Lucrarea 2

.1 Simulink

Simulink este un mediu pentru modelarea, simularea şi implementarea sistemelor dinamice şi sistemelor înglobate. Permite simularea numerică şi analiza sistemelor liniare, neliniare, discrete, continue, hibride prin dezvoltarea şi manipularea de blocuri şi diagrame. Oferă facilităţi pentru design bazat pe modelare, inclusiv modelare în domeniul fizicii, generare automată a codului, verificare şi validare. Are o arhitectură deschisă facilitând integrarea modelelor provenind de la alte unelte. Este utilizat în aplicaţii de control, procesare de semnale, comunicaţii şi alte domenii ale ingineriei de sistem.

Simulink oferă o interfaţă grafică cu utilizatorul, biblioteci cu blocuri personalizabile pentru crearea şi editarea diagramelor (modelelor). Aceste diagrame bloc (modele *mdl*) reprezintă sistemul modelat sub forma ecuaţiilor diferenţiale liniare şi neliniare şi a ecuaţiilor cu diferenţe finite care descriu dinamica sistemului.

Este integrat în Matlab, permiţând încorporarea algoritmilor Matlab în modele şi exportarea rezultatelor simulării în Matlab pentru analiza ulterioară.

Algoritmul general pentru crearea unui model Simulink pentru un sistem:

- găsirea modelului analitic al sistemului. Acesta poate consta într-o varietate de tipuri de ecuaţii, împreună cu condiţiile lor iniţiale şi de limită, precum ecuaţii algebrice simple sau ecuaţii diferenţiale mai complicate (de ordin mare);

- identificarea variabilelor de stare şi a variabilelor de control. De exemplu, în cazul unei simple translaţii liniare, variabilele de stare sunt viteza şi poziţia, iar variabila de control este forţa aplicată. Pentru evitarea redundanţei este de dorit ca aceste variabile să fie independente;

- identificarea blocurilor Simulink necesare pentru implementarea modelului pornind de la ecuaţiile găsite;
- implementarea modelului luând în considerare constrângerile externe dorite, condiţiile iniţiale şi condiţiile de limită.

Pentru rulare, se tastează în linia de comandă >> *simulink* urmată de apăsarea tastei *Enter* sau se apasă butonul *Simulink* din interfaţa Matlab. Pentru rularea programelor demonstrative, este disponibilă comanda >> *demo simulink*.

.1.1 Blocuri Simulink

Simulink oferă o mare varietate de blocuri pentru construirea modelelor matematice. O serie de blocuri des utilizate în continuare sunt prezentate în Figura .1 şi descrise pe scurt în cele ce urmează.

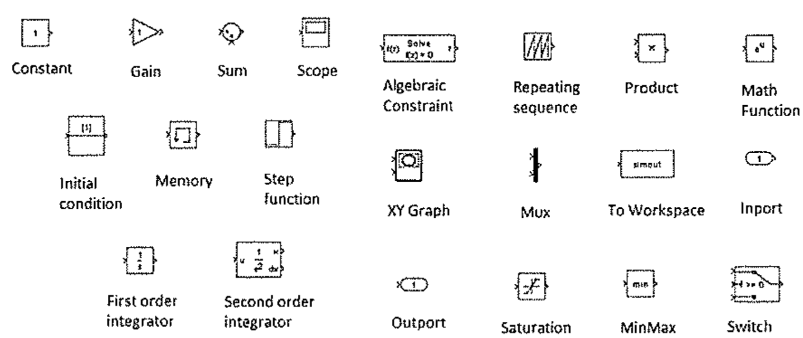

Figure .1: Blocuri Simulink

- Blocul ***Constant***, aflat în biblioteca ***Sources***, generează o valoare constantă reală sau complexă;
- Blocul ***Summation***, aflat în biblioteca ***Math Operations***, primeşte ca intrare două sau mai multe semnale, ex., x_1, x_2, x_3. Modificând parametrii blocului, ieşirea obţinută poate consta în combinaţia necesară de adunări şi scăderi precum $x_1 + x_2 + x_3$, $-x_1 + x_2 + x_3$ etc. Este util pentru ***adunări*** şi ***scăderi***;

- Blocul ***Gain***, aflat în biblioteca ***Math Operations***, primeşte la intrare un semnal, ex., x_1 şi returnează o versiune scalată a acestui semnal, $k \cdot x_1$. Se utilizează pentru ***înmulţire***;

- Blocul ***Scope***, aflat în biblioteca ***Sinks***, afişează semnalele de intrare în raport cu timpul de simulare;

- Blocul ***Initial Condition***, aflat în biblioteca ***Signal Attributes***, setează condiţia iniţială a semnalului de intrare, spre exemplu, valoarea semnalului la începutul simulării. La începutul simulării, indiferent de valoarea semnalului, blocul va furniza la ieşire condiţia iniţială specificată, urmând ca ulterior să returneze valoarea semnalului;

- Blocul ***Memory***, aflat în biblioteca ***Signal Attributes***, reţine şi întârzie intrarea cu un pas de timp de integrare

- Blocul ***Step Function***, aflat în biblioteca ***Sources***, generează o treaptă între două nivele de amplitudine configurabile, la un moment specificat. Ieşirea blocului constă în valoarea parametrului *Initial Step* dacă timpul de simulare este mai mic decât valoarea parametrului *Step time*, respectiv în valoarea parametrului *Final Step* dacă timpul de simulare este mai mare sau egal cu valoarea parametrului *Step time*;

- Blocul ***First Order Integrator***, aflat în biblioteca ***Continuous***, generează la ieşire integrala intrării la pasul de timp curent;

- Blocul ***Second Order Integrator***, aflat în biblioteca ***Continuous***, rezolvă probleme cu valori iniţiale pentru ecuaţii de ordinul doi:

$$\frac{d^2x}{dt^2} = u,$$
$$\frac{dx}{dt}|_{t=0} = dx_0,$$
$$x|_{t=0} = x_0,$$

  unde u este intrarea sistemului. Blocul este astfel un sistem dinamic cu două stări continue: x şi dx/dt;

- Blocul ***Repeating Sequence***, aflat în biblioteca ***Sources***, generează semnale periodice cu o formă de undă specificată prin intermediul parametrilor *Time values* (specifică vectorul timpilor de ieşire, perioada este dată de diferenţa dintre prima şi ultima valoare a parametrului) şi *Output values* (specifică un vector cu amplitudini ale semnalului la timpii de ieşire corespunzători). Implicit, ambii parametrii sunt [0 2], specificând un semnal dinte de fierăstrău cu perioada de două secunde şi amplitudine maximă 2;

- Blocul ***Product***, aflat în biblioteca ***Math Operations***, în mod implicit, generează la ieşire rezultatul înmulţirii a două intrări: doi scalari, un scalar şi o valoare care nu e scalară sau două valori care nu sunt scalare şi au aceleaşi dimensiuni. Poate fi configurat să funcţioneze ca un block ***Divide*** sau ***Product of Elements***;

- Blocul ***Math Function***, aflat în biblioteca ***Math Operations*** realizează numeroase funcţii matematice comune;

- Blocul ***XY Graph***, aflat în biblioteca ***Sinks***, afişează un grafic $X-Y$ al semnalelor de intrare într-o figură Matlab;

- Blocul ***Mux***, aflat în biblioteca ***Signal Routing*** combină semnalele de intrare (scalari sau vectori) într-un singur vector de ieşire. Toate intrările trebuie să fie numerice şi de acelaşi tip de date. Elementele vectorului de ieşire sunt ordonate conform semnalelor de intrare de sus în jos sau de la stânga la dreapta;

- Blocul ***To Workspace*** aflat în biblioteca ***Sinks*** primeşte la intrare un semnal şi exportă valorile semnalului în spaţiul de lucru al Matlabului;

- Blocul ***Inport (In1)***, aflat în biblioteca ***Sources***, creează un port de intrare pentru un subsistem sau o intrare externă;

- Blocul ***Outport (Out1)***, aflat în biblioteca ***Sinks***, creează un port de ieşire pentru un subsistem sau o ieşire externă;

- Blocul ***Saturation***, aflat în biblioteca ***Discontinuities***, impune limite superioare sau inferioare unui semnal de intrare;

- Blocul ***MinMax***, aflat în biblioteca ***Math Operations***, generează, conform funcţiei selectate, minimul sau maximul elementului sau elementelor de intrare;

- Blocul ***Switch***, aflat în biblioteca ***Signal Routing***, lasă să treacă spre ieşire prima sau a treia intrare (intrări de date) conform valorii celei de a doua intrări (intrare de control). Permite specificarea condiţiei de mapare a primei intrări la ieşire prin intermediul parametrilor *Criteria for passing first input* şi *Threshold.*

.1.2 Bila în cădere

Modelul unei bile în cădere (Fig. .2) poate fi reprezentat ca un sistem care implică atât dinamici continue cât şi tranziţii discrete, cu o singură stare discretă şi două stări continue. Dinamica continuă a unei bile în cădere este dată de:

$$\frac{dv}{dt} = -g, \qquad \frac{dx}{dt} = v, \tag{.1}$$

unde g acceleraţia gravitaţională, $x(t)$ este poziţia şi $v(t)$ este viteza.

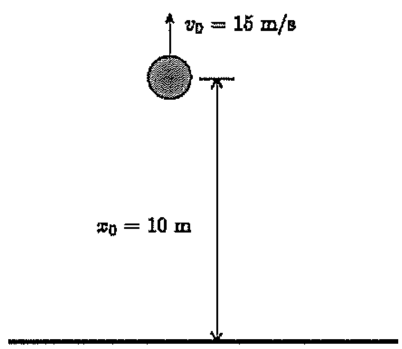

Figure .2: Starea iniţială pentru modelul bilei în cădere

Presupunând o coliziune parţial elastică cu pământul, relaţia dintre viteza înainte de coliziune (v^-) şi viteza după coliziune (v^+) poate fi scrisă sub forma:

$$v^+ = -\kappa v^-, \qquad x = 0 \tag{.2}$$

Prin urmare modelul prezintă un salt al valorii variabilei de stare continue (viteza) la condiţia de tranziţie, $x = 0$.

Pentru modelarea bilei în cădere în Simulink, se identifică variabilele de stare ca fiind poziţia şi viteza. Ecuaţiile care trebuie implementate sunt

$$\frac{dv}{dt} = -g, \qquad \frac{dx}{dt} = v, \tag{.3}$$

cu constrângerile

$$v^+ = -\kappa v^-, \qquad \text{at } x = 0, \tag{.4}$$

şi condiţiile iniţiale $x(t = 0) = 10m$, $v(t = 0) = 15m/s$.

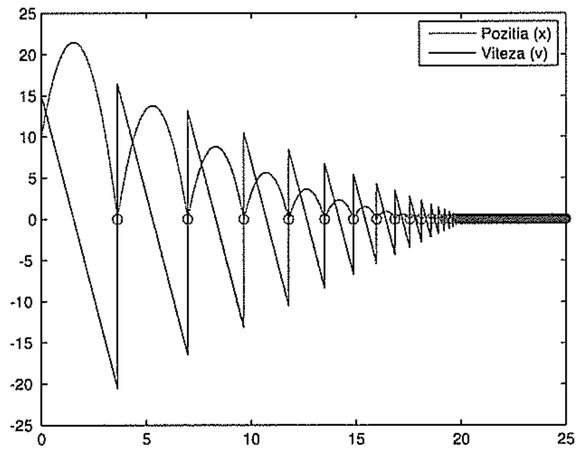

Figure .3: Variaţia în timp a poziţiei şi vitezei pentru bila în cădere

Modelul *mdl* asociat cu acest exemplu se află între aplicaţiile demostrative ale Simulink, putând fi deschis scriind în linia de comandă >> *sldemo_bounce*.

Blocurile utilizate în acest model sunt:

- *Constant*, pentru acceleraţia gravitaţională;
- *Initial Condition*, pentru setarea poziţiei şi a vitezei iniţiale;
- *Gain*, pentru coeficientul de atenuare κ.
- *Memory*, utilizat în bucla pentru calculul vitezei după coliziunea cu pământul şi are rolul de a reţine viteza bilei v^- dinaintea coliziunii.
- *Second order integrator*, poate fi utilizat ca alternativă la modelarea ecuaţiei cu ajutorul a două blocuri integratoare de ordinul unu. Cea de a doua ecuaţie $dx/dt = v$ este internă blocului. Analizând proprietăţile blocului se va observa că x are setată limita de jos la zero, şi este bifată opţiunea *Reinitialize dx/dt when x reaches saturation*, parametru care

permite reinițializarea dx/dt la o nouă valoare în momentul în care x ajunge la limita de saturație. În cazul modelului bilei în cădere, această opțiune implică faptul că în momentul în care bila atinge pământul, viteza ei se poate seta la o valoare diferită, și anume la valoarea de după impact.

Exerciții: Simulați modelul și încercați să înțelegeți funcționalitatea fiecărui bloc.
Reconstruiți modelul de la început. (Pentru crearea unui model nou se poate utiliza opțiunea din meniul *Open*, *New* → *Model*).

.1.3 Pendulul simplu

Figura .4 prezintă un pendul simplu (punctul de masă m) suspendat de un fir de lungime l neextensibil și fără greutate.

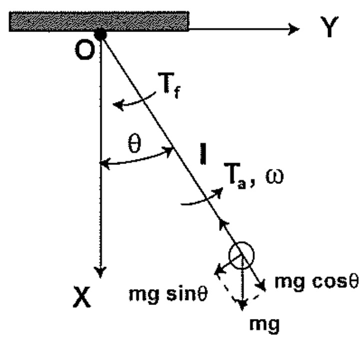

Figure .4: Pendulul simplu

Derivarea ecuațiilor de mișcare presupunând că forța de frecare este o funcție liniară de viteza unghiulară ($T_f = B_m\omega$), B_m este coeficientul de frecare vâscoasă):

Forța care tinde sa aduca pendulul spre pozitia verticala este:

$$F_{rest} = -mg\sin\theta \qquad (.5)$$

Momentul net în jurul punctului de fixare O este:

$$\sum M = T_{rest} + T_a - T_f = -mgl\sin(\theta) + T_a - B_m\omega \qquad (.6)$$

unde T_a, este cuplul aplicat.

Ecuaţiile diferenţiale de ordinul doi pentru modelarea mişcării pendulului

$$J\alpha = J\frac{d^2\theta}{dt^2} = -mgl\sin\theta + T_a - B_m\omega \quad \text{, sau}$$
$$\frac{d^2\theta}{dt^2} = \frac{1}{J}(-mgl\sin\theta + T_a - B_m\omega) \tag{.7}$$

unde J este momentul de inerţie al masei în jurul punctului O.

Ţinând cont că $\frac{d\theta}{dt} = \omega$, sistemul de două ecuaţii diferenţiale devine:

$$\frac{d\omega}{dt} = \frac{1}{J}(-mgl\sin\theta + T_a - B_m\omega) \quad ,$$
$$\frac{d\theta}{dt} = \omega \tag{.8}$$

Momentul de inerţie este $J = ml^2$, prin urmare ecuaţiile pot fi scrise sub forma:

$$\frac{d\omega}{dt} = -\frac{B_m}{ml^2}\omega - \frac{g}{l}\sin\theta + \frac{1}{ml^2}T_a \quad ,$$
$$\frac{d\theta}{dt} = \omega \tag{.9}$$

Exerciţii: Simulaţi modelul cu următorii parametri $l = 4$, $g = 9.81$, $B_m = 0.8$, $m = 1$, pasul maxim pentru simularea numerică 0.1, cuplul aplicat reprezentat printr-un semnal dreptunghiular cu amplitudine 1 şi frecvenţa de 0.1 rad/sec.