Tesina del corso di
IDENTIFICAZIONE DEI MODELLI ED ANALISI DEI DATI
27 maggio 2015

Istruzioni: Scrivere una relazione corredata da grafici che descriva lo svolgimento. La tesina deve essere OBBLIGATORIAMENTE consegnata nel mio studio (o in segreteria didattica nei giorni precedenti) entro **luned 22 giugno** per coloro che vogliono svolgere l'esame il 23 giugno ed entro **venerdi 17 luglio** per chi svolgerà l'esame il 20 luglio. Per la prova di settembre saranno comunicate le date in seguito. In ogni caso la relazione dovrà essere consegnata 3-4 giorni prima dell'esame.

Il giorno dell'esame (orale) **dovrete presentarvi con i files matlab** utilizzati per svolgere la relazioni (ad esempio con una chiavetta USB).

Esercizio 1. [Costruzione di un codificatore di tipo DPCM] I dati forniti $\{y_t\}_{t=1,.,N}$ (file `dpcm-dati-xx.mat`) sono una traiettoria di un processo y_t che può essere modellato tramite un modello autoregressivo del tipo

$$y_t = a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + v(t) \tag{1}$$

dove $v(t)$ è bianco, a media nulla e varianza σ_v^2 incognita.Tuttavia i coefficienti a_1, a_2, a_3 sono lentamente varianti nel tempo. Si vuole progettare un codificatore per il segnale y_t che utilizzi il minimo numero di bits, con un fissato errore di ricostruzione. In particolare si vuole codificare y_t con un segnale y_t^Q (segnale quantizzato) in modo che la distorsione

$$d_t^Q := y_t - y_t^Q$$

sia piccola (ad esempio imponendo che $Var\{d_t^Q\} \leq \sigma_{max}^2$).

Sia $P(z)$ la funzione di trasferimento di un sistema lineare, con $P(\infty) = 0$, che viene utilizzato per eseguire un "pre-processing" dei dati:

$$\hat{y}_{t|t-1}^P := P(z) y_t^Q$$

A partire dal segnale $\hat{y}_{t|t-1}^P$ si costruisce il segnale di "l'errore" $e_t := y_t - \hat{y}_{t|t-1}^P = y_t - P(z) y_t^Q$ che viene quantizzato $\mathcal{Q}$

$$e_t^Q := \mathcal{Q}[e_t]$$

dove $\mathcal{Q}$ è la mappa che definisce il quantizzatore. Noi assumeremo di usare un quantizzatore uniforme, con soglie $\pm T_Q$ e N_L livelli, definito dalla il seguente relazione

$$\mathcal{Q}[x] = \begin{cases} T_Q & x > T_Q \\ \frac{\Delta_Q}{2} + \Delta_Q \cdot \lfloor \frac{x}{\Delta_Q} \rfloor & |x| \leq T_Q \\ -T_Q & x < -T_Q \end{cases}$$

dove $\Delta_Q := \frac{2T_Q}{N_L - 1}$.

Uno schema per ridurre il numero di bits necessari per la codifica del segnale y_t è il così detto *Differential Pulse Code Modulation (DPCM)*, il cui funzionamento è descritto dalle seguenti equazioni (si veda anche lo schema di sinistra in Figura 1):

$$\begin{aligned} e_t &= y_t - \hat{y}_{t|t-1}^P = y_t - P(z) y_t^Q && (2) \\ e_t^Q &= \mathcal{Q}[e_t] && (3) \\ y_t^Q &= P(z) y_t^Q + e_t^Q && (4) \end{aligned}$$

Schema con Quantizzatore

Schema con Rumore equivalente

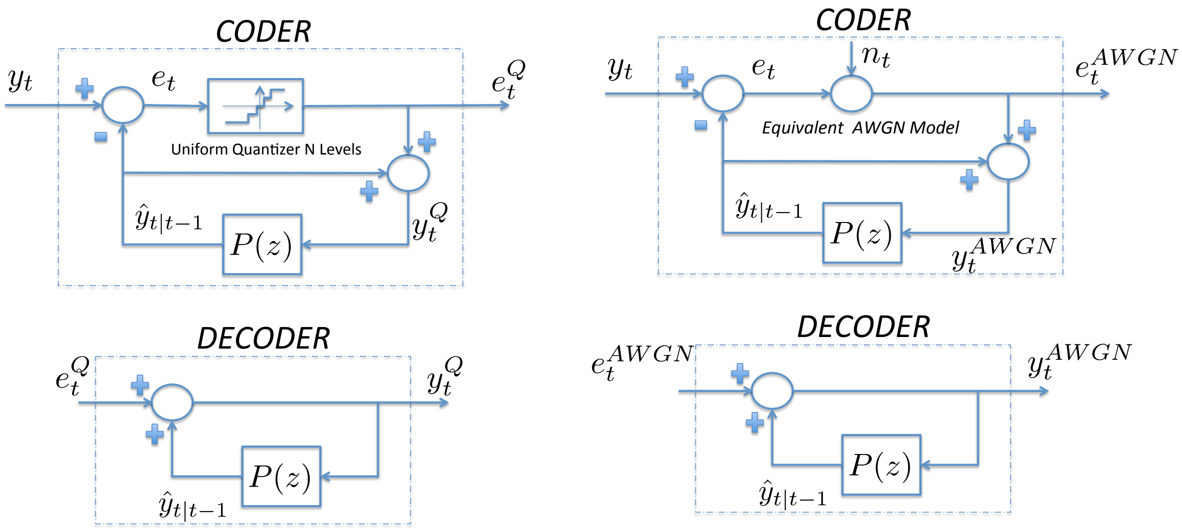

Figure 1: Schema di Codifica e Decodifica: a sinistra lo schema che si implementa in pratica con il quantizzatore uniforme. A destra lo schema con il modello additivo per l'errore di quantizzazione.

Le operazioni esequite dal codificatore (COD) e decodificatore (DEC) si possono riassumere come segue:

COD: Riceve il segnale y_t, esegue il prepocessing in (2) calcolando il segnale e_t, ne calcola la versione quantizzata e_t^Q utilizzando (3) e, infine, ricostruisce il segnale quantizzato y_t^Q necessario per il prepocessing (2) al passo successivo.

DEC: Riceve il segnale quantizzato e_t^Q e si costruisce la versione quantizzata y_t^Q del segnale y_t utilizzando l'equazione (4)

Si risponda alle seguenti domande:

1. Si dimostri che l'errore di quantizzazione $d_t^Q = y_t^Q - y_t$ soddisfa l'equazione
$$d_t^Q = e_t^Q - e_t = n_t$$

2. Si simuli il comportamento del sistema utilizzando un quantizzatore uniforme con N_b bits, con soglie definite da $\pm T_Q := \pm k\sqrt{\mathrm{Var}\{e_t\}}$ ($k \in [3,4]$) e assumendo che $P(z) = z^{-1}$

3. Utilizzando un algoritmo di identificazione ricorsivo (ad esempio i minimi quadrati con fattore di oblio oppure il filtro di Kalman) si costruisca uno stimatore dei coefficienti a_k del modello (1) assumendo siano "lentamente" tempo varianti.

4. Assumendo che il valore di soglia sia dato da $T_Q = k\sqrt{\mathrm{Var}\{e_t\}}$ ($k \in [3,4]$), e che si utilizzino N_b bits (e quindi $N_L = 2^{N_b}$), si può dimostrare che l'errore di quantizzazione $n_t = e_t^Q - e_t$ si può modellare come un processo bianco, a media nulla, e varianza
$$\sigma_Q^2 = \frac{\Delta_Q^2}{12} \qquad \Delta_Q = \frac{2T_Q}{2^{N_b}-1} = \frac{2k\sqrt{\mathrm{Var}\{e_t\}}}{2^{N_b}-1}$$
dove N_b è il numero di bits utilizzato dal quantizzatore.
Utilizando le stime dei coefficienti a_k al punto precendente, si costruisca un modello

(tempo-variante) per il processo y_t e, ricorrendo all'algoritmo del filtro di Kalman, si progetti il filtro ottimo $P_{ott}(z)$ in modo che

$$P_{ott}(z)y_t^Q = \mathbb{E}\left[y_t | y_s^Q, s < t\right].$$

Suggerimento: si osservi che

$$\sigma_Q^2 = \mathrm{Var}\{n_t\} = \frac{4k^2 \mathrm{Var}\{e_t\}}{12 \cdot (2^{N_b} - 1)^2} = \alpha \mathrm{Var}\{e_t\} \quad \alpha = \frac{k^2}{3(2^{N_b} - 1)^2}$$

con

$$e_t := y_t - P_{ott}(z)y_t^Q = y_t - \mathbb{E}\left[y_t | y_s^Q, s < t\right]$$

5. Si simuli il comportamento di questo sistema confrontando l'errore di ricostruzione ottenuto con il quantizzatore e $P(z)$ utilizzati al punto 3 e quello ottenuto sostituendo al quantizzatore il modello lineare additivo (Additive White Gaussian Noise, AWGN):

$$e_t^{AWGN} = e_t + n_t \quad n_t \simeq \mathcal{N}(0, \sigma_Q^2)$$

cioè usando il modello:

$$\begin{aligned} e_t &= y_t - \hat{y}_{AWGN}(t|t-1) = y_t - P_{ott}(z)y_t^{AWGN} & (5)\\ e_t^{AWGN} &= e_t + n_t & (6)\\ y_t^{AWGN} &= P_{ott}(z)y_t^{AWGN} + e_t^{AWGN} & (7) \end{aligned}$$

I due schemi sono descritti in Figura 1. Studiare, al variare del numero di bits, l'accuratezza del modello additivo del quantizzatore. In particolare, quanti bits per campione sono necessari affinchè il modello lineare additivo sia ragionevole, cioè affinchè

$$\frac{1}{N}\sum_{t=1}^{N}(y_t - y_t^Q)^2 \simeq \frac{1}{N}\sum_{t=1}^{N}(y_t - y_t^{AWGN})^2$$

OSSERVAZIONE: *Si noti che lo schema prevede che il* Decoder *abbia accesso ai coefficienti $\hat{a}_1$, $\hat{a}_2$, $\hat{a}_3$ del* Coder. *Quindi i coefficienti devono essere trasmessi assieme ai dati. Chiaramente questo richiede un costo di comunicazione. Quindi in realt non è vantaggioso utilizzare coefficienti $\hat{a}_k$ tempo varianti, ma si utilizzano coefficienti fissi in finestre di opportuna lunghezza entro le quali il segnale si può considerare stazionario. Quindi è necessario spedire al decoder i coefficienti una sola volta all'inizio della finestra all'interno della quale sono costanti. Questo schema (con l'update ricorsivo dei parametri a_k) è stato proposto come esercitazione solo allo scopo di far pratica con l'utilizzo dei Minimi Quadrati Ricorsivi.*

Esercizio 2. [Progetto di un sensore virtuale]

Si consideri lo schema in Figura 2. Lo scopo finale è quello di progettare un *sensore virtuale* che sia in grado di stimare l'andamento della grandezza z_t a partire da misure di y_t e u_t, in presenza di un ingresso di "disturbo" d_t incognito. Si assuma che il disturbo n_t sia bianco e di varianza incognita (ad esempio potrebbe essere dovuto ad un errore di quantizzazione) e che i processi u_t, d_t ed n_t siano tra di loro indipendenti.

Il progetto si divide in due parti: la prima prevede la possibilità di eseguire un esperimento con dei sensori "aggiuntivi" (che mi permettono di misurare anche z_t e ξ_t) al termine del quale sono in grado di ottenere una stima delle funzioni di trasferimento $G_1(z)$, $G_2(z)$ e $G_3(z)$ e di un modello del disturbo d_t. Nella seconda parte, si devono utilizzare le stime ottenute per progettare uno stimatore (sensore virtuale) di z_t. In particolare:

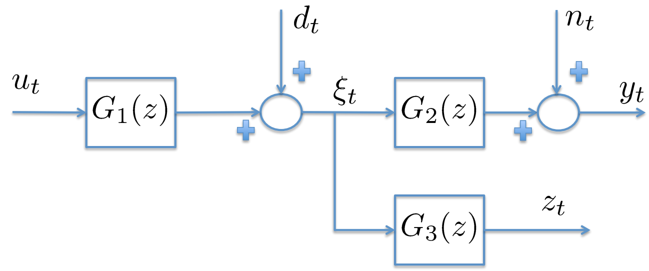

Figure 2: Schema per il progetto del sensore virtuale

1. Il set di dati nel file `virtual-sensor-dati-id-xx.mat` contiene traiettorie dei segnali $\{u_t, \xi_t, y_t, \bar{z}_t\}_{t=1,..,N}$ dal sistema in Figura 2 con $\bar{z}_t = z_t + v_t$, dove v_t è un processo stazionario, indipendente da u_t, n_t e d_t. Si trovi una stima[1] di $G_1(z)$, $G_2(z)$ e $G_3(z)$ e di un modello del disturbo d_t.

2. Utilizzando l'algoritmo del filtro di Kalman, si progetti un sensore virtuale che sia in grado di stimare la grandezza non misurabile z_t sulla base delle misure u_t e y_t, tenendo conto della presenza del disturbo d_t (non misurabile). I dati ingresso-uscita sono contenuti nel file `virtual-sensor-dati-kalman-xx.mat`. Ovviamente il filtro di Kalman dovrà essere opportunamente tarato (ad esempio il rumore n_t presente sui dati utilizzati in questo secondo punto potrebbe avere una varianza diversa da quella dei dati utilizzati al punto 1. (Ad esempio perchè si utilizza un quantizzatore diverso o perchè il sensore che misura y_t ha precisione diversa da quello utilizzato al punto 1)

[1] Potrà essere utile utilizzare il *System Identification Toolbox* di Matlab.