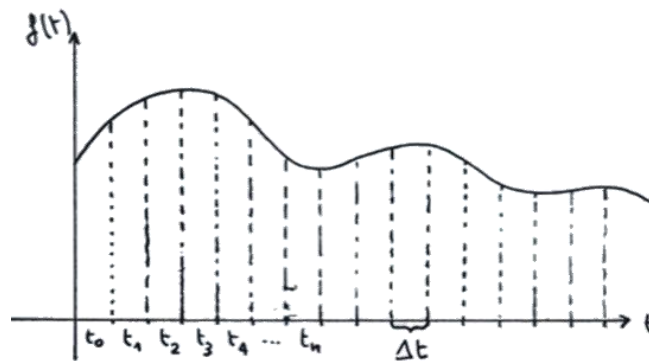


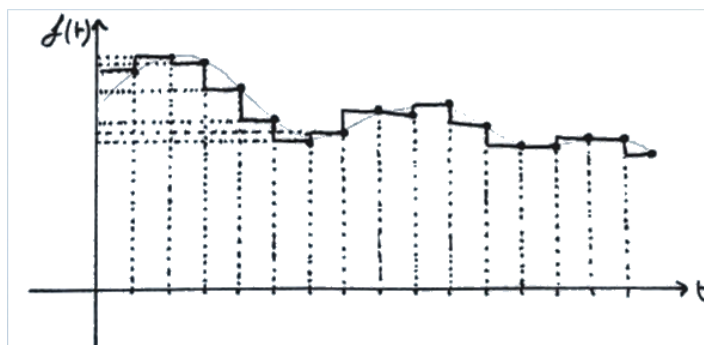


1) Descrivere (possibilmente attraverso un grafico) i concetti di discretizzazione e campionamento necessari alla trasformazione di una grandezza analogica continua nel tempo in una grandezza digitale.

Per trasformare una grandezza analogica continua nel tempo (rappresentabile con una curva irregolare), in una grandezza digitale comprensibile e manipolabile da un Sistema digitale, è necessario ricorrere ad operazioni di campionamento e discretizzazione dei valori reali continui. Il campionamento permette di passare da un numero infinito di valori ad un finito, suddividendo la realtà (un dato reale continuo) in intervalli di tempo (Δt). Il grafico di una funzione reale si presenta come una **curva irregolare** difficile da comprendere. Se suddividiamo il tempo in intervalli definiti otteniamo un numero finito di campioni. La frequenza di campionamento è data da $1/\Delta t$. Ad esempio la frequenza di campionamento della voce umana è di 8 kHz, quindi viene utilizzato un intervallo di campionamento (Δt) di 0.125 secondi.



Il passo successivo è rendere i campioni reali ottenuti, numeri discreti con un numero finito di cifre in modo da essere codificati da sistemi digitali. La discretizzazione è un'operazione in cui numeri reali, ad infinite cifre, sono approssimati al numero discreto più vicino, fissando a priori la precisione, ovvero un margine di errore che verrà commesso. La discretizzazione può essere rappresentata con una **spezzata poligonale**, che rappresenta la curva irregolare della funzione reale, assegnando ad ogni campione un valore discreto, con margine di errore. In questo modo dal grafico possiamo ricavare campioni razionali.



2) Sottrarre il numero binario 1101 dal numero binario 0101011, interpretando entrambi i numeri come codificati in complemento a due, ed esprimere il risultato come numero in base dieci. Quale risultato si sarebbe ottenuto se si fossero considerati i numeri codificati in modulo?

$(0101011)_2$ = ha il bit significativo a 0, quindi il numero è positivo.

La decodifica in decimale sarà positiva: $(2^0 \times 1 + 2^1 \times 1 + 2^2 \times 0 + 2^3 \times 1 + 2^4 \times 0 + 2^5 \times 1 + 2^6 \times 0) = (+43)_{10}$

$(1101)_2$ = ha il bit significativo a 1, quindi il numero è negativo. Per sottrarlo all'altro numero binario devo negarlo e sommargli 1, in modo da renderlo positivo.

$$\begin{array}{r} \text{Negato: } (0010)_2 \rightarrow 0010 + \\ 0001 = \\ \hline 0011 \rightarrow (2^0 \times 1 + 2^1 \times 1 + 2^2 \times 0 + 2^3 \times 0) = (+3)_{10} \end{array}$$

Somma: $0101011 + (+43)_{10} =$

L'estensione del segno nelle operazioni in C2 viene fatta in base al bit del segno (0 per il +)

$$\begin{array}{r} 0000011 = (+3)_{10} = \\ \hline 0101110 \rightarrow (2^0 \times 0 + 2^1 \times 1 + 2^2 \times 1 + 2^3 \times 1 + 2^4 \times 0 + 2^5 \times 1 + 2^6 \times 0) = (+46)_{10} \end{array}$$

Se i numeri fossero stati codificati in modulo, sarebbero stati entrambi positivi, perché questa codifica non considera il bit di segno. La codifica in modulo rappresenta tutti i numeri interi assoluti all'interno del campo di esistenza $0 \leq N \leq 2^n - 1$, in cui n è il numero di bit su cui vogliamo rappresentare il numero.

Per la stringa $(0101011)_2$ la codifica sarebbe stata ugualmente $(+43)_{10}$, mentre la codifica della stringa $(1101)_2$ sarebbe stata $\rightarrow (2^0 \times 1 + 2^1 \times 0 + 2^2 \times 1 + 2^3 \times 1) = (+13)_{10}$

Il risultato della sottrazione sarebbe stato: $0101011 - (+43)_{10} =$

L'estensione del segno nelle operazioni in modulo viene fatta con gli 0 per non alterare la codifica

$$\begin{array}{r} 0001101 = (+13)_{10} = \\ \hline 0011110 \rightarrow (2^0 \times 0 + 2^1 \times 1 + 2^2 \times 1 + 2^3 \times 1 + 2^4 \times 1 + 2^5 \times 0) = (+30)_{10} \end{array}$$

3) Si definiscano, possibilmente aiutandosi con un esempio, i concetti di: host, programma, processo, applicazione di rete.

- **Host:** è un sistema di elaborazione hardware come un PC, uno smartphone, un sistema medicale collegato in rete o un qualsiasi dispositivo collegato in rete. È sempre identificato da un indirizzo IP a cui, opzionalmente, può essere associato un Domain Name System (DNS)
- **Programma:** è l'insieme delle istruzioni da eseguire per avviare un processo. Finché non viene eseguito è un file statico. Ad esempio un file di codice.
- **Processo:** è un programma in esecuzione sull'host che trasmette/riceve pacchetti (PDU) verso/da altri processi su altri host attraverso la rete. È identificato da un numero di porta a 16 bit nell'intervallo da 0 a 65535.
- **Applicazione Di Rete:** sono insiemi di processi sparsi sulla rete, in esecuzione su host diversi, che si scambiano informazioni attraverso la rete, per realizzare un servizio utile per l'utente. Ad esempio Web, chat, e-mail, telemedicina. Hanno meccanismi di base differenti: unsolicited message, in cui le informazioni trasmesse non sono richieste e sono unidirezionali, notificano un evento; o richiesta/risposta, in cui l'applicazione crea un dialogo tra processi. I modelli più utilizzati per lo scambio di messaggi sono: client/server e publisher/subscriber.

4) Si definiscano, possibilmente aiutandosi con un esempio, i concetti di: **implicante**, **implicante primo** e **implicante primo essenziale**.

- **Implicante:** è il prodotto dei letterali di una funzione, dello spazio Booleano d'ingresso, tale che quando il prodotto dei letterali vale 1 anche la funzione vale 1. È un sottocubo di soli mintermini, ovvero punti dello spazio booleano d'ingresso in cui la funzione vale 1, a distanza di Hamming unitaria.

a, b \ c, d	00	01	11	10
00	0	1	0	1
01	1	0	0	1
11	1	0	0	1
10	0	1	0	0

Gli implicanti in questa mappa di Karnaugh sono quelli cerchiati, ad esempio:

Implicante $a\bar{b}c$

Implicante $a\bar{b}d$

Implicante $\bar{a}bd$

Implicante $\bar{a}bd$

...

- **Implicante Primo:** è un implicante che copre il maggior numero di mintermini (punti dello spazio Booleano d'ingresso in cui la funzione vale 1) non coperti da nessun altro implicante primo. Quindi non esiste nessun altro implicante primo di dimensioni maggiori che lo contenga interamente.

a, b \ c, d	00	01	11	10
00	0	1	0	1
01	1	0	0	1
11	1	0	0	1
10	0	1	0	0

L'implicante primo A è $\bar{b}d$

Perché copre 4 implicanti ($a\bar{b}d + \bar{a}bd + \bar{b}cd + \bar{b}cd$)

Non ci sono altri implicanti primi di dimensioni maggiori

- **Implicante Primo Essenziale:** è un implicante primo che copre almeno un mintermine (punto dello spazio Booleano d'ingresso in cui la funzione vale 1) non coperto da nessun altro implicante primo della funzione data.

a, b \ c, d	00	01	11	10
00	0	1	0	1
01	1	0	0	1
11	1	0	0	1
10	0	1	0	0

Gli implicanti primi essenziali sono B($\bar{a}bd$), C($a\bar{b}c$)

Perché C copre anche il mintermine $a\bar{b}cd$ che non è coperto dall'implicante primo A. Mentre B copre due mintermini coperti solo da se stesso.

Inoltre possiamo considerare l'implicante primo A anch'esso essenziale perché copre mintermini non coperti da altri implicanti.