

# Proiect - Control Inteligent

Alexandru Zigler

2025

Se consideră funcția de transfer:

$$H_f(s) = \frac{3}{8s^2 + 6s + 1}.$$

Se dorește determinarea unei rețele neuronale care modelează comportamentul acestei funcții de transfer. În acest sens, se generează date de antrenare pentru rețeaua neuronală, simulând răspunsul funcției de transfer la semnal de intrare de tip zgomot alb. Un aspect important este alegerea timpului de eşantionare, pentru înregistrarea semnalelor din simulare. S-a ales  $T_e = 0.01$  secunde. Datele de intrare, respectiv de ieșire sunt prelucrate (transpusă și stocate ca celule), acest format fiind acceptat pentru pregătirea datelor de antrenare. Se definește o rețea neuronală autoregresivă cu intrări exogene (NARX). Aceasta are ca parametri întârzieri de intrare (5), întârzieri de feedback (5) și numărul de neuroni din stratul ascuns (10). Se stabilesc parametrii de antrenare: numărul de epoci (1000), eroarea dorită ( $10^{-40}$ ) și gradientul minim ( $10^{-3000}$ ). De asemenea, se specifică faptul că funcțiile de activare ale straturilor sunt liniare. Pe baza datelor salvate anterior, de intrare și ieșire, se construiesc secvențele  $X_s$  și  $T_s$  cu care se antrenează rețeaua.

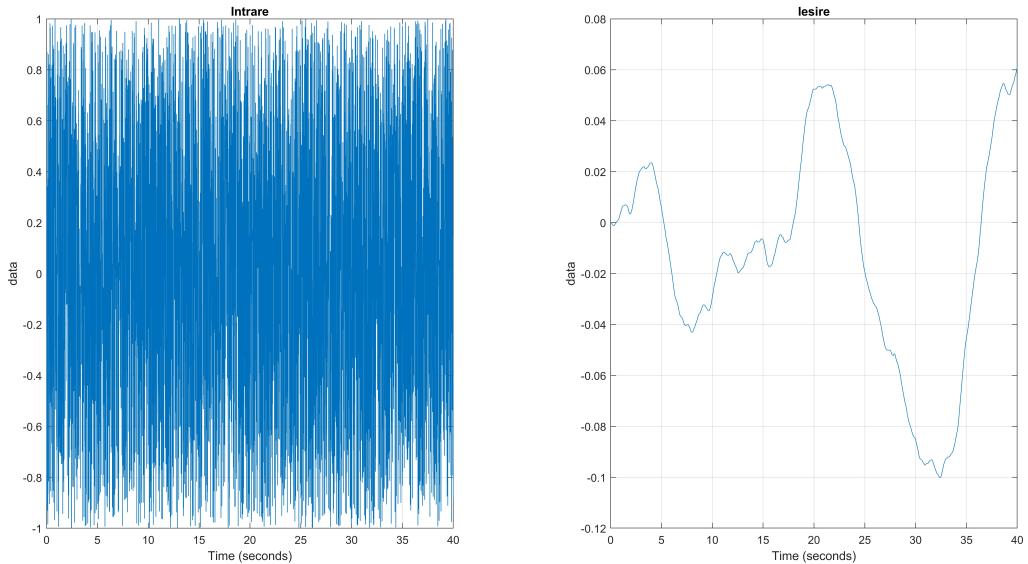


Figure 1: Datele de antrenare

Pentru validare, se compară ieșirea reală din datele de antrenare, cu răspunsul rețelei neuronale. De asemenea, se poate folosi o funcție (*perform*), care calculează performanța rețelei neuronale.

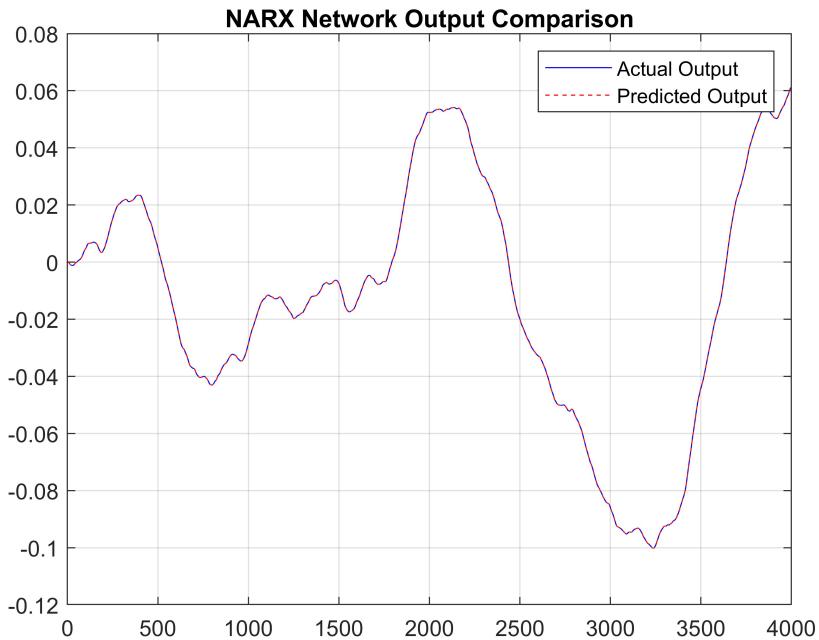


Figure 2: Ieșirea reală comparativ cu răspunsul rețelei neuronale

În timpul antrenării, rețeaua folosește semnalul de ieșire cunoscut din setul de date de antrenare. În schimb, în timpul predicției sau testării, semnalul de ieșire nu este accesibil, de aceea, rețeaua se bazează pe propriile ei predicții. Astfel, este necesară închiderea buclii pentru ca rețeaua să aibă acces la propriile predicții.

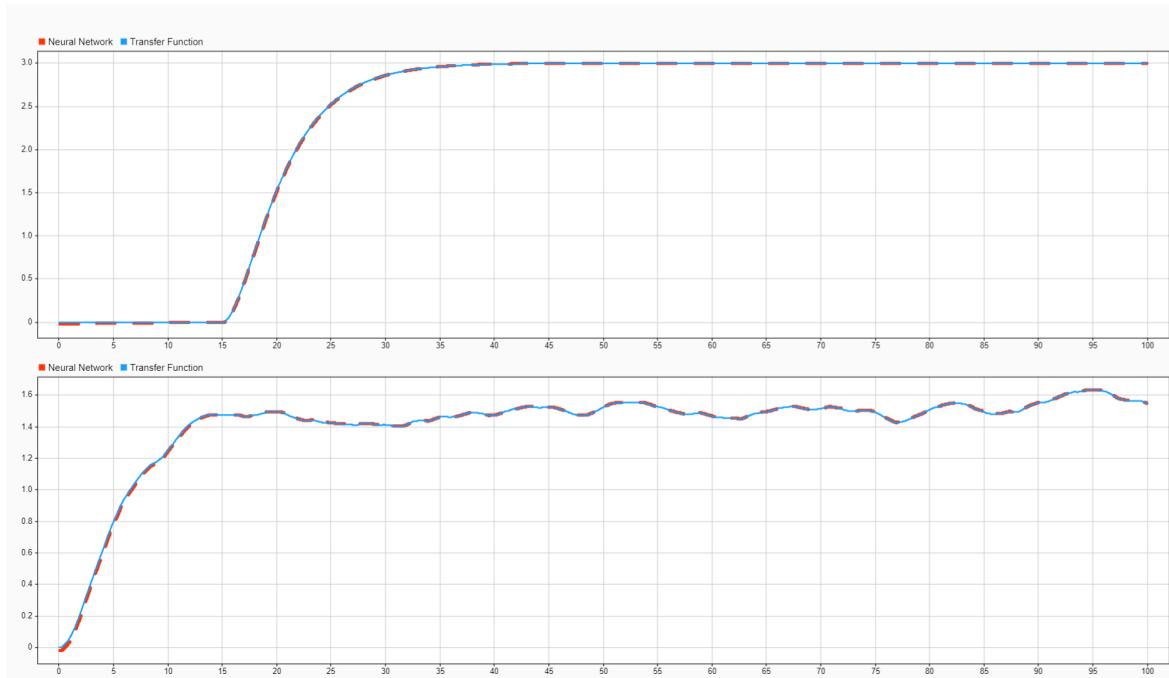


Figure 3: Răspunsul la treaptă (sus) și la zgomot alb (jos)

Se constată că rețeaua neuronală reușește să modeleze cu performanțe bune funcția de transfer impusă, atât în cazul intrării de tip treaptă, cât și în cazul zgomotului alb. Eroarea medie pătratică la antrenare este de ordinul  $10^{-12}$ .

În continuare, se dorește proiectarea unui regulator corespunzător sistemului considerat și determinarea performanțelor acestuia. Astfel, propunem metoda Ziegler-Nichols în buclă deschisă pentru un regulator de tip PID.

$$H_{\text{PID}} = k_p \left( 1 + \frac{1}{k_i} \cdot \frac{1}{s} + k_d \cdot \frac{N}{1 + N \cdot \frac{1}{s}} \right)$$

Cu metoda Ziegler-Nichols, obținem  $k_p = 24.2$ ,  $k_i = 0.6$  și  $k_d = 0.15$ . Totuși, pentru performanțe superioare, s-au recalibrat parametrii:  $k_p = 5.37$ ,  $k_i = 4.47$ ,  $k_d = 1.12$  și s-a ales  $N = 100$  ca filtru pentru componenta derivativă.

Suplimentar, pentru a asigura o formă strict proprie a regulatorului, s-a adăugat un filtru de forma  $H_{\text{filtru}} = \frac{1}{(0.01s + 1)^2}$ .

Forma finală a regulatorului va fi  $H_r = H_{\text{PID}} \cdot H_{\text{filtru}}$ .

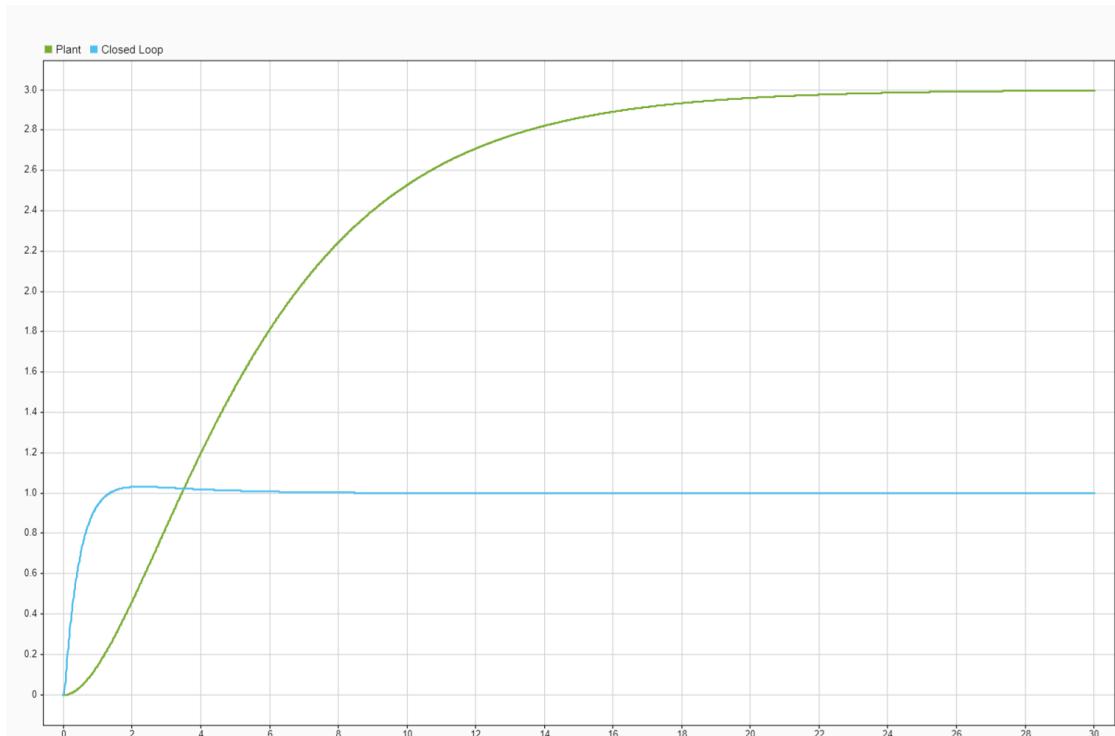


Figure 4: Răspunsul indicial al sistemelor în buclă deschisă, respectiv închisă

Performanțele regulatorului sunt:

- timp de răspuns 3.95 secunde;
- suprareglaj 3.15%;
- timp de ridicare 0.82 secunde;
- eroare staționară la poziție nulă.

Pentru învățarea comportamentului regulatorului se procedează în mod analog cu cazul anterior, al procesului. La intrarea funcției de transfer a regulatorului se folosește semnal de intrare de tip zgomot alb și se înregistrează atât intrarea, cât și ieșirea regulatorului. De această dată, s-a extins timpul de simulare la 200 secunde, pentru a crește performanța de învățare.

Parametrii de antrenare a rețelei neuronale sunt:

- timpul de eşantionare 0.01 secunde;
- întârzieri de intrare și de feedback: 7;
- număr de neuroni în stratul ascuns: 20;
- număr de epoci: 1000;
- eroare:  $10^{-40}$ ;
- gradient minim:  $10^{-3000}$ .

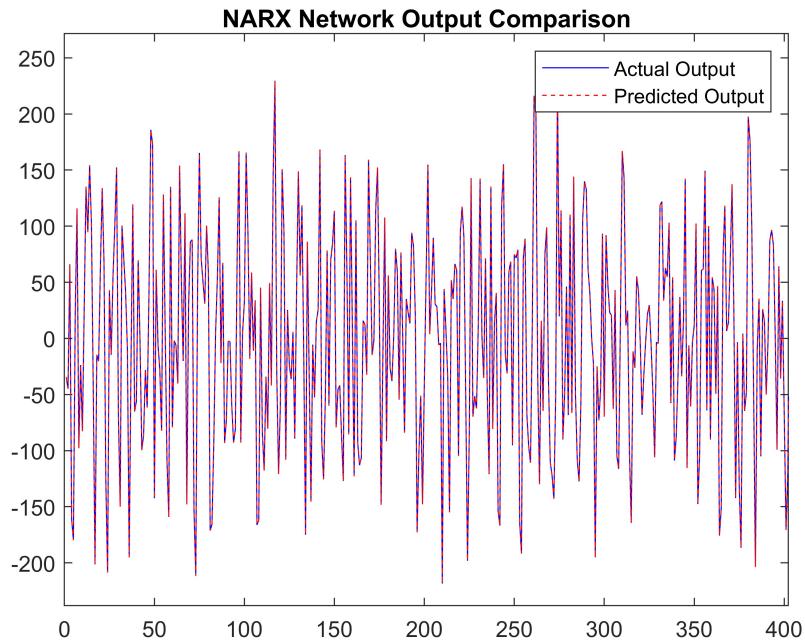


Figure 5: Răspunsul rețelei, comparat cu ieșirea din datele de antrenare (*zoom*)

Pentru validarea rețelei neuronale ce modelează regulatorul proiectat, se consideră schema din Figura 6, care testează în buclă închisă ambele regulațoare. Rezultatele se pot observa în Figura 7.

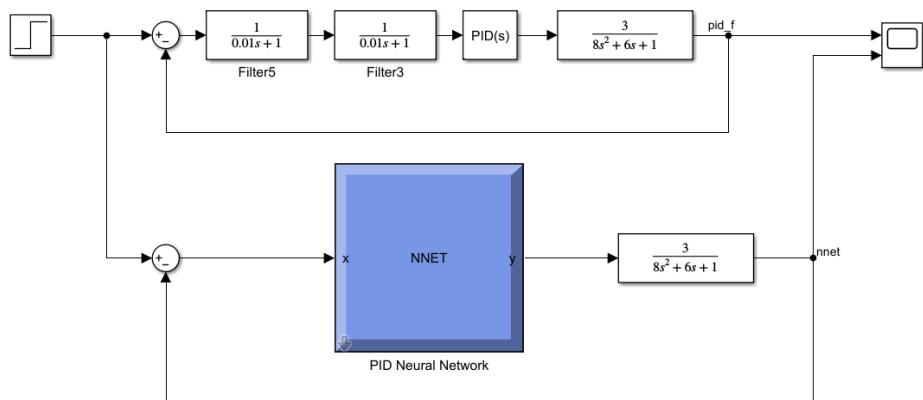


Figure 6: Schemă de control pentru validarea rețelei neuronale

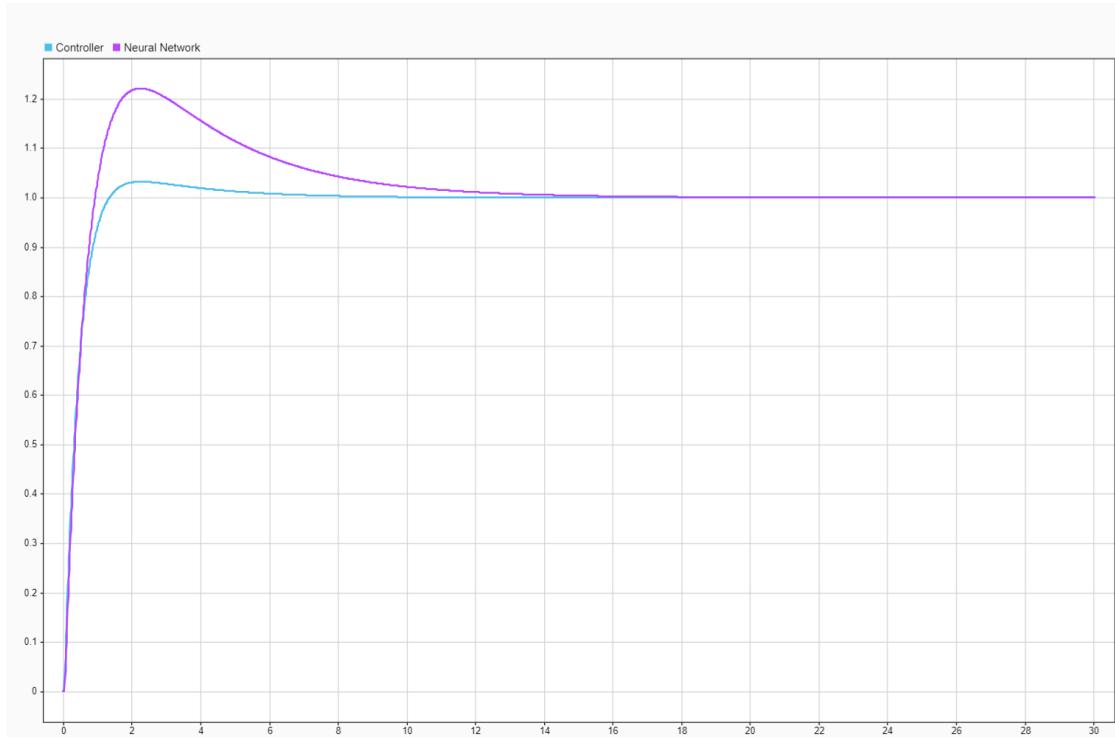


Figure 7: Răspunsurile în buclă închisă cu regulator ( $H_r$ ) și cu rețea neuronală

Performanțele de regim tranzitoriu ale rețelei neuronale, anume suprareglajul și timpul de răspuns, s-au înrăutățit vizibil, cu prețul menținrii erorii la poziție foarte mici ( $\approx 10^{-4}$ ).

În continuare, se dorește implementarea unei structuri de reglare de tip IMC, cu regulator neuronal. În acest context, se folosește modelul invers al părții fixate pe post de regulator.

$$H_{inv} = \frac{8s^2 + 6s + 1}{3}.$$

Fiind o funcție de transfer impropriu, introducem un filtru de ordinul 3, de forma:

$$H_{filtru1} = \frac{1}{(0.1s + 1)^3}.$$

Obținem astfel regulatorul  $H_{r1} = \frac{8s^2 + 6s + 1}{3(0.1s + 1)^3}$ . Dorim, de asemenea, să antrenăm o

rețea neuronală care să modeleze comportamentul său. În mod asemănător cu cazurile anterioare, folosim un semnal de intrare de tip zgomot alb, timp de 40 de secunde.

Parametrii de antrenare a rețelei neuronale sunt:

- timpul de eşantionare 0.01 secunde;
- întârzieri de intrare și de feedback: 4;
- număr de neuroni în stratul ascuns: 14;
- număr de epoci: 2000;
- eroare:  $10^{-60}$ ;

- gradient minim:  $10^{-3000}$ .

Un prim pas în validarea rețelei neuronale este analiza comportării ei în buclă închisă, comparativ cu regulatorul care se dorește a fi modelat. Se realizează o schemă de control asemănătoare cu Figura 6.

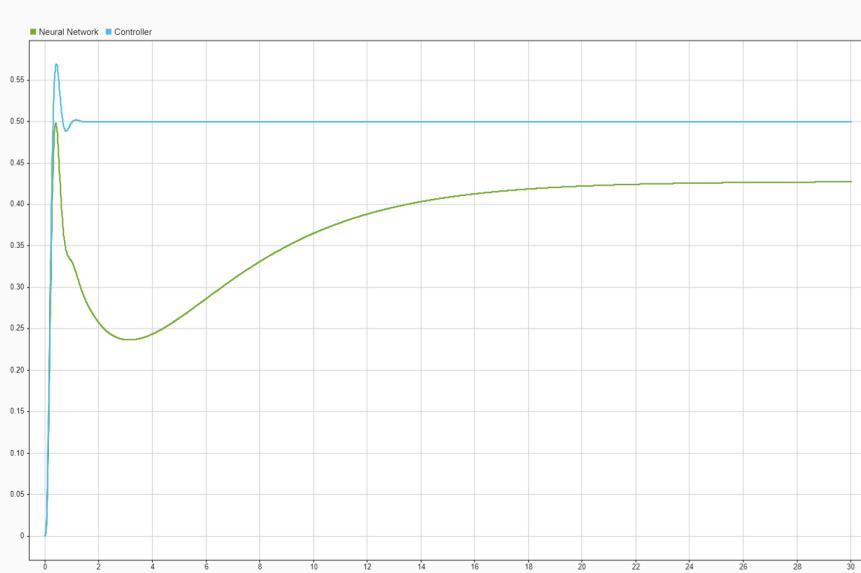


Figure 8: Răspunsurile în buclă închisă cu regulator ( $H_{r1}$ ) și cu rețea neuronală

Observăm că performanțele tranzitorii sunt asemănătoare la formă cu cele ale regulatorului, însă eroare la poziție în regim staționar este nesatisfăcătoare. În acest sens, se reia procesul de antrenare.

$$\text{Se schimbă și filtrul folosit, } H_{\text{filtru}2} = \frac{1}{(0.7s + 1)^3} \implies H_{r2} = \frac{8s^2 + 6s + 1}{3(0.7s + 1)^3}.$$

Pentru a îndeplini și cerințele de regim staționar, vom antrena rețeaua neuronală cu un set de date care, pe lângă zgomotul alb, conține porțiuni de tip treaptă. În plus, se mai schimbă numărul de neuroni din stratul ascuns de la 14 la 15 și numărul de întârzieri de la 4 la 6.

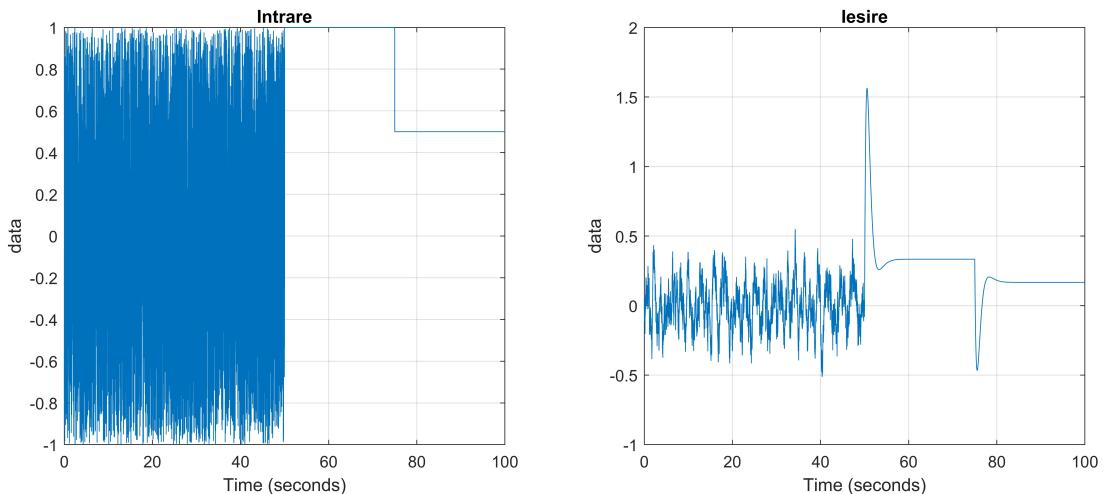


Figure 9: Date de intrare și ieșire pentru reantrenarea rețelei

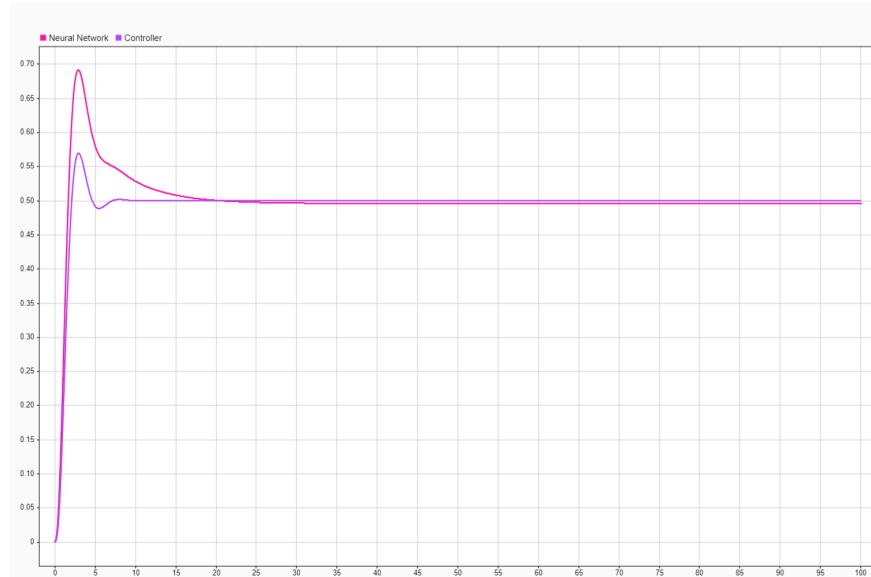


Figure 10: Răspunsurile în buclă închisă cu regulator ( $H_{r2}$ ) și cu rețea neuronală

Observăm că performanțele s-au îmbunătățit, atât în regimul staționar, cât și în cel tranzitoriu. Eroarea de aproximare în regim staționar este de ordinul  $10^{-3}$ .

Rețeaua neuronală obținută, care modelează regulatorul  $H_{r2}$  cu erori acceptabile, va fi introdusă în structura de reglare de tip IMC (Figura 11), alături de rețeaua neuronală care modelează procesul.

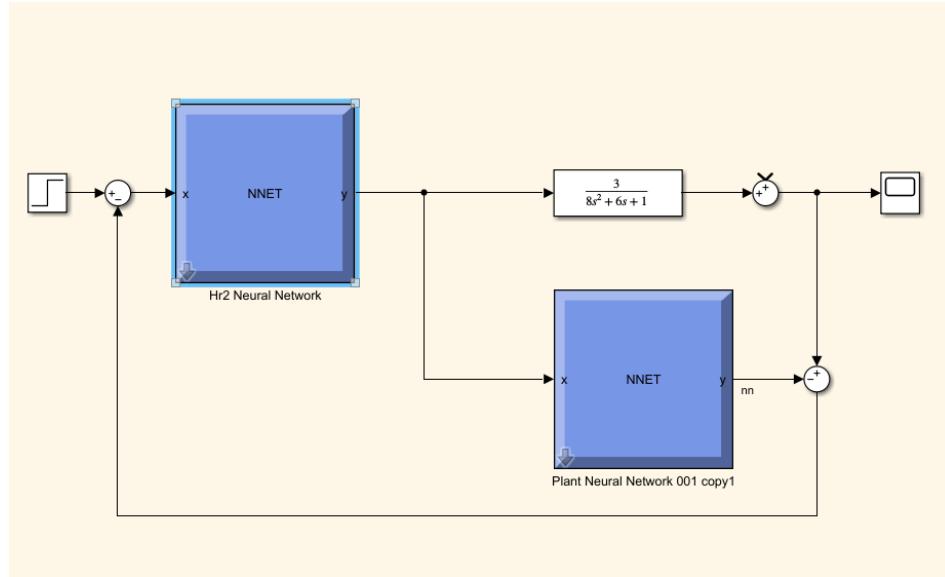


Figure 11: Structură de reglare IMC, folosind rețele neuronale

În Figura 12 se reprezintă simularea fără perturbații. Eroarea staționară la poziție este 0.02.

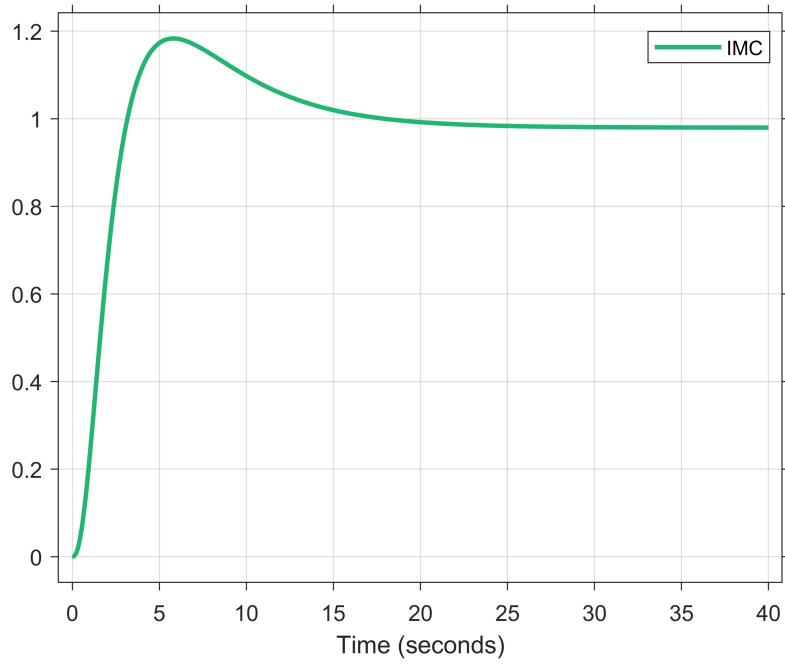


Figure 12: Sistemul controlat cu structura de tip IMC, fără perturbații

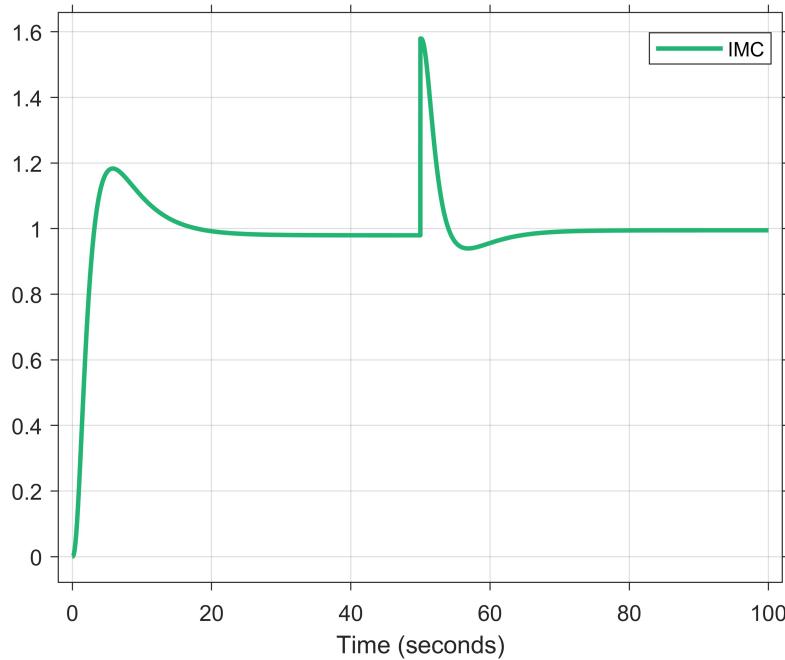


Figure 13: Sistemul controlat cu structura IMC, cu perturbație treaptă la secunda 50

În Figura 13 se poate observa că perturbația este rejectată complet de reglarea IMC cu rețele neuronale.

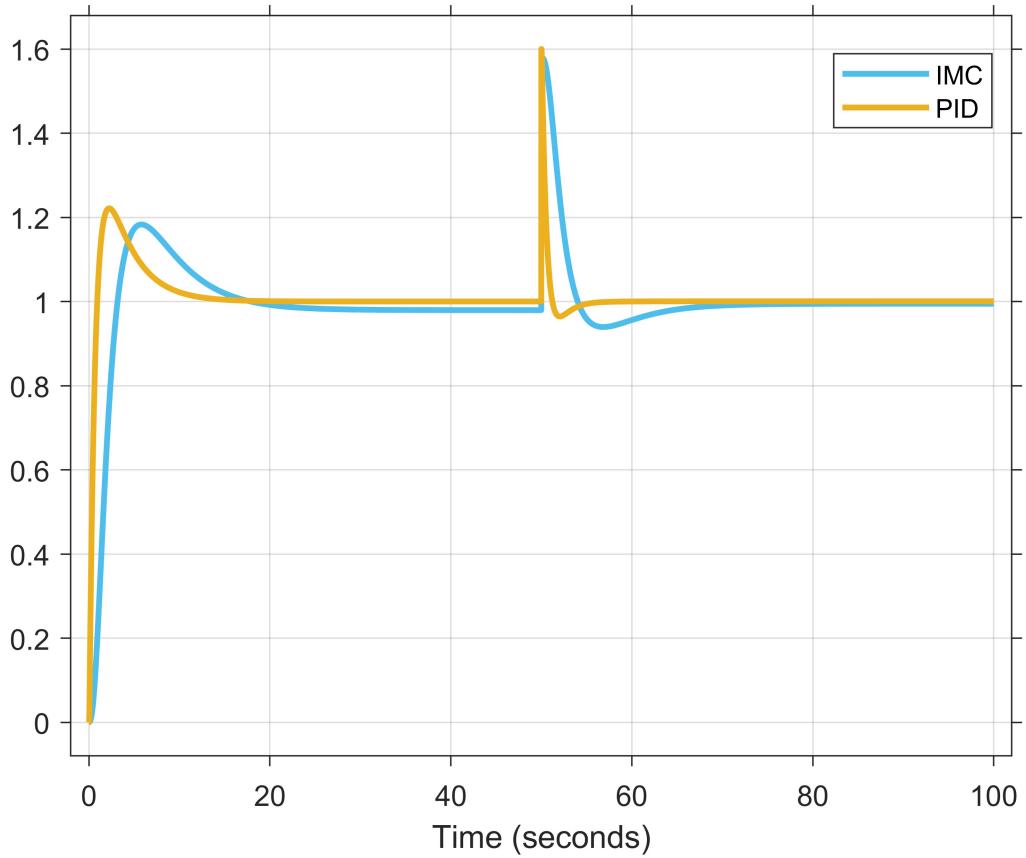


Figure 14: Reglare IMC versus reglare monocontur

Pentru comparația dintre cele 2 tipuri de control, se va considera răspunsul indicial unicitar, respectiv comportamentul la o perturbatie de valoare 0.6. În cazul regimului staționar, reglarea monocontur are o eroare la poziție de  $10^{-4}$ , iar reglarea IMC are o eroare de 0.0204. Reglarea monocontur are timpul de raspuns de 17.8 secunde și suprareglaj de 22.2%. Reglarea IMC are timp de răspuns de 10.3 secunde, iar suprareglajul este 20.3%. Timpul de urcare este mai mic la reglarea monocontur.

În final, se dorește proiectarea unui regulator online, folosind blocul din Simulink NARMA L2- Controller. În acest scop, s-a creat un model Simulink suplimentar care poate fi simutat programatic.

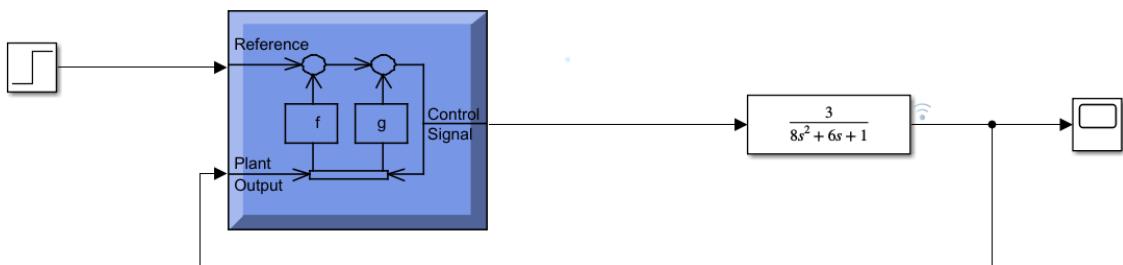


Figure 15: Schemă Simulink pentru proiectarea și testarea regulatorului online

Parametrii de antrenare s-au ales cei din Figura 16. Se generează automat date de

antrenare, conform parametrilor impuși.

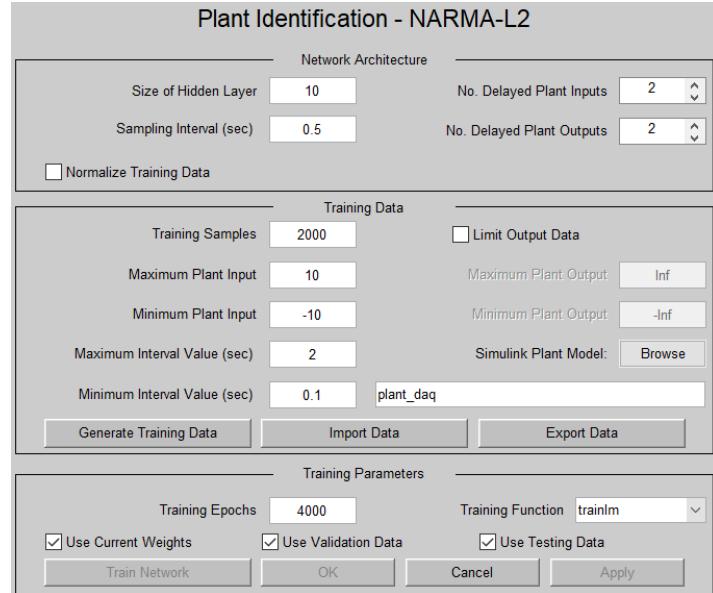


Figure 16: Parametrii de antrenare a regulatorului online

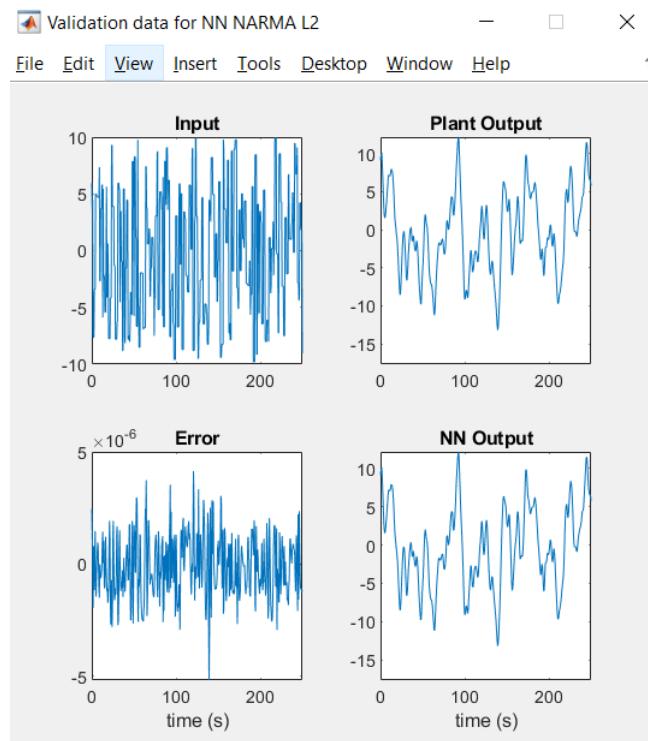


Figure 17: Date de validare a rețelei neuronale corespunzătoare regulatorului online

Performanța antrenării rețelei este  $10^{-12}$ .

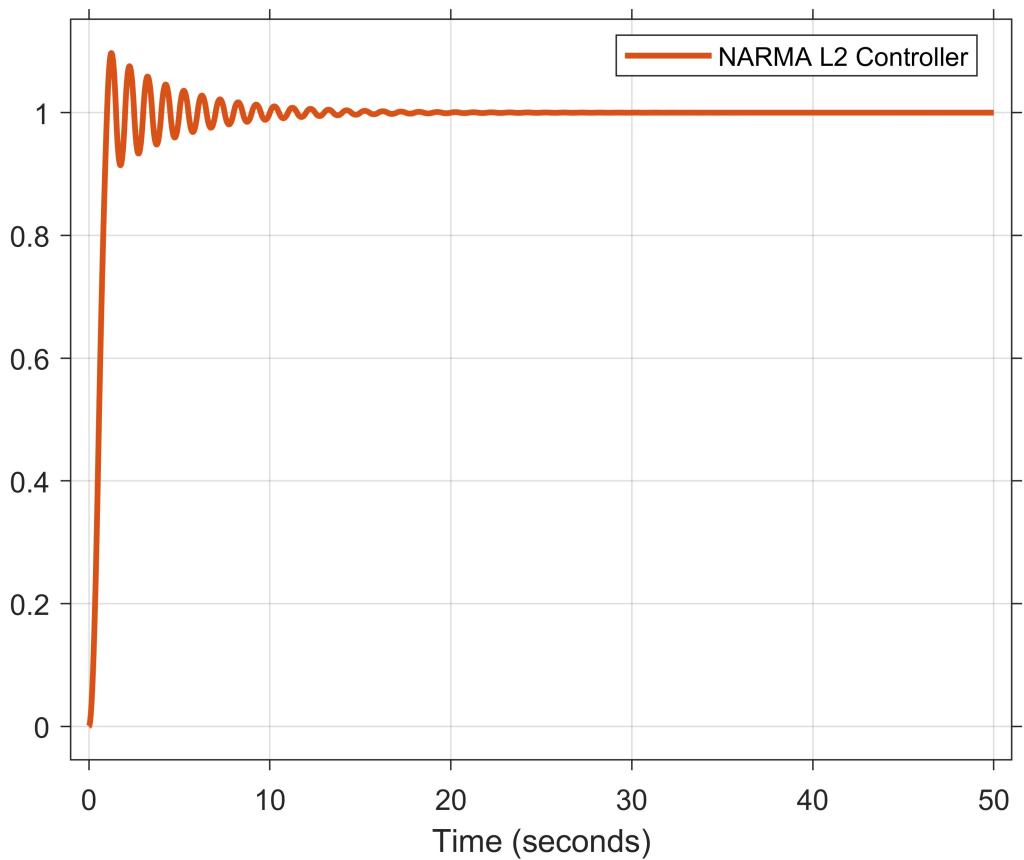


Figure 18: Răspunsul sistemului în buclă închisă cu regulator online

Observăm că performanțele regulatorului sunt foarte sensibile la datele de antrenare. Eroarea staționară la poziție este  $10^{-7}$ , timpul de răspuns este aproximativ 10 secunde, iar suprareglajul 10%.