



Universitatea
Transilvania
din Brașov

UNIVERSITATEA TRANSILVANIA DIN BRAȘOV

Departamentul de Automatică și tehnologia informației
Disciplina Sisteme de conducere în Robotică



PROIECT DE AN LA DISCIPLINA Sisteme de conducere în Robotică

Sistem Robotic cu Două Cuple de Translație

Autor: Student Andrei-Constantin BORICEAN
Programul de studii: Robotică
Grupa 4LF801A

Coordonator: Prof. univ. dr. ing. Claudiu POZNA

2022



CUPRINS

1. Introducere.....	3
a. Definiţie Roboţi.....	3
b. Introducerea în temă.....	3
c. Structura controlului.....	3
2. Structura robotului. Schema structurală.....	5
3. Modelul Geometric.....	6
a. Model Geometric Direct.....	6
b. Ecuatiile Jacobianului.....	8
c. Model Geometric Invers.....	10
4. Modelul Cinematic.....	11
a. Model Cinematic Direct.....	11
b. Model Cinematic Invers.....	14
5. Modelul Dinamic.....	15
6. Simulare Virtuală.....	18
a. Alegerea Motoarelor.....	18
b. Prototip Virtual 1.....	21
c. Prototip Virtual 2.....	23
7. Concluzii.....	30
8. Anexe.....	31
9. Bibliografie.....	33

1. Introducere

a. Definiţie Roboţi

Robotul poate fi definit ca un sistem sau un echipament cu funcţionare automată, adaptabilă prin programarea condiţiilor unui mediu complex şi variabil în care acţionează, înlocuind sau amplificând una sau mai multe din funcţiunile umane în acţiunea acestuia asupra mediului.

b. Introducerea în temă

Robotul cu post fix, manipulatorul, este un etalon al produselor mecatronice atât prin complexitatea sa cât şi prin succesul cu care a fost integrat în procesele industriale şi din ce în ce mai mult în sfera serviciilor. Astfel, pentru automatişti, studiul roboţilor devine un studiu de caz al modelării şi conducerii unui sistem complex, iar pentru roboticieni fundamentul domeniului în care se pregătesc.

Sinteza controlerului se realizează cu strategii care au în vedere exclusiv controlul motorului de curent continuu, apoi cele care se referă la modelul cinematic al structurii iar în final sunt prezentate acele strategii care utilizează modelul dinamic.

c. Structura controlului

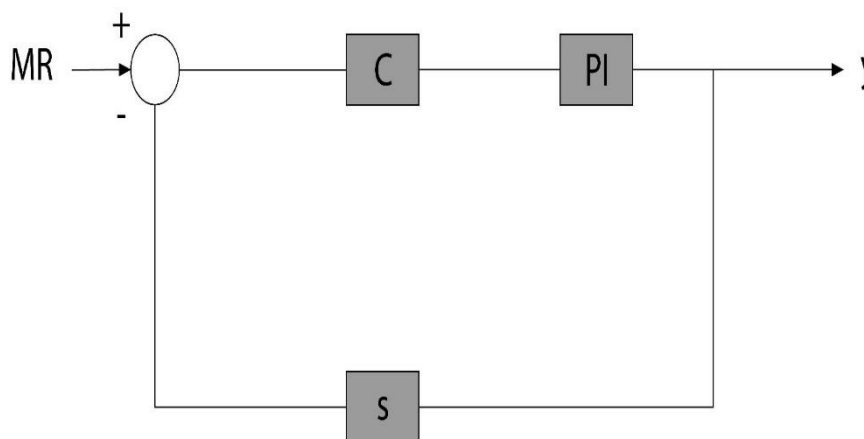


Fig.1 Schemă bloc a controlului unui robot



În figura precedentă se prezintă schema bloc pentru controlul unui robot. Pentru acest proiect se foloseşte un sistem neliniar de tip MIMO. Elementele care se regăsesc în schemă sunt următoarele:

- MR – Mărimi de referinţă
- C – Controller
- S – Senzor
- Pl – Plant (MG, MC, MD)
 - MG - Model Geometric
 - MC – Model Cinematic
 - MD – Model Dinamic
- y – Cuplu către motor

Modelul Geometric este împărţit în două:

- Model Geometric Direct** – se cunoaşte lungimea cuplelor robotului şi trebuie aflată poziţia punctului final;
- Model Geometric Invers** – se cunoaşte poziţia punctului final şi trebuie aflată lungimea cuplelor robotului;

Modelul Cinematic este împărţit în două:

- Model Cinematic Direct** – se cunoaşte lungimea cuplelor robotului şi trebuie aflată poziţia punctului final luând în considerare şi timpul;
- Model Cinematic Invers** – se cunoaşte poziţia punctului final şi trebuie aflată lungimea cuplelor robotului luând în considerare şi timpul;

Modelul Dinamic este împărţit în trei:

- Model Dinamic Direct** – se cunoaşte vectorul tensiunilor de alimentare şi trebuie aflate poziţia, viteza şi acceleraţia cuplelor şi a efectorului;
- Model Dinamic Invers** – se cunosc poziţia, viteza şi acceleraţia cuplelor şi a efectorului şi trebuie aflat vectorul tensiunilor de alimentare;
- Model Dinamic Mixt** – se cunoaşte doar o parte din mărimile menţionate anterior, iar pentru aflarea celorlalte părţi rezultă o combinaţie de ecuaţii diferenţiale şi algebrice;

2. Structura robotului. Schema structurală.

Pentru acest proiect se va prezenta un sistem robotic având două cuple de translaţie precum şi controlul acestuia. Scopul principal este de a se deplasa până la orice punct dat în limita spaţiului de lucru definit ca fiind un pătrat cu latura egală cu 1m. Pentru realizarea acestuia s-au folosit în principiu soft-urile Matlab + Simulink şi CatiaV5.

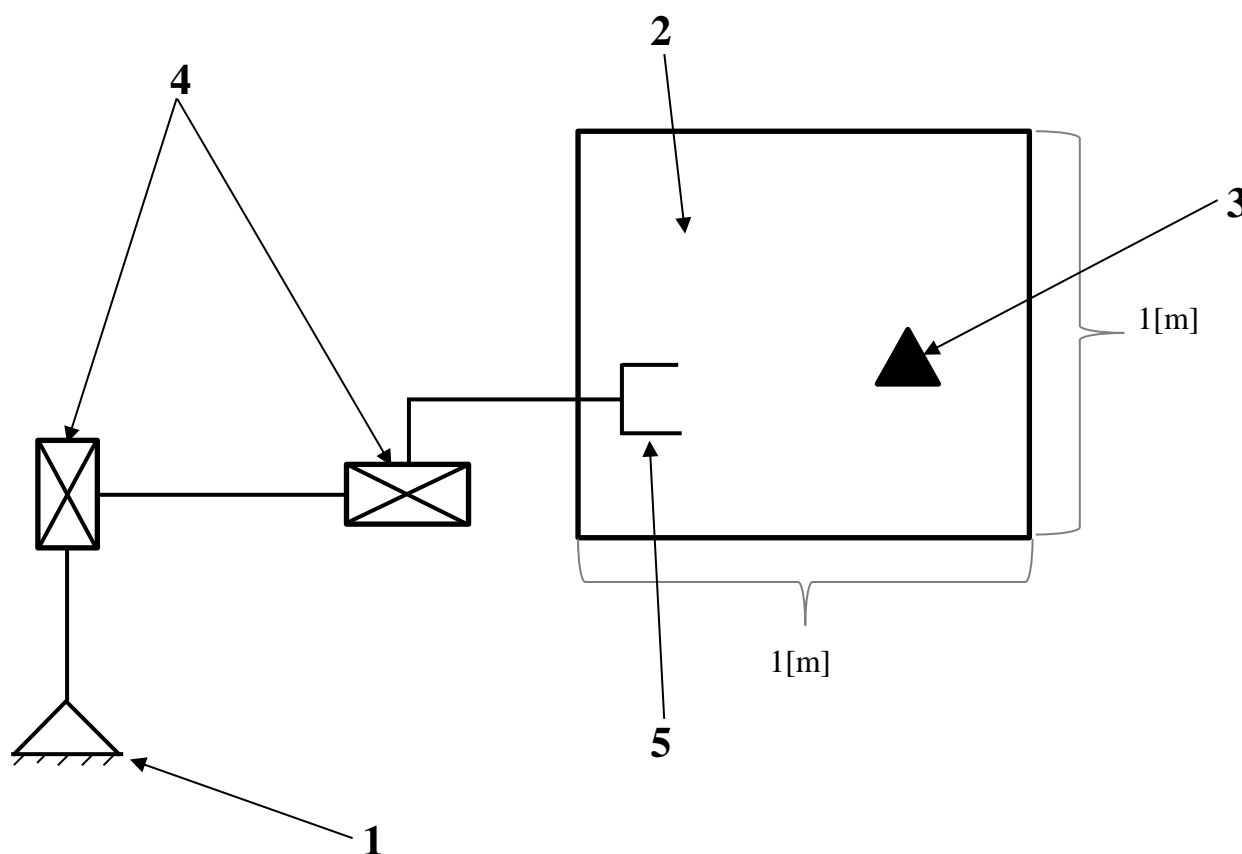


Fig.2 Schemă braţ robotic cu două cuple de translaţie

În schema precedentă (fig.2) se observă un braţ robotic unde sarcina acestuia este să realizeze operaţia de Pick & Place (sau de urmărire a traiectoriei) pe un plan cu aria de 1 m^2 . Elementele care se regăsesc în schemă sunt:

- 1 – reprezintă baza
- 2 – reprezintă planul de desfăşurare
- 3 – reprezintă un obiect aleatoriu
- 4 – reprezintă cuplele de translaţie
- 5 – reprezintă efectorul

Mobilitatea este o măsură a capacităţii unui obiect de a se deplasa şi reprezintă numărul total de grade de libertate (restricţii) pe care le are sistemul. Pentru robotul nostru avem:

Cupla 1 : *Translatie*₀₋₁, 1 grad de libertate

Cupla 2 : *Translatie*₁₋₂, 1 grad de libertate

$$\text{Mobilitate} = 3(n-1) - 2 \cdot C_1 - C_2 = 2 \quad (1)$$

Unde: n – numărul total de elemente

C_1 - numărul de elemente cu 1 grad de libertate

C_2 - numărul de elemente cu mai multe grade de libertate

3. Modelul Geometric

a. Model Geometric Direct

Modelul Geometric Direct reprezintă o problemă de geometrie statică pentru calculul poziţiei şi orientării efecteurului robotului în spaţiul cartezian folosind poziţiile şi orientările din spaţiul articular.

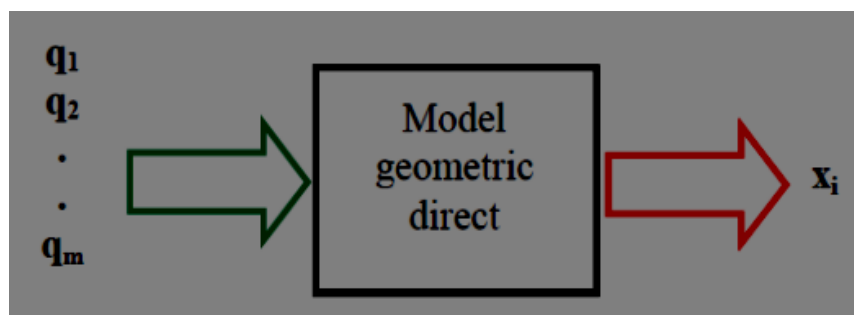


Fig. 3.1 Modelul Geometric Direct

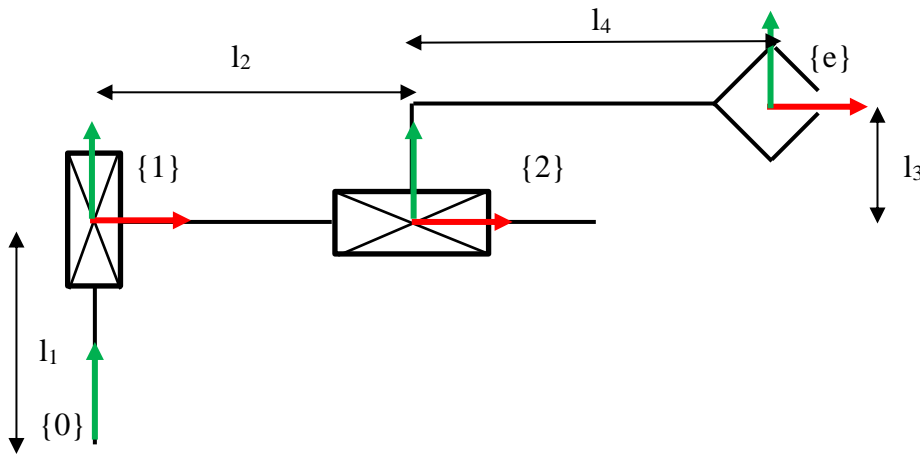


Fig. 3.2 Braţ robotic cu două cuple de translaţie

Modelul Geometric Direct are în alcătuirea sa matricea de transformare 0_eT prin intermediul căreia se pot calcula valorile coordonatelor. Pentru aflarea termenului 0_eT se vor înmulţi matricile de transformare pentru translaţie de la un reper la altul.

$$1 \rightarrow 0 \quad q = 0$$

$${}^0_1T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y + q \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow {}^0_1T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l1 + q1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matricea de translaţie de la 0 la 1

$$1 \rightarrow 2 \quad q = 0$$

$${}^1_2T = \begin{bmatrix} 1 & 0 & 0 & t_x + q \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow {}^1_2T = \begin{bmatrix} 1 & 0 & 0 & l2 + q2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Matricea de translaţie de la 1 la 2

$$2 \rightarrow e \quad q = 0$$

$${}^2_eT = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow {}^2_eT = \begin{bmatrix} 1 & 0 & 0 & l_4 \\ 0 & 1 & 0 & l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Matricea de translaţie de la 2 la e

$${}^0_eT = {}^0_1T {}^1_2T {}^2_eT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_1 + q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 + q_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_4 \\ 0 & 1 & 0 & l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$${}^0_eT = \begin{bmatrix} 1 & 0 & 0 & l_2 + l_4 + q_2 \\ 0 & 1 & 0 & l_1 + l_3 + q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matricea de translaţie finală

În etapa de translaţie pe axa y de la 1→0 s-a utilizat matricea $D_y(q_1)$;
În etapa de translaţie pe axa x de la 1→2 s-a utilizat matricea $D_x(q_2)$;
 l_1, l_2, l_3 şi l_4 sunt parametrii geometrici ai elementelor

b. Ecuaţiile Jacobianului

Jacobianul este o matrice utilizată în mecanica roboticii şi în controlul sistemelor. Acesta conţine derivatele parţiale ale mişcărilor unui sistem mecanic în raport cu parametrii de control ai acestuia. Este utilizat pentru a analiza relaţiile dintre parametrii de control şi mişcărilor sistemului şi pentru a optimiza acţiunile de control ale robotului.

Din matricea 0T_e obţinem $P = {}^0T_e(1:3,4)$ (3)

$$v = J_v \cdot \dot{q} \quad (4)$$

$$v = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \dot{x}_P = \frac{\partial x_P}{\partial q_1} \cdot \dot{q}_1 + \dots + \frac{\partial x_P}{\partial q_n} \cdot \dot{q}_n \quad (4.1)$$

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (5)$$

$$J_v = \begin{bmatrix} \frac{\partial x_P}{\partial q_1} & \frac{\partial x_P}{\partial q_2} & \dots & \frac{\partial x_P}{\partial q_n} \end{bmatrix} \quad (5.1)$$

$$J_v = \begin{bmatrix} \frac{\partial e_{px}}{\partial q_1} & \frac{\partial e_{px}}{\partial q_2} \\ \frac{\partial e_{py}}{\partial q_1} & \frac{\partial e_{py}}{\partial q_2} \\ \frac{\partial e_{pz}}{\partial q_1} & \frac{\partial e_{pz}}{\partial q_2} \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.2)$$

Cum nu avem rotaţie în structură, rezultă că J_ω este egal cu:

$$J_\omega = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.3)$$

Jacobianul final:

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.4)$$

Inversa Jacobianului:

$$J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.5)$$

v - reprezintă viteza liniară

J_v - reprezintă Jacobianul vitezei liniare

J_w - reprezintă Jacobianul vitezei unghiulare

J - reprezintă Jacobianul

J^{-1} - reprezintă inversa Jacobianului

Cunoscând Jacobianul putem determina atât Modelul Cinematic Direct cât şi Modelul Cinematic Invers.

c. Model Geometric Invers

Modelul Geometric Invers permite determinarea configuraţiei articulaţiilor unui robot pentru a obţine o anumită poziţie dorită. Pentru a determina acest model trebuie cunoscute diferite informaţii care se leagă de cinematica robotului cum ar fi relaţiile de transformare între articulaţii şi punctele finale.

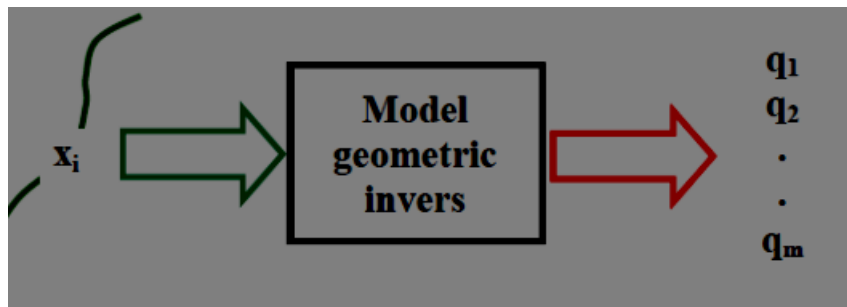


Fig. 4 Model Geometric Invers

Calculul pentru a realiza configuraţia necesară astfel încât efectorul să ajungă la punctul final este făcut printr-o metodă iterativă dată de algoritmul de aproximare numerică Newton.

$$q_{k+1} = q_k + \alpha * J^{-1}(q) * \Delta x \quad (6)$$



Unde:

q_k – reprezintă poziţia efectorului la pasul k

q_{k+1} – reprezintă poziţia efectorului la pasul $k+1$

$J^{-1}(q)$ – reprezintă inversul Jacobianului

Δx – reprezintă eroarea de poziţie (diferenţa între punctul iniţial şi cel final)

α – reprezintă coeficientul de învăţare

Prin aplicarea acestui algoritm se poate afla eroarea de poziţionare a efectorului. Prin fiecare iteraţie eroarea scade până ajunge să aibă o valoare admisibilă.

4. Modelul Cinematic

Modelul Cinematic este un nivel de cunoaştere care porneşte de la următoarele ipoteze:

- Fenomenele descrise de model sunt incluse de relaţia de cauzalitate: viteza unghiulară (din cuple) \rightarrow poziţia şi orientarea prehensorului; viteza prehensorului în spaţiul cartezian;
- Fenomenele nu au în vedere inerţia;
- Nu există interacţiune cu mediul.

a. Model Cinematic Direct

Modelul Cinematic Direct permite determinarea şi orientarea efectorului prin ajutorul coordonatelor operaţionale în funcţie de coordonatele articulare. De asemenea este utilizată pentru a prezice mişcările efectoului în funcţie de informaţiile legate de forţe şi viteză.

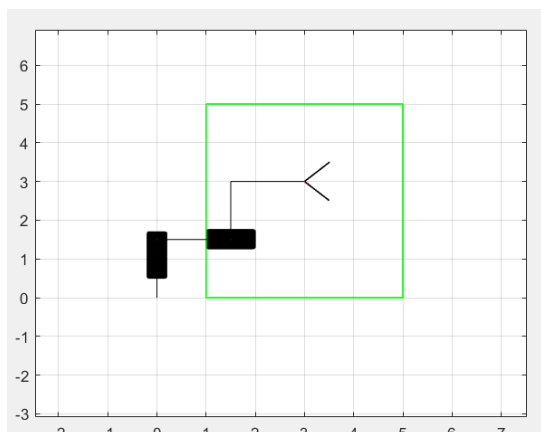


Fig 5.1 Model robot în poziția inițială
(Matlab)

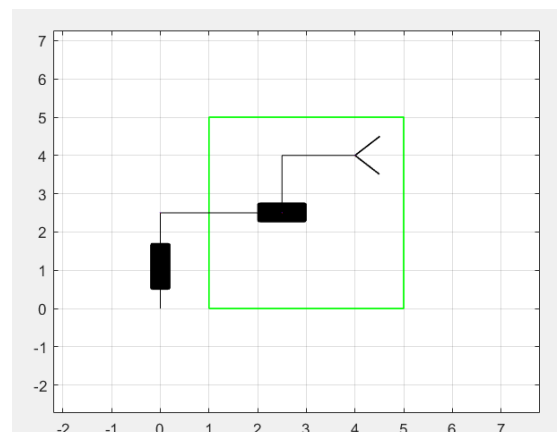


Fig 5.2 Model robot în poziția finală
(Matlab)

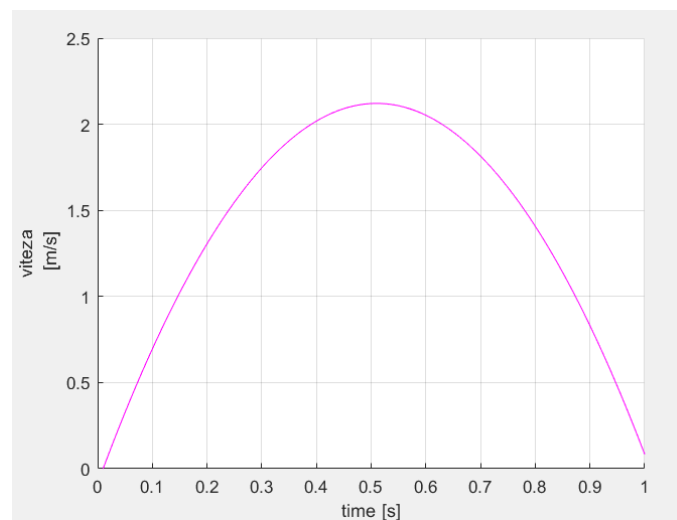


Fig 5.3 Grafic ecuații de viteză



În cazul în care sarcina robotului este de a ajunge de la un punct iniţial la un punct final, pentru interpolare se poate folosi un polinom de gradul 3 – cu patru coeficienţi:

$$q_i(t) = a_{i,0} + a_{i,1}t + a_{i,2}t^2 + a_{i,3}t^3 \quad (7.1)$$

$$\dot{q}_i(t) = a_{i,1} + 2a_{i,2}t + 3a_{i,3}t^2 \quad (7.2)$$

$$\ddot{q}_i(t) = 2a_{i,2} + 6a_{i,3}t \quad (7.3)$$

În care:

i este numărul cuplei (1 sau 2)

necunoscutele $a_{i,0}$, $a_{i,1}$, $a_{i,2}$, $a_{i,3}$ se calculează pornind de la următoarele condiţii:

1. la momentul $t=0$ se impune poziţia şi viteza iniţială $q_{i,1}$, respectiv $\dot{q}_{i,1}$
2. la momentul $t=T$, unde T este durata mişcărilor, se impune poziţia şi viteza finală, $q_{i,n}$, respectiv $\dot{q}_{i,n}$, şi n este ultimul element al vectorului q_i

Ceea ce conduce la următoarele rezultate:

$$a_{i,0} = q_{i,1} \quad (8.1)$$

$$a_{i,1} = \dot{q}_{i,1} \quad (8.2)$$

$$a_{i,2} = \frac{3}{T^2}(q_{i,n} - q_{i,1}) - \frac{1}{T}(\dot{q}_{i,n} + 2\dot{q}_{i,1}) \quad (8.3)$$

$$a_{i,3} = \frac{2}{T^3}(q_{i,1} - q_{i,n}) + \frac{1}{T^2}(\dot{q}_{i,n} + \dot{q}_{i,1}) \quad (8.4)$$

În general vitezele iniţiale şi finale sunt nule (pleacă din repaus şi ajunge în repaus), rezultând ecuaţia modificată:

$$a_{i,0} = q_{i,1} \quad (9.1)$$

$$a_{i,1} = 0 \quad (9.2)$$

$$a_{i,2} = \frac{3}{T^2}(q_{i,n} - q_{i,1}) \quad (9.3)$$

$$a_{i,3} = \frac{2}{T^3}(q_{i,1} - q_{i,n}) \quad (9.4)$$

b. Model Cinematic Invers

Modelul Cinematic Invers permite determinarea configuraţiei în care trebuie să ajungă structura mecanică a robotului astfel încât efectorul să ajungă în poziţia dorită. Se foloseşte modelul cinematic direct pentru realizarea acestui model, deoarece se pot folosi relaţiile între configuraţia articulaţiilor şi poziţia şi orientarea punctului final.

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J(q)^{-1} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (10.1)$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (10.2)$$

$$\dot{q}_1 = v_y \quad (10.3)$$

$$\dot{q}_2 = v_x \quad (10.4)$$

Pentru calculul momentelor s-au folosit:

$$\Gamma = J(q)^T F \quad (11.1)$$

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} \quad (11.2)$$

$$\Gamma = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} F \quad (11.3)$$

5. Modelul Dinamic

În general modelul dinamic este o aproximare a realităţii care permite stabilirea unei relaţii cauzale în condiţiile în care starea sistemului este cunoscută. În acest mod dinamica poate face predicţii asupra viitorului doar dacă, pe lângă noile cauze care acţionează asupra sistemului, sunt cunoscute şi stările anterioare ale acestuia.

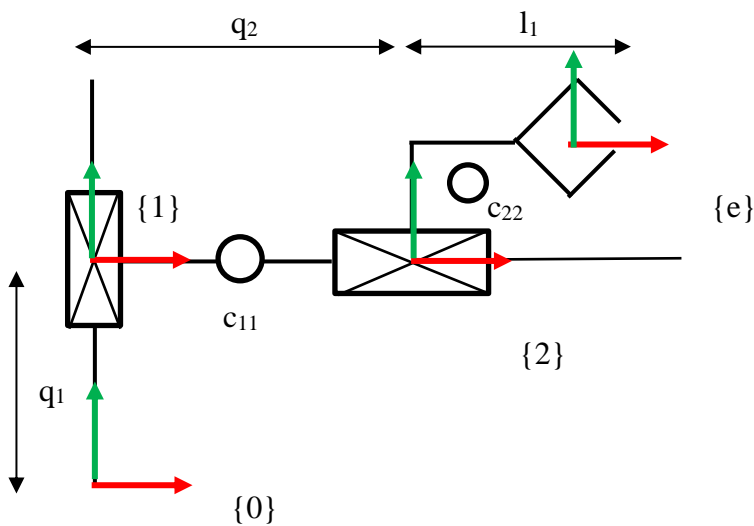


Fig. 6 Braţ Robotic cu două cuple de translaţie

Formula generalizată pentru determinarea forţelor şi a momentelor generate de un sistem este:

$$M(q)[\ddot{q}] + C(q)[\dot{q}^2] + B(q)[\dot{q}\dot{q}] + G(q) = T \quad (12)$$

În care:

$M(q)$ – reprezintă matricea de inerţie

$C(q)$ – reprezintă matricea forţelor Centrifuge

$B(q)$ – reprezintă matricea forţelor Coriolis

$G(q)$ – reprezintă matricea forţelor gravitaţionale

Pentru determinarea Modelului Dinamic se pot folosi două metode distincte de calcul:

1. Metoda Newton-Euler. Aceasta este o tehnică utilizată în dinamica sistemelor mecanice pentru a descrie şi analiza mişcarea şi comportamentul acestora. Ea se bazează pe două seturi de ecuaţii: ecuaţiile lui Newton şi ecuaţiile lui Euler. Aceste ecuaţii sunt combinate pentru a obţine informaţii detaliate despre forţele şi momentele care acţionează asupra sistemului.

2. Metoda Lagrange-Euler. Aceasta este şi ea o tehnică utilizată în dinamica sistemelor mecanice pentru a descrie şi analiza mişcarea şi comportamentul acestora bazându-se pe ecuaţiile lui Lagrange şi Euler. Ea se foloseşte de principiile energetice şi este mai uşor de implementat pentru sisteme cu un număr mare de articulaţii şi corpuri rigide interconectate, unde se pot lua în considerare forţele care apar în sistem, cum ar fi forţele de inerţie şi sarcinile dinamice.

Pentru acest proiect am ales să determin Modelul Dinamic folosindu-mă de **Metoda Lagrange-Euler**:

$$p_{C1}^0 = \begin{bmatrix} c_{11} \\ q_1 \\ 0 \end{bmatrix} \quad (13.1) \quad J_{v1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (13.2)$$

$$p_{C2}^0 = \begin{bmatrix} q_2 + c_{22,x} \\ q_1 + c_{22,y} \\ 0 \end{bmatrix} \quad (14.1) \quad J_{v2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (14.2)$$

$$I_{C1,2} = \begin{bmatrix} I_{xx1,2} & -I_{xy1,2} & -I_{xz1,2} \\ -I_{yx1,2} & I_{yy1,2} & -I_{yz1,2} \\ -I_{zx1,2} & -I_{zy1,2} & I_{zz1,2} \end{bmatrix} \quad (15)$$

$$J_{\omega 1,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (16.1)$$

m_1, m_2 – masa elementului

$$M = m_1 * J_{v1}^T * J_{v1} + m_2 * J_{v2}^T * J_{v2} + J_{\omega 1}^T * I_{C1} * J_{\omega 1} + J_{\omega 2}^T * I_{C2} * J_{\omega 2} \quad (17.1)$$

$$J_{\omega 1}^T * I_{C1} * J_{\omega 1} + J_{\omega 2}^T * I_{C2} * J_{\omega 2} = 0 \quad (17.2)$$

$$M = m_1 * \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} + m_2 * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (18.1)$$

$$M = m_1 * \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + m_2 * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 * m_1 + m_2 * 1 & 0 \\ 0 & 1 * m_2 \end{bmatrix} \quad (18.2)$$

$$C_{(q)[\dot{q}^2]} = \begin{bmatrix} b_{111} & b_{122} \\ b_{211} & b_{222} \end{bmatrix} * \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} \quad (19)$$

$$b_{111} = \frac{1}{2} \left(\frac{m_{11}}{\sigma q_1} + \frac{m_{11}}{\sigma q_1} - \frac{m_{11}}{\sigma q_1} \right) = 0 \quad (20.1)$$

$$b_{122} = \frac{1}{2} \left(\frac{m_{12}}{\sigma q_2} + \frac{m_{12}}{\sigma q_2} - \frac{m_{22}}{\sigma q_1} \right) = 0 \quad (20.2)$$

$$b_{211} = \frac{1}{2} \left(\frac{m_{21}}{\sigma q_1} + \frac{m_{21}}{\sigma q_1} - \frac{m_{11}}{\sigma q_2} \right) = 0 \quad (20.3)$$

$$b_{222} = \frac{1}{2} \left(\frac{m_{22}}{\sigma q_2} + \frac{m_{22}}{\sigma q_2} - \frac{m_{22}}{\sigma q_2} \right) = 0 \quad (20.4)$$

Nu avem forţă centripedă $B_{(q)[qq]}$ şi forţă coriolis $C_{(q)[\dot{q}^2]}$

$$G_{(q)} = -[J_{v1}^T J_{v2}^T] \begin{bmatrix} m_1 * g \\ m_2 * g \end{bmatrix} - \text{braşul robotic este perpendicular cu axa gravitaţională ceea} \quad (21)$$

$$\text{ce rezultă că } G_{(q)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

În final, avem:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix} * \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \quad (22)$$

6. Simulare Virtuală

Simularea virtuală a unui robot este o tehnică folosită pentru a crea şi testa un model computerizat al unui robot într-un mediu virtual, oferind o platformă de dezvoltare şi evaluare a performanţei robotului într-un mod eficient şi sigur.

Pentru simularea virtuală a robotului cu structura formată din două cuple de translaţie au fost folosite soft-urile Matlab (partea de Simulink în special) şi CatiaV5. Acestea au fost folosite pentru a realiza modelul virtual al robotului şi pentru controlul acestuia odată ce a fost importat în Simulink. Au fost utilizate diferite metode de control pentru robot, acestea fiind exemplificate în prototipurile virtuale create.

a. Alegerea Motoarelor

Pentru ambele cuple se v-a alege acelaşi model de motor electric de curent continuu cu scopul de a pune în mişcare ansamblul robotic. Pentru a putea face această alegere este nevoie să cunoaştem momentele şi forţele care apar la ansamblul robotului, dar în cazul structurii de tip TT, se vor lua în calcul doar forţele rezultate (referinţă fig. 7).

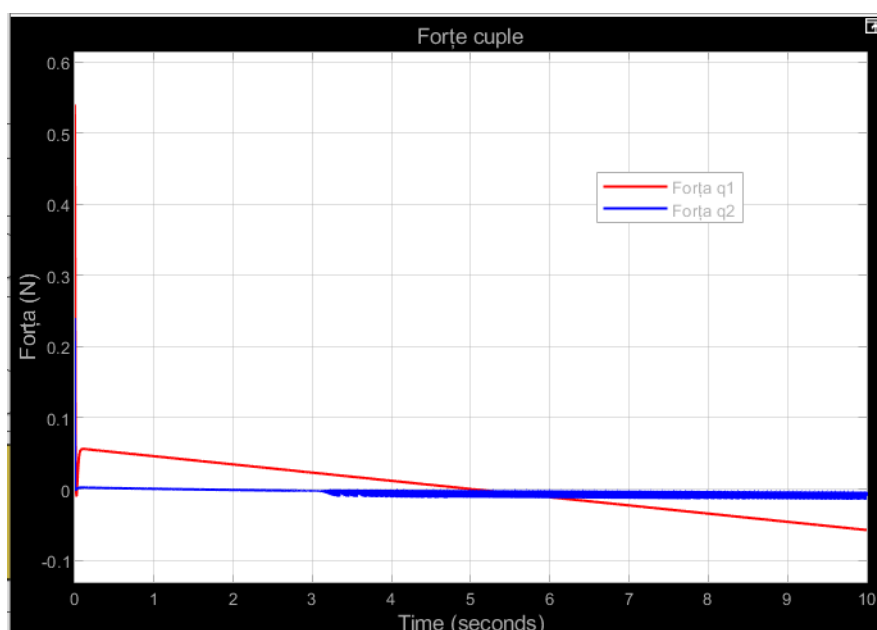
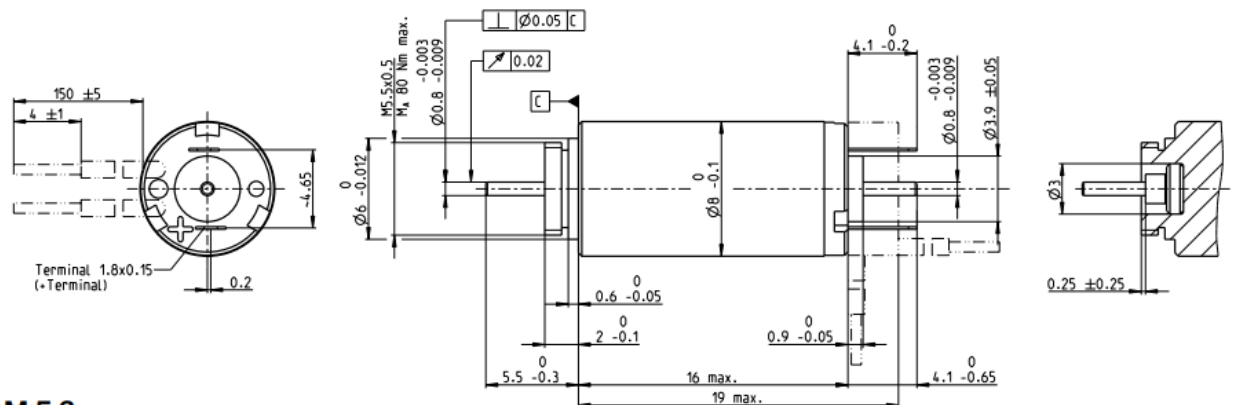


Fig. 7 Graficul forţelor în cuple (Prototip Virtual 2)

În funcţie de acestea, s-a ales un motor de curent continuu cu perii din gama largă a Companiei Maxon împreună cu un reductor. Specificaţiile acestora sunt următoarele:

RE 8 Ø8 mm, precious metal brushes, 0.5 watt

RE



M 5:2

- Stock program
- Standard program
- Special program (on request)

Part Numbers

		Part Numbers					
B with cables		462207	463219	463220	463221	463222	463223
A with terminals		347723	347724	347725	347728	347726	347727
Motor Data							
Values at nominal voltage							
1 Nominal voltage	V	2.4	4.2	6	7.2	9	12
2 No load speed	rpm	13900	14200	13300	14300	14400	15600
3 No load current	mA	19.2	11.2	7.3	6.66	5.35	4.44
4 Nominal speed	rpm	4320	4480	3500	4220	4760	5410
5 Nominal torque	mNm	0.63	0.624	0.616	0.596	0.626	0.589
6 Nominal current (max. continuous current)	A	0.412	0.237	0.155	0.134	0.113	0.0865
7 Stall torque	mNm	0.925	0.932	0.857	0.866	0.957	0.925
8 Stall current	A	0.581	0.34	0.207	0.187	0.166	0.13
9 Max. efficiency	%	67	67	66	66	68	67
Characteristics							
10 Terminal resistance	Ω	4.13	12.3	29	38.5	54.3	92.2
11 Terminal inductance	mH	0.03	0.09	0.206	0.257	0.4	0.606
12 Torque constant	mNm/A	1.59	2.74	4.15	4.63	5.77	7.11
13 Speed constant	rpm/V	6000	3490	2300	2060	1650	1340
14 Speed / torque gradient	rpm/mNm	15600	15700	16100	17200	15500	17400
15 Mechanical time constant	ms	6.31	6.3	6.34	6.44	6.29	6.49
16 Rotor inertia	gcm ²	0.0388	0.0383	0.0375	0.0358	0.0387	0.0355

Fig. 8 Desenul tehnic şi parametrii motorului

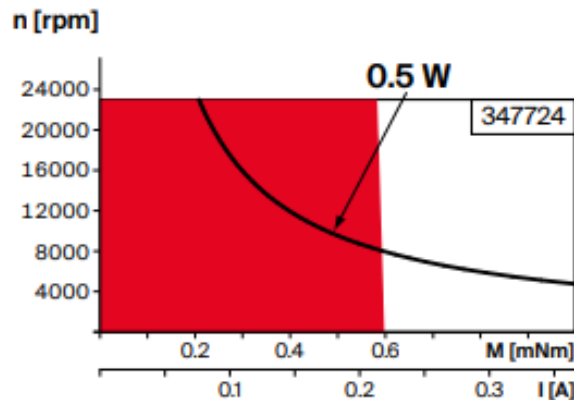
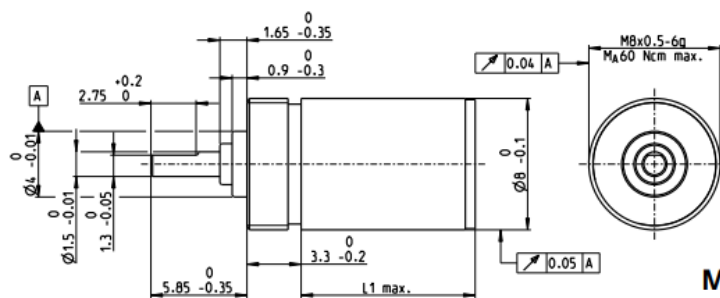


Fig. 9 Caracteristica mecanică de funcţionare a motorului

În figura precedentă se poate observa caracteristica mecanică de funcţionare a motorului ales. Zona marcată cu roşu reprezintă funcţionalitatea continuă a motorului, iar zona cu alb semnifică zona în care acesta funcţionează pe o perioadă scurtă de timp.

Planetary Gearhead GP 8 A Ø8 mm, 0.01–0.1 Nm



Technical Data	
Planetary Gearhead	straight teeth
Output shaft	stainless steel, hardened
Bearing at output	ball bearing
Radial play, 5 mm from flange	max. 0.08 mm
Axial play	max. 0.08 mm
Max. axial load (dynamic)	5 N
Max. force for press fits	10 N
Direction of rotation, drive to output	=
Max. continuous input speed	12 000 rpm
Recommended temperature range	-15...+80°C
Number of stages	1 2 3 4 5
Max. radial load, 5 mm from flange	5 N 6 N 7 N 8 N 8 N

gear

Gearhead Data	Part Numbers							
	468999	468998	474124	468997	474127	468996	474129	468995
1 Reduction	4:1	16:1	36:1	64:1	216:1	256:1	1296:1	1024:1
2 Absolute reduction	4	16	36	64	216	256	1296	1024
3 Max. motor shaft diameter	mm 1	1	0.65	1	0.65	1	0.65	1
4 Number of stages	1	2	2	3	3	4	4	5
5 Max. continuous torque	Nm 0.01	0.020	0.008	0.060	0.020	0.080	0.040	0.100
6 Max. intermittent torque at gear output	Nm 0.015	0.030	0.012	0.090	0.030	0.120	0.060	0.150
7 Max. efficiency	% 90	81	76	73	66	65	57	59
8 Weight	g 2.6	3.2	3.2	3.8	3.8	4.4	4.4	5.0
9 Average backlash no load	° 1.80	2.0	2.4	2.2	2.6	2.50	2.8	2.80
10 Mass inertia	gcm ² 0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004
11 Gearhead length L1	mm 5.5	8.1	8.3	10.7	11.1	13.3	13.9	15.9

Fig. 10 Desenul tehnic şi parametrii reductorului

b. Prototip Virtual 1

Primul prototip virtual conţine un model realizat în CatiaV5 similar conceptului din capitolul 2 (referinţă fig. 2) iar controlul acestui prototip va fi făcut prin intermediul motoarelor care sunt controlate prin intermediul voltajului de la intrare.

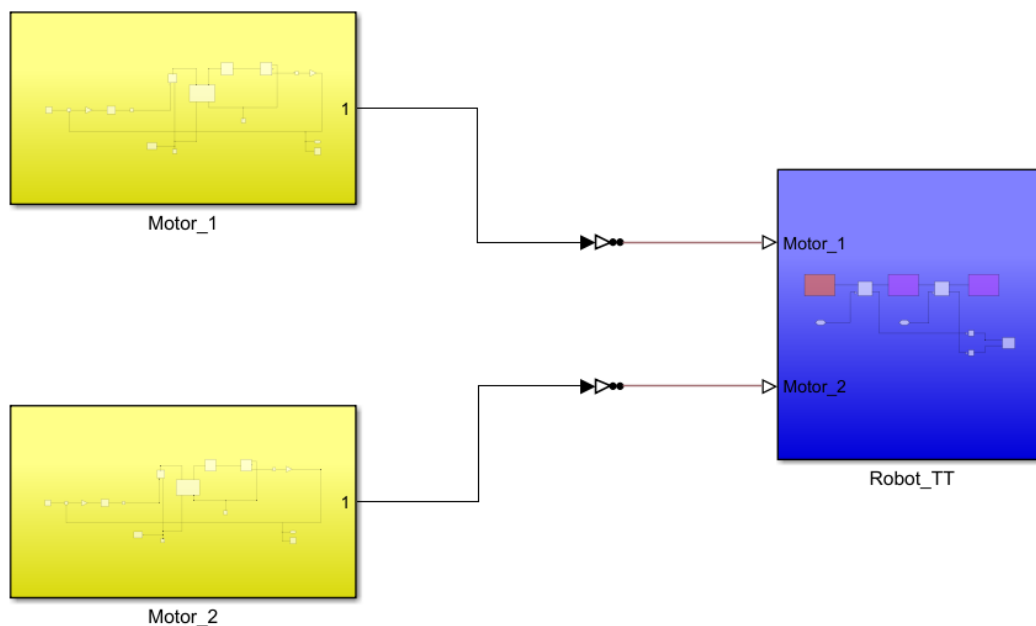


Fig. 11.1 Schemă Sistem Prototip Virtual 1

Sistemul realizat în Simulink prezintă două subsisteme identice care sunt pentru cele două motoare şi un subsistem care conţine structura robotului importată din CatiaV5.

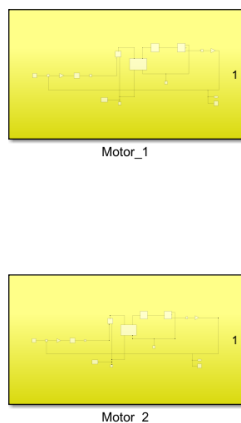


Fig. 11.2.1 Blocurile motoarelor

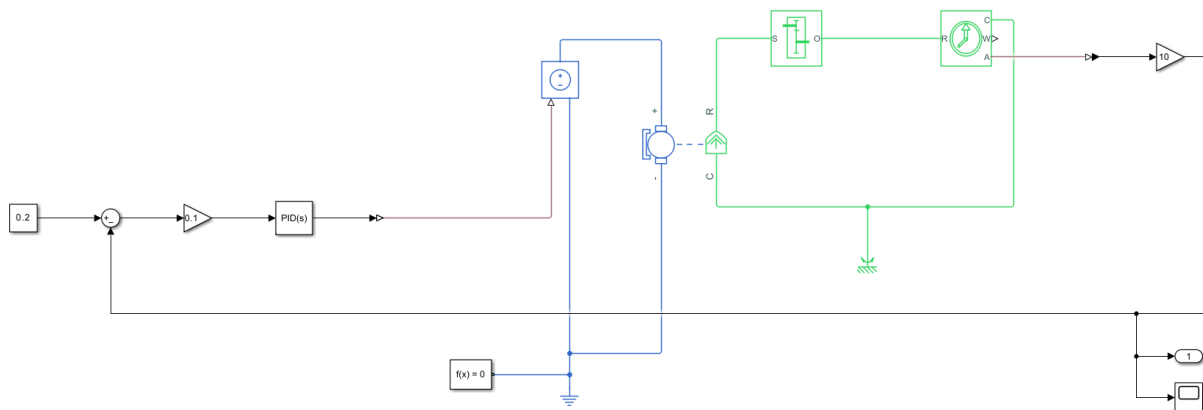


Fig. 11.2.2 Subsistemul blocului motoarelor

Se observă prezenţa unui regulator PID care are rolul de a calcula şi amplifica eroarea care ajunge să fie trimisă la motoare sub formă de voltaj.

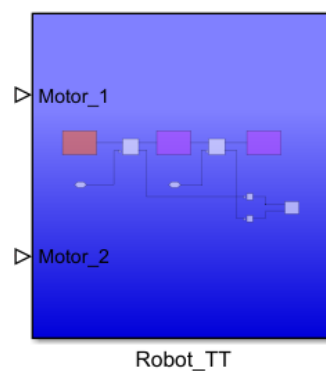


Fig. 11.3.1 Blocul Robotului TT

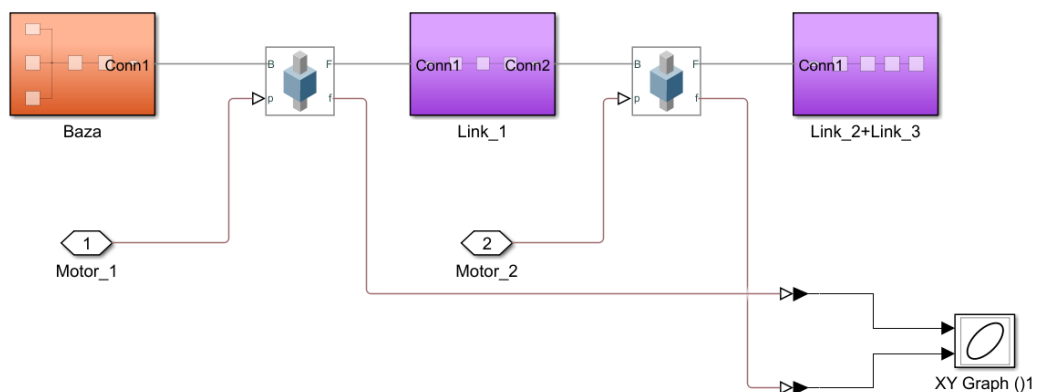


Fig. 11.3.2 Subsistemul Robotului TT

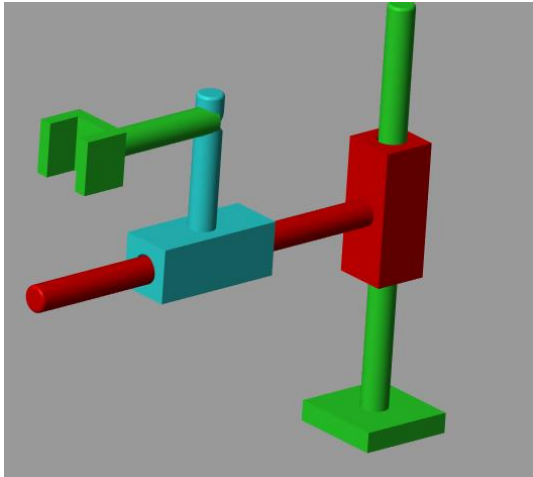


Fig. 11.3.3 Robot TT v1 în poziţia iniţială

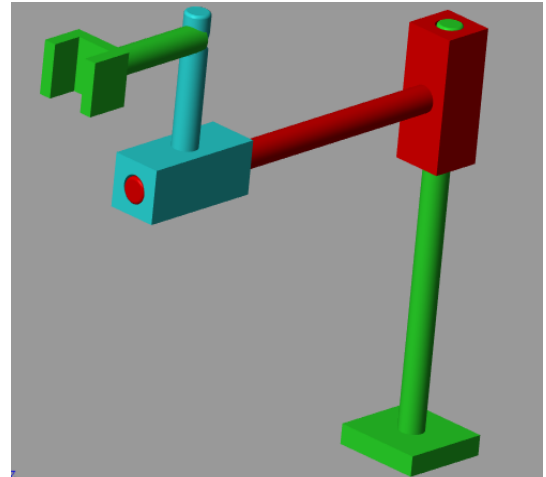


Fig. 11.3.4 Robot TT v1 în poziţia finală

c. Prototip Virtual 2

Prototipul virtual 2 foloseşte un nou model făcut în CatiaV5 pentru a reprezenta robotul de tip TT şi utilizează pentru control legea de mişcare formată din polinomul de gradul 3 care duce la aflarea poziţiei, vitezei şi a acceleraţiei robotului. De asemenea sunt incluse caracteristici precum funcţia de transfer care descrie relaţia matematică între intrările şi ieşirile sistemului, matricea de inerţie, traiectoria dată de poziţiile în spaţiul cartezian.

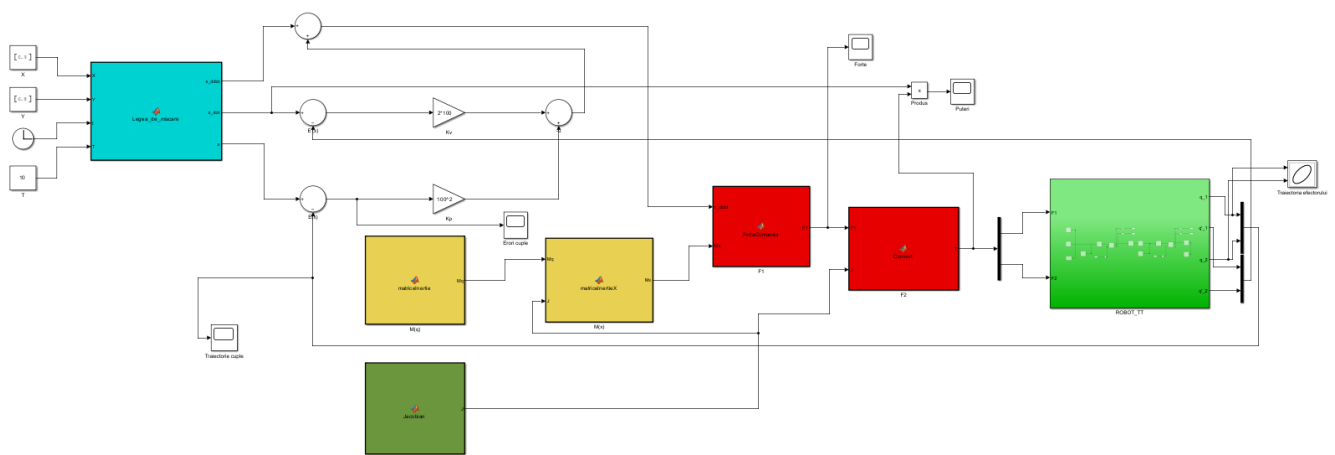


Fig. 12.1 Schemă sistem Prototip Virtual 2

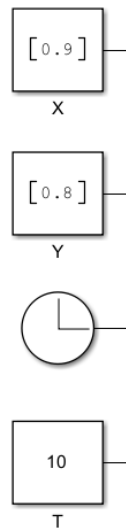


Fig. 12.1.1 Date de intrare



Fig. 12.1.2 Coeficienţi

K_v - reprezintă coeficientul de viteză

K_p - reprezintă coeficientul de proporţionalitate

$M(q)$ – reprezintă matricea de inerţie

$M_x(q)$ – reprezintă matricea de inerţie în reper cartezian

$J(q)$ – reprezintă Jacobianul

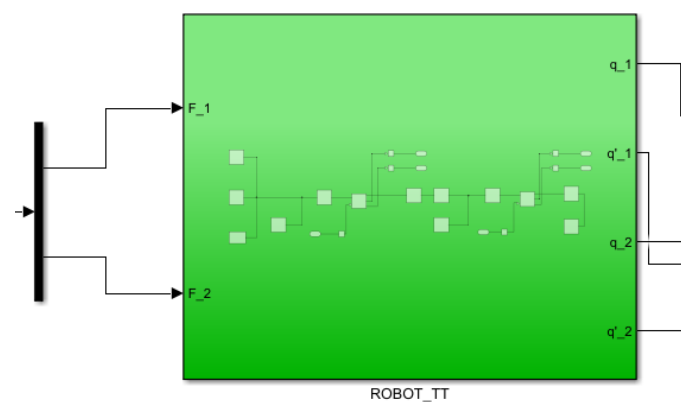


Fig. 12.2.1 Blocul Robotului TT

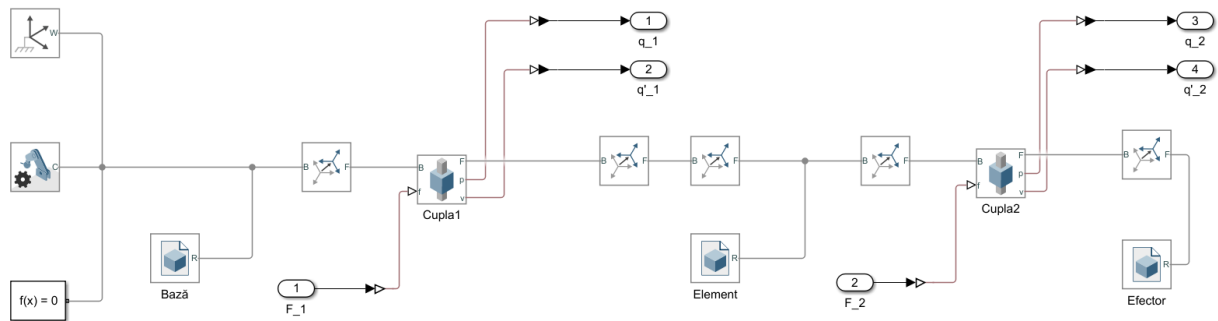


Fig. 12.2.2 Subsistemul Robotului TT

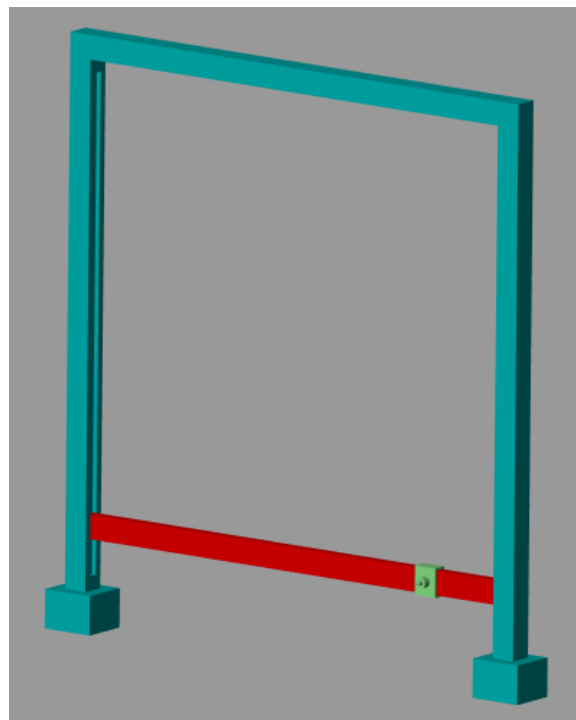


Fig. 12.2.3 Robot TT v2 în poziția [0.9;0.8]

Noua variantă de robot importată din CatiaV5 a fost făcută pentru a reprezenta mai bine un robot având două cuple de translație. Efectorul are o traiectorie bine definită şi se poate deplasa oriunde pe planul său de desfăşurare având aria de 1 m².

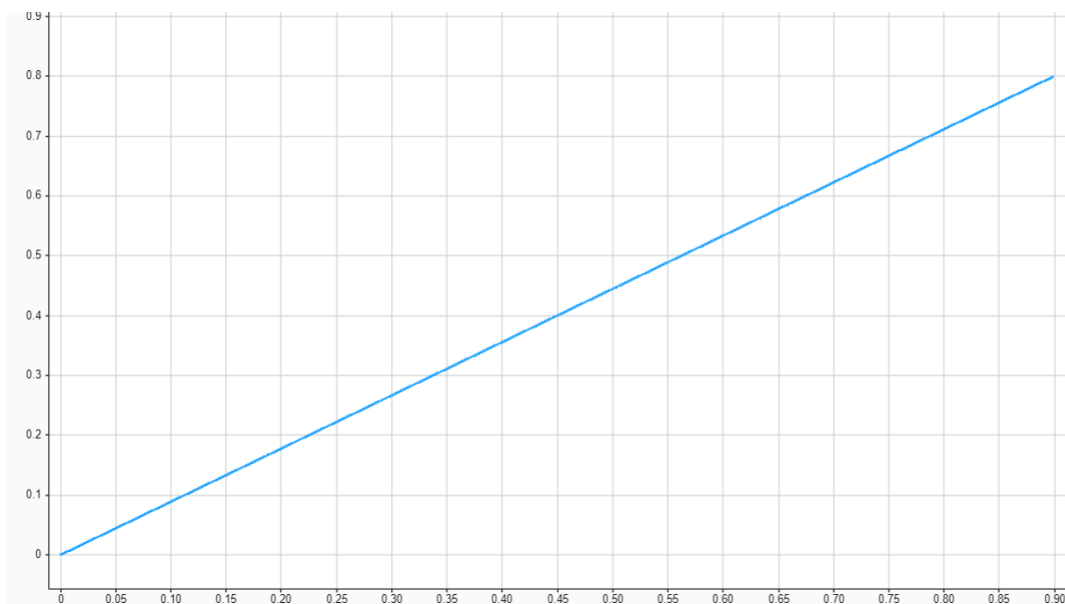


Fig. 12.2.4 Traectoria efectorului

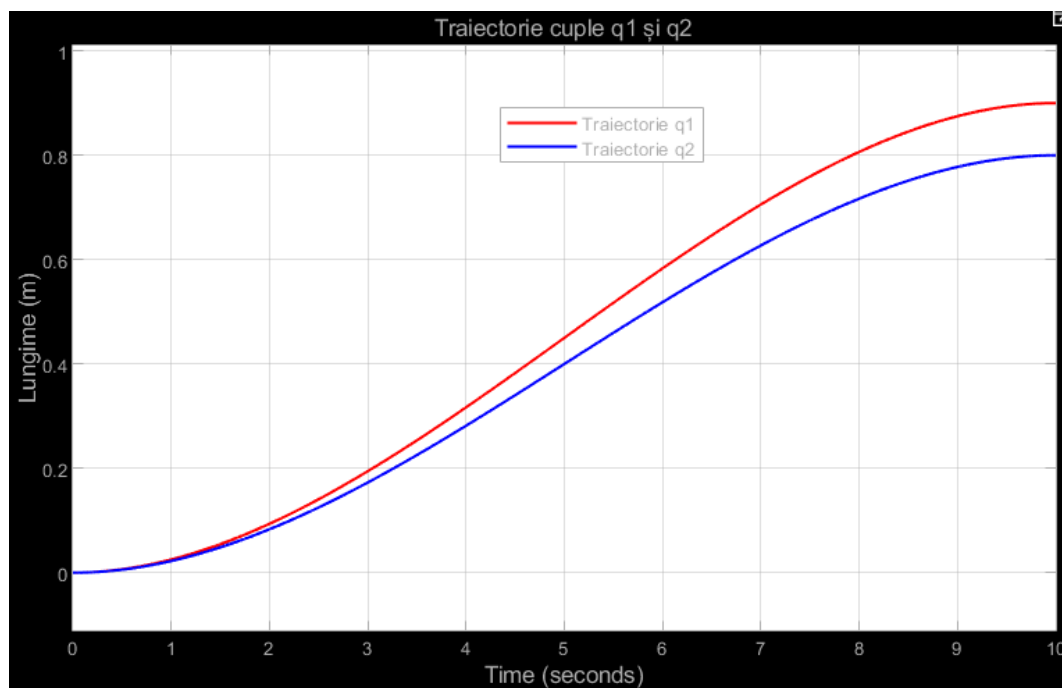


Fig. 12.2.5 Traectoria cuplelor în raport cu timpul

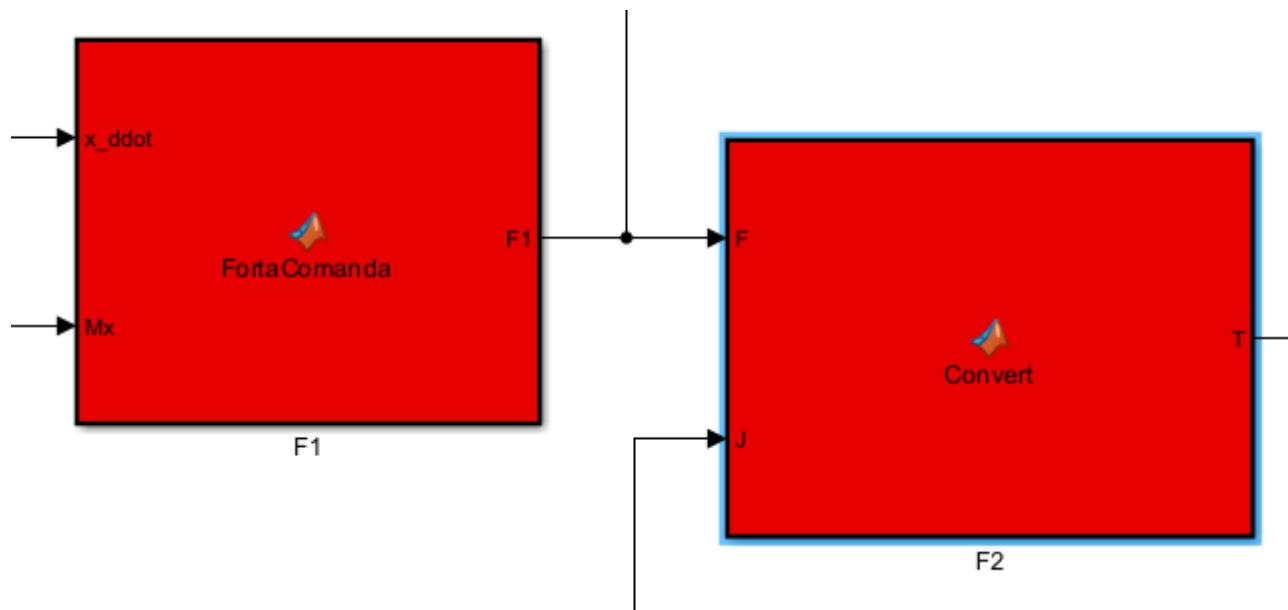


Fig. 12.3.1 Blocurile F1 şi F2

Blocul F1 reprezintă Forţa de comandă în reperul cartezian şi are formula:

$$F1 = M(x) * \ddot{x} \quad (23)$$

Blocul F2 reprezintă Forţa de comandă calculată anterior cum este convertită din spaţiul cartezian în spaţiul articular folosindu-se formula:

$$J(q)^T * F = T \quad (24)$$

T – reprezintă momentul

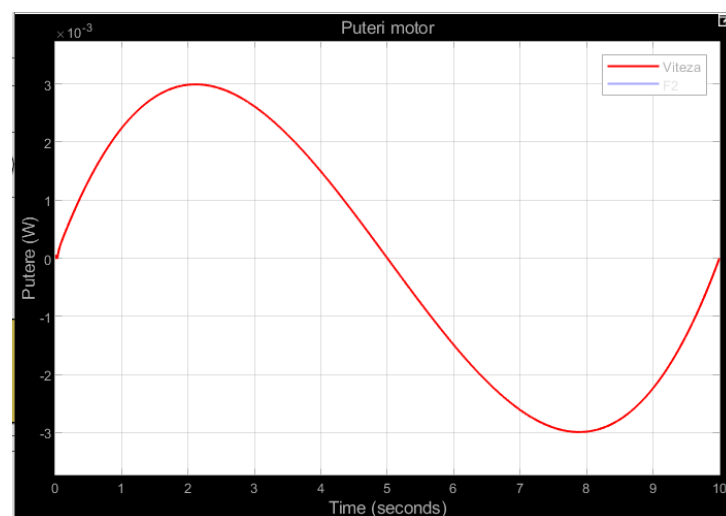


Fig 12.3.2 Graficul puterii cuplelor

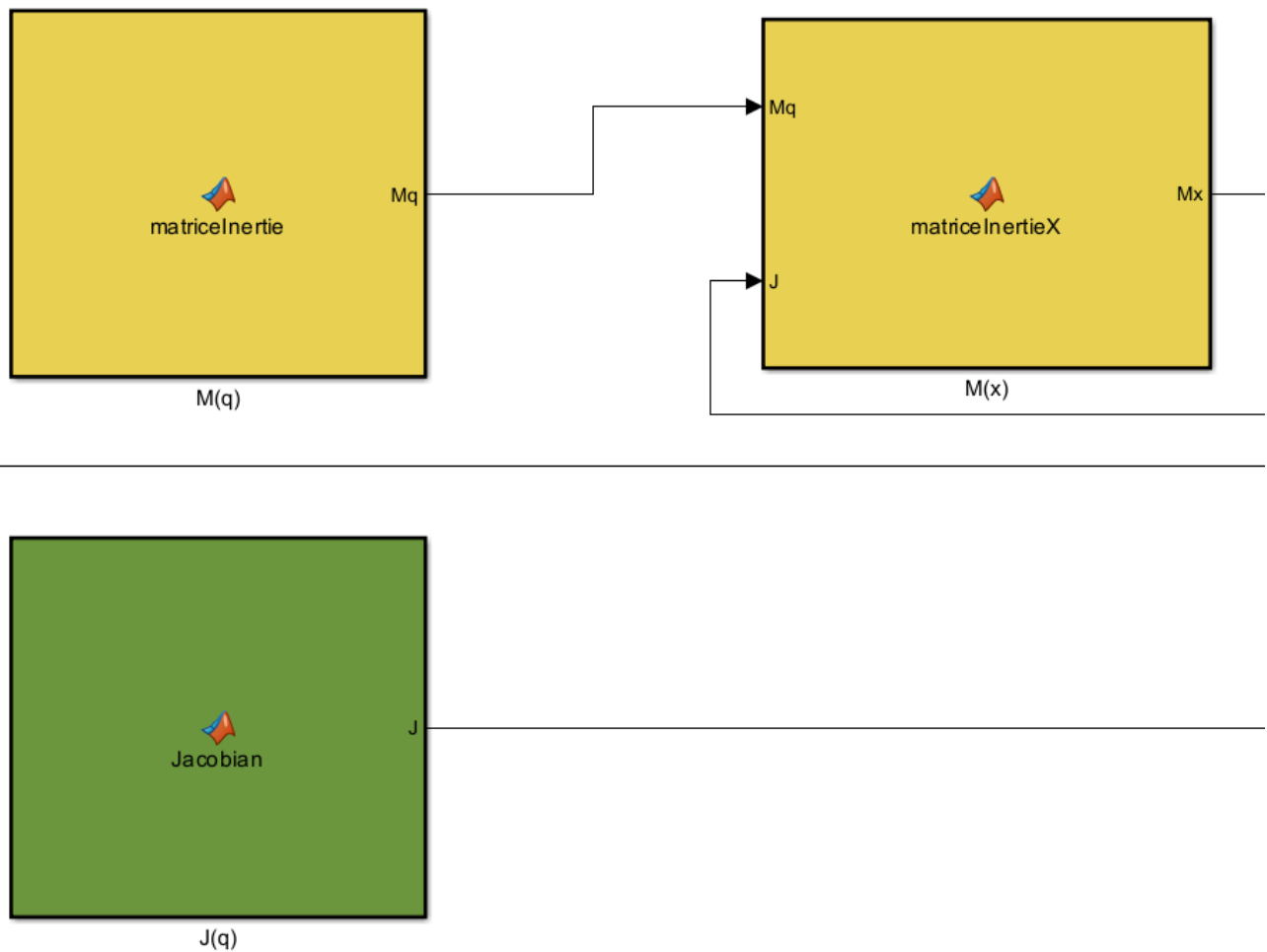


Fig. 12.4.1 Blocurile $M(q)$, $M(x)$ și $J(q)$

Blocul $M(q)$ reprezintă matricea de inerție, aceasta fiind calculată la modelul dinamic (referință capitolul 5, formula nr. 22).

Blocul $M(x)$ reprezintă matricea de inerție în reper cartezian și este egală cu:

$$J(q)^{-T} * M(q) * J(q)^{-1} \quad (25)$$

Blocul $J(q)$ reprezintă Jacobianul calculat anterior (referință capitolul 3, formula 5.4)

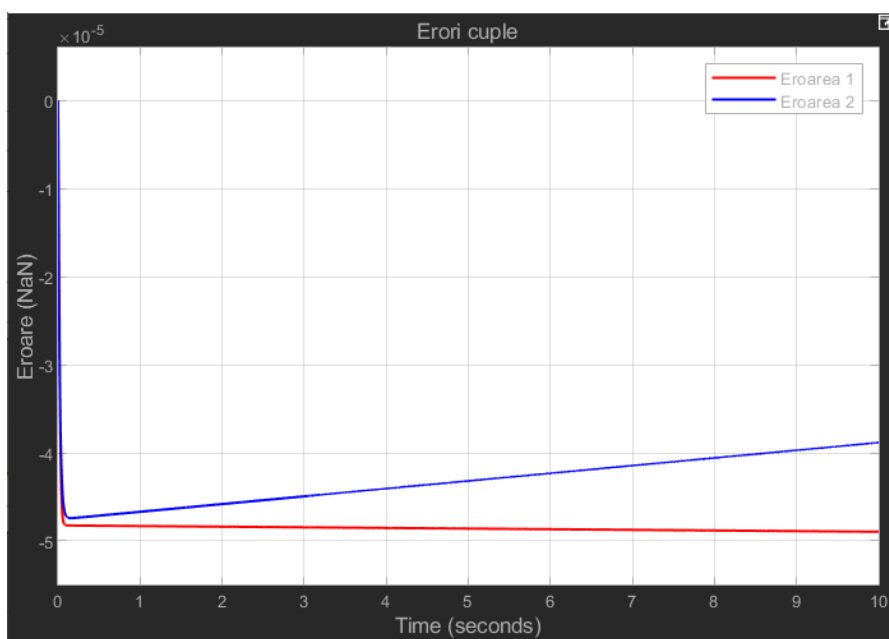


Fig. 12.4.2 Graficul erorilor cuplelor



Fig. 12.5.1 Blocul legii de mişcare



În acest bloc se prezintă legea de mișcare după care funcționează ansamblul robotului TT, aceasta fiind formată din ecuațiile polinomiale următoare:

$$x = [X; Y] * (-2 * \left(\frac{t}{T}\right)^3 + 3 * \left(\frac{t}{T}\right)^2) \quad (26)$$

$$\dot{x} = [X; Y] * (-6 * \frac{t^2}{T^3} + 6 * \frac{t}{T^2}) \quad (27)$$

$$\ddot{x} = [X; Y] * (-12 * \frac{t}{T^3} + \frac{6}{T}) \quad (28)$$

Unde:

$[X; Y]$ – reprezintă coordonatele punctelor

x – reprezintă poziția

\dot{x} – reprezintă viteza

\ddot{x} – reprezintă accelerația

t/T – reprezintă timpul

7. Concluzii

Acest proiect a reprezentat o provocare semnificativă, care a solicitat abilități și resurse considerabile pentru a fi finalizat. Prin intermediul lui am aplicat cunoștințele preluate de la cursuri și am reușit să proiectez un sistem robotic având două cuple de translație care poate îndeplini sarcinile date de a urmări o traiectorie și de a se deplasa în orice punct din planul de desfășurare.



8. Anexe

În această secţiune vor fi prezentate codurile folosite în Matlab.

```
function TT_Robot(q,l)

for t=0:0.01:1
cla

axis([-5 5 -5 5])
P_1 = [1;0]
P_2 = [1;5]
P_3 = [5;5]
P_4 = [5;0]
Square = [P_1,P_2,P_3,P_4,P_1]

q_1 = -2*q(1,:)*t.^3 + 3*q(1,:)*t.^2
q_2 = -2*q(1,:)*t.^3 + 3*q(1,:)*t.^2

%q_1(0) = 0 q_1(0) = a_0 = 0
%q_1(t_f) = q(1,:) q_1(1) = a_3 + a_2 + a_1 = q(1,:) -2*a_2/3 +3*a_2/3 =
%a_2 = 3*q(1,:)
%a_3 = -2*q(1,:)

%q_1'(t) = 3*a_3*t.^2 + 2*a_2*t +a_1
%q_1'(0) = 0 q_1'(0) = a_1 = 0
%q_1'(t_f) = 0 q_1'(1) = 3*a_3 + 2*a_2 = 0 a_3 = -2*a_2/3

x = [0.01;0;0]
y = [0;0.01;0]
o = [0;0;0]

T_01 = [1 , 0 , 0 , 0; ...
        0 , 1 , 0 , l(1,:) + q_1; ...
        0 , 0 , 1 , 0; ...
        0 , 0 , 0 , 1]
T_12 = [1 , 0 , 0 , l(2,:) + q_2; ...
        0 , 1 , 0 , 0; ...
        0 , 0 , 1 , 0; ...
        0 , 0 , 0 , 1]
T_23 = [1 , 0 , 0 , 0 ;...
        0 , 1 , 0 , l(3,:); ...
        0 , 0 , 1 , 0; ...
        0 , 0 , 0 , 1]
T_3e = [1 , 0 , 0 , l(4,:); ...
        0 , 1 , 0 , 0; ...
        0 , 0 , 1 , 0; ...
        0 , 0 , 0 , 1]

T_02 = T_01 * T_12
T_03 = T_02 * T_23
T_0e = T_03 * T_3e

Ox = T_01*[x;1]
Oy = T_01*[y;1]
```

```
Oo = T_01*[o;1]

Sx = T_02*[x;1]
Sy = T_02*[y;1]
So = T_02*[o;1]

Qx = T_03*[x;1]
Qy = T_03*[y;1]
Qo = T_03*[o;1]

Px = T_0e*[x;1]
Py = T_0e*[y;1]
Po = T_0e*[o;1]

Q = [T_01(1,4);T_01(2,4)]'
S = [T_02(1,4);T_02(2,4)]
P = [T_0e(1,4);T_0e(2,4)]

R_matrix = [[o;1],Oo,So,Qo,Po]

r1=rectangle('Position',[-0.2 0.5 0.4 1.2],'Curvature',0.2)
r1.FaceColor = 'black';
r2=rectangle('Position',[So(1,:)-0.5 Sx(1,:)-0.25 1 0.5],'Curvature',0.2)
r2.FaceColor = 'black';

plot([Po(1,:),Px(1,:)+0.5],[Po(2,:),Px(2,:)+0.5],'black','LineWidth',1)
%plot([Po(1,:)+0.3,Px(1,:)+0.2],[Po(2,:)+0.3,Px(2,:)+0.2],'black','LineWidth',1)
plot([Po(1,:),Py(1,:)+0.5],[Po(2,:),Py(2,:)-0.5],'black','LineWidth',1)
%plot([Po(1,:)-0.2,Py(1,:)-0.4],[Po(2,:)-0.2,Py(2,:)-0.2],'black','LineWidth',1)

plot(Square(1,:),Square(2,:), 'green','LineWidth',1)
plot(R_matrix(1,:),R_matrix(2,:), 'black','LineWidth',0.5)

plot([Oo(1,:),Ox(1,:)],[Oo(2,:),Ox(2,:)], 'r','LineWidth',1)
plot([Oo(1,:),Oy(1,:)],[Oo(2,:),Oy(2,:)], 'b','LineWidth',1)

plot([So(1,:),Sx(1,:)],[So(2,:),Sx(2,:)], 'r','LineWidth',1)
plot([So(1,:),Sy(1,:)],[So(2,:),Sy(2,:)], 'b','LineWidth',1)

plot([Po(1,:),Px(1,:)],[Po(2,:),Px(2,:)], 'r','LineWidth',1)
plot([Po(1,:),Py(1,:)],[Po(2,:),Py(2,:)], 'b','LineWidth',1)

hold on
grid on

pause (0.001)
```

Fig. 13.1 Generarea braţului robotic şi deplasarea lui


```
function RobotJ(q,l)
V = zeros(3,101);
D = zeros(3,101);
figure
hold
grid
axis([0 1 0 2.5])
xlabel('time [s]')
ylabel('velocity [m/s]')
for t=0:1:100
    q_1 = -2*q(1,:)*(t/100).^3 + 3*q(1,:)*(t/100).^2;
    q_2 = -2*q(2,:)*(t/100).^3 + 3*q(2,:)*(t/100).^2;

    q_1_diff = -6*q(1,:)*(t/100).^2 + 6*q(1,:)*(t/100);
    q_2_diff = -6*q(2,:)*(t/100).^2 + 6*q(2,:)*(t/100);

    Jv = [0,1;
          1,0;
          0,0];
    V(:,t+1) = Jv * [q_1_diff;q_2_diff];
    if t>0
        plot([t/100,(t+1)/100],[sqrt(V(1,t).^2+V(2,t).^2),sqrt(V(1,t+1).^2+V(2,t+1).^2)], 'magenta', 'Linewidth',0.5)
    end
end
```

Fig. 13.2 Generare lege de mişcare polinomială
Şi grafic viteze

9. Bibliografie

- [1] Pozna, C. (2021). Bazele Cinematicii Robotilor Industriali.
- [2] Pozna, C. (2021). Dinamica Robotilor
- [3] Pozna, C. (2021). Sisteme de Conducere in Robotica.
- [4] <https://www.maxongroup.com/maxon/view/content/index>