

Introducere în Matlab

Prof. Vasilescu Gheorghe – Colegiul Tehnic Metalurgic Slatina

Matlab-ul este un limbaj de nivel foarte înalt care prezintă performanțe deosebite în ceea ce privește calculul tehnic (Matlab reprezintă o prescurtare a cuvintelor “Matrix laboratory”). Pe lângă interpretorul de comenzi sunt prezente instrumente cum ar fi cele pentru vizualizarea datelor, prelucrarea imaginilor și sunetelor, analiza circuitelor electrice, etc.

Spre deosebire de alte limbaje, elementele de bază cu care se lucrează sunt vectorii. Utilizatorul poate defini și folosi vectori ai căror dimensiuni inițiale nu trebuie specificate. Problemele legate de gestionarea memoriei la operațiile care presupun creșterea dimensiunii unui vector se fac automat, transparent utilizatorului, ceea ce este un avantaj față de limbaje de programare cum ar fi C sau C++. Funcțiile specifice unui anumit domeniu sunt grupate în colecții de funcții sau “toolboxes”. Acestea ușurează foarte mult folosirea programului în scop educational sau de cercetare deoarece utilizatorul se poate concentra direct pe aplicarea unei serii de operații asupra unui set de date fără a se îngriji exclusiv de definirea acestor operații. Există mai multe colecții de funcții specifice domeniului electronicii cum ar fi cea pentru prelucrarea de semnale sau pentru domeniul comunicațiilor. Fereastra principală a programului permite accesul direct la interpretorul de comenzi. Acesta este un instrument care execută o secvență de cod, linie cu linie. Secvența de cod poate fi introdusă direct de la tastatură, iar după fiecare linie se apasă tasta Enter sau poate fi scrisă într-un fișier de tip text, care se salvează cu extensia “.M” și se execută prin simpla scriere a numelui fișierului. Limbajul Matlab respectă principiile programării structurale, astfel că există o foarte mare asemănare între sintaxa și structurile sale cu cea a limbajului C.

Notiuni despre Simulink

Simulink este un pachet de programe pentru modelarea, simularea și analiza sistemelor dinamice. Pot fi simulate atât sisteme liniare cât și neliniare, modelate în timp continuu, discret sau într-o combinație a celor două. Sistemele pot avea porțiuni esanționate cu frecvențe de esanționare diferite.

Pentru modelarea de sistem este furnizată o interfață grafică intuitivă. Blocurile sunt plasate și interconectate cu ajutorul mouse-ului ceea ce reprezintă un mare avantaj (față de scrierea directă a ecuațiilor diferențiale ce definesc un sistem). Simulink oferă o colecție de blocuri cum ar fi: generatoare de semnal, instrumente de vizualizare, blocuri care realizează funcții matematice, componente liniare și neliniare, etc. Setul de blocuri furnizat poate fi extins oricând cu noi blocuri – este furnizată documentație completă despre felul cum se poate crea un nou bloc.

Mai multe blocuri pot fi grupate oricând într-un bloc nou, oferind astfel posibilități extinse de analiză la un nivel superior de organizare. După definirea unui model nou, simularea se poate efectua atât în mod grafic cât și cu ajutorul interpretorului. Cele două instrumente sunt legate între ele și astfel se poate opta pentru orice modalitate de analiză.

Pentru lansarea programului Simulink se tastează în mediul Matlab comanda:
‘simulink’.

Fereastra care se deschide conține toate blocurile disponibile grupate pe categorii (figura 1.1 -pentru Matlab 6.0 sub sistemul de operare Windows). Dacă se deschide biblioteca “Simulink” se observă următoarele zone:

Continuous - blocuri ce furnizează funcții specifice circuitelor analogice: derivare, integrare, funcție de transfer, întârziere în domeniul timp, etc.;

Discrete - blocuri ce furnizează funcții specifice circuitelor discrete: funcție de transfer discretă, filtru discret, întârziere în domeniul timp cu un pas, integrator discret, etc.;

Function & Tables - blocuri care permit extinderea setului de blocuri existent cu blocuri create de utilizator;

Math Operations - funcții matematice de ordin general: sumă, produs, modul, amplificarea, fază, funcții trigonometrice, etc.;

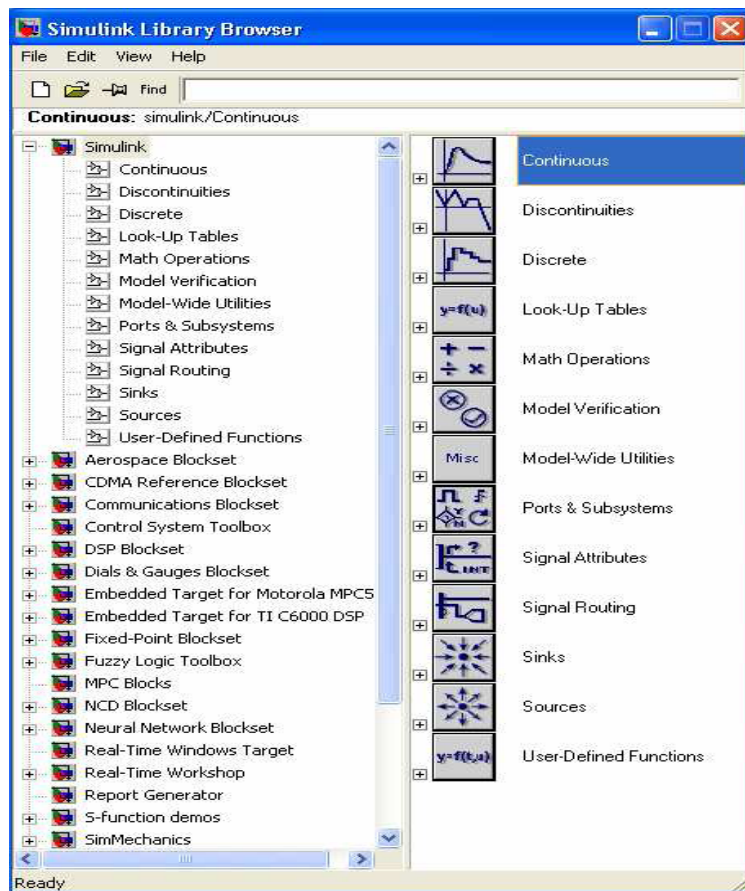
Nonlinear - funcții specifice circuitelor neliniare;

Signal Routing - blocuri necesare pentru definirea semnalelor: masă, multiplexor de mai multe semnale, funcții pentru preluarea și salvarea valorilor în spațiul Matlab, etc.;

Sinks - aparate de măsură: multimetru, osciloscop, graphic XY, etc.;

Sources - surse de semnal: generatoare de semnal sinusoidal, triunghiular, dreptunghiular, zgomot, rampă, pulsuri, etc. Se poate obține o descriere detaliată a fiecărui bloc dacă se selectează și se urmărește în partea de sus a ferestrei explicațiile aferente. O structurare asemănătoare se întâlnește în cadrul fiecărui modul. Celelalte module introduc blocuri noi care realizează funcții complexe prin combinarea blocurilor elementare descrise anterior.

Pentru plasarea unui bloc nou în schemă, acesta este selectat în fereastra bibliotecilor și se poziționează în schemă folosind “drag and drop”.



Biblioteca

Coduri Hamming

Codurile Hamming reprezintă o clasă aparte a codurilor bloc corectoare de erori.

Aceste coduri și variantele derivate din ele sunt larg utilizate pentru corecția erorilor în

comunicații digitale și în sistemele de stocare a datelor. Pentru orice întreg pozitiv $m \geq 3$, există un cod Hamming având următorii parametri:

- lungimea codului: $n = 2^m - 1$;
- numărul simbolurilor de informație: $k = 2^m - m - 1$;
- numărul simbolurilor de control: $n - k = m$;
- capacitatea de corecție: $t = 1$;

Matricea H are următoarea formă:

$$H = [I \ Q]_m$$

unde I este matricea unitate de ordinul m .

Matricea generatoare a codului este:

$$G = [I \ Q^T]_k$$

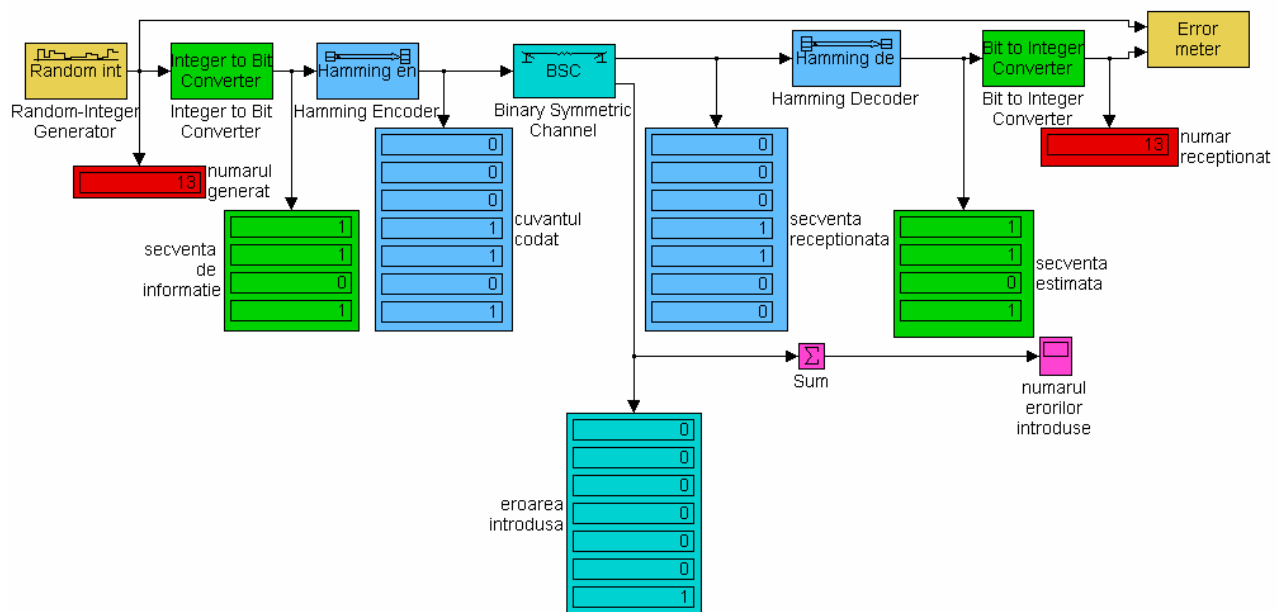
$$rG = Q^T I$$

unde I este matricea unitate de ordinul k .

Aplicație coduri Hamming

Codurile Hamming fac parte din categoria codurilor bloc. Aceste coduri sunt coduri perfecte, corectoare de o eroare, corectează toate structurile de erori simple dar nici o combinație de erori duble, și detectează toate structurile de eroare cu două sau mai puține erori.

Pentru a arăta utilitatea codului Hamming utilizăm următorul model:



Folosim urmatoarele blocuri:

- **Random-Integer Generator**: generează numere întregi distribuite în intervalul $[0, M-1]$. Parametrii blocului sunt:
 - 'M-ary number' este 2^4 deoarece codul Hamming utilizat este **C(7,4)** si numerele generate sunt reprezentate în binar pe 4 biTi.
 - 'Initial seed' este [12345]. Modificând acest parametru se modifică secvența de numere generate.
 - 'Sample time' este 1. Generează câte un număr la fiecare secundă.
- **Integer to Bit Converter**: transformă un vector de întregi într-un vector de biTi.

Parametrul blocului este:

- 'Number of bits per integer' este 4.
- **Hamming Encoder**: crează un cod Hamming din datele vectorului binar. Parametrii blocului sunt:
 - 'Codeword length N' este 7 .
 - 'Message length K' este 4 deoarece se utilizează codul C(7,4).
- **Binary Symmetric Channel**: introduce erori binare. Parametrii blocului sunt:
 - 'Error probability' este 0.29, pentru a introduce o singură eroare.
 - 'Input vector length' este 7 deoarece cuvântul de cod cu care se adună este reprezentat pe 7 biti.
 - 'Initial seed' este 1234.
 - 'Sample time' este 1 pentru a se genera un esantion la fiecare secundă.
- **Hamming Decoder**: decodează un cod Hamming pentru a reface vectorul binar transmis.

Parametrii blocului sunt:

- 'Codeword length N' este 7 .
 - 'Message length K' este 4 deoarece se utilizează codul C(7,4)

Pentru ca **Random-Integer Generator** să genereze un singur simbol am stabilit timpul de simulare ca fiind de la '0' la '0.5' deoarece primul simbol se generează la '0' si al doilea la '1'.

Pentru a genera mai multe simboluri timpul de simulare este stabilit de la 0 la 19 generându-se astfel 20 de numere. Numărul erorilor introduse pentru cele 20 de numere sunt reprezentate pe diagrama blocului **Scope**.