

Achiziția de date folosind NI6001 și LabView și dezvoltarea unui sistem de învățare automată folosind datele achiziționate, în vederea modelării unui proces de transfer termic

GHEORGHE Andrei-Marian, RĂCMAN Eduard-Nicolae
Mecatronica sistemelor biotehnice – Fac. Ingineria Sistemelor Biotehnice

Rezumat:

Lucrarea prezintă construcția și utilizarea unui modul pentru achiziția de date folosind NI6001 și LabView și dezvoltarea unui sistem de învățare automată bazat pe datele colectate. În cadrul experimentului, a fost încălzit un capăt al unei bare de aluminiu și apoi au fost măsurate temperaturile în patru puncte distincte, utilizând o diagramă LabView. Scopul a fost modelarea și identificarea procesului de transfer de căldură prin bara de aluminiu și evaluarea aplicabilității modelului în practică. Rezultatele au aratat că limitările modelului clasic, datorate neliniarităților și condițiilor experimentale specifice, pot fi depășite printr-o identificare de sistem cu ajutorul modelelor de *machine learning*. Datele obținute au fost prelucrate utilizând metode de regresie, cu rezultate foarte promițătoare. Acest demers se dovedește deosebit de util în proiectarea și optimizarea sistemelor biotehnice, având în vedere importanța proceselor de transfer de căldură în instalațiile agricole și din industria alimentară. Lucrarea sugerează că metodele de învățare automată, aplicate pe date experimentale precise, pot îmbunătăți semnificativ rezultatele modelării acestor procese complexe.

A.Introducere

Achiziția de date cât mai precisă este esențială în modelarea și controlul proceselor termice din agricultură și industria alimentară, deoarece acuratețea datelor influențează direct calitatea modelelor și a deciziilor derivate din acestea. Procesele termice implică variații complexe și rapide ale temperaturii, iar orice eroare în măsurare poate conduce la interpretări incorecte și la adoptarea unor strategii ineficiente. Precizia datelor este crucială pentru validarea modelelor teoretice, pentru calibrarea algoritmilor de control și pentru monitorizarea în timp real a proceselor. În lipsa unor date precise, riscurile includ ineficiența energetică, deteriorarea calității produselor și compromiterea siguranței alimentare.

Achiziția de date cu modulul NI 6001 și interfața LabView reprezintă o soluție fezabilă datorită mai multor avantaje. Modulul NI 6001 este un dispozitiv de achiziție de date compact, accesibil și ușor de integrat în diverse sisteme experimentale. Modulul oferă o rezoluție de 14 biți, ceea ce asigură o precizie adecvată pentru majoritatea aplicațiilor industriale și de cercetare. De asemenea, NI 6001 dispune de un număr suficient de canale analogice pentru a permite monitorizarea simultană a mai multor puncte de măsurare, așa cum a fost necesar în experimentele noastre cu bara de aluminiu.

LabView, ca mediu de dezvoltare grafică, facilitează configurarea și gestionarea sistemelor de achiziție de date. Interfața sa intuitivă permite realizarea rapidă a diagramelor de control și achiziție, reducând timpul și efortul necesar pentru implementarea sistemului. În plus, LabView oferă capacități avansate de procesare a

datelor, incluzând filtrare, analiză spectrală și transformate Fourier, care sunt esențiale pentru prelucrarea și interpretarea corectă a datelor achiziționate.

Un alt avantaj al utilizării NI 6001 și LabView este flexibilitatea și scalabilitatea soluției. Sistemul poate fi adaptat cu ușurință pentru a răspunde cerințelor specifice ale diferitelor aplicații din agricultură și industria alimentară. De exemplu, în procesele de uscare, unde distribuția temperaturii de-a lungul materialului este foarte importantă, se pot adăuga senzori suplimentari și se pot ajusta parametrii de achiziție fără a necesita modificări majore ale infrastructurii hardware sau software.

Pasteurizarea și sterilizarea sunt utilizate în industria alimentară pentru distrugerea microorganismelor patogene și pentru prelungirea duratei de viață a produselor alimentare. Prin modelarea procesului de transfer termic, se pot determina valorile optime ale temperaturii și timpului de încălzire/răcire, asigurând eficacitatea procesului de sterilizare fără a compromite calitatea senzorială și nutrițională a alimentelor. Răcirea și congelarea produselor alimentare sunt esențiale pentru păstrarea prospețimii și prelungirea duratei de viață a produselor perisabile. Modelarea proceselor de transfer termic ajută la proiectarea echipamentelor de răcire și congelare, optimizând distribuția temperaturii și minimizând consumul energetic, astfel încât să se poată menține calitatea produselor și să se evite formarea cristalelor de gheață mari, care pot deteriora structura alimentelor. De asemenea, fermentarea controlată este utilizată în producția de produse lactate, vin, bere și alte alimente. Controlul temperaturii în timpul fermentării este deosebit de important pentru a asigura dezvoltarea corectă a microorganismelor benefice și pentru a obține produse de calitate superioară. Modelarea proceselor de transfer termic poate ghida proiectarea și operarea echipamentelor de fermentare pentru a menține condițiile optime.

De asemenea, producția și prelucrarea uleiurilor vegetale implică procese termice complexe, iar modelarea transferului de căldură poate optimiza etapele de extracție și rafinare, reducând pierderile de ulei și consumul de energie și asigurând, în același timp, menținerea calității produsului final.

În cadrul experimentului nostru, a fost încălzită o bară de aluminiu la un capăt utilizând un tranzistor și apoi a fost măsurată temperatura în patru puncte distincte de-a lungul barei.

Pentru a descrie transferul de căldură în bara de aluminiu, se poate recurge la ecuația clasică a transferului de căldură prin conducție, cunoscută sub numele de ecuația lui Fourier:

$$\partial t / \partial T = \alpha \cdot \nabla^2 T$$

unde T este temperatura, t este timpul, și α este difuzivitatea termică a materialului.

Deși ecuația lui Fourier oferă o bază teoretică solidă pentru analiza proceselor de conducție termică în materiale omogene și izotrope, aplicarea sa în practică în procesele din agricultură și industria alimentară prezintă câteva limitări semnificative, evidente și în contextul experimentului nostru:

1. ecuația lui Fourier este adesea utilizată sub ipoteza că sistemul a ajuns la o stare de echilibru termic sau se apropie de aceasta. În cazul nostru, încălzirea cu un tranzistor poate induce variații rapide de temperatură, ceea ce face dificilă aplicarea acestei ipoteze;
2. în procesele reale, materialele pot prezenta variații ale proprietăților termice (cum ar fi conductivitatea termică) de-a lungul lungimii barei, fie din cauza impurităților, fie din

cauza defectelor structurale. Ecuația lui Fourier presupune o conductivitate termică constantă, ceea ce poate duce la erori în predicțiile modelului;

3. În cadrul experimentului, au loc pierderi de căldură prin convecție și radiație, mai ales că experimentul are loc într-un mediu deschis. Ecuația lui Fourier nu ține cont în mod direct de aceste mecanisme de pierdere de căldură, necesitând astfel termeni suplimentari pentru a obține un model precis;
4. În practică, barele de aluminiu au dimensiuni finite și condiții de frontieră complexe (de exemplu, capetele barei pot fi supuse unor condiții de temperatură specifice). Rezolvarea ecuației lui Fourier cu aceste condiții necesită metode numerice complexe și poate introduce erori suplimentare.

Limitările menționate evidențiază necesitatea unui model mai cuprinzător și a unei abordări experimentale care să poată descrie complexitatea sistemului.

B.Descrierea modului experimental

Modulul de aplicații facilitează lucrul cu dispozitivul de tip DAQ¹ *National Instruments USB-6001* prin intermediul perifericelor I/O² atașate celor patru tipuri de interfețe disponibile:

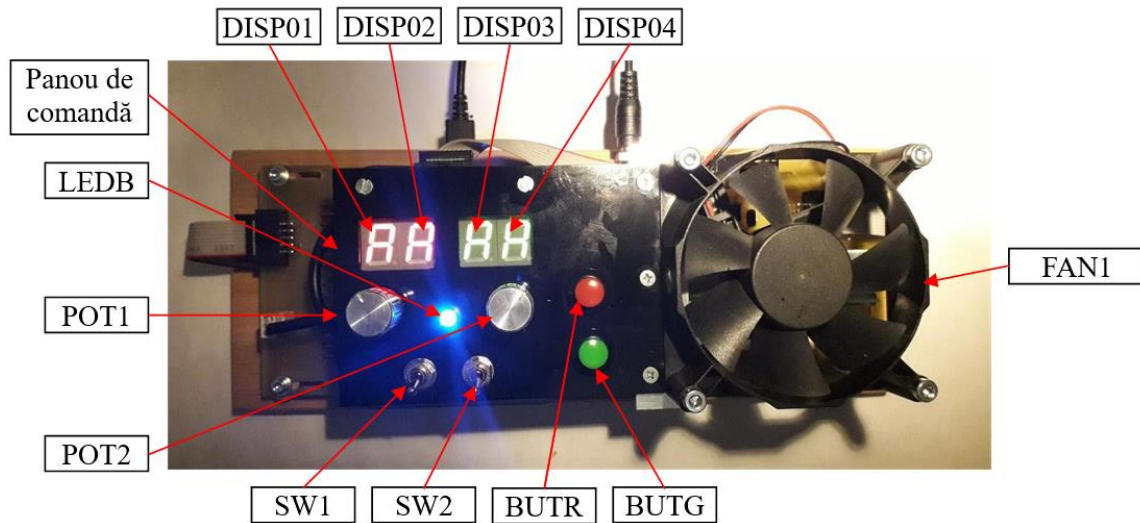


Fig.1 Aspectul fizic și schema de ansamblu a modului bazat pe NI-USB6001

- intrări analogice: senzori de temperatură (4 + 1 AI³), potențiometre de uz general (2 AI);
- ieșiri analogice: comandă rezistență încălzire bară aluminiu și comandă ventilator (1 + 1 AO⁴);
- intrări digitale: comutatoare bipoziționale și butoane (2 + 2 DIO⁵);
- ieșiri digitale: LED albastru(1 DIO), decodificator BINAR/7segmente și comandă afișaje 7 segmente (4 DIO + 4 DIO).

1 DAQ – Data Acquisition system
2 Input/Output
3 Analog Input
4 Analog Output
5 Digital Input/Output

Elementele din componența modului, vizibile în imaginea din figura 1 sunt:

- Panoul de comandă
- Afișaje cu 7 segmente: DISP01, DISP02 (de culoare roșie), respectiv DISP03, DISP04 (de culoare verde)
- Led albastru: LEDB
- Potențiometre: POT1, POT2 (de tip liniar)
- Comutatoare: SW1, SW2
- Butoane: BUTR (roșu), BUTG (verde)
- Ventilator: FAN1 (80mm/12V)

Pentru procesul supus comenzii de încălzire/răcire a fost folosită o bară rectangulară de aliaj de aluminiu prevăzută cu patru senzori de temperatură dispuși în lungul barei, elementul de încălzire de tip semiconductor fiind plasat la unul din capete (fig. 2.1, fig. 2.2).

1. Ieșire analogică – comandă ventilator

Ventilatorul (fig.1.1) se alimentează la o tensiune continuă nominală de 12V (bornele CON2-1 și CON2-2), putându-se obține o viteză de rotație diferită (în sensul scăderii sub turația nominală) prin modificarea tensiunii de alimentare, după o caracteristică liniară. Montarea acestuia deasupra barei de încălzire permite răcirea acesteia în sensul unei perturbații sau a unei comenzi.



Fig.1.1 Ventilator 90mm

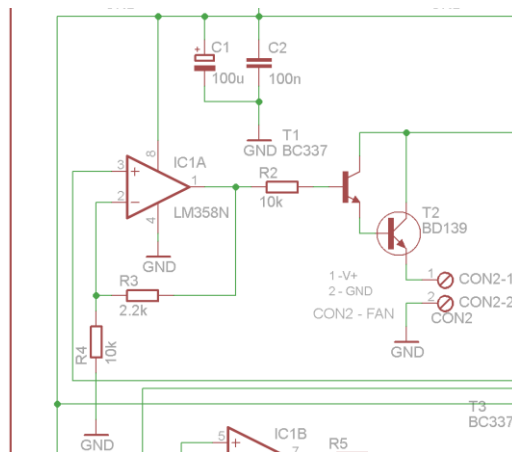


Fig.1.2 Schema electrică secțiune comandă ventilator

În acest sens, tensiunea electrică de la o ieșire analogică a NI-USB6001 (AO1) este conectată la intrarea neinvertată (pinul 3) a amplificatorului operațional IC1(A) (fig.

1,2), de tip LM358 (Texas Instruments[2]). Acesta amplifică ieșirea analogică din domeniul 0-10V (plaja maximă pozitivă de care este capabil NI-USB6001) în domeniul 0-12V (factor de amplificare 1.4). În acest sens, se impun rezistențele electrice R3 și R4 cu valorile din schemă:

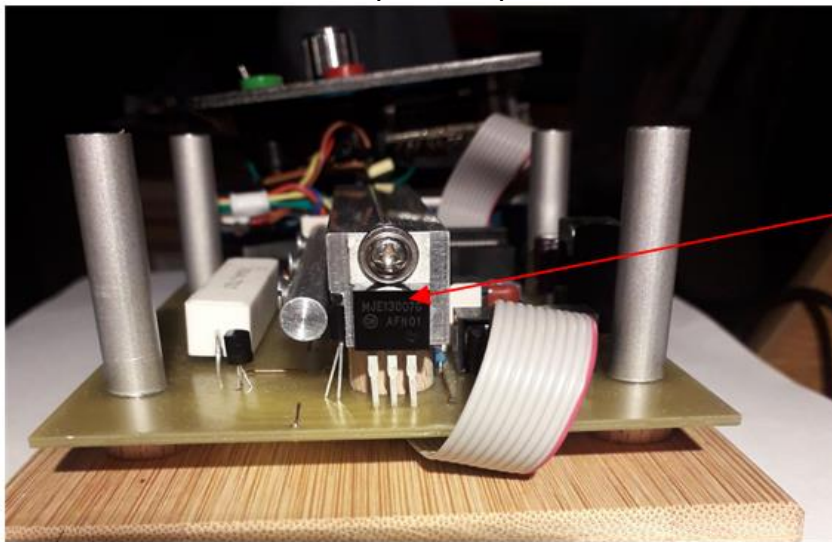
$$Gain=1+\frac{R3}{R4}=1+\frac{2.2k\Omega}{10k\Omega}=1+0.22=1.22$$

Se obține o valoare ușor diferită din cauza valorilor standard ale rezistențelor electrice. Astfel, pentru o tensiune maximă la ieșire de 10V => $V_{out} = Gain * 10V = 1.22 * 10V = 12.2V$.

Tranzistorul de putere medie BD139 [3] (T2 în fig.1.2) va disipa o putere maximă de aprox. 1W. Din specificul schemei, se disipă puterea maximă atunci când ventilatorul este alimentat cu tensiune joasă și implicit are o viteză de rotație redusă ($V_{out} > \sim 3V$). Utilizarea amplificatorului operațional și a tranzistorilor este justificată de necesitatea amplificării tensiunii de comandă de la 10V la 12V și totodată de limitarea dată de modulul NI-USB6001 asupra curentului de ieșire ($I_{out} < 4mA$, conform documentației tehnice).

2. Ieșire analogică – comandă element de încălzire

Ca element de încălzire s-a ales un tranzistor bipolar de uz comun (polaritate *npn*), de putere mare, MJE13007 [4] (elementul de încălzire din fig. 2.1, T4 în fig. 2.2). Acesta posedă o putere de disipare a căldurii de maxim 80W (suficientă pentru schema curentă), la un curent maxim de $I_{Cmax}=8A$ și o tensiune $V_{CEmax} = 400V$. Capsula acestuia de tip TO220 este optimă din punct de vedere mecanic pentru prinderea cât mai exactă la capătul barei de aluminiu ce reprezintă procesul efectiv de încălzire.



Element de
încălzire

Fig.2.1 Elementul de încălzire (tranzistor npn MJE13007 – ieșirea analogică AO0)

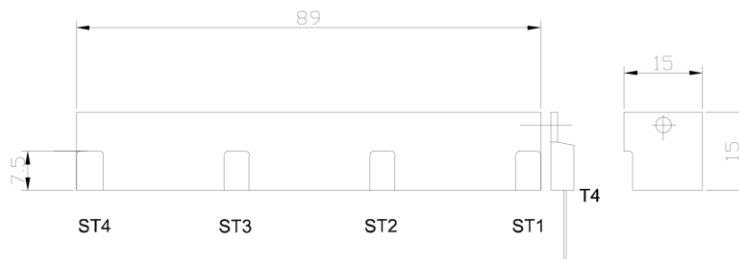


Fig.2.2 Bara de aluminiu în vedere laterală și frontală (s-au notat pozițiile celor 4 senzori și a tranzistorului de încălzire)

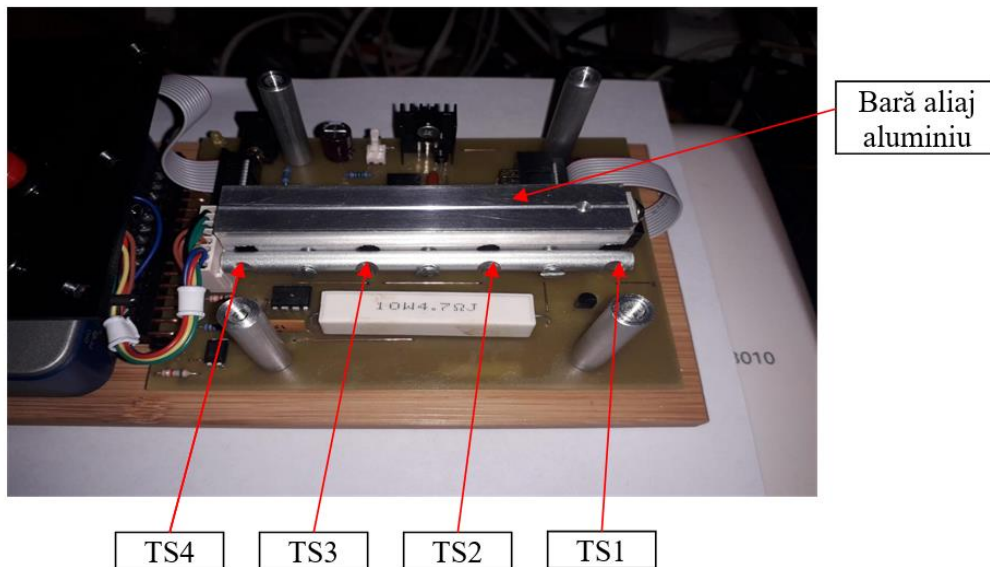


Fig.2.3 Bara de aliaj de aluminiu și senzorii de temperatură

Pentru a avea o caracteristică statică cât mai liniară a elementului de comandă, s-au ales componentele electronice și schema bazată pe un amplificator operațional în comandă de curent constant IC1(B). Conectarea directă dintre emitorul tranzistorului T4 și intrarea inversoare a IC1(B) (pinul 6) asigură egalitatea permanentă dintre tensiunea de intrare V_i (intrarea neinversoare – pinul 5) și tensiunea aflată la ieșirea lui T4 (emitorul tranzistorului – punctul V_e). Astfel, se poate respecta tot timpul relația $V_i = V_e$, care, luând în considerare valoarea constantă a rezistenței de putere (R11 – fig.2.4) va determina un curent constant I_c comandat de tensiunea de intrare V_i .

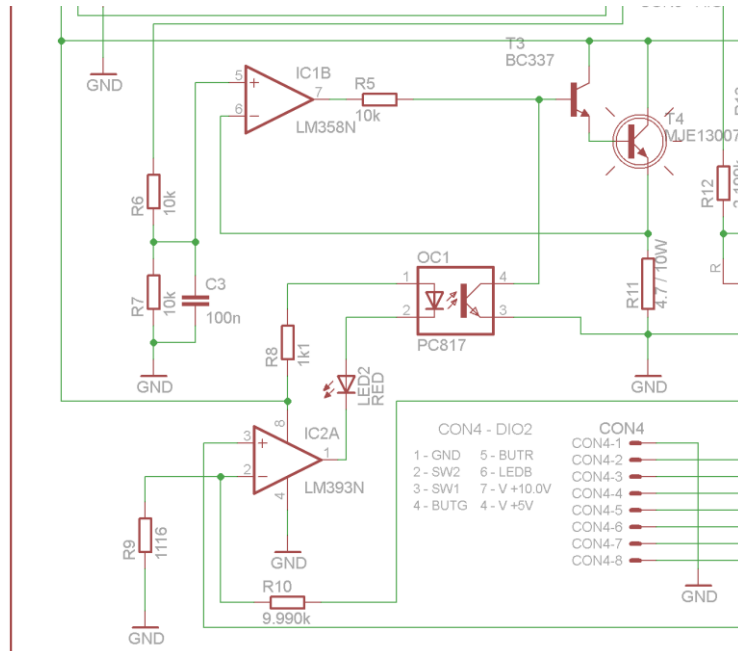


Fig.2.4 Schema electrică a secțiunii de comandă a încălzirii

Tensiunea de intrare V_i se obține divizând tensiunea de la ieșirea analogică a NI USB-6001 (AO2) cu ajutorul rezistențelor R6, R7 egale ca valoare. Astfel, dacă la intrarea schemei avem plaja de tensiune continuă 0 – 10V, la pinul 5 al IC1B corespunde $V_i = 0 - 5V$. Din egalitatea $V_i = V_e$, rezultă că se obține un curent prin tranzistorul Q1:

$$I_c = \frac{V_i = V_e}{R_{12}}$$

Valorile limită ale I_c sunt:

$$V_{i_{min}} = 0V \Rightarrow I_c = \frac{0V}{5\Omega} = 0A$$

$$V_{i_{max}} = 5V \Rightarrow I_c = \frac{5V}{5\Omega} = 1A$$

Puterea disipată de tranzistorul T4 și implicit caracteristica statică a întregului bloc este:

$$P_d = V_{CE} \cdot I_c = (15V - V_e) \cdot I_c = \frac{(15V - V_e) \cdot V_e}{R_e}$$

Întrucât la numărător avem pătratul tensiunii V_e , caracteristica obținută (fig. 2.5) nu este liniară (parabolă), însă prin alegerea valorilor tensiunii de intrare V_i și a rezistenței de putere R11 s-a realizat un bun compromis.

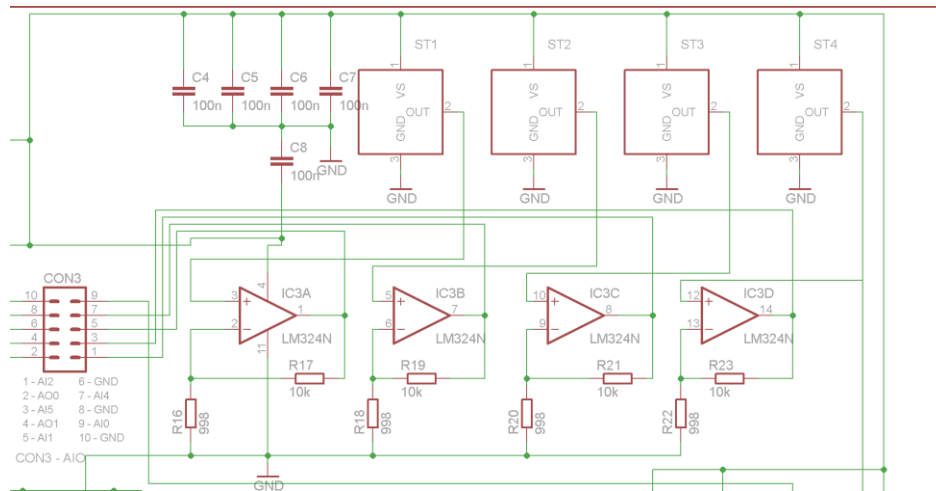


Fig.3.2 Secțiunea senzorilor de temperatură și a circuitelor de adaptare

Senzorii de temperatură ST1, ST2, ST3 și ST4 producând o tensiune continuă de valoare relativ mică (maxim 1.0 V pentru temperatura de 100°C), este necesară o amplificare obținută cu ajutorul a patru amplificatoare operaționale aflate integrate în circuitul IC3 – LM324. Schema de conectare a circuitului este de tip amplificator neinversor, cu un factor de amplificare:

$$Gain = 1 + \frac{R17}{R16} = 1 + \frac{10\text{ k}\Omega}{0.998\text{ k}\Omega} = 1 + 10.02 = 11.02$$

Rezistențele R17, R19, R21 și R23 îndeplinesc același rol, la fel ca R16, R18, R20 și R22. Dacă notăm V_{AIx} tensiunea amplificată (regăsită pe pinii de ieșire: IC3A-1, IC3B-7, IC3C-8, IC3D-14) ce se conectează la intrările NI – AIx, atunci temperatura se calculează cu:

$$Temp = \frac{V_{AIx}}{Gain} \cdot 100 = \frac{V_{AIx}}{11.02} \cdot 100 = V_{AIx} \cdot 9.074 \text{ [}^\circ\text{C]}$$

4. Intrări analogice – potențiometre de comandă

La intrările analogice AI3 și AI7, configurate în modul Single-Ended (RSE) sunt conectate două potențiometre cu valoarea de 50k ohmi fiecare.



Fig.4.1 Aspect fizic și schema de conectare potențiometrului de comandă

Acestea se alimentează de la o sursă de tensiune stabilizată de precizie având valoarea tensiunii de ieșire de 10.0V (fig.4.2). Această sursă este construită în jurul diodei stabilizatoare de precizie, programabilă VR1 – TL431. Cu ajutorul rezistențelor de programare R12 și R13 se obține o valoare de 10.0V.

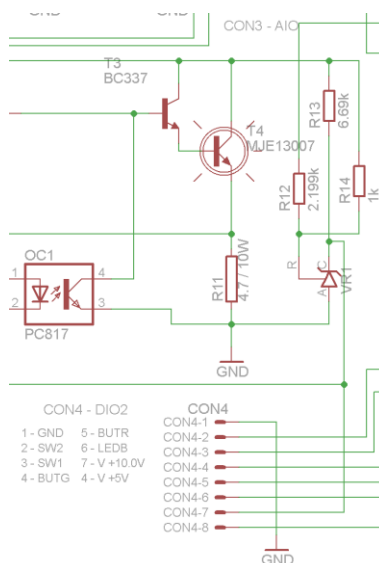


Fig.4.2 Secțiunea stabilizatorului de precizie programabil

Prin manevrarea butoanelor potențiometrelor se obține o variație liniară a tensiunii continue între 0.0 – 10.0V, acestea fiind conectate la intrările analogice AI3 (POT1), AI7 (POT2).

5. Intrări digitale – comutatoare și butoane ON/OFF

Pentru a oferi posibilitatea de realizare a unei game cât mai mari de aplicații, la intrările digitale (se presupune că s-a asigurat setarea corectă a sensului I/O: P1.0, P1.1, P1.2, P1.3) ale modulului NI, s-au conectat două butoane cu revenire (BUTR-red, BUTG-green) și două comutatoare (SW1, SW2), conform schemei din fig.5.1. Tensiunea continuă de alimentare pentru schemă este de +5V, fiind preluată de la ieșirea dedicată a modulului NI.

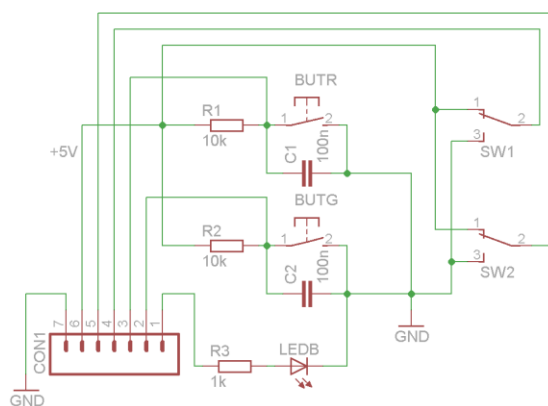


Fig.5.1. Secțiunea de intrări digitale plus LEDB

Condensatoarele C1, C2 cu valoarea de 100nF sunt utilizate pentru filtrarea oscilațiilor parazite care apar la apăsarea butoanelor cu revenire (filtru RC de ordinul I).

6. Ieșiri digitale – afișaj binar/7 segmente

Din cele 13 intrări/ieșiri digitale (P0-0...7, P1-0...3, P2.0), s-au configurat în schemă un număr de 9 ieșiri (restul de 4 intrări sunt explicate mai sus). Ieșirea aflată pe portul P2.0 este conectată la LEDB (blue), restul de 8 ieșiri (P0.0...P0.7) sunt utilizate pentru comanda unui afișaj multiplexat cu LED-uri în configurație 7 segmente. Acestea sunt divizate în două secțiuni:

- comanda efectivă a cifrei afișate (0, 1, 2, 3, 4, 5, 6, 7, A, B, C, D, E, F)
- comanda fiecărei cifre – starea ON/OFF

Pentru a reduce numărul de conexiuni fizice, este utilizat un decodificator BINAR – 7 SEGMENTE particularizat, realizat cu un microcontroller, IC1 – PIC16F676P (fig.6.1). S-a optat pentru această soluție întrucât s-a dorit afișarea cifrelor hexazecimale și utilizarea în acest mod a tuturor combinațiilor posibile pe 4 ieșiri (P0.0-A, P0.1-B, P0.2-C, P0.3-D). Tranzistorii bipolari de tip *nnp* T1, T2, T3 și T4 sunt comandați de restul de 4 ieșiri (P0.4-DISP01 red, P0.5-DISP02 red, P0.6-DISP03 green, P0.7-DISP04 green) pentru amplificarea curentului disponibil pentru LED-urile afișajului 7 segmente și posibilitatea opririi/pornirii individuale a celulelor de afișare.

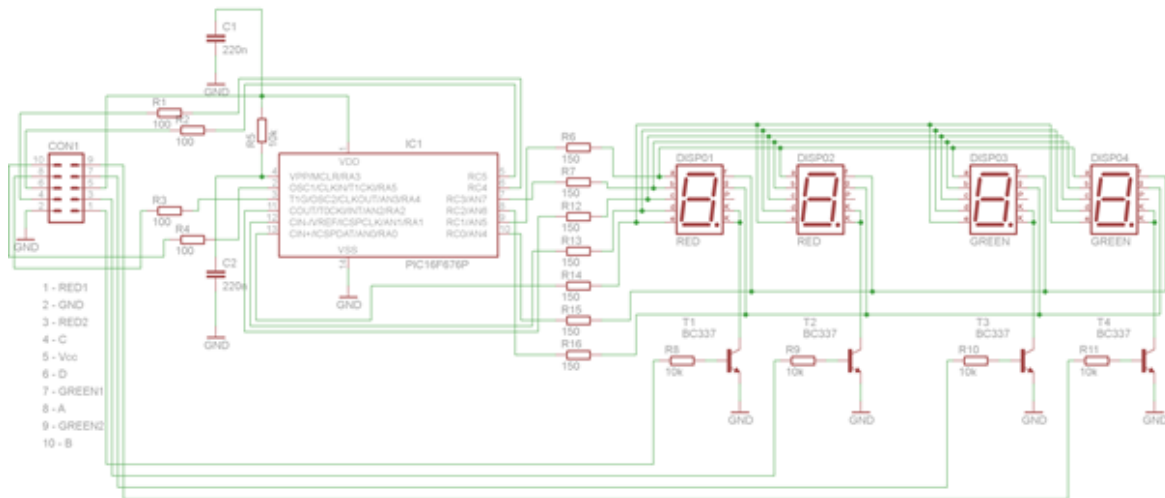


Fig.6.1 Secțiunea de afișare digitală cu celule 7 segmente

C. Diagrama Labview

Diagrama Labview este inclusă într-o buclă exterioară *while*, care este executată până la apăsarea butonului de stop sau până la oprirea rulării programului din bara de acțiuni a programului Labview.

Daq Assistant este un obiect care facilitează achiziția de date cu un dispozitiv National Instruments (NI6001, în cazul proiectului nostru). Pot fi alese limitări ale tensiunilor achiziționate, respectiv între 0 și 10 V, în cazul nostru. Toate aceste setări pot fi introduse în mod individual pentru fiecare port. Datele ies din acest obiect sub forma unui singur „fir” de date din „data

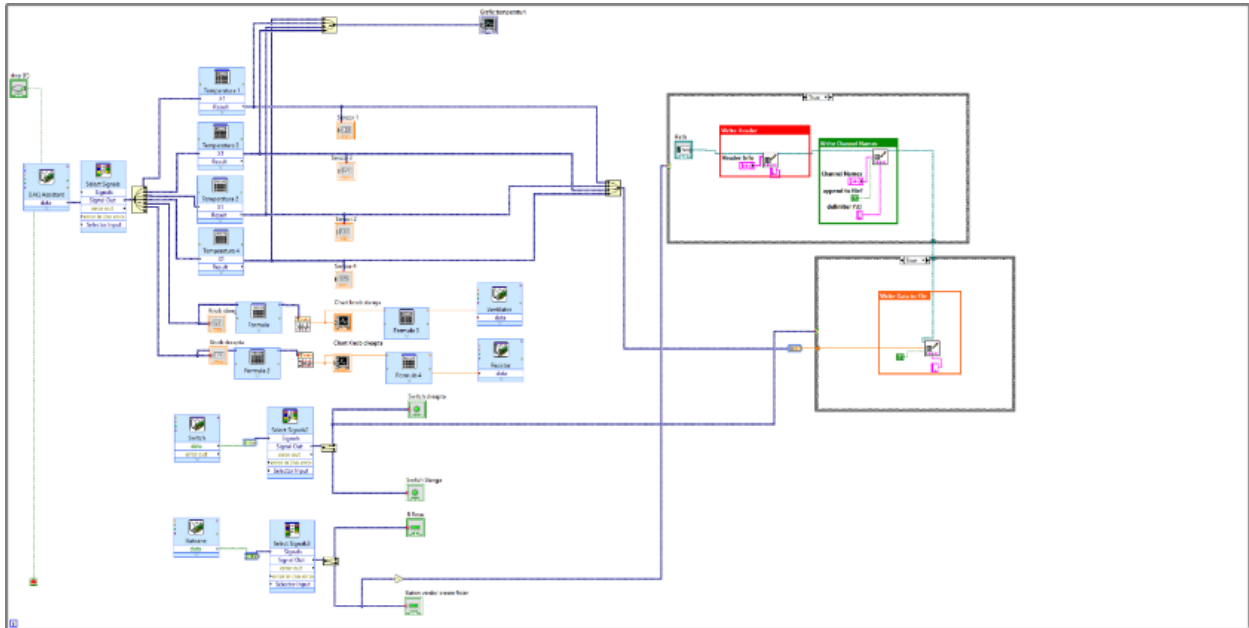


Fig. 7.1 Diagrama Labview

Selectorul de semnale oferă posibilitatea organizării și selecționării semnalelor pe care le obținem din Daq Assistant. În cazul de față rolul său este doar de organizare. Obiectul apare în mai multe locuri din diagramă, iar scopul său este același de fiecare dată.

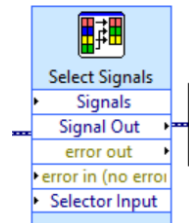


Fig. 7.2 Selectorul de semnale

Separatorul de semnale (Split Signal) se ocupă cu separarea semnalelor din firul principal pentru fiecare semnal, în cazul nostru fiind separate individual semnalele de la cei patru senzori și de la cele două potențiometre.

Formula este aplicată valorilor în volți citite de la senzori pentru a obține la final valoarea în grade Celsius. Formula utilizată în interiorul acestui obiect este oferită de producătorul senzorilor (Texas Instrument).

Pentru vizualizarea valorilor și a evoluției acestora, valorile în grade Celsius ale celor 4 senzori, au fost unite într-un singur „fir” de date și le-am introdus în graficul temperaturilor, vizibil în Panoul Frontal.

Pentru a avea acces la valorile în grade Celsius în timpul rulării programului, a fost creat câte un afișaj numeric al temperaturilor pentru fiecare dintre cei 4 senzori. Acestea sunt vizibile pe Panoul Frontal.

O altă etapă necesară în construcția diagramei a fost normalizarea semnalelor achiziționate de la potențiometre. Chiar dacă, folosind proprietățile obiectului Daq Assistant, potențimetrelor le-au fost asociate un interval de valori între 0 și 10 V, din motive constructive, în poziția de maxim semnalele depășesc valoarea de 10 V, oferind semnale de genul 10,00324V. Această depășire a valorii de 10 V are ca rezultat apariția unei erori. Pentru a corecta această problemă am fost nevoiți să găsim o metodă prin care să limităm semnalul strict între 0 și 10 V. Metoda aplicată este reprezentată de structurile de mai jos, aplicate pentru fiecare dintre cele două potențiometre:

Semnalul trece prin următoarele transformări:

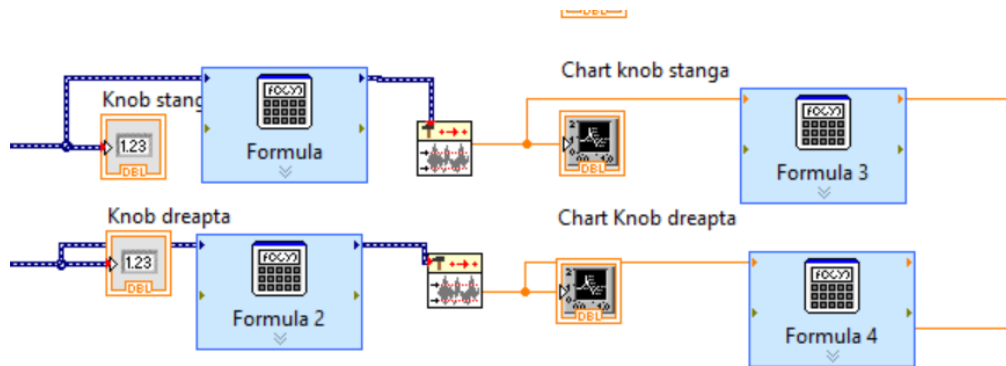


Fig. 7.3 Transformările semnalului

- este adus de la 0-10 V, la 0-1 V, prin intermediul formulelor „Formula” și „Formula 2”;
- este introdus în limitatorul de semnale, setat pentru 0-1 (valorile din interval ies neschimbate, ceea ce depășește valoarea 1, iese ca 1); acest semnal este afișat și pe graficele celor două potențiometre.
- este readus la intervalul 0-10 V folosind formulele „Formula 3” și „Formula 4”.

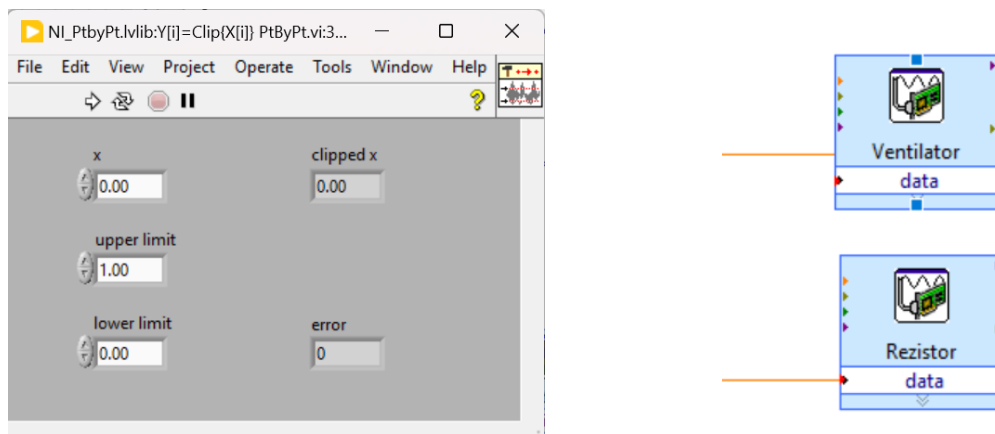


Fig. 7.4 DAQ Assistant pentru ventilator și „rezistor” (elementul de încălzire este un tranzistor)

Pentru controlul ventilatorului și al elementului de încălzire sunt necesare două obiecte Daq Assistant - unul pentru ventilator și unul pentru elementul de încălzire, setate pe ieșirile analogice (Volți), intervalul 0-10V pentru ventilator și 0-8V pentru tranzistor (pentru protejarea acestuia). Modul de generare a semnalului 1 este *sample (on demand)*. În fiecare dintre acestea va intra semnalul normalizat al unuia dintre cele două potențiometre, în diagrama Labview fiind asociate astfel: potențiometrul din stânga, comandă ventilatorul, iar cel din dreapta, comandă elementul de încălzire. A fost necesar ca semnalele digitale să fie achiziționate separat, un Daq Asistent pentru butoane și un Daq Asistent pentru Switch-uri. Acestea trec prin procese similare de selectare a semnalelor și apoi de separare pentru fiecare buton, respectiv, pentru fiecare switch.

Înainte de intrarea în selectorul de semnale, semnalul a trebuit convertit pentru a putea opera cu valorile achiziționate.

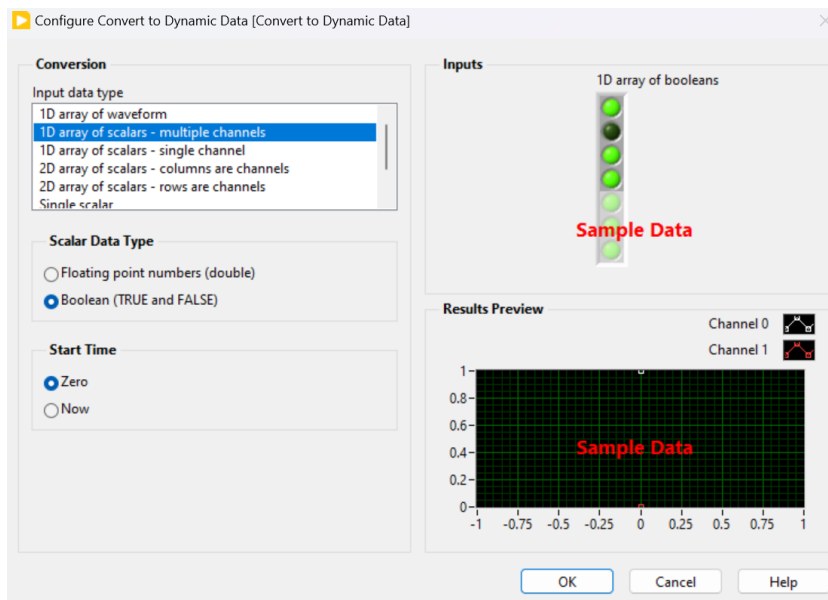


Fig. 7.5 Achiziția semnalelor digitale (butoane și switch-uri)

Butoanele sunt normal închise (semnalul butoanelor este în mod normal ADEVĂRAT), motiv pentru care am introdus o funcție logică NOT la semnalul butonului verde pentru a controla secvența de creare a documentului pentru exportarea datelor.

Switchul dreapta este folosit pentru a controla secvența de exportare a datelor în fișierul CSV. Secvența de creare a documentului de exportare a datelor CSV este reprezentată de structura „Case”, care acționează ca o structură IF. În momentul în care este introdusă valoarea ADEVĂRAT (1) la intrare, programul va crea un document CSV la locația precizată de calea (PATH). Primul lucru pe care îl va scrie în document va fi tabelul de date, prezentat în Panoul Frontal, folosind ca element de separare virgula. Al doilea lucru scris în document este tabelul de canale. Datele nu sunt suprascrise, deoarece modul de adăugare a datelor (*append to file*) a fost setat pe TRUE. În cazul în care valoarea de intrare a structurii Case este FALSE (0), nu se întâmplă nimic, dar este

necesară introducerea căii de scriere pentru continuarea programului de exportare a datelor.

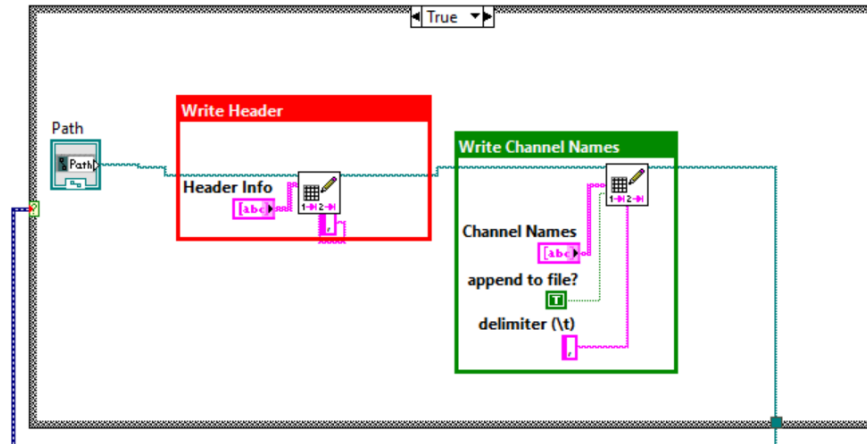


Fig. 7.6 Crearea fișierului csv

Secvența de exportare a datelor este realizată tot într-o structură CASE, care are ca scop scrierea datelor citite de la cei patru senzori pe 4 coloane, scriere care permite citirea cu ușurință a datelor de către scriptul Python care va rula modele de *machine learning*.

```
File Edit View
Senzor1, Senzor2, Senzor3, Senzor4,
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.409, 35.134
34.678, 34.654, 34.409, 35.134
34.678, 34.654, 34.409, 35.134
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.409, 35.122
34.678, 34.654, 34.409, 35.134
34.689, 34.654, 34.420, 35.122
34.678, 34.654, 34.420, 35.134
34.678, 34.654, 34.420, 35.134
34.689, 34.654, 34.420, 35.122
34.689, 34.654, 34.420, 35.122
34.678, 34.654, 34.420, 35.122
34.689, 34.643, 34.420, 35.122
34.689, 34.654, 34.420, 35.122
34.689, 34.654, 34.420, 35.122
34.678, 34.654, 34.467, 35.192
34.619, 34.643, 34.701, 35.461
34.549, 34.631, 35.005, 35.812
34.456, 34.619, 35.333, 36.198
34.385, 34.608, 35.637, 36.549
34.315, 34.608, 35.906, 36.877
34.257, 34.596, 36.163, 37.169
Ln 17, Col 28 46,877 characters 100% Windows (CRLF) UTF-8
```

Fig. 7.7 Structura fișierului csv

Condiția de rulare a structurii Case de scriere este comandată de switch-ul dreapta. Pe poziția 0, structura Case nu are nici o acțiune asociată acesteia, iar pe poziția 1 structura Case scrie în documentul creat anterior valorile pe care le primește de la cei 4 senzori. Aceste valori sunt adăugate în fișier, opțiunea de *append to file* fiind TRUE.

Au fost folosite următoarele instrumente (obiecte Labview):

- un buton “Stop (F)”, ce are ca scop întreruperea rulării programului în momentul apăsării;
- 4 afișaje numerice, „Senzor 1” , „Senzor 2” , „Senzor 3” și „Senzor 4”, pe care sunt afișate valorile numerice obținute de la cei patru senzori, în grade Celsius;
- 2 afișaje numerice, „Knob stânga” și „Knob dreapta”, pe care sunt afișate valorile numerice ale tensiunilor fiecărui potențiomtru (între 0 și 10 V);
- 2 căi, ce precizează calea spre dispozitivul de stocare, unde se creează documentul de exportare al datelor;
- 2 leduri, “Switch dreapta” și “Switch stanga”, care arată starea celor două switch-uri ale dispozitivului experimental;
- 2 leduri, “B roșu” și “Buton verde/creare fisier”, care vizualizează starea celor două butoane prezente pe dispozitivul experimental;
- 1 grafic, numit „Grafic temperaturi”, unde sunt afișate citirile de temperatură a celor 4 senzori;
- 2 diagrame, „Chart knob stanga” și „Chart knob dreapta”, pe care sunt afișate valorile fiecărui potențiomtru în parte, sub forma unei linii care variază în funcție de setarea introdusă din potențiometrele de pe dispozitivul experimental (între 0 și 10 V); sunt setate pe *auto scale*, modificându-se scala în funcție de valorile afișate;
- 1 tabel, „Header Info, care permite introducerea anumitor informații la începutul documentului exportat, în momentul creării acestuia; poate fi lăsat necompletat;
- 1 tabel, „Chanel names”, unde sunt introduse numele și numărul fiecărui element exportat, respectiv, valorile celor 4 senzori, „Senzor 1” , „Senzor 2” , „Senzor 3” și „Senzor 4”, cărora le corespunde câte o coloană de date.

D. Identificare de sistem folosind modele de regresie

În continuarea acestui studiu, s-a realizat o identificare de sistem aplicată pe datele experimentale (temperaturile în cele patru puncte ale barei), utilizând modele de regresie. Obiectivul principal a fost predicția temperaturii la capătul unei bare pe baza măsurătorilor de temperatură în trei puncte distincte ale barei. Aceste măsurători au fost achiziționate și stocate într-un fișier CSV, iar pentru analiza datelor s-au folosit, utilizând limbajul Python, diferite modele de regresie implementate în biblioteca *scikit-learn*.

Modelele de regresie au fost ales datorită simplității și eficienței lor în învățarea relațiilor (liniare sau polinomiale) dintre variabilele independente (temperaturile măsurate în cele trei puncte) și variabila dependentă (temperatura de la capătul barei). Această metodă este bine cunoscută în literatură pentru aplicabilitatea sa în diverse domenii de inginerie și în științele aplicate, asigurând, de multe ori, o bună capacitate de generalizare și interpretare a rezultatelor.

Implementarea modelului a urmat pașii standard specifici unui proces de învățare automată: preprocesarea datelor, antrenarea modelului și validarea acestuia. Datele au fost prelucrate pentru a elimina eventualele valori anormale. Modelul a fost antrenat pe un set de date reprezentativ și validat pentru a asigura acuratețea predicțiilor.

Codul complet în Python este următorul:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```



```

from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.metrics import mean_squared_error
# încărcarea datelor
data = pd.read_csv('temperaturi.csv')
# separarea datelor de antrenare de datele folosite ca predicție
X = data[['temp1', 'temp2', 'temp3']]
y = data['temp4']
# împărțirea datelor în seturi de antrenament și testare
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# definirea modelelor
model_linear = LinearRegression()
model_logistic = LogisticRegression()
model_polynomial = make_pipeline(PolynomialFeatures(degree=2),
LinearRegression())
# antrenarea modelelor
model_linear.fit(X_train, y_train)
model_logistic.fit(X_train, y_train)
model_polynomial.fit(X_train, y_train)
# evaluarea modelelor
y_pred_linear = model_linear.predict(X_test)
y_pred_logistic = model_logistic.predict(X_test)
y_pred_polynomial = model_polynomial.predict(X_test)
mse_linear = mean_squared_error(y_test, y_pred_linear)
mse_logistic = mean_squared_error(y_test, y_pred_logistic)
mse_polynomial = mean_squared_error(y_test, y_pred_polynomial)
print(f'MSE pentru regresia liniară: {mse_linear}')
print(f'MSE pentru regresia logistică: {mse_logistic}')
print(f'MSE pentru regresia polinomială: {mse_polynomial}')
# testarea modelelor cu date noi
new_data = np.array([[63, 62, 62.5]])
new_pred_linear = model_linear.predict(new_data)
new_pred_logistic = model_logistic.predict(new_data)
new_pred_polynomial = model_polynomial.predict(new_data)
print(f'Predicția pentru regresia liniară cu date noi: {new_pred_linear[0]}')
print(f'Predicția pentru regresia logistică cu date noi: {new_pred_logistic[0]}')
print(f'Predicția pentru regresia polinomială cu date noi: {new_pred_polynomial[0]}')

```

Inițial sunt importate bibliotecile esențiale pentru manipularea și prelucrarea datelor (numpy, pandas), precum și pentru construirea și evaluarea modelelor de regresie (scikit-learn). Datele experimentale sunt citite dintr-un fișier CSV utilizând biblioteca pandas, iar coloanele reprezentând temperaturile măsurate în cele trei puncte ale barei (temp1, temp2, temp3) sunt separate de coloana de predicție (temp4), care reprezintă temperatura la capătul barei.

Setul de date este împărțit în două subseturi: unul pentru antrenament (train) și unul pentru testare (test), utilizând funcția `train_test_split()` din `scikit-learn`. Această împărțire permite evaluarea performanței modelelor pe date neutilizate în procesul de antrenament, asigurând o evaluare obiectivă.

Au fost definite trei modele de regresie:

- regresia liniară (`LinearRegression`);
- regresia logistică (`LogisticRegression`);
- regresia polinomială (`PolynomialFeatures`).

Fiecare model a fost antrenat pe setul de date de antrenament (`X_train`, `y_train`). Modelele au fost evaluate pe setul de testare (`X_test`, `y_test`) utilizând eroarea pătratică medie (MSE - `mean_squared_error`). MSE este o metrică standard în evaluarea modelelor de regresie, oferind o măsură a diferenței dintre valorile prezise și cele reale.

Pentru a testa capacitatea modelelor de a generaliza au fost realizate predicții pentru un nou set de date (variabila `new_data`). Aceste predicții sunt în final afișate, oferind o imagine de ansamblu asupra performanței fiecărui model pe date noi de intrare.

E.Rezultate și concluzii

În urma analizei și evaluării modelelor de regresie, se pot formula următoarele concluzii:

1. Împărțirea setului de date în subseturi pentru antrenament și testare este esențială pentru a asigura o evaluare obiectivă a performanței modelelor. Aceasta permite observarea capacității modelelor de a generaliza pe date noi și neutilizate în procesul de antrenament;

2. Performanța modelelor de regresie:

- modelul de regresie liniară a avut un MSE de 0.0036847780951287656. Deși are o eroare relativ mică, nu a fost cea mai performantă dintre modelele testate.

- modelul de regresie logistică a obținut un MSE de 0.003791139781560431, similar cu regresia liniară, însă puțin mai ridicat, indicând o performanță ușor inferioară în această sarcină specifică;

- modelul de regresie polinomială a demonstrat cea mai bună performanță, cu un MSE de 0.00017238861674443757. Aceasta indică faptul că modelul polinomial a reușit să capteze mai bine relațiile complexe dintre variabile, oferind predicții mai precise.

3. Generalizarea modelelor:

- Modelele au fost testate și pe un nou set de date (`new_data` în codul Python) pentru a testa capacitatea lor de a face predicții în afara datelor de antrenament. Rezultatele predicțiilor temperaturii sunt (oferim aici doar un singur exemplu):

- regresia liniară: 60.4092284174624

- regresia logistică: 61.2997528781232

- regresia polinomială: 62.03354299752878

Deși predicțiile diferă ușor, modelul de regresie polinomială a obținut o performanță mai bună și pe datele noi.

În contextul specific al predicției temperaturii la capătul unei bare pe baza temperaturilor măsurate în alte trei puncte, modelul de regresie polinomială, comparativ cu celelalte modele testate, s-a dovedit a fi cel mai adecvat. În timp, odată cu creșterea volumului de date achiziționate și reantrenarea modelelor, există posibilitatea ca acuratețea predicțiilor să crească.

BIBLIOGRAFIE

1. Documentație dispozitiv de achiziție multifuncții National Instruments – USB6001: <https://www.ni.com/ro-ro/support/model.usb-6001.html>;
2. Documentație circuit integrat de tip amplificator operațional LM358 – Texas Instruments: <https://www.ti.com/lit/ds/snosbt3j/snosbt3j.pdf?ts=1663618756434>;
3. Documentație tranzistor de putere medie BD139 – On Semiconductor: <https://www.onsemi.com/pdf/datasheet/bd139-d.pdf>;
4. Documentație tranzistor de putere MJE13007 – On Semiconductor: <https://www.onsemi.com/pdf/datasheet/mje13007-d.pdf>;
5. Documentație circuit integrat comparator de tip LM393N – Texas Instruments: <https://www.ti.com/lit/ds/symlink/lm393-n.pdf?ts=1663627853212>;
6. Documentație senzor de temperatura analogic de tip LM35 – Texas Instruments: <https://www.ti.com/lit/ds/symlink/lm35.pdf>;
7. Anicuța Stoica, Marta Stroescu, Tănase Dobre, *Probleme de transfer termic cu aplicații în industria alimentară*, Electra, București, 2015;
8. John Essick, *Hands-On Introduction to LabVIEW for Scientists and Engineers*, Oxford University Press, 2019;
9. Perry Xiao, *Artificial Intelligence Programming with Python*, Wiley, 2022;
10. Kevin Murphy, *Probabilistic Machine Learning*, MIT Press, 2022;
11. Aurelien Geron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly, 2019.