Problemi implementativi di Controllori Digitali

Gianmaria De Tommasi[1]

[1]Università degli Studi di Napoli Federico II
detommas@unina.it

Ottobre 2012

Corsi AnsaldoBreda

Outline

1. Realizzazioni dell'algoritmo di controllo
2. Implementazione su Digital Signal Processor
3. Messa in scala dell'algoritmo di controllo

Controllori e filtri digitali

Un controllore digitale è un **filtro digitale ricorsivo**, rappresentabile da un'equazione alle differenze del tipo

$$u_k = \sum_{i=1}^{n} a_i u_{k-i} + \sum_{j=0}^{m} b_j e_{k-j}$$

Controllori e filtri digitali

Un controllore digitale è un **filtro digitale ricorsivo**, rappresentabile da un'equazione alle differenze del tipo

$$u_k = \sum_{i=1}^{n} a_i u_{k-i} + \sum_{j=0}^{m} b_j e_{k-j}$$

Scrivendo l'algoritmo di controllo in termini di **operatore di ritardo** z^{-1}, si ottiene

$$u_k \left(1 - \sum_{i=1}^{n} a_i z^{-i}\right) = \sum_{j=0}^{m} b_j z^{-j} e_k$$

ovvero

$$u_k = \frac{\sum_{j=0}^{m} b_j z^{-j}}{1 - \sum_{i=1}^{n} a_i z^{-i}} e_k = D(z) e_k$$

Funzione di trasferimento e sequenza ponderatrice

Funzione di trasferimento discreta

$$D(z) = \frac{\sum_{j=0}^{m} b_j z^{-j}}{1 - \sum_{i=1}^{n} a_i z^{-i}} = \frac{b_0 z^m + b_1 z^{m-1} + \ldots + b_m}{z^n + a_1 z^{n-1} + \ldots + a_n}$$

Funzione di trasferimento e sequenza ponderatrice

Funzione di trasferimento discreta

$$D(z) = \frac{\sum_{j=0}^{m} b_j z^{-j}}{1 - \sum_{i=1}^{n} a_i z^{-i}} = \frac{b_0 z^m + b_1 z^{m-1} + \ldots + b_m}{z^n + a_1 z^{n-1} + \ldots + a_n}$$

Sequenza ponderatrice del controllore

$$D(z) = \sum_{k=0}^{\infty} d_k z^{-k}$$

con d_k sequenza ponderatrice del controllore.

Rumore di quantizzazione e rumore di moltiplicazione - 1

In un anello di controllo possono presentarsi particolari problemi derivanti

- dalle limitazioni nella risoluzione dei convertitori A/D e D/A
- dalla precisione di elaborazione del processore su cui viene eseguito il codice del controllore

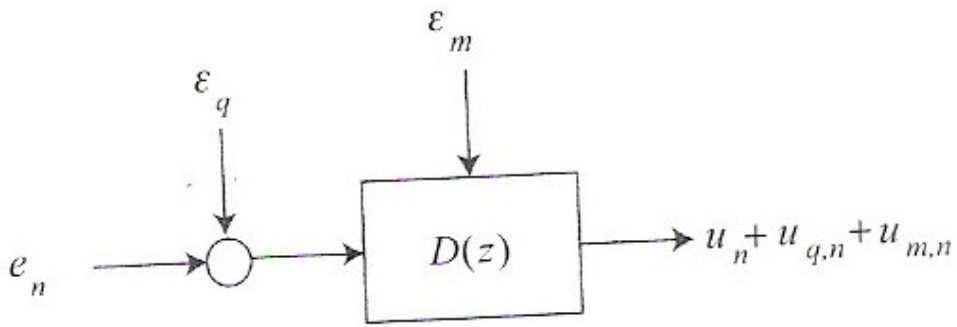
Rumore di quantizzazione e rumore di moltiplicazione - 1

In un anello di controllo possono presentarsi particolari problemi derivanti

- dalle limitazioni nella risoluzione dei convertitori A/D e D/A
- dalla precisione di elaborazione del processore su cui viene eseguito il codice del controllore

In entrambi i casi, gli effetti di tali problemi possono essere descritti attraverso rumori additivi

Rumore di quantizzazione e rumore di moltiplicazione - 2



rumore di quantizzazione ε_q - è dovuto alla rappresentazione delle grandezze con un numero finito di cifre. È caratterizzato da valor medio $\bar{\varepsilon}_q$ e varianza σ_q^2

rumore di moltiplicazione ε_m - è dovuto alla precisione finita dell'unità di elaborazione nella moltplicazione tra parametri e valori di una variabile. È caratterizzato da valor medio $\bar{\varepsilon}_m$ e varianza σ_m^2

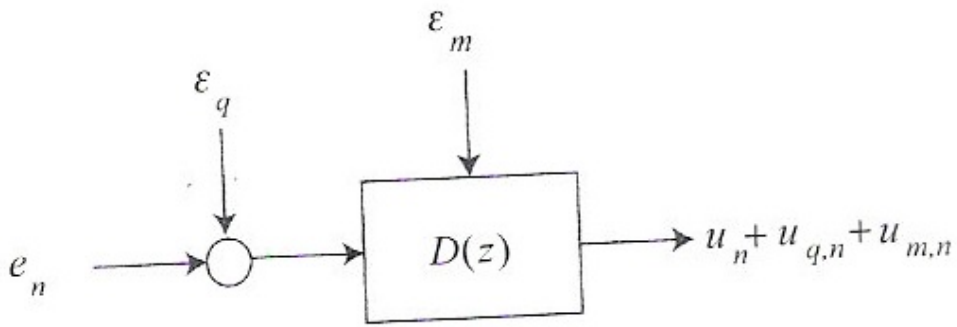
Rumore di quantizzazione e rumore di moltiplicazione - 3

- Il processo di sintesi porta alla definizione matematica dell'algoritmo di controllo

Rumore di quantizzazione e rumore di moltiplicazione - 3

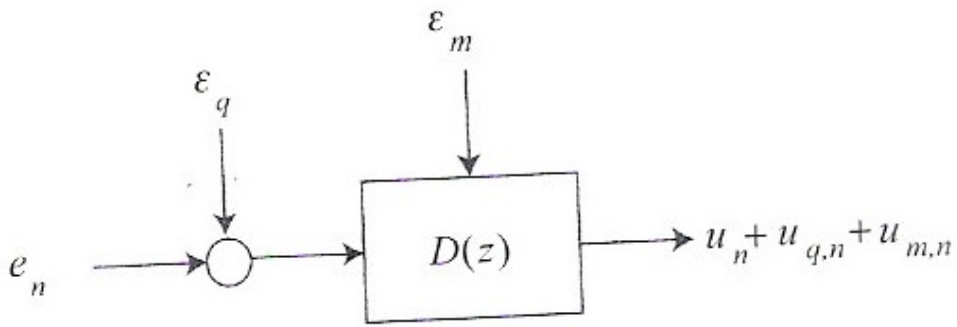
- Il processo di sintesi porta alla definizione matematica dell'algoritmo di controllo
- Il processo di implementazione software deve mirare a
  - ridurre il ritardo dovuto all'elaborazione
  - **minimizzare gli effetti dovuti alla propagazione degli errori di quantizzazione (sia dovuti alla conversione che alla moltiplicazione)**

Propagazione degli errori - 1



- Il controllore riceve in ingresso l'errore di controllo e_k sommato ai campioni del rumore di quantizzazione $\varepsilon_{q,k}$

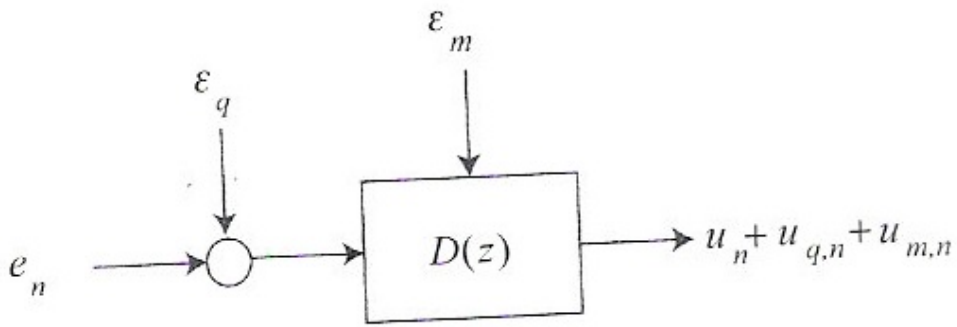
Propagazione degli errori - 1



- Il controllore riceve in ingresso l'errore di controllo e_k sommato ai campioni del rumore di quantizzazione $\varepsilon_{q,k}$
- Il rumore di moltiplicazione è dovuto alla moltiplicazione di un coefficiente costante con un segnale $\rightarrow$ ogni qualvolta si effettua una moltiplicazione nella realizzazione dell'algoritmo di controllo si introduce un termine additivo di rumore ε_m^i

$$\alpha_i u_{k-i} \rightarrow \alpha_i u_{k-i} + \varepsilon_m^i$$

Propagazione degli errori - 2



Per la linearità dell'algoritmo di controllo, l'uscita del controllore sarà data dalla somma di tre termini

$$u_k + u_{q,k} + u_{m,k}$$

con

- u_k uscita in assenza di rumore dovuta all'errore di controllo e_k
- $u_{q,k}$ uscita dovuta al solo rumore di quantizzazione (conversione)
- $u_{m,k}$ uscita dovuta ai rumori di moltiplicazione

Propagazione degli errori - 3

Per effettuare l'analisi di propagazione degli errori si consideri il controllore del secondo ordine

$$D(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} = D_1(z) + D_2(z) = \frac{\beta_1}{1 - \alpha_1 z^{-1}} + \frac{\beta_2}{1 - \alpha_2 z^{-1}}$$

con

$$b_0 = \beta_1 + \beta_2 \,, \quad b_1 = -\left(\beta_1 \alpha_2 + \beta_2 \alpha_1\right)$$

$$a_1 = -\left(\alpha_1 + \alpha_2\right) \,, \quad a_2 = \alpha_1 \alpha_2$$

Realizzazione parallela

- Il software di controllo relativo a $D(z) = D_1(z) + D_2(z)$ può essere realizzato in diversi modi

Realizzazione parallela

- Il software di controllo relativo a $D(z) = D_1(z) + D_2(z)$ può essere realizzato in diversi modi
- Si parla di **realizzazione parallela** quando $D(z)$ è implementata lasciando separate le funzioni di trasferimento $D_1(z)$ e $D_2(z)$

Realizzazione parallela

- Il software di controllo relativo a $D(z) = D_1(z) + D_2(z)$ può essere realizzato in diversi modi
- Si parla di **realizzazione parallela** quando $D(z)$ è implementata lasciando separate le funzioni di trasferimento $D_1(z)$ e $D_2(z)$
- In questo caso si possono avere due varianti
  - realizzazione parallela con pre-moltiplicazione
  - realizzazione parallela con post-moltiplicazione

Realizzazione parallela

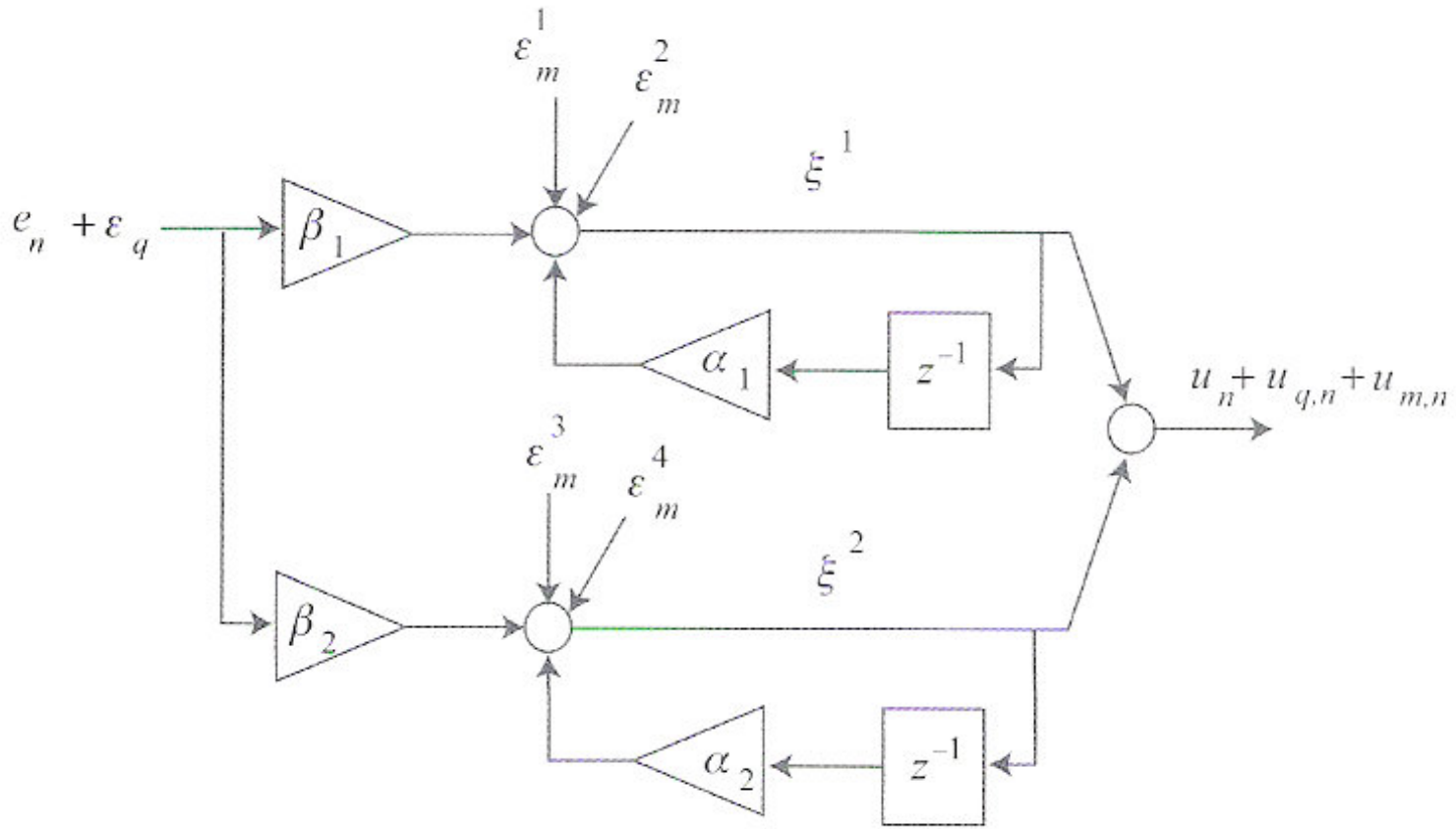
- Il software di controllo relativo a $D(z) = D_1(z) + D_2(z)$ può essere realizzato in diversi modi
- Si parla di **realizzazione parallela** quando $D(z)$ è implementata lasciando separate le funzioni di trasferimento $D_1(z)$ e $D_2(z)$
- In questo caso si possono avere due varianti
  - realizzazione parallela con pre-moltiplicazione
  - realizzazione parallela con post-moltiplicazione
- Se il controllore viene implementato con l'unica funzione di trasferimento $D(z)$ si parla di **realizzazione diretta**

Realizzazione parallela

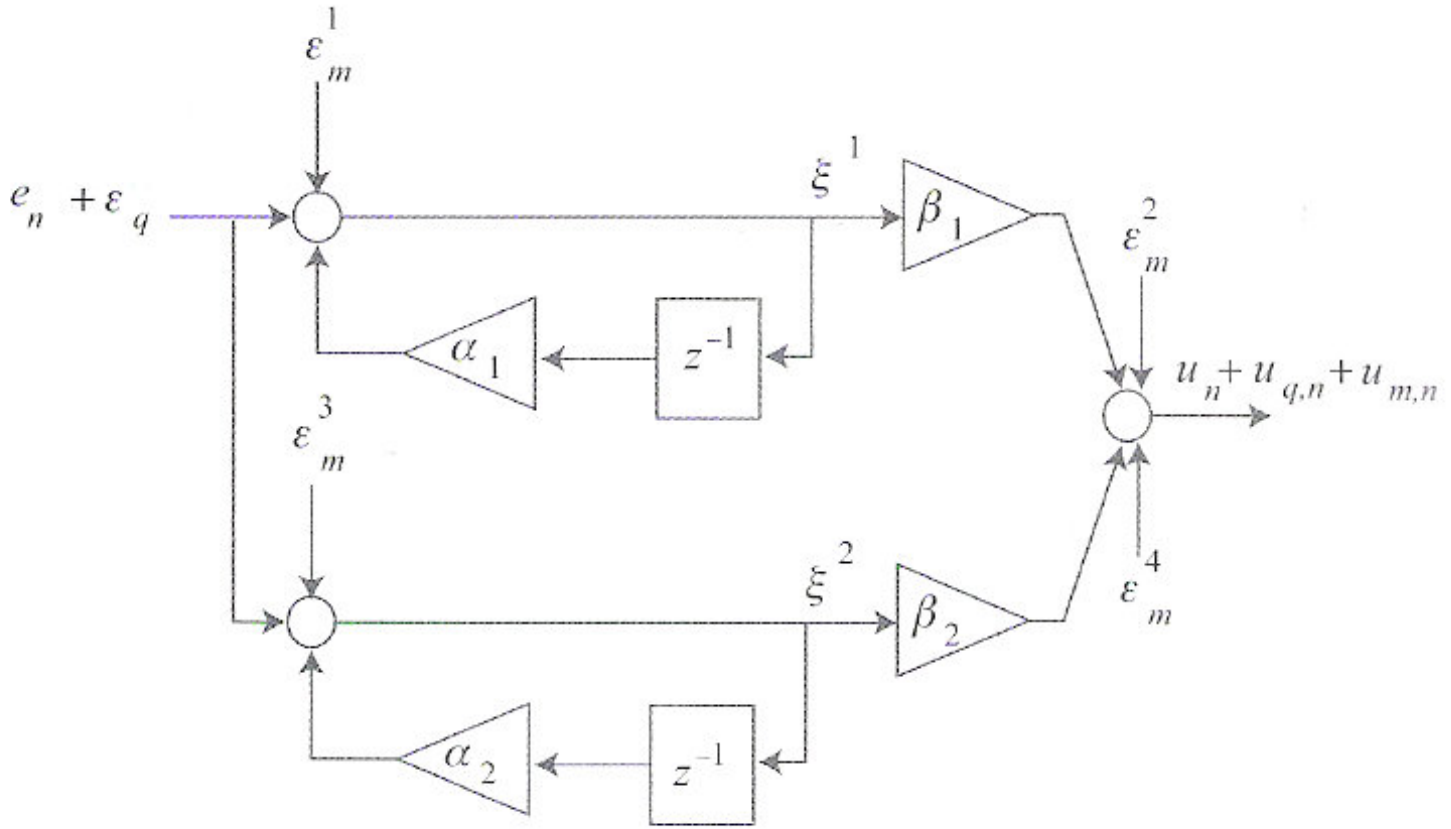
- Il software di controllo relativo a $D(z) = D_1(z) + D_2(z)$ può essere realizzato in diversi modi
- Si parla di **realizzazione parallela** quando $D(z)$ è implementata lasciando separate le funzioni di trasferimento $D_1(z)$ e $D_2(z)$
- In questo caso si possono avere due varianti
  - realizzazione parallela con pre-moltiplicazione
  - realizzazione parallela con post-moltiplicazione
- Se il controllore viene implementato con l'unica funzione di trasferimento $D(z)$ si parla di **realizzazione diretta**

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} + b_0 e_k + b_1 e_{k-1}$$

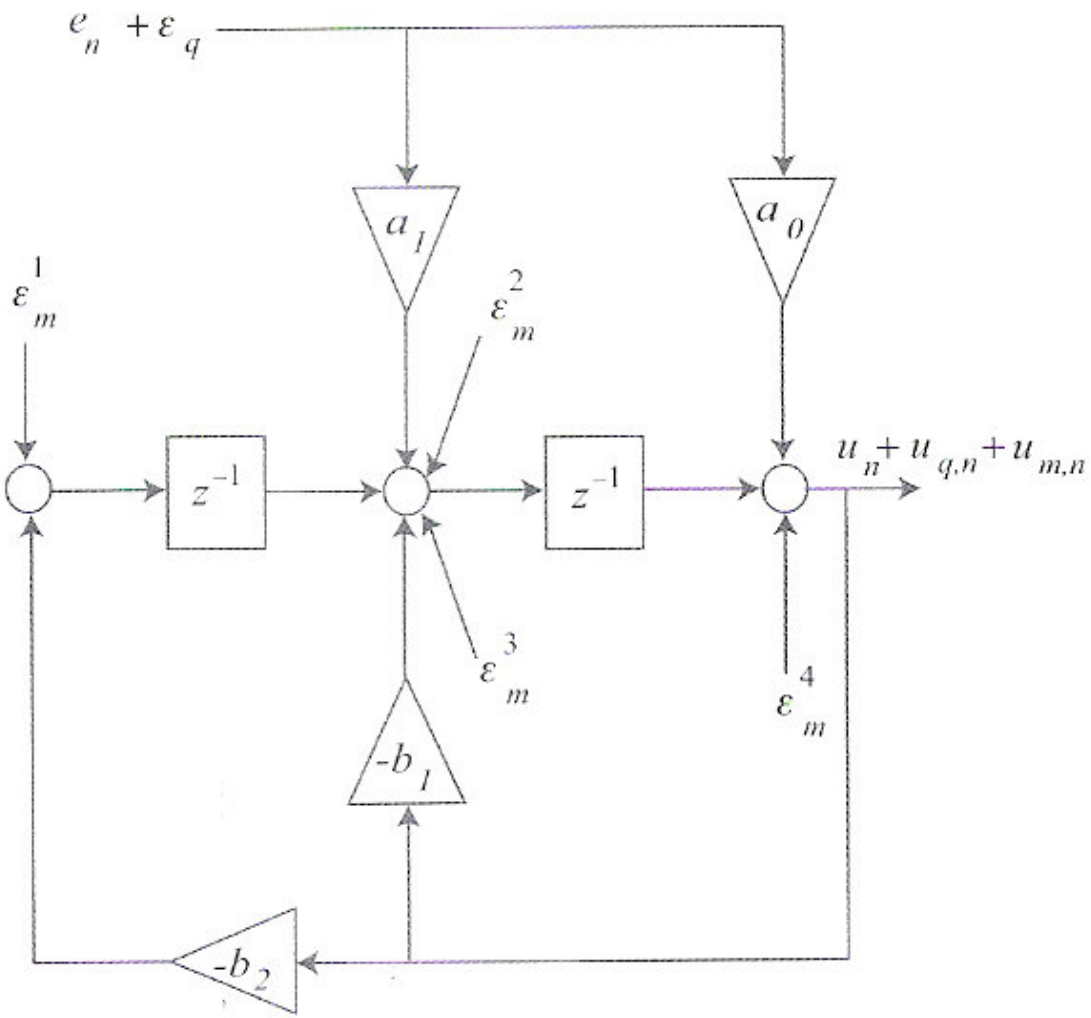
Realizzazione parallela con pre-moltiplicazione



Realizzazione parallela con post-moltiplicazione



Realizzazione diretta



Propagazione dell'errore di quantizzazione

- Sia per la realizzazione parallela che per quella diretta, si può dimostrare che

$$\bar{u}_q = \bar{\varepsilon}_q \lim_{z \to 1} D(z)$$

$$\sigma^2_{u_q} = \sigma^2_q \sum_{i=1}^{n} \lim_{z \to z_i} \left((z - z_i)\, D(z) D(z^{-1}) z^{-1} \right)$$

con z_i i-mo polo di $D(z)$

Propagazione dell'errore di quantizzazione

- Sia per la realizzazione parallela che per quella diretta, si può dimostrare che

$$\bar{u}_q = \bar{\varepsilon}_q \lim_{z \to 1} D(z)$$

$$\sigma^2_{u_q} = \sigma^2_q \sum_{i=1}^{n} \lim_{z \to z_i} \left((z - z_i)\, D(z) D(z^{-1}) z^{-1} \right)$$

  con z_i i-mo polo di $D(z)$

- **L'influenza del rumore di quantizzazione (conversione) sull'azione di controllo dipende dal controllore ma non dalla sua realizzazione software**

Propagazione degli errori di moltiplicazione - Realizzazione diretta

- Nell'esaminare l'effetto degli errori di moltiplicazione sull'uscita del controllore si assuma che tutti i rumori di moltiplicazione abbiano lo stesso valor medio $\bar{\varepsilon}_m$ e la stessa varianza σ_m^2.

Propagazione degli errori di moltiplicazione - Realizzazione diretta

- Nell'esaminare l'effetto degli errori di moltiplicazione sull'uscita del controllore si assuma che tutti i rumori di moltiplicazione abbiano lo stesso valor medio $\bar{\varepsilon}_m$ e la stessa varianza σ_m^2.
- Nel caso di realizzazione diretta del controllore del secondo ordine considerato, si può dimostrare che

$$\bar{u}_m = \bar{\varepsilon}_m \frac{4}{(1-\alpha_1)(1-\alpha_2)}$$

$$\sigma_{u_m}^2 = 4\sigma_m^2 \frac{1+\alpha_1\alpha_2}{(1-\alpha_1\alpha_2)(1-\alpha_1^2)(1-\alpha_2^2)}$$

Propagazione degli errori di moltiplicazione - Realizzazione parallela

Realizzazione parallela - Pre-moltiplicazione

$$\bar{u}_m = 2\bar{\varepsilon}_m \left(\frac{1}{1-\alpha_1} + \frac{1}{1-\alpha_2} + 2 \right)$$

$$\sigma^2_{u_m} = 2\sigma^2_m \left(\frac{1}{1-\alpha_1^2} + \frac{1}{1-\alpha_2^2} + 2 \right)$$

Propagazione degli errori di moltiplicazione - Realizzazione parallela

Realizzazione parallela - Pre-moltiplicazione

$$\bar{u}_m = 2\bar{\varepsilon}_m \left(\frac{1}{1-\alpha_1} + \frac{1}{1-\alpha_2} + 2 \right)$$

$$\sigma^2_{u_m} = 2\sigma^2_m \left(\frac{1}{1-\alpha_1^2} + \frac{1}{1-\alpha_2^2} + 2 \right)$$

Realizzazione parallela - Post-moltiplicazione

$$\bar{u}_m = 2\bar{\varepsilon}_m \left(\frac{\beta_1}{1-\alpha_1} + \frac{\beta_2}{1-\alpha_2} + 2 \right)$$

$$\sigma^2_{u_m} = 2\sigma^2_m \left(\frac{\beta_1^2}{1-\alpha_1^2} + \frac{\beta_2^2}{1-\alpha_2^2} + 2 \right)$$

Propagazione degli errori di moltiplicazione

- La generazione e la propagazione dell'errore di moltiplicazione dipende dalla particolare struttura realizzativa del controllore

Propagazione degli errori di moltiplicazione

- La generazione e la propagazione dell'errore di moltiplicazione dipende dalla particolare struttura realizzativa del controllore
- In generale strutture realizzative di tipo parallelo sono preferibili per quanto riguarda la propagazione degli errori di moltiplicazione (si ricordi che solitamente i poli del controllore α_1 , α_2 , . . . sono in modulo minore di 1)

Digital Signal Processor - 1

- Il software relativo ad un algoritmo di controllo $D(z)$ viene solitamente eseguito da un processore

Digital Signal Processor - 1

- Il software relativo ad un algoritmo di controllo $D(z)$ viene solitamente eseguito da un processore
- Un controllore digitale è un filtro digitale, vale a dire un algoritmo che elabora segnali digitali

Digital Signal Processor - 1

- Il software relativo ad un algoritmo di controllo $D(z)$ viene solitamente eseguito da un processore
- Un controllore digitale è un filtro digitale, vale a dire un algoritmo che elabora segnali digitali
- In molte applicazioni si preferisce utilizzare processori progettati in maniera specifica (sia dal punto di vista architetturale che funzionale) per l'elaborazione dei segnali digitali: tali sistemi di elaborazione sono detti Digital Signal Processor (DSP)

Digital Signal Processor - 1

- Il software relativo ad un algoritmo di controllo $D(z)$ viene solitamente eseguito da un processore
- Un controllore digitale è un filtro digitale, vale a dire un algoritmo che elabora segnali digitali
- In molte applicazioni si preferisce utilizzare processori progettati in maniera specifica (sia dal punto di vista architetturale che funzionale) per l'elaborazione dei segnali digitali: tali sistemi di elaborazione sono detti Digital Signal Processor (DSP)
- I DSP
  - sono particolarmente efficienti nel eseguire operazioni su segnali digitali
  - costituiscono un *punto di unione* tra ASIC e microprocessori general purpose

Digital Signal Processor - 2

- I DSP ottimizzano l'esecuzione di somme e moltiplicazioni su segnali digitali
- Le ALU sono dotate di moltiplicatori hardware che consentono di effettaure in una solo colpo di clock la sequenza di moltiplicazione e somma

$$w = x \cdot y + z$$

Digital Signal Processor - 2

- I DSP ottimizzano l'esecuzione di somme e moltiplicazioni su segnali digitali
- Le ALU sono dotate di moltiplicatori hardware che consentono di effettaure in una solo colpo di clock la sequenza di moltiplicazione e somma

$$w = x \cdot y + z$$

- I DSP possono essere classificati a seconda del metodo utilizzato per rappresentare in aritmetica binaria i numeri reali
  - DSP con aritmetica in virgola fissa (fixed-point)
  - DSP con aritmetica in virgola mobile (floating-point)

Notazione mantissa/esponente

- In generale un numero reale z può essere rappresentato con una notazione mantissa/esponente

$$z = x \cdot 2^y$$

  - x **numero intero** detto mantissa
  - y **numero intero** detto esponente

Notazione mantissa/esponente

- In generale un numero reale z può essere rappresentato con una notazione mantissa/esponente

$$z = x \cdot 2^y$$

  - x **numero intero** detto mantissa
  - y **numero intero** detto esponente
- Quindi z può essere rappresentato attraverso due numeri interi che possono essere codificati in binario

Aritmetica Fixed-point

- Se si assume fissato il valore dell'esponente y (definendolo in fase di progettazione), z si può rappresentare utilizzando la sola mantissa x
- In questo caso si parla di rappresentazione a virgola fissa o fixed-point, dato che il valore dell'esponente indica la posizione della virgola che separa la parte intera di z dalla sua parte frazionaria

Aritmetica Fixed-point

- Se si assume fissato il valore dell'esponente y (definendolo in fase di progettazione), z si può rappresentare utilizzando la sola mantissa x
- In questo caso si parla di rappresentazione a virgola fissa o fixed-point, dato che il valore dell'esponente indica la posizione della virgola che separa la parte intera di z dalla sua parte frazionaria
- L'utilizzo di un'aritmetica fixed-point permette di **gestire numeri reali e operazioni tra questi** mediante un **processore ad aritmetica intera** che è tipicamente molto più economico di un processore capace di gestire numeri reali

Rappresentazione $Q - N$

Solitamente la rappresentazione di un numero fixed-point viene indicata come **rappresentazione** $Q - N$, dove N indica la posizione della virgola rispetto al bit meno significativo

Esempio

In aritmetica $Q - 3$ sia ha

$$(11011011)_{Q-3} = (11011.011)_2$$
$$= \left(2^4 + 2^3 + 2 + 1 + 2^{-2} + 2^{-3}\right) = (27.375)_{10}$$

Rappresentazione Fixed-point

- La rappresentazione fixed-point richiede molta attenzione in fase di programmazione; è compito del programmatore scalare le grandezze in modo che risultino rappresentabili dal processore ad aritmetica intera
- Alcune operazioni aritmetiche possono essere effettuate solo se gli operandi sono compatibili (hanno lo stesso formato $Q - N$)
- Quando si utilizzano processori ad aritmetica fixed-point è necessario effettuare una messa in scala degli operandi prima di poter effettuare le operazioni

Aritmetica floating-point

- In aritmetica a virgola mobile o floating-point è possibile rappresentare il numero reale mediante una parola binaria in cui un certo numero di bit rappresentano la mantissa e i restanti l'esponente
- In questo caso è possibile variare la posizione della virgola che separa la parte intera di z dalla sua parte frazionaria semplicemente variando il valore dell'esponente

Aritmetica floating-point

- In aritmetica a virgola mobile o floating-point è possibile rappresentare il numero reale mediante una parola binaria in cui un certo numero di bit rappresentano la mantissa e i restanti l'esponente
- In questo caso è possibile variare la posizione della virgola che separa la parte intera di z dalla sua parte frazionaria semplicemente variando il valore dell'esponente
- La rappresentazione floating-point permette, a parità di bit utilizzati, di ottenere un range di valori rappresentati e una risoluzione molto maggiore rispetto al caso fixed point

Floating-point vs Fixed-point - 1

- parola a 8 bit, rappresentazione fixed-point $Q-4$
  - valore minimo non nullo (a meno del segno) $2^{-4} = 0.0625$
  - valore massimo 15.9375
- parola 8 bit, rappresentazione floating-point 4 bit per la mantissa e 4 per l'esponente
  - valore minimo non nullo (a meno del segno) $2^{-16} \cong 0.000015$
  - valore massimo $15 \cdot 2^{15} = 491520$

Floating-point vs Fixed-point - 2

- Sebbene la rappresentazione floating-point sia *migliore* rispetto a quella fixed-point, la complessità delle istruzioni aritmetiche e dell'hardware necessario è molto maggiore
- Questa diversità di complessità comporta un aumento notevole del costo del processore (DSP)

Floating-point vs Fixed-point - 2

- Sebbene la rappresentazione floating-point sia *migliore* rispetto a quella fixed-point, la complessità delle istruzioni aritmetiche e dell'hardware necessario è molto maggiore
- Questa diversità di complessità comporta un aumento notevole del costo del processore (DSP)
- Grazie alla loro semplicità (che comporta anche una maggiore affidabilità) e al loro basso costo i DSP ad aritmetica fixed-point sono molto utilizzati in applicazioni industriali

Messa in scala dell'algoritmo di controllo

- In fase di implementazione software di un algoritmo di controllo occorre tenere conto del DSP (processore) che eseguirà effettivamente l'algoritmo

Messa in scala dell'algoritmo di controllo

- In fase di implementazione software di un algoritmo di controllo occorre tenere conto del DSP (processore) che eseguirà effettivamente l'algoritmo
- In generale, quando si progetta un algoritmo di controllo lo si fa considerando l'impianto dal punto di vista fisico
  - si considerano sia l'errore di controllo che l'azione di controllo espressi nele grandezze fisiche proprie dell'impianto (con relative unità di misura)

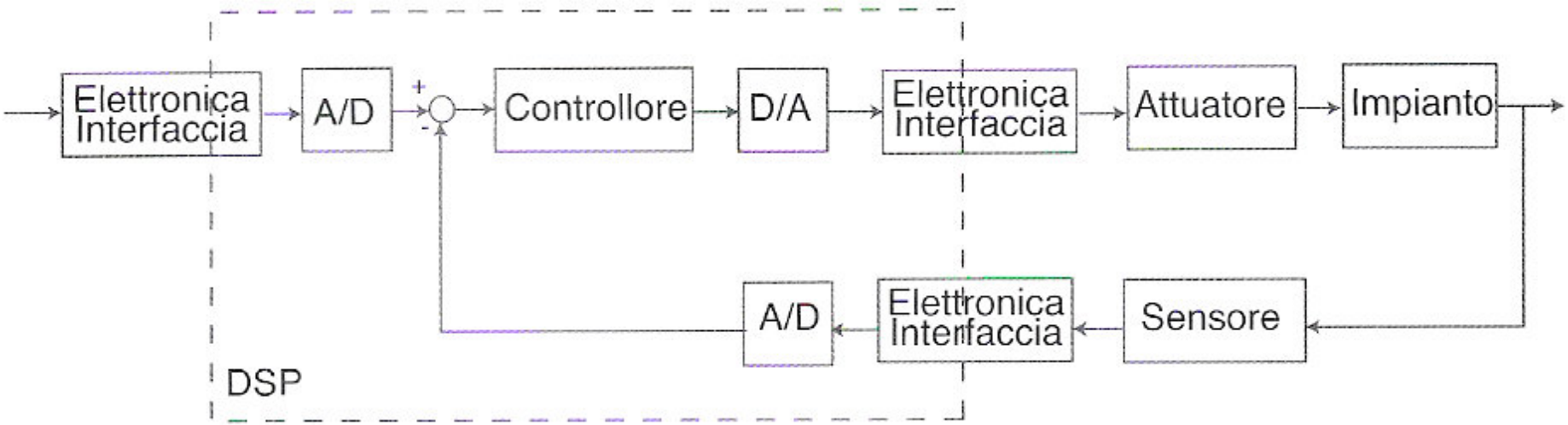
Messa in scala dell'algoritmo di controllo

- In fase di implementazione software di un algoritmo di controllo occorre tenere conto del DSP (processore) che eseguirà effettivamente l'algoritmo
- In generale, quando si progetta un algoritmo di controllo lo si fa considerando l'impianto dal punto di vista fisico
  - si considerano sia l'errore di controllo che l'azione di controllo espressi nele grandezze fisiche proprie dell'impianto (con relative unità di misura)
- **In realtà l'implementazione dell'algoritmo impone che esso si riferisca alle grandezze adimensionali derivanti dalla conversione in digitale delle grandezze fisiche**

Messa in scala dell'algoritmo di controllo

- In fase di implementazione software di un algoritmo di controllo occorre tenere conto del DSP (processore) che eseguirà effettivamente l'algoritmo
- In generale, quando si progetta un algoritmo di controllo lo si fa considerando l'impianto dal punto di vista fisico
  - si considerano sia l'errore di controllo che l'azione di controllo espressi nele grandezze fisiche proprie dell'impianto (con relative unità di misura)
- **In realtà l'implementazione dell'algoritmo impone che esso si riferisca alle grandezze adimensionali derivanti dalla conversione in digitale delle grandezze fisiche**
- Il procedimento con il quale si adatta l'algoritmo di controllo alla particolare implementazione è detto **messa in scala**

Schema a blocchi implementativo



Catene tecnologiche di acquisizione e attuazione

- L'insieme dei sistemi fisici che trasformano la variabile di controllo nell'azione di controllo (elettronica di interfaccia e condizionamento del segnale, convertitore D/A e attuatore) è detto catena tecnologica di attuazione
- L'insieme dei sistemi fisici trasformano l'uscita dell'impianto e il riferimento nell'ingresso del controllore (sensore, elettronica d'interfaccia, convertitore A/D) è detto catena tecnologica di acquisizione

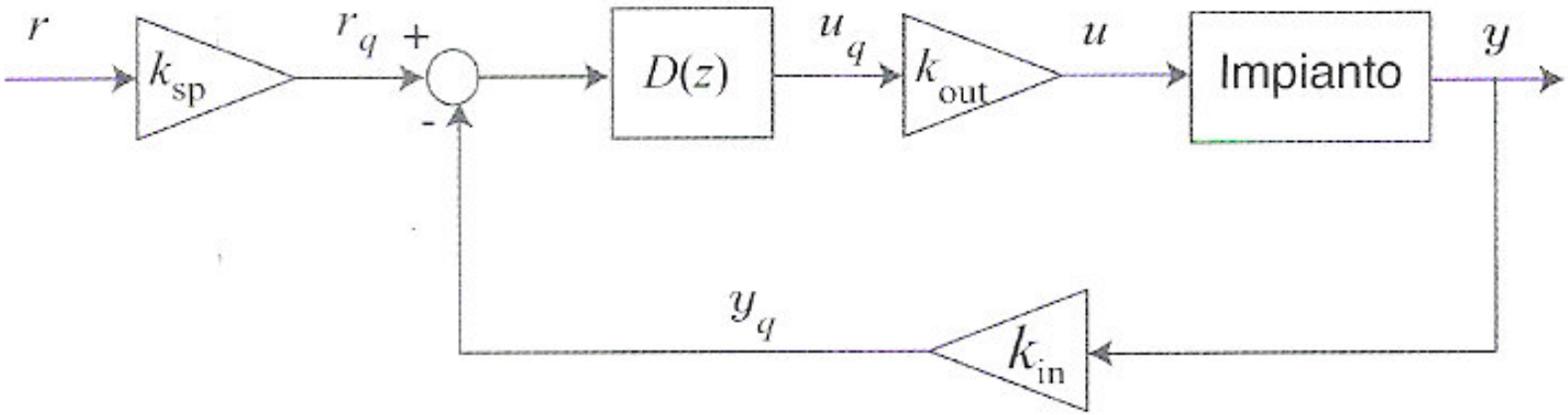
Modello statico e lineare dei sistemi d'interfaccia

- Per effettuare la messa in scala dell'algoritmo di controllo, si considerano blocchi implementativi con caratteristiche **statiche e lineari**, ovvero si suppone che tutti i sottosistemi di interfaccia siano approssimabili con i seguenti guadagni
  - k_{ad} per il convetitore A/D
  - k_{da} per il convetitore D/A
  - k_a per l'attuatore
  - k_s per il sensore
  - k_c per l'elettronica di interfacciamento

Modello statico e lineare dei sistemi d'interfaccia

- Per effettuare la messa in scala dell'algoritmo di controllo, si considerano blocchi implementativi con caratteristiche **statiche e lineari**, ovvero si suppone che tutti i sottosistemi di interfaccia siano approssimabili con i seguenti guadagni
  - k_{ad} per il convetitore A/D
  - k_{da} per il convetitore D/A
  - k_a per l'attuatore
  - k_s per il sensore
  - k_c per l'elettronica di interfacciamento
- Tale ipotesi non è riduttiva in quanto tutti i blocchi di interfaccia dovrebbero essere progettati/dimensionati in maniera da non influenzare la dinamica del sistema a ciclo chiuso nella banda d'interesse

Schema a blocchi per la messa in scala



- r_q , y_q e u_q sono le immagini di processo delle grandezze fisiche (così come rappresentate nel processore)
- Le costanti $k_{sp} = k_c k_{ad}$, $k_{in} = k_s k_c k_{ad}$ e $k_{out} = k_{da} k_c k_a$ sono note una volta progettato lo schema tecnologico

Grandezze fisiche e immagini di processo

- Le relazioni tra le grandezze fisiche r , y , u e le immagini di processo quantizzate r_q , y_q , u_q all'istante di campionamento k sono
  - $r_{q,k} = k_{sp} r_k$
  - $y_{q,k} = k_{in} y_k$
  - $u_{q,k} = \frac{1}{k_{out}} u_k$

Grandezze fisiche e immagini di processo

- Le relazioni tra le grandezze fisiche r , y , u e le immagini di processo quantizzate r_q , y_q , u_q all'istante di campionamento k sono
  - $r_{q,k} = k_{sp} r_k$
  - $y_{q,k} = k_{in} y_k$
  - $u_{q,k} = \frac{1}{k_{out}} u_k$
- Il progetto dell'algoritmo di controllo $D(z)$ viene solitamente fatto utlizzando le grandezze fisiche, quindi trascurando la catena tecnologica
- L'implementazione dell'algoritmo, utilizzando le immagini di processo al posto delle grandezze fisiche, porterebbe a risultati errati

Grandezze fisiche e immagini di processo

- Le relazioni tra le grandezze fisiche r , y , u e le immagini di processo quantizzate r_q , y_q , u_q all'istante di campionamento k sono
  - $r_{q,k} = k_{sp} r_k$
  - $y_{q,k} = k_{in} y_k$
  - $u_{q,k} = \frac{1}{k_{out}} u_k$
- Il progetto dell'algoritmo di controllo $D(z)$ viene solitamente fatto utlizzando le grandezze fisiche, quindi trascurando la catena tecnologica
- L'implementazione dell'algoritmo, utilizzando le immagini di processo al posto delle grandezze fisiche, porterebbe a risultati errati
- Occorre compensare gli effetti della catena tecnologica scalando l'algoritmo di controllo

Messa in scala

Dato l'algoritmo di controllo

$$u_k = \sum_{i=1}^{n} a_i u_{k-i} + \sum_{j=0}^{m} b_j e_{k-j}$$

esistono due possibili soluzioni al problema della scalatura dell'algoritmo di controllo

1. utilizzare l'algoritmo di controllo progettato e adattare le variabili in gioco (**messa in scala tecnologica delle variabili**)
2. utilizzare le variabili acquisite e adattare l'algoritmo di controllo (**messa in scala tecnologica delle equazioni**)

Messa in scala tecnologica delle variabili - 1

1. Messa in scala dei valori d'ingresso, calcolando i valori *veri* di tali grandezze

$$r_k = \frac{1}{k_{sp}} r_{q,k}$$

$$y_k = \frac{1}{k_{in}} y_{q,k}$$

Messa in scala tecnologica delle variabili - 1

1. Messa in scala dei valori d'ingresso, calcolando i valori *veri* di tali grandezze

$$r_k = \frac{1}{k_{sp}} r_{q,k}$$

$$y_k = \frac{1}{k_{in}} y_{q,k}$$

2. Esecuzione dell'algoritmo di controllo con i parametri originali

$$e_k = r_k - y_k$$

$$u_k = \sum_{i=1}^{n} a_i u_{k-i} + \sum_{j=0}^{m} b_j e_{k-j}$$

Messa in scala tecnologica delle variabili - 1

1. Messa in scala dei valori d'ingresso, calcolando i valori *veri* di tali grandezze

$$r_k = \frac{1}{k_{sp}} r_{q,k}$$

$$y_k = \frac{1}{k_{in}} y_{q,k}$$

2. Esecuzione dell'algoritmo di controllo con i parametri originali

$$e_k = r_k - y_k$$

$$u_k = \sum_{i=1}^{n} a_i u_{k-i} + \sum_{j=0}^{m} b_j e_{k-j}$$

3. Scalatura dell'uscita del controllore

$$u_{q,k} = \frac{1}{k_{out}} u_k$$

Messa in scala tecnologica delle variabili - 2

Il processo di messa in scala delle variabili

- richiede la **scalatura dei segnali acquisiti ad ogni istante di campionamento**

Messa in scala tecnologica delle variabili - 2

Il processo di messa in scala delle variabili

- richiede la **scalatura dei segnali acquisiti ad ogni istante di campionamento**
- richiede operazioni da effettuare online che potrebbero essere computazionalmente gravose

Messa in scala tecnologica delle equazioni- 1

1. Se $k_{in} \neq k_{sp}$ si mettono in scala relativa le due variabili d'ingresso

$$k_{in} = k' k_{sp} \Rightarrow e_k = r_k - y_k = \frac{1}{k_{in}} \left(k' r_{q,k} - y_{q,k} \right)$$

da cui

$$e_{q,k} = k' r_{q,k} - y_{q,k}$$

Se $k_{in} = k_{sp}$ allora $k' = 1$

Messa in scala tecnologica delle equazioni- 1

1. Se $k_{in} \neq k_{sp}$ si mettono in scala relativa le due variabili d'ingresso

$$k_{in} = k' k_{sp} \Rightarrow e_k = r_k - y_k = \frac{1}{k_{in}} \left(k' r_{q,k} - y_{q,k} \right)$$

da cui

$$e_{q,k} = k' r_{q,k} - y_{q,k}$$

Se $k_{in} = k_{sp}$ allora $k' = 1$

2. Esecuzione dell'algoritmo di controllo *scalato*

$$k_{out} u_{q,k} = \sum_{i=1}^{n} a_i k_{out} u_{q,k-i} + \sum_{j=0}^{m} b_j \frac{1}{k_{in}} e_{q,k-j}$$

da cui

$$u_{q,k} = \sum_{i=1}^{n} a_i u_{q,k-i} + \sum_{j=0}^{m} b_j \frac{1}{k_{out} k_{in}} e_{q,k-j}$$

Messa in scala tecnologica delle equazioni - 2

La messa in scala delle equazioni

- è un procedimento che può essere effettuato in fase di progettazione (offline)

Messa in scala tecnologica delle equazioni - 2

La messa in scala delle equazioni

- è un procedimento che può essere effettuato in fase di progettazione (offline)
- **non rappresenta un appesantimento dell'onere computazionale del processore, pertanto è la soluzione da preferire per la messa in scala**

Messa in scala aritmetica

- Quando si utilizzano DSP con aritmetica fixed-point le immagini di processo sono grandezze intere
- I parametri dell'algoritmo, invece, sono solitamente grandezze reali

Messa in scala aritmetica

- Quando si utilizzano DSP con aritmetica fixed-point le immagini di processo sono grandezze intere
- I parametri dell'algoritmo, invece, sono solitamente grandezze reali
- Occorre scalare l'algoritmo di controllo affinché le immagini di processo e i parametri dell'algoritmo di controllo siano rappresentabili all'interno del DSP
- tale procedimento prende il nome di **messa in scala aritmetica**

Messa in scala aritmetica

- Quando si utilizzano DSP con aritmetica fixed-point le immagini di processo sono grandezze intere
- I parametri dell'algoritmo, invece, sono solitamente grandezze reali
- Occorre scalare l'algoritmo di controllo affinché le immagini di processo e i parametri dell'algoritmo di controllo siano rappresentabili all'interno del DSP
- tale procedimento prende il nome di **messa in scala aritmetica**
- Per processori ad aritmetica intera, la messa in scala aritmetica si combina con quella tecnologica

Messa in scala aritmetica - Esempio

- Supponiamo di voler rappresentare il seguente regolatore con un'aritmetica intera a 4 cifre con segno (ovvero un'aritmetica che permette di rappresentare i numeri da -9999 a +9999)

$$u_{q,k} = 0.93u_{q,k-1} + 0.132e_{q,k}$$

Messa in scala aritmetica - Esempio

- Supponiamo di voler rappresentare il seguente regolatore con un'aritmetica intera a 4 cifre con segno (ovvero un'aritmetica che permette di rappresentare i numeri da -9999 a +9999)

$$u_{q,k} = 0.93u_{q,k-1} + 0.132e_{q,k}$$

- Si scala l'algoritmo di controllo premoltiplicando per la massima potenza di 10 che rende ancora rappresentabile il più grande coefficiente

$$u^s_{q,k} = 10^4 \left(0.93u_{q,k-1} + 0.132e_{q,k}\right) = 9300u_{q,k-1} + 1320e_{q,k}$$

Messa in scala aritmetica - Esempio

- Supponiamo di voler rappresentare il seguente regolatore con un'aritmetica intera a 4 cifre con segno (ovvero un'aritmetica che permette di rappresentare i numeri da -9999 a +9999)

$$u_{q,k} = 0.93u_{q,k-1} + 0.132e_{q,k}$$

- Si scala l'algoritmo di controllo premoltiplicando per la massima potenza di 10 che rende ancora rappresentabile il più grande coefficiente

$$u^s_{q,k} = 10^4 \left(0.93u_{q,k-1} + 0.132e_{q,k}\right) = 9300u_{q,k-1} + 1320e_{q,k}$$

- Il valore dell'azione di controllo $u^s_{q,k}$ va riscalato

$$u_{q,k} = 10^{-4}u^s_{q,k}$$

Messa in scala tecnologica e aritmetica delle equazioni - 1

1. Messa in scala delle equazioni, ottenendo l'algoritmo di controllo

$$u_{q,k} = \sum_{i=1}^{n} a_i u_{q,k-i} + \sum_{j=0}^{m} b'_j e_{q,k-j}\,, \quad \text{con } b'_j = b_j \frac{1}{k_{in} k_{out}}$$

Messa in scala tecnologica e aritmetica delle equazioni - 1

1. Messa in scala delle equazioni, ottenendo l'algoritmo di controllo

$$u_{q,k} = \sum_{i=1}^{n} a_i u_{q,k-i} + \sum_{j=0}^{m} b'_j e_{q,k-j}\,, \quad \text{con } b'_j = b_j \frac{1}{k_{in} k_{out}}$$

2. Determinazione della massima potenza di 2 (2^p) che renda ancora rappresentabile, con il numero di bit disponibili (tenendo conto del bit di segno), il coefficiente più grande tra a_i e b'_j

Messa in scala tecnologica e aritmetica delle equazioni - 1

1. Messa in scala delle equazioni, ottenendo l'algoritmo di controllo

$$u_{q,k} = \sum_{i=1}^{n} a_i u_{q,k-i} + \sum_{j=0}^{m} b'_j e_{q,k-j}\,, \quad \text{con } b'_j = b_j \frac{1}{k_{in} k_{out}}$$

2. Determinazione della massima potenza di 2 (2^p) che renda ancora rappresentabile, con il numero di bit disponibili (tenendo conto del bit di segno), il coefficiente più grande tra a_i e b'_j
3. Riscalare l'algoritmo di controllo

$$u^s_{q,k} = \sum_{i=1}^{n} 2^p a_i u_{q,k-i} + \sum_{j=0}^{m} 2^p b'_j e_{q,k-j}$$

Messa in scala tecnologica e aritmetica delle equazioni - 2

4. Riscalatura dell'azione di controllo

$$u_{q,k} = 2^{-p} u^s_{q,k}$$

Si noti che tale operazione non rappresenta un aumento del carico computazionle, perché può essere realizzata con un operazioni di shift sulla parola binaria

Messa in scala tecnologica e aritmetica delle equazioni - 2

4. Riscalatura dell'azione di controllo

$$u_{q,k} = 2^{-p} u^s_{q,k}$$

Si noti che tale operazione non rappresenta un aumento del carico computazionle, perché può essere realizzata con un operazioni di shift sulla parola binaria

5. Limitare il valore dell'azione di controllo calcolato al massimo valore rappresentabile con il numero di bit del convertitore D/A

Messa in scala tecnologica e aritmetica delle equazioni - 2

4. Riscalatura dell'azione di controllo

$$u_{q,k} = 2^{-p} u^s_{q,k}$$

Si noti che tale operazione non rappresenta un aumento del carico computazionle, perché può essere realizzata con un operazioni di shift sulla parola binaria

5. Limitare il valore dell'azione di controllo calcolato al massimo valore rappresentabile con il numero di bit del convertitore D/A
   In generale, per motivi di costo del convertitore D/A, DSP e convertitore utilizzano un numero diverso di bit

Messa in scala tecnologica e aritmetica delle equazioni - 2

4. Riscalatura dell'azione di controllo

$$u_{q,k} = 2^{-p} u^s_{q,k}$$

Si noti che tale operazione non rappresenta un aumento del carico computazionle, perché può essere realizzata con un operazioni di shift sulla parola binaria

5. Limitare il valore dell'azione di controllo calcolato al massimo valore rappresentabile con il numero di bit del convertitore D/A
   In generale, per motivi di costo del convertitore D/A, DSP e convertitore utilizzano un numero diverso di bit
   Questo passo è importante perché cercare di convertire in analogico un numero maggiore di quello massimo rappresentabile comporta l'attuazione di un valore completamente sbagliato

Messa in scala tecnologica e aritmetica delle equazioni - 2

4. Riscalatura dell'azione di controllo

$$u_{q,k} = 2^{-p} u^s_{q,k}$$

Si noti che tale operazione non rappresenta un aumento del carico computazionle, perché può essere realizzata con un operazioni di shift sulla parola binaria

5. Limitare il valore dell'azione di controllo calcolato al massimo valore rappresentabile con il numero di bit del convertitore D/A
   In generale, per motivi di costo del convertitore D/A, DSP e convertitore utilizzano un numero diverso di bit
   Questo passo è importante perché cercare di convertire in analogico un numero maggiore di quello massimo rappresentabile comporta l'attuazione di un valore completamente sbagliato

Esempio

- DSP a 16 bit senza segno $\rightarrow u_{q,k} = 4142$ (0001 0000 0010 1110)
- D/A a 12 bit senza segno $\rightarrow u_{D/A,,k} = 46$ (0000 0010 1110)

Problemi per la messa in scala

- Il processo di messa in scala può avere dei problemi nel caso in cui l'algoritmo di controllo presenti coefficienti con ordini di grandezza molto differenti tra loro
- In questo caso potrebbe accadere che i coefficienti con valore più piccolo risultino non rappresentabili (underflow)

Esempio

- Si supponga di voler scalare il seguente algoritmo

$$u_k = 0.972u_{k-1} - 0.0432e_k + 0.00322e_{k-1} - 0.000531e_{k-2}$$

utilizzando un'aritmetica intera a tre cifre con segno

Esempio

- Si supponga di voler scalare il seguente algoritmo

$$u_k = 0.972u_{k-1} - 0.0432e_k + 0.00322e_{k-1} - 0.000531e_{k-2}$$

utilizzando un'aritmetica intera a tre cifre con segno

- Scalando per un unico coefficiente pari a 10^3 si ottiene

$$u_k = 972u_{k-1} - 43e_k + 3e_{k-1} - 0e_{k-2}$$

l'ultimo coefficiente non viene rappresentato!

Messa in scala differenziale

- Per risolvere il problema dovuto all'underflow è possibile effettuare la **messa in scala aritmetica differenziale**

Messa in scala differenziale

- Per risolvere il problema dovuto all'underflow è possibile effettuare la **messa in scala aritmetica differenziale**
- Nel caso dell'esempio considerato si ha

$$h_1^s = 10^3(0.972)u_{k-1} = 972u_{k-1}$$
$$h_2^s = 10^4(-0.0432)e_k = -432e_k$$
$$h_3^s = 10^5(0.00322)e_{k-1} = 322e_{k-1}$$
$$h_4^s = 10^6(-0.000531)e_{k-2} = -531e_{k-2}$$

L'azione di controllo viene poi ricostruita con una riscalatura differenziale

$$w_3 = h_3^s + 10^-1 h_4^s$$
$$w_2 = h_2^s + 10^-1 w_3$$
$$w_1 = h_1^s + 10^-1 w_2$$
$$u_k = 10^-3 w_3$$

Bibliografia


C. Bonivento, L. Gentili, A. Paoli,
Sistemi di Automazione Industriale
McGraw-Hill, 2011