

Identificazione dei Parametri Caratteristici di un Plasma Circolare Tramite Rete Neuronale

Introduzione

Si è utilizzata una rete neuronale multipiano con il fine di determinare i parametri caratteristici di un plasma circolare partendo dalla conoscenza del flusso magnetico in alcuni punti definiti.

Si è utilizzata la formula di Grad-Shafranov, risolta nel caso di plasma circolare e rapporto $r/R \ll 1$ [1], per generare una banca dati in cui, per ogni valore delle tre variabili indipendenti (R,Z e Λ), si ottengono i valori dei flussi nei test point considerati.

Ora, con l'utilizzazione di una rete neuronale, si tenta di risolvere il problema inverso che consiste appunto nell'identificare le variabili indipendenti conoscendo il valore del flusso nei punti prima definiti.

Ciò è possibile attraverso 5 passi fondamentali:

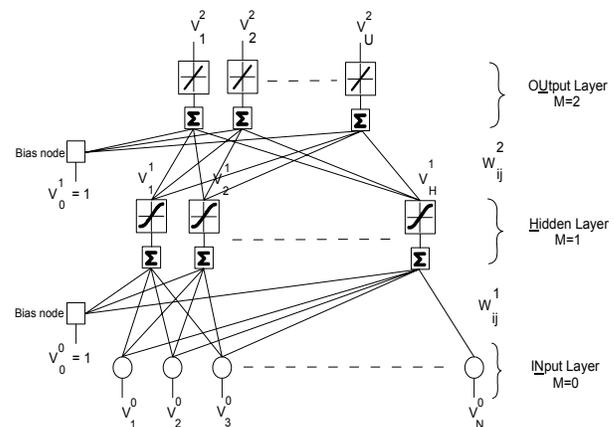
- scelta dell'architettura della rete neuronale;
- generazione dei pesi iniziali da attribuire ad ogni singolo ramo;
- generazione di una banca dati che costituisce la base sia della fase di training che di testing della rete stessa;
- fase di training, che consiste nel trovare i pesi da attribuire ai singoli rami e che risolvono il problema nel modo più soddisfacente;
- fase di testing, che consiste nel sottoporre la rete, ormai assestata, a dei nuovi casi e registrarne la risposta.

In ultimo vi è l'analisi dei risultati ottenuti nell'ultima fase con la visione delle distribuzioni generate ed il calcolo degli errori.

Architettura della rete

Si è presa in considerazione [2,3] la soluzione di tre Layers, in cui si hanno 10 neuroni per il Layer di Input, 8 per quello Hidden e 3 (R,Z e Λ) per il layer di Output.

Lo schema della rete è riportato nella Fig.1 in cui compaiono i layers con i neuroni, le connessioni aggiuntive che lavorano come bias, le funzioni di attivazione sigmoidali dei neuroni Hidden e le funzioni di attivazione lineari di Output.



Definizioni

V_i^M = Output della i-esima unità nel piano M (per $M=0$ V_i^0 = Input)

W_{ik}^M = peso della connessione da V_k^{M-1} a V_i^M

N = Numero di Input

H = Numero di unità Hidden

U = Numero di Outputs

Fig. 1

Generazione dei pesi iniziali

I pesi iniziali sono variabili tra -0.5 e 0.5 e sono stati generati in modo random secondo una distribuzione uniforme.

Generazione della banca dati

Come detto si è utilizzata la funzione:

$$\psi = \frac{\mu_0 \cdot I}{2\pi} R \left(\ln \frac{8R}{r} - 2 \right) - \frac{\mu_0 \cdot I}{4\pi} \left(\ln \frac{r}{a} + \left(\Lambda + \frac{1}{2} \right) \left(1 - \frac{a^2}{r^2} \right) \right) \cos \omega$$

dove il significato dei simboli è visibile dalla Fig. 2, essendo Λ il fattore di asimmetria.

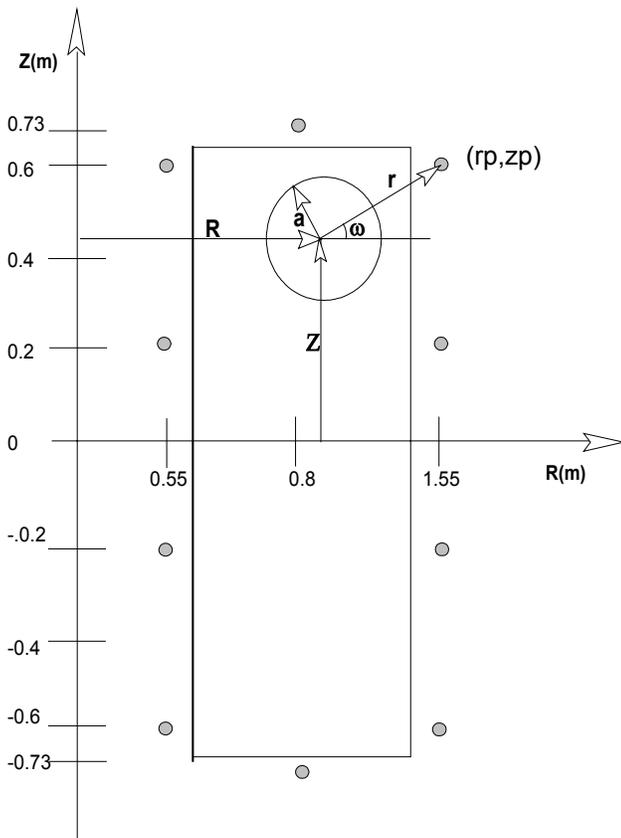


Fig.2

Le variabili indipendenti R,Z e Λ sono scelte in modo random nei ranges:

$$R \in [0.80, 0.90]$$

$$Z \in [-0.48, 0.48]$$

$$\Lambda \in [0.50, 4.00]$$

ed il raggio del plasma $a = 0.24 - |R - 0.8|$.

Sono stati generati 1300 serie di dati: ogni serie è composta dai 3 valori delle variabili R,Z, Λ e dai 10 valori conseguenti del flusso nei punti scelti come test point.

La banca dati verrà utilizzata sia in fase di apprendimento che in fase di verifica. Si consideri, però, che i valori di R, Z e Λ realmente scritti nella banca dati e su cui si effettuano tutte le operazioni di learning e di test sono valori normalizzati (Val. medio = 0; Val. estremi = ± 1).

Training della rete

L'obiettivo del training è quello di trovare un set di pesi che, per l'architettura fissata, porti a minimizzare le differenze tra le uscite che si vorrebbe avere dalla rete e quelle che, invece, la rete dà. Come misura della differenza tra le uscite desiderate e quelle ottenute, si prende l'errore quadratico definito come

$$ER2 = \frac{1}{2} \sum (\text{Val. Desiderato} - \text{Val. Ottenuto})^2$$

essendo la sommatoria estesa a tutti i P casi con cui si effettua il training: nel presente caso si utilizzano 800 serie di dati o pattern. La procedura di seguito descritta [4] utilizza l'algoritmo della "back propagation" e fornisce una metodologia per variare i pesi, durante la fase di learning, al fine di minimizzare la precedente usuale misura dell'errore.

Utilizzando la simbologia riportata in Fig. 1 si ha:

$$\text{Vettore dati di input (livello 0): } \overline{V}_k^0 \quad [k=0, \dots, N] \Rightarrow \begin{matrix} V_0^0 = -1 \\ V_1^0 \\ \vdots \\ V_N^0 \end{matrix}$$

$$\begin{aligned} \text{Vettore output hidden (livello 1): } \bar{\mathbf{V}}_i^1 \quad [i=0, \dots, H] &\Rightarrow \begin{matrix} \mathbf{V}_0^1 = -1 \\ \mathbf{V}_1^1 \\ \vdots \\ \mathbf{V}_H^1 \end{matrix} \\ \text{Vettore dati di output (livello 2): } \bar{\mathbf{V}}_j^2 \quad [j=1, \dots, U] &\Rightarrow \begin{matrix} \mathbf{V}_1^2 \\ \vdots \\ \mathbf{V}_U^2 \end{matrix} \end{aligned}$$

da cui:

$$V_i^1 = g\left(\sum_{k=1}^N W_{i,k}^1 * V_k^0 - \theta_i^1\right) \quad [i=1, \dots, H] \quad (1)$$

$$V_j^2 = g\left(\sum_{i=1}^H W_{j,i}^2 * V_i^1 - \theta_j^2\right) \quad [j=1, \dots, U] \quad (2)$$

sostituendo la (1) nella (2) si ha :

$$V_j^2 = g\left(\sum_{i=1}^H \sum_{k=1}^N W_{j,i}^2 * g\left(\sum_{k=1}^N W_{i,k}^1 * V_k^0 - \theta_i^1\right) - \theta_j^2\right) \quad [j=1, \dots, U] \quad (3)$$

fissando nella (1) $V_0^1 = -1$ e $W_{i,0}^1 = \theta_i^1$

e nella (2) $V_0^2 = -1$ e $W_{j,0}^2 = \theta_j^2$

si ha nella (3) :

$$V_j^2 = g\left(\sum_{i=1}^H \sum_{k=1}^N W_{j,i}^2 * g\left(\sum_{k=1}^N W_{i,k}^1 * V_k^0\right) - W_{j,0}^2 * V_0^1\right) \quad [j=1, \dots, U]$$

ed in ultimo :

$$\left\{ \begin{array}{l} V_j^2 = g\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right) \quad [j=1, \dots, U] \\ \text{con la condizione che per } i=0 \text{ sia } g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right) = -1 \text{ e } W_{j,i}^2 = \theta_j^2 \\ \text{e ricordando che per } k=0 \text{ risulta } V_0^1 = -1 \text{ e } W_{i,0}^1 = \theta_i^1 \end{array} \right.$$

Per la definizione delle matrici dei pesi si chiamano:

matrice pesi livello 1 (tra il piano 1 ed il piano 0): $|\mathbf{W}_{i,k}^1|$

matrice pesi livello 2 (tra il piano 2 ed il piano 1): $|\mathbf{W}_{j,i}^2|$

$$|\mathbf{W}_{i,k}^1| = \begin{matrix} \begin{matrix} W_{1,0}^1 (= \theta_1^1) & W_{1,1}^1 & W_{1,2}^1 & \dots & W_{1,N}^1 \\ W_{2,0}^1 (= \theta_2^1) & W_{2,1}^1 & \dots & \dots & W_{2,N}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{H,0}^1 (= \theta_H^1) & W_{H,1}^1 & \dots & \dots & W_{H,N}^1 \end{matrix} & \begin{matrix} [i=1, \dots, H] \\ [k=0, \dots, N] \end{matrix} \end{matrix}$$

$$|\mathbf{W}_{j,i}^2| = \begin{matrix} \begin{matrix} W_{1,0}^2 (= \theta_1^2) & W_{1,1}^2 & W_{1,2}^2 & \dots & W_{1,H}^2 \\ W_{2,0}^2 (= \theta_2^2) & W_{2,1}^2 & \dots & \dots & W_{2,H}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{U,0}^2 (= \theta_U^2) & W_{U,1}^2 & \dots & \dots & W_{U,H}^2 \end{matrix} & \begin{matrix} [j=1, \dots, U] \\ [i=0, \dots, H] \end{matrix} \end{matrix}$$

Avendo chiamato l'errore:

$$E = \frac{1}{2} \sum_j \sum_p [U_j(p) - V_j^2(p)]^2$$

si ha :

$$E = \frac{1}{2} \sum_j \sum_p \left[U_j(p) - g\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right)(p) \right]^2$$

derivando l'errore rispetto ai termini $W_{j,i}^2$ si ha :

$$\frac{\partial E}{\partial W_{j,i}^2} = - \sum_p \left[\left[U_j(p) - g\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right)(p) \right] * \left[g'\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right) * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right) \right] \right]$$

ossia :

$$\frac{\partial E}{\partial W_{j,i}^2} = - \sum_p [U_j(p) - V_j^2(p)] * g'(V_j^2(p)) * V_i^1(p)$$

chiamando $\delta_j^p = [U_j(p) - V_j^2(p)] * g'(V_j^2(p))$

si ha :

$$\frac{\partial E}{\partial W_{j,i}^2} = - \sum_p (\delta_j^p * V_i^1(p))$$

ed ancora :

$$\frac{\partial E}{\partial W_{i,k}^1} = - \sum_j \sum_p (\delta_j^p * W_{j,i}^2 * g'(V_i^1(p)) * V_k^0)$$

derivando invece rispetto ai termini $W_{i,k}^1$ si ha :

$$\frac{\partial E}{\partial W_{i,k}^1} = - \sum_j \sum_p \left[\left[U_j(p) - g\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right)(p) \right] * \left[g'\left(\sum_{i=0}^H \sum_{k=0}^N W_{j,i}^2 * g\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right)\right) * W_{j,i}^2 * g'\left(\sum_{k=0}^N W_{i,k}^1 * V_k^0\right) * V_k^0 \right] \right]$$

ossia :

$$\frac{\partial E}{\partial W_{i,k}^1} = - \sum_j [U_j(p) - V_j^2(p)] * g'(V_j^2(p)) * W_{j,i}^2 * g'(V_i^1(p)) * V_k^0$$

chiamando ora $\delta_i^p = g'(V_i^1(p)) * \sum_j (\delta_j^p * W_{j,i}^2)$

si ha :

$$\frac{\partial E}{\partial W_{i,k}^1} = - \sum_p (\delta_i^p * V_k^0)$$

Ora si può pensare di trovare il minimo dell'errore E attribuendo all'incremento dei pesi W un valore proporzionale alla derivata parziale dell'errore stesso, tramite un fattore usualmente chiamato "Learning rate" η .

Per i pesi livello 2 (compreso tra i piani 2 ed 1) si ha :

$$\Delta W_{j,i}^2 = -\eta * \frac{\partial E}{\partial W_{j,i}^2} = \eta * \sum_p (\delta_j^p * V_i^1(p)) \text{ con } \delta_j^p = [U_j(p) - V_j^2(p)] * g'(V_j^2(p))$$

i nuovi pesi saranno $W_{j,i}^{2,new} = W_{j,i}^{2,old} + \Delta W_{j,i}^2$

similmente per i pesi livello 1 (tra i piani 1 e 0) si ha :

$$\Delta W_{i,k}^1 = -\eta * \frac{\partial E}{\partial W_{i,k}^1} = \eta * \sum_p (\delta_i^p * V_k^0(p)) \text{ con } \delta_i^p = g'(V_i^1(p)) * \sum_j (\delta_j^p * W_{j,i}^2)$$

i nuovi pesi saranno $W_{i,k}^{1,new} = W_{i,k}^{1,old} + \Delta W_{i,k}^1$

Nel presente caso la funzione di attivazione "g" è definita come:

$g(x) = \tanh(x)$ la cui derivata è $g'(x) = 1 - g(x)^2$ mentre la funzione di attivazione in uscita è lineare $g(x) = \text{cost} * x$.

Test della rete ed analisi dei risultati

Ora è possibile sottoporre alla rete i rimanenti patterns della banca dati al fine di valutare il comportamento della rete stessa.

Nel nostro caso si sono utilizzati gli altri 500 casi e si è generato un file in cui, a fianco di ogni valore desiderato, vi è il valore ottenuto.

I parametri significativi dell'esempio sono riportati nella Fig. 3, mentre l'analisi delle distribuzioni dei dati e delle loro differenze è mostrato nelle figure 4 e 5: la figura 4 mostra le posizioni reciproche dei valori attesi e di quelli ottenuti, mentre la figura 5 mostra gli istogrammi relativi alle singole popolazioni dei valori e degli errori.

Dalle Fig. 4 e 5 si può vedere che il sistema segue bene nella media i valori attesi (gli errori percentuali sulla media, riferiti al valore massimo, sono per R: E%=0; per Z:

E%=8% e per Λ : E%=7%), ma produce una dispersione grande intorno ad essi.

Le popolazioni attese ed ottenute sono molto simili sia per valore medio che per deviazione standard ma gli errori hanno una deviazione abbastanza grande.

L'errore **rms** assoluto, definito come:

$$rms = \left[\frac{1}{P} \sum_{i=1}^P e_i^2 \right]^{\frac{1}{2}}$$

risulta, su tutte le tre variabili, uguale a 0.283 essendo su

$$R : rms = 0.140$$

$$Z : rms = 0.169$$

$$\Lambda : rms = 0.179$$

Quanto mostrato è stato ottenuto con un programma che racchiude tutti i passi descritti: tale programma permette di variare l'architettura dell'intera rete (ad esclusione, al momento, del numero di Layers), la numerosità della banca dati, la numerosità dei patterns per il training e quelli di test, le costanti che compaiono nella procedura di back-propagation ed è quasi completata la possibilità di scelta tra le varie tipologie delle funzioni sigmoidali di attivazione..

Un capitolo importante è come identificare la costante Learning rate e la cost. di linearità che compaiono nell'algoritmo di calcolo, insieme ai pesi iniziali attribuiti ai vari rami. In questo caso tali costanti ed il set di pesi iniziali sono trovati per tentativi senza una adeguata procedura di ottimizzazione. .

~~Il listing del programma con cui sono stati ottenuti i risultati visti è riportato in allegato 1.~~

Attività per il futuro

Nello svolgimento dell'attività si sono poste una quantità di problemi su cui lavorare: analizzare l'importanza dei pesi iniziali delle connessioni; verificare il comportamento della rete assumendo costanti multiple e/o procedure di adattamento automatico durante la fase di Learning [4]; confrontare i risultati con quelli ottenibili con diverse funzioni sigmoidali di attivazione; passare a diverse architetture con diversi Hidden neurons e/o diversi input e/o più Hidden Layers; pensare a reti che varino automaticamente la propria architettura ed assestino la quantità di dati di apprendimento in funzione del compito e dei risultati che, via via, ottengono.

Conclusioni

Quello mostrato è soltanto un caso che prende lo spunto dalle presentazioni in mio possesso, al fine di entrare nell'argomento e produrre un minimo di risultati su cui discutere.

Considerando questo lavoro un primo approccio al problema (semplicità delle funzioni di attivazione, mancanza di una procedura adeguata di ottimizzazione dei parametri, basso numero di Input, momento nullo), sembra che i risultati ottenuti siano molto incoraggianti: sarebbe opportuno, a mio parere, seguire ad approfondire il problema almeno nelle linee di attività a più breve termine.

RIFERIMENTI

- [1] J.Wesson "Tokamaks", Clarendon Press, Oxford, 1987
- [2] E.Coccoresse, R.Martone and C.Morabito "A Neural Network Approach for the Solution of Electric and Magnetic Inverse Problems", IEE Transactions on Magnetics, Vol. 30, n°5, Sept. 1994
- [3] E.Coccoresse, R.Martone and C.Morabito "On Line Plasma Shape Identification in a Tokamak Reactor via Neural Network", 5° Workshop Italiano 12-15 Maggio 1992
- [4] J.Hertz, A.Krogh, R.G.Palmer "Introduction to the Theory of Neural Computation", Santa Fe Institute, Addison-Wesley Publishing Company, 1991
- [5] R.Albanese, E.Coccoresse et al. "Identification of Plasma Equilibria in Iter from Magnetic Measurements via Functional Parametrization and Neural Networks", Fusion Technology, 30 June 1995

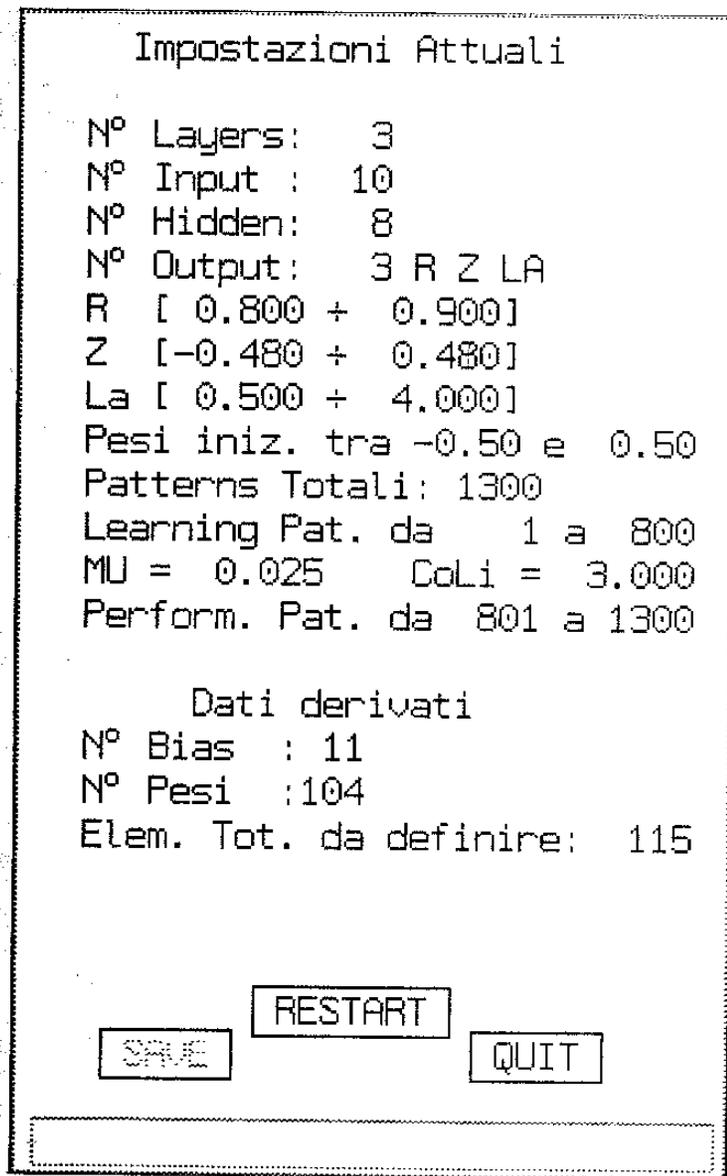


Fig. 3

Confronto Valori ATTESI - Valori OTTENUTI

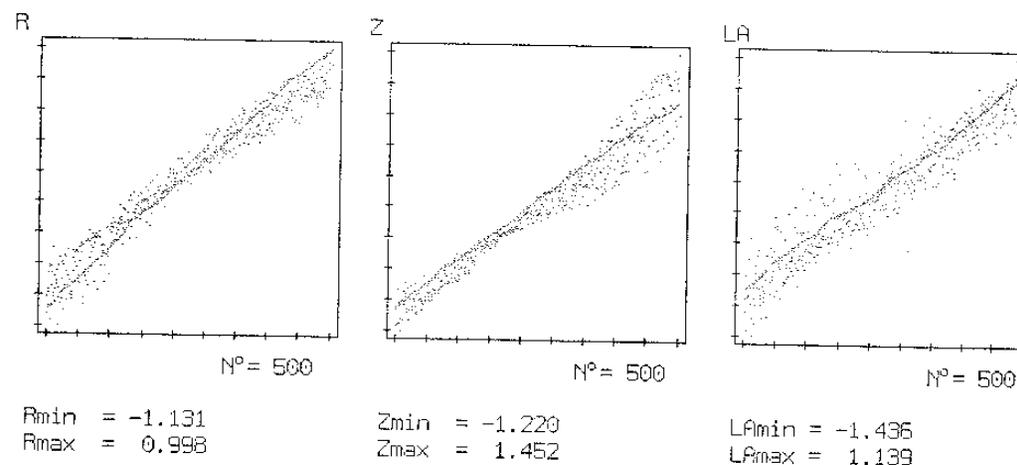
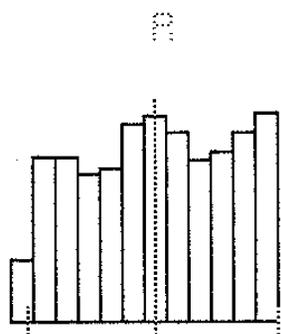
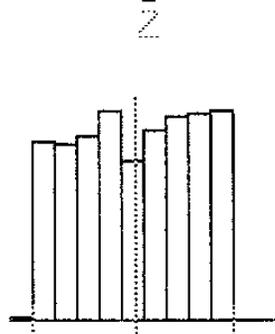


Fig. 4

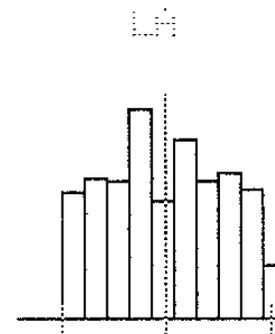
Distribuzioni degli outputs desiderati



Um= 0.03
Ds= 0.58
Mx= 1.00
Mi=-1.00

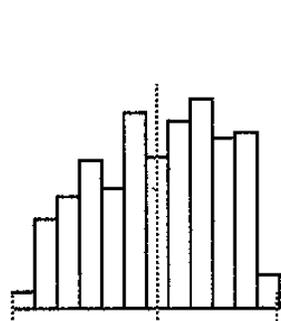


Um= 0.03
Ds= 0.58
Mx= 1.00
Mi=-1.00

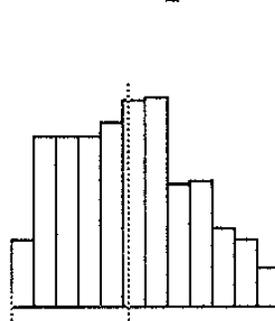


Um=-0.01
Ds= 0.56
Mx= 1.00
Mi=-1.00

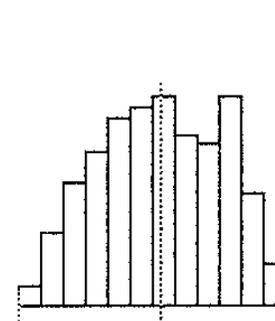
Distribuzioni degli outputs ottenuti



Um= 0.03
Ds= 0.50
Mx= 0.97
Mi=-1.13

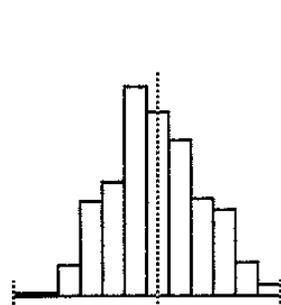


Um=-0.05
Ds= 0.63
Mx= 1.45
Mi=-1.22

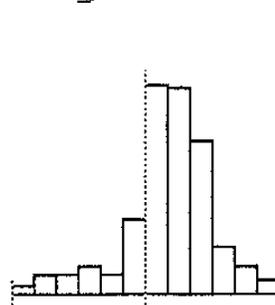


Um=-0.07
Ds= 0.58
Mx= 1.14
Mi=-1.44

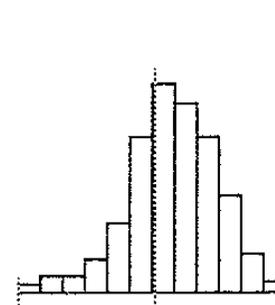
Distribuzione degli Errori (U.desid.-U.otten.)



Um=-0.00
Ds= 0.14
Mx= 0.37
Mi=-0.44



Um= 0.08
Ds= 0.15
Mx= 0.46
Mi=-0.46



Um= 0.07
Ds= 0.17
Mx= 0.48
Mi=-0.50

Esc ← →

Fig. 5