....	75
8 Test e valutazioni	
8.1 Introduzione.....	79
8.2 PCA su dati di IquEye.....	80
8.3 L'algoritmo <i>ufolding</i>	84
8.4 Risultati per <i>ufolding</i>	86
8.5 Calcolo della KLT.....	92
8.6 Risultati KLT.....	94
9 Conclusioni	
9.1 Conclusioni.....	101
9.2 Sviluppi futuri.....	102
Bibliografia.....	103

CAPITOLO 1

Introduzione

1.1 Abstract

Questa tesi, svolta presso l'Unità Organizzativa di Supporto (UOS) di Padova dell'Istituto di Fotonica e Nanotecnologie (CNR IFN), si pone l'obiettivo di realizzare un nuovo metodo per l'analisi e il denoising di segnali per mezzo di uno strumento matematico ancora poco noto e utilizzato: la trasformata di Karhunen-Loève o KLT.

I segnali che hanno portato alla necessità di ricercare tale metodo, sono caratterizzati da un forte rumore e dalla grande mole di dati che li rappresenta. Essi sono il risultato dell'osservazione di alcune stelle pulsar per mezzo di due telescopi, prima del telescopio Copernico di Asiago e poi del telescopio NTT situato in Cile. Il dispositivo che ha prodotto i dati è un prototipo di nome IquEye, realizzato tramite una collaborazione tra il dipartimento di elettronica e informatica e il dipartimento di astronomia dell'università di Padova.

In queste pagine viene descritto il funzionamento dello strumento usato per l'acquisizione dei dati e il lavoro che è stato svolto sul segnale presso i laboratori Luxor. In seguito viene dato spazio alla trasformata KLT nelle sue varie versioni esistenti in letteratura, dipendenti dall'applicazione che se ne intende fare, che hanno generato non poca confusione nel trovare la giusta via da seguire per arrivare allo scopo prefissato.

La difficoltà di reperire documentazione utile allo svolgimento del lavoro, unita al forte rumore del segnale ed alla grande quantità di dati, che implica un grande impiego di risorse, ha portato a numerosi problemi che verranno spiegati con maggiore dettaglio.

Infine viene fatto un bilancio della situazione attuale, di cosa si è riuscito ad ottenere, degli obiettivi che, per mancanza di mezzi o informazioni, non è stato possibile raggiungere, e delle eventuali applicazioni future del lavoro svolto.

1.2 L'informazione della luce

In astronomia, quasi tutti gli strumenti adibiti allo studio delle radiazioni elettromagnetiche si occupano di registrare la direzione di arrivo dei fotoni provenienti dalla sorgente osservata (camere) oppure l'energia con cui il fotone arriva allo strumento (spettrometri).

E' stato dimostrato con studi teorici che per studiare il processo fisico a monte della radiazione e conoscere la composizione della sorgente emittente è possibile utilizzare l'informazione trasportata dai singoli fotoni, o da gruppi di essi, che fino ad ora non è stato possibile apprendere. Singoli fotoni possono trasportare il *momento angolare orbitale* (oltre al loro normale momento angolare) e la statistica di arrivo dei fotoni in una stringa temporale può fornire informazioni su come tali fotone siano stati creati o modificati durante la loro propagazione. Tali informazioni si possono apprendere superando l'analisi del primo ordine di coerenza e ricavando le proprietà del secondo e dei più alti ordini di coerenza della luce. Perché questo sia possibile è necessario misurare le statistiche di arrivo dei fotoni con una precisione dell'ordine dei picosecondi.

I telescopi più grandi offrono un enorme incremento di sensibilità per studiare le variabilità astrofisiche su una scala temporale del milli-, micro- o nano-secondo. Siccome i fenomeni astrofisici sono solitamente non periodici e la loro esatta scala temporale è sia sconosciuta che variabile, gli studi sulle stringhe di fotoni devono essere di tipo statistico ed eseguiti su un grande numero di essi. Le funzioni di correlazione del secondo ordine aumentano col quadrato dell'intensità della luce raccolta, quindi al raddoppiare del diametro del telescopio l'area si quadruplica e il segnale aumenta di un fattore sedici; da qui si capisce l'utilità di telescopi di grandi dimensioni o ELT (Extremely Large Telescopes).

1.3 QuantEye

La realizzazione di strumenti capaci di grandi prestazioni nel conteggio dei fotoni è uno degli obiettivi dell'astronomia moderna, e in tale ambito trovano spazio i prototipi AquEye e IquEye, costruiti per arrivare in futuro a QuantEye, lo strumento che avrà la più alta risoluzione temporale per applicazioni astronomiche.

QuantEye è stato pensato considerando le possibilità che avrebbe potuto offrire il telescopio OWL (Over-Whelmingly Large), progetto del 2005 di un telescopio di 100 m per l'Osservatorio Europeo Meridionale (ESO), il quale però, nonostante sia stato considerato dall'International Review Panel come un progetto fondamentale, è stato poi ridimensionato considerevolmente ad un telescopio di 42 metri di diametro. Causa di questa riduzione è stato l'enorme costo e la notevole complessità di realizzazione.

Nonostante il progetto finale sia stato modificato in corso d'opera, mantiene la sua validità scientifica e potrà essere facilmente adattato ad altri telescopi di tipo ELT.

QuantEye è stato concepito per essere il fotometro a conteggio fotoni più veloce esistente. Il progetto prevede 100 canali paralleli, ognuno dei quali adibito al campionamento di un centesimo della pupilla del telescopio e in grado di assegnare il tempo di arrivo di ogni fotone in arrivo con una accuratezza dell'ordine dei 10 ps; l'intero sistema sarebbe così in grado di mantenere un conteggio globale di 1 GHz per ore.

Il progetto di QuantEye segue due specifiche. La prima, ovviamente è di sfruttare le caratteristiche prestazionali offerte da OWL, che nel progetto originale aveva un'apertura di 100 m con rapporto focale di $f / 6$; inoltre esso era stato completamente corretto da aberrazioni geometriche e limitato nell'osservazione dall'assenza di un'ottica adattiva (AO System). Poichè QuantEye è essenzialmente un fotometro a conteggio fotoni ad altissima velocità, non necessita di un sistema di ottica adattiva se i detector sono abbastanza grandi. In normali condizioni di "seeing" era stato previsto che OWL riuscisse a concentrare una grande frazione della luce proveniente da un punto sorgente posto all'infinito all'interno di un'immagine di 1 arcosecondo; considerando la focale di 600 m, ciò significa che un oggetto "puntiforme" in cielo dovrebbe riempire normalmente uno spot di 3 mm al fuoco.

La seconda specifica che ha guidato il progetto di QuantEye è stata la limitata disponibilità di veloci detectors che contassero fotoni, o meglio l'inadeguatezza di quelli già esistenti. Ad esempio, sia i CCD che i CMOS non permettono una veloce rilevazione degli eventi, i rivelatori MCP (Microchannel Plate Detector), che possono avere una risoluzione temporale inferiore a poche decine di picosecondi, sono limitati da un conteggio massimo troppo basso, di pochi kHz, e da una bassa efficienza del fotocatodo nel visibile.

Sfortunatamente, allo stato dell'arte, nessun contatore di fotoni non-imaging è abbastanza performante da soddisfare le specifiche richieste. In pratica, le uniche possibilità sono: i fotomoltiplicatori, che in ogni caso hanno una bassa efficienza quantica; oppure i fotodiodi contatori di fotoni a valanga, (SPAD: Single Photon Avalanche photoDiode).

Attualmente, proprio quest'ultimi hanno le migliori qualità: con le moderne tecniche di realizzazione, gli SPAD raggiungono una risoluzione temporale inferiore ai 50 ps e una massima velocità di conteggi globali superiore a 10 MHz. Anche se la risoluzione temporale è accettabile per l'applicazione, rimane irrisolto il problema del massimo "*count rate*" che è almeno due ordini di grandezza inferiore al necessario.

Viste le limitazioni e la difficoltà di creare arrays di SPAD, è stato deciso di adottare una soluzione ottica nella quale la luce raccolta dal telescopio va a cadere su un array di SPAD "distribuiti", cioè un array di singoli SPAD separati.

1.4 Da QuantEye a IquEye

Prima di realizzare lo strumento QuantEye per il telescopio OWL, si è deciso di applicare gli stessi concetti per costruirne una versione molto più ridotta, compatibilmente anche con i fondi disponibili, per un'applicazione al telescopio di 182 cm di Cima Ekar di Asiago: questo prototipo è stato chiamato AquEye (Asiago Quantum Eye).

Tale telescopio offre buone possibilità di tempi di osservazione per poter testare lo strumento. Inoltre, uno degli strumenti presenti al piano focale del telescopio, AFOSC

(Asiago Faint Object Spectrograph and Camera), può essere facilmente utilizzato per agevolare la progettazione di AquEye (Fig. 1).

Ovviamente, le specifiche di AquEye sono molto meno stringenti rispetto a quelle richieste a QuantEye.

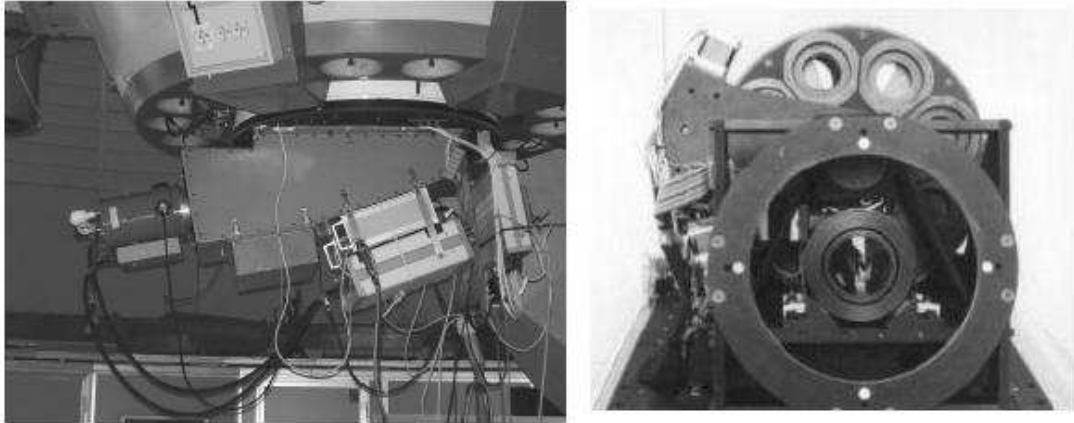


Figura 1 A sinistra: una vista di AFOSC montato sul fuoco del telescopio Cassegrain di 182 cm di Asiago. A destra: la lente di uscita di AFOSC al centro della flangia di montaggio di AquEye.

Un modo molto semplice di realizzare questo piccolo prototipo è stato quello di considerare una configurazione ottica in cui la pupilla del telescopio venga divisa in solo quattro parti. Seguendo questa premessa, una strada facile per ottenere quattro percorsi ottici indipendenti è di inserire uno specchio piramidale all'uscita di AFOSC, dove il fascio di radiazione diverge (Fig. 2). I raggi riflessi dalla piramide vengono indipendentemente spediti lungo quattro direzioni ed ognuno di essi può essere collimato tramite una lente appropriata. A questo punto possono essere inseriti sia filtri che polarizzatori, mentre successivamente il fascio viene focalizzato tramite una seconda lente sullo SPAD.

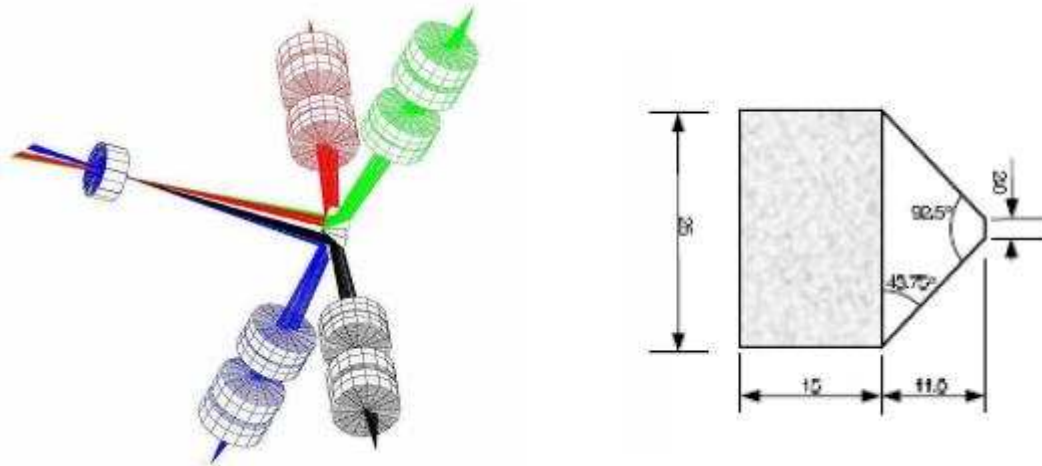


Figura 2 A sinistra: dopo l'ultima lente di AFOSC mostrato sulla sinistra, una piramide divide la luce in quattro canali separati per gli SPAD. A destra: dettagli meccanici della piramide.

Dopo mesi di operazioni compiute con AquEye, sia in laboratorio che sul telescopio di Asiago di 182 cm, i risultati ottenuti sono stati più che soddisfacenti e hanno confermato le aspettative per cui tale strumento è stato realizzato. Grazie ad AquEye è stata acquisita, da parte degli studiosi, la prima esperienza con l'astronomia quantica nei suoi differenti aspetti (statistiche delle stringhe di fotoni, spettroscopia di foto correlazione, interferometria di intensità). Sono stati, inoltre, riscontrati diversi limiti dovuti alle dimensioni, al sito del telescopio usato per l'elettronica ed alla meccanica.

Per confermare definitivamente la validità dei concetti proposti con AquEye e per superare alcuni dei suoi limiti, è stato progettato e realizzato IquEye (Italian Quantum Eye).

CAPITOLO 2

IquEye

2.1 IquEYE : Italian Quantum Eye

Il progetto di IquEye segue direttamente dallo sviluppo di AquEye, infatti lo schema concettuale rimane molto simile. Uno dei limiti di AquEye è il tilt del fascio dal centro della piramide. Come mostrato in Fig. 3, nel caso avvenga un tilt rispetto all'asse del telescopio, esso finisce in maniera non centrata sulla piramide specchiata. Quando il fascio non è più simmetrico rispetto alle quattro facce della piramide esso viene riflesso in maniera non più omogenea sui quattro SPAD, cosicchè può capitare che uno di loro riceva più radiazione luminosa, con un conseguente aumento del conteggio rispetto a quello opposto. Inoltre, senza la piattaforma AFOSC lo strumento non riuscirebbe a lavorare indipendentemente, in quanto necessita di alcune strumentazioni, tra cui un proprio specchio di campo e un riduttore focale; in più la radiazione luminosa raccolta dal telescopio di cima Ekar è troppo limitata per l'osservazione di stelle più deboli.

IquEye è stato progettato proprio per risolvere la maggior parte di queste problematiche e per aumentare le potenzialità degli strumenti di questo tipo.

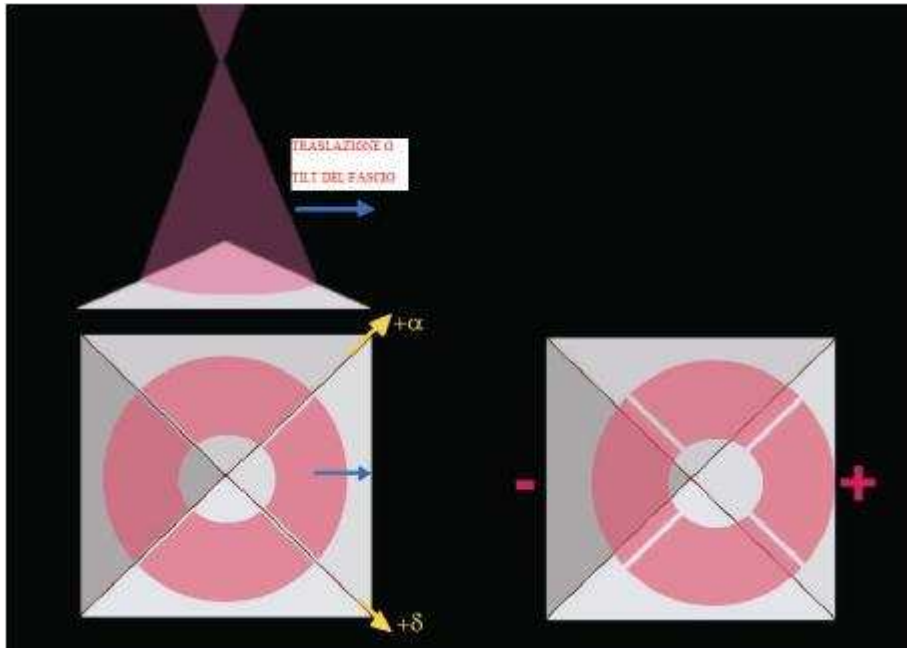


Figura 3 Nel caso di traslazione o tilt del fascio, in questo esempio verso destra, accade che la faccia destra riceverà più radiazione a discapito delle altre.

2.2 Il telescopio NTT di La Silla

Il sito scelto per questa applicazione di IquEye è il telescopio NTT (New Technology Telescope) dell'Osservatorio Europeo Australe situato a 2400 m di altezza in località La Silla, Cile. Questo telescopio è un Richey-Chretien, le cui aberrazioni sferiche e il coma sono corrette perfettamente; i fattori limitanti sono l'astigmatismo e la curvatura di campo. Lo specchio primario è realizzato in Zerodur e ha un diametro di 3580 mm. La qualità ottica del telescopio è assicurata dall'ottica attiva che consiste di 75 attuatori assiali, 3 punti fissi e 24 attuatori laterali. La lunghezza focale è di 38.5 m, il fattore di scala è $f/10.85$, il piano focale è di $187 \mu\text{m} / \text{arcsec}$; il campo di vista (FoV) che riesce a raccogliere è di 25 primi (5 arcsec verranno utilizzati da IquEye) che si traducono 0.935 mm al fuoco del telescopio.



Figura 4 Telescopio NTT di La Silla. Nella foto sono visibili gli attuatori per l'ottica adattiva.

2.3 IquEye: caratteristiche ottiche

Come il suo predecessore AquEye, anche IquEye divide il fascio in quattro parti grazie a una piramide specchiata e poi lo invia ad altrettanti SPAD. Diversamente da AquEye però, è dotato di un riduttore di focale composto da due coppie di lenti che riducono l'immagine fornita dal telescopio di 5'' in cielo ($935 \mu\text{m}$) fino a $290 \mu\text{m}$ con fattore di demagnificazione di $1/3.25$. Le lenti successive riducono ulteriormente il fascio fino ad ottenere uno spot di $80 \mu\text{m}$ di diametro, con ulteriore fattore di demagnificazione di $1/3.6$. Il disegno ottico è mostrato in Fig. 5. La figura mostra le componenti e il percorso che il fascio luminoso deve seguire dall'interfaccia dello strumento col telescopio fino agli SPAD; le componenti verranno illustrate con maggiore dettaglio nella sezione seguente.

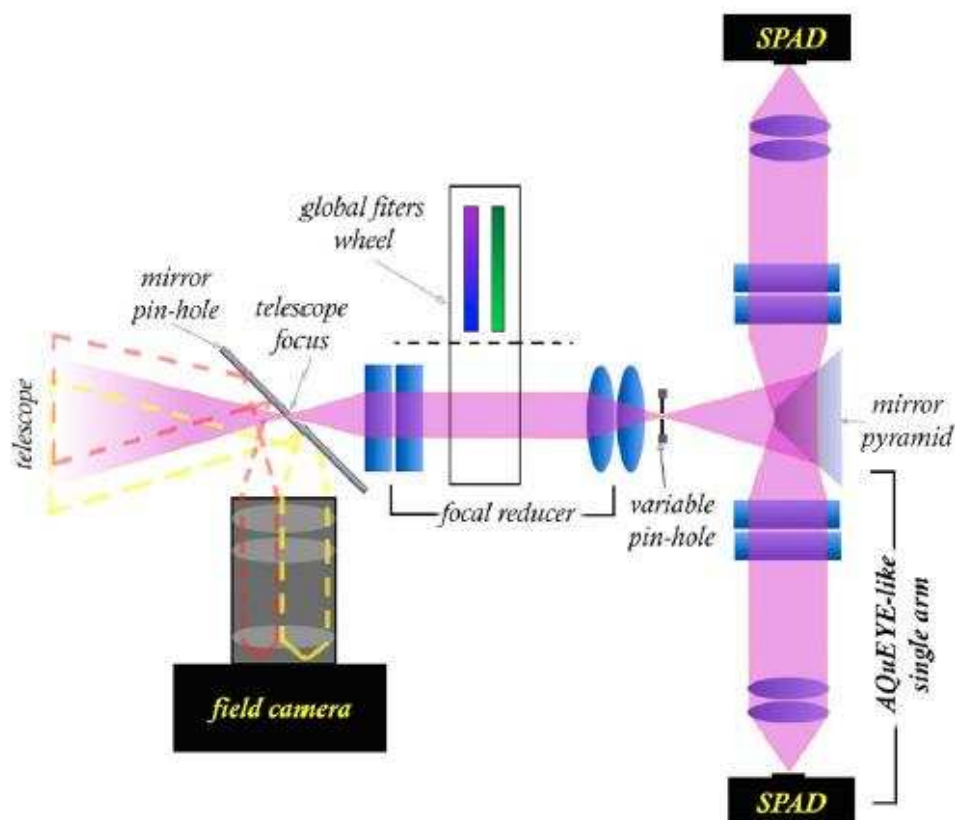


Figura 5 Schema ottico di IquEye.

Le prestazioni ottiche di questo strumento sono state stimate attraverso simulazioni di ray-tracing e risultano eccellenti nell'area attiva circolare di $100\ \mu\text{m}$ tra il blu ($420\ \text{nm}$) e il rosso ($720\ \text{nm}$). In Fig. 6 è mostrato lo spot della sorgente estesa ottenuto considerando sia il centrale che i quattro campi di vista laterali per una sorgente avente diametro di $5\ \text{arcsec}$; in Fig. 7 viene mostrata la corrispondente encircled energy. È evidente la capacità dello strumento di focalizzare più del 90% del flusso incidente sulla parte attiva del rivelatore. Altre perdite del sistema (meno del 15%) sono dovute alla luce dispersa dai bordi della piramide dove la radiazione luminosa viene divisa e alla retro riflessione delle lenti.

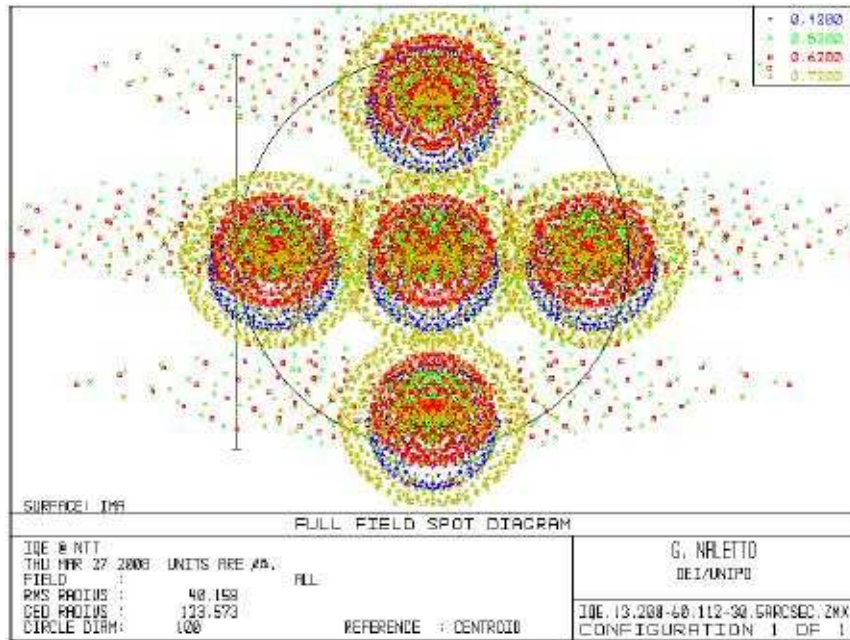


Figura 6 Concentrazione di energia per un fascio di radiazione avente distribuzione spettrale compresa tra i 420 e i 700 nm, per cinque posizioni (al centro e ai bordi) per un campo di vista assunto di 5 arcosec. Il cerchio rappresenta l'area sensibile dello SPAD.

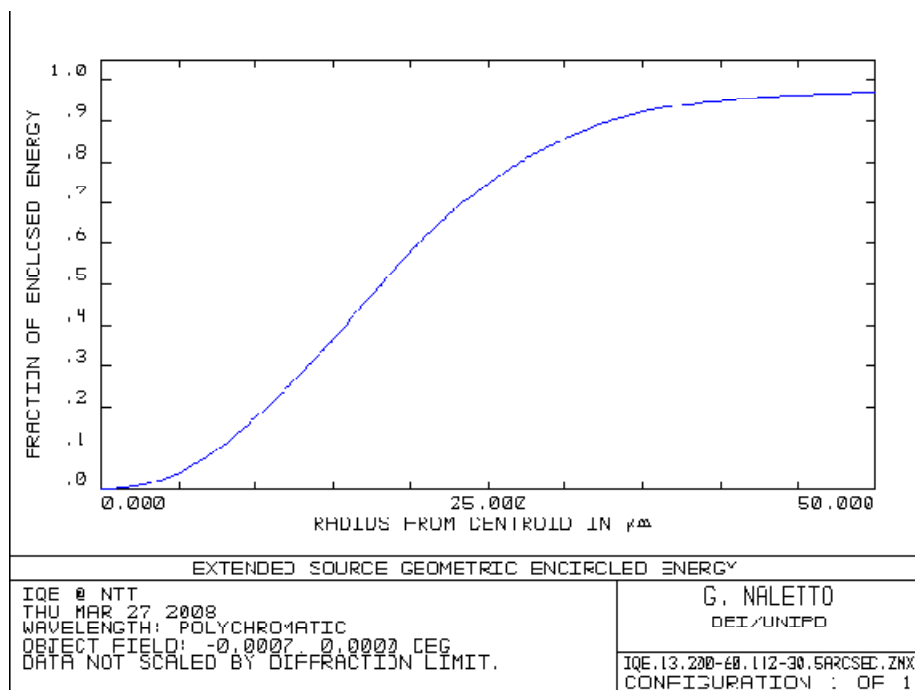


Figura 7 Tracciato dell'encircled energy per sorgente estesa: percentuale dell'energia che cade all'interno di un dato raggio per una sorgente estesa uniformemente illuminata di 5 arcosec. Il limite dei 50 μm corrisponde all'area sensibile degli SPAD.

2.4 IquEye: caratteristiche progettuali

Lo strumento si compone di due corpi assemblati: il primo, di forma semicilindrica, contenente le prime due coppie di lenti e il secondo, di forma ottagonale, che contiene la piramide, i quattro bracci costituiti da altrettanti corpi di lenti, e gli SPAD. Qui di seguito verrà presentata in dettaglio la composizione di tale strumento; si usi come riferimento la Fig. 8 che è un disegno concettuale usato per la progettazione. Come si vede in figura, tutto lo strumento è ancorato ad una flangia circolare progettata appositamente per essere agganciata al telescopio NTT già menzionato. Il fascio dal telescopio focalizza sulla faccia anteriore di uno specchio di campo (visibile in figura all'interno della flangia); tale specchio è posto a 45° rispetto all'asse ottico e serve a riflettere l'immagine della volta celeste osservata dalla camera sottostante; al centro dello specchio è presente un foro svasato (anch'esso con inclinazione a 45°) che permette il passaggio del flusso luminoso. Tutte le lenti attraversate dal fascio sono di tipo commerciale, in particolare si è scelto di utilizzare quelle prodotte dalla Edmund Optics. Dopo lo specchio, la luce viene raccolta dalla prima coppia di lenti, distanziate da uno spessore di 3.6 mm; quest'ultimo serve ad evitare il contatto tra le superfici in vetro ed i relativi problemi ottici connessi all'aderenza tra ottiche. La coppia di lenti, adeguatamente posizionata, provvede a rendere il fascio collimato. Tra la prima e la seconda coppia di lenti sono state posizionate due ruote porta filtri in serie comandate elettronicamente dall'utilizzatore. Le ruote hanno otto posizioni ciascuna; la Tabella 1 riporta le posizioni ed i relativi filtri assegnati per ogni posizione. È ovvio che il sistema delle ruote permette l'utilizzo di un singolo filtro o di una qualsiasi combinazione tra loro.

Dopo aver passato entrambe le ruote porta filtri il fascio viene raccolto da un'altra coppia di lenti, anch'esse intervallate da uno spessore di 3.6 mm. Queste lenti focalizzano il fascio dopo 48.55 mm su uno dei tre pinhole; i pinhole sono montati tutti e tre su di una slitta meccanica, anch'essa comandabile dall'utilizzatore. La slitta (OWIS), visibile in figura 8 colorata in verde è ancorata all'ottagono e contiene i tre pinhole: il più grande è di 500 μm , il secondo è di 300 μm , mentre l'ultimo e più piccolo è di 200 μm di diametro.

Dopodichè il fascio viene suddiviso e riflesso in quattro parti dalla piramide a base quadrata posta nel centro dell'ottagono.

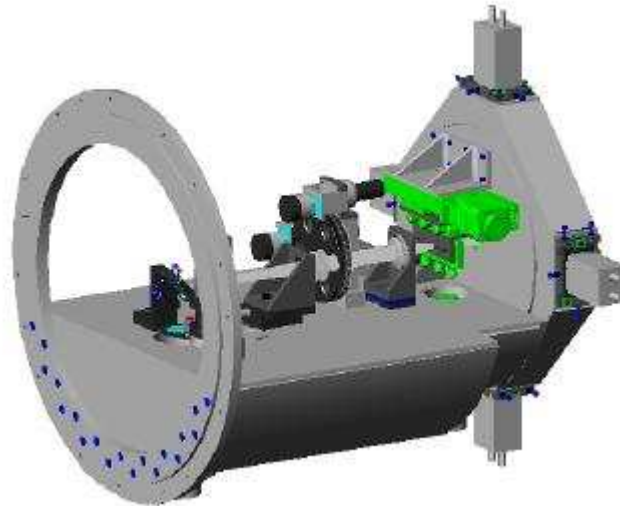


Figura 8 Schema tridimensionale dell'intero corpo di IquEye

Per ognuno dei quattro fasci è stato previsto un percorso ottico così composto: una prima coppia di lenti, (intramezzate anch'esse da uno spessore di 3.6 mm) addette alla collimazione del fascio che viene poi focalizzato da un'altra coppia di lenti, sulla superficie sensibile degli SPAD (anche quest'ultima coppia è intervallata da uno spessore di 3.6 mm).

POS.	RUOTA 1	RUOTA 2
1	Blu	Neutro 1
2	Vis	Neutro 3
3	Rosso	468 nm HE II
4	IR	630 nm O I
5	O3	Polarizzatore visibile asse orizzontale
6	H alfa	Polarizzatore UV asse verticale
7	Otturatore	Neutro 2
8	Vuoto	Vuoto

Tabella 1 Riassunto schematico dei filtri presenti sulle ruote con le relative posizioni.

2.5 IquEye: sistema di acquisizione dati

La parte più innovativa di IquEye consiste nel sistema di acquisizione dati che, accoppiato ai rivelatori ad alta velocità, un oscillatore al rubidio e un ricevitore GPS, permette di assegnare un tempo di arrivo ai fotoni riferito a UTC con una accuratezza finale della rms migliore di 0.5 ns in un'ora di osservazione.

Il compito del sistema acquisizione dati di IquEye (IDAS) è concettualmente molto semplice: deve raccogliere le pulsazioni prodotte da ogni SPAD quando rivela un fotone, assegnare un'etichetta temporale ad ogni evento e immagazzinarle in una memoria esterna. La realizzazione di questo compito non è per nulla semplice a causa dei requisiti estremamente rigidi: è necessario assegnare un tempo ad ogni fotone in arrivo con una accuratezza dell'ordine dei 500 ps per almeno un'ora di osservazione continua, tenendo conto che il conteggio di fotoni è variabile da poche decine di Hertz fino a 8 MHz.

Il punto di partenza per definire IDAS (Fig. 9) è la pulsazione prodotta dagli SPAD al rivelamento di un fotone. Gli SPAD utilizzati hanno due diversi output: uno TTL, che può sostenere conteggi fino a 12 MHz in regime lineare, con una accuratezza temporale di circa 250 ps e un output NIM, con conteggio massimo di circa 2 MHz ma molto più accurato (35 ps). Poiché l'obiettivo con IquEye era di raggiungere una risoluzione temporale di 100 ps, si è adottato l'output NIM come baseline mentre il TTL può essere impiegato in parallelo come parametro di controllo.

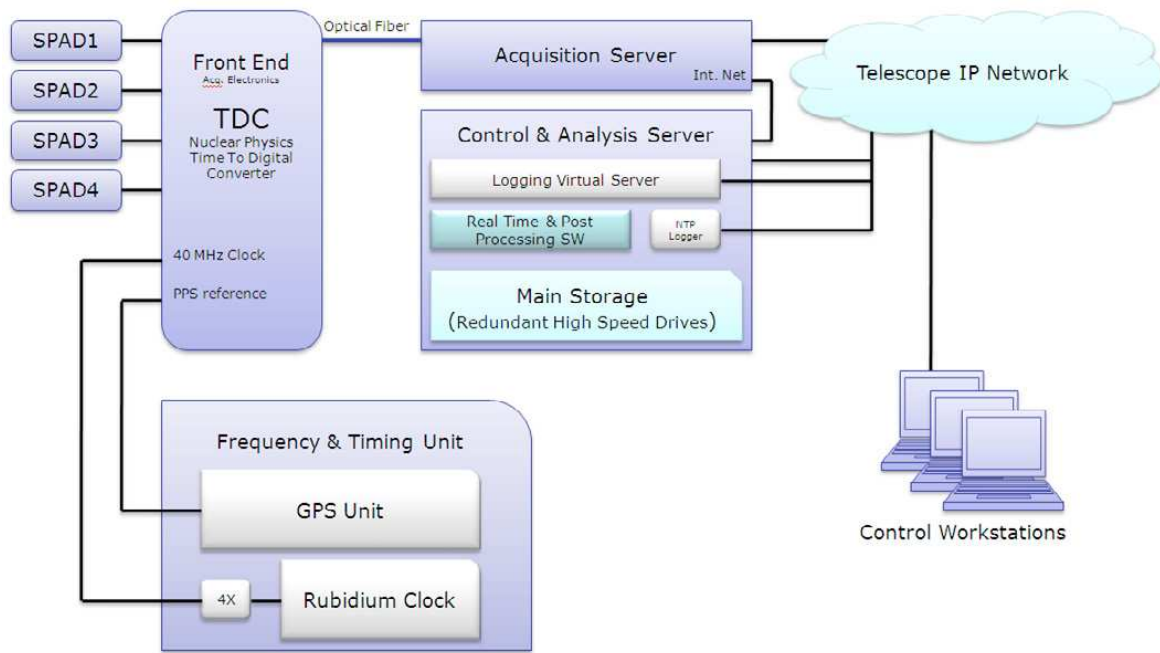


Figura 9 Schema concettuale del sistema di acquisizione e temporizzazione di IquEye.

Ognuno dei 4 segnali NIM viene trasferito dagli SPAD, tramite cavi coassiali calibrati in lunghezza per assicurare lo stesso ritardo elettrico, ad uno dei canali di input di un convertitore tempo-digitale (TDC). Tale convertitore è nominalmente in grado di assegnare un'etichetta temporale agli impulsi di voltaggio che riceve come input con una risoluzione di 24.4 ps; un campionamento così alto è dovuto all'uso di un oscillatore interno a 40 MHz, la cui frequenza viene moltiplicata per 1024 grazie a un phase locked loop (PLL) e a un delay locked loop (DLL), arrivando così a un clock di 40 GHz. Sfortunatamente la qualità dello strumento è insufficiente per soddisfare i rigidi vincoli di stabilità richiesti dall'applicazione; si è reso necessario ottenere la stabilità su tempi brevi tipica di un oscillatore al quarzo e per il lungo termine la stabilità data da un riferimento temporale esterno. La soluzione compatibile con le risorse economiche a disposizione consiste nell'uso combinato di un oscillatore al rubidio, un ricevitore GPS e un algoritmo post acquisizione.

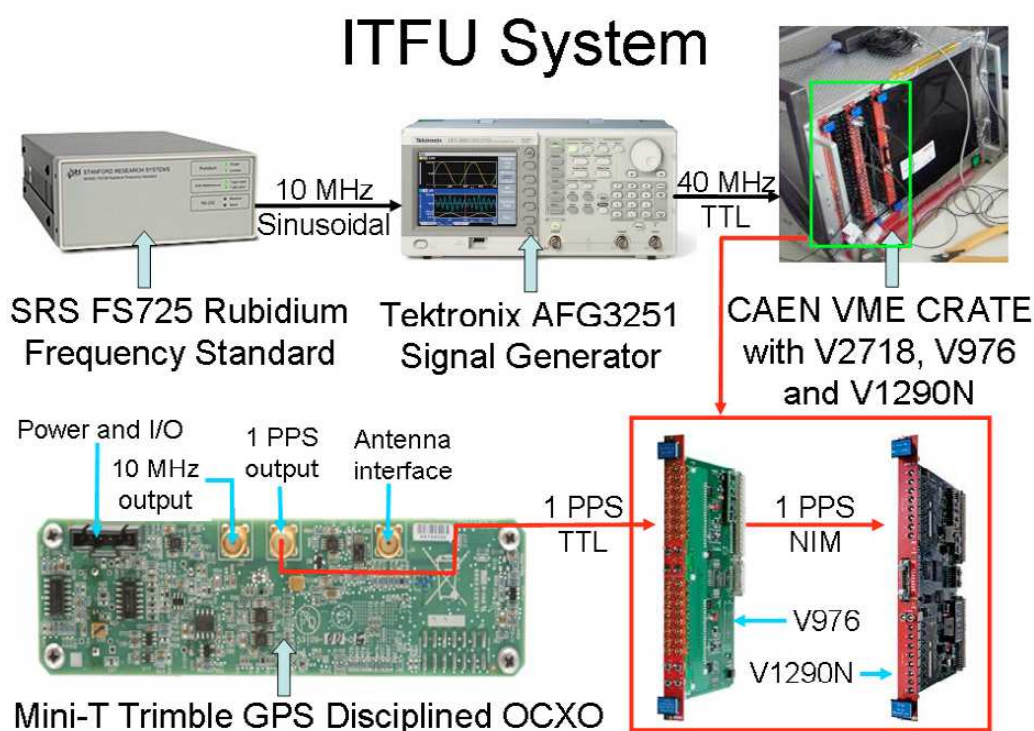


Figura 10 Diagramma a blocchi dell'unità tempo-frequenza di IquEye

L'unità tempo-frequenza di IquEye (ITFU), in figura 10, è concepita per correggere la deriva della frequenza di riferimento fornita dall'oscillatore al rubidio tramite un algoritmo che utilizza il segnale PPS (Pulse Per Second) ricevuto da un ricevitore GPS mini-T Trimble. Il GPS provvede anche alla sincronizzazione con UTC.

L'oscillatore al rubidio produce un segnale sinusoidale a 10 MHz che viene poi convertito in un segnale TTL a 40 MHz per mezzo di una moltiplicazione in frequenza da un generatore di impulso e da un generatore di segnale. Il segnale TTL è l'input per IDAS. L'orologio atomico al rubidio è un ottimo trade off tra performance e costi: assicura una buona stabilità per un tempo non troppo lungo (inferiore a 10^4 secondi), mantenendo contenuti i costi. Test effettuati presso il *Laboratorio Tempo & Frequenza* dell'Osservatorio Astronomico di Cagliari (OAC), hanno evidenziato che il rubidio accumula una deriva di fase di circa 65 ns dopo un'ora e 160 ns dopo due. Comunque, rimuovendo questa deriva lineare, rimane solamente un errore stocastico residuo, il cui valore rms è inferiore a 50 ps per una durata di oltre 1000 secondi.

Sommando l'errore temporale degli SPAD (35 ps), l'errore della catena elettronica (50 ps) e il residuo di fase (50 ps in un'ora) alla discretizzazione interna del campionamento (24.4 ps), si ottiene un errore totale di circa 100 ps, che soddisfa le specifiche richieste per lo strumento. Questo è il livello di accuratezza temporale *relativa* che possiamo raggiungere con IquEye per osservazioni di un'ora.

Per rimuovere la deriva lineare ed ottenere la sincronizzazione UTC, IquEye utilizza il segnale PPS del ricevitore GPS. Algoritmi post acquisizione eseguono una stima dell'errore dell'oscillatore al rubidio e provvedono alla sua rimozione. Per determinare l'inizio dell'acquisizione viene considerato il primo PPS trovato nella stringa di dati, la cui incertezza è di circa ± 25 ns (valore associato a un ricevitore GPS di buona qualità). Tuttavia, grazie all'interpolazione della deriva lineare, l'errore decresce col numero di PPS acquisiti: si può facilmente dimostrare che dopo un'ora di osservazione l'errore è calato fino a 0.4 ns. Considerando l'errore relativo di 100 ps si giunge ad avere un errore *assoluto* rispetto a UTC di circa 0.5 ns.

Una limitazione del convertitore tempo-digitale (TDC) è l'impiego di 21 bit per rappresentare i valori temporali in uscita. Usando come unità 24.4 ps, è possibile avere intervalli di tempo privi di ambiguità di lunghezza massima $\Delta t = 51.2 \mu s$ ed è necessario assegnare ad ogni tempo maggiore di questo intervallo una fase iniziale $N\Delta t$, con N da determinare. Per realizzare questo processo, definito "de-rollover", viene dato come input al TDC un impulso di riferimento a 20 kHz, che permette di determinare il valore di N.

I dati in uscita dal TDC vengono immagazzinati in un server dedicato e poi spediti mediante fibra ottica al server di acquisizione con velocità massima di circa 60 Mb/s. Il trasferimento dati si è rivelato essere il collo di bottiglia poichè limita il conteggio massimo del sistema; tuttavia si prevede di integrare una memoria di acquisizione nel blocco TDC per superare tale ostacolo. L'interfaccia utente, implementata su un secondo server, è stata realizzata in codice Java: essa permette di controllare l'acquisizione e la memorizzazione dei dati, il controllo di ogni meccanismo e strumento del sistema, di eseguire alcune veloci analisi statistiche e di memorizzare immagini del campo di vista attorno all'oggetto selezionato per l'osservazione da utilizzare per un'ulteriore analisi a posteriori delle condizioni del cielo.

In tabella 2 si può vedere un riassunto delle caratteristiche principali e performance di IquEye.

Sensibilità del sistema	Conteggio fotoni
Accuratezza temporale relativa	100 ps (per un'ora di osservazione)
Accuratezza temporale assoluta	500 ps (per un'ora di osservazione)
Dark count rate	< 100 Hz
Conteggio massimo	8 MHz
Range dinamico	> 40 Db
Magnitudine limite	$m_v = 24$ con tempo di esposizione di 2 ore e $\frac{S}{N} = 10$
Campo di vista effettivo	Selezionabile tra 3.5 5.2 e 6.1 arcsec
Intervallo spettrale operativo	$\Delta\lambda = [350, 925]$ nm
Efficienza totale del sistema	33% (picco a 550 nm) 18% (media su $\Delta\lambda = [350, 925]$ nm)

Tabella 2 Caratteristiche principali e performance di IquEye applicato a NTT

CAPITOLO 3

Pulsar

3.1 Pulsating stars

Una stella a neutroni è il prodotto della morte di una stella con massa sufficientemente grande (da 4 a 8 volte la massa del sole); una volta finita la combustione del materiale nel nucleo si verifica una esplosione da supernova e la regione centrale della stella subisce un collasso gravitazionale. La forza di gravità è talmente elevata che protoni ed elettroni si fondono e formano neutroni, che compongono quasi interamente la stella e dai quali prende il nome. Il collasso a questo punto si interrompe, coerentemente con il principio di esclusione di Pauli secondo cui due neutroni (o due fermioni in generale) non possono occupare lo stesso luogo e stato quantico nello stesso momento.

La massa di queste stelle è pari a 1.4 – 2.1 volte il sole e il loro raggio è di circa una decina di km, di conseguenza la loro gravità superficiale è molto elevata, circa $2 * 10^{11}$ volte quella terrestre. Durante il collasso, il nucleo della stella mantiene buona parte del suo momento angolare. Questo fenomeno, combinato alla riduzione del raggio (e quindi alla diminuzione del momento di inerzia), fa sì che ogni stella di neutroni abbia alla nascita un'elevata velocità di rotazione.

Una pulsar (contrazione di pulsating star) è un particolare tipo di stella di neutroni che emette, durante la rotazione, un fascio di radiazioni lungo l'asse magnetico (Fig. 11).

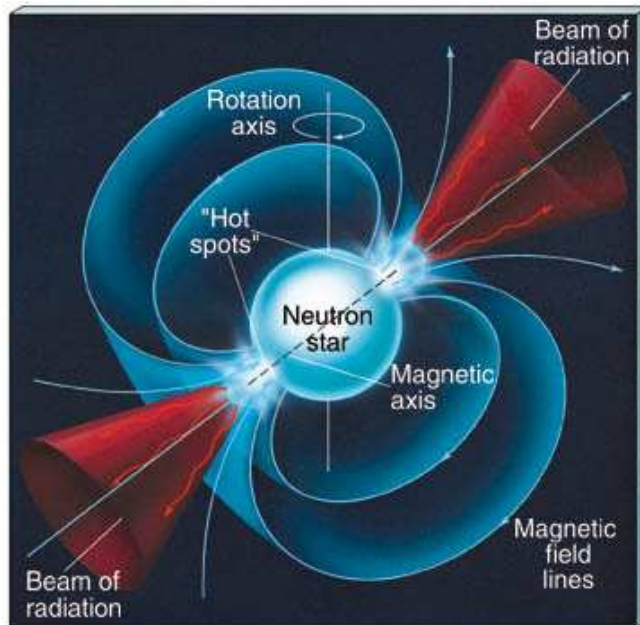


Figura 11 Schema di una stella a neutroni: sono visibili l'asse di rotazione e l'asse magnetico dai cui poli hanno origine i coni di radiazioni elettromagnetiche

Il tipo di radiazioni è variabile a seconda della pulsar: raggi x, raggi gamma, onde radio, onde nel campo della luce visibile o una combinazione di questi (una pulsar che emette luce visibile emette sempre anche onde radio). Nel nostro lavoro si sono considerate solo pulsar che emettono luce visibile in quanto IquEye è sensibile solo in questa regione spettrale.

I poli magnetici determinano la direzione del fascio, ma l'asse magnetico non coincide necessariamente con l'asse di rotazione della stella. Tale disallineamento è la causa della natura a "pulsazione" di questa categoria di stelle: essendo molto lontane e deboli, sono visibili solo quando la terra viene investita da uno o entrambi i coni di radiazioni, fenomeno che capita una volta per ogni rotazione.

Il fascio di radiazioni ha origine dall'energia di rotazione e dal forte campo magnetico della stella: si forma così un campo elettrico che causa un'accelerazione di protoni ed elettroni presenti sulla superficie della stella fino al crearsi di un raggio elettromagnetico uscente dai poli del campo magnetico.

A causa dell'energia irradiata, una stella a neutroni rallenta ad ogni rotazione di una frazione di secondo (tra 10^{-10} e 10^{-21} secondi); si ritiene che al raggiungimento di una

velocità di rotazione sufficientemente bassa, l'emissione dei coni di radiazioni venga meno per mancanza di energia. Il periodo delle pulsar individuate fino ad ora varia da 1.4 ms a 8 secondi; considerando la decelerazione media si pensa che la durata della vita di una pulsar sia di qualche decina di milioni di anni.

3.2 Curva di luce

In astronomia, una curva di luce (*light curve*) è un grafico dell'intensità luminosa di un corpo celeste o di una regione di spazio, in funzione del tempo. Le curve di luce possono essere aperiodiche, nel caso riguardino una nova, supernova o una stella variabile cataclismica, oppure periodiche come nel caso di variabili cefeidi o pulsar. Lo studio di una curva di luce può portare una considerevole quantità di informazioni riguardanti il processo fisico che la genera. Esse vengono utilizzate anche per avere una stima del periodo di rotazione di pianeti o lune, analizzando la separazione temporale dei picchi di luce.

Nello studio delle pulsar, l'obiettivo è arrivare ad una curva di luce che metta in evidenza la luce della stella dal rumore, proveniente dalla nebulosa circostante la stella, dal fondo del cielo e nel caso di IquEye anche dal rumore di buio degli SPAD. In Figura 12 si può vedere un esempio di light curve su due periodi di rotazione di una pulsar molto facile da analizzare (una pulsar nella nebulosa del Granchio); i picchi nel conteggio dei fotoni corrispondono alla luce appartenente ai due coni della pulsar. Arrivare a una curva in cui la pulsar è distinguibile dal rumore significa che il periodo di rotazione su cui si basa il *folding* dei dati è corretto, come verrà spiegato dettagliatamente nel capitolo 4.

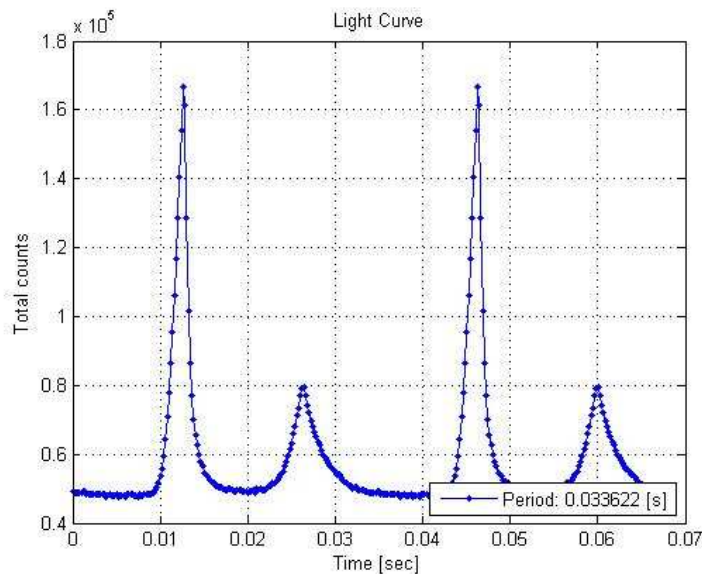


Figura 12 Light curve generata dalla pulsar Crab

3.3 Oggetto dell'osservazione

Tra gli oggetti analizzati con IquEye (stelle brillanti, transiti planetari, variabili cataclismiche), si è dedicata maggiore attenzione all'osservazione di tre pulsar specifiche. La particolarità di queste tre stelle a neutroni sta nel fatto che sono le uniche pulsar, monitorabili con il telescopio NTT, che emettono radiazioni nel campo della luce visibile e che quindi si prestano all'impiego di IquEye.

La prima pulsar si chiama PSR J1921+2153 o più semplicemente Pulsar della Crab, poichè situata nella nebulosa del Granchio (Fig. 13), ed è uno dei corpi celesti più studiati nel campo dell'astronomia. Essa è la più brillante pulsar ottica ed è stata la prima ad essere riconosciuta come sorgente di pulsazioni nella banda visibile¹, anche se le radiazioni che emette comprendono onde radio, potenti raggi X e raggi gamma. Lo scopo delle analisi su questa stella è di studiare la stabilità della forma della pulsazione ottica (*curva di luce*) e di testare le performance temporali dello strumento determinandone il periodo di rotazione e la derivata prima del periodo. Per tale obiettivo la Crab è l'ideale poichè le sue caratteristiche temporali e la sua curva di luce sono note con grande precisione, essendo un oggetto che è stato studiato da molti team di ricerca tramite il monitoraggio delle sue onde radio

¹ Cocke et al. 1969; Lynds et al. 1969

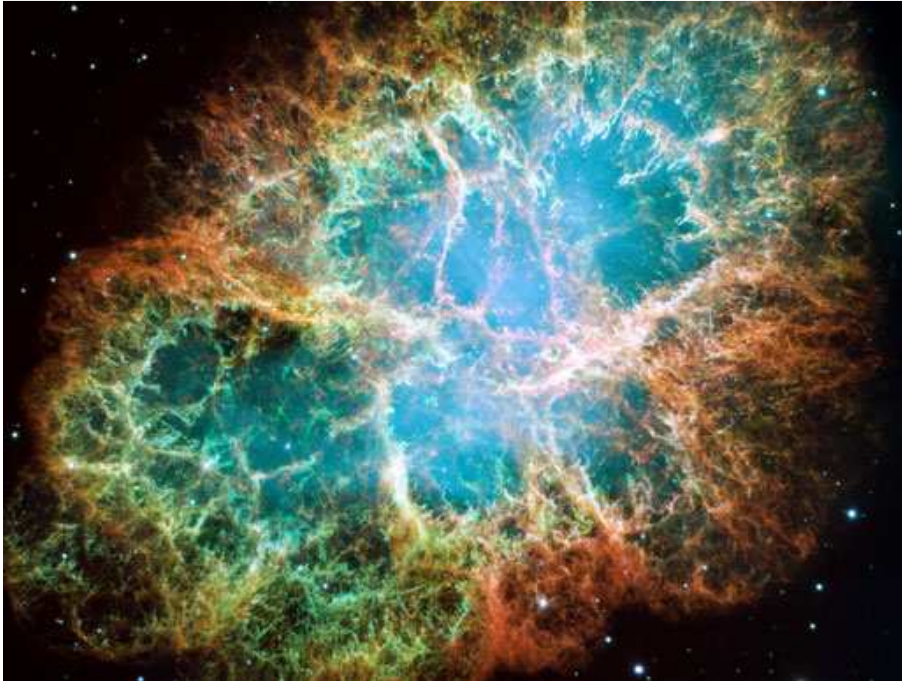


Figura 13 Nebulosa del Granchio, luce visibile

Un'altra pulsar analizzata è PSR B0540-69, la seconda più brillante nel campo del visibile, situata nella Large Magellanic Cloud (Pulsar LMC). Essa è stata osservata negli ultimi anni sia con telescopi di terra che con il telescopio spaziale Hubble, tuttavia solo in poche occasioni è stato possibile distinguere una curva di luce corretta e pubblicare quindi dei risultati².

Diversamente dalla Crab, la luce di questa stella è immersa in un forte rumore che rende molto difficile individuarne il periodo di rotazione esatto al momento dell'osservazione e arrivare a “vedere” la stella, ovvero ottenere una light curve in cui la luce della stella si distingua dal rumore. Il fatto che fino ad ora sia stato possibile vedere la stella solo con pochi telescopi, è stato uno stimolo a mettere alla prova la qualità di IquEye. Si è arrivati infine ad avere una buona light curve per Pulsar LMC [10] utilizzando metodi di analisi “convenzionali”, ovvero senza dover ricorrere a strumenti di denoising.

² Middleditch et al. 1987, Boyd et al. 2005, Mignani et al. (1998), Deeter et al. (1999), Gouiffes et al. (1992), Ulmer et al. (1997).

Infine si è puntato il telescopio su PSR 0833-45, una pulsar meglio conosciuta con il nome di Vela. L'unica rivelazione di questa stella risale al 1979 ad opera di Wallace et al. per mezzo di un sistema a doppio fotometro montato sul telescopio anglo-australiano situato a Coonabarabran (New South Wales). In questo caso il SNR è estremamente basso, il segnale appare totalmente coperto dal rumore e negli anni successivi nessuno è riuscito ad ottenere una curva di luce che evidenziasse la pulsazione della stella.

La pulsar Vela rappresenta la vera sfida per IquEye ed è il motivo per cui è nato questo lavoro; data la grande quantità di rumore, è necessario ricorrere a metodi di denoising per ripulire il segnale e trovare il corretto periodo di rotazione, parametro fondamentale per arrivare alla light curve. La scelta di utilizzare la KLT ha origine da [1], in cui la trasformata si è rivelata utile per individuare un segnale sommerso da rumore, tuttavia poichè la tipologia di dati e il metodo di applicazione sono completamente diversi dal nostro caso, l'articolo ha dato solo un'idea di partenza che si è dovuta sviluppare con altra documentazione.

CAPITOLO 4

Analisi dati

4.1 Il segnale ed il suo preprocessing

Prima di iniziare qualunque analisi alla ricerca del periodo e della light curve, i dati in uscita da IquEye devono essere sottoposti a un'ulteriore correzione. I tempi di arrivo dei fotoni vengono riferiti al baricentro del sistema solare, cioè il loro valore viene modificato in modo che corrisponda al valore che si sarebbe ottenuto posizionando il telescopio nel baricentro del sistema solare. Questo preprocessing corregge l'errore nella misurazione dei tempi causato dal moto di rotazione terrestre: se l'osservazione è prolungata nel tempo, tra l'inizio e la fine dell'acquisizione dati la terra avrà compiuto una parte della sua rotazione. La correzione dei tempi viene effettuata usando l'ultima release del software Tempo2 (Hobbs, Edwards & Manchester 2006), creato appositamente per l'analisi dei fenomeni legati alle pulsar.

I dati trattati con Tempo2 sono in formato MJD (Modified Julian Date) e sono suddivisi in un centinaio di file contenenti sia valori numerici rappresentanti i tempi, sia porzioni di testo con informazioni relative allo stato dello strumento, la data e le condizioni di acquisizione. Per questo motivo sono stati sottoposti a un altro preprocessing allo scopo di unire i file, estrarne i tempi e creare un unico array in cui i valori non sono più in MJD, bensì in distanza temporale dal primo elemento dell'array che diventa lo 0, l'inizio dell'acquisizione.

Il data-frame risultante contiene tutti i fotoni rivelati dallo strumento IquEye, esso comprende quindi i fotoni provenienti dalla pulsar oggetto dell'osservazione, ma anche una grande quantità di fotoni che costituiscono il rumore del segnale. Le sorgenti di rumore identificate sono tre.

La prima è costituita dai conteggi di buio (*dark count*) degli SPAD: anche in condizione di assenza di luce i quattro SPAD di IquEye danno un certo numero di falsi conteggi. Tuttavia

il *dark count rate* è relativamente basso (meno di 50 conteggi al secondo per ogni SPAD) e per questo motivo si può considerare la fonte di rumore con peso minore.

Occorre poi considerare il fondo del cielo: rumore introdotto dalla luminosità del cielo, il che comprende sia la luce all'interno dell'atmosfera che la luce nello spazio tra la Terra e l'oggetto che si sta osservando.

L'ultima fonte di rumore è dovuta alla natura stessa della pulsar. Ogni pulsar si trova all'interno di una nebulosa, la quale non è altro che il residuo dell'esplosione della supernova che ha dato origine alla pulsar stessa; di conseguenza la stella che viene osservata è immersa nella radiazione luminosa della nebulosa circostante. Risulta impossibile, coi mezzi a disposizione, isolare completamente l'immagine della pulsar dalla nebulosa per due motivi.

Si può assumere che la pulsar sia una sorgente luminosa puntiforme posta all'infinito; per ottenere l'immagine della stella non è possibile usare un campo visivo puntiforme perchè, a causa delle aberrazioni derivanti dalle lenti dello strumento e dall'atmosfera, la sorgente si traduce sempre in un'immagine non puntiforme. Di conseguenza per includere nello *spot* tutta l'energia della pulsar è necessario osservare una porzione di cielo di 2-3 arcsec, ma in questo modo si riceve anche l'immagine parziale di altre sorgenti puntiformi poste vicino alla pulsar. Inoltre occorre considerare che il *field of view* (fov) dello strumento è 5 arcsec, quindi puntando lo strumento sulla pulsar si ricevono i fotoni provenienti dalla stella e altri fotoni originati dalla porzione di nebulosa attorno alla pulsar che costituiscono il rumore.

4.2 Analisi di Fourier

Il primo e più importante passo per avere una curva di luce corretta è individuare il periodo della pulsar in esame, con una risoluzione temporale di almeno 10^{-9} s. Lo strumento più diffuso e di facile utilizzo per individuare la periodicità di un segnale è certamente la Trasformata di Fourier, ma per utilizzarla è necessario convertire i dati da un vettore contenente una serie di istanti temporali, in un segnale nel dominio del tempo.

Per “binnare” i frame-dati è stato deciso, dopo qualche test, di utilizzare dei *time bin* pari a circa 1/20 del periodo di rotazione delle stelle. Considerando che i periodi di rotazione delle tre stelle sono valori compresi tra i 30 ms e i 90 ms, i *time bin* utilizzati variano da 1 a 3 ms. Nel caso della Pulsar Crab, il cui periodo è all’incirca 33 ms, si è costruito il segnale utilizzando *time bin* di 1 ms contenenti il numero di fotoni rivelati dagli SPAD ad ogni millisecondo per tutta la durata dell’osservazione.

Una volta ottenuto un segnale idoneo per l’analisi di Fourier, è stato impiegato l’algoritmo standard FFT (Fast Fourier Transform) sugli array di dati delle pulsar ed è stata subito evidente la grande differenza tra le tre stelle, come si può vedere in Fig. 14, in cui è raffigurato lo spettro ricavato da tre osservazioni considerate particolarmente buone per condizioni atmosferiche.

Nel caso della pulsar Crab (Fig. 14.a) il rumore è trascurabile, infatti le componenti in frequenza si vedono nitidamente analizzando i dati corrispondenti a 30 minuti di osservazione binnati a 1 ms; la componente di interesse tuttavia non è la più grande in ampiezza, posizionata all’incirca a 59 Hz: tale componente è dovuta alla presenza di due picchi di segnale per ogni periodo di rotazione. La terra viene investita dai fotoni provenienti da entrambi i poli magnetici della Crab, quindi ad ogni rotazione i picchi di fotoni sono due; per questo motivo il periodo di rotazione corrisponde alla componente con una frequenza di 29.74319458 Hz (pari a circa la metà della maggiore) che equivale a un periodo di 0.0336211363 s.

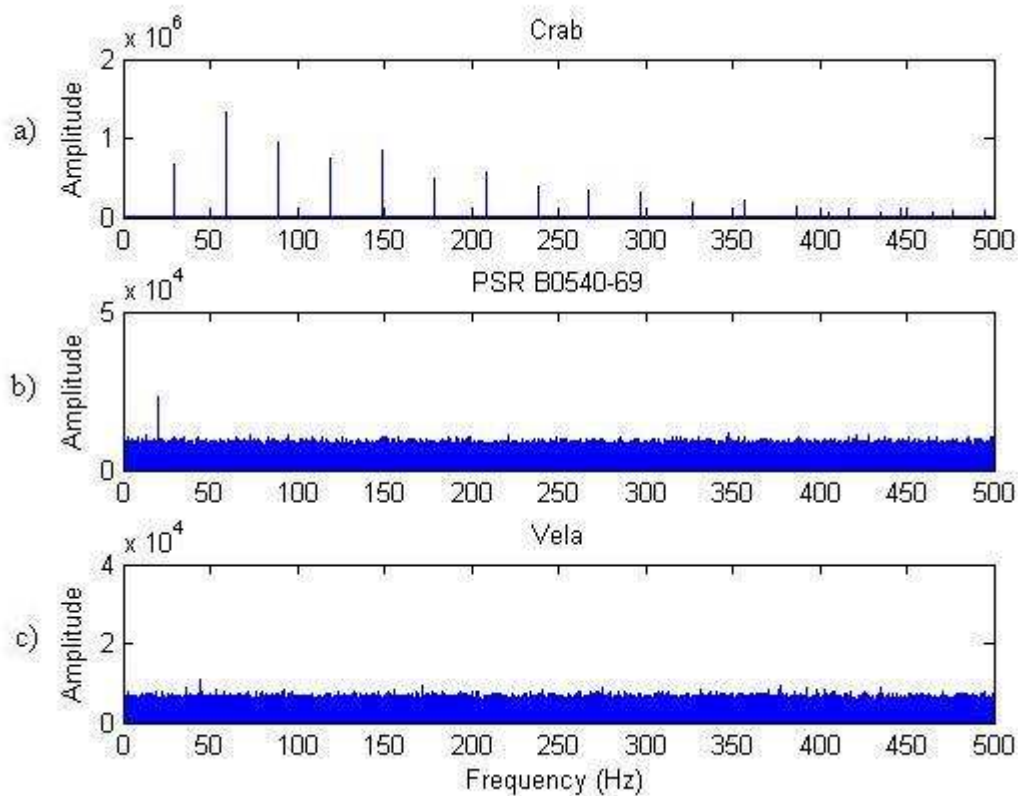


Figura 14 Spettro della pulsar Crab (a), PSR B0540-69 (b) e Vela (c).

Diversamente dal primo caso, nello spettro di PSR B0540-69 (Fig. 14.b) in cui si sono analizzati i dati derivanti da 60 minuti di osservazione mantenendo a 1 ms la dimensione dei time bin, la presenza di un forte rumore è evidente. È comunque visibile un picco nello spettro a 19.75 Hz, che corrisponde a 0.050632911 s, un valore molto simile ai periodi individuati in precedenza da altri gruppi di ricerca.

Infine in Fig.14.c si può vedere lo spettro della pulsar Vela per 60 minuti di osservazione con condizioni atmosferiche ottimali: come già accennato in precedenza, il forte rumore copre interamente il segnale ed è infatti impossibile ricavare il periodo di rotazione dallo spettro. Solo una delle ultime osservazioni, durata 180 minuti, ha permesso di individuare qualcosa dallo spettro del segnale (Fig.15). I time bin sono di 4 ms dato che, stando all'unico articolo esistente in letteratura [4], il periodo della pulsar Vela è molto maggiore delle altre, ed è possibile individuare un picco nello spettro a 11.1899 Hz.

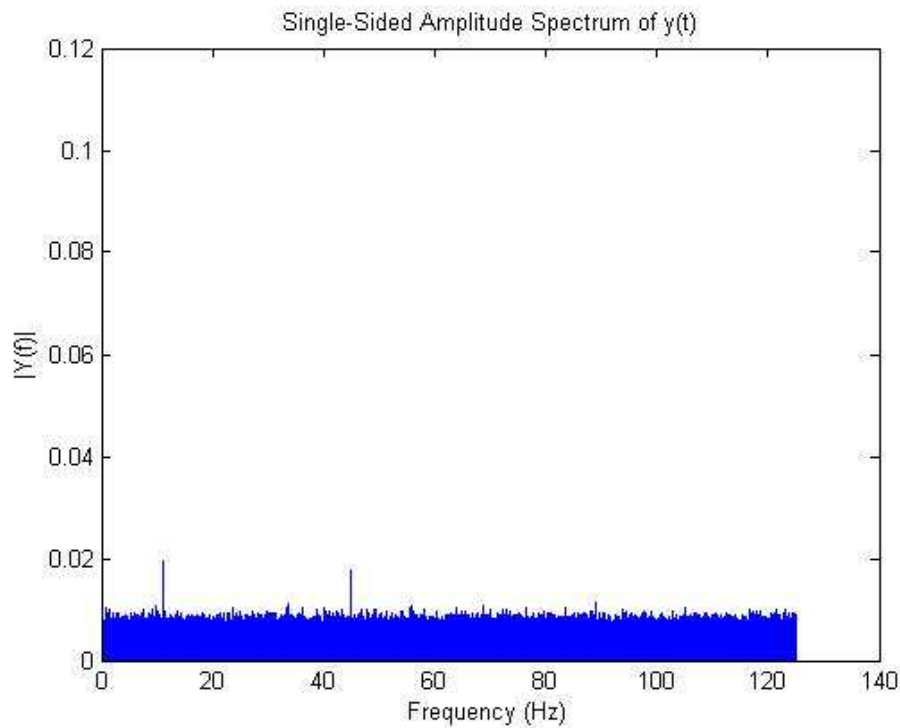


Figura 15 Spettro per 180 minuti di dati della pulsar Vela.

L'aver individuato, tramite Fourier, un periodo di rotazione per ognuna delle pulsar non è sufficiente per ottenere una curva di luce corretta. Come spiegato nel capitolo 3, una stella a neutroni perde velocità ad ogni rotazione, di conseguenza è necessario calcolare il periodo di rotazione ad ogni acquisizione dati; bisogna anche considerare che in caso di acquisizioni prolungate (oltre 30 minuti), la variazione nella velocità di rotazione influenza sia la trasformata di Fourier, sia la costruzione della light curve stessa, almeno nel caso di periodi brevi, qual è quello della Crab.

4.3 Folding dei dati per Crab pulsar

Una volta trovato un periodo della pulsar in esame considerato corretto, si procede al *folding* dei dati: il vettore viene spezzato in molti sottovettori la cui lunghezza copre un tempo pari al periodo e tali sottovettori vengono sommati tra di loro in modo da evidenziare i picchi di segnale. Essendo Crab la pulsar più monitorata, sono disponibili in rete molte informazioni, tra cui la derivata del periodo di rotazione che viene aggiornata ad ogni mese [15]; l'algoritmo tiene conto anche di questa informazione per ottenere la curva di luce migliore possibile.

Gli input dell'algoritmo sono l'array (non binnato) di dati riguardanti la pulsar Crab, in cui gli elementi sono una successione di istanti temporali, il periodo da utilizzare per il folding trovato con la trasformata di Fourier e la derivata del periodo. L'array viene diviso in N sotto-array (V_1, V_2, \dots, V_N), dove N è un valore arbitrario deciso in base alla durata dell'osservazione ed alla quantità di rumore presente; nel caso della Crab i dati coprono 30 minuti di osservazione e il rumore è basso, quindi dopo alcuni test si è scelto di utilizzare $N = 100$. Poichè ogni valore dell'array corrisponde alla distanza temporale dall'inizio dell'acquisizione, è possibile ottenere per mezzo della derivata e del periodo iniziale, una serie di N periodi (p_1, p_2, \dots, p_{100}), calcolati all'istante iniziale di ogni sotto-array. Eseguire un folding composto da una serie di sotto-folding permette di aumentare la precisione della curva di luce; come già detto, la variazione del periodo nel tempo causa un errore che aumenta con il passare del tempo. Ricalcolando il periodo a intervalli regolari (in questo caso ogni $30 * 60/100 = 18$ s) si minimizza l'errore.

Ottenuti i periodi, si passa al calcolo della fase di tutti gli elementi che vengono memorizzati sotto array V'_i , trovando il modulo di ogni valore per il periodo corrispondente all'array di appartenenza; così facendo i dati nell'array V'_i saranno tutti valori di fase normalizzata al periodo p_i e quindi compresi tra 0 e p_i . Si sceglie poi il tempo di binning, che incide sul numero di punti che generano la curva in output all'algoritmo; anche questo valore è arbitrario ed è consigliabile un valore tanto più grande quanto maggiore è il rumore del segnale. Si è scelto di usare inizialmente $dt = 2 * 10^{-4}$ s poichè il rapporto segnale rumore per la Crab è alto; questo valore che viene poi modificato affinché sia un

divisore del periodo. Avendo $N = 100$ periodi differenti, avremo anche N valori di time bin $(dt_1, dt_2, \dots, dt_N)$, uno per ognuno dei sotto-array, il cui valore è:

$$dt_i = p_i / \text{round}\left(\frac{p_i}{dt}\right)$$

La differenza tra i p_i è minima; per questo motivo il valore $k = \text{round}(p_i/dt)$ è lo stesso per tutti i periodi e corrisponde al numero di intervalli temporali $(\Delta t_{i1}, \Delta t_{i2}, \dots, \Delta t_{ik})$ di lunghezza dt_i in cui viene diviso ogni periodo p_i , inoltre k è anche pari al numero di punti che costituiscono la curva di luce in output.

Per ognuno dei sotto-array V_i si procede alla divisione dei fotoni secondo il tempo di arrivo allo strumento, usando come time bin i Δt_{ij} e se ne esegue il conteggio, arrivando ad ottenere N vettori L_i di k elementi che non contengono istanti temporali, bensì il numero di fotoni rivelati da IquEye suddivisi secondo i time bin. La somma degli N vettori L_1, \dots, L_N , componente per componente, è il vettore L che corrisponde alla light curve.

Inserendo come parametri un periodo iniziale uguale a quello ottenuto tramite la trasformata di Fourier $p_1 = 0.0336211363$ s e un time bin $dt = 2 * 10^{-4}$ s, la light curve risultante non è corretta (Figura 16), poichè non sono visibili i due picchi nel segnale caratteristici della Crab pulsar.

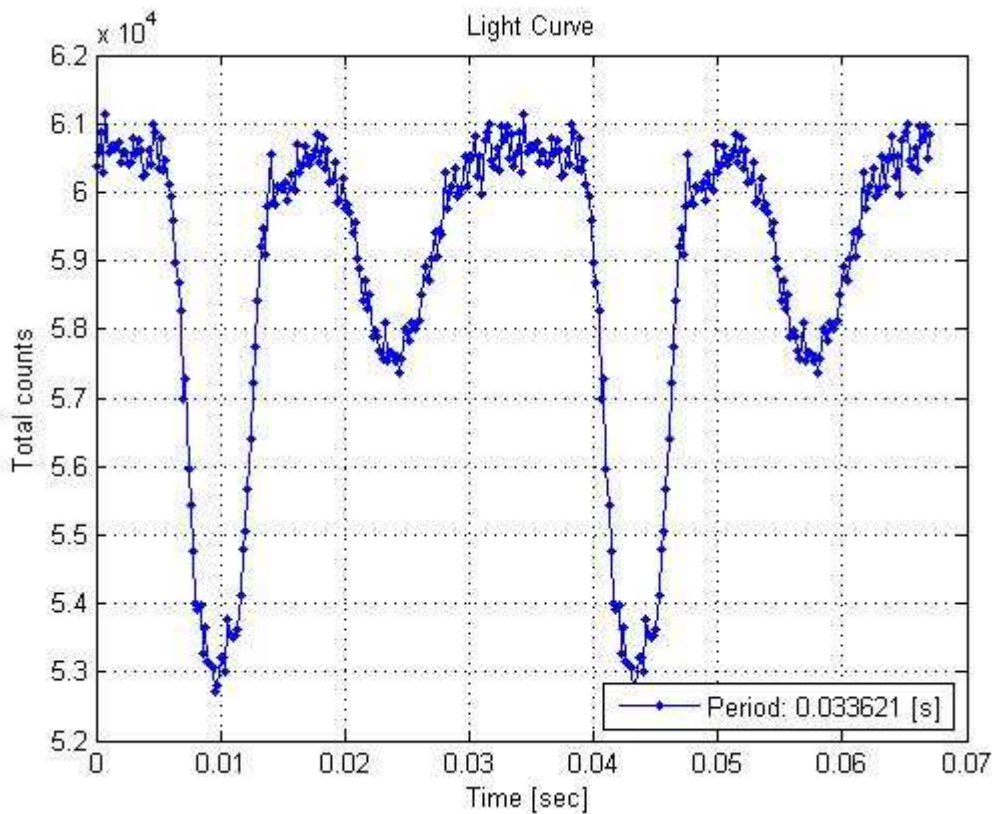


Figura 16 Light curve errata per pulsar Crab, due periodi visualizzati

Questo risultato indica che il periodo trovato con la trasformata di Fourier non è esatto, nonostante il segnale proveniente dalla Crab pulsar sia il più pulito tra quelli analizzati in questo lavoro. Per meglio studiare la natura della light curve si è deciso di modificare l'algoritmo in modo che nell'output venga inclusa una matrice costruita con i vettori L_i , in cui ogni riga della matrice è costituita dal vettore di indice corrispondente. La matrice risultante è quindi di dimensioni $N \times k$ e viene detta *waterfall diagram* poichè è una cascata di light curve costruite su brevi periodi. Un grafico cromatico di questa matrice (Figura 17) permette di individuare eventuali errori commessi nella costruzione della light curve corrispondente all'intera osservazione.

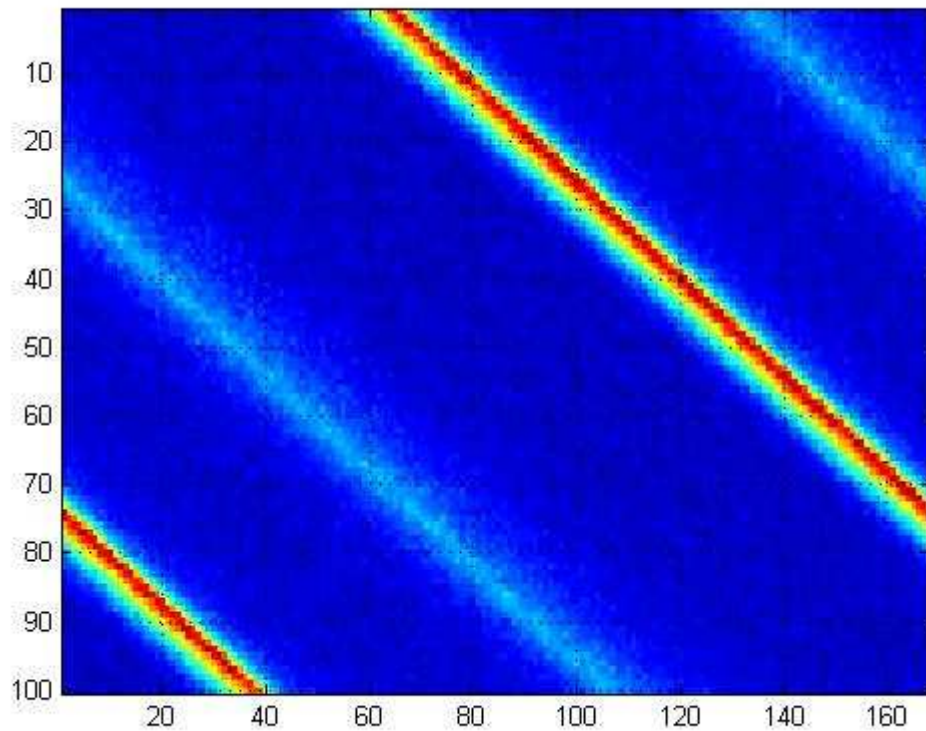


Figura 17 Waterfall corrispondente alla light curve errata per pulsar Crab

Nel grafico cromatico il modulo degli elementi del waterfall viene rappresentato per mezzo di una scala di colori in cui il valore massimo corrisponde a un rosso scuro e il valore di modulo minore a un blu scuro; i valori intermedi vengono rappresentati con tonalità dall'azzurro al rosso chiaro, a seconda della vicinanza con il valore massimo o minimo. In Figura 17 si può vedere come per ogni riga del waterfall siano presenti due picchi nei valori del segnale, uno rosso con un fronte di salita giallo e un picco più basso di colore azzurro; questi due picchi sono proprio quello che ci si aspetta di vedere nella light curve della pulsar. Il nostro errore è causato dal fatto che le righe del waterfall (e quindi le 100 sotto-light curve) sono disallineate tra di loro e al momento di sommarle tra di loro i picchi si distribuiscono su tutta la curva, riportandoci al risultato di Figura 16.

Il fatto che per le singole righe, la light curve sia coerente con la natura della pulsar, mentre la light curve totale su 30 minuti di osservazione non lo sia, è indice del fatto che il periodo utilizzato sia un numero vicino al periodo di rotazione che si desidera trovare. A

sostegno di questa tesi è anche la disposizione in diagonale dei picchi nel grafico, che altrimenti sarebbero non individuabili (in caso di periodo molto errato) o per lo meno in ordine sparso. Una deviazione dei picchi di quel tipo e in quella direzione suggerisce che il periodo trovato grazie alla Fourier Transform sia troppo basso.

Data la facilità di analisi del waterfall diagram e nota la lieve entità dell'errore, per arrivare alla curva di luce desiderata è stato sufficiente eseguire qualche test aumentando il periodo in ingresso all' algoritmo da utilizzare per il folding. Questo metodo empirico è applicabile solo coi dati della Crab pulsar, poichè con le altre due stelle, come vedremo nei capitoli seguenti, il waterfall è totalmente corrotto dal rumore ed è impossibile usarlo come base di partenza per decidere quali modifiche apportare; inoltre i picchi di luce non sono individuabili neppure in caso di folding corretto data la quantità di dati troppo bassa contenuta in una singola riga.

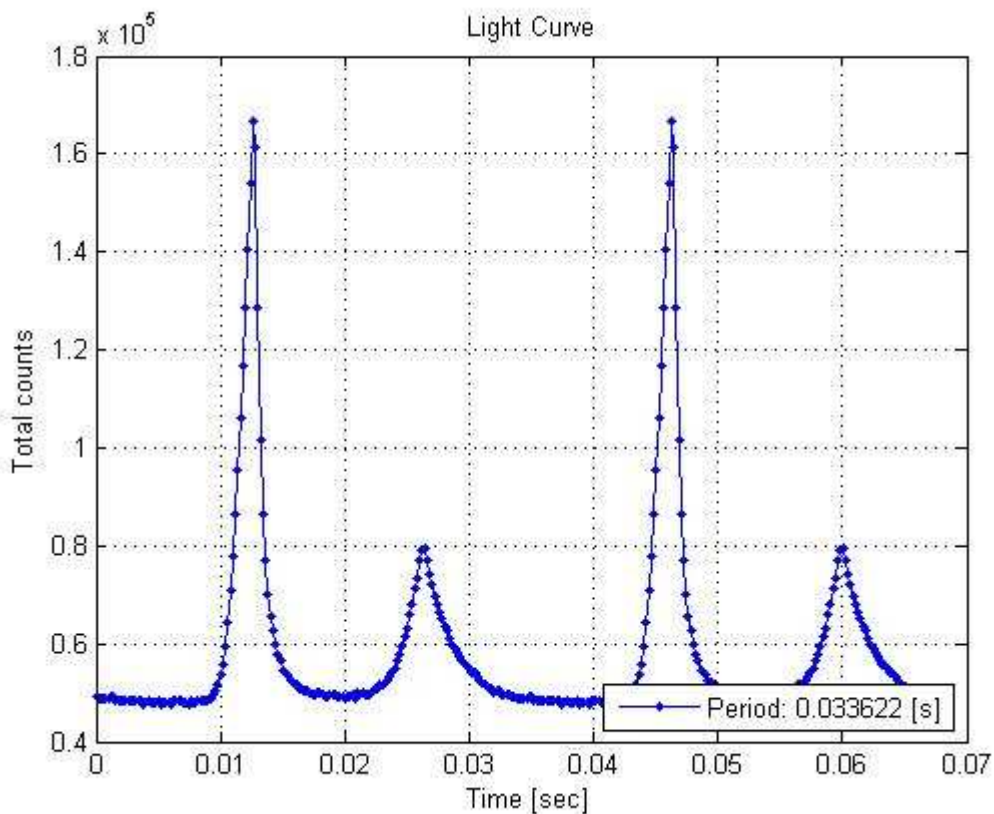


Figura 18 Light curve di Crab pulsar con periodo di rotazione esatto. Due periodi visualizzati

Il risultato finale si può vedere in Figura 18 in cui i fotoni appartenenti ai coni di radiazioni sono ben visibili nella light curve e in Figura 19 dove i picchi di ogni riga del waterfall sono allineati.

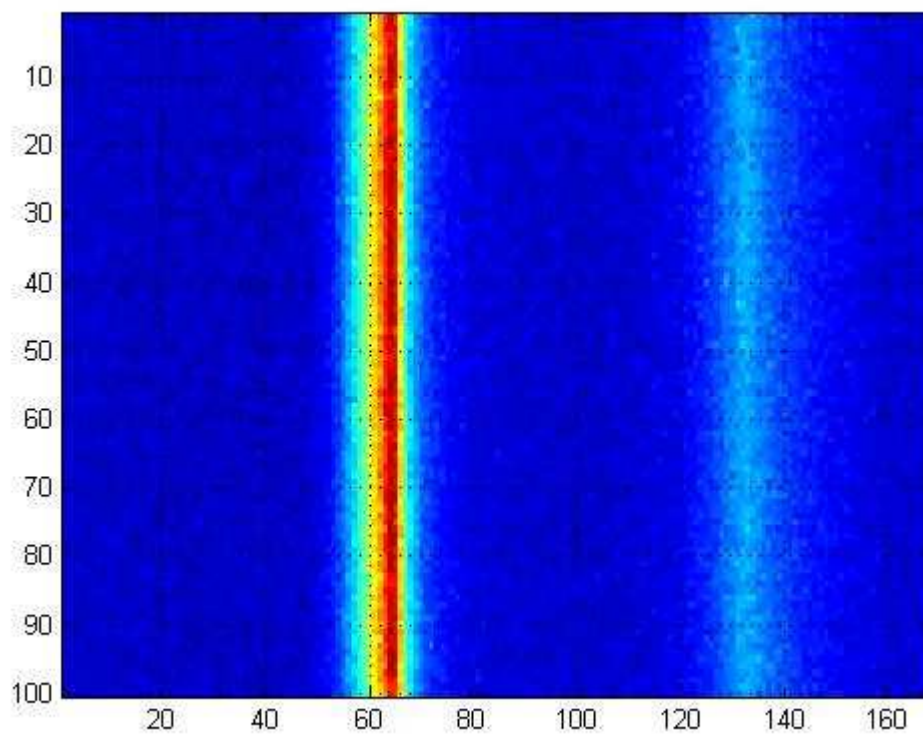


Figura 19 Waterfall diagram per Crab pulsar.

4.4 Folding dei dati per PSR B0540-69

Per il folding dei dati provenienti da PSR B0540-69 è stato usato lo stesso algoritmo adoperato per la Crab; alcuni parametri però sono stati modificati data la diversa natura del segnale generato da IquEye.

La dimensione dei time bin è stata aumentata fino a 10 volte ($dt = 2 * 10^{-3}$) perchè la variazione del numero dei fotoni tra un istante di “buio” e un istante in cui il telescopio è investito da uno dei coni della pulsar è estremamente bassa; inoltre, sempre a causa del rumore introdotto dalla luce circostante la stella, sono presenti alcuni picchi minori di luminosità; riducendo la granularità della curva di luce l’immagine appare più visibile a occhio nudo ed è possibile distinguere con più facilità i picchi che vogliamo evidenziare.

A differenza di Crab, PSR B0540-69 non è monitorata costantemente e non è disponibile la stessa quantità di informazioni. La derivata nel tempo del periodo di rotazione è stata calcolata un numero limitato di volte e a distanza di anni dalla data di acquisizione dei dati da parte di IquEye; poichè la derivata è variabile nel tempo, non è stato possibile considerare l’utilizzo di questi valori. Senza questo parametro, la curva di luce creata utilizzando un periodo costante risulta necessariamente meno precisa, tuttavia è possibile ottenere un risultato accettabile trovando un valore ottimale corrispondente al periodo “medio” dell’osservazione che si vuole analizzare.

Non potendo usare la derivata del periodo, la divisione in sotto-array dei dati per l’aggiornamento del periodo di rotazione perde gran parte del suo significato; sono stati usati comunque alcuni sotto-array per la creazione del waterfall diagram anche se, una volta eseguito il folding dei dati, appare evidente come il waterfall (Fig. 20) non sia utile come nel caso della pulsar Crab per decidere come modificare i parametri di ingresso all’algoritmo.

Per la Crab pulsar, la differenza nel numero di fotoni tra un momento di “buio” e uno dei momenti in cui la terra viene colpita dal cono di radiazioni è evidente; in Figura 18 si può vedere come piegando su se stessi i dati corrispondenti a 30 minuti di segnale, il massimo numero di fotoni contati in un dato dt $1.7 * 10^5$ mentre il minimo è circa $0.5 * 10^5$. Nel

caso di PSR B0540-69 tale differenza è minima a causa del rumore: per un'ora di osservazione il numero di fotoni contati al picco di segnale è circa $3.35 * 10^5$ mentre il conteggio al punto di minimo è di circa $3.3 * 10^5$ fotoni, corrispondenti al rumore. Una SNR così bassa, si traduce nel fatto che per avere un buon risultato è necessario eseguire il folding su una grande mole di dati, mentre le righe del waterfall diagram, che corrispondono alle light curve ottenute da sotto-array, risultano invece essere incorrelate tra loro. Dividendo i dati in un numero molto basso di sotto-array (sono stati costruiti waterfall a 100 righe, 25, 20, 10 fino a 5), il waterfall rimane inutile se osservato a occhio nudo.

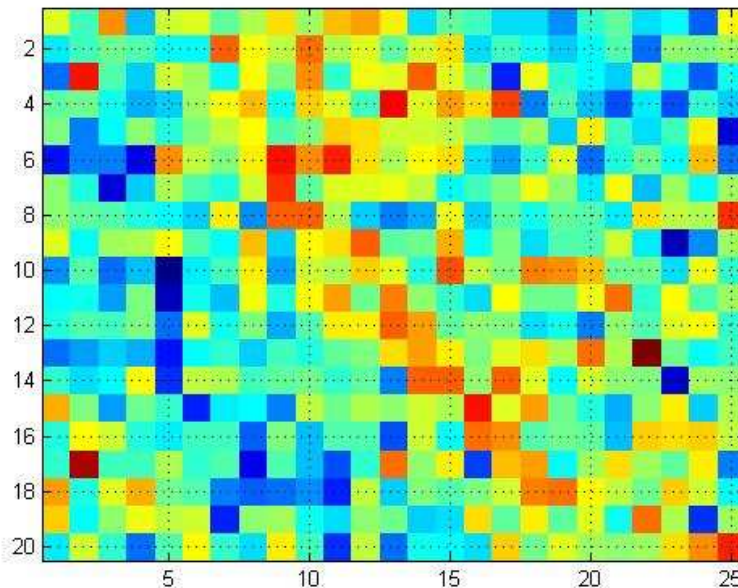


Figura 20 Waterfall diagram per PSR B0540-69 su 20 righe e con periodo trovato tramite Fourier Transform

La light curve ottenuta (Fig. 21) usando il periodo trovato con la trasformata di Fourier non è corretta se confrontata con le pubblicazioni riguardanti PSR B0540-59. Non è stato possibile correggere il periodo di rotazione basandosi sul waterfall diagram poichè la disposizione dei picchi appare casuale. Per questo si è dovuto ricorrere ad un altro metodo per trovare il valore ottimo da inserire nell'algoritmo di folding.

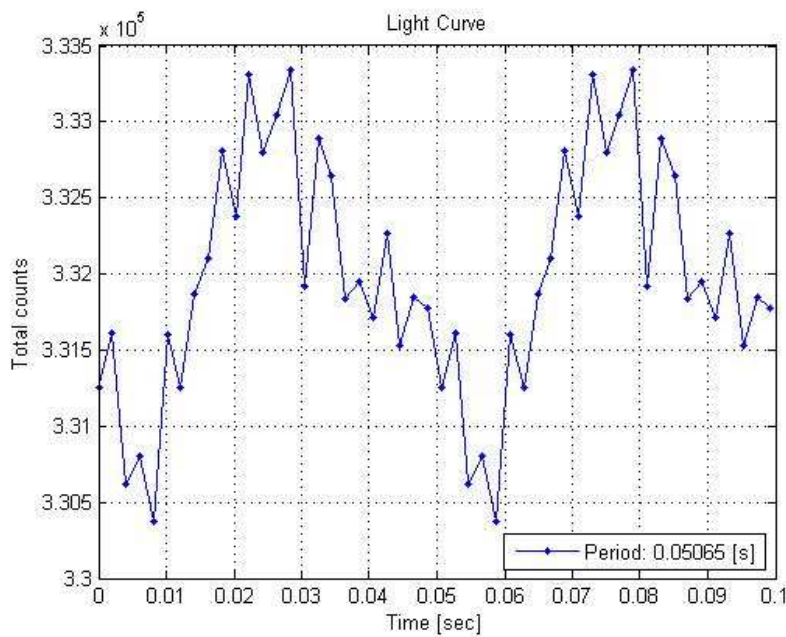


Figura 21 Light curve su due periodi per PSR B0540-69 ottenuta con periodo trovato tramite Fourier Transform

4.5 Ricerca del periodo di rotazione di PSR B0540-69

L'algoritmo per trovare il periodo ottimale per il folding prende come punto di partenza il periodo p_F ottenuto grazie all'analisi dello spettro del segnale e si sposta iterativamente in un suo intorno, fino ad ottenere un p_O che minimizza la merit function.

Ad ogni iterazione, l'algoritmo esegue il folding dei dati con un periodo $p_i = p_F + k\Delta t$, dove Δt può essere modificato dopo una prima serie di iterazioni per diminuire la variazione tra un periodo e l'altro al fine di avere una maggiore precisione temporale del valore in uscita dell'algoritmo.

Una volta eseguito il folding con periodo p_i , la curva di luce ottenuta viene sovrapposta all'intero segnale originale, binnato con un dt uguale a quello usato nel folding, e viene calcolata la distanza tra la light curve e il segnale tramite:

$$RMS_i = \sqrt{\sum_{j=1}^N I_j^2} \quad (4.1)$$

dove N è il numero di elementi del segnale binnato su cui si calcola la distanza e I_j è la distanza tra il punto della curva di luce e il punto del segnale corrispondente con indice j .

L'algoritmo, dopo un numero arbitrario di iterazioni, fornisce come output il valore p_i a cui corrisponde la RMS minima tra quelle calcolate; inserendo come nuovo input il periodo trovato e diminuendo il valore di Δt è possibile raffinare il valore di p_0 .

Con questo metodo è stato possibile trovare un periodo di rotazione per PSR B0540-69 che ha permesso di costruire la curva di luce in Figura 22.

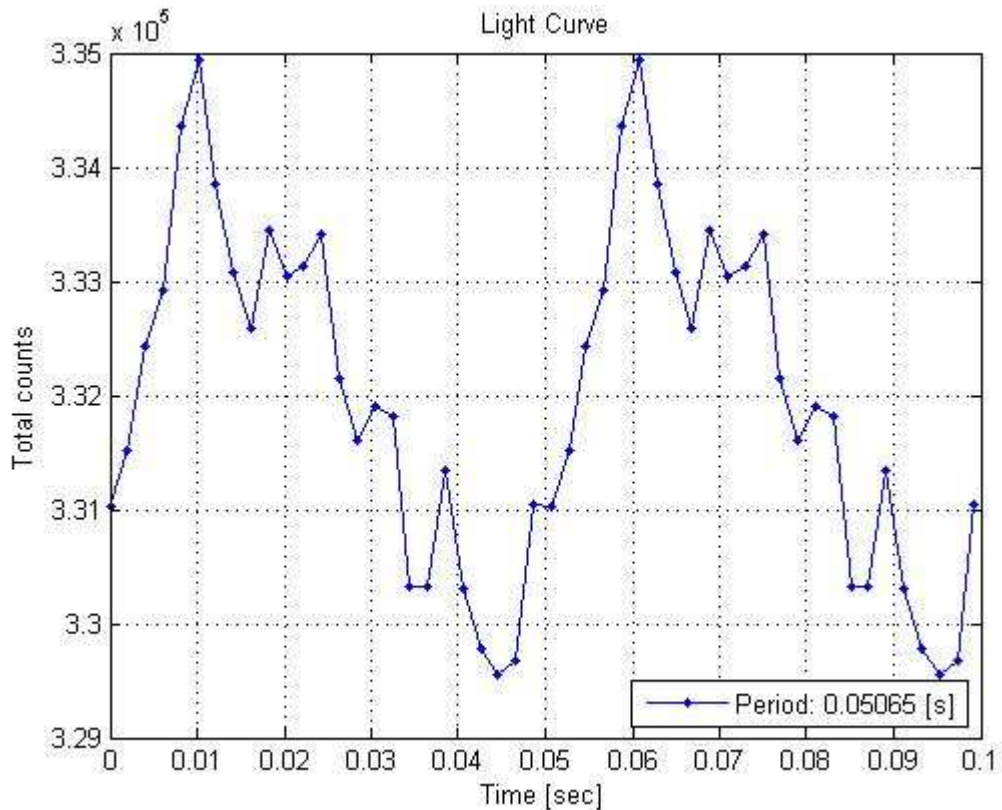


Figura 22 Light curve per PSR B0540-69 con periodo di rotazione corretto.

La bontà del risultato è confermata dal confronto con le curve di luce di PSR B0540-69 pubblicate finora; ad esempio, come si può vedere nelle Fig. 23-24, l'istogramma rappresentante la nostra curva di luce e la curva ottenuta da Boyd et al. [13] nel 1994 per mezzo del telescopio Hubble, sono molto simili tra loro e con le altre light curve esistenti.

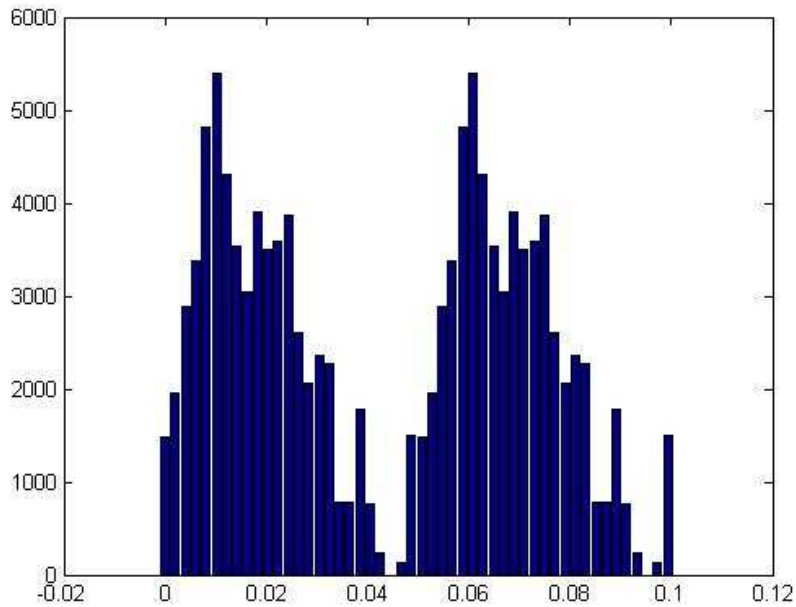


Figura 23 Light curve di PSR B0540-69, dati di IquEye su NTT

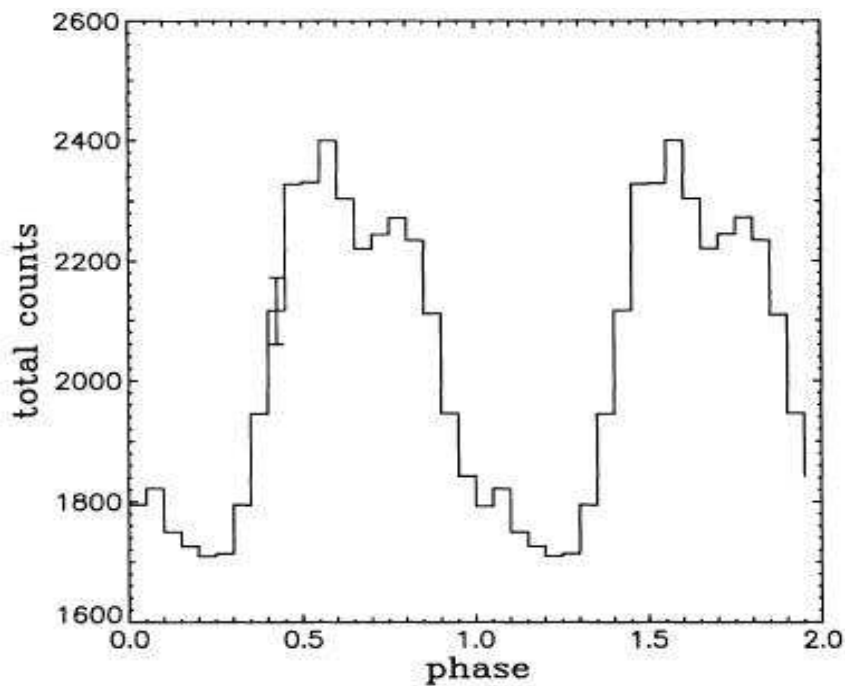


Figura 24 Light curve di PSR B0540-69, dati provenienti da HST

4.6 L'idea della KLT per Vela pulsar

Nell'analisi dei dati sui fotoni provenienti dalla pulsar Vela, i problemi incontrati fino ad ora si sono ripresentati con maggiore intensità.

Affinchè sia possibile vedere almeno una componente nello spettro del segnale è stato necessario prolungare l'osservazione della stella per compensare il SNR molto basso delle radiazioni in arrivo sulla Terra. Nel caso in esame i dati coprono 180 minuti di osservazione, un tempo molto maggiore rispetto alla durata standard per le altre stelle; di conseguenza il periodo di rotazione, durante l'osservazione, ha subito una variazione non trascurabile che ha contribuito a falsare l'analisi con la trasformata di Fourier. Bisogna anche considerare che un'acquisizione così lunga ha portato ad una maggiore quantità di dati che rendono inevitabilmente più lento e difficoltoso ogni tipo di elaborazione; il vettore costruito coi tempi di arrivo dei fotoni provenienti dalla pulsar Vela conta circa 15 milioni di elementi.

Come per PSR B0540-69, non è stato possibile ricorrere alla derivata del periodo nè all'uso del waterfall per correggere il folding dei dati. L'algoritmo di ricerca di un periodo medio ottimo non ha portato a risultati soddisfacenti e si è reso necessario trovare una nuova strategia che si rivelasse efficace.

L'idea di ricorrere alla trasformata di Karhunen-Loève è nata dalla pubblicazione di Bean et al. [1], in cui la KLT è stata utilizzata come strumento di denoising per un segnale totalmente immerso nel rumore. Il campo di ricerca e la natura dei dati presenti nell'articolo sono completamente diversi da quelli su cui si basa questa tesi, ma la pubblicazione ha comunque fornito una direzione da seguire.

All'inizio del lavoro, mancando qualunque conoscenza sulla KLT, non si sapeva se questo strumento sarebbe stato applicabile ai dati di IquEye, se avrebbe prodotto dei risultati utili e se sì, di quale natura. La speranza era, grazie alla KLT, di riuscire a ricostruire il segnale eliminando parte del rumore in modo da facilitare tutte le operazioni di analisi standard; oppure, in alternativa, di trovare un periodo di rotazione medio migliore di quello fornito dalla Fourier Transform da inserire nell'algoritmo di folding.

CAPITOLO 5

Principal Component Analysis (PCA)

5.1 Introduzione

La *Karhunen-Loève Transform* (KLT) è un argomento che spazia molti campi della scienza contemporanea; viene usato principalmente nella compressione ed analisi di immagini, ma anche in statistica, matematica e signal processing. A seconda del campo di applicazione, questa tecnica di analisi apporta delle piccole variazioni all'algoritmo ed al nome: la trasformata che trova spazio nell'analisi e compressioni di immagini e nello studio di dati statistici viene chiamata *Principal Component Analysis* (PCA), KLT è invece il nome con cui ci si riferisce allo strumento orientato al signal processing, in particolare si usa per il denoising di segnali con SNR particolarmente basso e, pur mantenendo l'idea alla base della trasformata, l'algoritmo e il metodo di applicazione sono diversi da quello della PCA. Altri nomi con cui tale trasformata è nota sono *Independent Component Analysis* (ICA) e *Hotelling Transform* (HT), ma questi ultimi sono solo modi diversi per definire lo stesso strumento matematico.

In questo capitolo verrà spiegata nel dettaglio la teoria riguardante la PCA, il capitolo successivo è invece dedicato alla KLT per signal processing poichè entrambe sono state applicate ai dati delle pulsar.

5.2 Il fondamento della Principal Component Analysis

La Principal Component Analysis (PCA) è uno strumento utilizzato nella moderna analisi di dati riguardanti diversi campi di ricerca: dalle neuroscienze alla computer graphic. Essa è un metodo relativamente semplice, non parametrico, per l'estrazione di informazioni rilevanti da dati di difficile interpretazione.

L'idea alla base della PCA è di ridurre la dimensionalità del data set in cui sono presenti un gran numero di variabili correlate, mantenendo quanta più varianza possibile nei dati. La riduzione viene fatta passando a un nuovo set di variabili, le componenti principali, incorrelate tra loro e ordinate in modo che le prime mantengano la maggior parte della varianza presente in tutte le variabili originali.

Per meglio capire lo scopo della PCA, si immagini di studiare la natura di un fenomeno con una serie di misure (lo spettro, il voltaggio, la velocità) del sistema; i dati acquisiti possono essere confusi, ridondanti e sembrano non fornire informazioni valide. Si può prendere ad esempio lo studio del moto di una sfera di massa m attaccata a una molla ideale, priva di massa e che non considera la forza di attrito (Fig. 25). La sfera è rilasciata a una piccola distanza dal punto di equilibrio e la molla è quindi in tensione all'istante 0; essendo la molla ideale, la sfera oscilla lungo l'asse x intorno al suo punto di equilibrio ad una certa frequenza. Il movimento della massa m può essere espressa in funzione della variabile x nel tempo, ma, se si suppone che l'osservatore non sia a conoscenza di questo, le misure del moto potrebbero essere eseguite su più dimensioni, anche non perpendicolari tra loro.

Sembra un'assurdità, ma nella sperimentazione reale accade spesso che una scarsa conoscenza a priori del fenomeno induca ad effettuare misurazioni che non descrivono al meglio la dinamica del sistema, inoltre nella pratica occorre considerare la non idealità del sistema e quindi la presenza di rumore che complica ulteriormente l'analisi.

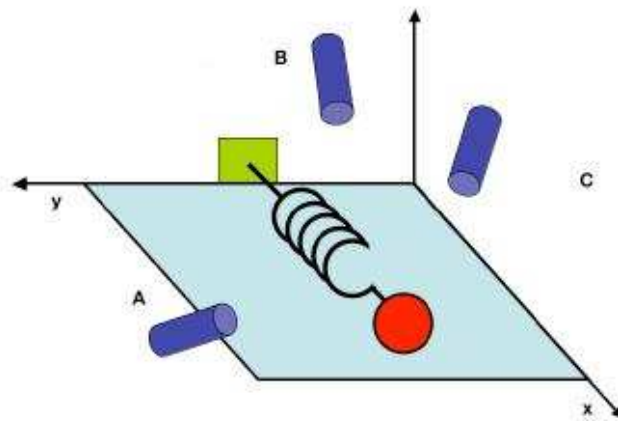


Figura 25 Esempio di problema fisico, il cui moto è misurato in 2 dimensioni da tre telecamere A, B, C.

Nell'esempio della molla la PCA ha lo scopo di trovare il vettore \vec{x} , la base che descrive le dinamiche del fenomeno lungo l'asse x , che è la dimensione più importante. Misurando la posizione della sfera dal punto A, si ottengono una serie di coordinate (x_A, y_A) . Una misurazione su tutte le tre telecamere si può rappresentare col vettore riga $\vec{X} = [x_A, y_A, x_B, y_B, x_C, y_C]$, in cui ogni telecamera contribuisce a una proiezione bidimensionale della posizione della sfera rispetto a \vec{X} . Ogni campione della misura è un vettore di dimensione m , quindi è contenuto in uno spazio vettoriale m -dimensionale originato da una base ortonormale; se si pensa a una base per uno spazio a m dimensioni, la prima e più semplice da trovare è la matrice identità I_m , quindi per le misure dal punto A la base vettoriale "standard" è $[(1,0), (0,1)]$.

L'obiettivo della PCA è di trovare una base vettoriale alternativa, combinazione lineare della base originale, che meglio esprima le proprietà del data set, filtrando il rumore e rivelando la struttura prima nascosta dei dati. Il termine lineare è molto importante: la PCA infatti pone il vincolo della linearità. Questa assunzione restringe il set di basi potenziali e semplifica notevolmente il problema, la PCA semplicemente ri-esprime i dati come una combinazione lineare dei suoi vettori base.

Sia \mathbf{X} una matrice $n \times m$ in cui ogni colonna è la misurazione in un istante temporale del fenomeno fisico, e sia \mathbf{Y} un'altra matrice $m \times n$ prodotta da una trasformazione \mathbf{P} sul nostro data set.

$$\mathbf{P}\mathbf{X} = \mathbf{Y} \quad (5.1)$$

L'equazione (5.1) rappresenta un cambio di base e si può interpretare come una trasformazione da \mathbf{X} a \mathbf{Y} per mezzo di \mathbf{P} , oppure come una rotazione geometrica secondo la matrice \mathbf{P} , oppure si può vedere \mathbf{P} come un set di nuovi vettori base (disposti per righe) che esprimono le colonne di \mathbf{X} . Quest'ultima interpretazione si vede esplicitando il prodotto

$$\mathbf{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{bmatrix} [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n]$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p}_1\mathbf{x}_1 & \cdots & \mathbf{p}_1\mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{p}_m\mathbf{x}_1 & \cdots & \mathbf{p}_m\mathbf{x}_n \end{bmatrix}$$

Per ogni colonna di \mathbf{Y}

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{p}_1\mathbf{x}_i \\ \vdots \\ \mathbf{p}_m\mathbf{x}_i \end{bmatrix}$$

Ogni coefficiente di \mathbf{y}_i è il prodotto esterno di \mathbf{x}_i con la riga corrispondente di \mathbf{P} . In altre parole il j -esimo coefficiente di \mathbf{y}_i è una proiezione di \mathbf{x}_i sulla j -esima riga di \mathbf{P} , quindi l'intero vettore \mathbf{y}_i è una proiezione sulla base $[\mathbf{p}_1, \dots, \mathbf{p}_m]$ e le righe di \mathbf{P} sono un nuovo set di vettore base per la rappresentazione delle colonne di \mathbf{X} .

Presupponendo la linearità, il problema si riduce a trovare l'appropriato cambio di base. I vettori base che massimizzano la varianza del segnale saranno i *componenti principali* di \mathbf{X} .

5.3 Base ottimale del data set

Qualunque sia il data set in esame, è auspicabile che il rumore sia il minimo possibile per poter analizzare al meglio il segnale di interesse; non esistendo una scala assoluta per la quantità di rumore si ricorre al *Signal to Noise Ratio* (SNR), calcolato come il rapporto tra le varianze del segnale e del rumore.

$$SNR = \frac{\sigma_{signal}^2}{\sigma_{noise}^2} \quad (5.2)$$

In Figura 26 si può vedere una simulazione della rappresentazione su (x,y) dei dati raccolti dalla telecamera posta nel punto di osservazione A per l'esperimento della molla.

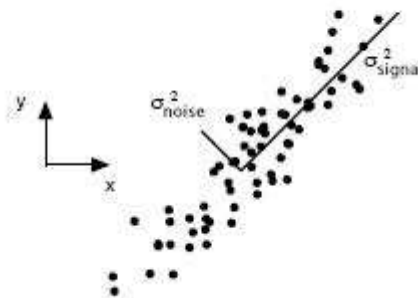


Figura 26 Dati (x,y) simulati per la telecamera in A, sono visibili la varianza di segnale e rumore.

Poichè la molla si muove in una direzione (x) , i dati raccolti da ogni telecamera dovrebbero essere posti su una linea retta, di conseguenza ogni deviazione da quella linea è da considerarsi rumore. Ipotizzando che la misura sia stata effettuata con sufficiente perizia e strumentazione, si può assumere che la direzione con una varianza maggiore nello spazio di misura sia quella che contiene le dinamiche importanti per lo studio del fenomeno. Nel caso della Figura 26, la direzione di interesse non è lungo gli assi x o y , cioè la base formata da $[(1,0), (0,1)]$, ma lungo l'asse più esteso della nuvola costituita dai dati.

Massimizzare la varianza (e per assunzione anche il SNR), significa trovare la matrice di rotazione che sposti il vettore (1, 0) sulla direzione di σ_{signal}^2 , in modo da rivelare la direzione del moto della molla nel caso a 2 dimensioni in esame.

In un modello a n dimensioni, la PCA trasforma i dati in un nuovo sistema di coordinate in cui la varianza maggiore per ogni proiezioni dei dati è posta nella prima coordinata (la prima componente principale), la seconda varianza più grande nella seconda componente e così via fino alla n -esima. L'informazione di interesse è posta nelle prime k componenti, con k che va deciso caso per caso a seconda del data set.

5.4 Matrice di covarianza

Si considerino due set di misure dello stesso fenomeno con media nulla $A = \{a_1, \dots, a_n\}$ e $B = \{b_1, \dots, b_n\}$, dove l'indice indica il numero del campione. Le varianze di A e B sono definite come

$$\sigma_A^2 = \frac{1}{n} \sum_i a_i^2 \quad (5.3.a)$$

$$\sigma_B^2 = \frac{1}{n} \sum_i b_i^2 \quad (5.3.b)$$

mentre la covarianza tra le due misure è

$$\sigma_{AB}^2 = \frac{1}{n} \sum_i a_i b_i \quad (5.4)$$

La covarianza misura il grado di relazione lineare tra le due variabili ed è tanto maggiore quanto i dati sono correlati tra di loro, fino al valore massimo $\sigma_{AB}^2 = \sigma_A^2$ per il caso $A = B$. Il valore assoluto della covarianza misura il grado di ridondanza nell'acquisizione dei dati.

Prendendo al posto di A e B , una serie di vettori $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$, si ottiene la matrice \mathbf{X} di dimensione $m \times n$.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix}$$

Ogni riga di \mathbf{X} corrisponde a tutte le misure di un particolare tipo (uno stesso sensore ad esempio), ogni colonna invece è un set di misure di una particolare osservazione (corrispondenti a $\vec{\mathbf{X}}$ nel paragrafo 5.2).

La matrice di covarianza per \mathbf{X} è definita con

$$\mathbf{C}_X = \frac{1}{n} \mathbf{X} \mathbf{X}^T \quad (5.5)$$

Questa matrice ha tre importanti proprietà:

- È una matrice simmetrica $m \times m$.
- I valori posti sulla diagonale corrispondono alla varianza tra i vari tipi di misura (le righe di \mathbf{X}).
- I valori posti al di fuori della diagonale corrispondono alla covarianza tra i tipi di misura.

Riassumendo quanto detto sino ad ora, l'obiettivo è massimizzare la varianza nelle direzioni contenenti le informazioni significative, rappresentata dalla diagonale della matrice \mathbf{C}_X , e minimizzare la ridondanza dei dati e il rumore, rappresentati dalla covarianza che si trova negli elementi non in diagonale.

5.5 Costruzione di \mathbf{P}

Per valorizzare la varianza e minimizzare la covarianza di \mathbf{C}_X , occorre trasformare la matrice di covarianza in una matrice diagonale \mathbf{C}_Y i cui valori sulla diagonale principale sono ordinati secondo la varianza. Il metodo più semplice per farlo è assumere che i vettori base $\{\mathbf{p}_1, \dots, \mathbf{p}_m\}$ della matrice di trasformazione siano ortonormali. In uno spazio a molte dimensioni i passi da eseguire sono:

1. Scegliere una direzione normalizzata nello spazio a m dimensioni, in cui la varianza di \mathbf{X} sia massimizzata e assegnare a \mathbf{p}_1 il vettore risultante.
2. Trovare un'altra direzione in cui la varianza è massimizzata; a causa della condizione di ortonormalità, cercare tra le direzioni ortogonali a quelle già scelte ed assegnare il vettore \mathbf{p}_i .
3. Creare in questo modo m vettori.

La matrice \mathbf{P} risultante conterrà le componenti principali ordinate per riga. L'idea alla base della costruzione di \mathbf{P} è di assegnare ad ogni riga \mathbf{p}_i un autovettore di \mathbf{C}_X . Questa operazione è possibile grazie ai teoremi [1] e [2].

Teorema [1]. *Una matrice è simmetrica se e solo se è ortogonalmente diagonalizzabile.*

Teorema [2]. *Una matrice simmetrica A è diagonalizzata da una matrice E composta da autovettori ortonormali di A .*

Dim [2].

Sia A una matrice simmetrica $n \times n$ i cui autovettori siano $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$. Sia E la matrice avente come i -esima colonna l'autovettore \mathbf{e}_i . Il teorema stabilisce che esiste una matrice diagonale D tale che $A = EDE^T$.

La prova è in due parti, nella prima si dimostra che ogni matrice può essere diagonalizzata ortogonalmente se e solo se i suoi autovettori sono linearmente indipendenti; nella seconda si dimostra che una matrice simmetrica ha la proprietà di avere autovettori non solo linearmente indipendenti ma anche ortogonali.

Sia A una matrice, non necessariamente simmetrica, con autovettori indipendenti e sia $E = [\mathbf{e}_1, \dots, \mathbf{e}_n]$ la matrice avente tali autovettori come colonne. Sia D una matrice diagonale in cui l' i -esimo autovalore è posto nella posizione ii .

Esaminando i prodotti AE e ED si può vedere che $AE = ED$

$$\begin{aligned} AE &= [A\mathbf{e}_1, \dots, A\mathbf{e}_n] \\ ED &= [\lambda_1\mathbf{e}_1, \dots, \lambda_n\mathbf{e}_n] \end{aligned}$$

Se $AE = ED$, allora $A\mathbf{e}_i = \lambda_i\mathbf{e}_i \forall i$, ma questa uguaglianza è sempre verificata per la definizione di autovalore. Di conseguenza $A = EDE^{-1}$.

La seconda parte della dimostrazione prova che una matrice simmetrica ha sempre autovettori ortogonali. Data una matrice simmetrica, siano λ_1 e λ_2 due autovalori distinti per gli autovettori \mathbf{e}_1 e \mathbf{e}_2 .

$$\begin{aligned} \lambda_1\mathbf{e}_1 \cdot \mathbf{e}_2 &= (\lambda_1\mathbf{e}_1)^T \mathbf{e}_2 \\ &= (A\mathbf{e}_1)^T \mathbf{e}_2 \\ &= \mathbf{e}_1^T A^T \mathbf{e}_2 \\ &= \mathbf{e}_1^T (\lambda_2\mathbf{e}_2) \\ &= \lambda_2\mathbf{e}_1 \cdot \mathbf{e}_2 \end{aligned}$$

Dall'ultima equazione si può dire che $(\lambda_1 - \lambda_2)\mathbf{e}_1 \cdot \mathbf{e}_2 = 0$ ed essendo gli autovalori unici per ipotesi, significa che $\mathbf{e}_1 \cdot \mathbf{e}_2 = 0$. Di conseguenza gli autovettori della matrice simmetrica sono ortogonali.

Tornando al postulato originale per cui A è una matrice simmetrica, per la seconda parte della dimostrazione, si sa che gli autovettori di A sono tutti ortonormali (si è scelto di normalizzarli); di conseguenza E è una matrice ortonormale e $E^T = E^{-1}$. Si può quindi scrivere che $A = EDE^{-1}$. □

La matrice diagonale C_Y , ottenuta diagonalizzando la C_X tramite la matrice P così costruita, contiene nell' i -esimo elemento della sua diagonale la varianza della matrice X dei dati lungo p_i , che è anche l'autovalore associato al vettore p_i . A questo punto il calcolo della PCA si riduce ad eseguire la proiezione di X sulle componenti principali che meglio rappresentano i dati.

5.6 La scelta delle componenti principali

Nell'esempio della sfera agganciata alla molla, l'unica informazione importante è il moto lungo l'asse x , la cui varianza viene massimizzata dalla proiezione dei dati sulla prima componente principale (PC), mentre scartando la seconda componente si elimina il rumore della misurazione. Nelle applicazioni pratiche non è così facile capire quali componenti tenere in considerazione e quali scartare. Scegliendo solo la prima PC o le prime m con un m troppo piccolo, si rischia di perdere informazioni importanti; se invece la scelta è di tenere troppe PC, si mantiene buona parte del rumore e delle informazioni indesiderate, vanificando l'uso della PCA.

È possibile che le informazioni cercate siano contenute nelle ultime PC invece che nelle prime con maggiore varianza, tuttavia è un evento raro e dipendente dal tipo di dati in analisi e non esiste una regola generale per scegliere quali PC considerare.

Se lo scopo è di ridurre la dimensionalità dei dati usando le prime m componenti invece delle p variabili di partenza (con $m < p$), esistono alcune regole; tuttavia la loro origine non è dovuta a dimostrazioni formali, ma piuttosto a una serie di intuizioni plausibili supportate da una validità pratica.

Il criterio più diffuso consiste nello scegliere la percentuale della varianza totale che si desidera mantenere (valori tipici sono 80%-90%) e sommare i valori di varianza associati alle varie PC fino al raggiungimento della percentuale prefissata. Il numero di elementi sommati corrisponde al numero m di PC da tenere.

Sia data la varianza totale

$$t_{TOT} = \sum_{j=1}^p l_j \quad (5.6)$$

dove p è il numero totale di PC e l_j è la varianza della j -esima PC, il cui valore è in \mathbf{C}_Y con indice jj ; si scelga poi una percentuale, ad esempio 90%, e si ponga $t^* = 0.9 * t_{TOT}$. Per trovare il numero minimo m di PC da quindi considerare, basta sommare le varianze

$$t_m = \sum_{j=1}^m l_j \quad (5.7)$$

fino ad ottenere che $t_m \geq t^*$.

Altri criteri sono stati ideati per la KLT costruita con la matrice di correlazione invece che con la matrice di covarianza (si veda il capitolo 6). Il primo è detto *Kaiser rule*, secondo cui le PC da considerare sono tutte quelle la cui varianza associata è $l_k \geq 1$; in seguito Jolliffe [8] ha suggerito di prendere un valore di *cut-off* più basso e considerare le PC per cui $l_k \geq 0.7$. Queste due regole possono essere adattate alla PCA calcolata con l'uso della matrice di covarianza grazie a una semplice modifica: è sufficiente moltiplicare i valori di *cut-off* per la media delle varianze.

$$\bar{l} = \frac{1}{p} \sum_{j=1}^p l_j \quad (5.8)$$

5.7 L'algoritmo per il calcolo della PCA

Qui di seguito sono elencati i passi da compiere per determinare, grazie all'analisi delle componenti principali, la matrice \mathbf{Y} , di ridotta dimensionalità rispetto ai dati di ingresso, ma che mantiene gran parte delle informazioni considerate rilevanti.

1. Si prendono i dati organizzati in N vettori $\mathbf{x}_1, \dots, \mathbf{x}_N$ (uno per ogni osservazione) di lunghezza M e si costruisce la matrice \mathbf{X} di dimensione $M \times N$.
2. Viene calcolato il vettore contenente la media per ogni colonna e lo si sottrae a \mathbf{X} ottenendo $\tilde{\mathbf{X}}$.

$$\mathbf{u}[m] = \frac{1}{N} \sum_{n=1}^N \mathbf{X}[m, n] \quad (5.9)$$

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{u}\mathbf{h} \quad (5.10)$$

$$h[n] = 1 \\ \text{per } n = 1, \dots, N$$

3. Si calcola la matrice di covarianza di $\tilde{\mathbf{X}}$.

$$\mathbf{C}_X = \frac{1}{n} \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T \quad (5.11)$$

4. Si calcola la matrice \mathbf{V} di autovettori che diagonalizza \mathbf{C}_X

$$\mathbf{V}^{-1}\mathbf{C}_X\mathbf{V} = \mathbf{D} \quad (5.12)$$

Dove \mathbf{D} è una matrice diagonale $M \times M$ contenente gli autovalori di \mathbf{C}_X e \mathbf{V} è un'altra matrice $M \times M$ in cui ogni colonna è un autovettore di \mathbf{C}_X . La m -esima colonna di \mathbf{V} è l'autovettore associato all'autovalore in posizione $[m, m]$ di \mathbf{D} .

5. Si ordinano le colonne di \mathbf{V} e di \mathbf{D} secondo l'ordine decrescente degli autovalori contenuti in \mathbf{D} , mantenendo l'accoppiamento tra autovalori ed autovettori.
6. Si sceglie il numero q di PC da utilizzare per il calcolo della trasformata e si crea la matrice \mathbf{Q} , di dimensione $q \times q$ contenente le prime q colonne di \mathbf{V} .
7. Si calcola il vettore \mathbf{s} , di lunghezza M , contenente la deviazione standard degli elementi sulla diagonale della matrice di covarianza \mathbf{C}_X

$$\mathbf{s} = \{s[m]\} = \sqrt{\mathbf{C}_X[m, m]} \quad (5.13)$$

per $m = 1, \dots, M$

e si utilizza per calcolare la matrice $N \times M$ di z -score a partire dai dati sorgente, dividendo elemento per elemento

$$\mathbf{Z} = \frac{\mathbf{B}}{\mathbf{s} \cdot \mathbf{h}} \quad (5.14)$$

Questo passo non fa parte dell'algoritmo classico per la PCA, ma normalizza il data set rispetto alla varianza ed è utile a molte applicazioni.

8. I dati (o gli z -score) vengono proiettati sulla nuova base e si ottiene la trasformata. Le colonne di \mathbf{Y} rappresentano la trasformata delle colonne di \mathbf{X} .

$$\mathbf{Y} = \mathbf{Q}^T \cdot \mathbf{Z} = \text{PCA}\{\mathbf{X}\} \quad (5.15)$$

CAPITOLO 6

Karhunen-Loève Transform (KLT)

6.1 Un approccio intuitivo alla KLT per signal processing

Il concetto alla base della KLT è molto semplice: essenzialmente si tratta di una trasformazione degli assi coordinati nello spazio di Hilbert. Nel caso dell'elaborazione numerica di segnale questo significa che, mediante lo sviluppo in serie KL di un segnale affetto da rumore e da una conseguente analisi degli autovalori associati (o più probabilmente di K autovalori dominanti), è possibile ottenere una rivelazione del segnale coerente, ed ancora di più una ricostruzione dello stesso. Ad un grande autovalore dominante, ottenuto grazie ad un livello sufficientemente alto di rapporto segnale rumore (SNR), corrisponde un'autofunzione che è una buona rappresentazione del segnale stesso. Questo è di particolare interesse per i segnali a larga banda, per i quali la trasformata di Fourier non è poi così utile, fornendo un potente strumento matematico per il recupero ottimale di qualunque segnale immerso in qualsiasi forma di rumore.

La PCA opera su una matrice di dati, che può rappresentare diverse osservazioni di un fenomeno oppure una immagine, e ne calcola la matrice di covarianza; la KLT per signal processing invece può operare su un unico vettore, che rappresenta il segnale nel tempo, e ne calcola l'autocorrelazione che riveste un compito analogo alla matrice di covarianza per la PCA. Lo scopo della trasformata è ovviamente lo stesso per entrambi i casi e cioè di valorizzare le informazioni di interesse contenute nei dati e minimizzare il contributo del rumore, il metodo di applicazione sul data set di IquEye però è diverso a causa della natura dei due algoritmi: la KLT si calcola su un segnale rappresentato da un vettore di dati, mentre la PCA necessita di una matrice di valori.

Nonostante le principali proprietà della KLT siano conosciute e ne sia stata dimostrata la sua efficacia in diversi campi, non è possibile non considerare le importanti difficoltà computazionali insite negli algoritmi di sviluppo, dalle quali non è semplice svincolarsi,

con il conseguente scarso utilizzo della KLT stessa. In questo stesso lavoro l'ingente quantità di risorse richieste per processare la grande mole di dati a disposizione ha rappresentato il problema principale.

6.2 La teoria per il caso continuo

Sia $X(t)$ un segnale rumoroso nello spazio di Hilbert, si definisce sviluppo KL del segnale la serie infinita

$$X(t) = \sum_{n=1}^{\infty} Z_n \phi_n(t) \quad (6.1)$$

dove gli Z_n sono variabili aleatorie casuali e le $\phi_n(t)$ sono funzioni che rappresentano la dipendenza temporale; il tempo varia all'interno dell'intervallo chiuso $0 \leq t \leq T$, dove T è detto tempo di integrazione. Le funzioni $\phi_n(t)$ sono le autofunzioni ottenute dalla soluzione dell'equazione integrale

$$\int_0^T E\{X(t_1)X(t_2)\} \phi_n(t_2) dt_2 = \lambda_n \phi_n(t_1) \quad (6.2)$$

dove $E\{X(t_1)X(t_2)\}$ è l'autocorrelazione del processo $X(t)$ che è data da

$$E\{X(t_1)X(t_2)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f_{X(t_1), X(t_2)}(x_1, x_2) dx_1 dx_2 \quad (6.3)$$

in cui $f_{X(t_1), X(t_2)}(x_1, x_2)$ è la densità di probabilità Gaussiana bidimensionale.

Per il calcolo dell'equazione (6.2) si suppone che l'autocorrelazione sia una funzione nota di t_1 e t_2 ; la convergenza dell'espansione KL richiede anche che la media del processo in esame sia zero.

$$E\{X(t)\} = \int_{-\infty}^{\infty} x f_{X(t)}(x) dx = 0 \quad (6.3)$$

Poichè la media di un segnale non sempre è nulla, e in particolare non lo è per il segnale costituito dai dati prodotti da IquEye, prima del calcolo dell'espansione KL il segnale $X(t)$ viene sostituito con $X(t) - E\{X(t)\}$.

L'equazione (6.2) costituisce l'*eigen problem*³ in cui l'operatore ha un nucleo simmetrico. Questo rende l'operatore autoaggiunto e, come conseguenza, gli autovalori sono numeri reali e le autofunzioni sono ortogonali tra loro. Questa ortogonalità si può esprimere con

$$\int_0^T \phi_m(t) \phi_n(t) dt = \delta_{mn} = \begin{cases} 0 & \text{per } m \neq n \\ 1 & \text{per } m = n \end{cases} \quad (6.4)$$

Le autofunzioni non solo sono ortogonali tra di loro, ma sono anche complete, cioè le $\phi_n(t)$ formano una base ortonormale nello spazio di Hilbert $L^2(0, T)$.⁴

Moltiplicando l'espansione (6.1) per una delle autofunzioni $\phi_m(t)$ e integrando nel tempo, l'ortonormalità delle autofunzioni (6.4) porta a una espressione formale per i coefficienti Z_m .

$$Z_m = \int_0^T X(t) \phi_m(t) dt \quad (6.5)$$

³ Si veda a proposito il Capitolo 7

⁴ Spazio delle funzioni ad energia finita definite nell'intervallo $[0, T]$

Ora è possibile dimostrare che le variabili Z_n sono ortogonali tra di loro nel senso della correlazione:

$$\begin{aligned}
 E\{Z_m Z_n\} &= E\left\{\int_0^T X(t_1) \phi_m(t_1) dt_1 \int_0^T X(t_2) \phi_n(t_2) dt_2\right\} \\
 &= \int_0^T \phi_m(t_1) dt_1 \int_0^T \phi_n(t_2) dt_2 E\{X(t_1)X(t_2)\} \\
 &= \lambda_n \int_0^T \phi_m(t_1) \phi_n(t_1) dt_1 \\
 &= \lambda_n \delta_{mn}
 \end{aligned}$$

e quindi

$$E\{Z_m Z_n\} = \lambda_n \delta_{mn} \quad (6.6)$$

L'espressione (6.5) permette di dimostrare anche che la media delle variabili Z_n è nulla

$$\begin{aligned}
 E\{Z_n\} &= E\left\{\int_0^T X(t) \phi_n(t) dt\right\} \\
 &= \int_0^T E\{X(t)\} \phi_n(t) dt \\
 &= \int_0^T 0 \phi_n(t) dt \\
 &= 0
 \end{aligned}$$

Combinando questo risultato e l'espressione (6.6), ponendo $m = n$, la varianza $\sigma_{Z_n}^2$ di ogni Z_n risulta uguale all'autovalore associato, come è dimostrato dalla formula

$$\sigma_{Z_n}^2 = E\{Z_n^2\} - E^2\{Z_n\} = E\{Z_n^2\} - 0 = E\{Z_n Z_n\} = \lambda_n \delta_{nm} = \lambda_n \quad (6.7)$$

dalla quale si ottiene che la deviazione standard di ciascun Z_n attorno alla media zero, è uguale a $\pm\sqrt{\lambda_n}$.

Costruendo l'espressione dell'autocorrelazione usando l'espansione di $X(t)$ (6.1) e l'ortogonalità delle variabili Z_n (6.6), si può ottenere l'autocorrelazione in termini dell'espansione delle autofunzioni (autovettori).

$$\begin{aligned}
E\{X(t_1)X(t_2)\} &= E\left\{\sum_{m=1}^{\infty} Z_m\phi_m(t_1)\sum_{n=1}^{\infty} Z_n\phi_n(t_2)\right\} \\
&= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} E\{Z_mZ_n\}\phi_m(t_1)\phi_n(t_2) \\
&= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \lambda_n\delta_{nm}\phi_m(t_1)\phi_n(t_2) \\
&= \sum_{n=1}^{\infty} \lambda_n\phi_n(t_1)\phi_n(t_2)
\end{aligned} \tag{6.8}$$

Poichè $X(t)$ ha media nulla e ponendo $t_1 = t_2 = t$ nella (6.8) si può esprimere la varianza di $X(t)$ nel modo seguente

$$\sigma_{X(t)}^2 = E\{X(t)X(t)\} - E^2\{X(t)\} = \sum_{n=1}^{\infty} \lambda_n\phi_n^2(t) - 0 = \sum_{n=1}^{\infty} \lambda_n\phi_n^2(t) \tag{6.9}$$

L'equazione (6.9) si può integrare rispetto a t nell'intervallo $0 \leq t \leq T$ ottenendone una versione semplificata poichè, per la (6.4), l'integrale di $\phi_n^2(t)$ è pari a 1.

$$\int_0^T \sigma_{X(t)}^2 dt = \sum_{n=1}^{\infty} \lambda_n \tag{6.10}$$

Quest'ultima formula stabilisce l'uguaglianza tra l'energia totale e la sommatoria degli autovalori, che d'ora in poi assumeremo essere ordinati in ordine decrescente

$$\lambda_1 > \lambda_2 > \dots > \lambda_n > 0$$

indicando con λ_1 l'autovalore *dominante*.

La convergenza dell'espansione (6.1) richiede che sia soddisfatto il limite

$$\lim_{n \rightarrow \infty} \lambda_n = 0$$

È stato dimostrato che ogni autovalore λ_n equivale alla varianza della corrispondente variabile casuale Z_n e che la media di queste variabili è zero. Con l'aumentare di n , la loro deviazione standard $\pm\sqrt{\lambda_n}$ decresce attorno alla media; di conseguenza si può scrivere

$$\lim_{n \rightarrow \infty} Z_n = 0$$

L'andamento delle variabili Z_n è visibile in Figura 27.

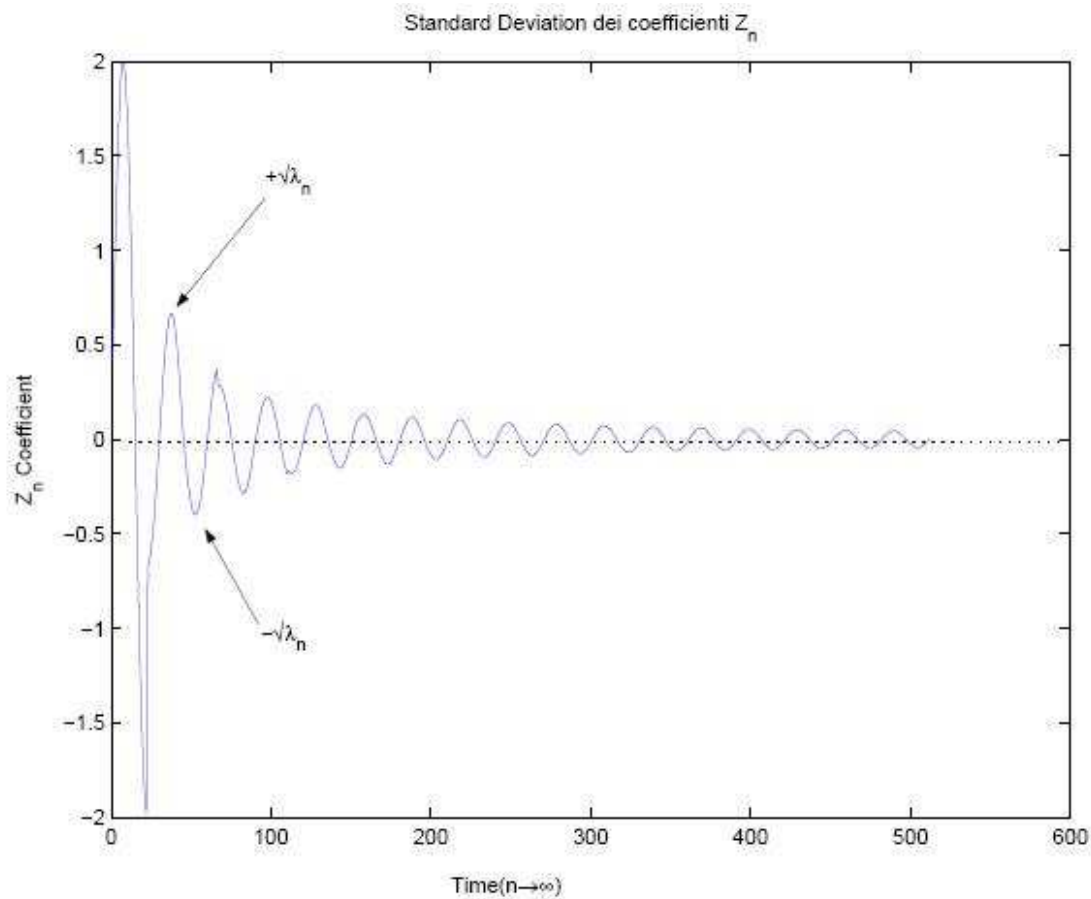


Figura 27 Varianza dei coefficienti Z_n

6.3 Il calcolo della KLT nel caso discreto

È buona norma procedere con l'analisi matematica partendo dal caso continuo poichè è il miglior modo per dimostrare proprietà che altrimenti sarebbero di difficile comprensione, ma il passo successivo è quello di utilizzare quanto detto nell'ambito discreto poichè sarà quello in cui tale trasformata verrà utilizzata.

Sia $X(t)$ un segnale stazionario costituito da un vettore contenente N campioni di cui si vuole calcolare la KLT, ovvero lo sviluppo in serie $X(t) = \sum_{n=1}^{\infty} Z_n \phi_n(t)$. L'operazione in questione viene effettuata calcolando la funzione di autocorrelazione di $X(t)$.

Definizione 1. Si definisce autocorrelazione il valore atteso del prodotto di due campioni del segnale $X(t) \in \mathbb{R}$ rispettivamente ai tempi t_1 e t_2 :

$$R_{XX}(t_1, t_2) = E\{X(t_1)X(t_2)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f_{X(t_1), X(t_2)}(x_1, x_2) dx_1 dx_2$$

che, nel caso discreto, sostituendo la variabile t con n e gli integrali con sommatorie di opportuni indici, diventa

$$R_{XX}(n_1, n_2) = E\{X(n_1)X(n_2)\} = \sum_{x_1=0}^T \sum_{x_2=0}^T x_1 x_2 f_{X(n_1), X(n_2)}(x_1, x_2) \quad (6.11)$$

Se il processo è stazionario allora la funzione di autocorrelazione diventa funzione della sola variabile $m = |n_2 - n_1|$

$$R_{XX}(m) = E\{X(n+m)X(n)\} \quad (6.12)$$

Un'altra proprietà dell'autocorrelazione è la sua simmetria

$$R_{XX}(m) = R_{XX}(-m);$$

di conseguenza l'autocorrelazione ottenuta sarà costituita da un vettore di lunghezza $(2N - 1)$, con $(N - 1)$ elementi ridondanti nella forma

$$R_{xx}(m) = [R_{xx}(1), R_{xx}(2), \dots, R_{xx}(N - 1), R_{xx}(N), R_{xx}(N - 1), \dots, R_{xx}(1)]$$

Il passo successivo per il calcolo della trasformata KL del segnale $X(t)$ è la costruzione della matrice di autocorrelazione, che altro non è che una matrice di Toeplitz costruita con gli elementi non ridondanti del vettore di autocorrelazione.

La matrice T risultante è:

$$T = \begin{bmatrix} R_{xx}(1) & R_{xx}(2) & \dots & \dots & R_{xx}(N) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(2) & \dots & R_{xx}(N-1) \\ \vdots & R_{xx}(2) & R_{xx}(1) & R_{xx}(2) & \vdots \\ \vdots & \dots & R_{xx}(2) & R_{xx}(1) & R_{xx}(2) \\ R_{xx}(N) & R_{xx}(N-1) & \dots & R_{xx}(2) & R_{xx}(1) \end{bmatrix} \quad (6.13)$$

Vengono poi calcolati gli autovalori λ_i della matrice T e i relativi autovettori ϕ_i . I primi compongono il vettore $\vec{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_N]$ mentre gli autovettori disposti per colonne danno origine alla matrice V .

$$V = \begin{bmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_N(1) \\ \phi_1(2) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \phi_1(N) & \phi_2(N) & \dots & \phi_N(N) \end{bmatrix}$$

Il calcolo degli autovettori ed autovalori rappresenta il passo più importante dal punto di vista del tempo di calcolo e delle risorse computazionali richieste. La dimensione della matrice di autocorrelazione è quadratica rispetto alla lunghezza del segnale in analisi; poichè i vettori contenenti i segnali analizzati in questa tesi sono estremamente lunghi, è stata subito evidente l'impossibilità di calcolare tutte le coppie autovalore-autovettore, anche considerando solo piccole porzioni di segnale. Per risolvere almeno in parte questo problema, è stato necessario ricorrere ad algoritmi per il calcolo approssimato degli autovalori più significativi (con modulo maggiore), come spiegato nel capitolo 7.

Gli autovettori ϕ_i così trovati formano una base ortonormale nello spazio di Hilbert; essi sono la migliore base possibile per descrivere il segnale rumoroso $X(t)$. Gli autovalori corrispondenti invece sono legati allo sviluppo della serie (6.1) dalla relazione con le variabili Z_i (6.7): sono infatti le varianze delle variabili casuali a media nulla che vengono moltiplicate con gli autovettori per ottenere la KLT.

Come per la PCA, si mantengono solo le prime K coppie autovalore-autovettore, contenenti la maggior parte dell'informazione legata al segnale, e si scartano le restanti $N - K$ con l'obiettivo di non considerare buona parte del rumore nella ricostruzione del segnale. Per decidere il valore K di autovalori da considerare, è possibile ricorrere ai metodi usati per la PCA, oppure, analizzando l'*eigen-spectrum* generato dagli autovalori, scegliere caso per caso il valore di soglia che divide gli autovalori legati al segnale da quelli considerati rumorosi.

Una volta trovati gli autovettori ϕ_i , costruite le variabili Z_i a partire dalle varianze λ_i e scelto il numero di componenti da considerare, si può calcolare l'espansione KL che ricostruisce il segnale $X(t)$ senza considerare buona parte del rumore ad esso associato.

$$\tilde{X}(t) = \sum_{n=1}^K Z_n \phi_n(t) \quad (6.14)$$

6.4 L'eigen spectrum

Definizione 2. Sia $\vec{\lambda}$ il vettore avente componenti gli autovalori $\lambda_1, \lambda_2, \dots, \lambda_N$; definiamo l'eigen spectrum il grafico dell'andamento di tali valori ordinati in maniera decrescente.

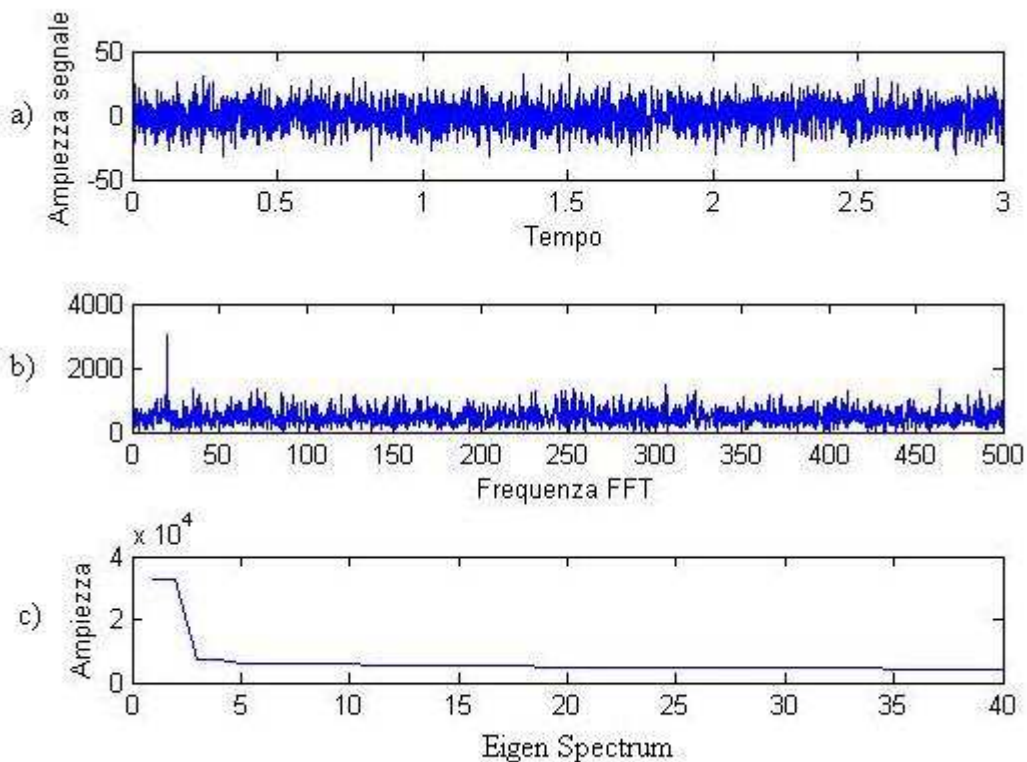


Figura 28 Eigen Spectrum per una sinusoide rumorosa.

Se il SNR è sufficientemente alto, l'eigen spectrum evidenzia un andamento decrescente nel quale le primissime componenti hanno dei valori molto più alti rispetto alle altre, che quindi possono essere assunte come responsabili del carattere rumoroso del segnale.

In Figura 28.a è visualizzato l'andamento nel tempo di una sinusoide con frequenza fondamentale a 20 Hz sommata a rumore bianco gaussiano (AWGN) fino ad avere un SNR pari a 0.02; in 28.c è visibile l'andamento dell'eigen Spectrum per i primi 40 autovalori.

Il problema consiste nello scegliere un valore di soglia opportuno al di sopra del quale si assume che gli autovalori siano relativi al segnale; quanto maggiore è il SNR, tanto più netta è la distinzione tra autovalori dominanti e autovalori relativi al rumore, di conseguenza la scelta della soglia è più semplice. Anche nel caso di un SNR basso come quello della sinusoide in Figura 28, la scelta del valore di soglia è immediato.

In Figura 29 è visibile la differenza tra lo spettro della sinusoide non processata e lo spettro della KLT costruita sulle prime due autofunzioni, i cui autovalori corrispondenti sono maggiori del valore di soglia. Le componenti spettrali causate dal rumore sono state eliminate, poichè l'informazione a loro relativa è contenuta nelle coppie autovalore-autovettore di peso minore; la frequenza fondamentale del segnale invece è ben evidenziata. L'utilizzo della KLT per trovare il periodo del segnale non è necessario in questo caso; ulteriori test, che verranno riportati nel capitolo seguente, mostrano come in certi casi la KLT permetta di individuare componenti spettrali altrimenti invisibili a causa del SNR troppo basso.

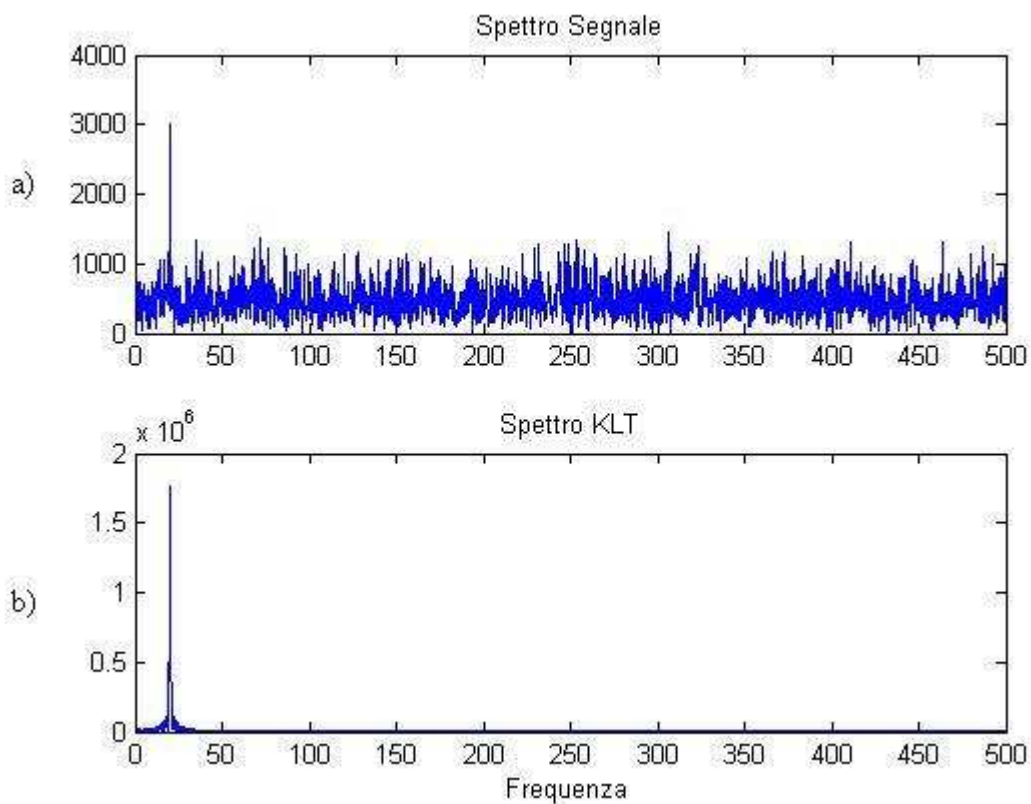


Figura 29 Analisi spettrale del segnale (a) e della KLT del segnale (b).

In Figura 30 invece si possono vedere gli eigen spectrum ottenuti dall'elaborazione di piccole porzioni (pari a pochi secondi) dei segnali riguardanti le tre pulsar analizzate in questa tesi. Si può notare come l'eigen spectrum di Crab abbia una discesa netta dopo i primi autovalori; nel caso di PSR 0540-69 e Vela invece, l'energia del segnale è spalmata su tutti gli autovalori a causa del SNR troppo basso.

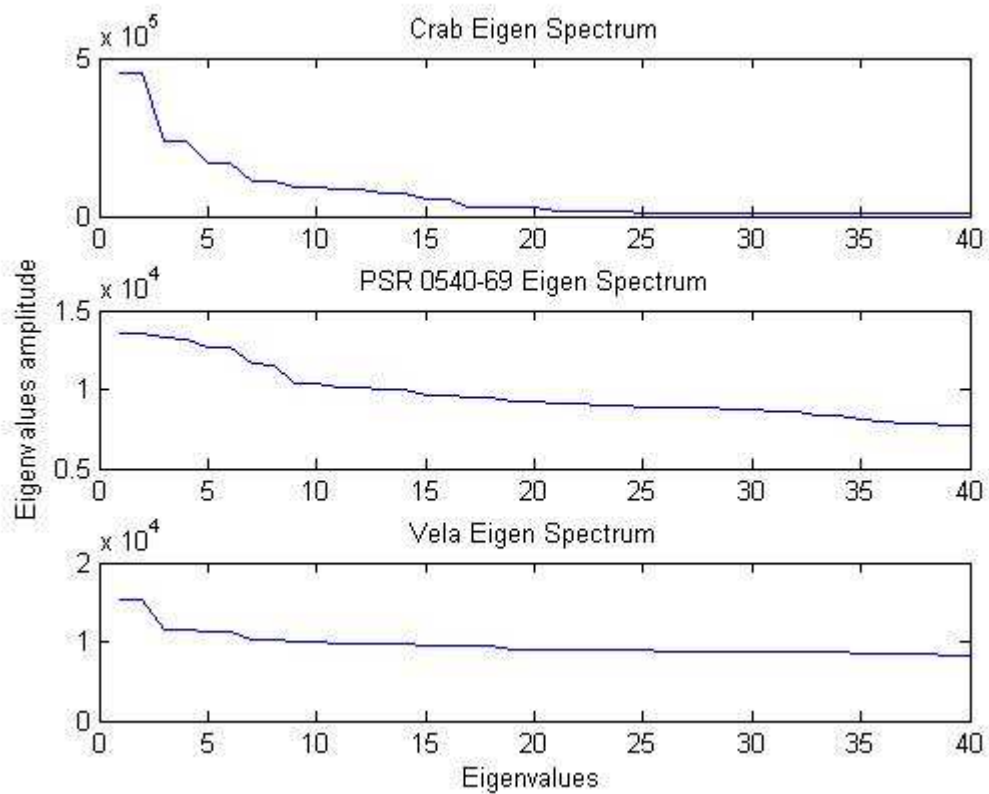


Figura 30 Eigen Spectrum per 5 secondi (N=5000), di segnale

Per scegliere il valore di soglia in caso di indecisione, si può utilizzare il metodo citato da R. Dixon [18], secondo la formula

$$M_{KLT} = \frac{2 \max(\lambda_i)}{N R_{xx}(N)} \quad (6.15)$$

dove con N si intende il numero di elementi nel frame dati, $R_{xx}(N)$ è l'elemento centrale del vettore di autocorrelazione e ne rappresenta l'energia, e $\max(\lambda_i)$ è l'autovalore di modulo maggiore. Il rapporto segnale rumore è legato direttamente all'equazione (6.15): tanto più grande è il SNR, tanto maggiore è la sensibilità dell'equazione alla presenza di rumore.

CAPITOLO 7

Autovalori ed autovettori

7.1 Introduzione al problema

Il problema della ricerca degli autovalori e autovettori interessa vari campi di applicazione, dallo studio della stabilità di strutture, ai problemi di statistica, dai problemi di fisica (stati dell'energia) alla propagazione dei segnali. Il calcolo degli autovalori è un problema di tipo non lineare e quindi per la loro risoluzione si potrebbero usare tecniche risolutive per sistemi non lineari. Il calcolo della KLT richiede appunto la soluzione di un problema di questo tipo: infatti, come già citato in precedenza, quello che si vuole ottenere è un sistema di assi privilegiati che descrivano il fenomeno in analisi nella maniera migliore possibile.

Occorre precisare però che non si intende utilizzare tutti gli autovalori, poichè nel nostro caso il loro numero è estremamente grande ed il calcolo è sostanzialmente impossibile anche con i più potenti computer attualmente disponibili. Quindi ci limiteremo a calcolare solo i K autovalori di valore maggiore, scegliendo opportunamente K in base alle nostre esigenze. Con un numero ridotto di autovalori rispetto al totale si può avere una buona ricostruzione del segnale, e per individuare la presenza di un segnale può essere sufficiente calcolarne anche uno soltanto.

Questo modo di procedere ha portato alla ricerca di algoritmi di estrazione di pochi, più grandi, autovalori piuttosto che all'estrazione di tutti gli autovalori di una matrice. In tal senso la ricerca ha condotto all'utilizzo dell'algoritmo generale di Arnoldi che, per matrici simmetriche come in questo caso, diventa l'algoritmo di Lanczos [16]. Questo algoritmo permette l'abbattimento dei tempi di calcolo in maniera considerevole e rende possibile quindi l'elaborazione di una porzione di dati maggiore.

7.2 Soluzione del problema

Si dà la definizione formale del problema come segue, tenendo presente che per matrici simmetriche gli autovalori sono reali

Definizione 3. Si chiama autovalore della matrice A uno scalare $\lambda \in \mathbb{R}$ per il quale esiste un autovettore $x \in \mathbb{R}^N$ non nullo tale che

$$Ax = \lambda x \quad (7.1)$$

inoltre possiamo anche dire che se esiste una matrice P non singolare tale che:

$$P^{-1}AP = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n] \equiv: D \quad (7.2)$$

allora la matrice A è detta diagonalizzabile. Le colonne di P sono gli autovettori di A e i corrispondenti autovalori sono i λ_i della matrice diagonale D .

7.3 Metodo delle potenze

Per spiegare il metodo adottato nello sviluppo di questa tesi occorre presentare l'algoritmo base che mostra in maniera più chiara il concetto. Si tratta di un metodo iterativo che è adatto ad approssimare l'autovalore di modulo massimo ed il relativo autovettore. Supponiamo che una matrice qualsiasi $A \in \mathbb{R}^{N \times N}$ sia diagonalizzabile, cioè che esista $X^{-1}AX = \text{diag}(\lambda_1, \dots, \lambda_N)$ tale che $X = [x_1, \dots, x_N]$; allora, dato $q^{(0)} \in \mathbb{R}^N$, il metodo delle potenze produce le successioni $q^{(k)}$ come segue:

for $k = 1, 2, \dots$

$$z^{(k)} = Aq^{(k-1)}$$

$$q^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$$

$$\lambda^{(k)} = [q^{(k)}]^T Aq^{(k)}$$

end

Poichè $q^{(0)} \in \mathbb{R}^N$ si può scrivere $q^{(0)} = a_1 x_1 + a_2 x_2 + \dots + a_n x_n$; supponendo poi che $a_1 \neq 1$ e tenendo presente che per A^k si ha che gli autovalori sono elevati della rispettiva potenza k , allora otteniamo

$$A^k q^{(0)} = a_1 \lambda_1^k \left(x_1 + \sum_{j=2}^N \frac{a_j}{a_1} \left(\frac{\lambda_j}{\lambda_1} \right)^k x_j \right) \quad (7.3)$$

Poichè si ha $\frac{\lambda_j}{\lambda_1} < 1$ per $j \geq 2$, allora la direzione di $q^{(k)}$ tende a quella di x_1 . In conclusione, il metodo delle potenze consiste in una modifica iterativa delle componenti del vettore di partenza q_0 in modo da convergere all'autovettore dominante⁵ ottenendo anche il relativo autovalore. In questo caso l'indice k indica l'iterazione e non la k -esima componente. La convergenza dipende dal rapporto tra l'autovalore dominante λ_1 e λ_2 . A questo punto però si può fare un'osservazione: questo metodo calcola un solo autovalore/autovettore per volta, la convergenza può essere estremamente lenta o addirittura il metodo può non raggiungere la convergenza. Occorre quindi un algoritmo che calcoli più autovalori (parametrizzabili come numero) e che abbia una buona convergenza.

7.4 Il metodo di Jacobi

Il metodo di Jacobi, molto popolare per matrici di piccole dimensioni, approssima tutti gli autovalori di una matrice simmetrica utilizzando trasformazioni ortogonali di tipo "rotazione". Ad ogni iterazione viene annullato un elemento al di fuori della diagonale principale. Il principio del metodo è il seguente

$$A_k = G_k^T A_{k-1} G_k \quad (7.4)$$

⁵ Per autovettore dominante si intende l'autovettore associato all'autovalore di modulo massimo.

dove G_k è una matrice di rotazione di angolo θ che definisce i seguenti scalari $c = \cos\theta$, $s = \sin\theta$. Per la costruzione della A_k allora devono essere calcolate le seguenti operazioni:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)}, \quad \text{per } i \neq p, q; j \neq p, q$$

$$a_{pj}^{(k)} = ca_{pj}^{(k-1)} - sa_{qj}^{(k-1)}, \quad \text{per } j \neq p, q$$

$$a_{qj}^{(k)} = sa_{pj}^{(k-1)} + ca_{qj}^{(k-1)}, \quad \text{per } j \neq p, q$$

$$a_{pq}^{(k)} = cs(a_{pp}^{(k-1)} - a_{qq}^{(k-1)}) + (c^2 - s^2)a_{pq}^{(k-1)}$$

$$a_{pp}^{(k)} = c^2 a_{pp}^{(k-1)} + s^2 a_{qq}^{(k-1)} - 2cs a_{pq}^{(k-1)}$$

$$a_{qq}^{(k)} = s^2 a_{pp}^{(k-1)} + c^2 a_{qq}^{(k-1)} + 2cs a_{pq}^{(k-1)}$$

Poichè fra un'iterazione ed un'altra alcuni elementi che erano stati annullati possono ridiventare non nulli, tale metodo è del tipo "a terminazione non finita". Per questo motivo esistono vari criteri per la scelta dell'elemento da annullare fra i quali il criterio classico che elimina l'elemento di modulo massimo e il metodo con soglia. Il metodo di Jacobi però è molto lento su matrici di grandi dimensioni ed approssima tutti gli autovalori, motivi per cui, nel nostro caso, è inapplicabile.

7.5 Il metodo di Lanczos

L'algoritmo di Lanczos è quello che risponde al meglio alla nostra esigenza, in quanto consente una buona convergenza e una ricerca di un numero di autovalori parametrizzabile. Questo metodo permette di ottenere una fattorizzazione della matrice A . Come si può vedere in Figura 31, la matrice in ingresso viene decomposta dall'algoritmo: V contiene i k autovettori disposti per colonne, mentre una importante proprietà della matrice tridiagonale H è che i suoi autovalori sono una buona approssimazione dei k autovalori di modulo maggiore di A .

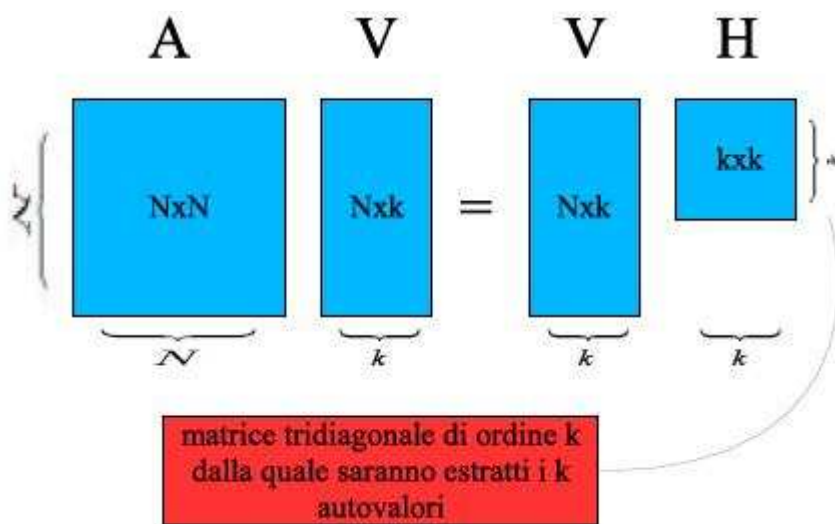


Figura 31 Fattorizzazione tramite Lanczos

Sia $A^T = A \in \mathbb{R}^{N \times N}$ e indichiamo con $\lambda_i(A)$ l' i -esimo autovalore più grande di A ; allora si ha $\lambda_N(A) \leq \lambda_{N-1} \leq \dots \leq \lambda_2(A) \leq \lambda_1(A)$. Per un noto teorema di Rayleigh si ha che gli autovalori di una matrice simmetrica sono i punti stazionari della funzione che mappa $x \rightarrow x^T A x / x^T x$ con $x \neq 0$; inoltre essi soddisfano il seguente teorema:

Teorema 3. (Courant-Fischer Minimax Theorem). Se $A \in \mathbb{R}^{N \times N}$ è simmetrica, allora

$$\lambda_k(A) = \max_{\dim(S)=k} \min_{0 \neq y \in S} \frac{y^T A y}{y^T y} \quad (7.5)$$

per $k=1 \dots n$.

dove con S si intende il sottospazio invariante di dimensione k . Senza addentrarci troppo nella teoria, poichè molto vasta, possiamo dire che il metodo di Lanczos genera una successione di vettori ortonormali che costituisce il sottospazio invariante S appena sopra citato.

Se nel teorema di Courant-Fischer assumiamo che i vettori y siano unitari allora noteremo come il numeratore è una trasformazione di similarità su uno spazio 1-dimensionale. A questo punto si può descrivere brevemente l'iterazione Lanczos

$$r_0 = q_1; \beta_0 = 1; q_0 = 0; j = 0$$

while ($\beta_j \neq 0$)

$$q_{j+1} = r_j / \beta_j; j = j + 1; \alpha_j = q_j^T A q_j;$$

$$r_j = (A - \alpha_j I) q_j - \beta_{j-1} q_{j-1};$$

$$\beta_j = \|r_j\|_2$$

end

Il caso più fortunato che si può avere è quello in cui si ottiene, dopo N passi, $\beta_N = 0$: ciò sta ad indicare che si ha uno spazio invariante di dimensione N dove le basi sono date dai q_j . Ponendo poi

$$Q = [q_1, q_2, \dots, q_n]$$

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & \beta_{n-1} & \alpha_n \end{pmatrix} \quad (7.6)$$

si è ottenuta la tridiagonalizzazione di A ed i vettori colonna q_i costituiscono la matrice Q, si può notare anche che

$$T = Q^T A Q \quad (7.7)$$

premultiplicando a sinistra, e ricordando che per matrici ortogonali si ha che $Q^T Q = I$, otteniamo

$$A Q = Q T \quad (7.8)$$

È importante notare che l'algoritmo di Lanczos soffre di grandi problemi di stabilità numerica quando $\beta_j \rightarrow 0$ causando la non ortogonalità delle basi del sottospazio invariante. Questo problema diventa tanto più evidente quando la dimensione di S diventa grande. Nel nostro caso, poichè la dimensione del sottospazio invariante che cerchiamo è molto piccola (rispetto al numero di punti N del frame-dati), si è calcolata la completa ri-ortogonalità delle basi ottenute. A questo punto si può riscrivere la fattorizzazione in modo più completo come segue

$$A V = V H + f e_k^T \quad (7.9)$$

dove $V \in R^{N \times k}$, $V^T = I_k$, $H \in R^{k \times k}$ tridiagonale simmetrica e $f \in R^N$ con $0 = V^T f$. L'ultimo termine dell' eq. (4.9) è direttamente correlato alla distanza rispetto un sottospazio invariante.

Questa fattorizzazione è ottenuta dal seguente algoritmo

Input $\rightarrow (A, v_0)$

Output $\rightarrow (V_k, H_k, f_k)$

Begin

$$v_1 \leftarrow v_0 / \|v_0\|$$

$$w \leftarrow Av; \alpha_1 = v_1^T w$$

$$H_1 = (\alpha_1); V_1 = (v_1); f_1 \leftarrow w - v_1 \alpha_1$$

for $j = 1, \dots, k - 1$

$$\beta_j = \|f_j\|; v_{j+1} \leftarrow f_j / \beta_j$$

$$V_{j+1} = (V_j, v_{j+1}); H_j \leftarrow \begin{pmatrix} H_j \\ \beta_j e_j^t \end{pmatrix}$$

$$h \leftarrow Av_{j+1}$$

$$h \leftarrow V_j^T z; H = (H_j, h)$$

$$f \leftarrow z - V_{j+1} h;$$

end

end

CAPITOLO 8

Test e valutazioni

8.1 Introduzione

Durante il lavoro di ricerca sono state trovate e considerate promettenti le due tipologie di trasformate di Karhunen-Loève illustrate nei capitoli 5 e 6. La Principal Component Analysis (Cap. 5) è la più diffusa ed utilizzata, di conseguenza è disponibile su di essa una sufficiente documentazione che conferma la solidità della teoria alla base della trasformata, composta da paper scientifici e libri che ne descrivono svariate applicazioni [7] [8]. Per questo motivo, al momento di decidere quale metodo usare per l'elaborazione dei dati prodotti da IquEye, la PCA è stata la prima scelta. Occorreva tenere in considerazione però che le applicazioni trovate in letteratura non hanno molte analogie con il nostro lavoro: solitamente la PCA viene utilizzata per l'analisi e compressione di immagini, o su dati statistici; in ogni caso essa richiede che i dati in input siano in forma matriciale mentre i dati in nostro possesso sono vettori che rappresentano segnali nel tempo.

Nei paragrafi seguenti è spiegato il metodo usato per l'applicazione della PCA ai dati e ne vengono riportati i risultati. In seguito sono state effettuate ulteriori ricerche fino a trovare materiale riguardante la KLT per signal processing. Questo diverso tipo di trasformata sembrava ancora più promettente della PCA in quanto concepita proprio per l'analisi di un segnale rumoroso come il nostro, tuttavia il metodo appare ancora sperimentale e la teoria non è mai stata applicata interamente prima d'ora; come si vedrà più avanti questo ha generato non poche difficoltà nel tentativo di ripulire il segnale dal rumore.

Sia per la PCA che per la KLT il modo di procedere è stato lo stesso: il punto di partenza sono i dati della pulsar Crab, sui quali è stata applicata la trasformata al fine di trovare un metodo migliore per trovare il periodo corretto di folding o una ricostruzione del segnale con un migliore SNR.

La curva di luce ricavata dai dati della Crab è ottima e confrontabile con i molti risultati disponibili in letteratura, per questo è stata utilizzata come benchmark per testare la qualità dell'output delle trasformate. In caso di risultati incoraggianti, sia in termini di SNR della light curve, sia in termini di algoritmi per la ricerca del periodo di rotazione, il passo successivo è di testare il procedimento prima con PSR B0540-69 e infine con Vela.

8.2 PCA su dati di IquEye

Come già spiegato in precedenza, l'algoritmo per il calcolo delle componenti principali richiede come input una matrice di dati, poichè è impossibile costruire la matrice di covarianza a partire da un unico vettore.

Il modo di procedere più immediato è stato quello di trasformare il vettore contenente i dati di Crab in forma matriciale, binnando i dati con bin temporali di 1 ms e poi dividendo il vettore di lunghezza N in \sqrt{N} sotto-vettori (di lunghezza \sqrt{N} ciascuno) che sono andati a costituire le righe di una matrice quadrata; tale matrice è stata data in input all'algoritmo per il calcolo della PCA. Questo procedimento tuttavia è stato ben presto scartato poichè dalle PC risultanti non è stato possibile ricavare nessuna informazione rilevante: i dati proiettati sulla base costituita dalle PC di varianza maggiore non costituivano un segnale adatto a successive elaborazioni. Il folding dei dati trasformati era impossibile poichè i valori non rappresentavano più il numero di fotoni rivelati ad ogni millisecondo, senza contare che per un folding corretto è necessario come punto di partenza un vettore binnato con un time bin divisore del periodo di rotazione; anche l'analisi con la trasformata di Fourier non mostrava nessun miglioramento rispetto all'analisi dello spettro del segnale non trasformato. Tutto ciò era facilmente prevedibile poichè la PCA non è concepita per l'analisi di un segnale nel tempo e utilizzarla direttamente sul segnale non aveva nessuna base teorica.

Dato che la PCA viene impiegata maggiormente per l'analisi di immagini, si è cercato di ricavare informazioni utili dall'analisi del waterfall diagram in output dall'algoritmo che esegue il folding dei dati.

Come già spiegato in precedenza, il waterfall diagram è un grafico cromatico composto da una serie di N light curve poste una sotto all'altra; ognuna di queste light curve si riferisce a una frazione del segnale originale.

Il calcolo delle PC sulla matrice che costituisce il waterfall diagram della pulsar Crab ha permesso di ricavare alcune informazioni utili. Dopo alcuni test si è scelto di mantenere il numero di PC sufficiente a mantenere il 90 % dell'informazione ed è stato fatto un confronto tra la PCA del waterfall ottenuto con il periodo di rotazione corretto (Fig. 32) e la PCA del waterfall corrispondente al periodo trovato con l'analisi spettrale del segnale (Fig. 33).

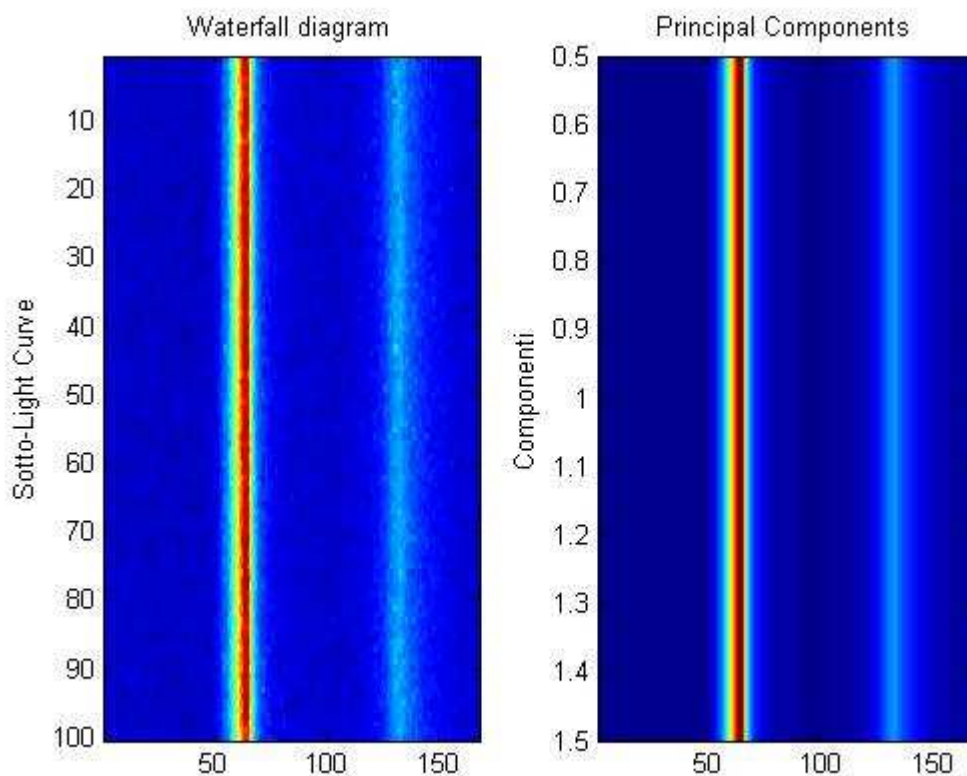


Figura 32 Waterfall diagram per folding dei dati di Crab con periodo esatto (a sinistra) e proiezione del waterfall sulla prima PC (a destra).

Come si può vedere in Figura 32, l'output dell'algoritmo è costituito dalla proiezione della matrice in ingresso sulla prima PC, contenente più del 90 % dell'informazione (circa il 98.5 % della varianza totale).

Diversamente, nel caso la PCA venga applicata al waterfall costruito con il periodo di rotazione trovato grazie all'analisi spettrale dei dati, la varianza viene distribuita maggiormente tra le varie componenti e di conseguenza il numero di PC che vengono considerate aumenta.

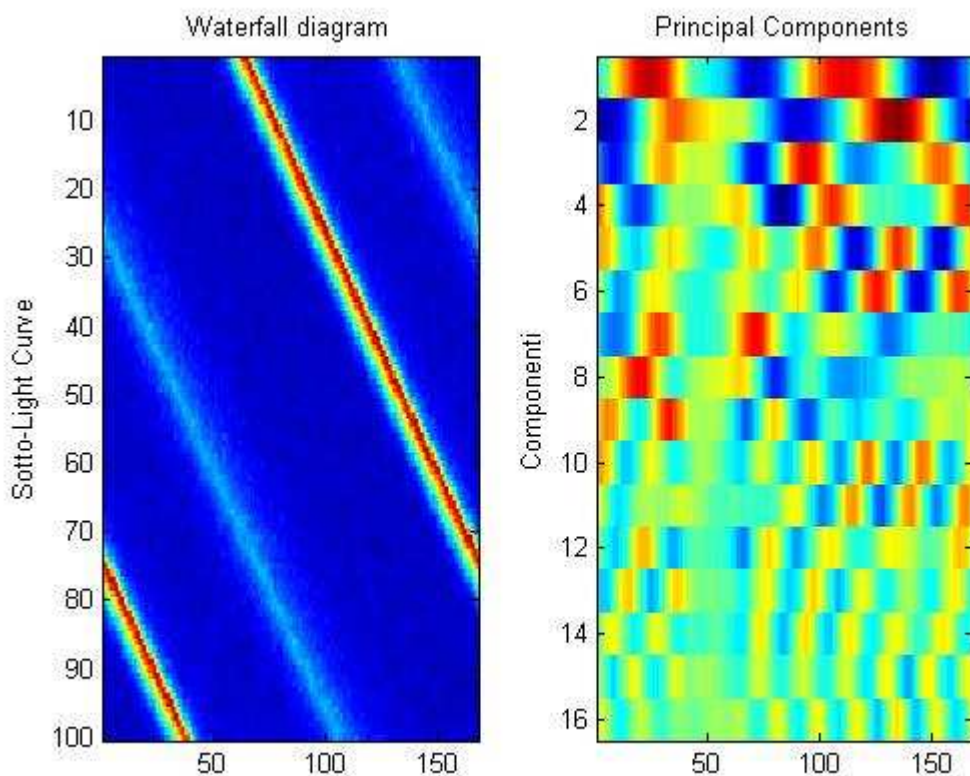


Figura 33 Waterfall diagram per folding dei dati di Crab con periodo trovato con Fourier (a sinistra) e proiezione del waterfall sulle PC (a destra).

Il grafico a destra in Figura 33 mostra come l'algoritmo abbia dovuto utilizzare 16 componenti principali per conservare la percentuale desiderata di varianza.

L'output della PCA, cioè la proiezione del waterfall sulla base ortonormale costituita dalle PC selezionate, non rappresenta in sé un risultato utile al raggiungimento dell'obiettivo del lavoro, e cioè la costruzione della light curve corretta; tuttavia, osservando il comportamento della PCA al variare della matrice in ingresso, è nata l'idea di utilizzare questa trasformazione dei dati per creare un sistema di feedback con lo scopo di trovare il periodo di rotazione esatto.

Alla base del sistema vi è il legame tra la varianza delle PC e la struttura del waterfall. Le righe del waterfall diagram, per costruzione, sono tanto più simili tra di loro quanto più è corretto il periodo utilizzato per il folding dei dati; se il periodo di rotazione è esatto le righe appaiono uguali tra loro e i picchi di segnale (rappresentati da colori più caldi) sono perfettamente allineati verticalmente, di conseguenza la varianza calcolata sulle colonne della matrice è minore rispetto a un waterfall in cui i picchi sono disallineati.

La somiglianza tra le righe del waterfall si riflette anche sulle componenti principali: sono stati sufficienti pochi test per scoprire l'esistenza di una relazione che lega la varianza tra le colonne del waterfall e la distribuzione della varianza nelle PC. La percentuale di varianza totale contenuta nelle prime componenti, quelle legate al segnale, cresce al calare della varianza tra le colonne del waterfall; se il waterfall è costruito con un periodo errato la varianza viene distribuita maggiormente tra le componenti, comprese quelle rappresentate il rumore. Questa proprietà è stato il punto di partenza per lo sviluppo di un algoritmo iterativo in grado di ottimizzare il valore del periodo in ingresso al folding dei dati.

8.3 L' algoritmo *ufolding*

L' algoritmo, chiamato *ufolding* (Fig. 34), utilizza come punto di partenza il periodo di rotazione ricavato dall' analisi spettrale del segnale e si sposta in un intorno di quel valore alla ricerca del periodo per il folding ottimo grazie a una merit function valutata sulle varianze delle PC.

L' input è il vettore contenente i tempi dei fotoni: prima di cominciare il ciclo di iterazioni, l' algoritmo costruisce un vettore distribuendo i dati in time bin di grandezza appropriata (1 ms per Crab e PSR B0540-69 e 4 ms per Vela) e ne calcola la trasformata di Fourier.

Sia p_{FFT} il periodo di rotazione ottenuto grazie alla trasformata di Fourier; nella sua prima iterazione l' algoritmo esegue un normale folding dei dati usando come periodo di rotazione $p_1 = p_{FFT} + k * \Delta t$. Il valore Δt , il cui valore iniziale è stato impostato a $\Delta t = 10^{-8} s$, costituisce lo spostamento nell' intorno del periodo ad ogni iterazione, mentre k è un parametro impostabile a piacere che determina il numero di iterazioni. Il folding con p_1 determina una light curve e un waterfall diagram su N righe, con $N = 100$ per Crab e $N = 20$ per PSR B0540-69 e Vela; l' algoritmo calcola poi la PCA del waterfall e fa la somma delle varianze corrispondenti alle prime h componenti principali, con h variabile a seconda della pulsar in esame (i risultati migliori si sono ottenuti per $h = 3$ durante l' analisi della pulsar Vela), prendendo nota del risultato (σ_{max}) e del periodo di rotazione $p_0 = p_1$.

Alla seconda iterazione ripete il folding dei dati con $p_2 = p_1 - \Delta t$ e calcola la PCA sul waterfall: se la somma delle prime h varianze è maggiore del valore trovato in precedenza, aggiorna il valore σ_{max} e memorizza il periodo $p_0 = p_2$, altrimenti mantiene il vecchio valore di σ_{max} e p_0 per poi iniziare una nuova iterazione.

Dopo $2 * k$ iterazioni, quando il periodo in analisi p_i prende il valore $p_i = p_{FFT} - k * \Delta t$, l' algoritmo interrompe il primo ciclo di iterazioni e ne inizia un secondo, cambiandone i parametri iniziali. Alla prima iterazione del secondo ciclo utilizza $p_1 = p_0 + k * \Delta t$, con $\Delta t = 10^{-10} s$, per aumentare la precisione del valore del periodo p_0 .

Infine viene eseguito un terzo ciclo di iterazioni con $\Delta t = 10^{-12}s$, al termine del quale, viene restituita come output la light curve costruita con il valore del periodo di rotazione p_0 .

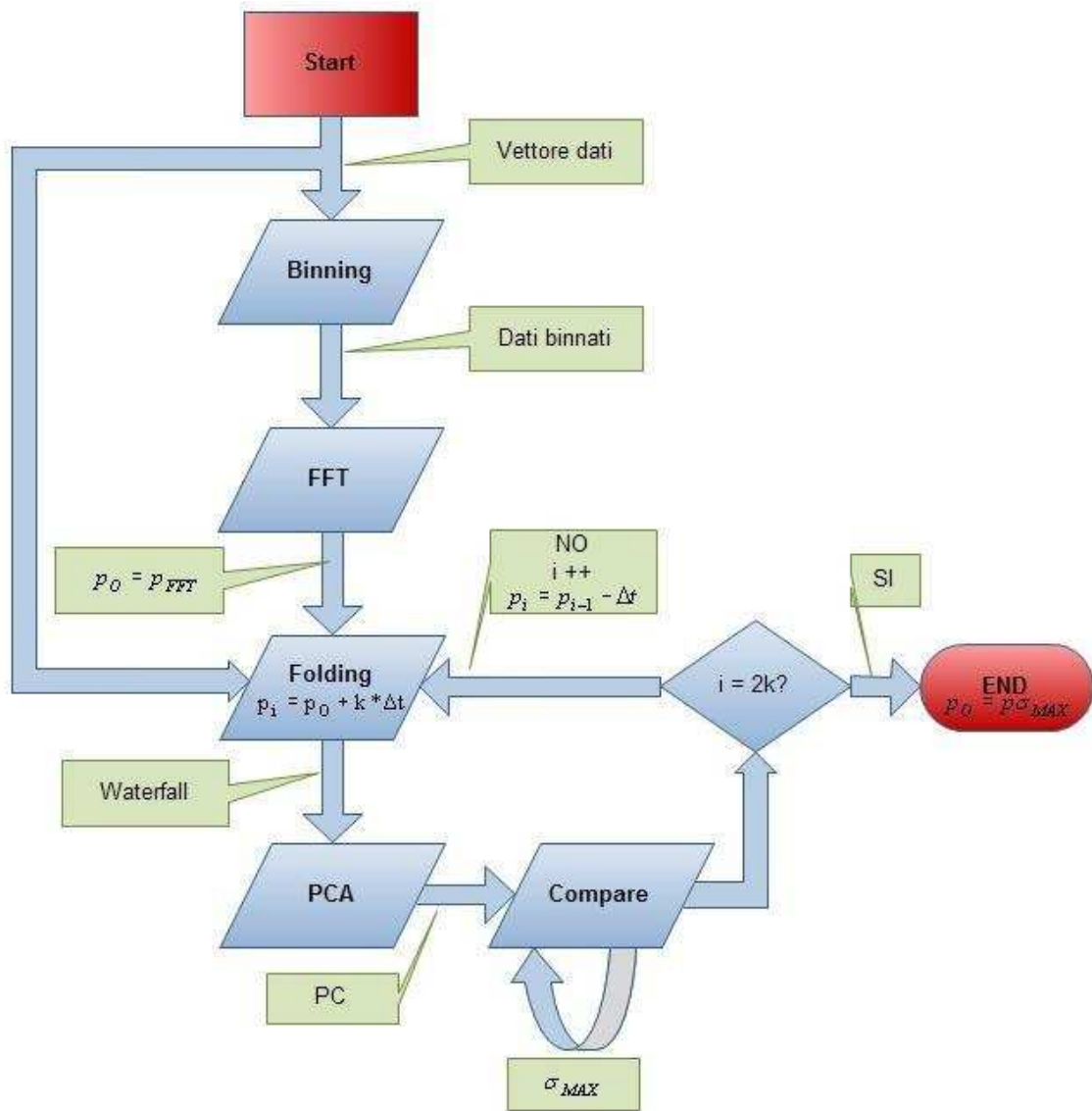


Figura 34 Diagramma a blocchi dell'algoritmo *ufolding*

8.4 Risultati per *ufolding*

I primi test per *ufolding* sono stati eseguiti sui dati di Crab per verificare l'efficacia dell'algoritmo e i risultati sono stati molto promettenti: è stato sufficiente valutare la varianza della prima PC del waterfall diagram ($h = 1$) ed eseguire il folding con il periodo di rotazione associato al massimo valore di varianza per ottenere una light curve identica a quella esatta di Figura 18.

Questo risultato incoraggiante ha permesso di provare *ufolding* sui dati di PSR B0540-69. Sono stati eseguiti numerosi test per trovare i parametri in ingresso che permettono di ottimizzare l'output: a causa del livello di rumore, il numero di righe del waterfall diagram su cui viene calcolata la PCA è stato ridotto da 100 a 20, inoltre si ottengono risultati migliori se la valutazione delle varianze viene eseguita su più componenti invece che su una sola. Il valore migliore per il numero di componenti principali da considerare è $h = 3$, con un numero maggiore di PC si tende a includere il rumore e con un numero inferiore non viene considerata una quantità di segnale sufficiente per stabilire il periodo ottimale.

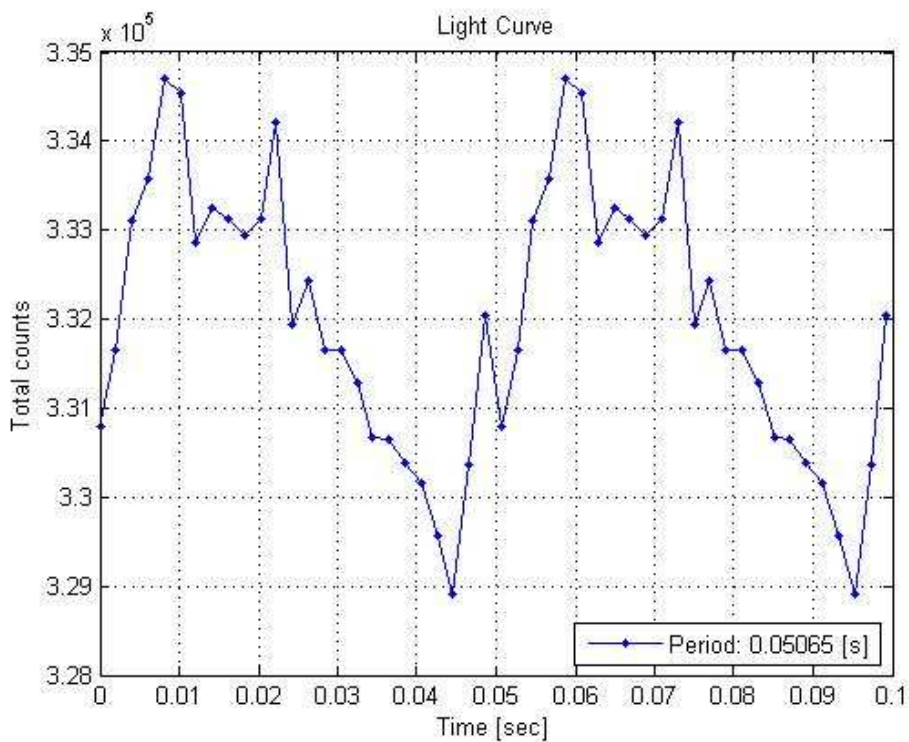


Figura 35 Light curve creata con *ufolding*

In Figura 35 è visibile la light curve su due periodi così ottenuta, il periodo di rotazione trovato è $p = 0.050649972293684$ s. Confrontando il diagramma a barre di questa curva di luce con altre ottenute negli ultimi anno con altre osservazioni, ci si può rendere conto della similitudine tra i risultati (Fig. 36).

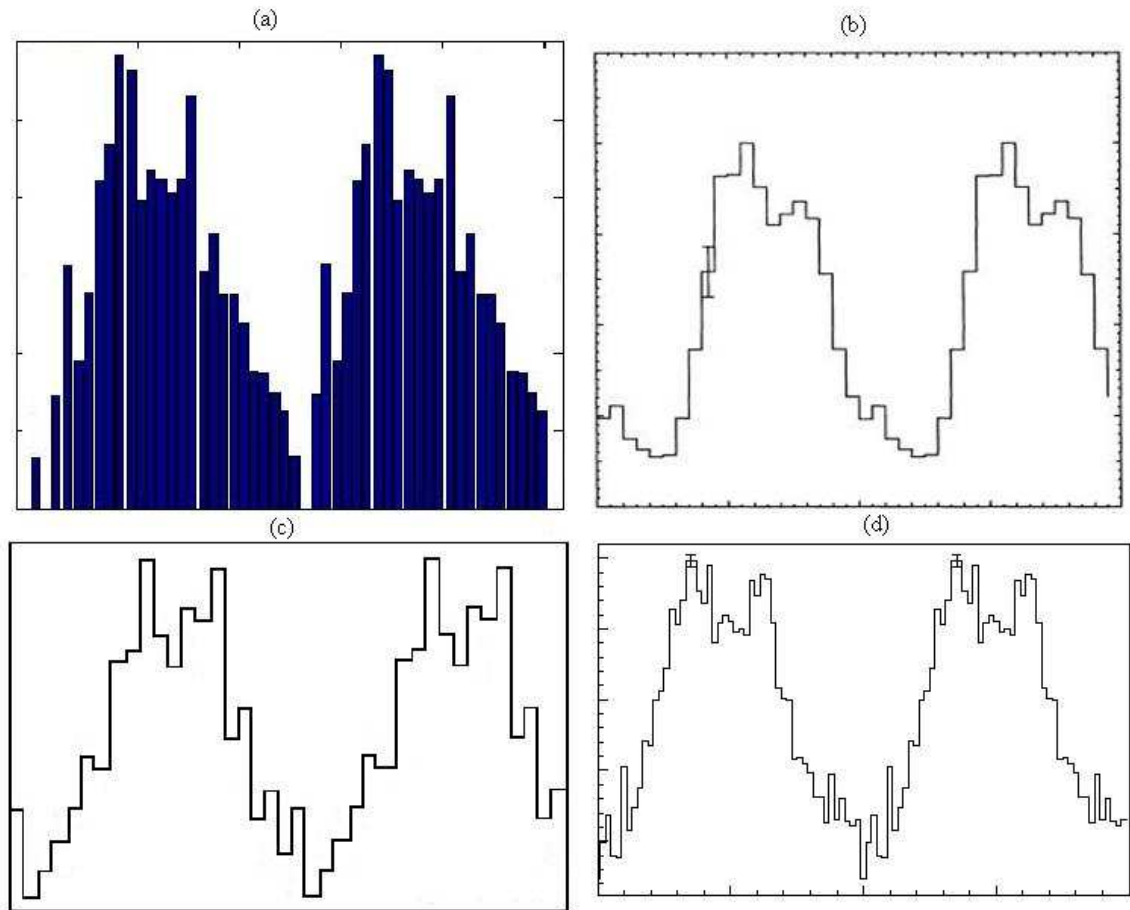


Figura 36 Light curve per *ufolding* (a), Boyd et al. (2005) (b), Ulmer et al. (2005) (c), Gradari et al. (2009) (d)

È interessante osservare l'andamento delle varianze delle componenti principali al variare del periodo considerato: in Figura 37 è riportata la somma delle prime 3 PC durante il primo ciclo di iterazioni (in alto) e durante il secondo (in basso), mentre in Figura 38 sono visibili le singole varianze delle tre PC.

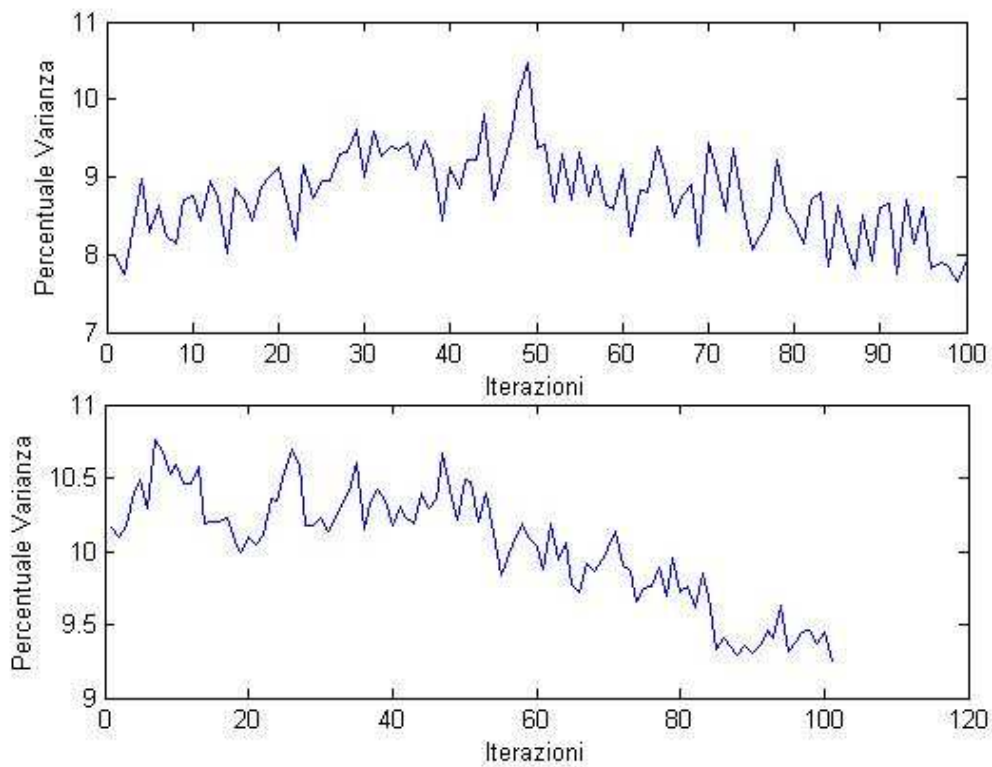


Figura 37 Andamento della somma delle varianze per le prime 3 PC nel primo ciclo di iterazioni (grafico in alto) e nel secondo ciclo (grafico in basso)

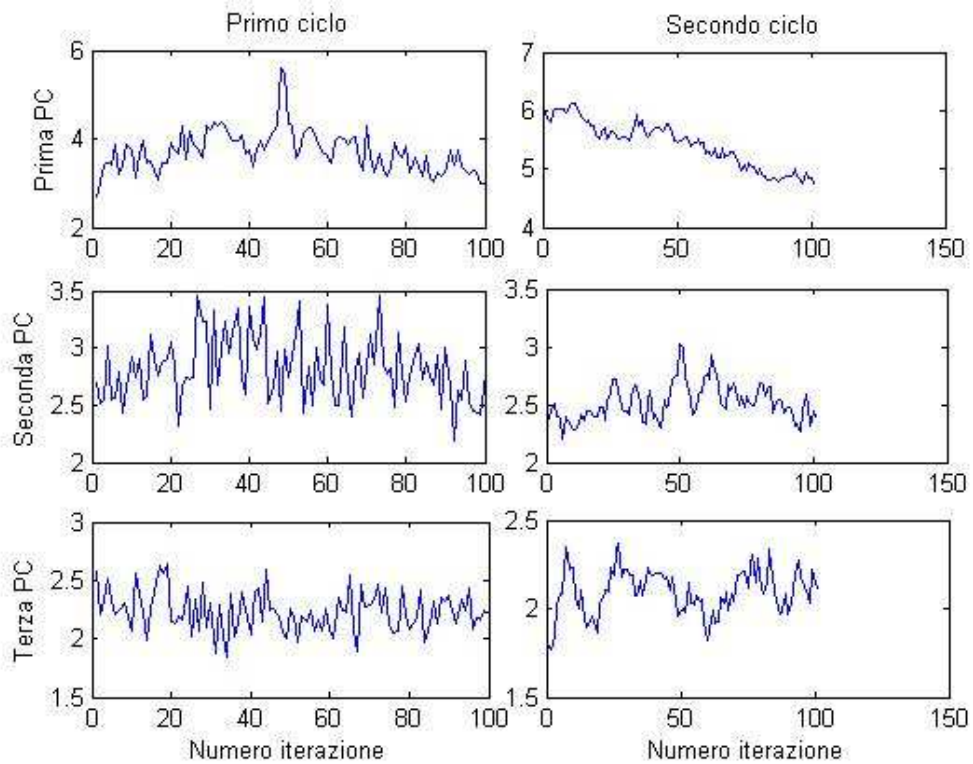


Figura 38 Andamento delle varianze per le singole PC di PSR B0540-69

Osservando la Figura 37 si nota che la crescita della varianza non è costante con l'avvicinarsi all'ottimo globale: il valore oscilla continuamente e sono presenti molti ottimi locali ma si può vedere nel complesso un andamento in salita che culmina nel punto di massimo, nettamente superiore a tutti gli altri. Il valore del periodo migliore è senz'altro dovuto in gran parte alla prima componente che ha un peso maggiore rispetto alle altre, ma nel secondo ciclo di iterazioni sono determinanti anche la seconda e terza PC.

Infine l'algoritmo è stato utilizzato sui dati di Vela pulsar, il vero obiettivo di questo lavoro di tesi. Anche in questo caso i parametri ottimali individuati sono $h = 3$ e un waterfall di 20 righe; il periodo di rotazione che è stato trovato ha portato alla curva di luce in Figura 39.

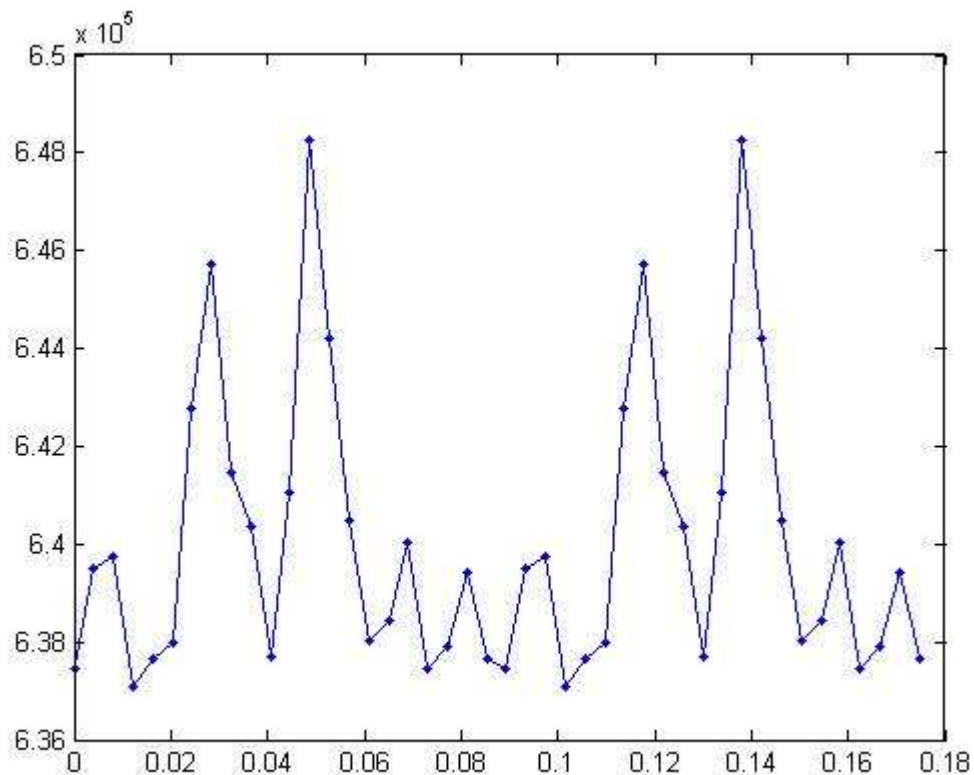


Figura 39 Light curve per vela pulsar, periodo di rotazione $p = 0.089366997345886$ s

I due picchi ben evidenti testimoniano la buona qualità della curva di luce, che è stata confrontata con l'unica altra curva pubblicata fino ad ora [4] (Figura 40).

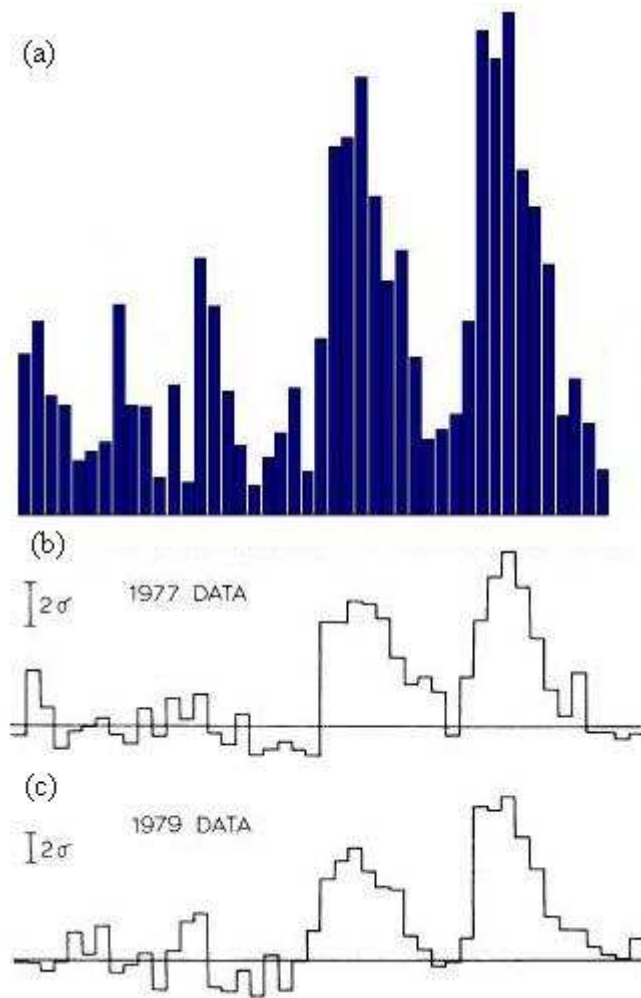


Figura 40 Light curve di Vela pulsar: *unfolding* (a), Wallace, Peterson, Elliott (b) e (c)

La somiglianza tra le curve è evidente, il che porta a pensare che il periodo di rotazione trovato grazie alla PCA sia estremamente buono. Non è possibile avere la certezza che sia la curva migliore ottenibile dai dati a disposizione poichè non esistono informazioni dettagliate sulla pulsar Vela, ma sicuramente il risultato è molto vicino all'ottimo.

L'analisi delle varianze delle componenti principali mostra come la prima componente giochi un ruolo ancora più importante che per PSR B0540-69, tuttavia la presenza della seconda e della terza rendono più preciso il risultato nel secondo ciclo iterativo.

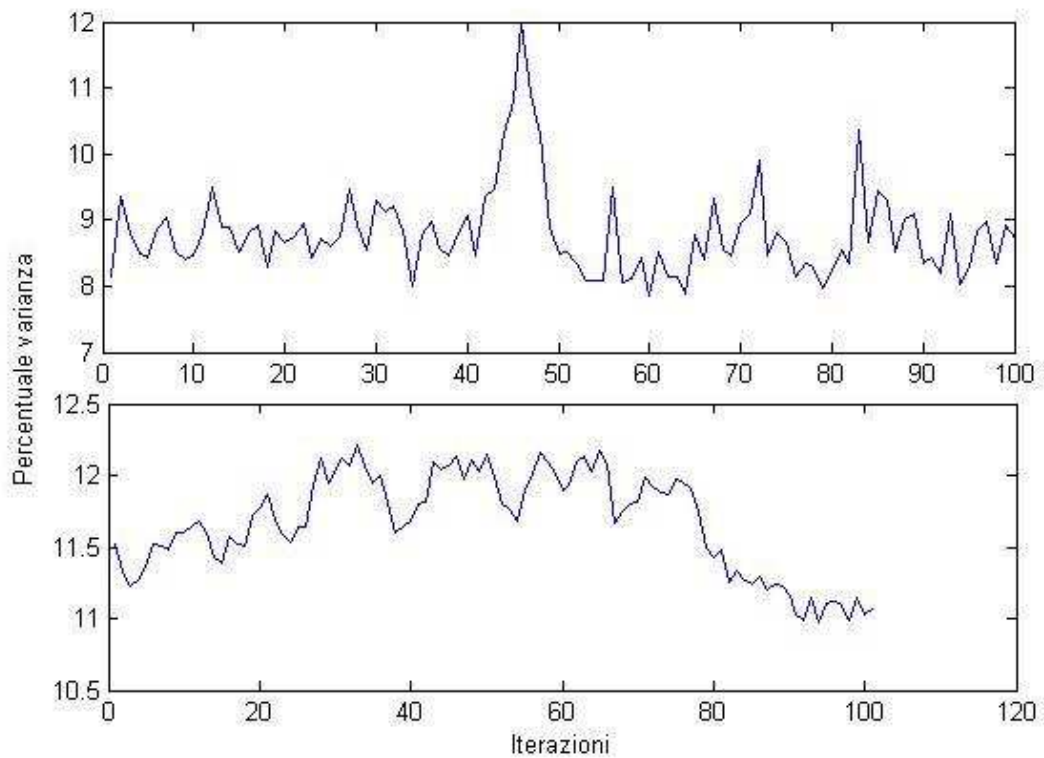


Figura 41 Andamento della somma delle varianze per le prime 3 PC nel primo ciclo di iterazioni (grafico in alto) e nel secondo ciclo (grafico in basso)

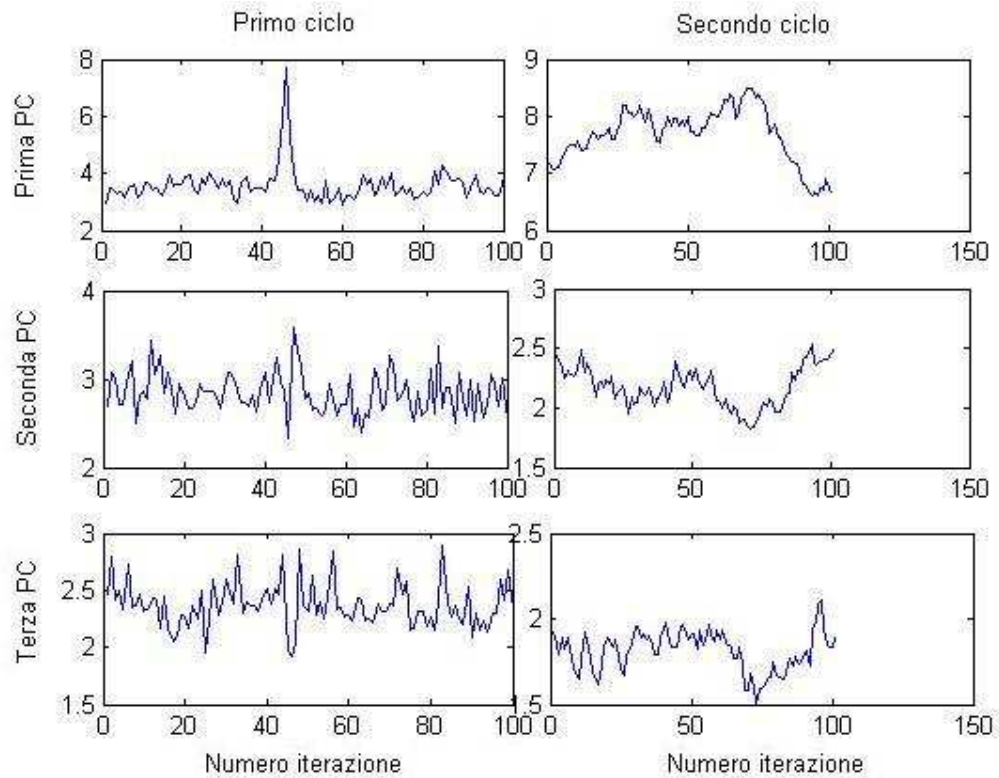


Figura 42 Andamento delle varianze per le singole PC di Vela pulsar

8.5 Calcolo della KLT

La KLT per signal processing, a differenza della PCA, non necessita di una matrice come input ed è quindi stata calcolata direttamente sul vettore contenente i dati binnati; tuttavia nell'applicazione di questa trasformata ai dati in nostro possesso si sono incontrati numerosi problemi.

Fin dall'inizio si è manifestata l'inadeguatezza dei mezzi disponibili rispetto al carico computazionale necessario. L'algoritmo per la KLT esegue l'autocorrelazione del segnale e la utilizza per costruire una matrice di Toeplitz; se il segnale è un vettore costituito da N elementi, l'autocorrelazione è un vettore di lunghezza $2N - 1$ e la matrice di Toeplitz contenente gli N elementi non ridondanti dell'autocorrelazione è una matrice $N \times N$. Considerando che le osservazioni delle pulsar sono rispettivamente di 30 minuti per Crab, 60 minuti per PSR B0540-69 e 180 minuti per Vela, ci si rende conto immediatamente che i vettori contenenti i dati suddivisi in time bin di 1 ms (per Crab e PSR B0540-69) e 2-4 ms (per Vela) hanno un numero di elementi dell'ordine $N = 10^6$. Vettori di questa lunghezza portano ad avere matrici di Toeplitz che contengono $N * N = 10^{12}$ elementi di tipo double e, utilizzando 64 bit per ognuno di essi, lo spazio in memoria necessario a contenere l'intera matrice diventa $64 * 10^{12} \simeq 2^{46} \text{ bit} \simeq 7.2 \text{ TB}$; questo valore in realtà è sottostimato se si considera, ad esempio, il vettore binnato per PSR B0540-69, di lunghezza $6 * 10^6$, che porta ad una matrice per la quale servono 262 TB di memoria.

Anche ipotizzando di poter ignorare i problemi di spazio in memoria, un altro ostacolo è la quantità di tempo necessaria a trovare le coppie autovalore-autovettore di una matrice di queste dimensioni. La fattorizzazione di Lanczos, che è stata inserita nell'algoritmo per il calcolo della KLT, riduce di molto i tempi di calcolo ed effettua una buona approssimazione degli autovalori, ma è effettiva con matrici di dimensioni contenute ($N = 10^3 - 10^4$); con una quantità di dati così grande da processare il tempo resta il problema principale indipendentemente dalla qualità dell'algoritmo utilizzato.

Vista l'impossibilità di calcolare la KLT sui dati di un'intera osservazione, si è deciso di effettuare i test su porzioni ridotte dei nostri segnali, rappresentative di pochi secondi di

osservazione, e su altri segnali costruiti per l'occasione con lo scopo di verificare l'efficacia del metodo.

L'ultimo è più grande problema incontrato è dovuto alla teoria su cui si basa la trasformata stessa. Gli elementi che permettono la ricostruzione del segnale sono le autofunzioni $\phi_n(t)$ ricavate dalla matrice di autocorrelazione e i valori Z_n . Il problema consiste nel determinare i valori Z_n che, stando alla teoria del Prof. Maccone [2], sono variabili casuali a media nulla e varianza pari a $\sqrt{\lambda_n}$. Se, nel calcolo della trasformata, si costruiscono gli Z_n secondo queste specifiche, il risultato sono delle variabili casuali estremamente rumorose, poichè gli autovalori dominanti calcolati sulla matrice di autocorrelazione sono molto grandi in modulo. La composizione della KLT utilizzando questi Z_n e le autofunzioni non genera una ricostruzione del segnale originale, ma piuttosto un segnale che sembra costituito unicamente da rumore dal quale non si è riuscito a ricavare nessuna informazione utile.

Non è stato possibile verificare se ci sono stati errori nell'applicazione della trasformata o se la teoria stessa è inesatta, poichè non esistono pubblicazioni in cui tale trasformata è stata calcolata completamente; gli unici lavori che citano il lavoro del Prof. Maccone ([6] e [9]) riportano le stesse difficoltà nel calcolo dei pesi Z_n delle autofunzioni. Si è quindi deciso di non utilizzare le variabili Z_n , ma di utilizzare come pesi degli autovettori nella sommatoria gli stessi autovalori λ_n . Ciò che ne risulta è una versione tronca della KLT che non porta ad avere una ricostruzione del segnale originale, ma che si rivela estremamente utile per l'individuazione delle componenti spettrali di un segnale immerso nel rumore. La speranza è che la KLT tronca permetta di individuare il periodo di rotazione per i dati rumorosi di Vela con una precisione maggiore della semplice analisi con la trasformata di Fourier.

8.6 Risultati KLT

In primo luogo, la versione tronca della KLT per signal processing è stata testata su segnali a cui è stato aggiunto rumore. In seguito sono stati processati vettori costituiti da piccole porzioni dei segnali originali delle pulsar, fino a un massimo di 5 secondi di osservazione, a causa della mancanza di risorse computazionali.

Come segnale di partenza è stata considerata una sinusoida con varianza 2, campionata a 1 kHz per 5 secondi, la cui frequenza fondamentale è a 300 Hz.

Come primo test, alla sinusoida è stato aggiunto AWGN (rumore bianco gaussiano) con varianza pari a 40, portando avere un SNR a 0.05. Lo spettro del segnale risultante è in Figura 43, in cui la componente a 300 Hz si individua immediatamente nonostante il rumore.

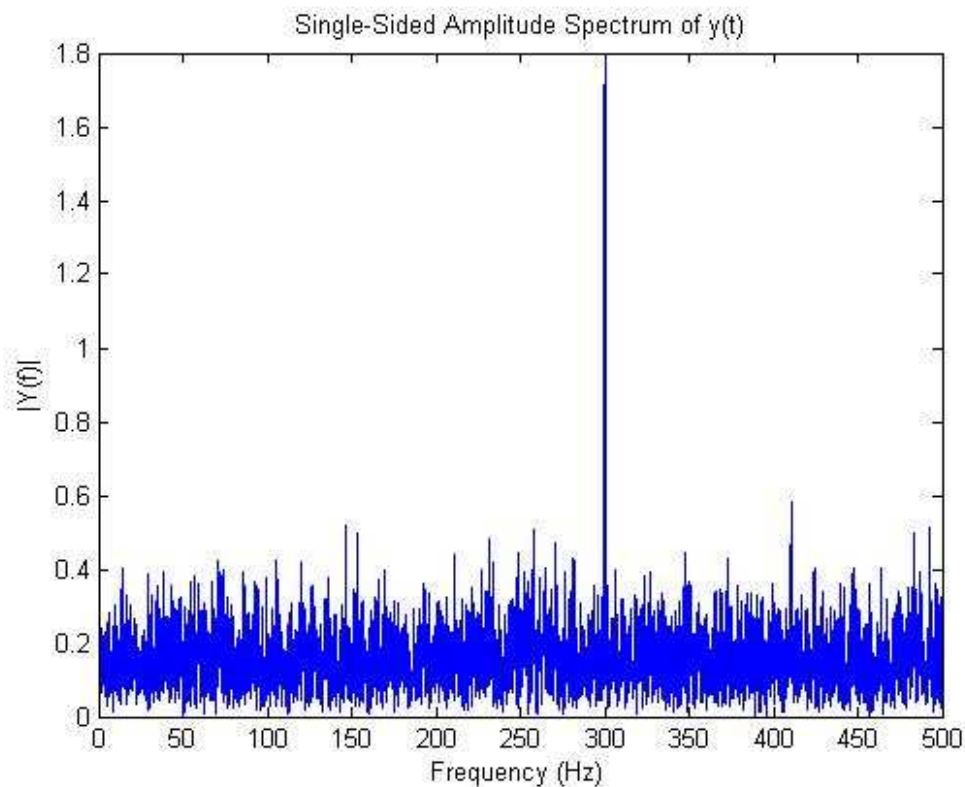


Figura 43 Analisi in frequenza di segnale + rumore, SNR = 0.05

È sufficiente analizzare lo spettro della KLT calcolata solo sull'autovalore dominante per ottenere il grafico in Figura 44. La componente a 300 Hz è stata isolata dalla KLT e non c'è traccia del rumore.

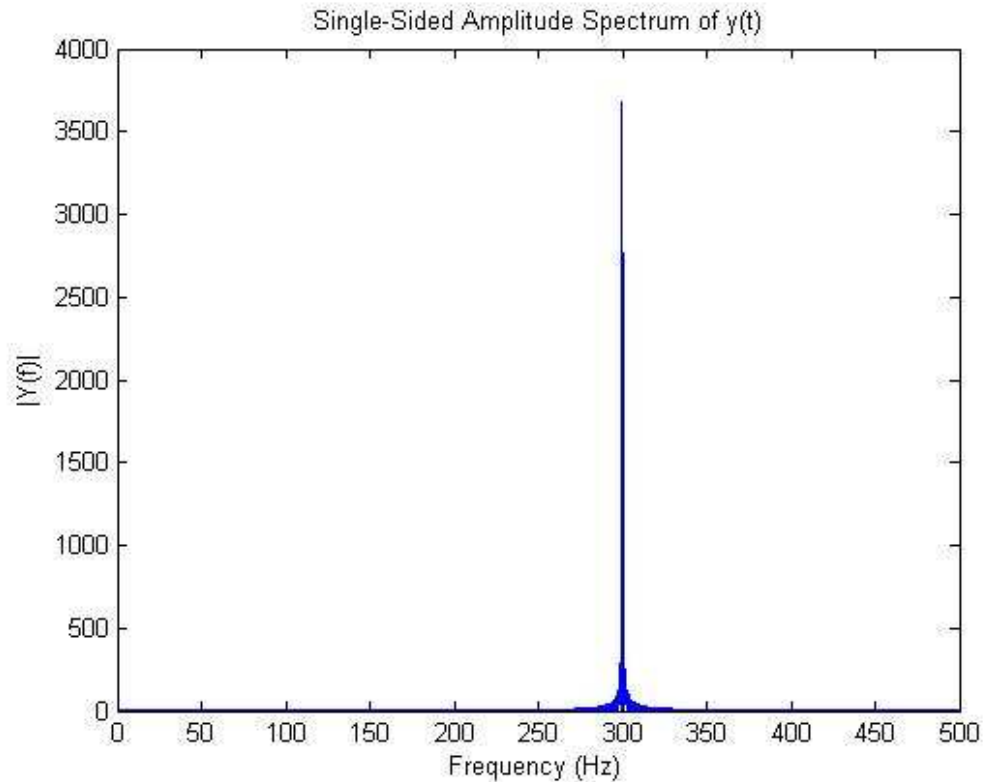


Figura 44 Spettro per la KLT calcolata sulla prima autofunzione pesata per l'autovalore dominante.

La varianza del rumore è stata aumentata successivamente a 400; con un $SNR = 0.005$; l'analisi spettrale risultante è riportata in Figura 45. In questo caso il rumore rende difficoltosa l'individuazione della frequenza a 300 Hz: si può notare un picco in corrispondenza della frequenza esatta, ma ce ne sono altri di intensità simile (anche maggiore) che impediscono la separazione delle componenti dovute al rumore da quella del segnale.

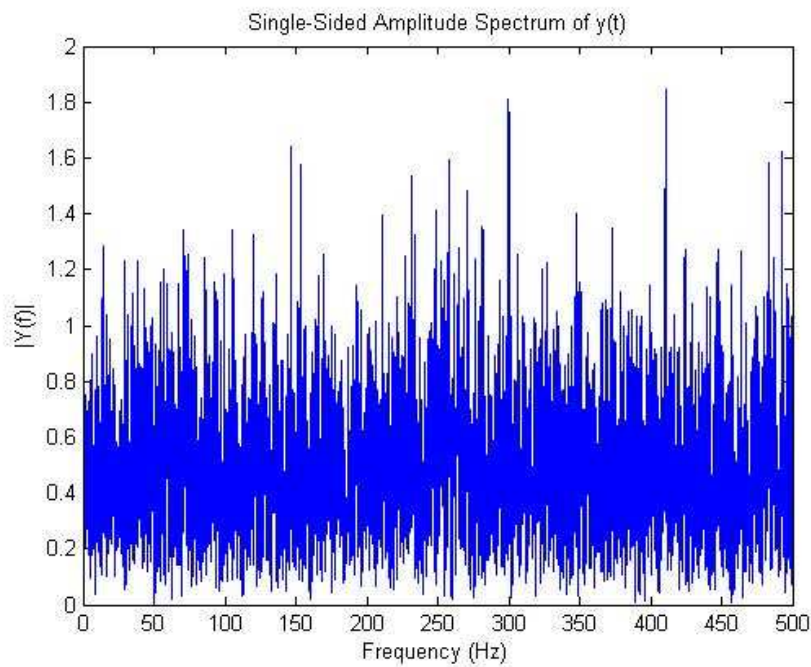


Figura 45 Analisi in frequenza di segnale + rumore, SNR = 0.005

Con un SNR così basso si può vedere il vantaggio della KLT sulla FFT: è sufficiente anche in questo caso analizzare solamente la prima componente, legata all'autovalore dominante per ottenere il grafico in Figura 46.

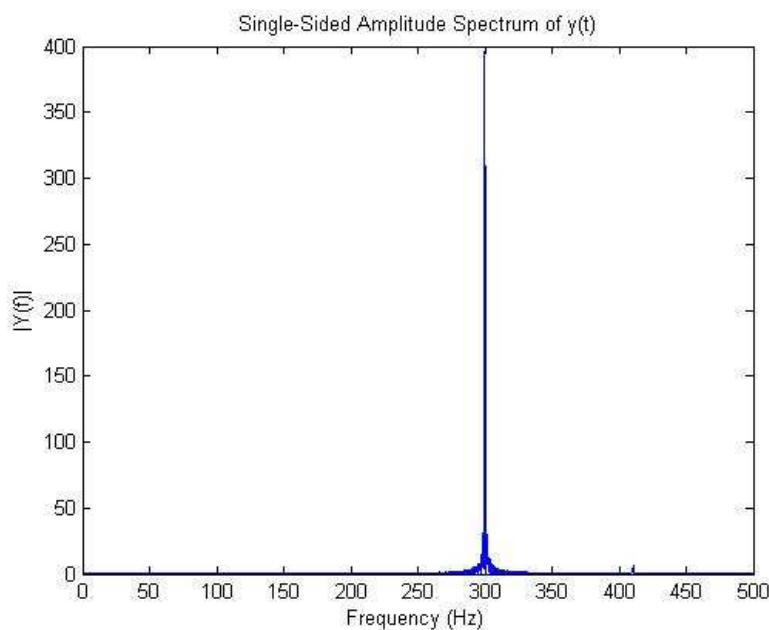


Figura 46 Spettro per la KLT calcolata sulla prima autofunzione.

Al peggiorare del SNR, l'analisi dello spettro con la sola trasformata di Fourier non restituisce nessuna informazione utile; grazie alla KLT è possibile individuare le componenti spettrali del segnale, anche se, dato l'elevato rumore, si rende necessario considerare più autofunzioni per mantenere l'informazione desiderata legata al segnale. In Figura 47 vengono posti a confronto lo spettro del segnale sommato a un rumore di varianza pari a 600, e lo spettro della KLT calcolata su quel segnale. In questo caso sono state considerate le prime sette coppie autovalore-autovettore per la costruzione della KLT.

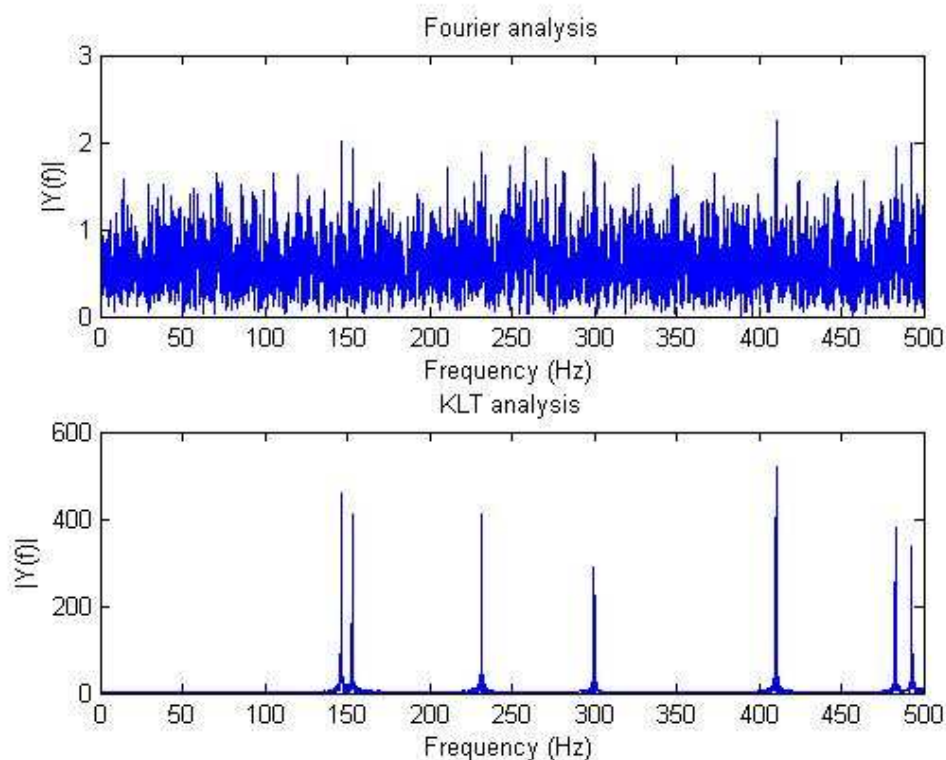


Figura 47 Confronto tra FFT e KLT per SNR = 0.0033

Infine è stata testata la KLT su parti di segnale già binnato delle osservazioni delle tre pulsar. Mantenendo il numero di campioni $N = 5000$, come per il segnale precedente, sono stati processati 5 secondi di osservazione, o 20 secondi nel caso della pulsar Vela, dove il tempo di binning è 4 ms.

Nel caso di PSR B0540-69 e Vela è stato necessario eseguire l'analisi dello spettro rispettivamente su 60 minuti e 180 minuti di segnale per individuare una componente in frequenza corrispondente al periodo di rotazione. Di conseguenza è impossibile che processando 5 o 20 secondi di segnale sia possibile ricavare qualcosa, indipendentemente dalla potenza della KLT. Dall'analisi dei dati di Crab pulsar, invece, è stato possibile fare alcune considerazioni sull'efficacia della KLT.

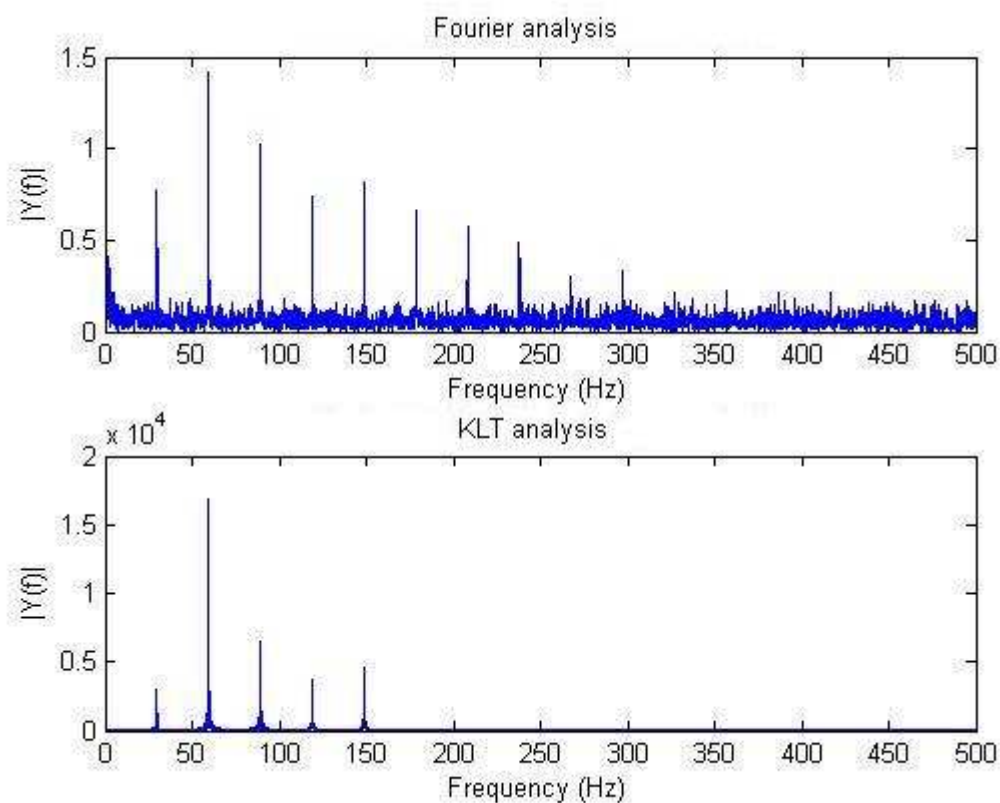


Figura 48 Confronto tra spettro del segnale e KLT per 5 secondi di osservazione di Crab pulsar.

In Figura 48 è riportato il confronto tra FFT e KLT di 5 secondi di osservazione della pulsar Crab. Per vedere il periodo di rotazione, posto a circa 30 Hz, sono state mantenute le prime 5 coppie autovalore-autovettore. Anche se le componenti visibili nella KLT e nella FFT sembrano poste alla stessa frequenza, il valore in frequenza è diverso. Il picco nel grafico in alto di Figura 48 individua un periodo di 0.033436734 s, mentre con la KLT il periodo trovato sulla stessa porzione di segnale è 0.033573770 s; il secondo

valore è significativamente migliore se comparato col periodo di rotazione utilizzato per la light curve di Crab che è pari a 0.033621669 s. Questa importante differenza porta a pensare che se fosse possibile calcolare la KLT sull'intero segnale di Vela o PSR B0540-69, rappresentante ore di osservazione, si potrebbe individuare un periodo di rotazione più preciso di quello individuato con la semplice analisi di Fourier.

CAPITOLO 9

Conclusioni

9.1 Conclusioni

L'obiettivo prefissato all'inizio di questo lavoro di tesi era di realizzare un nuovo metodo, grazie allo studio della Karhunen-Loève Transform, per l'analisi dei dati ottenuti dall'osservazione delle tre pulsar per mezzo di IquEye. Il lavoro di ricerca ha portato a considerare validi due tipi diversi di approccio al problema: la Principal Component Analysis e la KLT per signal processing.

L'implementazione della KLT per segnali, considerata inizialmente la più promettente, è stato ostacolata dall'enorme carico computazionale richiesto, che ne ha limitato fortemente l'impiego in altri campi di ricerca, e dall'assenza di altre applicazioni riscontrabili in letteratura che permettessero di risolvere le difficoltà incontrate nella stesura dell'algoritmo. Grazie all'utilizzo di algoritmi approssimati (fattorizzazione di Lanczos), è stato comunque possibile eseguire dei test con la versione tronca della KLT che hanno dimostrato come essa sia uno strumento potente per determinare le componenti spettrali di segnali rumorosi con SNR estremamente basso. Alla fine del lavoro è stata provata la bontà dello strumento ma non si è comunque ottenuto un metodo in grado di effettuare il denoising dei nostri segnali.

L'analisi delle componenti principali (PCA) invece è stata determinante per ottenere la light curve di PSR B0540-69 ed in particolare di Vela, della quale non esistevano, fino ad oggi, curve di luce successive al 1979. Grazie a questo strumento l'obiettivo del lavoro di tesi, e cioè riuscire a vedere la pulsar Vela, è stato raggiunto.

9.2 Sviluppi futuri

L'algoritmo *ufolding*, che implementa la PCA per trovare il periodo di rotazione migliore e quindi la curva di luce corrispondente, è concepito unicamente per operare sui dati di una pulsar. Esso costituisce una delle tante applicazioni della PCA che, come è già stato detto, trova largo impiego nella compressione di dati, algoritmi per il riconoscimento facciale e analisi di dati statistici. In futuro *ufolding* potrebbe essere usato per l'analisi dei dati ottenuti da altre pulsar, meno luminose e con un periodo di rotazione è dell'ordine di 1 ms, che si prevede di osservare con il telescopio NTT di La Silla.

La KLT per signal processing invece ha potenzialmente un numero di applicazioni molto più alto, ma è utilizzabile solo su segnali con un ridotto numero di campioni o in caso di una grande disponibilità di risorse computazionali. Attualmente una KLT analoga viene utilizzata in ambito radioastronomico presso SETI Italia con lo scopo di verificare la presenza o meno di segnali nel rumore cosmico. In futuro, con maggiore potenza di calcolo a disposizione, questo tipo di KLT potrebbe trovare largo impiego nella rivelazione di segnali o, tramite la versione completa, nel campo del denoising di segnali.

Bibliografia

- [1] Alice Bean, John P. Ralston, James Snow, *Evidence for observation of virtual radio Cherenkov fields*, Nuclear Instruments and Methods in Physics Research A, Vol. 595, 2008.
- [2] C. Maccone, *Telecommunication KLT and Relativity*, IPI Press, Colorado, 1994.
- [3] Naletto G., Barbieri C. Dravins, D., Occhipinti T., Tamburini F., Da Deppo V., Fornasier S., D’Onofrio M., Fosburye R. A. E., Nilsson R., Uthas H., Zampieri L., “*QuantEye: A Quantum Optics Instrument for Extremely Large Telescopes*”.
- [4] P.T. Wallace, Bruce A. Peterson, K. H. Elliott, *The optical pulse profile of the Vela pulsar*.
- [5] Salvatore Paolo Ciacco, *Integrazione e caratterizzazione dello strumento IquEye per una sua applicazione su NTT*.
- [6] Pierpaolo Pari, *Rilevazione di segnali tramite trasformata KLT nel progetto SETI*.
- [7] Aapo Hyvärinen, Juha Karhunen, Erkki Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.
- [8] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
- [9] Schillirò F., Pluchino S., Montebugnoli S., *La KL Transform: considerazioni generali sulle metodologie di analisi ed impiego nel campo della Radioastronomia*.
- [10] S. Gradari, M. Barbieri, P. Zoccarato, C. Barbieri, G. Naletto, I. Capraro, T. Occhipinti, E. Verroi, A. Possenti, L. Zampieri, A. Di Paola, C. Facchinetti, *The optical light curve of the LMC pulsar B0540-69 in Jan 2009*.
- [11] Germanà C., Zampieri L., Barbieri C, Naletto G., Cadèz A., Calvani M., Capraro I., Di Paola A., Facchinetti C., Occhipinti T., Verroi E., Zoccarato P., *Crab pulsar optical observations with AquEye*.

- [12] Anil K. Jain, *A fast Karhunen-Loève Transform for a class of random processes*.
- [13] P. T. Boyd, G. W. Van Citters, J. F. Dolan, K. G. Wolinski, J. W. Percival, R. C. Bless, J. L. Elliot, M. J. Nelson, M. J. Taylor, *High-speed photometer observations of the LMC pulsar B0540-69*.
- [14] Jonathon Shlens, *A tutorial on Principal Component Analysis*, 2009.
- [15] Jodrell Bank, www.jb.man.ac.uk
- [16] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, H.M.S.O.
- [17] J. Middleditch, C. R. Pennypacker, M. S. Burns, *Optical color, polarimetric, and timing measurements of the 50 ms Large Magellanic Cloud pulsar, PSR 0540-69*.
- [18] R. Dixon, *On the detection of unknown signals*, ASP Conference Series, Vol. 47 1993.