

Tesi

Giacomo Caiazzo

February 2023

1 Introduzione

L'industria 4.0 si fonda su due elementi cardine: la digitalizzazione e l'interconnessione dei diversi dispositivi presenti all'interno dell'ambiente industriale. L'obiettivo che si prefigge quindi l'industria 4.0 è quello di costruire ambienti industriali all'interno dei quali poter controllare ed ottimizzare ogni aspetto del processo produttivo e dove avere dispositivi capaci di comunicare e scambiare informazioni tra loro senza l'intervento di operatori umani.

1.1 La manutenzione predittiva

Nel contesto appena descritto, un ruolo di primaria importanza è ricoperto dalla manutenzione predittiva. Questa differisce essenzialmente dalla manutenzione preventiva, che si basa sull'esecuzione di attività di controllo e di manutenzione periodiche programmate al fine di evitare guasti futuri. La manutenzione predittiva, al contrario, si fonda sull'idea di individuare sul nascere eventuali problemi legati al funzionamento di diverse tipologie di macchinari, che se dovessero perdurare nel tempo potrebbero portare al danneggiamento degli stessi. Si parla quindi di una strategia che va ad abbattere notevolmente i costi legati ad operazioni di controllo periodiche, e che soprattutto può essere eseguita da sensori che siano in grado di riconoscere quando un macchinario o un componente si trovano a lavorare in condizioni di funzionamento anomalo.

1.2 Il monitoraggio dei motori elettrici

Il focus principale di questa tesi è la manutenzione predittiva dei motori elettrici, e in particolare modo dei motori ad induzione, più comunemente conosciuti come motori asincroni. I motori ad

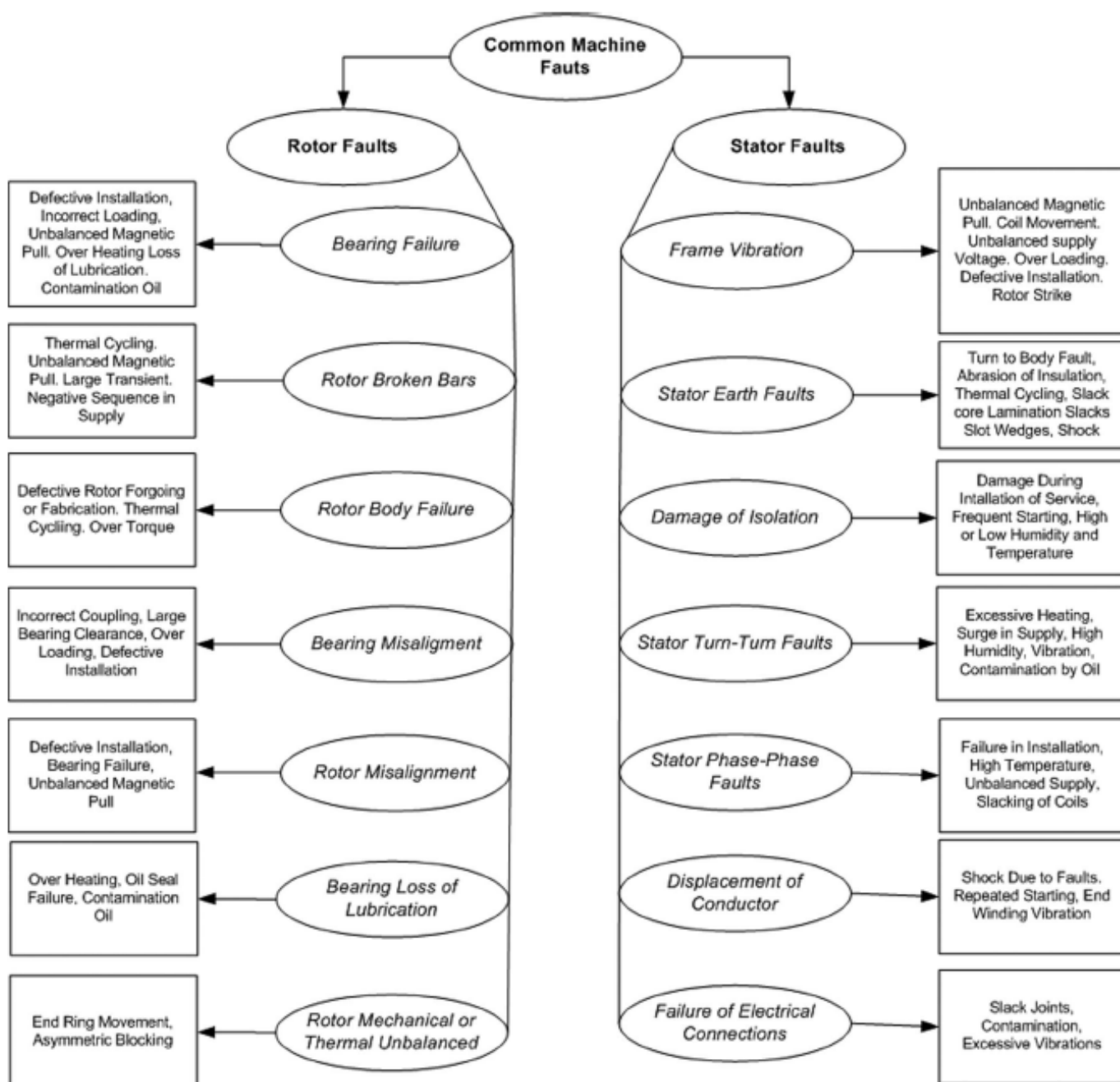


Figure 1.1: Tipologie di guasto e le loro cause

induzione rappresentano infatti uno dei punti focali dell'industria moderna, dal momento che circa il 90% dei processi industriali dipende da essi. Risulta quindi di fondamentale importanza

monitorare le condizioni di salute dei motori. In funzione della natura della condizione di guasto o di malfunzionamento, è possibile operare una prima distinzione tra problemi di natura meccanica e di natura elettrica, mentre in base a dove è localizzato il guasto tra guasti al rotore o allo statore. L'insorgere di una di queste condizioni potrebbe portare il motore a lavorare in regime di funzionamento anomalo, causando gravi perdite in termini di rendimento e danneggiandone le componenti. Inoltre, nel caso peggiore, il motore potrebbe incorrere in un malfunzionamento catastrofico. Bisogna quindi individuare le principali categorie di guasti che possono verificarsi all'interno di un motore chiamato ad operare per lunghi periodi di tempo. Tra queste, quelle di maggiore interesse sono:

- Danneggiamento dei cuscinetti: possono essere causati da vari fattori, come mancanza di lubrificazione, stress meccanici sulle sfere dei cuscinetti che possono portare al danneggiamento degli elementi rotanti.
- Danneggiamento della pista interna ed esterna
- Guasto dello statore o del rotore

Questi guasti porterebbero ad una serie di effetti negativi, quali ad esempio una maggiore quantità di calore generato, un maggior consumo di energia o un minor valore di coppia d'uscita. Risulta quindi essenziale essere in grado di monitorare le condizioni di salute di un motore, così da essere in grado di intervenire in maniera tempestiva qualora esso si trovasse a lavorare in condizioni non ordinarie e contenere la natura del danno. Definito quindi il target principale del lavoro di tesi, bisogna capire di quali strumenti abbiamo bisogno per effettuare la manutenzione predittiva dei motori elettrici.

1.3 Analisi delle vibrazioni

Sono diverse le tecniche che possono essere utilizzate per il monitoraggio delle condizioni di salute e di funzionamento dei motori elettrici. Una delle più popolari è la MCSA (motor current signature analysis). Questa tecnica si basa sul monitoraggio della corrente di statore del motore, e si propone di associare allo specifico guasto preso in esame un certo valore di segnale di corrente misurato. Il segnale viene processato attraverso la trasformata di Fourier veloce (FFT): analizzando lo spettro del segnale è possibile individuare quale condizione di guasto o di malfunzionamento insorge nel motore, in quanto ad ognuna di queste corrisponde una "firma" in

termini di corrente che presenta delle armoniche all'interno dello spettro. Queste firme dipendono sia dalle caratteristiche del motore che dalle sue condizioni di funzionamento. Si tratta di una tecnica che ha dimostrato di poter ottenere buoni risultati, ma che presenta un grande svantaggio: per ogni specifica condizione di funzionamento anomalo è richiesta strumentazione dedicata ed un algoritmo per il riconoscimento automatico della stessa.

Un'altra tecnica adoperata nell'ambito della manutenzione predittiva è la vibration signature analysis (VSA); quasi il 100 % dei guasti nei motori si riflette infatti in variazioni nelle vibrazioni. La VSA, in maniera simile alla MCSA, ha lo scopo di riconoscere le condizioni di funzionamento anomalo del motore, e di associare a queste una firma in termini di vibrazioni a cui esso è soggetto. Ogni motore è caratterizzato da una vibrazione propria, e l'analisi spettrale di tali vibrazioni consente di monitorare le condizioni di salute del motore. L'analisi nel dominio della frequenza risulta estremamente efficace, in quanto rispetto a quella nel dominio del tempo fornisce informazioni più dettagliate riguardo lo stato di funzionamento del motore. Come mostrato nella tabella 1, sono state stabilite delle correlazioni tra alcune specifiche componenti armoniche e possibili condizioni di guasto. Anche in questo caso, quindi, viene adoperata la FFT come strumento matematico di elaborazione dei dati raccolti, ovvero misure di accelerazione lungo i 3 assi.

POSSIBLE FAILURE RELATED TO THE VIBRATION FREQUENCY
(RPM IS THE MOTOR SPINDLE)

Frequency	Possible Failure
1 x RPM	Unbalance, looseness, misalignment of gear or pulley, resonance, electrical problems, alternative forces
2 x RPM	Mechanical offset, misalignment, alternative forces, resonance
3 x RPM	Misalignment, axial mechanical breath
< 1 x RPM	Slip, oil whirl
Supply frequency	Electrical problems
RPM harmonics	Broken bars, Damaged gears, aerodynamic forces, hydraulic forces, mechanical set, alternative forces

Table 1: Relazione tra RPM e frequenza

La tecnica che è stata scelta per il lavoro di tesi prevede l'analisi delle vibrazioni del mo-

tore elettrico per ricavare informazioni utili riguardo lo stato di funzionamento dello stesso. Pertanto, è stato necessario l'utilizzo di un sensore che fosse in grado di misurare valori di accelerazione, misure da elaborare successivamente attraverso la FFT per comprendere come le armoniche presenti all'interno dello spettro siano correlate alle condizioni di funzionamento del motore.

1.4 Realizzazione di un modello di apprendimento automatico

Nel corso degli ultimi decenni, approcci che prevedono l'utilizzo di modelli di machine learning sono stati ampiamente utilizzati nell'ambito della manutenzione predittiva per la diagnosi delle condizioni di funzionamento dei motori ad induzione. Metodologie di classificazione come alberi decisionali (decision trees) o reti neurali (neural networks) hanno dimostrato di essere molto efficienti quando combinati a tecniche di elaborazione dei segnali. Il paradigma di apprendimento utilizzato nel lavoro di tesi rientra nella categoria dell'apprendimento supervisionato: viene raccolta una certa quantità di campioni, che vengono opportunamente "etichettati" in base allo stato di funzionamento dell'oggetto vibrante, il motore elettrico in questo caso. Le misure di accelerazione vengono acquisite dal sensore, processate attraverso l'utilizzo della FFT e successivamente etichettate. Un passaggio fondamentale all'interno di questo processo è la scelta delle "features", che consente di ridurre il numero di variabili di input del modello predittivo che si intende costruire. Avere un numero di variabili di ingresso limitato consente infatti di ridurre il costo computazionale del modello e di migliorarne le performance.

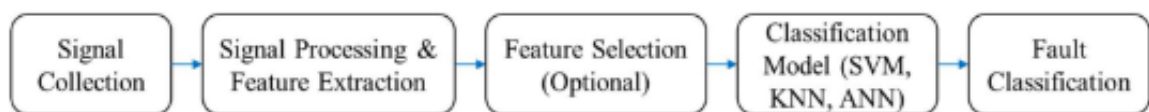


Figure 1.2: Processo di apprendimento automatico

Una scelta possibile, ad esempio, è quella di prelevare per uno dei tre assi un certo numero di frequenze all'interno dello spettro del segnale e i valori delle ampiezze corrispondenti. In questo caso le variabili di input sarebbero 2, un'ampiezza e una frequenza. Una volta determinato il criterio dietro la scelta delle frequenze e delle ampiezze da prelevare bisogna etichettare i dati ricavati dalle misure. Le etichette che vengono assegnate alle coppie rappresentano le variabili di uscita. Il set di dati etichettati così costruito diviene il "training set" del modello di apprendimento, ovvero l'insieme di dati sul quale questo viene addestrato. Scopo dell'algoritmo

di addestramento è quello di far comprendere al modello la relazione che sussiste tra le variabili di ingresso e quelle di uscita, cosicché esso sia in grado di eseguire previsioni anche qualora non conoscesse a priori il valore dell'uscita.

2 Kit di sviluppo STM

Per eseguire le misure di accelerazione è stato utilizzato un kit prodotto da STMicroelectronics, lo "STEVAL-BFA001V2B". Si tratta di un kit di progettazione pensato per il monitoraggio delle condizioni e la manutenzione predittiva, il cui layout soddisfa i requisiti IEC61000-4-2/4 e EN60947 per applicazioni industriali. Il kit consta di 3 componenti principali:

- Sensor board (IDP005V2)
- Communicationn adapter board (STEVAL-UKI001V2)
- Strumento di programmazione e debug (STLINK-V3MINI):

La sensor board è una scheda compatta che rappresenta l'hardware di misura e di controllo del kit. Su di essa troviamo diverse tipologie di sensori, come sensori di umidità, di temperatura o di pressione. Il dispositivo STLINK-V3MINI è invece una sonda di programmazione e debug stand-alone per i microcontrollori STM32. È composto da un modulo principale e da un adapter board che si interfaccia e si collega fisicamente alla sensor board. Il firmware, che lavora sul microcontrollore a 32 bit ad elevate prestazioni STM32F469AI, ARM Cortex-M4, contiene algoritmi dedicati per l'elaborazione avanzata di segnali nel dominio del tempo e della frequenza. I risultati dell'analisi dei dati dei sensori possono essere inoltre visualizzati su un emulatore di terminale PC tramite connettività cablata.

2.0.1 Accelerometro IIS3DWB

Come già accennato in precedenza, sulla sensor board è presente un vasto assortimento di sensori. Tra questi, quello utile per eseguire le misure di accelerazione è il sensore IIS3DWB, in quanto dotato di un accelerometro digitale a 3 assi capace di operare su un'ampia gamma di frequenze. La possibilità di operare su un intervallo di frequenze ultra-ampio, unito alla sensibilità molto stabile e la capacità di lavorare in un range di temperatura molto ampio (fino a 105°C), rendono il IIS3DWB ideale per il monitoraggio delle vibrazioni nelle applicazioni industriali.

Questo dispositivo è installato direttamente sull'oggetto che vibra: in questo modo l'energia delle vibrazioni viene trasformata in un segnale elettrico proporzionale all'accelerazione istantanea dell'oggetto vibrante. L'IIS3DWB è un accelerometro capacitivo. Questo usa la variazione della capacità elettrica di un condensatore al variare della distanza tra le due armature per rilevare lo spostamento della massa.

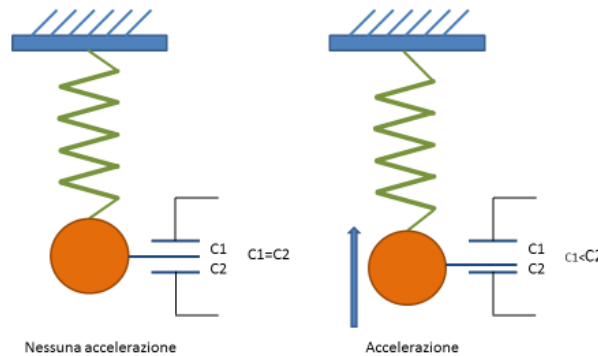


Figure 2.1: Modello di accelerometro capacitivo

Il principio di funzionamento di questo dispositivo si basa sul posizionare un peso installato su una delle due molle. Uno dei due elementi elastici è fissato alle armature del condensatore, l'altra, invece, al peso. La forza che agisce sul sensore causa lo spostamento del peso sulle molle, in questo modo varia la distanza tra l'elemento capacitivo e la massa, e ha luogo una variazione della capacità. In particolare, l'IIS3DWB in particolare è realizzato con la tecnologia MEMS(micro electro-mechanical systems), la più utilizzata in questo ambito. Il sensore ha un intervallo di accelerazione a scala completa selezionabile di $\pm 2/\pm 4/\pm 8/\pm 16$ g ed è in grado di misurare accelerazioni con una larghezza di banda fino a 5 kHz con una velocità dati di uscita di 26,7 kHz.

2.1 Software STM32CUBEMX

L'utilizzo del software STM32CUBEMX, prodotto e distribuito da STM, è stato di notevole supporto, in quanto ha consentito di impostare e di attivare ad un livello alto, e dunque evitando di farlo a livello dell'IDE di programmazione, tutti i canali di comunicazione, le funzioni da utilizzare e le periferiche necessarie per eseguire le misure di accelerazione e trasmettere i dati relativi alle stesse. Creato quindi un progetto su CubeMX e selezionato il microcontrollore utilizzato, basta consultare il datasheet del kit per vedere quali sono i pin utili per configurare e programmare l'accelerometro.

3D accelerometer

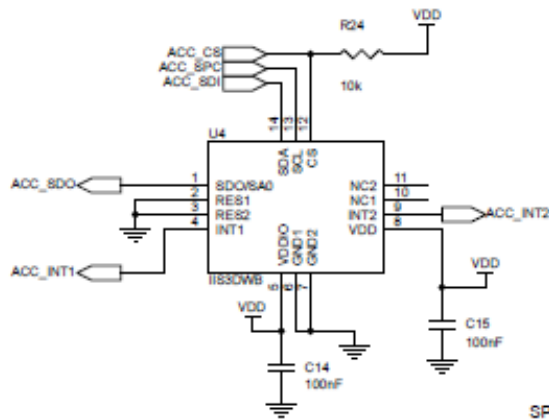


Figure 2.2: Circuito schematico IIS3DWB

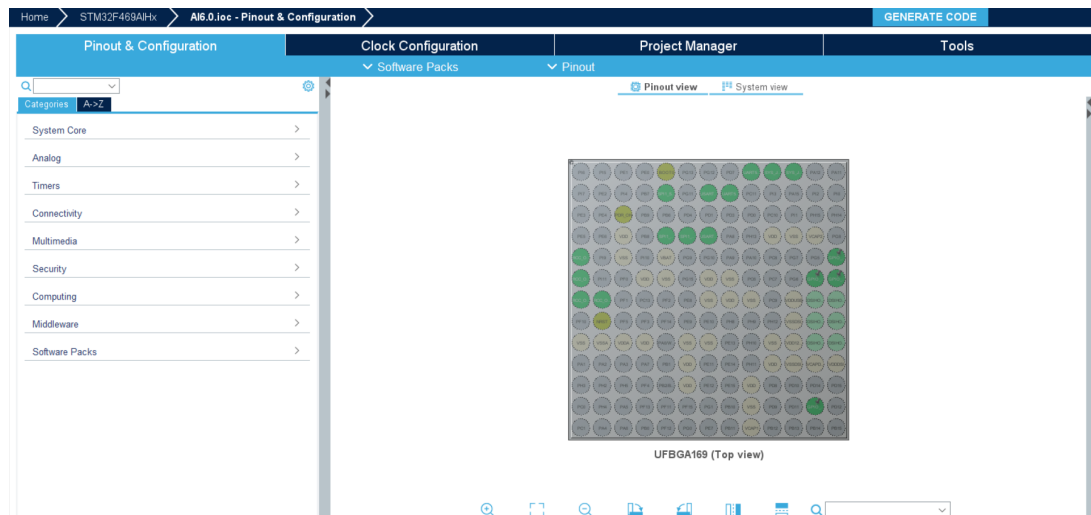


Figure 2.3: Interfaccia CubeMX

I pinout di interesse sono:

- PB3,PB4,PB5: sono i pin SPI, configurati all'attivazione del canale SP1
- PG2: corrisponde al GPIO di output che individua il chip select del sensore
- PD2,PC12: UART RX e TX per la comunicazione seriale dei dati

2.1.1 Protocollo SPI

La comunicazione con il sensore IIS3DWB avviene attraverso un protocollo di tipo SPI: si tratta di una comunicazione singolo Master e singolo Slave di tipo sincrono e full-duplex. Ad avviarla è il master, che abilita lo slave tramite il segnale CS ed impone il segnale clock (SPC) sulla linea

di comunicazione. Sia il dispositivo master che quello slave sono dotati di registri a scorrimento interni la cui dimensione è uguale, in cui i bit vengono emessi e contemporaneamente immessi. I registri a scorrimento sono interfacce che consentono di impartire comandi e di trasmettere dati che arrivano serialmente e che vengono prelevati in parallelo. Ad ogni impulso di clock, i dispositivi che stanno comunicando sulle linee del bus SPI emettono un bit dal loro registro e lo rimpiazzano con un bit proveniente dal loro interlocutore. Questo avviene tramite le linee MISO/SDI e MOSI/SDO. Il segnale MISO è la linea attraverso cui il dispositivo (master o slave) riceve il dato seriale emesso dalla controparte. Sullo stesso fronte di commutazione del clock, il dispositivo emette, con la stessa cadenza, il suo output ponendo il dato sulla linea MOSI (linea di output di dato).

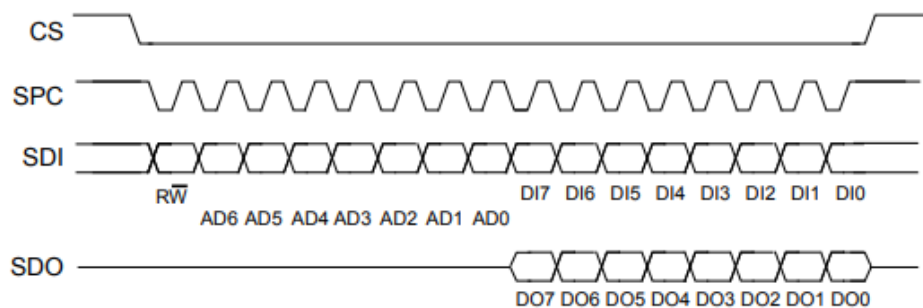


Figure 2.4: Protocollo di lettura e scrittura

Nel caso preso in esame, il ruolo di master è interpretato dal microcontrollore, mentre quello di slave dal sensore di accelerazione. La sincronizzazione avviene sul fronte di salita o di discesa del clock ed è regolata da due parametri configurabili:

- CPOL: determina la polarità del clock. Quando è impostato a 0, il clock si porta a livello logico basso quando si trova nel suo stato di riposo, mentre si porta a livello logico alto se impostato ad 1.
- CPHA: determina il fronte di clock in cui il ricevente campiona il segnale di ingresso

Le modalità di funzionamento più spesso utilizzate dai dispositivi in commercio sono quelle con CPHA=CPOL=0 e con CPHA=CPOL=1. Il segnale CS viene utilizzato per l'abilitazione della porta seriale ed è controllato dal master SPI, ovvero il microcontrollore. CS diventa basso all'inizio della trasmissione e torna alto alla fine. Attraverso l'uso del software CubeMX, è stata

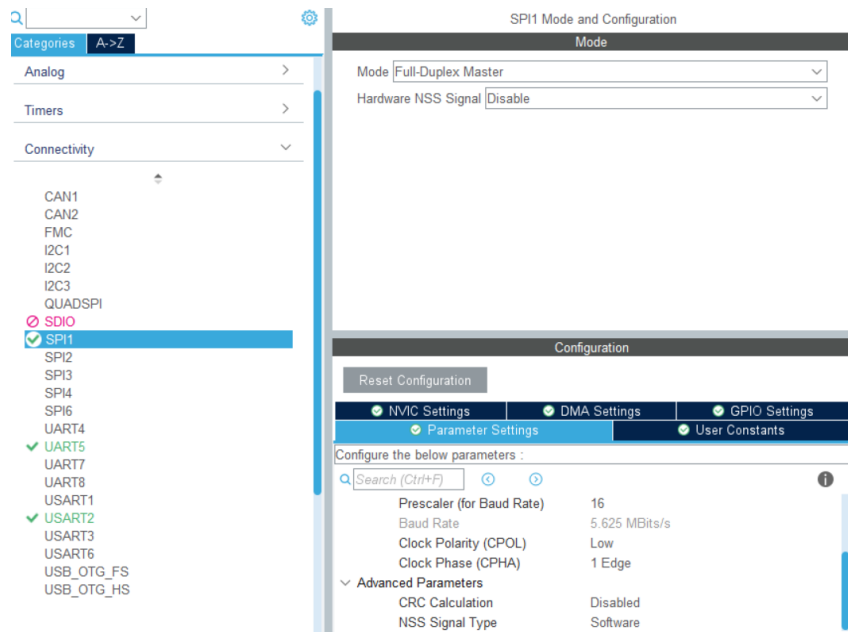


Figure 2.5: Configurazione SP1

selezionato e configurato il protocollo SPI abilitando la comunicazione SPI1 in "Connectivity" ed impostandolo in modalità Master Full-Duplex.

Bisogna successivamente configurare i pin di interrupt INT1 ed INT2 corrispondenti ai pin EXTI1 ed EXTI2, ed il pin PG2, al quale corrisponde il pin di output dove il sensore scrive i risultati delle misure di accelerazione.

2.1.2 Configurazione del segnale di clock

Dal diagramma a blocchi in figura, si può verificare che la CPU, attraverso il bus AHB MATRIX, può accedere alle linee AHBx/APBx per leggere e scrivere dati ad una velocità di 180MHz. Per ricevere la lettura delle misure di accelerazione, la CPU abilita la periferica relativa all'SPI1 tramite il bus AHB/APB2 (che lavora a 90MHz) inviando un segnale di enable ed interrogandolo per salvare in memoria i dati. Attraverso CubeMX è possibile accedere al pannello che consente di configurare il segnale di clock abilitando la sorgente di quarzo esterna nella sezione RCC. Dal system Clock configuration è possibile poi impostare l'HSE(High-speed external) alla velocità di 24MHz.

2.1.3 Utilizzo del timer

L'utilizzo di una periferica timer risulta necessario. Supponendo infatti di voler accedere alle letture del sensore con frequenza pari a 20KHz, senza l'utilizzo di un timer quello che succed-

erebbe è che si potrebbero verificare casi in cui si andrebbe ad interrogare il sensore attraverso l'SPI ad una velocità maggiore rispetto a quella con cui questo acquisisce le misure di accelerazione, con il risultato di ottenere o la stessa misura dell'istante precedente, oppure un valore NULL. Per sincronizzare le misure alla velocità di 20KHz viene quindi abilitato ed interrogato un timer attraverso il bus AHB/APB1, operante a 45MHz. Per generare dunque un segnale di sincronia alla frequenza scelta, bisogna impostare un numero di conteggi pari a 2250.

2.1.4 Interfaccia UART

L'interfaccia UART(Universal Asynchronous Receiver-Transmitter) è un dispositivo hardware che converte flussi di bit di dati da un formato parallelo ad uno seriale asincrono e viceversa. I due segnali di ciascun dispositivo UART sono denominati:

- Trasmettitore(Tx)
- Ricevitore(Rx)

L'UART trasmittente è collegato a un bus dati di controllo che invia i dati in forma parallela. Da qui, i dati vengono trasmessi sulla linea di trasmissione in serie, bit per bit, all'UART ricevente. Questo, a sua volta, convertirà i dati seriali in parallelo per il dispositivo ricevente. Un dispositivo UART presenta quindi un pin di trasmissione e ricezione dedicato per l'una o l'altra. La velocità di trasmissione delle informazioni (baud rate) deve essere uguale per dispositivo ricevente e trasmittente. L'interfaccia UART non utilizza un segnale di clock per sincronizzare i dispositivi che comunicano, in quanto trasmette i dati in modo asincrono. Il punto di sincronizzazione è gestito avendo lo stesso baud rate su entrambi. Nel lavoro di tesi, la porta UART presente sulla sensor board che è stata abilitata è la UART5, e il baud rate impostato è pari a 230400.

2.2 Ambiente di programmazione STMCubeIDE

Una volta terminata questa fase preliminare sul CubeMX, bisogna spostarsi all'interno di un ambiente di programmazione, in questo caso STMCubeIDE. Da CubeIDE è possibile creare un nuovo progetto partendo dalla configurazione appena determinata su CubeMX. In questo modo verranno generate automaticamente le librerie necessarie per inizializzare ed utilizzare le periferiche configurate.

Per effettuare le operazioni di misura con il sensore è necessario inviare un'istruzione tramite SPI (Transmit) contenente l'indirizzo che indica l'operazione da svolgere. Per fare un esempio immediato, quando viene definita la variabile `spiSndX[0] = 0x29—0x80`, quello che viene comunicato tramite la funzione `transmit` è di eseguire l'operazione descritta dal registro 29, ovvero quella che si occupa della lettura degli 8 bit più significativi relativi all'accelerazione lungo l'asse X. Essendo un'operazione di lettura va settato il bit più significativo ad 1; da qui l'utilizzo dell'OR con il valore 0x80. In modo del tutto analogo, per ricevere il risultato di misura bisogna utilizzare un'istruzione tramite SPI di Receive, così da memorizzare all'interno di una struttura dati definita dall'utente i valori delle misure di accelerazione ricavate dall'accelerometro. Questa operazione va eseguita poi per ognuno dei tre assi. Bisogna inoltre evidenziare l'importanza del timer, che attraverso un flag (`flag elapsed`) assicura che le misure vengano ogni inverso della frequenza di campionamento impostata dall'utente. L'ultimo passo consiste nel visualizzare a schermo i risultati delle operazioni di misura attraverso l'interfaccia UART, passando dalla sensor board alla debugging board attraverso l'utilizzo di un cavo SWD e dell'adapted board così da poterli leggere sulla com seriale di un PC.

3 Elaborazione dei dati attraverso la FFT

Lo strumento matematico scelto per elaborare le misure di accelerazione ricavate attraverso il kit utilizzato è la trasformata di Fourier veloce. Il motivo dietro la scelta della FFT come strumento matematico di elaborazione dei dati risiede nel fatto che l'analisi nel dominio della frequenza dei campioni raccolti è in grado di fornire informazioni molto dettagliate riguardo lo stato in cui il motore si trova ad operare. Molte anomalie nel funzionamento dei motori elettrici, infatti, sono il riflesso di ampiezze altrettanto anomale in corrispondenza di specifiche frequenze o intervalli di frequenze, che possono essere analizzati per rilevare eventuali condizioni di malfunzionamento o di guasto.

3.1 La DFT

Prima di approfondire il discorso relativo alla FFT, tuttavia, bisogna introdurre alcune nozioni preliminari. Dal momento che stiamo lavorando con un insieme discreto di campioni, ovvero le N misure di accelerazione ricavate dal sensore, avremo un range di frequenze testabili che varia tra 0Hz (la componente continua del segnale in ingresso) e la frequenza di Nyquist, ovvero $F_n =$

$F_s/2$, con F_s = frequenza di campionamento (nel nostro caso pari a 20KHz), che determina un limite superiore oltre il quale non è possibile andare. Poiché abbiamo un set di campioni finito pari ad N , lo strumento matematico di cui si ha bisogno non è la trasformata di Fourier, che lavora con un segnale in ingresso definito su un insieme tempo-continuo, ma la trasformata di Fourier discreta, ovvero la DFT. La DFT lavora su collezioni di dati discrete raccolti entro una finestra temporale di durata finita. La relazione che ci permette di determinare la durata della finestra temporale quando andiamo a raccogliere un numero N di campioni con una certa frequenza di campionamento è $T = N \frac{1}{F_s}$. Se N fosse uguale a 20000, ad esempio, parlare di una finestra temporale di 1 secondo o di 20000 campioni raccolti a frequenza 20KHz sarebbe equivalente. Se abbiamo N campioni prelevati a frequenza F_s , otteniamo in uscita dalla DFT un numero discreto di oscillazioni sinusoidali. Lo spettro risulta quindi costituito da componenti sinusoidali a frequenze che variano tra 0 ed F_n , la cui distanza è pari a $\Delta f = \frac{F_n}{N/2}$, grandezza definita come risoluzione spettrale. L'equazione che mostra la definizione matematica della DFT è la seguente:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-2\pi i n k}{N}}$$

dove

- $x(n)$: il segnale tempo discreto.
- N : il numero di campioni
- n : l'indice tempo-discreto
- k : l'indice in frequenza

È possibile rendere più compatta l'equazione appena descritta definendo il seguente termine:

$$W_N = e^{\frac{-2i\pi}{N}}$$

3.2 L'algoritmo di ottimizzazione della FFT

La DFT è implementabile su un calcolatore, in quanto richiede un numero finito di passaggi per essere eseguita, il che la rende uno strumento di elaborazione digitale dei segnali estremamente utile. Il grande svantaggio nell'utilizzarla, tuttavia, risiede nel fatto che il tempo di calcolo

necessario per eseguirla è direttamente proporzionale ad N^2 . il che la rende molto inefficiente. Ecco perché, al posto della DFT, si preferisce utilizzare una versione ottimizzata, la FFT. La famiglia di algoritmi di ottimizzazione sviluppati per realizzare l'FFT consta di un gran numero di procedure atte a rendere più efficiente la DFT. Queste si dividono principalmente in due categorie: algoritmi di decimazione in tempo (DIT) ed algoritmi di decimazione in frequenza (DIF). La differenza principale tra queste due tipologie di algoritmi è che nel caso della decimazione in tempo, l'input viene decomposto in sottosequenze, che dividono la sequenza di campioni $x(n)$ in campioni pari e dispari. Nel caso della decimazione in frequenza, invece, è l'output che viene decomposto secondo questa procedura. Nel primo caso, quindi, l'operazione di divisione viene eseguita nel dominio del tempo, mentre nel secondo in quello della frequenza. Alla base di questi algoritmi vi è l'assunzione che il numero N di campioni sui quali si va ad eseguire la trasformata sia una potenza di 2. Al giorno d'oggi esistono delle procedure e delle routine per il calcolo di FFT la cui lunghezza non è un multiplo di 2, ma che sono meno efficienti e richiedono maggiore potenza di calcolo. Nel caso del lavoro di tesi, lavorando con un'architettura embedded, si è scelto di utilizzare un algoritmo basato su FFT di lunghezza pari ad un multiplo di 2. La procedura di ottimizzazione utilizzata per implementare la FFT è l'algoritmo DIF radix-2. Questa procedura si basa su un approccio del tipo divide-and-conquer: la DFT viene divisa in due sommatorie, la prima delle quali viene computata per i primi $N/2$ punti, e la seconda per la restante metà, ottenendo la seguente relazione:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn}$$

Sostituendo nella seconda sommatoria $n = n + \frac{N}{2}$ è possibile ricondursi ad una forma compatta

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + W_N^{KN/2} \sum_{n=0}^{(N/2)-1} x(n + \frac{N}{2})W_N^{kn}$$

È possibile verificare che $W_N^{KN/2} = (-1)^k$, e mettendo insieme le 2 sommatorie si ottiene questa relazione:

$$X(k) = \sum_{n=0}^{(N/2)-1} [x(n) + (-1)^k x(n + \frac{N}{2})]W_N^{kn}$$

A questo punto avviene la decomposizione dell'uscita nelle due sequenze di campioni pari e dispari.

$$X(2r) = \sum_{r=0}^{(N/2)-1} [x(r) + x(n + \frac{N}{2})] W_N^{2rk}$$

$$X(2r+1) = \sum_{r=0}^{(N/2)-1} [x(r) - x(n + \frac{N}{2})] W_N^{(2r+1)k}$$

Per eseguire queste semplificazioni basta verificare che $(-1)^{2r} = 1$ e che $(-1)^{(2r+1)} = -1$. Questo procedimento di decimazione può essere iterato un numero $d = \log_2 N$, cosicché alla fine avremo d stadi di decimazione. Quello che fa l'algoritmo radix-2 è dunque dividere l'intero processo computazionale della DFT in processi più piccoli di moltiplicazioni ed addizioni, chiamati "radix-2 butterflies":

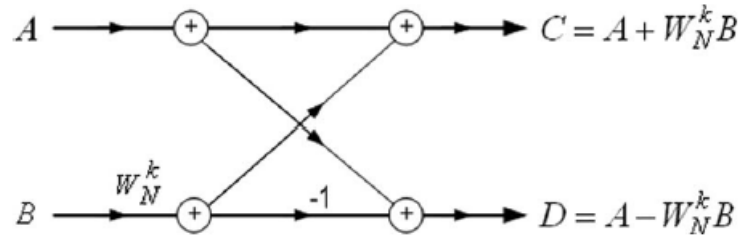


Figure 3.1: Radix-2 butterfly

Ad ogni stadio di decimazione avremo un numero pari ad $\frac{N}{2}$ di butterflies come quello mostrato in figura. Quello che bisogna evidenziare dell'applicazione di questo algoritmo di ottimizzazione è che attraverso il procedimento di decimazione in frequenza il numero di moltiplicazioni complesse da eseguire si riduce ad $\frac{N}{2} \log_2 N$, mentre il numero di addizioni complesse ad $N \log_2 N$, riducendo di molto il costo computazionale della DFT.

3.3 Utilizzo delle librerie DSP

Attraverso l'utilizzo di CubeMX, che consente di scaricare il plugin X-CUBE ALGOBUILD, è possibile implementare all'interno del progetto la libreria software CMSIS DSP. All'interno della libreria sono presenti diverse funzioni, ognuna delle quali copre una categoria specifica. Tra queste sono presenti anche quelle che permettono di implementare la FFT. Una volta inclusa la libreria, bisogna definire il numero di campioni rispetto ai quali viene eseguita la FFT (che deve essere una potenza di 2) e la dimensione del vettore che ospiterà i risultati della trasformata, che sarà pari alla metà del numero di campioni. Questo perché in uscita dalla FFT

avremo $N/2$ parti reali ed $N/2$ parti complesse di cui però bisogna effettuare il modulo, per cui l'output sarà costituito da $N/2$ valori. Nel caso preso in esame il numero di campioni scelto è $N = 2048$. L'ultima grandezza che resta da definire è la risoluzione spettrale che, come già visto in precedenza, è pari a:

$$\Delta f = \frac{F_n}{N/2}$$

Lo step sul quale bisogna soffermarsi è l'inizializzazione del modulo CFFT/CIFFT:

```
/* Inizializzo il modulo CFFT/CIFFT, ifftFlag = 0, doBitReverse = 1 */
arm_cfft_radix2_instance_f32 S;
arm_cfft_radix2_init_f32(&S, FFT_SIZE, 0, 1);
```

Figure 3.2: Inizializzazione modulo CFFT/CIFFT

La funzione di inizializzazione, oltre a prendere in la lunghezza della FFT, permette di settare altri due parametri. Il primo è l'ifft flag, che permette di impostare la direzione della trasformata; se settato ad 1 consente di eseguire la trasformata inversa. Il secondo, invece, è il bitReverseFlag. Se si assegna 1 a questo parametro, viene abilitata l'ordinamento inverso in termini di bit dell'output. Il motivo per cui questa operazione è richiesta si evidenzia osservando il diagramma che illustra in modo in cui opera il radix-2:

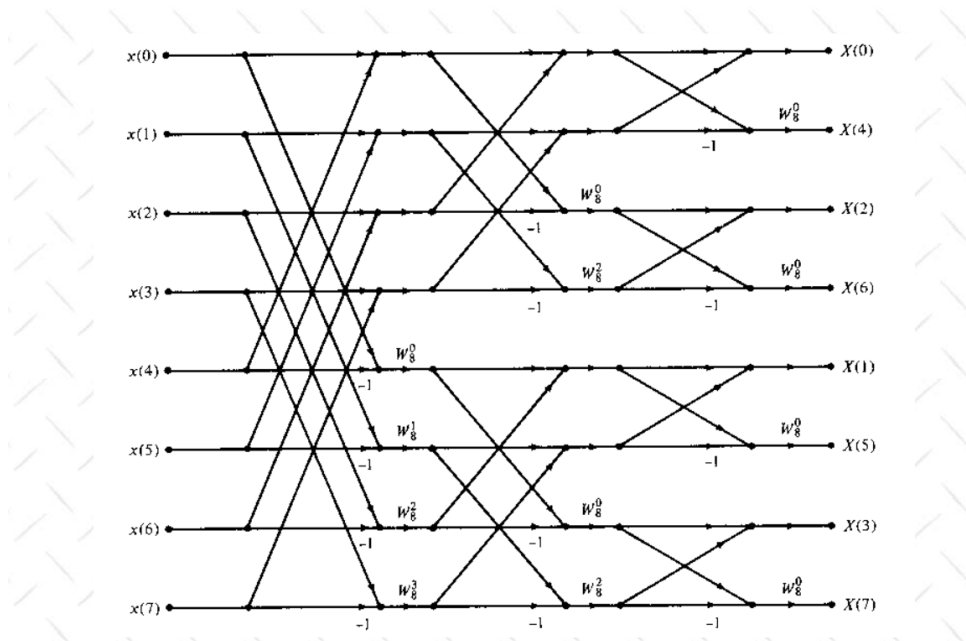


Figure 3.3: Decimazione in frequenza con N=8

Mentre la sequenza dei campioni in input è in ordine naturale, l'output è in ordine di bit

inverso, per cui risulta necessario settare il bitReverseFlag ad 1.

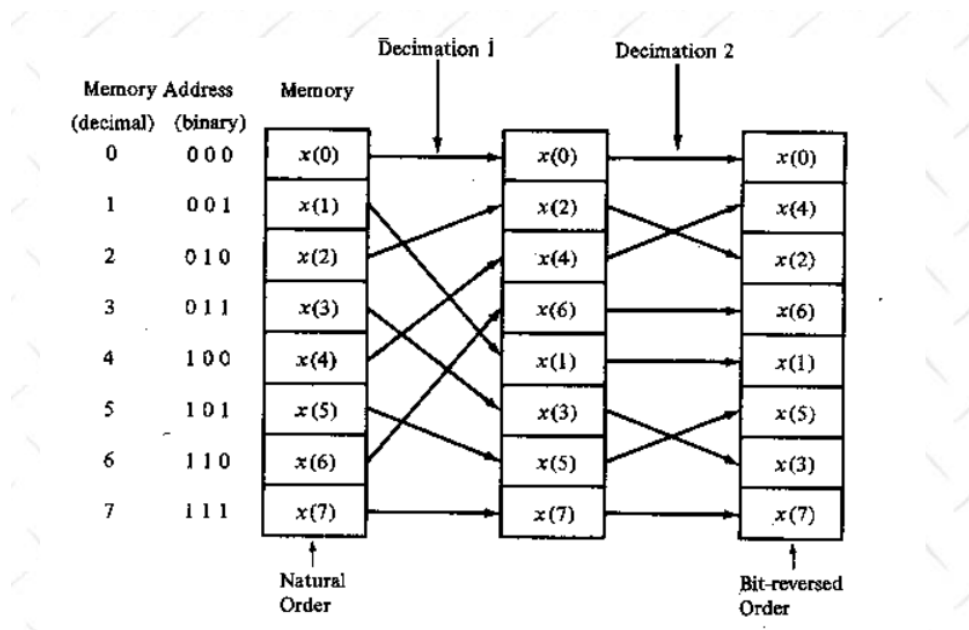


Figure 3.4: Ordine di bit dell'output del radix-2

A questo punto non resta che immagazzinare i dati che l'FFT dovrà processare all'interno di un buffer di input. Il modulo CFFT/CIFFT tratta il vettore in ingresso come se questo fosse diviso in coppie di valori, una parte reale ed una immaginaria, per cui all'atto in cui riempiamo il vettore bisogna opportunamente settare a zero la parte immaginaria, dal momento che i dati di input sono numeri reali. Fatto questo è possibile eseguire la FFT attraverso il radix-2, per poi usare una funzione apposita presente all'interno della libreria per ottenere i valori dei moduli dei numeri complessi in uscita:

```
//Raccolgo i campioni

for(int j=0;j<SAMPLES;j+=2){
//Parte reale
fft_in_buf_X[j] = ((float32_t)((float32_t)Vettx[j]));
//Parte immaginaria
fft_in_buf_X[j+1] = 0;
}

/* Processo i dati attraverso il modulo CFFT/CIFFT */
arm_cfft_radix2_f32(&S, fft_in_buf_X);

/* Processo i dati attraverso il modulo Complex Magnitude */
arm_cmplx_mag_f32(fft_in_buf_X, fft_out_buf_realX, FFT_SIZE);
```

Figure 3.5: Elaborazione campioni tramite il modulo CFFT/CIFFT

3.4 Limiti della FFT

La FFT risulta essere un grande miglioramento in termini di efficienza computazionale rispetto alla DFT. Tuttavia ci sono dei limiti che caratterizzano l'uso della FFT e l'accuratezza dei risultati che ne derivano, e sono principalmente legati al fenomeno della dispersione spettrale (spectral leakage). Come accennato in precedenza, quando utilizziamo la FFT, che è il risultato di un algoritmo di ottimizzazione della DFT, abbiamo in ingresso una collezione di campioni raccolti con una certa frequenza di campionamento F_s entro una finestra temporale la cui durata è finita. Vi sono quindi tre fattori che sono di importanza fondamentale per capire come interpretare i valori di uscita in termini di ampiezze e frequenze corrispondenti quando viene eseguita una FFT: la finestra temporale di osservazione, il numero di campioni raccolti e la frequenza di campionamento. Consideriamo un caso esemplificativo molto semplice, una sinusoide di ampiezza unitaria che ha frequenza pari a 10 Hz che viene campionata ad una frequenza di 100Hz e con una finestra di osservazione pari ad 1 secondo. Se riprendiamo la formula usata in precedenza per il calcolo della finestra temporale e ribaltandola per calcolare il numero di campioni raccolti otteniamo $N = T \cdot F_s = 100$.

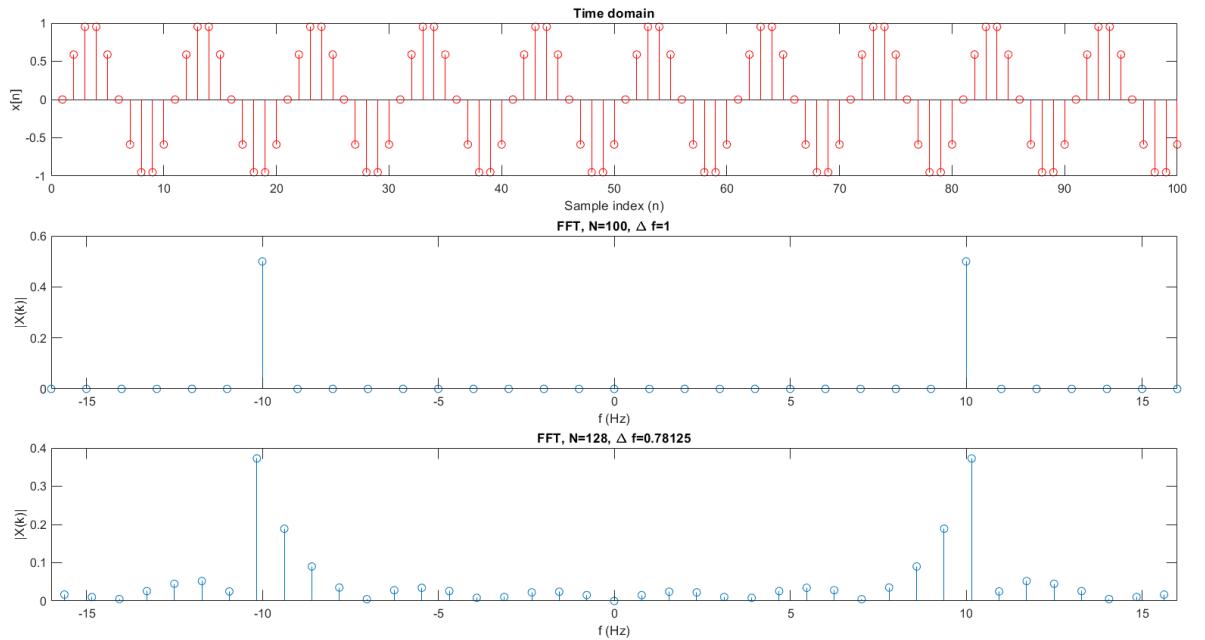


Figure 3.6: Esempio di spectral leakage

A questo punto consideriamo due casi: il primo è quello in cui eseguiamo una FFT su 100 punti, che corrisponde esattamente alla quantità di campioni del segnale, mentre nel secondo

utilizziamo la FFT assumendo $N = 128$, ovvero la potenza di 2 più vicina.

Osservando i due plot, si può osservare come nel secondo caso si ha una situazione essenzialmente diversa dalla prima. Quello che ci si aspetta infatti è di avere due impulsi a frequenza 10 e -10, ognuno dei quali con ampiezza pari a 0.5, che è quello che succede nel secondo caso. Quando invece eseguiamo l'FFT con $N = 128$ campioni, osserviamo che ci sono dei valori di ampiezza diversi da 0 in corrispondenza di frequenze diverse da 10 e -10. Questo fenomeno è dovuto alla differente risoluzione spettrale, che nel secondo caso è pari a 0.78125 e non ad 1 come nel primo. Il fatto che la risoluzione spettrale non sia unitaria si traduce nel fatto che sull'asse x non si potrà mai avere una frequenza pari esattamente a 10Hz. Il massimo che si può fare è prendere le due frequenze più vicine, che in questo caso sono 9.375Hz e 10.15Hz. Ecco perché attorno a 10Hz (e a -10Hz) sembra che lo spettro si sia "distribuito". È possibile quindi ricondurre il fenomeno dello spectral leakage al disallineamento tra la frequenza del segnale e la risoluzione spettrale (che dipende dalla frequenza di campionamento e dalla lunghezza della FFT).