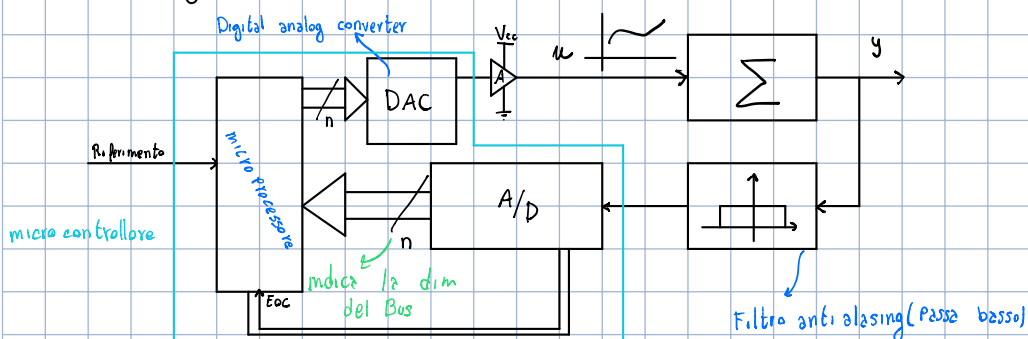


Le lezioni che seguono servono per vedere che tutto quello che abbiamo fatto può essere applicato ai sistemi digitali.

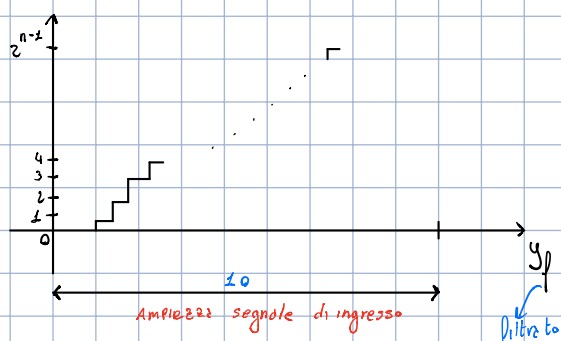
Prendiamo un generico processo



u e y sono analogici (continui)

che dobbiamo immettere nel sistema digitale

con n bit abbiamo 2^n simboli diversi.



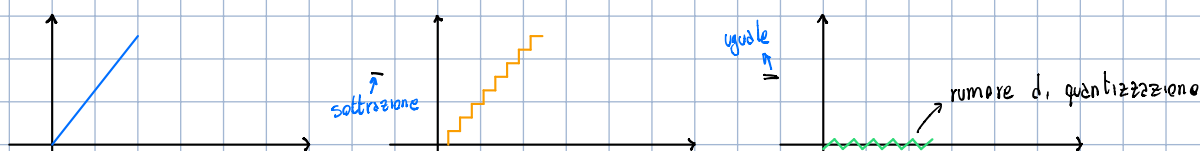
Il segnale deve essere quantizzato cioè si deve associare a ogni simbolo un valore riconoscibile. Spesso i convertitori A/D sono variabili, cioè si può scegliere un range di valori con cui rappresentare il segnale di ingresso.

Scelto un numero di bit n per il convertitore A/D si può calcolare un valore q che è detto "passo di quantizzazione" $q = \frac{\text{Amplitude segnale di ingresso}}{2^n}$. Si nota che all'aumentare di n diminuisce il passo di quantizzazione

e si nota pure che potrebbe accadere che la variazione di un simbolo rispetto al successivo non è abbastanza da superare il passo di quantizzazione perdendo il simbolo. Questo è un problema. Bisogna trovare un compromesso tra la giusta precisione del passo di quantizzazione e la quantità di disturbo che non deve essere campionato nel sistema. Del sistema precedente i due componenti che hanno questo problema sono A/D e il DAC.

Il problema si presenta perché ci troviamo in catena di controreazione e quindi deve essere fatto tutto alla perfezione. Quindi non possiamo avere un'uscita che ha una precisione migliore del passo di quantizzazione.

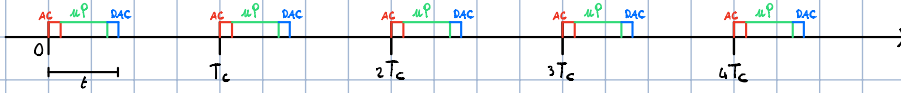
In alcuni casi questo difetto di precisione viene trattato come se fosse un rumore.



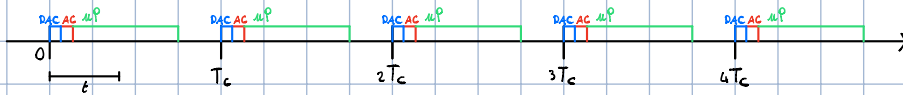
Delle volte l'approccio che si usa è di lavorare come se non ci fosse la quantizzazione e di valutare alla fine il rumore di quantizzazione.

Si noti che tutto quello che è all'interno del perimetro solitamente sono dentro lo stesso integrato.

Il microcontrollore deve necessariamente funzionare a tempo discreto.



La problematica è: Come scelgo T_c in modo corretto? Ricordiamoci che prendere un T_c troppo grande non fa altro che peggiorare la replica campionata del segnale. Inoltre notiamo che nel disegno precedente non stiamo sfruttando bene il tempo di campionamento poiché abbiamo un tempo $(T_c - t)$ inutilizzato. Questo può essere risolto cambiando approccio, introducendo un ritardo di un campione



Così facendo si ha tutto il tempo di campionamento per l'esecuzione dell'elaborazione nel microprocessore.

Anche con questa osservazione non siamo riusciti ad avere qualche indicazione su come scegliere il tempo di campionamento.

Ci serve il teorema del campionamento di Shannon per capire come fare. Lo vedremo la prossima lezione.