

# **Proiect Sisteme de Conducere în Robotică (SCR)**

## **Conducerea unui robot RT**

Student: Ilie Razvan-Mihai

Specializare: Robotica

Grupa: 4LF871

Anul: 2020

## **1. Introducere**

Un brat robotic este utilizat pentru a usura procesul de mutare a materialelor, miscare necesara in procesele de fabricatie. Procesul de transfer al materialelor este de obicei implinit folosind puterea omului, acesta poate provoca rani operatorului. Bratul robotic previne posibilele accidente ale omului si ajuta la cresterea eficientei productiei prin reducerea erorii umane.

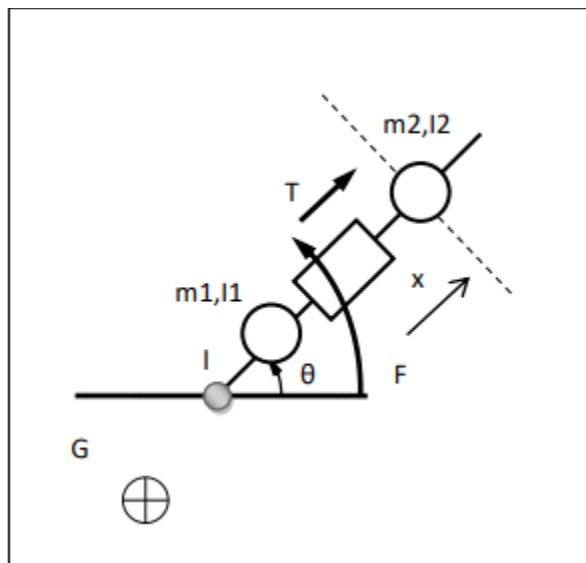
Cei mai multi roboti manipulatori gasiti in operatiile industriale sunt controlati prin algoritmi de tipul PID (instrumentul derivat proportional-integral ). Acest tip de control are o structura simpla pentru cele trei castiguri ale sale.

Exista multe tehnici utilizate pentru controlul bratului robotului. Cele mai cunoscute tehnici de control sunt control PID, control adaptiv, control optim si controlul robust.

Intrucat obiectivul final este proiectarea si fabricarea de roboti adevarati, este necesar sa avem simularea dinaintea investigatiilor cu roboti reali.

## 2. Modelarea geometrica a structurii

Bratul robotic are doua grade de libertate si este constituit dintr-o cupla de rotatie si una de translatie (RT) avand 2 laturi si 2 greutatea ce prin animarea structurii degaja 2 forte: momentul motor (T), forta motoare (F).



$$m_1 = 1 \text{ kg};$$

$$l_1 = 5.02 \text{ m};$$

$$m_2 = 1 \text{ kg};$$

$$l_2 = 0.2 \text{ m};$$

Vom folosi principiul coordonatelor generalizate pentru abstractizarea variabilelor cuplelor:

$i$  - indicele cuplei;

$$\epsilon_i = \{0\text{-rot}; 1\text{-trans}; !\epsilon_i = 1 - \epsilon_i;\}$$

$$q_i = !\epsilon_i * \theta_i + \epsilon_i * x_i;$$

Prin urmare miscarile din cuple devin:

$$q_1 = (1-0) \cdot \theta + 0 \cdot x = \theta;$$

$$q_2 = (1-1) \cdot \theta + 1 \cdot x = x;$$

Cu ajutorul principiului reperului traslatat construim matricile de transformare  ${}^{01}T$ ,  ${}^{12}T$ ,  ${}^{2E}T$ . Matricia de transformare de la originie la end effector:

$${}^{0E}T = {}^{01}T \cdot {}^{12}T \cdot {}^{2E}T;$$

$${}^{ij}T = \begin{bmatrix} {}^{ij}R(3 \times 3), & {}^{ji}D(3 \times 1), \\ 0(1 \times 3), & 1(1 \times 1) \end{bmatrix};$$

$${}^{01}T = \begin{bmatrix} R_z(q_1)(3 \times 3), & {}^{10}D(3 \times 1), \\ 0(1 \times 3), & 1(1 \times 1) \end{bmatrix};$$

$$R_z(q_1) = \begin{bmatrix} \cos(q_1), & -\sin(q_1), & 0, \\ \sin(q_1), & \cos(q_1), & 0, \\ 0, & 0, & 1 \end{bmatrix};$$

$${}^{10}D = \begin{bmatrix} 0, \\ 0, \\ 0 \end{bmatrix};$$

$${}^{12}T = \begin{bmatrix} I(3 \times 3), & {}^{21}D(3 \times 1), \\ 0(1 \times 3), & 1(1 \times 1) \end{bmatrix};$$

$$I = \begin{bmatrix} 1, 0, 0, \\ 0, 1, 0, \\ 0, 0, 1 \end{bmatrix};$$

$${}^{21}D = \begin{bmatrix} l_1, \end{bmatrix}$$

0,

0 ];

$${}^2E_T = [ I(3 \times 3), \quad E_{2D}(3 \times 1),$$

$$0(1 \times 3), \quad 1(1 \times 1) ];$$

$$E_{2D} = [ l_2 + q_2,$$

0,

0 ];

$${}^0E_T = [ \cos(q_1), -\sin(q_1), 0, \cos(q_1) \cdot l_1 + \cos(q_1) \cdot (l_2 + q_2),$$

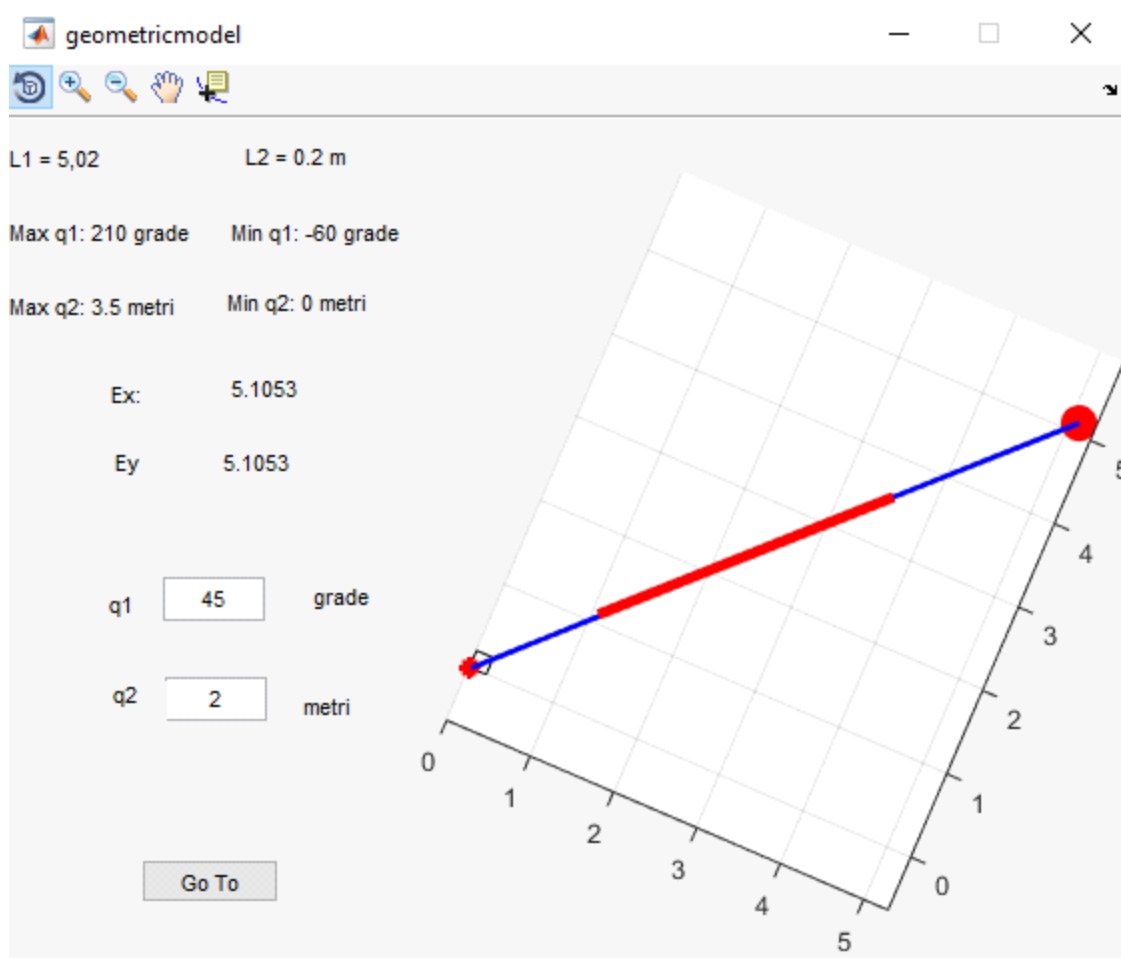
$$\sin(q_1), \cos(q_1), 0, \sin(q_1) \cdot l_1 + \sin(q_1) \cdot (l_2 + q_2),$$

$$0, \quad 0, \quad 1, \quad 0,$$

$$0, \quad 0, \quad 0, \quad 1 \quad ];$$

de unde rezulta punctul end effectorului:

$$E(x,y) = [ \cos(q_1) \cdot l_1 + \cos(q_1) \cdot (l_2 + q_2), \sin(q_1) \cdot l_1 + \sin(q_1) \cdot (l_2 + q_2) ];$$



### 3. Modelarea cinematica a structurii

#### 3.1 Cinematica directa

Modelarea cinematica presupune cunoasterea valorilor finale si initiale ale cuplelor din care rezulta incrementarea (schimbarea) la un moment de timp de la valoarea initiala la valoare urmatoare ( $dq = q_{final} - q_{init}$ ) a unei cuple, cu ajutorul acestora putem formula Jacobian-ului in cinematica directa pentru a afla viteza end effectorului la fiecare miscare.

Folosind formula Jacobianului avem:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = J * \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix};$$

Unde J este matricea Jacobianului de forma:

$$J = \begin{bmatrix} dp(f_1(q))/dp(q_1), dp(f_1(q))/dp(q_2), \\ dp(f_2(q))/dp(q_1), dp(f_2(q))/dp(q_2) \end{bmatrix};$$

Unde f1 si f2 sunt functii ce primesc ca input valorile finale ale cuplelor si returneaza proprietatea X si respectiv Y a punctului end effectorului.

$$f_1(q) = \cos(q_1) * l_1 + \cos(q_1) * (l_2 + q_2);$$

$$f_2(q) = \sin(q_1) * l_1 + \sin(q_1) * (l_2 + q_2);$$

$$dp(f_1(q))/dp(q_1) = -\sin(q_1) * l_1 - \sin(q_1) * (l_2 + q_2);$$

$$dp(f_1(q))/dp(q_2) = \cos(q_1);$$

$$dp(f_2(q))/dp(q_1) = \cos(q_1) * l_1 + \cos(q_1) * (l_2 + q_2);$$

$$dp(f_2(q))/dp(q_2) = \sin(q_1);$$

Din acestea rezulta:

$$\begin{bmatrix} -\sin(q_1) * l_1 - \sin(q_1) * (l_2 + q_2), \cos(q_1), \\ \cos(q_1) * l_1 + \cos(q_1) * (l_2 + q_2), \sin(q_1) \end{bmatrix} * \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \end{bmatrix};$$

$$V_x = (-\sin(q_1) * l_1 - \sin(q_1) * (l_2 + q_2)) * dq_1 + \cos(q_1) * dq_2;$$

$$V_y = (\cos(q_1) * l_1 + \cos(q_1) * (l_2 + q_2)) * dq_1 + \sin(q_1) * dq_2;$$

$$V = \sqrt{(V_x)^2 + (V_y)^2};$$

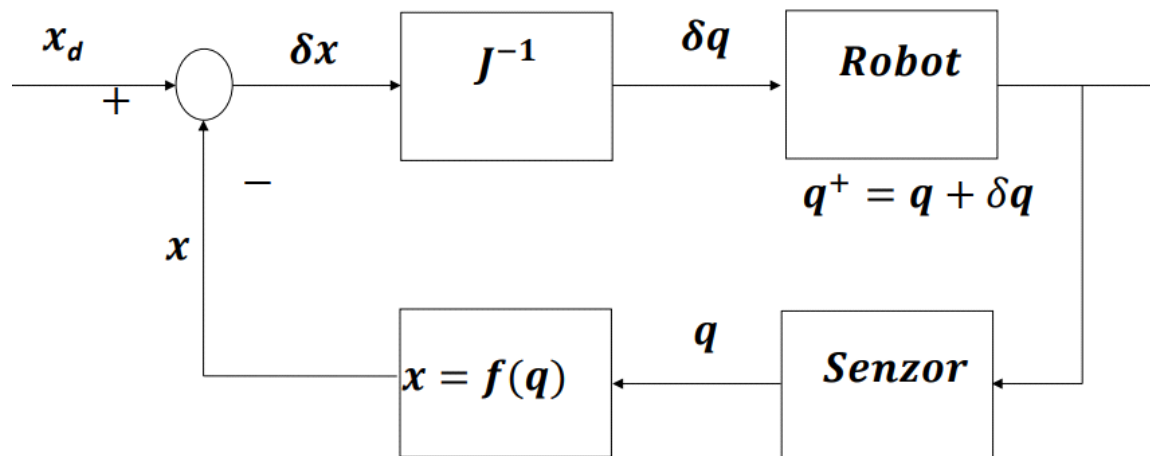
q1, q2 fiind functii ce depind de timp.

### 3.2 Cinematica inversa

Presupune cunoasterea vitezelor si a valorilor initiale si finale din cuple pentru deductia noilor valorilor din acestea. Cinematica inversa pentru controlul manipuloarelor se foloseste intr-o bucla inchisa pentru a putea ajunge la valorile dorite din cuple ( $x_d$ ).

$$\delta x = x_d - x$$

$$\delta q = J^{-1} \delta x$$



Formularea Jacobian-ului in cinematica inversa:

$$\begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix} = (J)^{-1} * \begin{bmatrix} V_x \\ V_y \end{bmatrix};$$

$$(J)^{-1} = 1/\det(J) * \text{adj}(J);$$

$\det(J) \neq 0$  (daca  $\det(J)=0$  inseamna ca avem un punct de singularitate si suntem inafara volumului de lucru al robotului, prin urmare nu se poate construi inversa Jacobianului);

$$\det(J) = (-\sin^2(q_1) * l_1 - \sin^2(q_1) * (l_2 + q_2)) - (\cos^2(q_1) * l_1 + \cos^2(q_1) * (l_2 + q_2));$$

$$\text{adj}(J) = \begin{bmatrix} \sin(q_1) & -\cos(q_1) \end{bmatrix};$$



$$[-\cos(q_1)l_1 - \cos(q_1)(l_2 + q_2), -\sin(q_1)l_1 - \sin(q_1)(l_2 + q_2)];$$

de unde rezulta:

$$(J)^{-1} = 1/((- \sin^2(q_1) * l_1 - \sin^2(q_1) * (l_2 + q_2)) - (\cos^2(q_1) * l_1 + \cos^2(q_1) * (l_2 + q_2))) * \text{adj}(J);$$

### 3.4 Definirea traiectoriilor

Se definește traiectoria din spațiul task ca fiind o linie dreaptă între două puncte  $P_0(x_0, y_0)$  și  $P_f(x_f, y_f)$ , iar pentru traiectoriile din spațiul articular se definește o funcție polinomială de gradul 3 în funcție de timp pentru fiecare cupla  $i$  (1 sau 2):

$$q_i(T) = a_{i,0} + a_{i,1} * T + a_{i,2} * (T)^2 + a_{i,3} * (T)^3;$$

$$\dot{q}_i(T) = a_{i,1} + a_{i,2} * T + a_{i,3} * (T)^2;$$

$$\ddot{q}_i(T) = a_{i,2} + a_{i,3} * T;$$

Unde se cunosc valorile și vitezele din cuple la momentul initial și la cel final ( $q_{1init}$ ,  $q_{1final}$ ,  $q_{2init}$ ,  $q_{2final}$ ,  $\dot{q}_{1init}$ ,  $\dot{q}_{1final}$ ,  $\dot{q}_{2init}$ ,  $\dot{q}_{2final}$ ).

La  $T = 0$  avem:

$$q_i(0) = a_{i,0}, \text{ din care rezulta } a_{i,0} = q_{i,init};$$

$$\dot{q}_i(0) = a_{i,1}, \text{ din care rezulta } a_{i,1} = \dot{q}_{i,init};$$

La  $T = t_{final}$  avem:

$$q_i(t_{final}) = q_{i,final};$$

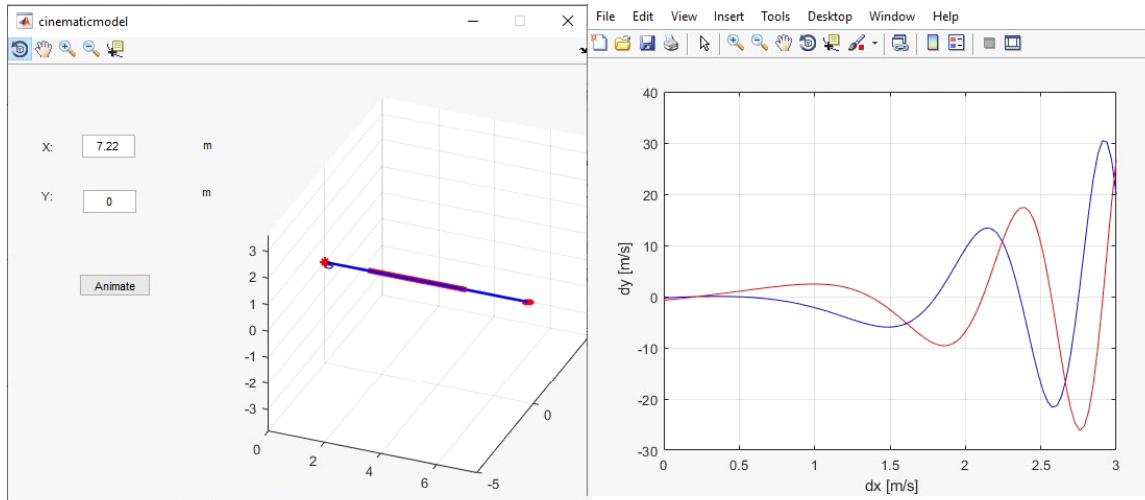
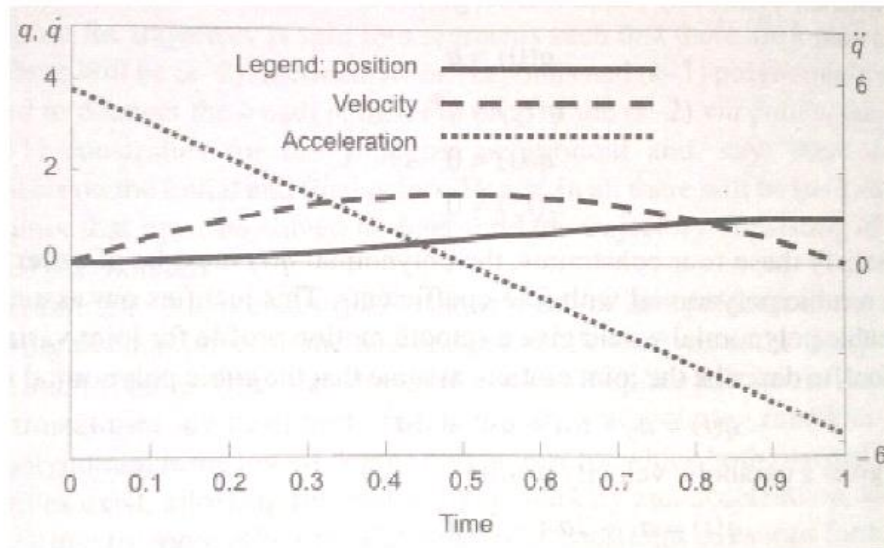
$$\dot{q}_i(t_{final}) = \dot{q}_{i,final};$$

Din acestea rezulta:

$$a_{i,2} = (3/(t_{final})^2) * (q_{i,final} - q_{i,init}) - (1/t_{final}) * (\dot{q}_{i,final} + 2 * \dot{q}_{i,init});$$

$$a_{i,3} = (2/(t_{final})^3) * (q_{i,init} - q_{i,final}) + (1/(t_{final})^2) * (\dot{q}_{i,final} + \dot{q}_{i,init});$$

# Cubic Polynomial (zero speed at end-points)



#### 4. Modelarea dinamica a structurii

Fortele ce afecteaza structura sunt: T (momentul motor), F (forta motoare). Generalizand pe acestea avem  $Q_{1,2}$ .

Folosind formularea Lagrangian-ului:

$$L(q, dq) = E_c(q, dq) - E_p(q);$$

Forta gravitationala G fiind perpendiculara pe planul in care opereaza robotul se considera energia potentiala egala cu 0 ( $E_p = 0$ ) si rezulta:

$$L(q, dq) = E_c(q, dq);$$

$$L = \left( \frac{1}{2} * m_1 * (V_1)^2 + \frac{1}{2} * MI_1 * (dq_1)^2 \right) + \left( \frac{1}{2} * m_2 * (V_2)^2 + \frac{1}{2} * MI_2 * (dq_2)^2 \right);$$

$$MI_1 = m_1 * (l_1)^2 - \text{Momentul de inertie 1};$$

$$MI_2 = m_2 * (l_2 + q_2)^2 - \text{Momentul de inertie 2};$$

$$(V_1)^2 = (l_1)^2 * (dq_1)^2 - \text{viteza corp 1};$$

$$(V_2)^2 = (dq_2)^2 + (dq_1)^2 * (q_2)^2 - \text{viteza corp 2};$$

Rezulta:

$$L = m_1 * (l_1)^2 * (dq_1)^2 + \frac{m_2}{2} * \left( (dq_2)^2 + (dq_1)^2 * (q_2)^2 + (dq_2)^2 * \left( (l_2)^2 + 2 * l_2 * q_2 + (q_2)^2 \right) \right);$$

$$\frac{d}{dt} \left( \frac{dp(L)}{dp(dq_{1,2})} \right) - \frac{dp(L)}{dp(q_{1,2})} = Q_{1,2};$$

$$\frac{dp(L)}{dp(dq_1)} = dq_1 * (2 * m_1 * (l_1)^2 + m_2 * (q_2)^2);$$

$$\begin{aligned} \frac{d}{dt} \left( \frac{dp(L)}{dp(dq_1)} \right) &= d^2 q_1 * (2 * m_1 * (l_1)^2 + m_2 * (q_2)^2) \\ &+ 2 * dq_1 * m_2 * q_2; \end{aligned}$$

$$\frac{dp(L)}{dp(q_1)} = 0;$$

din acestea rezulta:

$$T = d^2q_1 * ( 2 * m_1 * (l_1)^2 + m_2 * (q_2)^2) + 2 * dq_1 * m_2 * q_2;$$

$$dp(L)/dp(dq_2) = dq_2 * (m_2 + m_2 * (l_2)^2 + 2 * l_2 * q_2 + (q_2)^2);$$

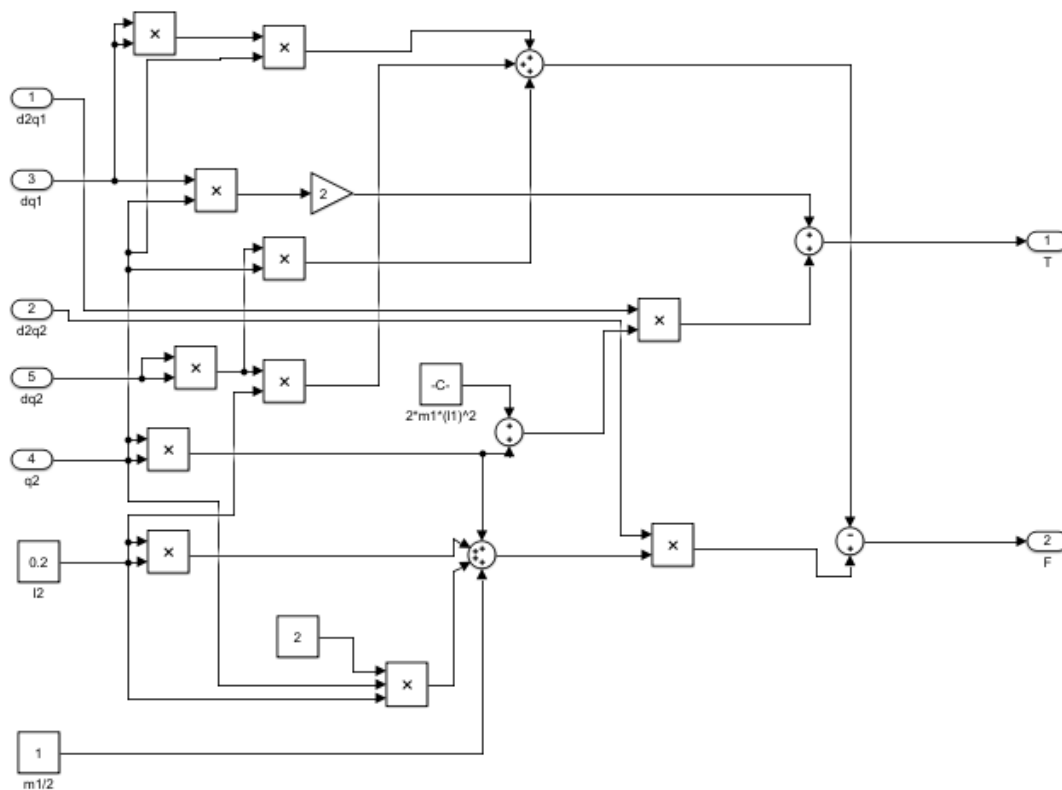
$$d/dt(dp(L)/dp(dq_2)) = d^2q_2 * (m_2 + m_2 * (l_2)^2 + 2 * l_2 * q_2 + (q_2)^2);$$

$$dp(L)/dp(q_2) = m_2 * ( (dq_1)^2 * q_2 + (dq_2)^2 * l_2 + (dq_2)^2 * q_2);$$

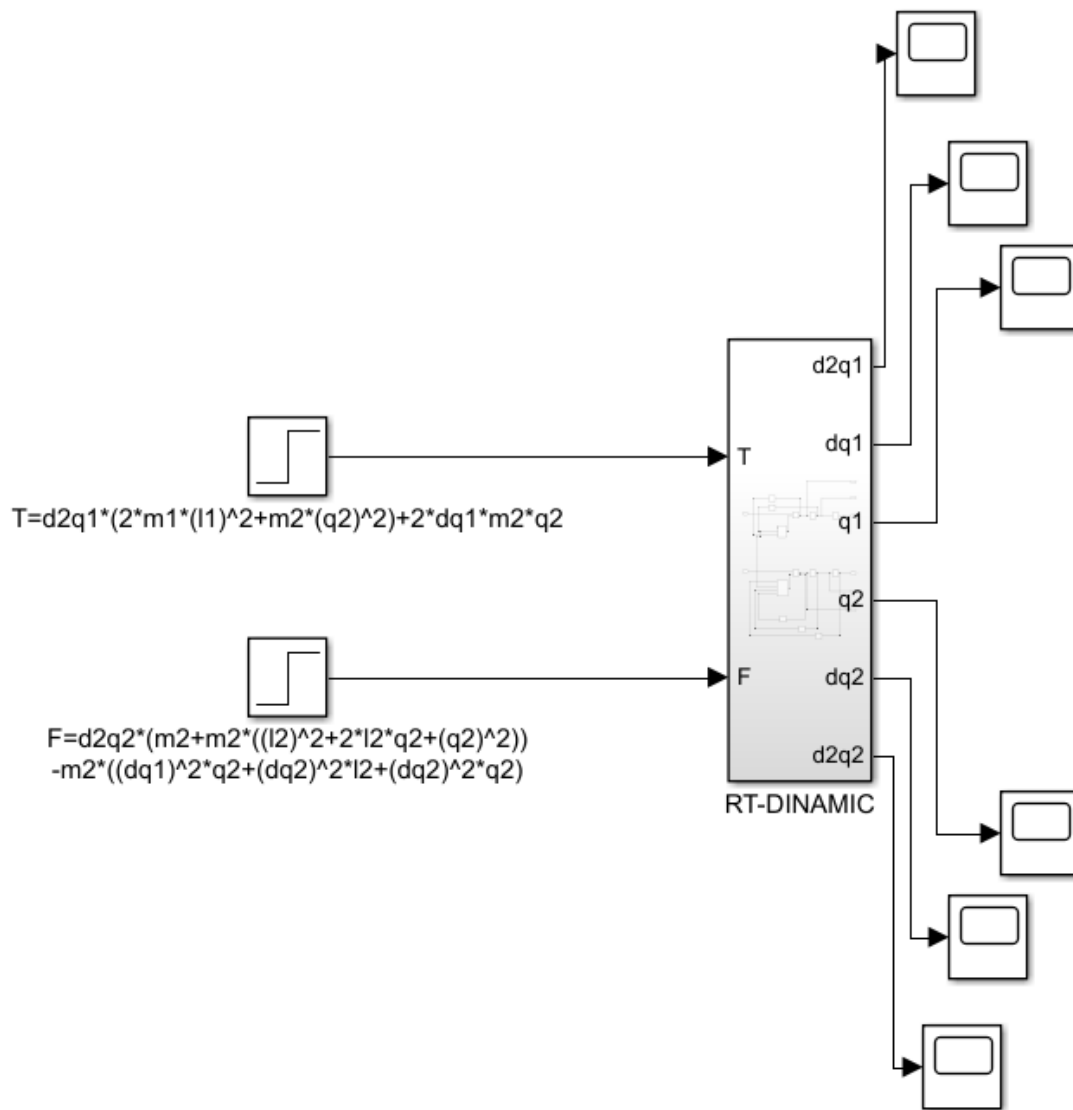
din acestea rezulta:

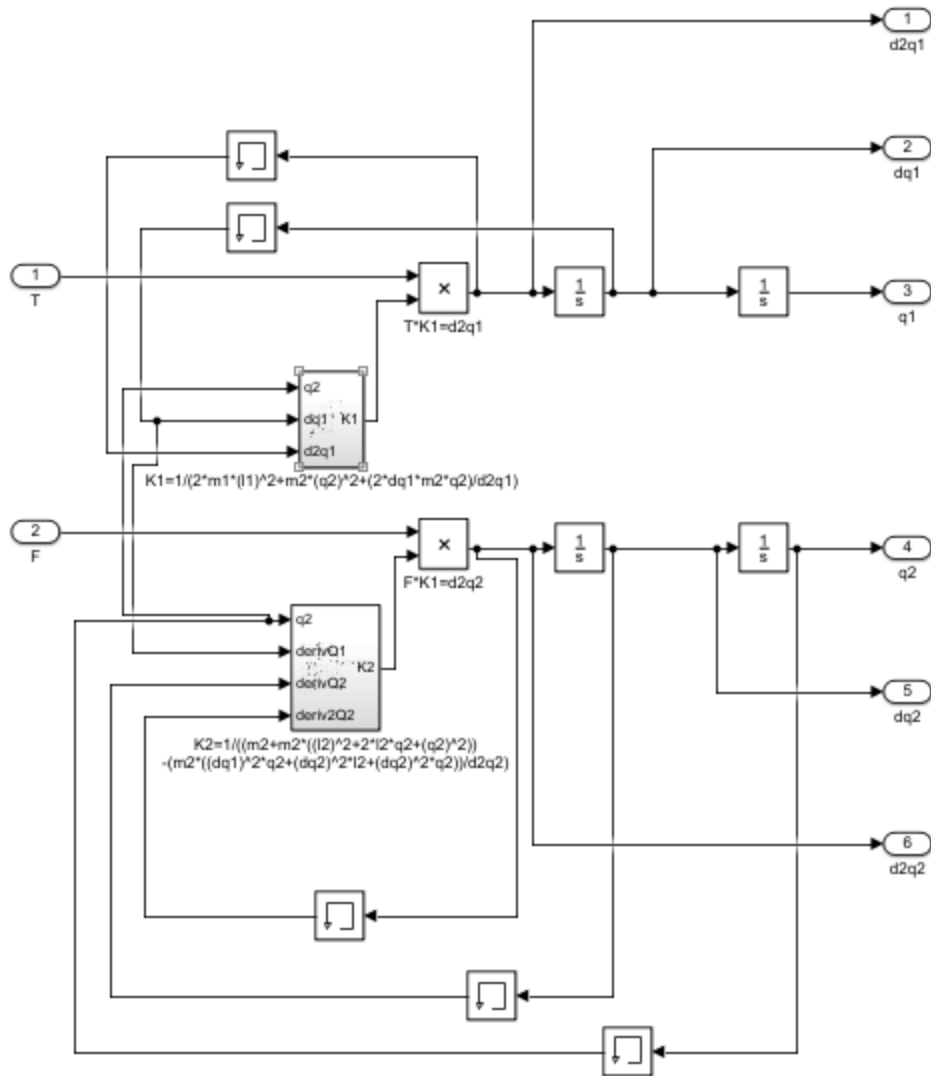
$$F = d^2q_2 * ( m_2 + m_2 * ( l_2)^2 + 2 * l_2 * q_2 + (q_2)^2 ) \\ - m_2 * ( (dq_1)^2 * q_2 + (dq_2)^2 * l_2 + (dq_2)^2 * q_2);$$

*Simulink: robot RT model dinamic direct*

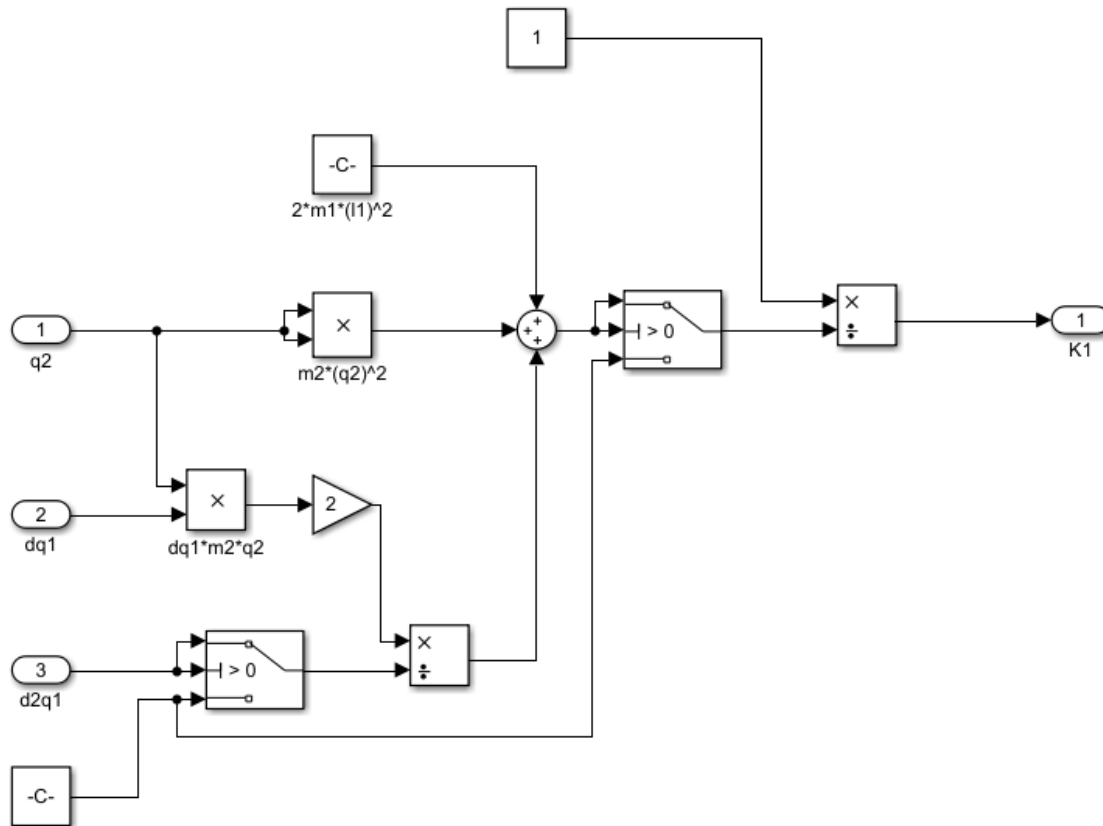


## Simulink: robot RT model dynamic invers

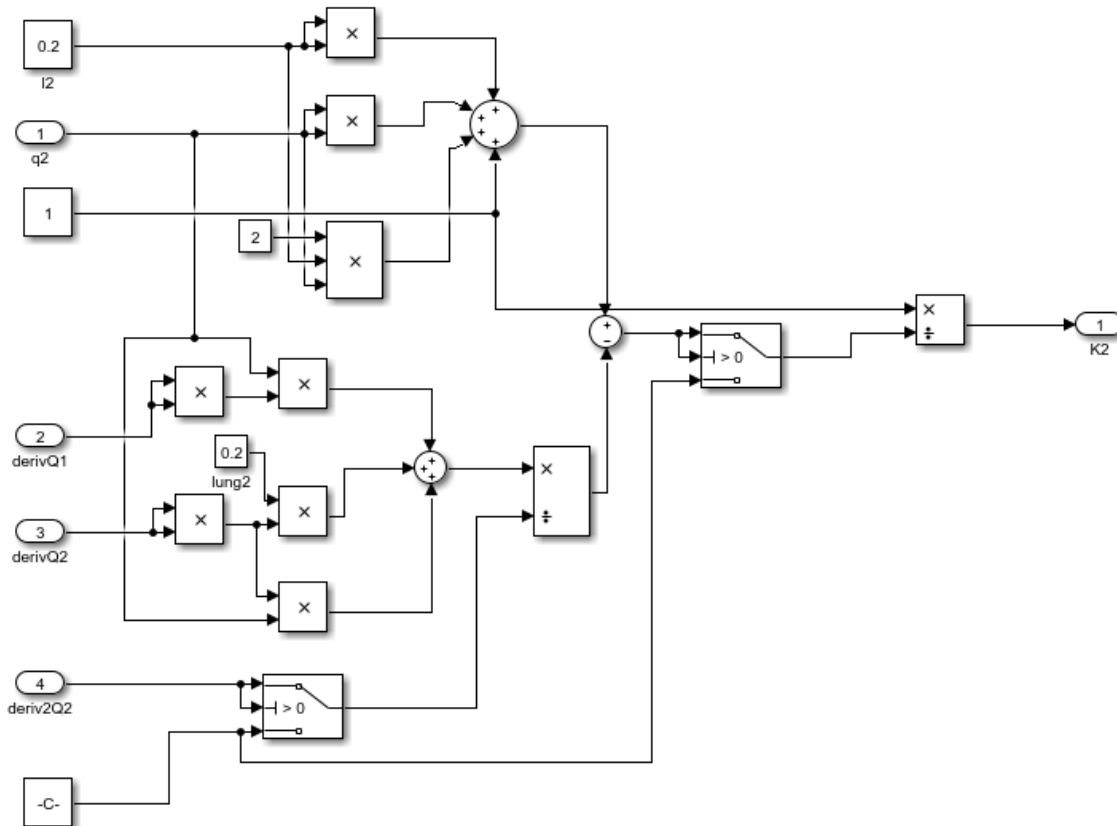




*Simulink: robot RT model dinamic invers, termenul K1*



Simulink: robot RT model dinamic invers, termenul  $K_2$



## 5. Controlul sistemului robotic

### 5.1 Etape parcurse

**Task plan** - obiectivul robotului este de a desena o linie dreapta din pozitia  $q_1=0$ ,  $q_2=0$  pana la  $E_x=7$ ,  $E_y=3$ .

**Action plan** - se genereaza prin interpolarea liniei un numar de 70 puncte ce se afla pe dreapta.

**Path plan** - pentru fiecare punct de pe dreapta se calculeaza prin modelul cinematic invers pozitiile si vitezele articulatilor.

**Trajectory plan** - pentru definirea spatiului articular se folosesc ca si pozitii, viteze si acceleratii dorite functiile cubice polinomiale de gradul 3 cu viteze si acceleratii initiale si finale inexistente.

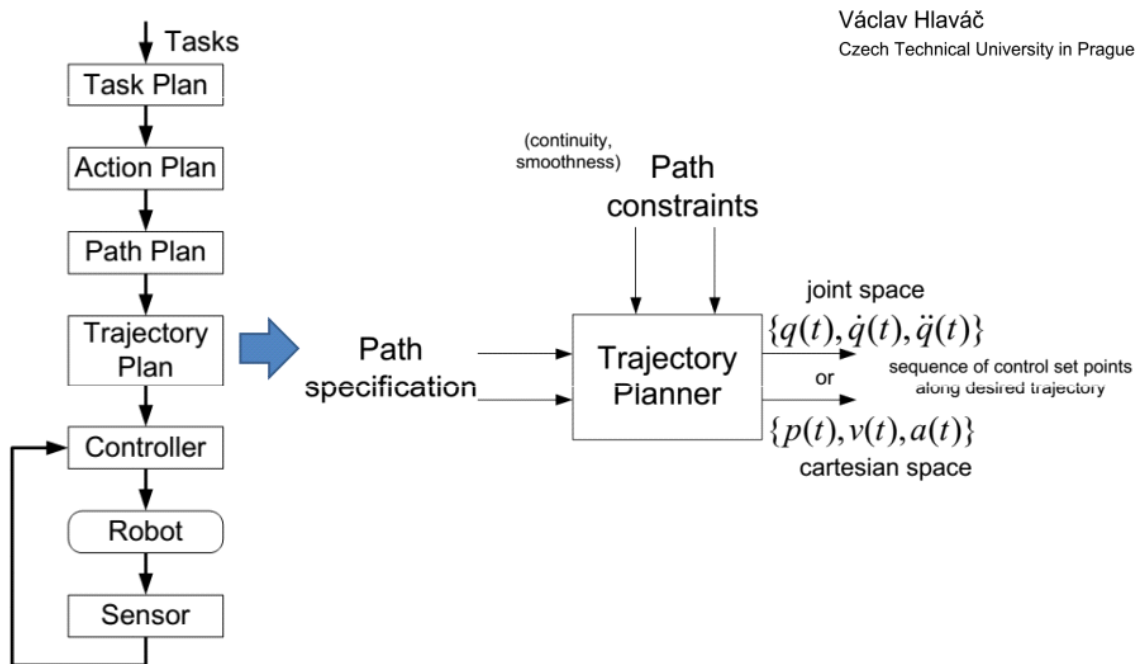


**Controller** - pentru controller-ul motorului a fost folosit un regulator de tip PD (proportional-derivative controller) ce modifica semnalul de iesire in functie de eroarea pozitiei si a vitezei.

**Robot** - este reprezentat de modelul matematic al motorului de cc + reductor + modelul dinamic direct al structurii.

**Sensor** - simulink

### *Etape parcurse*



## 5.2. Regulatorul PID

Regulatorul PID este un mecanism bucla de control care foloseste principiul feedback ce este utilizat pe scara larga in sistemele de control industriale. Un controler PID calculează continuu o valoare de eroare  $e(t)$  ca diferenta intre o valoare dorita si o variabila de proces masurata si aplica o corectie bazata pe proportionare, integrare si derivare, de unde și numele.

Comportamentul dinamic al sistemelor liniare de ordin 2:

$$m \cdot d^2x + b \cdot dx + K \cdot x = V;$$

avand forma canonica:

$$d^2x + 2 \cdot \gamma \cdot w \cdot dx + w^2 \cdot x = u;$$

Aplicand transformata Laplace avem functia de transfer:

$$G(s) = 1 / (s^2 + 2 \cdot \gamma \cdot w \cdot s + w^2);$$

unde:

$$\gamma = b / (2 \cdot \sqrt{K \cdot m}) = 1;$$

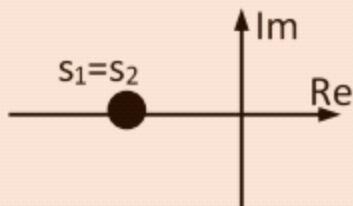
$$w = \sqrt{K / m} = 0.2;$$

Polii  $s_1$  si  $s_2$  dorim sa fie inclusi in multimea numerelor reale,  $s_1 = s_2$  si  $s_2 < 0$  pentru a avea un sistem critic amortizat.

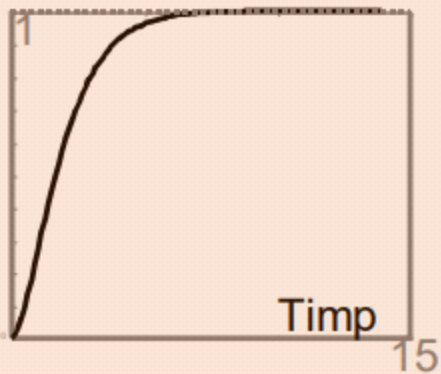
*Sistem critic-amortizat*

### Critic-amortizate

$$\zeta = 1$$



### Aspectul răspunsului indicial



Parametrii de acord  $K_v > -b$  și  $K_p > 0$  îi aflăm cu ajutorul relațiilor lui Viète:

$$s_1 + s_2 = -(K_v + b);$$

$$s_1 \cdot s_2 = K_p + K;$$

din care rezulta:

$$K_v = b - (s_1 + s_2);$$

$$K_p = s_1 \cdot s_2 - K;$$

De unde putem scrie:

$$u = K_v \cdot dx + K_p \cdot x + u^*;$$

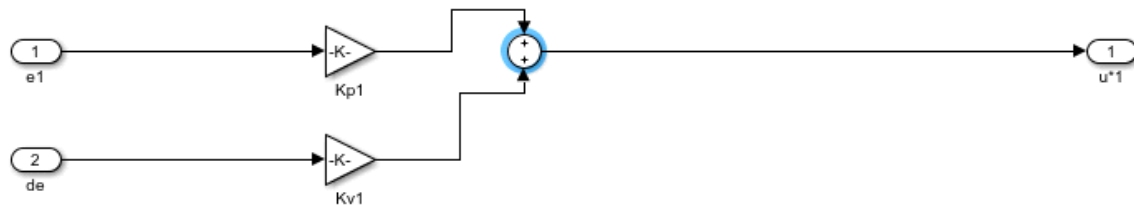
Se alege  $s_1 = -100$ ,  $s_2 = -100$ , din care rezulta:

$K_v = 0.4 - (-200) = 200.4$ ; (termenul de derivare din regulatorul PID)

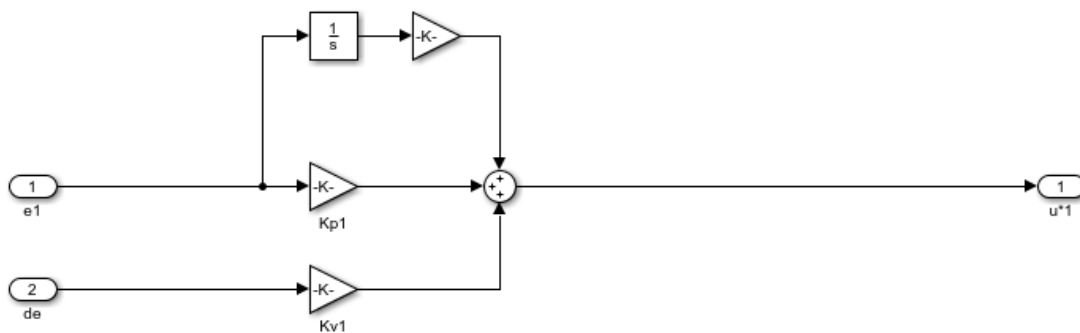
$K_p = 10000 - 0.04 = 10000.04$ ; (termenul de proportionare din regulatorul PID)

Astfel putem construi un controller PID pentru robotul nostru.

*Simulink: regulator PD*



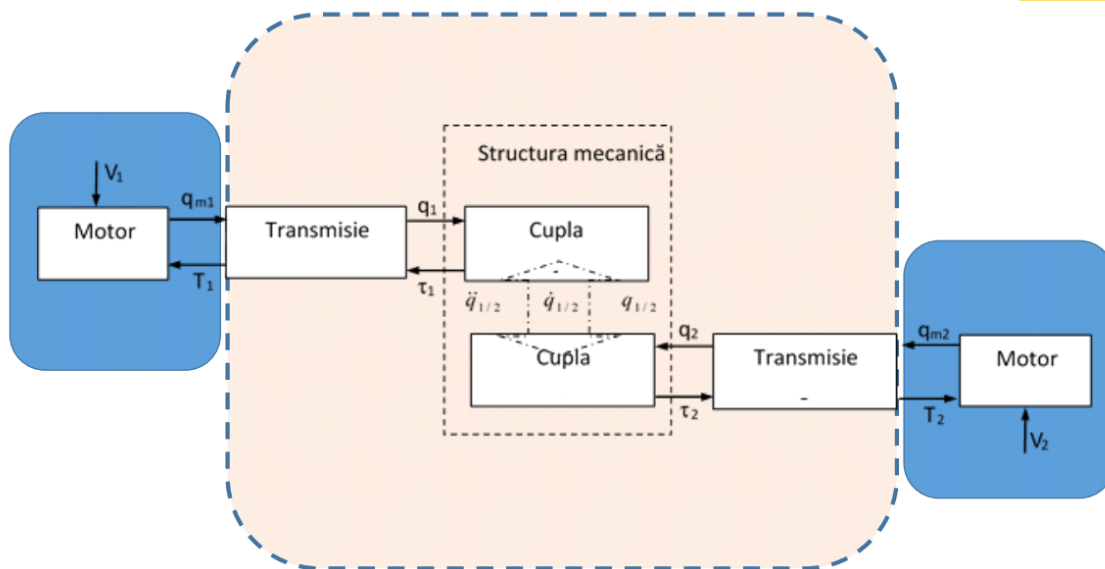
*Simulink: regulator PID*



### 5.3. Controlul modelului dinamic holist

Controlul modelului dinamic holist presupune utilizarea unui regulator PID/PD pentru controlul motorului de curent continuu ce este legat prin transmisie la modelul dinamic direct al robotului.

## Schema bloc structura motor cc



Motorul este definit atat electric cat si mecanic in relatiile urmatoare:

$$V_{1/2} = R_a \cdot i_{1/2} + L \cdot \frac{d i_{1/2}}{dt} + E_{1/2};$$

$$J \cdot d^2 q_{m1/2} = K_m \cdot i_{1/2} - b \cdot dq - T_{1/2};$$

## Ecuatii motorc cc

## Motorul de curent continuu

$$V_{1/2} = R_a i_{1/2} + L \frac{di_{1/2}}{dt} + E_{1/2}$$

$$J \ddot{q}_{m1/2} = K_M i_{1/2} - b \dot{q}_m - T_{1/2}$$

unde:

V este tensiunea de alimentare a motorului;

T este momentul rezistent;

R<sub>a</sub> este rezistența armăturii;

L este inductanța armăturii;

i este intensitatea curentului în armătură;

E este tensiunea contra electromotoare dată de relația

$E = K_b \dot{q}_m$  în care: K<sub>b</sub> este constanta contra electromotoare;

q<sub>m</sub> este poziția rotorului;  $\dot{q}_m = \omega$  este viteza de rotație a rotorului

J este momentul de inerție (rotor + arbore)

K<sub>M</sub>i este cuplul motor: K<sub>M</sub> este coeficientul momentului motor;

b este coeficientul de frecare vâscoasă

$$q_{m1/2} = i_{1/2} * q_{1/2};$$

$$T_{1/2} = 1/(i_{1/2}) * Q_{1/2};$$

Q<sub>1/2</sub> - forțele generalizate de afectează structura (momentul motor + forța motoare);

Motorul ales are un diametru de 32 mm, o putere mecanică de 80 watt și poate fi maxim alimentat la 48V. Reductorul ales este GP 32 BZ de diametru 32 mm și reacție medie de 0.4.

i<sub>R</sub> = 35; (raportul de transmisie)

J = 20 g(cm)<sup>2</sup> = 20\*(10)<sup>(-7)</sup> kg(m)<sup>2</sup>; (momentul de inerție)

K<sub>M</sub> = 40 mNm/A = 40\*(10)<sup>(-3)</sup> Nm/A; (constantă cuplului)

K<sub>b</sub> = 325 rpm/V = 60/325\*2\*pi V/(rad/s); (constantă vitezei)

R<sub>a</sub> = 5.42 ohm; (rezistența armăturii)

b = 0.4; (reacție medie)

$$m_{1,2} = 1 \text{ kg};$$

Putem deduce pentru modelul motorului:

$$K_m \cdot V_1 = (J \cdot i_R + m_2 / i_R) \cdot d^2 q_{1m} + i_R \cdot (b + K_b \cdot K_m / R_a) \cdot dq_{1m} + P_1;$$

$$K_m \cdot V_2 = (J \cdot i_R + (m_1 + m_2) / i_R) \cdot d^2 q_{2m} + i_R \cdot (b + (K_b \cdot K_m) / R_a) \cdot dq_{2m} + P_2;$$

$$(J \cdot i_R + m_2 / i_R) / K_m = 0.71575;$$

$$(i_R \cdot (b + K_b \cdot K_m / R_a)) / K_m = 350.1892;$$

$$(J \cdot i_R + (m_1 + m_2) / i_R) / K_m = 1.43032;$$

$P_1, P_2$  - perturbarea motorului 1, respectiv 2

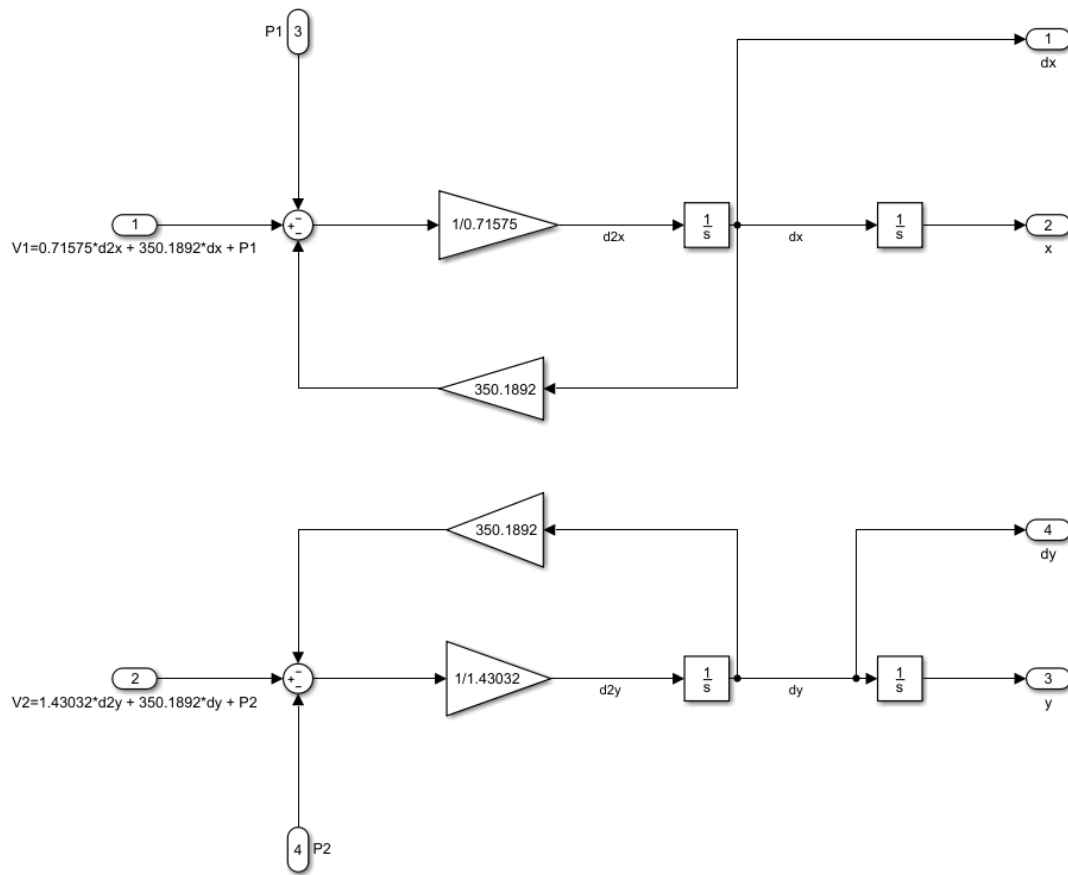
din care rezulta:

$$V_1 = 0.71575 \cdot d^2 q_{1m} + 350.1892 \cdot dq_{1m} + P_1;$$

$$V_2 = 1.43032 \cdot d^2 q_{2m} + 350.1892 \cdot dq_{2m} + P_2;$$

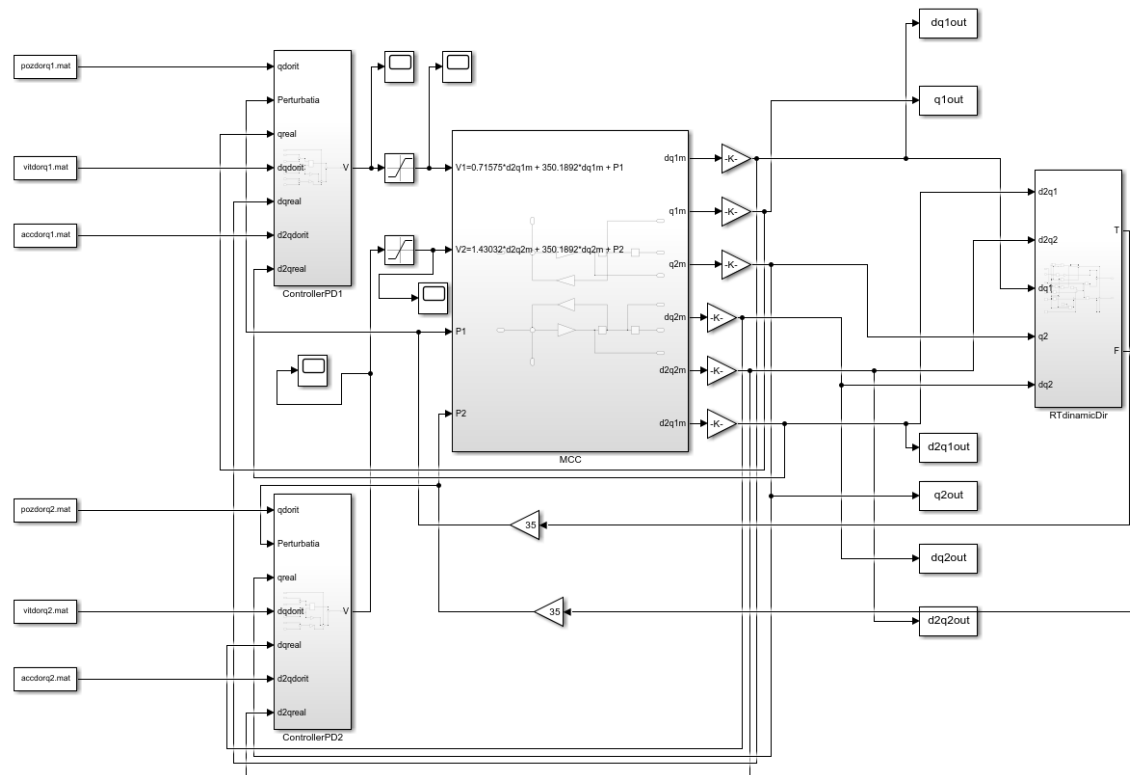
Se va tine cont de alimentarea maxima de 48V.

*Simulink: model motor cc*

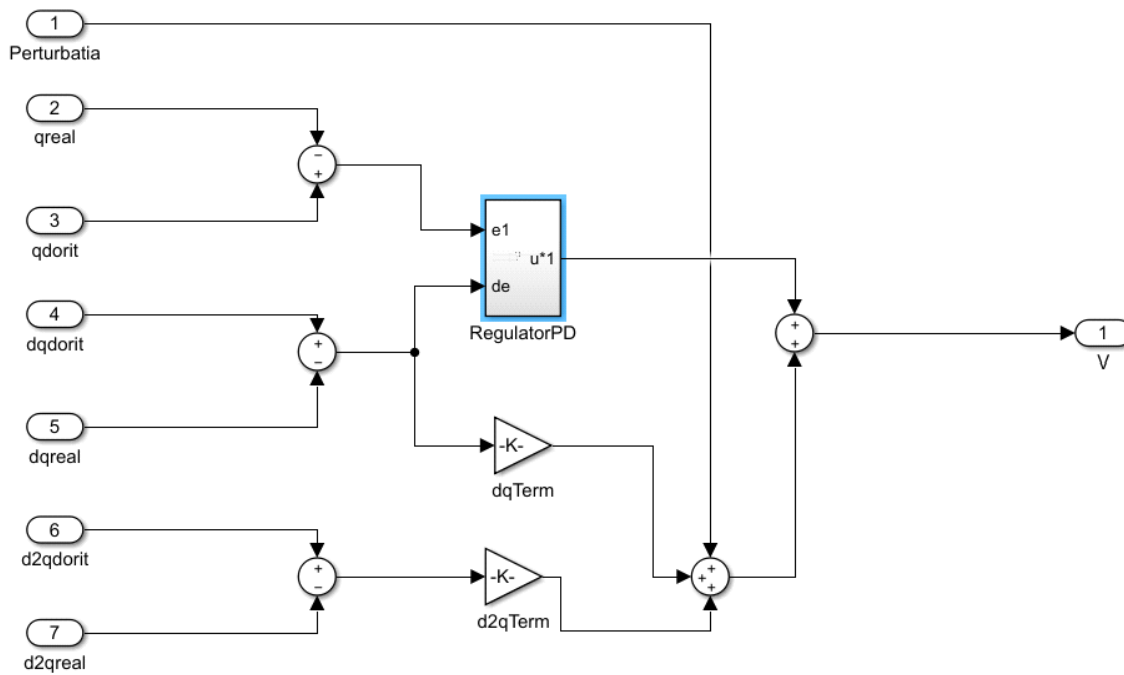


*Simulink: controlul modelului dinamic holist(PD)*

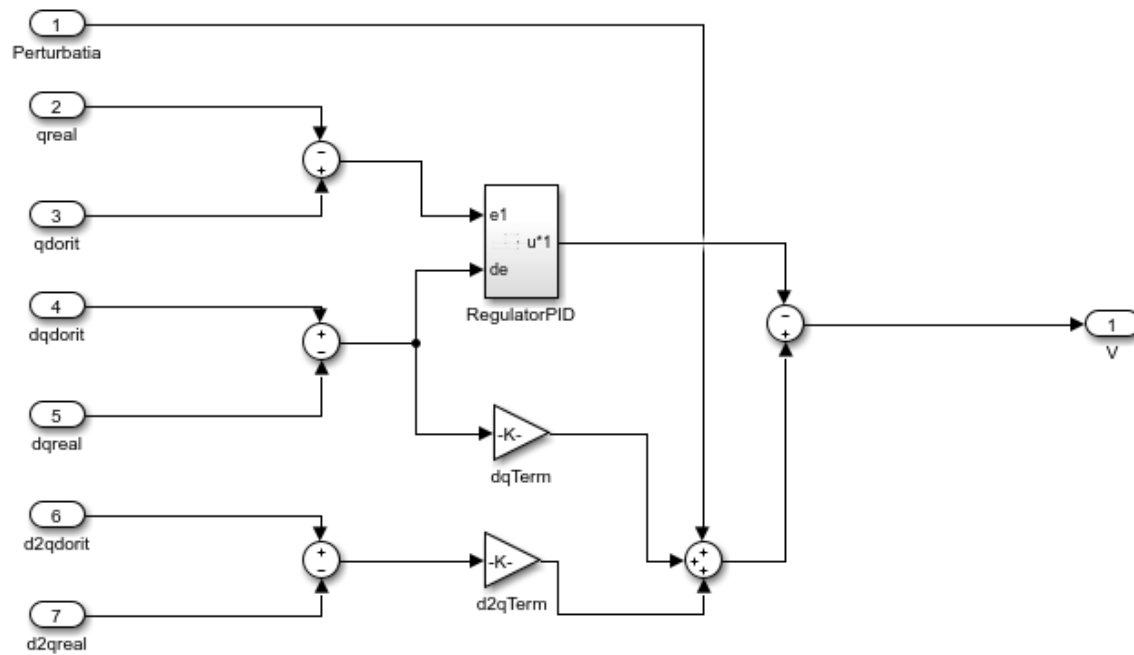




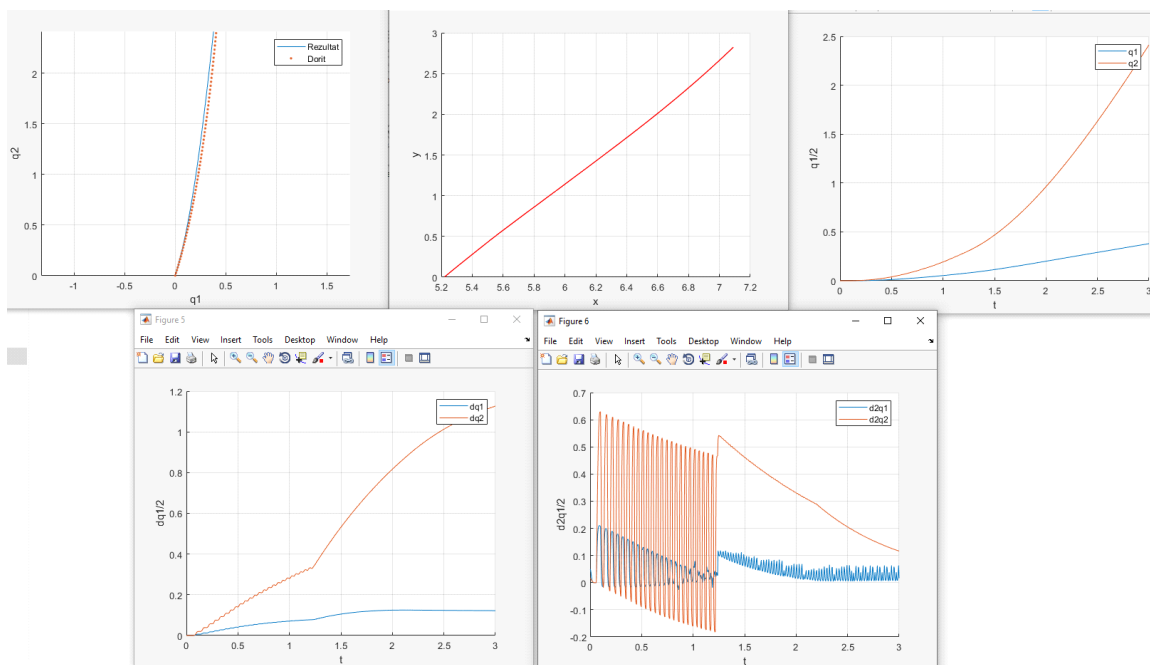
*Simulink: controlul modelului dinamic holist(PD), controller*



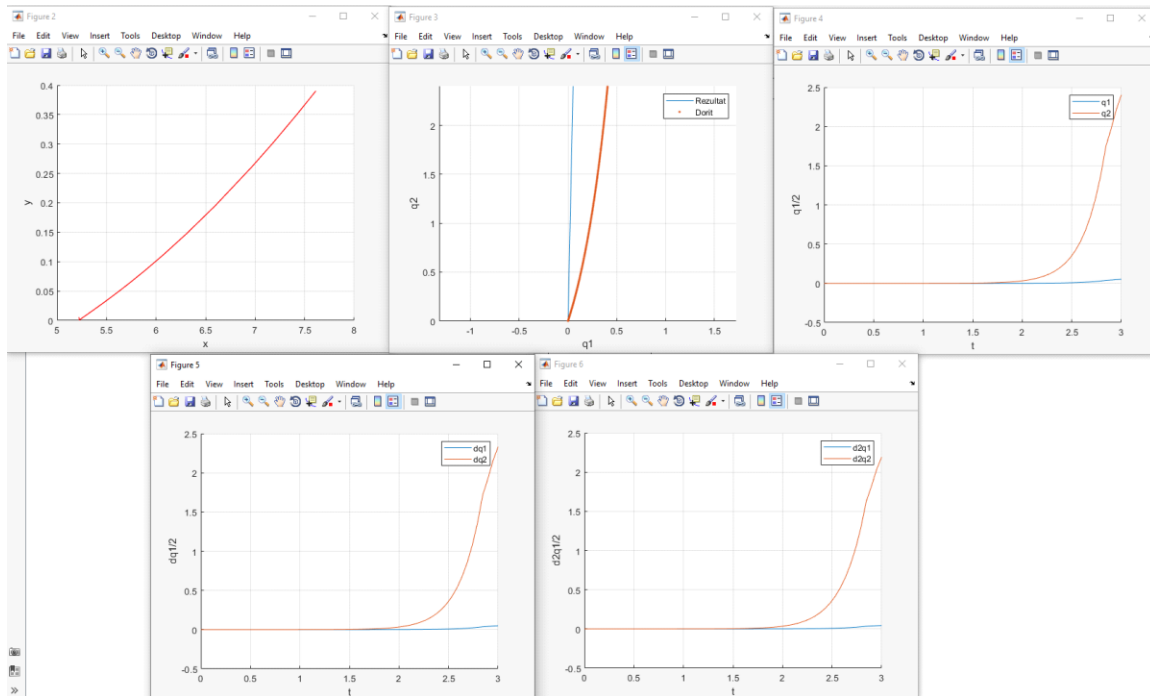
*Simulink: controlul modelului dinamic holist(PID), controller*



*Matlab: rezultate modelul holist PD*



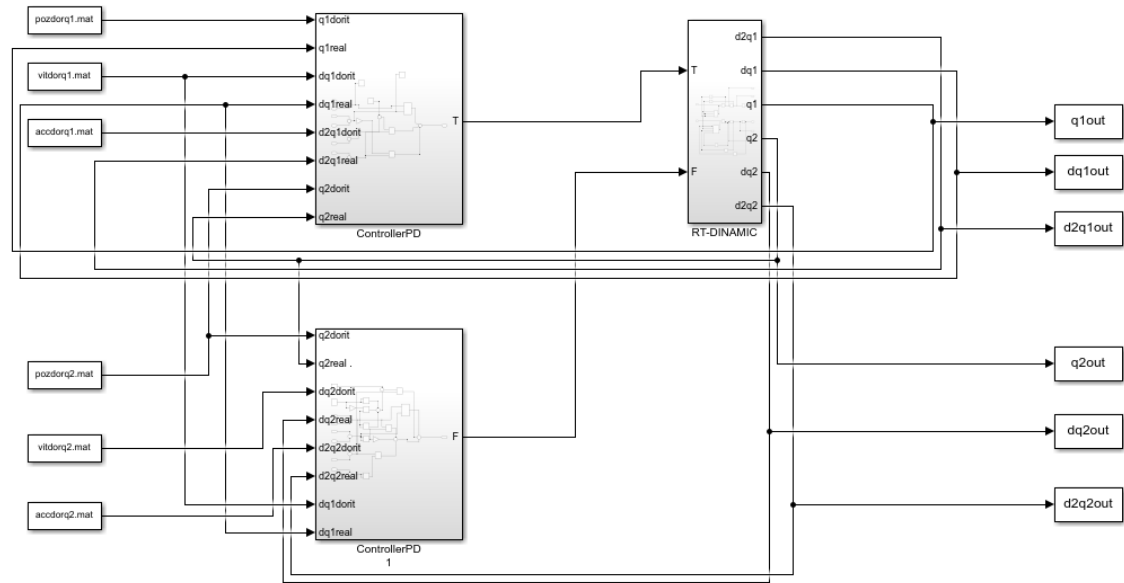
*Matlab: rezultate modelul holist PID*



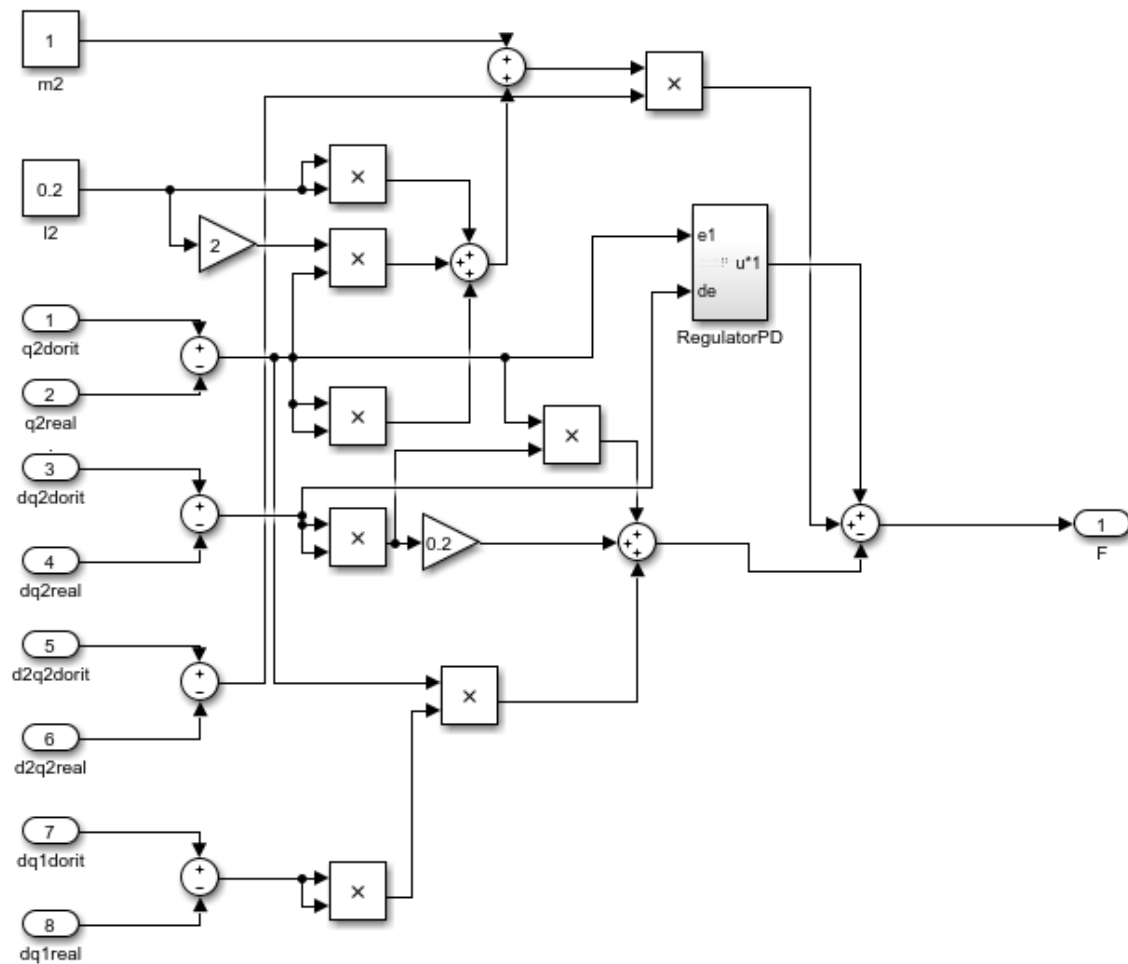
## 5.4 Controlul modelului dinamic invers

Fortele ce afecteaza structura T si F sunt generate de cate un controller ce foloseste regulatorul PID/PD.

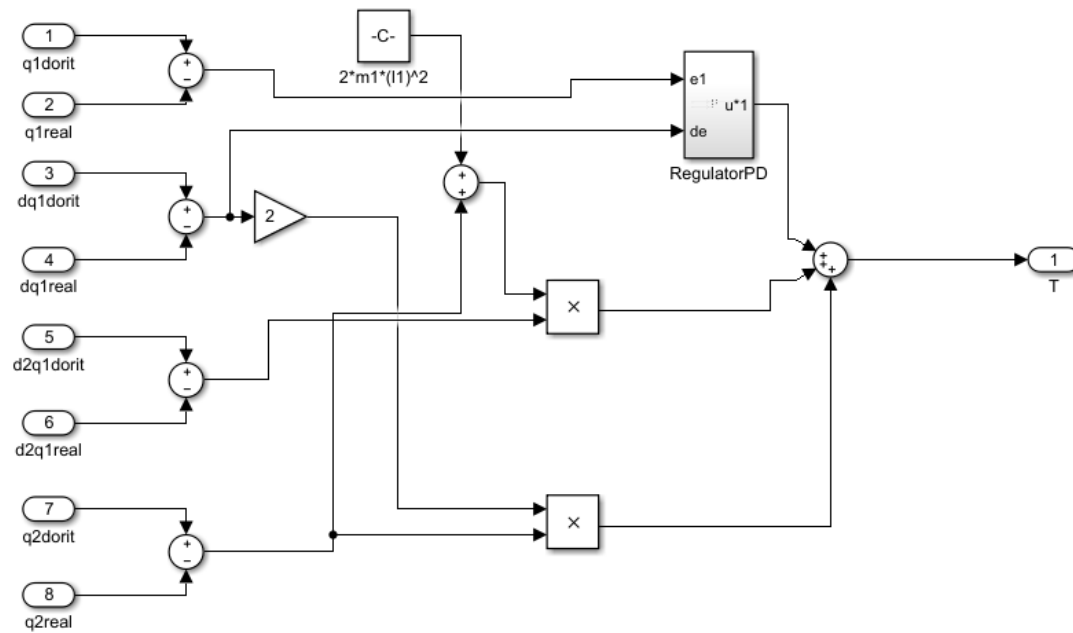
*Simulink: controlul modelului dinamic invers*



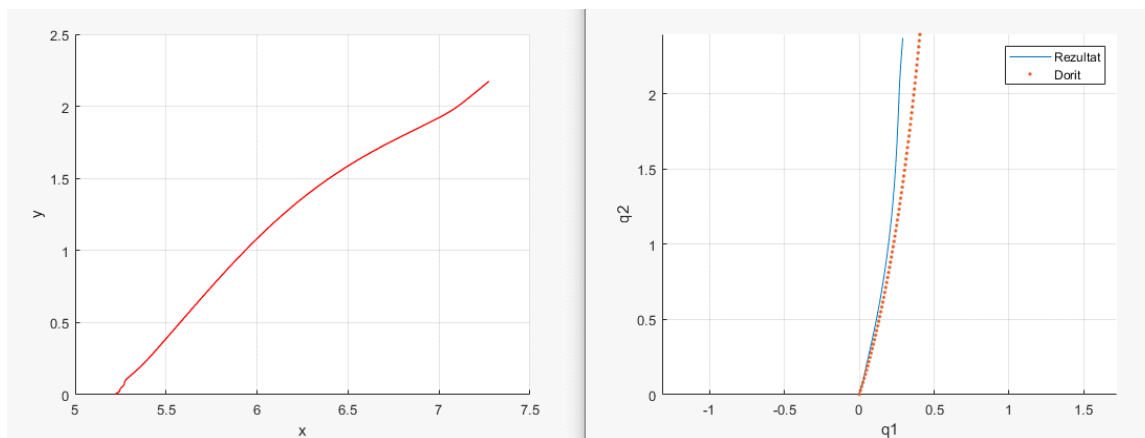
*Simulink: controlul modelului dinamic invers, controller F*

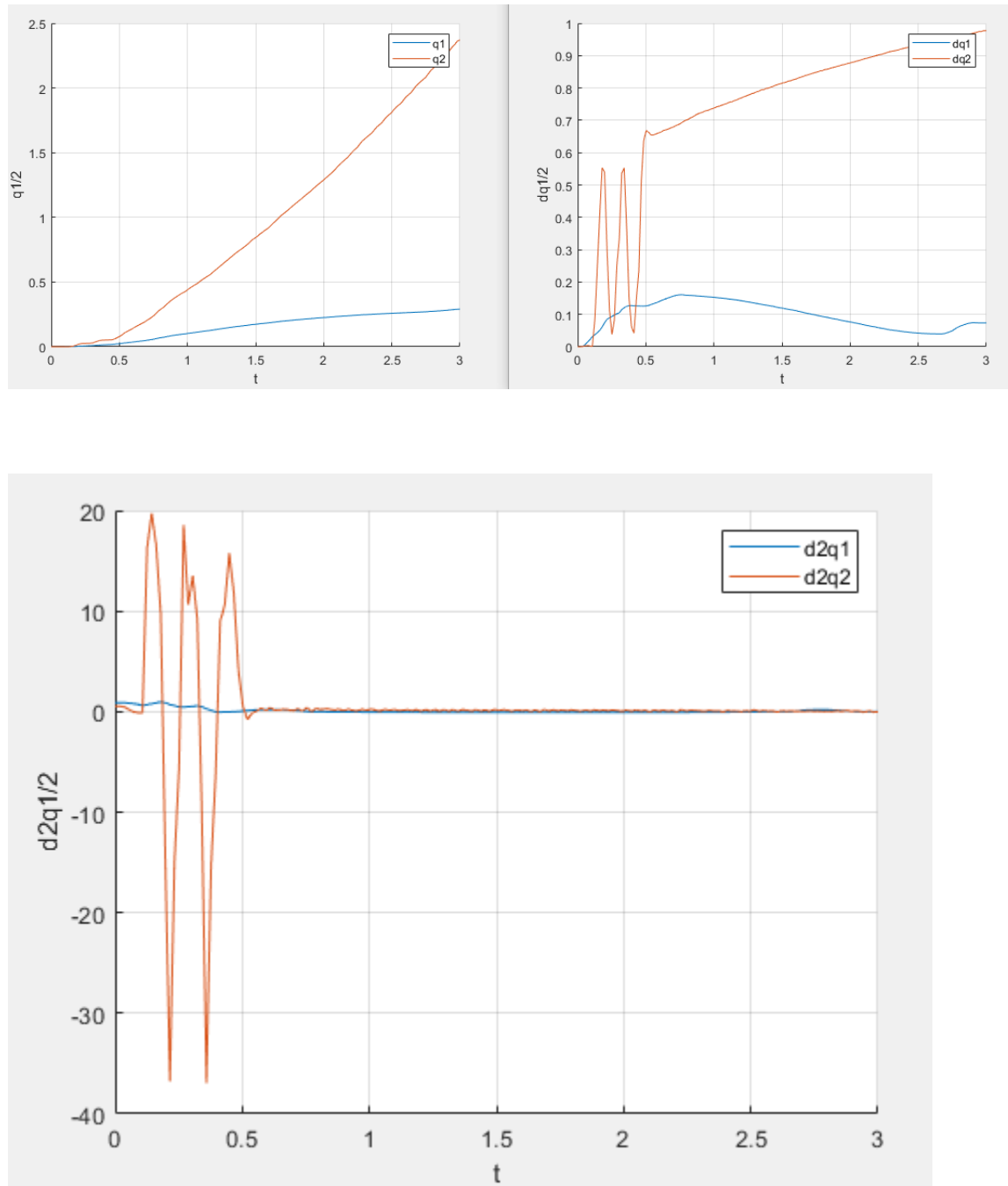


*Simulink: controlul modelului dinamic invers, controller  $T$*



*Matlab: rezultate controlul modelului dinamic invers PD*





*Matlab: rezultate controlul modelului dinamic invers PID*

