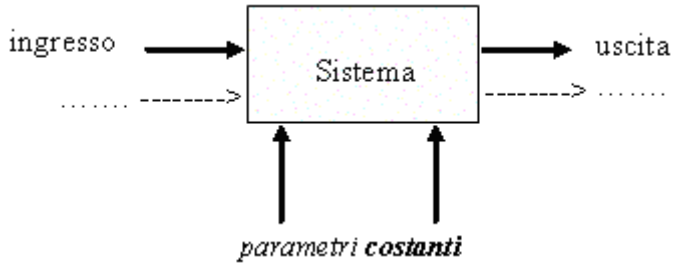
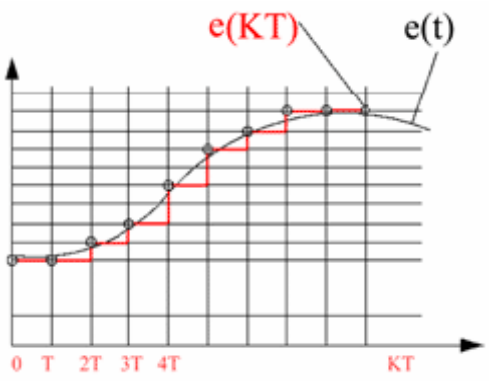
**Rappresentazione *sistemica*** (paradigma ingresso-uscita): descrizione a blocchi funzionali

Rappresentazione grafica che distingue tra variabili in ingresso *(grandezze su cui possiamo agire per introdurre modifiche)* e in uscita o risposte *(grandezze che risultano influenzate e possiamo osservare per studiare sperimentalmente l'andamento)* individuando gli eventuali *parametri costanti* :



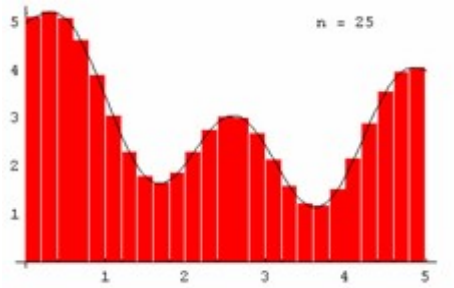

**nb**: Nel caso di simulazione al computer, uno dei *parametri* ***costanti*** è il tempo di discretizzazione **Δt** cioè l'intervallo tra campioni da fissare poiché un **esecutore discreto** è in grado di distinguere solo un **numero finito di valori** e non consente di analizzare infiniti valori istantanei.

**DISCRETIZZAZIONE DI UN SEGNALE ANALOGICO:** la discretizzazione è uno dei due processi che costituiscono la **digitalizzazione**. La discretizzazione può essere chiamata anche "quantizzazione" poichè sostituisce ai valori *continui* del segnale analogico dei valori *discreti* (quantificati, nel senso letterale di "quanto") cioè non-continui, "a salti. La figura qui sotto chiarisce quanto detto; **e(t)** rappresenta il segnale analogico continuo, mentre **e(KT)** quello discretizzato (o quantificato).

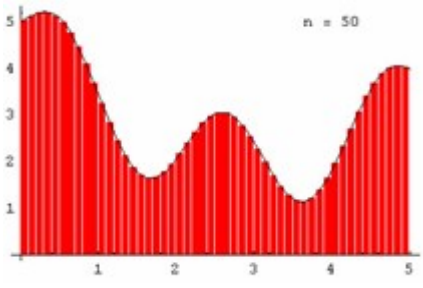



I circoletti indicano il valore che assume il segnale dopo la **discretizzazione**; collegandoli, supponendo noto il segnale esclusivamente negli istanti di tempo campionati, si ottiene un andamento "a gradini" rappresentativo del segnale **discretizzato** e **campionato**. Sull'asse verticale sono segnati i possibili salti (livelli) di valore che quantificano il segnale; più essi sono fitti, maggiore è la precisione della discretizzazione.

Caso numero campioni pari a 25




Caso numero campioni pari al doppio (scelta del tempo di discretizzazione **Δt** pari alla metà) : maggiore precisione




Tanto **minore è Δt**, tanto maggiore è la precisione

Maggiore è la "**precisione**" digitale (dimensione del digit) maggiore è lo spazio di memoria necessario a memorizzare i campioni ed il tempo per trasmetterli (in assenza di tecniche di compressione): nel fissare i criteri di scelta si dovrà anche tener conto dei limiti dell'operatore umano.

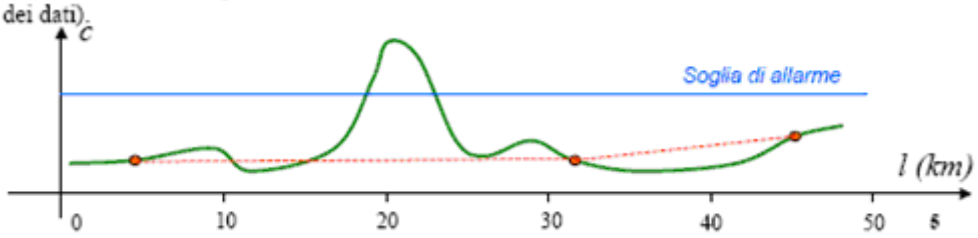
*Per approfondire*

## Il problema della frequenza di campionamento

Quando si raccolgono valori discreti e si desidera che siano rappresentativi della grandezza continua dalla quale sono estratti, è **fondamentale** che i punti o gli istanti di campionamento siano **sufficientemente vicini** tra loro.

Punti di campionamento troppo distanti possono dare luogo a significative **perdite di informazione**.

Punti di campionamento troppo vicini comportano uno **spreco di risorse** (tempo di misura, memoria per la conservazione dei dati).


Affinchè il maggior numero di campioni non richieda eccessivi spazi di memorizzazione e lentezze nella trasmissione si ricorre a tecniche di **compressione**.

### Compressione

La compressione può essere con perdita di informazione o *lossy* (ad esempio la compressione delle immagini) e senza perdita di informazione cioè fedele o *lossless* (necessaria per comprimere i dati).

Tra i metodi di compressione ricordiamo:

- quello a lunghezza di sequenza (*Run Length Encoding*) usato quando ci sono lunghe sequenze dello stesso simbolo; adatto per comprimere informazioni audio (intervalli di silenzio) o video (sequenze di pixel con stesso colore) o nei fax (sequenze di spazi). Invece di inviare molte volte lo stesso simbolo, si invia un solo simbolo e il numero di ripetizioni.
- quelli a lunghezza variabile ad esempio l'algoritmo di Huffman (1952) : si usa un codice breve per rappresentare i simboli più frequenti e più lungo per i simboli meno frequenti (come nell'alfabeto Morse)
- quello a dizionari di codifica: molto diffuso (usato da programmi come zip e arj): usa un buffer detto dizionario, inizialmente riempito col carattere che si presume più frequente (ad esempio in un testo lo spazio) e in cui vengono via via aggiunte nuove stringhe del testo; per la codifica, si cerca se una stringa è già presente nel buffer ed in tal caso si sostituisce con la coppia di valori (posizione, lunghezza) che la identifica nel buffer; in decodifica si segue il procedimento opposto ricostruendo man mano il buffer e il testo.

La compressione permette di aumentare la velocità di trasmissione riducendo la quantità di bit da trasmettere ed è sempre un processo a tre stadi: compressione, trasmissione, decompressione; questo comporta che l'apparato ricevente sia dotato della capacità di processare i segnali e non sia soltanto un "ricevitore" passivo.