
Introducere în modelarea și controlul roboților

Alexandru Codrean



UTPRESS
Cluj-Napoca, 2024
ISBN 978-606-737-723-1

Alexandru Codrean

Introducere în modelarea și controlul roboților



UTPRESS

Cluj-Napoca, 2024

ISBN 978-606-737-723-1



Editura U.T.PRESS
Str. Observatorului, nr. 34
400775 Cluj-Napoca
Tel: 0264-401 999
e-mail:
utpress@biblio.utcluj.ro
biblioteca.utcluj.ro/editura

Recenzia:

Ș.l.dr.ing. Cosmin Marcu
Ș.l.dr.ing. Octavian Ștefan

Pregătire format electronic on-line: Gabriela Groza

Copyright © 2024 Editura U.T.PRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii U.T.PRESS.

ISBN 978-606-737-723-1

Cuprins

Introducere	5
I Modelarea matematică a unui robot	7
1 Modelul geometric	9
1.1 Sisteme de coordonate	9
1.2 Modelul geometric direct	14
1.2.1 Metoda Denavit-Hartenberg	16
1.3 Modelul geometric invers	20
2 Cinematica vitezelor	26
2.1 Jacobianul 2D al unui robot	26
2.2 Jacobianul 3D al unui robot	29
2.3 Inversarea Jacobianului	33
3 Modelul dinamic	35
3.1 Ecuatiile Euler-Lagrange	35
3.2 Modelul dinamic al unui braț robotic cu două cuple de rotație	39
3.3 Modelarea elementelor de execuție	44
3.4 Modelul neliniar al unui robot	47
II Controlul unui robot	49
4 Controlul liniar	51
4.1 Determinarea modelului liniarizat	51
4.1.1 Liniarizarea prin aproximări	51
4.1.2 Metoda generală de liniarizare	52
4.2 Controlul liniar prin reacție după ieșire	54
4.3 Controlul liniar prin reacție după stare	58
4.3.1 Problema de stabilizare	59

Cuprins

4.3.2	Problema de urmărire	61
4.3.3	Estimarea stărilor	65
5	Generarea traiectoriei	68
5.1	Traietorii în spațiul cuplelor	68
5.1.1	Traietoria între două puncte în spațiul cuplelor . . .	69
5.1.2	Traietoria prin mai multe puncte în spațiul cuplelor .	70
5.2	Traietorii în spațiul cartezian	73
6	Controlul neliniar	77
7	Control și estimare pentru roboți mobili	81
	Bibliografie	89

Introducere

Odată cu dezvoltarea și progresul științei și a tehnicii, apar metode și tehnologii tot mai sofisticate, pe niveluri ierarhice multiple, care ne permit să proiectăm, să construim și să controlăm procesele în vederea atingerii anumitor scopuri. Deși în ultimii ani Inteligența Artificială a luat foarte mult avânt, aceasta operează de cele mai multe ori la un nivel ierarhic superior, iar abordările la un nivel ierarhic mai jos (ceea ce a făcut obiectul Automaticii în sens clasic), care interacționează direct cu procesul fizic și îi asigură stabilitatea și funcționarea în anumite condiții critice, tind să fie trecute cu vederea. O categorie de procese în care Automatica a înregistrat un succes spectaculos în ultimii 50 de ani o reprezintă roboții cu bază fixă, iar mai recent roboții mobili. În acest context, cartea de față își propune să sintetizeze într-o manieră introductivă cele mai des întâlnite abordări din Mecanică, Teoria Sistemelor și Ingineria Reglării, utilizate în modelarea, analiza, estimarea și controlul roboților.

Cartea este adresată inginerilor, cercetătorilor, studenților dar și amatorilor entuziaști, care sunt interesați de controlul roboților. Cunoștințele preliminare necesare sunt la nivel de noțiuni de bază de fizică și matematică care se predau în anul 1 al facultăților de inginerie ¹, dar se presupune și că cititorul a fost expus la cel puțin un curs introductiv în Teoria Sistemelor, Semnale și sisteme, sau ceva echivalent. S-a urmărit o prezentare cât mai apropiată de practica inginerescă, fără a insista pe demonstrații teoretice, cu multe exemple și simulări cât mai apropiate de realitate.

Cartea este structurată în două părți. Partea *I* se referă la tot ce ține de modelarea matematică a unui robot, începând de la modelul geometric și cinematic, și continuând cu modelul dinamic. Formalismul Euler-Lagrange va predomina în dezvoltarea modelului dinamic. Partea a *II*-a prezintă metodele de control liniar și neliniar folosite des în practică, insistând pe aspecte care țin de tratarea neliniarităților, dar și pe modalități de generare a traiectoriilor de referință. Dacă majoritatea exemplelor se referă mai mult la roboți cu bază fixă, ultimul capitol discută despre modelarea, estimarea și

¹O foarte bună carte introductivă, sau recapitulativă, în funcție de caz, în matematica pentru ingineri este [12].

Cuprins

controlul roboților mobili, apelând ca studii de caz la un exemplu de robot aerian și un robot terestru.

Toate rezultatele de simulări au fost efectuate în mediul de programare MATLAB/Simulink, dar algoritmi pot fi ușor adaptați și rescriși și în alte limbaje de programare (e.g. C sau Python). Metodele numerice folosite s-au bazat de multe ori pe *Robotics Toolbox* dezvoltat de Peter Corke.

Alexandru Codrean,
Cluj-Napoca, iulie 2024

Partea I

**Modelarea matematică a
unui robot**

Partea *I* descrie modelarea matematică a unui robot. Deși accentul se pune pe roboți cu bază fixă, o mare parte din metodele discutate se pot aplica și la roboții mobili. În Capitolul 1 se discută cum se poate determina modelul geometric direct și invers al unui robot. Pentru modelul geometric direct se va insista pe metoda Denavit-Hartenberg. Pentru modelul geometric invers nu există o metodă generală, ca urmare sunt ilustrate câteva metode analitice particulare, iar apoi este prezentată o abordare numerică a problemei. Capitolul 2 se referă la determinarea Jacobianului robotului, cât și inversarea lui. De interes practic este Jacobianul geometric, valabil în spațiul $3D$, pentru care se prezintă atât abordarea analitică, cât și cea numerică. Determinarea modelului dinamic al robotului prin formalismul Euler-Lagrange este discutată în Capitolul 3. Manipularea modelului pentru a fi adus la o formă standard din punct de vedere al Teoriei Sistemelor este de interes pentru ceea ce va urma în Partea *II*. De asemenea, se vor discuta aspecte legate de implementarea și simularea modelului.

Capitolul 1

Modelul geometric

Pe parcursul cărții ne vom referi predominant la roboții cu bază fixă, în particular la brațul robotic de tip serial. Un astfel de robot este alcătuit dintr-o succesiune de legături (solid rigid) și articulații (cuple). Un exemplu de astfel de robot, cu o cuplă de rotație și una de translație, este ilustrat în Figura 1.1. Prima legătură începe din baza robotului, iar ultima se termină la nivelul efectorului final. Cuplele sunt de regulă acționate de motoare electrice² și realizează mișcări de translație sau rotație³.

Pentru a putea pătrunde în lumea geometrică care descrie un robot, începem cu descrierea diverselor sisteme de coordonate ce se pot adopta, precum și legăturile dintre ele pe baza unor transformări de rotație și translație. Modelul geometric al unui robot face legătura dintre variabilele cuplelor (unghi sau deplasare liniară) și variabilele efectorului final (poziția și orientarea). Altfel spus, modelul geometric face maparea dintre spațiul cuplelor q și spațiul cartezian xyz - Figura 1.2. Determinarea modelului pornește de obicei de la schema constructivă a robotului. Metodele prezentate pentru determinarea modelului geometric direct sau invers vor fi exemplificate pe diverse configurații de braț robotic.

1.1 Sisteme de coordonate

Pentru a descrie mișcarea în spațiul 3D a diferitelor elemente de tip solid rigid ale unui robot, este necesar definirea și atașarea unor sisteme de coordonate. Sistemele de coordonate ne permit descrierea mișcării în termeni de poziție și orientare.

Standardul în Robotică este adoptarea unui sistem de coordonate pe

²Cuplele unui robot pot să fie acționate și pneumatic sau hidraulic.

³În Robotică există cuple și mai complexe, nu doar cele simple de rotație și translație, dar acestea trec dincolo de tema urmărită în această carte.

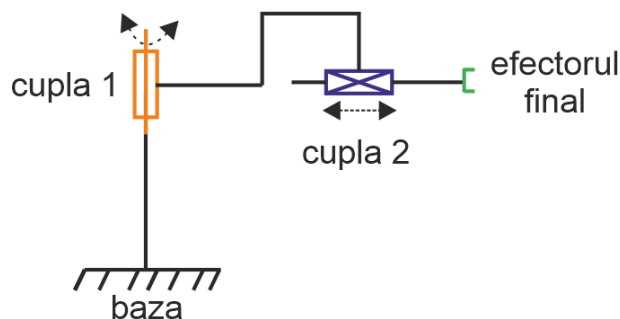


Figura 1.1: Exemplu de braț robotic cu o cuplă de rotație și una de translație

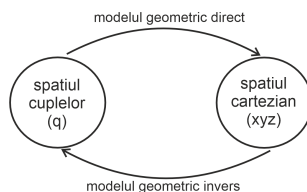


Figura 1.2: Maparea dintre spațiul cuplelor și spațiul cartezian prin modelul geometric

baza regulii mâinii drepte ([24]). Regula ne ajută să determinăm sensul pozitiv pentru fiecare axa, precum în Figura 1.3. Degetul mare ne indica sensul pozitiv pentru axa Z, restul degetelor ne indică sensul pozitiv pentru axa X, iar prin flexia degetelor cu 90 de grade (sens trigonometric față de Z) obținem sensul pozitiv și pentru axa Y.

Decierea poziției unui punct într-un sistem de coordonate este sub forma unui vector coloană $p = [p_x \ p_y \ p_z]^T$. Când vorbim de orientare, ne referim la orientarea unui sistem de coordonate atașat unui solid rigid, în raport cu un alt sistem de coordonate. De exemplu în Figura 1.4, avem două sisteme de coordonate, A și B. Originea sistemului de coordonate B, notată cu O_B , în raport cu originea sistemului A, este dată de un vector de poziție o_B^A , compus din cele trei proiecții ale punctului O_B pe axele $\{x_A, y_A, z_A\}$. Pentru a descrie orientarea sistemului de coordonate B în raport cu A, este necesară definirea unor matrici de rotație (uneori numite și matrici de orientare). O opțiune constă în a ne folosi de unghiurile de rotație din jurul fiecărei axe: pentru x_A avem unghiul ψ (numit *yaw*), pentru y_A avem unghiul θ (numit *pitch*), iar pentru z_A unghiul ϕ (numit *roll*). Astfel

putem defini o matrice de rotație ca o succesiune a trei rotații [31]⁴

$$R = R_{z_A, \phi} R_{y_A, \theta} R_{x_A, \psi} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}. \quad (1.1)$$

Dacă de exemplu cunoaștem poziția unui punct p în sistemul B ($p_B = [x_B \ y_B \ z_B]^T$), și dorim să determinăm poziția față de sistemul A, putem defini o transformare de forma

$$p_A = R_B^A p_B + d_B^A, \quad (1.2)$$

unde d_B^A este distanța dintre originea o_A și originea o_B , iar R_B^A definește o succesiune de trei rotații succesive în jurul axelor x_B, y_B, z_B . Este mai eficient matematic însă să definim maparea din B în A ca o transformare de forma

$$p_A = T_B^A p_B. \quad (1.3)$$

Ca urmare de obicei se adoptă un operator matriceal de dimensiune 4×4 de forma

$$\begin{bmatrix} p_A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & d_B^A \\ 000 & 1 \end{bmatrix} \begin{bmatrix} p_B \\ 1 \end{bmatrix}, \quad (1.4)$$

care se mai numește și transformare omogenă [8]. Dacă dorim să aflăm maparea de la A la B, putem să ne folosim de inversa transformării: $T_A^B = (T_B^A)^{-1}$.

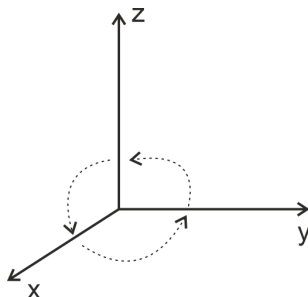


Figura 1.3: Sistemul de coordonate standard în Robotică

Se pot defini astfel 6 transformări omogene elementare, corespunzând celor trei tipuri de translații și rotații:

$$Trans(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & \alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Rot(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.5)$$

⁴Pentru un unghi oarecare α , folosim notațiile prescurtate $\sin(\alpha) = s_\alpha$ și $\cos(\alpha) = c_\alpha$.

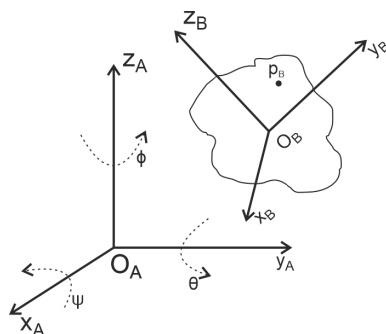


Figura 1.4: Poziția și orientarea unui corp cu ajutorul sistemelor de coordonate

$$Trans(y, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Rot(y, \alpha) = \begin{bmatrix} c_\alpha & 0 & s_\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -s_\alpha & 0 & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.6)$$

$$Trans(z, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix}, Rot(z, \alpha) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 & 0 \\ s_\alpha & c_\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.7)$$

Exercițiul 1. Considerăm că punctul o_B din Figura 1.4 are coordonatele în sistemul A $\{1, 4, 3\}$. Să se aplice următoarea succesiune de transformări: $transl(x_A, 4)$, $rot(x_A, 90^\circ)$, $rot(y_A, -90^\circ)$. Care este noua poziție?

Soluție. Începem prin a defini vectorul poziției inițiale $p_{o_B} = \begin{bmatrix} 1 \\ 4 \\ 3 \\ 1 \end{bmatrix}$.

Aplicarea primei transformări de translație presupune operația

$$p_{o_B}^1 = Trans(x_A, 4) p_{o_B} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 1 \end{bmatrix}.$$

La noul punct obținut, $p_{o_B}^1$, putem aplica a doua transformare de rotație:

$$p_{o_B}^2 = Rot(x_A, 90^\circ) p_{o_B}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90^\circ} & -s_{90^\circ} & 0 \\ 0 & s_{90^\circ} & c_{90^\circ} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \\ 4 \\ 1 \end{bmatrix}.$$

În mod similar se poate aplica și următoarea transformare pentru a obține poziția finală $p_{O_B}^3$. Acest calcul se poate efectua și simultan, având grijă la ordinea efectuării operațiilor ⁵:

$$p_{O_B}^3 = \begin{bmatrix} c_{-90^\circ} & 0 & s_{-90^\circ} & 0 \\ 0 & 1 & 0 & 0 \\ -s_{-90^\circ} & 0 & c_{-90^\circ} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90^\circ} & -s_{90^\circ} & 0 \\ 0 & s_{90^\circ} & c_{90^\circ} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 3 \\ 1 \end{bmatrix} \\ = \begin{bmatrix} -4 \\ -3 \\ 5 \\ 1 \end{bmatrix}.$$

□

Exercițiul 2. Considerăm problema amplasării unui bloc B (paralelipiped) în cadrul unui ansamblu A, ilustrată în Figura 1.5 ([24]). Coordonatele punctelor O_A și O_B sunt date în sistemul de referință R. Ce succesiuni de transformări trebuie aplicate asupra corpului B astfel încât în final (după asamblare) punctul O_B să se suprapună cu O_A , iar punctul s cu punctul p.

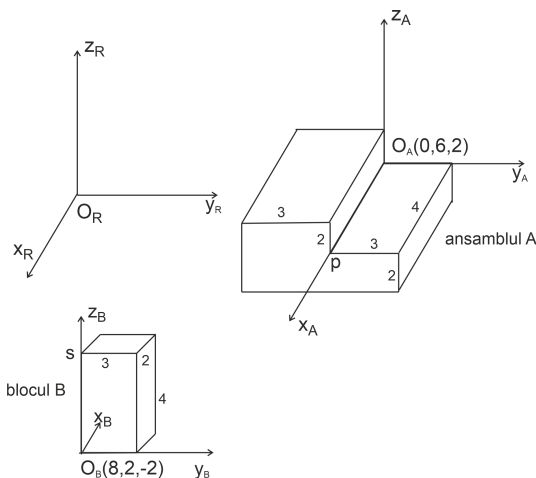


Figura 1.5: Amplasarea blocului B în ansamblul A

Soluție. O posibilă soluție constă în a aduce corpul B la orientarea dorită pentru asamblare, iar apoi asamblarea propriu-zisă constă doar într-o succesiune de trei translații. Pentru orientare este suficientă o rotație în jurul axei y_B (deci în sistemul de coordonate B) cu -90° . Apoi succesiunea de operații de translație în sistemul R, $trans(z, 4)$, $trans(y, 4)$ și $trans(x, -8)$, ne aduc corpul B în poziția dorită în ansamblu. □

⁵Înmulțirea între matrici nu este comutativă.

1.2 Modelul geometric direct

Modelul geometric direct realizează legătura dintre variabilele cuplelor și poziția și orientarea efectorului final. În vederea construirii modelului, vom atașa sisteme de coordonate fiecărei cuple, bazei robotului și efectorului final, iar apoi vom determina legăturile dintre sistemele de coordonate.

Considerăm ca exemplu un robot cu o cuplă de rotație și o cuplă de translație - Figura 1.6. Legătura dintre bază și prima cuplă are lungimea L_0 , iar cea dintre cupla 1 și cupla 2 are lungimea L_1 . Sistemul de coordonate fix (de referință) este cel atașat în baza robotului. Sistemul de coordonate 1 îl plasăm în centrul cuplei 1, iar sistemul de coordonate 2 îl plasăm direct în efectorul final (în loc de cupla 2) din motive de simplitate⁶. Pentru a determina legăturile dintre sistemele de coordonate, ne vom folosi de matricile de transformare omogenă definite în ecuațiile (1.5)-(1.7).

Din sistemul de coordonate 0 ajungem în sistemul de coordonate 1 pe baza unor operații de translație și rotație pe axa Z , definite prin matricea de transformare

$$T_{01} = \begin{bmatrix} c_{q_1} & -s_{q_1} & 0 & 0 \\ s_{q_1} & c_{q_1} & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Din sistemul de coordonate 1 ajungem în sistemul de coordonate 2 prin două translații pe axa Y ⁷

$$T_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 + q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Modelul geometric se obține prin înmulțirea matricelor de transformare

$$T_{02} = T_{01} T_{12} = \begin{bmatrix} c_{q_1} & -s_{q_1} & 0 & -s_{q_1}(L_1 + q_2) \\ s_{q_1} & c_{q_1} & 0 & c_{q_1}(L_1 + q_2) \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{02} & o_{02} \\ 0_{1 \times 3} & 1 \end{bmatrix},$$

și ne oferă poziția (vectorul o_{02} de dimensiune 3×1) și orientarea (matricea R_{02} de dimensiune 3×3) efectorului final în funcție de sistemul de coordonate fix din bază.

⁶A se observa că toate sistemele de coordonate au aceeași orientare.

⁷A se observa ca fiecare matrice de transformare depinde de o singură variabilă, și anume variabila cuplei.

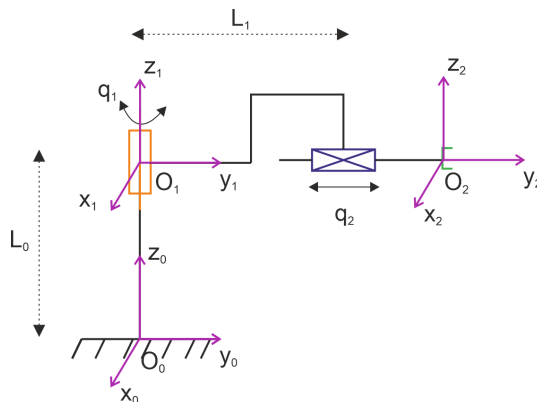


Figura 1.6: Exemplu de braț robotic cu o cuplă de rotație și una de translație

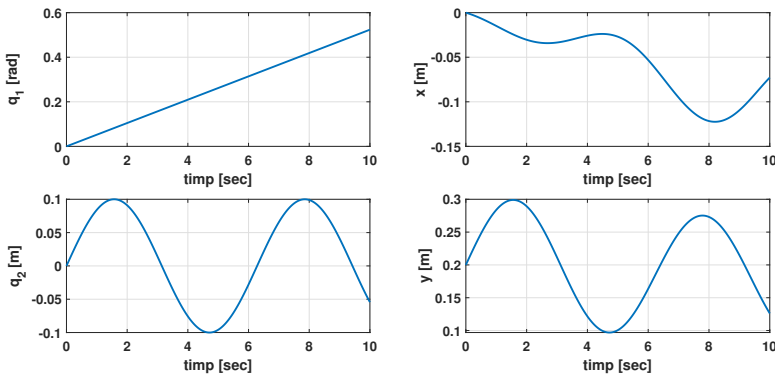


Figura 1.7: Traiectoriile efectorului final (dreapta) în funcție de traiectoriile cuplelor (stânga)

În general, pentru un sistem cu n cuple, modelul geometric direct se calculează ca

$$T_{0n} = T_{01} T_{12} \dots T_{n-1,n}. \quad (1.8)$$

Ca exemplu numeric la robotul din Figura 1.6, considerăm o traiectorie de tip rampă pentru cupla 1 ($q_1 = t \cdot \pi/60$), una sinusoidală pentru cupla 2 ($q_2 = 0.1 \cdot \sin(0.1t)$), și adoptăm valorile parametrilor: $L_1 = 0.2m$, $L_0 = 0.1m$. Pe baza modelului geometric direct se determină traiectoriile din Figura 1.7 (poziția z a efectorului final rămâne constantă la valoarea $0.1m$, ca urmare este omisă din figură).

Exercițiul 3. Să se determine modelul geometric direct pentru robotul din Figura 1.8, care presupune o cuplă de rotație și două cuple de translație. Deoarece cupla de rotație este în bază, sistemul de coordonate fix se supra-

pune cu sistemul de coordonate asociat primei cuple, pentru $q_1 = 0$.

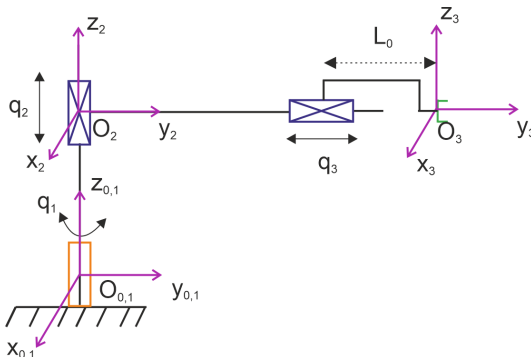


Figura 1.8: Braț robotic cu o cuplă de rotație și două de translație

Pentru structuri mai complexe de roboți abordarea prezentată, deși intuitivă, se complică considerabil. În practica roboticii se aplică anumite convenții care simplifică și sistematizează construirea modelului geometric direct. Cea mai cunoscută convenție este cea propusă de Denavit și Hartenberg [17], și va fi prezentată mai departe.

1.2.1 Metoda Denavit-Hartenberg

În cadrul convenției Denavit-Hartenberg (pe scurt, metoda *DH*) fiecare transformare omogenă $T_{i-1,i}$ se obține ca produsul a patru transformări de bază ([31],[24])⁸

$$\begin{aligned}
 T_{i-1,i} &= Rot(x, \alpha_{i-1}) Trans(x, a_{i-1}) Rot(z, \theta_i) Trans(z, d_i) \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.9)
 \end{aligned}$$

unde apar parametrii $\{\alpha_{i-1}, a_{i-1}, \theta_i, d_i\}$ asociați sistemelor de coordonate i și $i-1$, care se mai numesc parametrii *DH*.

Ne vom folosi de Figura 1.9 pentru a defini acești parametri, care prezintă legătura $i-1$ între cupla $i-1$ și cupla i [24]. Parametrul a_{i-1} reprezintă

⁸Metoda DH prezentată aici este referită în literatură ca metoda DH *modificată* [13].

perpendiculara comună pe axele $i - 1$ și i ⁹. Parametrul α_{i-1} reprezintă unghiul dintre cele două axe (se poate obține prin proiectarea în planul normal pe perpendiculara comună). Parametrul d_i reprezintă distanța de-a lungul axei i între perpendiculara comună a_{i-1} și perpendiculara comună (următoare) a_i , iar parametrul θ_i reprezintă unghiul dintre direcția legăturii $i - 1$ și direcția legăturii i .

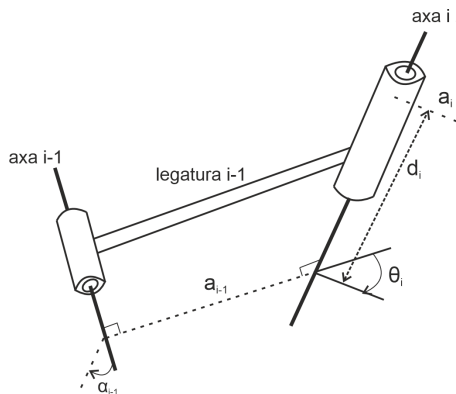


Figura 1.9: Reprezentare schematică pentru definirea parametrilor DH

Primul pas în aplicarea metodei DH constă în asocierea sistemelor de coordonate i pentru fiecare cuplă i . Acest lucru se poate face în trei pași:

Pasul 1. Se aleg axele z_i ca axele de mișcare pentru fiecare cuplă.

Pasul 2. Se aleg originile O_i ale sistemelor de coordonate pentru fiecare cuplă:

$$O_i = z_i \cap z_{i+1}.$$

Cazuri particulare:

- a) z_i și z_{i+1} colineare: O_i se poate alege arbitrar, dar este convenabil de obicei să fie ales în centrul cuplei i ,
- b) z_i și z_{i+1} paralele: $O_i = z_i \cap a_i$, unde a_i este normala comună.

Pasul 3. Se aleg axele x_i ale sistemelor de coordonate pentru fiecare cuplă:

$$x_i \perp \text{plan}(z_i, z_{i+1}).$$

Cazuri particulare:

- a) z_i și z_{i+1} colineare: $x_i \perp z_i$,
- b) z_i și z_{i+1} paralele: $x_i = a_i$.

⁹Axa unei cuple este axa de rotație, dacă cupla este de rotație, respectiv axa de-a lungul căreia se face translația, dacă cupla este de translație.

Sistemul de coordonate 0 (sistemul fix) care se plasează în baza robotului poate fi adoptat similar sistemului de coordonate 1. Ultimul sistem de coordonate n se poate alege arbitrar, dar de obicei ajută ca direcția lui x_n să fie aceeași cu x_{n-1} ¹⁰.

Pe baza sistemelor de coordonate atașate fiecărei cuple, în care este suficient să avem axele x și z , parametrii DH pot fi determinați în felul următor [24]:

- a_{i-1} este distanța dintre z_{i-1} și z_i , de-a lungul axei x_{i-1} ,
- α_{i-1} este unghiul dintre z_{i-1} și z_i , de-a lungul axei x_{i-1} ,
- d_i este distanța dintre x_{i-1} și x_i , de-a lungul axei z_i ,
- θ_i este unghiul dintre x_{i-1} și x_i , de-a lungul axei z_i .

De obicei parametrii α_i și a_i sunt constanți. Dacă cupla i este de rotație, atunci θ_i este variabil și d_i este constant. Dacă cupla i este de translație, atunci d_i este variabil și θ_i este constant.

Odată obținuți parametrii DH , se pot construi matricele de transformare (1.9), iar modelul geometric direct rezultă din (1.8).

Exercițiul 4. Să se determine modelul geometric direct prin metoda DH pentru robotul din Figura 1.10:

- analitic, pe baza ecuațiilor (1.9) și (1.8),
- numeric, folosind Robotics Toolbox în MATLAB [6].

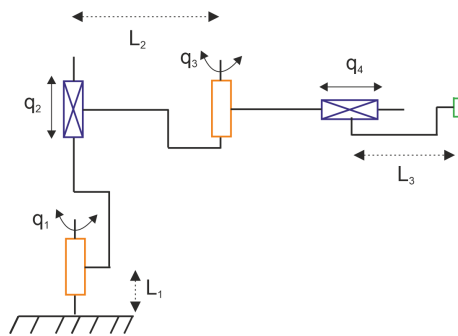


Figura 1.10: Braț robotic cu 2 cuple de rotație și 2 de translație

Soluție.

¹⁰Se observă că nu este o manieră unică de a adopta sistemele de coordonate, ca urmare matricele de transformare $T_{i-1,i}$ pot să difere, dar modelul geometric final obținut $T_{0,n}$ este același.

- a) Urmărind indicațiile anterioare pentru alegerea sistemelor de coordonate, rezultă Figura 1.11. Parametrii DH se pot determina sub forma tabelară pentru fiecare cuplă, în care fiecare linie i corespunde unei cuple i . Rezulta astfel Tabelul (1.1). Pe baza tabelului, matricile de transformare sunt

$$\begin{aligned}
 T_{0,1} &= \begin{bmatrix} c_{q_1} & s_{q_1} & 0 & 0 \\ s_{q_1}c_0 & c_{q_1}c_0 & -s_0 & -s_0L_1 \\ s_{q_1}s_0 & c_{q_1}s_0 & c_0 & c_0L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{1,2} &= \begin{bmatrix} c_{90^\circ} & s_{90^\circ} & 0 & 0 \\ s_{90^\circ}c_0 & c_{90^\circ}c_0 & -s_0 & -s_0q_2 \\ s_{90^\circ}s_0 & c_{90^\circ}s_0 & c_0 & c_0q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{2,3} &= \begin{bmatrix} c_{q_3-90^\circ} & s_{q_3-90^\circ} & 0 & L_2 \\ s_{q_3-90^\circ}c_0 & c_{q_3-90^\circ}c_0 & -s_0 & -s_0q_0 \\ s_{q_3-90^\circ}s_0 & c_{q_3-90^\circ}s_0 & c_0 & c_0q_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{3,4} &= \begin{bmatrix} c_0 & s_0 & 0 & 0 \\ s_0c_{-90^\circ} & c_0c_{-90^\circ} & -s_{-90^\circ} & -s_{-90^\circ}(L_3 + q_4) \\ s_0s_{-90^\circ} & c_0s_{-90^\circ} & c_{-90^\circ} & c_{-90^\circ}(L_3 + q_4) \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	L_1	q_1
2	0	0	q_2	90°
3	L_2	0	0	q_3-90°
4	0	-90°	$L_3 + q_4$	0

Tabela 1.1: Parametrii DH

Modelul geometric direct rezultă din înmulțirea matricilor și efectuarea unor simplificări:

$$T_{0,4} = \begin{bmatrix} -c_{q_1+q_3} & 0 & -s_{q_1+q_3} & L_2s_{q_1} - L_3s_{q_1-q_3} - q_4s_{q_1-q_3} \\ -s_{q_1+q_3} & 0 & -c_{q_1+q_3} & L_2c_{q_1} - q_4c_{q_1+q_3} - L_3c_{q_1+q_3} \\ 0 & -1 & 0 & L_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

- b) Pentru a determina modelul geometric direct, pornind de la Tabelul (1.1), se folosește codul MATLAB din Listing 1.1. Fiecare linie i din tabel este

introdusă cu funcția *Link*, în care variabila cuplei q_i este introdusă automat. Modelul se obține în final prin funcția *SerialLink*. Matricea $T_{0,4}$ care ne dă modelul geometric direct se poate determina pentru diferite valori numerice ale cuplelor $q = [q_1 \ q_2 \ q_3 \ q_4]^T$ prin funcția *fkine*.

Listing 1.1: Modelul geometric direct obținut numeric

```
L1=1;L2=2;L3=0.5;% m
%theta | d | a | alpha
L(1)=Link([0 L1 0 0], 'modified'); %met DH modificata
L(2)=Link([pi/2 0 0 0 1], 'modified');
% ultimul parametru: rot(0-default)/trans(1)
L(3)=Link([0 0 L2 0], 'modified');
% se suprascrive valoarea lui theta;
% trebuie adaugat offset la q3 cu -pi/2
L(4)=Link([0 0 0 -pi/2 1], 'modified');
%se suprascrive valoarea lui d;
%trebuie adaugat offset la q4 cu L3
robarm1 = SerialLink(L, 'name', 'four_link');

robarm1.offset(3)=-pi/2;
robarm1.offset(4)=L3;

%determinarea matricei T04 pentru q=[0 3 0.3 7]
robarm1.fkine([0 3 0.3 7])
%afisarea robotului in aceasta pozitie
robarm1.plot([0 3 0.3 7], 'workspace', [-5 5 -5 5 0 5])
```

□

Exercițiul 5. Să se determine analitic modelul geometric direct prin metoda DH pentru robotul cu trei cuple din Figura 1.12.

Exercițiul 6. Considerăm robotul colaborativ de tip UR5 cu șase cuple de rotație din Figura 1.13. Schema asociată robotului este ilustrată în Figura 1.14. Parametrii robotului sunt [32](în metri): $L_0 = 0.08$, $L_1 = 0.11$, $L_2 = 0.08$, $L_3 = 0$, $L_4 = 0.392$, $L_5 = 0$, $L_6 = 0.0945$, $L_7 = 0.082$. Să se determine tabelul cu parametrii DH, iar apoi, folosind Robotics Toolbox, să se determine numeric modelul geometric direct. Să se afișeze robotul pentru diferite valori ale cuplelor $q_1 \dots q_6$.

1.3 Modelul geometric invers

Modelul geometric invers ne permite determinarea variabilelor cuplelor în funcție de poziția și orientarea efectorului final în spațiul cartezian (Fi-

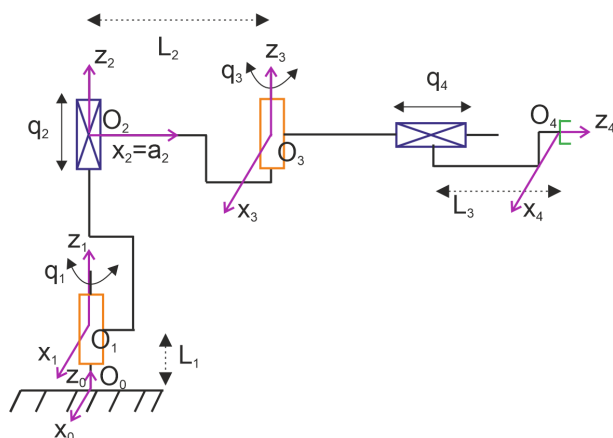


Figura 1.11: Sistemele de coordonate conform convenției *DH* pentru brațul robotic cu 2 cuple de rotație și 2 de rotație

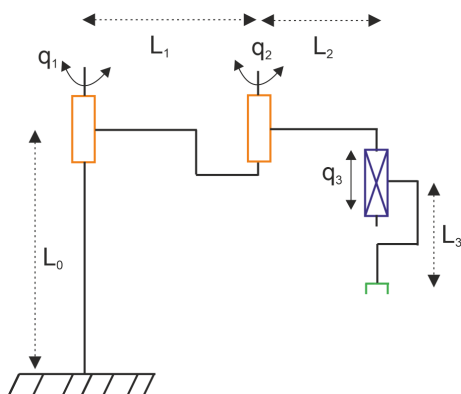


Figura 1.12: Braț robotic cu două cuple de rotație și una de translație

gura 1.2). Această problemă presupune rezolvarea unui sistem de ecuații algebrice neliniare, în care putem avea o soluție, mai multe sau niciuna. Problema se complică cu cât structura robotului este mai complexă, cu mai multe cuple, respectiv cu mulți parametri în tabelul *DH* diferiți de zero.

Problema generală a modelului geometric invers pentru un robot cu n cuple pornește de la modelul geometric direct

$$T_{0,n}(q_1, \dots, q_n) = H, \quad (1.10)$$

în care H este matricea care definește orientarea și poziția efectorului final pentru care dorim să determinăm valorile cuplelor q_1, \dots, q_n . Deci avem de rezolvat un sistem cu 12 ecuații neliniare și n necunoscute, în care fiecare

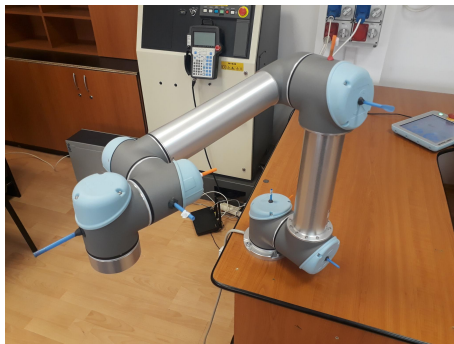


Figura 1.13: Robot colaborativ de tip UR5 cu 6 grade de libertate

ecuație are forma:

$$T_{i,j}(q_1, \dots, q_n) = h_{i,j}, \quad i = 1, 2, 3, \quad j = 1, \dots, 4. \quad (1.11)$$

Soluțiile care formează modelul geometric invers sunt de forma

$$q_k = f_k(h_{1,1}, \dots, h_{3,4}), \quad k = 1, \dots, n. \quad (1.12)$$

Găsirea unor soluții explicite analitice de forma (1.12) nu este întotdeauna posibilă, sau devine prea complicată. Nu există metode generale analitice, doar metode care funcționează pe anumite categorii de roboți. Avantajul unor soluții analitice constă în faptul că avem o soluție exactă la problemă, iar aplicarea unui model geometric invers dedus analitic este foarte eficientă din punct de vedere al puterii de calcul și ca timp de rulare. Ca regulă generală, soluții analitice se pot deduce la roboți cu minim 6 grade de libertate ¹¹, care au o structură specială cu mai mulți parametri *DH* zero ([13]–*Cap.2.7*), astfel încât:

- i) axele la trei cuple consecutive de rotație se intersectează într-un punct comun,
- ii) trei cuple de rotație consecutive sunt paralele.

Alternativa la metode analitice constă în apelarea la metode numerice, care au o aplicabilitate mai generală, dar ne furnizează doar soluții aproximative, respectiv sunt intensive din punct de vedere al calculului numeric și mai lente ca timp de rulare. O abordare numerică des întâlnită constă în formularea unei probleme de optimizare de tipul ([6])

$$q^* = \underset{q}{\operatorname{argmin}} \|T_{0,n}(q) - H^*\| \quad (1.13)$$

¹¹Numărul de grade de libertate reprezintă în *Robotică* numărul de cuple sau axe de mișcare.

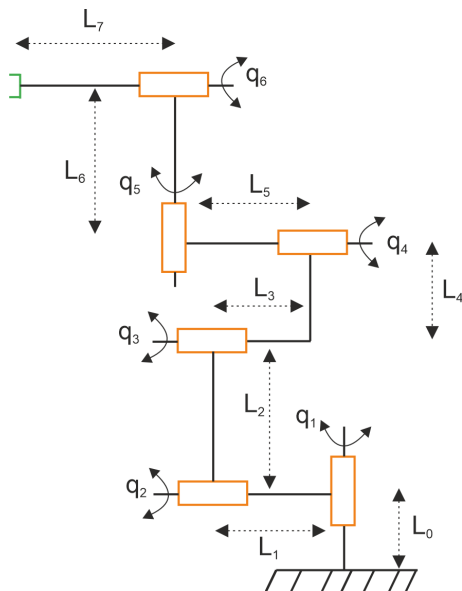


Figura 1.14: Reprezentare schematică a robotului de tip UR5 cu 6 grade de libertate

în care H^* reprezintă poziția și orientarea dorită a robotului. Practic dorim să găsim valoarea optimă a cuplelor q^* care minimizează eroarea dintre soluția modelului geometric direct și H^* . Soluția aproximativă obținută s-ar putea să fie doar un minim local, iar rezultatul este dependent de valoarea inițială a cuplei q_0 furnizată algoritmului de optimizare ¹², dar de multe ori în practică se obține un rezultat suficient de bun.

Următoarele trei exerciții exemplifică cum se poate obține modelul geometric invers, pe cale analitică sau numerică, pentru diverse configurații de roboți.

Exercițiul 7. Modelul geometric direct pentru robotul din Figura 1.8 este

$$T_{03} = \begin{bmatrix} c_{q1} & -s_{q1} & 0 & -s_{q1}(L_0 + q_3) \\ s_{q1} & c_{q1} & 0 & c_{q1}(L_0 + q_3) \\ 0 & 0 & 1 & q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Din ultima coloană putem extrage poziția efectorului final

$$\begin{aligned} x_{ef} &= -s_{q1}(L_0 + q_3), \\ y_{ef} &= c_{q1}(L_0 + q_3), \\ z_{ef} &= q_2. \end{aligned}$$

¹²Un exemplu de algoritm de optimizare care se poate folosi este algoritmul Levenberg-Marquardt.

Să se determine modelul geometric invers, analitic, adică soluții de forma $q_i = f_i(x_{ef}, y_{ef}, z_{ef})$ cu $i = \{1, 2, 3\}$. De multe ori în problema modelului geometric invers trebuie să impunem restricții la nivelul mișcării robotului, fie la nivelul domeniului admis de variație a cuplelor, fie la nivelul punctelor pe care efectorul final le poate atinge în spațiul cartezian. În acest exemplu considerăm că variabilele cuplelor de translație q_2 și q_3 nu pot să fie negative. Pentru q_2 sensul pozitiv este în sus, iar valoarea de zero la baza robotului. Pentru q_3 sensul pozitiv este spre dreapta, iar valoarea de zero la nivelul cuplei 2 a robotului.

Soluție. Împărțind prima ecuație la a doua obținem $\frac{x_{ef}}{y_{ef}} = -\frac{s_{q_1}}{c_{q_1}}$, de unde rezultă prima soluție $q_1 = -atan(\frac{x_{ef}}{y_{ef}})$. A doua soluție rezultă direct din ultima ecuație $q_2 = z_{ef}$. Ridicând la pătrat primele două ecuații și adunându-le, obținem o ecuație de ordinul 2 în q_3

$$q_3^2 + (2L_0)q_3 + (L_0^2 - x_{ef}^2 - y_{ef}^2) = 0,$$

care are două soluții $q_3^{1,2} = -L_0 \pm \sqrt{x_{ef}^2 + y_{ef}^2}$. Deoarece avem condiția pentru cupla 3 ca să fie pozitivă, din cele două soluții ne rămâne validă doar $q_3 = -L_0 + \sqrt{x_{ef}^2 + y_{ef}^2}$. În final, pe lângă cele 3 soluții $\{q_1, q_2, q_3\}$, trebuie să adăugăm restricțiile de tip inegalitate $z_{ef} > 0$ și $\sqrt{x_{ef}^2 + y_{ef}^2} > L_0$. Ca urmare, efectorul final nu poate ajunge mai jos de nivelul bazei robotului, și nu poate ajunge mai aproape de trunchiul robotului decât la o distanță minimă L_0 . □

Exercițiul 8. Considerăm robotul din Figura 1.10, la care modelul geometric direct a fost determinat folosind Listing 1.1. Să se determine numeric modelul geometric invers folosind funcția *ikine* din Robotics Toolbox [6].

Soluție. Listing 1.2 exemplifică cum putem obține și verifica o soluție la modelul geometric invers. Pornim de la o valoare inițială de test pentru vectorul cuplelor $q = [0 \ 3 \ 0.3 \ 7]^T$. Cu *fkine* determinăm, pe baza modelului geometric direct, poziția și orientarea efectorului final *Tr1*. Funcția *ikine*, care ne calculează soluția numerică la modelul geometric invers, are ca argumente de intrare: poziția și orientarea efectorului final (*Tr1* în cazul nostru), o valoare inițială pentru variabila cuplelor (am considerat $q_0 = [-0.1 \ 0.1 \ 0.1 \ 0.1]$), și o mască pentru roboții cu mai puțin de 6 grade de libertate. Masca este necesară deoarece în cazul roboților cu mai puțin de 6 grade de libertate, metoda nu poate calcula o soluție care să țină cont și de poziție și de orientare. Primii 3 parametri reprezintă poziția, iar ultimii 3 orientarea exprimată în funcție de unghiurile Euler. Fiindcă robotul nostru are 4 grade de libertate, masca se setează ca să se țină cont de 4 din acești

parametri (valorile de 1 din vectorul $[1\ 1\ 1\ 0\ 1\ 0]$). Verificarea constă în faptul că, prin rularea acestor instrucțiuni, obținem aceeași valoare pentru cuple de la care am pornit $q_{sol} = q$.

Listing 1.2: Modelul geometric invers obținut numeric

```
q=[0 3 0.3 7];
Tr1=robarm1.fkine(q);
q0=[-0.1 0.1 .1 .1];
masca=[1 1 1 0 1 0];
q_sol=robarm1.ikine(Tr1, 'q0', q0, 'mask', masca);
```

□

Exercițiul 9. *Să se determine modelul geometric invers numeric pentru robotul UR5 din Figura 1.14, folosind funcția $ikine$ ¹³ din Robotics Toolbox. Să se compare rezultatul obținut, cu cel obținut analitic (simbolic) prin funcția $ikine6s$, pentru diverse valori ale cuplelor q .*

¹³Deoarece robotul are 6 grade de libertate, nu mai este necesar folosirea argumentului *mask*.

Capitolul 2

Cinematica vitezelor

De multe ori nu este suficient să caracterizăm mișcarea unui robot doar în termeni de poziții, și este necesar să cunoaștem și evoluția vitezelor. Prin cipial viteza este definită ca derivata poziției. Dacă este vorba de viteza unui punct în spațiu, atunci aveam de a face doar cu viteza liniară¹⁴. Dacă suntem însă interesați de mișcarea unui corp (obiect sau solid rigid), atunci trebuie să facem distincția între viteza liniară (specifică unei mișcări de translație) și viteza unghiulară (specifică unei mișcări de rotație). Legătura dintre vitezele cuplelor și vitezele liniare și unghiulare ale efectorului final este dată de o transformare denumită Jacobianul robotului. Jacobianul generalizează noțiunea de derivată a unei funcții scalare, iar în Robotică este folosit în special în determinarea modelului dinamic al robotului, dar și în determinarea legăturii dintre forțele/momentele de forță la nivelul efectorului final și cele de la nivelul cuplelor. În acest capitol vom începe cu prezentarea Jacobianului $2D$, adică pentru roboți care se mișcă în plan, pentru o primă înțelegere intuitivă a lucrurilor. Ulterior vom descrie Jacobianul în cazul general $3D$ și exemplifica pe diverse configurații de roboți. În ultimul subcapitol vom menționa câteva aspecte referitoare la inversarea Jacobianului.

2.1 Jacobianul $2D$ al unui robot

În această secțiune o să explicăm noțiunea de **Jacobian $2D$** cu ajutorul unui exemplu. Să considerăm brațul robotic cu două cuple de rotație din Figura 2.1. Se poate observa ca ambele axe de rotație (axele Z) sunt perpendiculare pe planul XOY , deci avem de-a face cu un robot care se poate mișca doar în plan.

Pentru a putea discuta despre legătura dintre viteze, mai întâi trebuie să determinăm legătura dintre poziții, prin intermediul modelului geometric

¹⁴Noțiunea de viteză unghiulară a unui punct nu are sens fiindcă nu avem orientare.

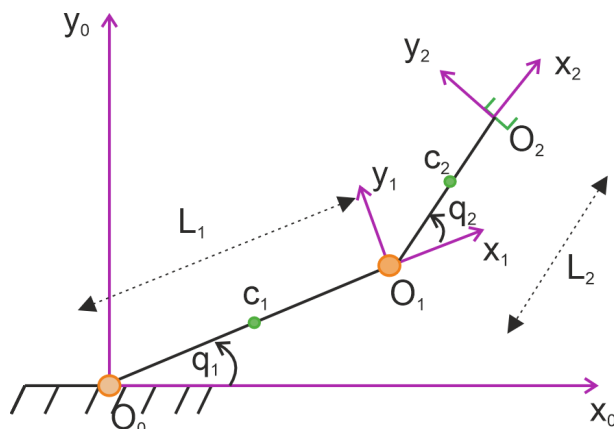


Figura 2.1: Robot cu două cuple de rotație care se poate mișca doar în planul XOY

direct. Dacă O_0 este originea sistemului fix de coordonate (plasat în cupla 1), și O_1 este originea sistemului de coordonate plasat în cupla 2, atunci avem:

$$x_1 = L_1 \cdot \cos(q_1), \quad y_1 = L_1 \cdot \sin(q_1), \quad (2.1)$$

iar poziția efectorului final (O_2) este dată de ecuațiile

$$x = x_2 = x_1 + L_2 \cdot \cos(q_1 + q_2), \quad y = y_2 = y_1 + L_2 \cdot \sin(q_1 + q_2). \quad (2.2)$$

În ambele cazuri $z = 0$, iar mișcarea este restricționată la planul XOY .

Pozițiile unghiulare în jurul fiecărei axe de rotație $\{x, y, z\}$ pot fi exprimate ca:

$$\theta_x = 0, \quad \theta_y = 0, \quad \theta_z = q_1 + q_2. \quad (2.3)$$

Jacobianul face legătura dintre vitezele cuplelor și vitezele liniare și unghiulare ale efectorului final¹⁵. Deci începem prin a defini vectorul pozițiilor liniare $p = [x \ y \ z]^T$ și vectorul pozițiilor unghiulare $\theta = [\theta_x \ \theta_y \ \theta_z]^T$. Legătura dintre acestea și variabilele cuplelor este dată de funcțiile

$$p = f_p(q), \quad \theta = f_\theta(q), \quad (2.4)$$

unde $q = [q_1 \ q_2]^T$ este vectorul variabilelor cuplelor. Vectorul vitezelor liniare se poate scrie ca

$$v = \dot{p} = \frac{\partial f_p}{\partial q} \cdot \dot{q} = J_v \cdot \dot{q},$$

¹⁵Jacobianul definit mai departe este valabil doar pentru mișcarea în plan, și se bazează pe descrierea din [1].

unde J_v este Jacobianul vitezelor liniare. De asemenea, vectorul vitezelor unghiulare se poate scrie ca

$$\omega = \dot{\theta} = \frac{\partial f_{\theta}}{\partial q} \cdot \dot{q} = J_{\omega} \cdot \dot{q},$$

unde J_{ω} este Jacobianul vitezelor unghiulare.

Matricele Jacobian ale vitezelor liniare pentru *legătura 1 și 2* sunt:

$$J_v = J_{v_2} = \begin{bmatrix} \frac{\partial x_2}{\partial q_1} & \frac{\partial x_2}{\partial q_2} \\ \frac{\partial y_2}{\partial q_1} & \frac{\partial y_2}{\partial q_2} \\ \frac{\partial z_2}{\partial q_1} & \frac{\partial z_2}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -L_1 \sin(q_1) - L_2 \sin(q_1 + q_2) & -L_2 \sin(q_1 + q_2) \\ L_1 \cos(q_1) + L_2 \cos(q_1 + q_2) & L_2 \cos(q_1 + q_2) \\ 0 & 0 \end{bmatrix}, \quad (2.5)$$

$$J_{v_1} = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} \\ \frac{\partial y_1}{\partial q_1} & \frac{\partial y_1}{\partial q_2} \\ \frac{\partial z_1}{\partial q_1} & \frac{\partial z_1}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -L_1 \sin(q_1) & 0 \\ L_1 \cos(q_1) & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.6)$$

Deci putem scrie că $v_1 = J_{v_1} \cdot \dot{q}$ și $v_2 = J_{v_2} \cdot \dot{q}$.

Matricele Jacobian ale vitezelor unghiulare pentru *elementele 1 și 2*¹⁶, sunt:

$$J_{\omega} = J_{\omega_2} = \begin{bmatrix} \frac{\partial \theta_x}{\partial q_1} & \frac{\partial \theta_x}{\partial q_2} \\ \frac{\partial \theta_y}{\partial q_1} & \frac{\partial \theta_y}{\partial q_2} \\ \frac{\partial \theta_z}{\partial q_1} & \frac{\partial \theta_z}{\partial q_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad (2.7)$$

$$J_{\omega_1} = \begin{bmatrix} \frac{\partial \theta_x}{\partial q_1} & \frac{\partial \theta_x}{\partial q_2} \\ \frac{\partial \theta_y}{\partial q_1} & \frac{\partial \theta_y}{\partial q_2} \\ \frac{\partial \theta_z}{\partial q_1} & \frac{\partial \theta_z}{\partial q_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}. \quad (2.8)$$

Deci putem scrie că $\omega_1 = J_{\omega_1} \cdot \dot{q}$ și $\omega_2 = J_{\omega_2} \cdot \dot{q}$.

În final, matricele Jacobian pentru legătura 1 și 2 sunt:

$$J = J_2 = \begin{bmatrix} J_{v_2} \\ J_{\omega_2} \end{bmatrix} \quad \text{și} \quad J_1 = \begin{bmatrix} J_{v_1} \\ J_{\omega_1} \end{bmatrix}.$$

Exercițiul 10. Pentru robotul din Figura 2.1, cu parametrii $L_1 = 0.5 \text{ m}$ și $L_2 = 0.5 \text{ m}$, să se determine vitezele liniare v ale efectorului final pentru vitezele cuplelor $\dot{q} = [0.1 \ 0.3]^T \text{ rad/sec}$, considerând valorile cuplelor $q = [0 \ 0]^T \text{ rad}$.

¹⁶Pentru legătura 1 avem $\theta_x = 0$, $\theta_y = 0$ și $\theta_z = q_1$.

Soluție. Pe baza Jacobianului J_v , putem calcula vitezele liniare ca

$$v = \begin{bmatrix} 0 & 0 \\ 1 & 0.5 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \text{ m/sec.}$$

□

2.2 Jacobianul 3D al unui robot

Jacobianul unui robot în cazul general 3D este mai complicat de calculat, deoarece include noțiunile generale de viteze liniare și unghiulare în spațiul 3D. În ce urmează, vom descrie ceea ce se numește Jacobianul geometric (prezentarea se bazează pe [31]- *Cap.6*).

Considerăm poziția efectorului final raportată la sistemul de referință fix (de obicei atașat în baza robotului) $o_{0n}(q) = [x \ y \ z]^T$, iar orientarea o specificăm prin matricea $R_{0n}(q)$ de dimensiune 3×3 , unde variabila cuplelor este definită ca $q = [q_1 \ \dots \ q_n]^T$, iar n reprezintă numărul cuplelor. Astfel, modelul geometric direct al robotului¹⁷ este de forma

$$T_{0n}(q) = \begin{bmatrix} R_{0n}(q) & o_{0n}(q) \\ 0_{1 \times 3} & 1 \end{bmatrix}. \quad (2.9)$$

Mai departe definim vectorul vitezelor efectorului final raportat la sistemul de referință fix

$$\zeta = [\dot{x} \ \dot{y} \ \dot{z} \ \omega_x \ \omega_y \ \omega_z]^T, \quad (2.10)$$

unde \dot{x} , \dot{y} și \dot{z} reprezintă vitezele liniare, iar ω_x , ω_y și ω_z reprezintă vitezele unghiulare¹⁸.

Matricea Jacobian ne va da legătura dintre vitezele cuplelor și vitezele efectorului final

$$\zeta = J \dot{q}, \quad (2.11)$$

unde $\dot{q} = [\dot{q}_1 \ \dots \ \dot{q}_n]^T$, iar dimensiunea matricei J este $6 \times n$. Pentru primele 3 linii ale Jacobianului care se referă la vitezele liniare definim matricea J_v , iar pentru următoarele 3 linii care se referă la viteze unghiulare definim matricea J_ω . Astfel avem $J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$.

¹⁷Presupunem că modelul geometric direct a fost obținut prin metoda *DH*, în care axele Z corespund axelor de mișcare.

¹⁸A se observa că pentru viteze unghiulare s-a evitat notația de forma $\dot{\theta}$, uzuală mișcării de rotație în plan cu unghi θ , deoarece aceasta nu se poate generaliza în cazul 3D. În cazul 2D avem întotdeauna o rotație în jurul unei singure axe fixe. În cazul 3D însă putem avea rotație în jurul unei axe care nu este fixă, sau putem avea mai multe rotații în jurul unor axe diferite [31].

Jacobianul vitezelor unghiulare J_ω are atâtea coloane câte cuple are robotul:

$$J_\omega = [J_{\omega 1} \quad \dots \quad J_{\omega i} \quad \dots \quad J_{\omega n}]. \quad (2.12)$$

Fiecare coloană i se determină în funcție de tipul cuplei: rotație sau translație. Dacă cupla i este de translație, nu contribuie la mișcarea de rotație, deci $J_{\omega i} = [0 \ 0 \ 0]^T$. Pentru cuple de rotație, mișcarea depinde de orientare. Dacă am adoptat convenția DH , atunci axa fiecărei cuple i este aliniată axei Z , deci va rezulta o viteză unghiulară ω_z doar în jurul acestei axe. Ca urmare, pentru o cuplă i de rotație, coloana Jacobianului o putem extrage din coloana a 3-a a matricei de orientare: $J_{\omega i} = R_{0,i}(:, 3)$.

Similar, Jacobianul vitezelor liniare J_v are n coloane:

$$J_v = [J_{v1} \quad \dots \quad J_{vi} \quad \dots \quad J_{vn}]. \quad (2.13)$$

Dacă cupla i este de translație, contribuie la mișcarea de rotație în funcție de orientare. Din convenția DH , axa de mișcare (axa de-a lungul căreia are loc translația) este axa Z . Deci pentru o cuplă de translație coloana Jacobianului este $J_{vi} = R_{0,i}(:, 3)$. Dacă cupla i este de rotație, atunci trebuie să ținem cont, pe lângă orientarea cuplei, și de distanța dintre cuplă și efectorul final. Coloana Jacobianului se calculează în acest caz ca produsul vectorial $J_{vi} = R_{0,i}(:, 3) \times (o_{0n} - o_{0,i})$.

În concluzie, **Jacobianul geometric**

$$J = [J_1 \quad \dots \quad J_i \quad \dots \quad J_n] \quad (2.14)$$

se poate calcula coloană cu coloană. Dacă cupla i este de rotație atunci

$$J_i = \begin{bmatrix} R_{0,i}(:, 3) \times (o_{0n} - o_{0,i}) \\ R_{0,i}(:, 3) \end{bmatrix}, \quad (2.15)$$

iar dacă este de translație atunci

$$J_i = \begin{bmatrix} R_{0,i}(:, 3) \\ 0_{3 \times 1} \end{bmatrix}. \quad (2.16)$$

Exercițiul 11. Considerăm brațul robotic cu 3 grade de libertate din Figura 2.2, cu parametrii $L_1 = L_2 = L_3 = 1m$.

- a) Să se determine modelul geometric direct cu metoda DH .
- b) Să se determine Jacobianul geometric.
- c) Să se compare Jacobianul geometric obținut cu cel determinat numeric prin funcția `jacob0` din *Robotics Toolbox* [6], pentru diferite valori ale cuplelor.

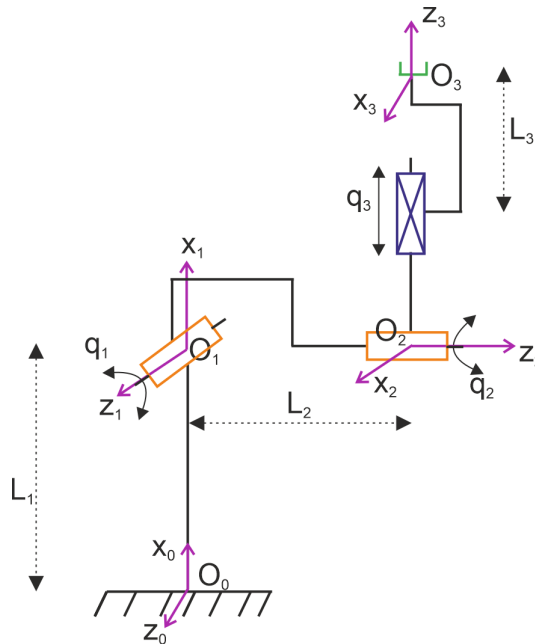


Figura 2.2: Robot cu două cuple de rotație și o cuplă de translație

Soluție.

- a) Pornind de la adoptarea sistemelor de coordonate precum în Figura 2.2, se poate determina Tabelul 2.1 cu parametrii DH . Pe baza tabelului, se pot scrie cele trei matrici de transformare:

$$T_{01} = \begin{bmatrix} c_{q_1} & -s_{q_1} & 0 & 1 \\ s_{q_1} & c_{q_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{12} = \begin{bmatrix} c_{q_2+90^\circ} & -s_{q_2+90^\circ} & 0 & 0 \\ 0 & 0 & -1 & -1 \\ s_{q_2+90^\circ} & c_{q_2+90^\circ} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_3 - q_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Modelul geometric direct rezultă din înmulțirea matricilor

$$T_{03} = T_{01}T_{12}T_{23}.$$

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	L_1	0	0	q_1
2	0	90°	L_2	q_2+90°
3	0	90°	q_3+L_3	0

Tabela 2.1: Parametrii DH

- b) Jacobianul robotului are trei coloane $J = [J_1 \ J_2 \ J_3]$. Prima și a doua coloană corespund unor cuple de rotație și se calculează ca

$$J_1 = \begin{bmatrix} R_{0,1}(:, 3) \times (o_{0n} - o_{0,1}) \\ R_{0,1}(:, 3) \end{bmatrix}, J_2 = \begin{bmatrix} R_{0,2}(:, 3) \times (o_{0n} - o_{0,2}) \\ R_{0,2}(:, 3) \end{bmatrix}.$$

A treia coloană corespunde unei cuple de translație și se calculează ca

$$J_3 = \begin{bmatrix} R_{0,3}(:, 3) \\ 0_{3 \times 1} \end{bmatrix}.$$

În urma calculului, rezultă matricea Jacobianului

$$J = \begin{bmatrix} c_{q_1} - s_{q_1} s_{q_2+90^\circ} (q_3 + 1) & -c_{q_1} s_{q_2} (q_3 + 1) & c_{q_1} c_{q_2} \\ s_{q_1} + c_{q_1} s_{q_2+90^\circ} (q_3 + 1) & -s_{q_1} s_{q_2} (q_3 + 1) & c_{q_2} s_{q_1} \\ 0 & c_{q_2} (q_3 + 1) & s_{q_2} \\ 0 & s_{q_1} & 0 \\ 0 & -c_{q_1} & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.17)$$

- c) Dacă evaluăm Jacobianul (2.17), de exemplu pentru valorile cuplelor $q = [0.1 \ 0.1 \ 0.1]^T \text{ rad}$, obținem același rezultat ca și cel furnizat de Listing 2.1, în care este utilizată funcția *jacob0* pentru calcularea numerică a Jacobianului.

Listing 2.1: Jacobianul robotului obținut numeric

```
%theta | d | a | alpha
% L1=1, L2=1, L3=1
L(1)=Link([0 0 1 0 0], 'modified');
L(2)=Link([0 1 0 pi/2 0], 'modified');
L(3)=Link([0 0 0 pi/2 1], 'modified');

robarm2 = SerialLink(L, 'name', 'three_link');

robarm2.offset(2)=pi/2;
robarm2.offset(3)=1;
```

```
robarm2.jacob0([0.1 0.1 0.1])
```

□

Exercițiul 12. Considerăm robotul cu șase cuple de rotație din Figura 1.14, prezentat în exercițiul 6, pentru care s-a determinat modelul geometric direct prin metoda DH.

- a) Să se determine Jacobianul pentru valorile cuplelor $\{q_1 = 0.1, q_2 = 0.5, q_3 = 0.1, q_4 = 0.8, q_5 = 0.1, q_6 = 0.2\}$ rad, folosind funcția `jacob0` din *Robotics Toolbox*.
- b) Să se determine vitezele (liniare și unghiulare) ale efectorului final pentru vitezele cuplelor $\dot{q} = [1 \ 1.4 \ 1.7 \ 0.2 \ 0.3 \ 0.1]^T$ rad/sec.

2.3 Inversarea Jacobianului

De multe ori în practică este mai important să aflăm vitezele cuplelor în funcție de vitezele efectorului final. Când Jacobianul este o matrice patrată nesingulară (dimensiune 6×6), aceasta se poate obține înmulțind la stânga în (2.11) cu **inversul matricei Jacobian**, rezultând

$$J^{-1}\zeta = \dot{q}. \quad (2.18)$$

Pentru roboții care au mai puțin de 6 cuple (matricea Jacobian are mai puțin de 6 coloane), în general Jacobianul nu este inversabil. În cazuri particulare, o soluție la (2.11) în termeni de \dot{q} există dacă se respectă condiția de rang

$$\text{rang } J(q) = \text{rang}[J(q) | \zeta],$$

și se poate determina de exemplu prin metoda de eliminare Gauss din algebră liniară [31]. Pentru roboți care au mai mult de 6 cuple, soluția este dată de expresia

$$J^+\zeta = \dot{q}, \quad (2.19)$$

unde J^+ este pseudoinversa definită ca

$$J^+ = J^T (J J^T)^{-1}. \quad (2.20)$$

În general, matricea Jacobian nu este constantă, depinzând de valoarea variabilelor cuplelor $J := J(q(t))$. Ca urmare și rangul matricei depinde de cuplele q . Definim ca puncte de singularitate acele valori ale cuplelor pentru care rangul matricei devine mai mic decât dimensiunea matricei. Determinarea posibilelor singularități ale unui robot este foarte importantă

în practică deoarece reprezintă poziții sau direcții de mișcare care nu pot fi atinse de robot, sau în care viteze mărginite ale efectorului final conduc la viteze nemărginite ale cuplelor.

Exercițiul 13. *Considerăm robotul cu șase cuple de rotație din Figura 1.14, prezentat în exercițiul 6, pentru care se cunoaște Jacobianul (vezi exercițiul 12). Să se calculeze vitezele cuplelor \dot{q} rezultate pentru o viteză liniară a efectorului final pe axa X de 1 m/sec.*

Capitolul 3

Modelul dinamic

În acest capitol vom discuta despre cum se poate obține modelul dinamic pentru un braț robotic. Deși există mai multe abordări în construirea modelului dinamic, în această carte ne vom concentra pe formalismul Euler-Lagrange, în care se pornește de la energia cinetică și energia potențială a robotului. Modelul dinamic, odată construit, se poate apoi utiliza fie pentru simularea comportamentului dinamic al robotului, fie pentru proiectarea unor structuri de conducere bazate pe model.

În prima secțiune vom prezenta ecuațiile Euler-Lagrange, precum și transpunerea lor într-o formă matriceală standard. Un robot în plan cu două cuple de rotație va fi folosit ca exemplu al modului de calcul implicat în contruirea modelului dinamic. În secțiunea 2 prezentăm cazul mai general, în care contruim un model dinamic pentru un robot care se mișcă în spațiul 3D. Secțiunea a 3-a descrie partea de modelare a actuatorilor electromecanice care acționează la nivelul cuplelor, iar în ultima secțiune aceste modele sunt integrate în modelul robotului.

3.1 Ecuațiile Euler-Lagrange

Începem prin a defini **Lagrangianul** ca

$$L = K - P, \tag{3.1}$$

unde K reprezintă energia cinetică totală a sistemului și P reprezintă energia potențială totală a sistemului.

Ecuațiile Euler Lagrange care descriu dinamica unui sistem mecanic

cu n grade de libertate sunt ¹⁹ :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, \dots, n, \quad (3.2)$$

unde q_i reprezintă coordonatele generalizate (în cazul nostru variabilele cuplelor) și τ_i forțe generalizate (în cazul nostru cuplurile generate de actuatori) ²⁰ . Energia cinetică totală este suma energiilor cinetice ale fiecărui element (solid rigid) al robotului. Pentru un element i , energia cinetică se exprimă de obicei în funcție de vitezele liniare ale centrului de greutate și vitezele unghiulare

$$K_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} m_i \omega_i^T I_i \omega_i, \quad (3.3)$$

unde I_i reprezintă momentul de inerție. Energia potențială totală este de asemenea suma energiilor potențiale ale elementelor, iar pentru un element i al robotului depinde de obicei de forța gravitațională ²¹

$$P_i = m g h_{ci}, \quad (3.4)$$

unde h_{ci} este înălțimea centrului de greutate c_i a elementului i , iar g este accelerația gravitațională.

Pentru o înțelegere intuitivă a modului în care se aplică ecuațiile Euler-Lagrange, în exercițiul de mai jos vom arăta cum se poate determina modelul dinamic al unui robot în plan.

Exercițiul 14. *Să se determine modelul dinamic pentru robotul în plan din Figura 2.1.*

Soluție. Lagrangianul robotului este definit ca diferența energiei cinetice totale și a energiei potențiale totale: $L = K - P$. Energia cinetică totală are o componentă de translație

$$K_{transl} = \frac{1}{2} m_1 v_{c1}^T v_{c1} + \frac{1}{2} m_2 v_{c2}^T v_{c2} = \frac{1}{2} \dot{q}^T (m_1 J_{vc1}^T J_{vc1} + m_2 J_{vc2}^T J_{vc2}) \dot{q}, \quad (3.5)$$

și una de rotație

$$K_{rot} = \frac{1}{2} \omega_2^T I_2 \omega_2 + \frac{1}{2} \omega_1^T I_1 \omega_1 = \frac{1}{2} \dot{q}^T (J_{\omega 2}^T I_2 J_{\omega 2} + J_{\omega 1}^T I_1 J_{\omega 1}) \dot{q}, \quad (3.6)$$

¹⁹Ecuatiile Euler-Lagrange pot fi folosite și pentru modelarea unor sisteme de altă natură decât mecanică, cum ar fi modelarea unor sisteme electro-magnetice [20], respectiv sunt folosite și în control optimal și calculul variațional [22].

²⁰Aceasta prezentare este bazată pe [31]. Pentru o demonstrație formală legată de cum se pot obține ecuațiile Euler-Lagrange din legile lui Newton, pornind de la principiul lucrului mecanic virtual, a se vedea capitolul 7.1.

²¹În funcție de natura sistemului modelat, energiile cinetice și potențiale mai pot să conțină și alți termeni, care de exemplu țin cont de elasticitate, sau de partea electrică, etc.

Capitolul 3. Modelul dinamic

unde I_1 și I_2 sunt momentele de inerție. Deoarece în expresiile de mai sus apar vitezele liniare raportate la centrele de greutate c_2 și c_1 ²², matricile Jacobian (2.5) și (2.6), determinate anterior, se vor recalcula raportat la centrele de greutate²³:

$$J_{vc_2} = \begin{bmatrix} \frac{\partial x_{c_2}}{\partial q_1} & \frac{\partial x_{c_2}}{\partial q_2} \\ \frac{\partial y_{c_2}}{\partial q_1} & \frac{\partial y_{c_2}}{\partial q_2} \\ \frac{\partial z_{c_2}}{\partial q_1} & \frac{\partial z_{c_2}}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -L_1 \sin(q_1) - L_{c_2} \sin(q_1 + q_2) & -L_{c_2} \sin(q_1 + q_2) \\ L_1 \cos(q_1) + L_{c_2} \cos(q_1 + q_2) & L_{c_2} \cos(q_1 + q_2) \\ 0 & 0 \end{bmatrix}, \quad (3.7)$$

$$J_{vc_1} = \begin{bmatrix} \frac{\partial x_{c_1}}{\partial q_1} & \frac{\partial x_{c_1}}{\partial q_2} \\ \frac{\partial y_{c_1}}{\partial q_1} & \frac{\partial y_{c_1}}{\partial q_2} \\ \frac{\partial z_{c_1}}{\partial q_1} & \frac{\partial z_{c_1}}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -L_{c_1} \sin(q_1) & 0 \\ L_{c_1} \cos(q_1) & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.8)$$

Deci avem $v_{c_1} = J_{vc_1} \cdot \dot{q}$ și $v_{c_2} = J_{vc_2} \cdot \dot{q}$.

Energia cinetică totală se poate scrie în forma condensată ca

$$K = K_{transl} + K_{rot} = \frac{1}{2} \dot{q}^T D(q) \dot{q}, \quad (3.9)$$

unde $D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$ se numește matricea de inerție, și are termenii

$$d_{11} = m_1 L_{c_1}^2 + m_2 (L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2}^2 + 2L_1 L_{c_2} \cos(q_2)) + I_1 + I_2,$$

$$d_{12} = m_2 (L_{c_2}^2 + L_1 L_{c_2} \cos(q_2)) + I_2,$$

$$d_{21} = d_{12},$$

$$d_{22} = m_2 L_{c_2}^2 + I_2.$$

Energia potențială totală se determină ca înmulțire a masei cu accelerația gravitațională și înălțimea centrului de greutate:

$$P_1 = m_1 g L_{c_1} \sin(q_1), \quad P_2 = m_2 g (L_1 \sin(q_1) + L_{c_2} \sin(q_1 + q_2)), \quad P = P_1 + P_2. \quad (3.10)$$

Particularizând ecuațiile Euler-Lagrange pentru robotul nostru cu două cuple obținem

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_1} \right) - \frac{\partial K}{\partial q_1} + \frac{\partial P}{\partial q_1} = \tau_1, \quad (3.11)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_2} \right) - \frac{\partial K}{\partial q_2} + \frac{\partial P}{\partial q_2} = \tau_2. \quad (3.12)$$

²²Viteza unghiulară nu este proprietatea unui punct individual, dar viteza liniară poate să fie.

²³Considerăm centrul de greutate al fiecărei legături la mijlocul fiecărui segment: $L_{c_2} = L_2/2, L_{c_1} = L_1/2$.

Capitolul 3. Modelul dinamic

Derivatele parțiale ale energiei potențiale sunt:

$$\begin{aligned}\frac{\partial P}{\partial q_1} &= (m_1 L_{c1} + m_2 L_1) g \cos(q_1) + m_2 L_{c2} g \cos(q_1 + q_2), \\ \frac{\partial P}{\partial q_2} &= m_2 L_{c2} \cos(q_1 + q_2).\end{aligned}$$

Derivatele parțiale ale energiei cinetice raportate la q și \dot{q} se pot calcula ca:

$$\begin{aligned}\frac{\partial K}{\partial q_1} &= 0, \\ \frac{\partial K}{\partial q_2} &= (-m_2 L_1 L_{c2} \sin(q_2))\dot{q}_1^2 - (m_2 L_1 L_{c2} \sin(q_2))\dot{q}_1 \dot{q}_2, \\ \frac{\partial K}{\partial \dot{q}_1} &= d_{11}\dot{q}_1 + d_{12}\dot{q}_2, \\ \frac{\partial K}{\partial \dot{q}_2} &= d_{12}\dot{q}_1 + d_{22}\dot{q}_2.\end{aligned}$$

Derivatele raportate la timp pentru ultimele două ecuații se determină în felul următor:

$$\begin{aligned}\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_1} \right) &= \dot{d}_{11} \dot{q}_1 + d_{11} \ddot{q}_1 + \dot{d}_{12} \dot{q}_2 + d_{12} \ddot{q}_2 \\ &= d_{11} \ddot{q}_1 + d_{12} \ddot{q}_2 - (2 m_2 L_1 L_{c2} \sin(q_2) \dot{q}_2) \dot{q}_1 - (m_2 L_1 L_{c2} \sin(q_2) \dot{q}_2) \dot{q}_2, \\ \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_2} \right) &= \dot{d}_{22} \dot{q}_2 + d_{22} \ddot{q}_2 + \dot{d}_{12} \dot{q}_1 + d_{12} \ddot{q}_1 \\ &= 0 + d_{22} \ddot{q}_2 - (m_2 L_1 L_{c2} \sin(q_2) \dot{q}_2) \dot{q}_1 + d_{12} \ddot{q}_1.\end{aligned}$$

În final modelul dinamic poate fi adus la o forma matriceală:

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau,$$

unde $q = [q_1 \quad q_2]^T$, $\tau = [\tau_1 \quad \tau_2]^T$, $G = [\frac{\partial P}{\partial q_1} \quad \frac{\partial P}{\partial q_2}]^T$ este vectorul gravitațional, $D(q)$ este matricea de inerție, iar restul termenilor sunt adunați în matricea $C(q, \dot{q})$ (termeni care țin cont de forța centrifugă și efectul de Coriolis). \square

Este de multe ori util ca să aducem ecuațiile Euler-Lagrange la **forma matriceală** standard [31]:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.13)$$

unde $q = [q_1, \dots, q_n]^T$, $\tau = [\tau_1, \dots, \tau_n]^T$.

Matricea $D(q)$ se numește matrice de inerție, este simetrică și pozitiv definită, și poate fi exprimată în funcție de termenii energiei cinetice:

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q} = \frac{1}{2} \sum_{i,j}^n d_{ij}(q) \dot{q}_i \dot{q}_j. \quad (3.14)$$

Matricea $C(q, \dot{q})$ ține cont de termenii forțelor centrifuge și efectul de Coriolis, iar fiecare element $\{k, j\}$ al matricei poate fi calculat ca:

$$c_{kj} = \frac{1}{2} \sum_{i=1}^n \underbrace{\left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\}}_{c_{ijk}} \dot{q}_i. \quad (3.15)$$

În final $G(q)$ este termenul care ține cont de gravitație, și este de obicei un vector coloană $G(q) = [g_1 \dots g_n]^T$, unde fiecare termen k se obține prin derivarea energiei potențiale:

$$g_k(q) = \frac{\partial P}{\partial q_k}, \quad k = 1, \dots, n. \quad (3.16)$$

3.2 Modelul dinamic al unui braț robotic cu două cuple de rotație

Pentru roboți industriali cu 6 grade de libertate, modelul dinamic presupune un volum mare de calcul, ca urmare de multe ori în practică se apelează la calcul simbolic²⁴. Ecuatiile Euler-Lagrange în forma matriceală se pretează la o astfel de implementare algoritmică.

În această secțiune vom descrie construirea modelului dinamic, pas cu pas, pornind de la modelul geometric și Jacobianul, pe un exemplu mai simplu de robot cu 2 grade de libertate, dar care se mișcă totuși într-un spațiu 3D. Figura 3.2 prezintă robotul fizic²⁵, iar Figura 3.1 este reprezentarea schematică. Deoarece prima axă de rotație este pe axa X, iar a doua axă de rotație este pe axa Y, robotul se poate mișca în spațiul 3D, deși are doar două grade de libertate.

Modelul geometric se poate obține prin determinarea matricelor de transformare de la sistemul fix (de referință) atașat bazei robotului, la sistemul de coordonate atașat efectorului final. Considerăm cazul simplificat în care $L_0 = 0$ deci sistemul de coordonate fix se va suprapune cu sistemul de coordonate atașat primei cuple a robotului (pentru $q_1 = 0$). Vom nota originea sistemului fix cu O_0 , iar originea sistemului de coordonate atașat cuplei 1 cu O_1 . Deci matricea de transformare între cele două sisteme de coordonate presupune doar o rotație în jurul axei X:

$$T_{01} = Rot(x, q_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_1) & -\sin(q_1) & 0 \\ 0 & \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

²⁴A se vedea de exemplu Symbolic Math Toolbox în MATLAB.

²⁵Robotul a fost proiectat și construit de Zoltan Nagy - a se vedea [25].

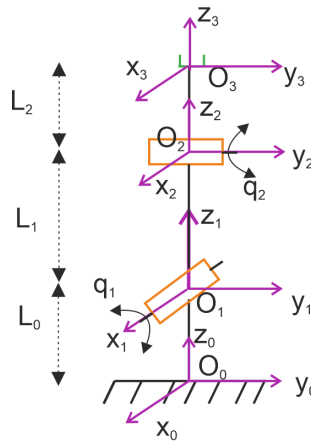


Figura 3.1: Reprezentare schematică a unui braț robotic cu 2 grade de libertate



Figura 3.2: Robotul cu două cuple de rotație din [25]

Din sistemul de coordonate 1 (atașat cuplei 1) putem ajunge în sistemul de coordonate 2 (atașat cuplei 2) printr-o translație pe axa Z și o rotație în jurul axei Y :

$$T_{12} = Transl(z, L_1) \cdot Rot(y, q_2) = \begin{bmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_2) & 0 & \cos(q_2) & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

În final ajungem la sistemul de coordonate atașat efectorului final printr-o

translație pe axa Z :

$$T_{23} = \text{Transl}(z, L_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matricea de transformare pentru a ajunge de la sistemul de referință la sistemul de coordonate al efectorului final, adică **modelul geometric**, se obține ca:

$$T = T_{03} = T_{01} \cdot T_{12} \cdot T_{23} = \begin{bmatrix} \cos(q_2) & 0 & \sin(q_2) & L_2 \sin(q_2) \\ \sin(q_1) \sin(q_2) & \cos(q_1) & -\cos(q_2) \sin(q_1) & -\sin(q_1)(L_1 + L_2 \cos(q_2)) \\ -\cos(q_1) \sin(q_2) & \sin(q_1) & \cos(q_1) \cos(q_2) & \cos(q_1)(L_1 + L_2 \cos(q_2)) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.17)$$

Poziția efectorului final în funcție de unghiurile cuplelor q_1 și q_2 este dată de primele 3 linii din coloana a 4-a:

$$\begin{aligned} x &= L_2 \sin(q_1), \\ y &= L_1 \sin(q_1) - L_2 \sin(q_1) \cos(q_2), \\ z &= L_1 \cos(q_1) + L_2 \cos(q_1) \cos(q_2). \end{aligned} \quad (3.18)$$

Orientarea efectorului final este dată de primele 3 linii și coloane, grupate în submatricea

$$R = \begin{bmatrix} \cos(q_2) & 0 & \sin(q_2) \\ \sin(q_1) \sin(q_2) & \cos(q_1) & -\cos(q_2) \sin(q_1) \\ -\cos(q_1) \sin(q_2) & \sin(q_1) & \cos(q_1) \cos(q_2) \end{bmatrix}. \quad (3.19)$$

Mai departe calculăm Jacobianul robotului coloană cu coloană:

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} = [J_1 J_2]. \quad (3.20)$$

Fiindcă ambele cuple sunt de rotație, fiecare coloană se calculează pe baza relației (2.15), cu o modificare importantă. În prezentarea din capitolul 2, Jacobianul este determinat pornind de la modelul geometric direct obținut prin metoda DH, în care axa de mișcare este tot timpul axa Z . În modelul geometric determinat de noi mai sus, axele de mișcare sunt X pentru cupla 1, respectiv Y pentru cupla 2. Ca urmare termenul $R_{0,i}(:, 3)$ din (2.15)

devine $R_{0,1}(:, 1)$ pentru cupla 1, respectiv $R_{0,2}(:, 2)$ pentru cupla 2. În urma calculului se obține **Jacobianul**:

$$J = \begin{bmatrix} 0 & L_2 \cos(q_2) \\ -L_2 \cos(q_1) \cos(q_2) - L_1 \cos(q_1) & L_2 \sin(q_1) \sin(q_2) \\ -L_2 \sin(q_1) \cos(q_2) - L_1 \sin(q_1) & -L_2 \cos(q_1) \sin(q_2) \\ 1 & 0 \\ 0 & \cos(q_1) \\ 0 & \sin(q_1) \end{bmatrix}. \quad (3.21)$$

Astfel, pentru elementul 2, matricele Jacobian pentru viteze liniare și unghiulare sunt

$$J_{vc2} = J_{vc} = \begin{bmatrix} 0 & \frac{L_2}{2} \cos(q_2) \\ -\frac{L_2}{2} \cos(q_1) \cos(q_2) - L_1 \cos(q_1) & \frac{L_2}{2} \sin(q_1) \sin(q_2) \\ -\frac{L_2}{2} \sin(q_1) \cos(q_2) - L_1 \sin(q_1) & -\frac{L_2}{2} \cos(q_1) \sin(q_2) \end{bmatrix}, \quad (3.22)$$

$$J_{\omega 2} = J_{\omega} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(q_1) \\ 0 & \sin(q_1) \end{bmatrix}, \quad (3.23)$$

iar pentru elementul 1 ²⁶:

$$J_{vc1} = \begin{bmatrix} 0 & 0 \\ -\frac{L_1}{2} \cos(q_1) & 0 \\ -\frac{L_1}{2} \sin(q_1) & 0 \end{bmatrix}, \quad (3.24)$$

$$J_{\omega 1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.25)$$

Trecând mai departe la **modelul dinamic** propriu-zis, începem prin a detalia componenta de translație și rotație a **energiei cinetice** ($K = K_{transl} + K_{rot}$):

$$K_{transl} = \frac{1}{2} m_1 v_{c1}^T v_{c1} + \frac{1}{2} m_2 v_{c2}^T v_{c2} = \frac{1}{2} \dot{q}^T (m_1 J_{vc1}^T J_{vc1} + m_2 J_{vc2}^T J_{vc2}) \dot{q}, \quad (3.26)$$

$$K_{rot} = \frac{1}{2} \dot{q}^T (J_{\omega 2}^T R_2 I_2 R_2^T J_{\omega 2} + J_{\omega 1}^T R_1 I_1 R_1^T J_{\omega 1}) \dot{q}, \quad (3.27)$$

²⁶Se calculează considerând ca elementul 2 nu există.

Capitolul 3. Modelul dinamic

cu²⁷

$$R_2 = R, \quad R_1 = T_{01}(1 : 3, 1 : 3), \quad I_2 = \text{diag}\{0, I_{2y}, 0\}, \quad I_1 = \text{diag}\{I_{1x}, 0, 0\}. \quad (3.28)$$

Dupa calcularea celor două componente ale energiei cinetice, putem obține matricea de inerție $D(q)$ sub forma

$$\begin{aligned} D(q) &= \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \\ &= \begin{bmatrix} I_{1x} + \frac{L_1^2 m_1}{4} + L_1^2 m_2 + \frac{L_2^2 m_2}{4} c_{q_2}^2 + L_1 L_2 m_2 c_{q_2} & 0 \\ 0 & \frac{m_2 L_2^2}{4} + I_{2y} \end{bmatrix}. \end{aligned} \quad (3.29)$$

În determinarea matricei $C(q, \dot{q})$, calculăm mai întâi termenii c_{ijk} din (3.15):

$$\begin{aligned} c_{111} &= \frac{\partial d_{11}}{\partial q_1} + \frac{\partial d_{11}}{\partial q_1} - \frac{\partial d_{11}}{\partial q_1} = 0, \\ c_{112} &= \frac{\partial d_{21}}{\partial q_1} + \frac{\partial d_{21}}{\partial q_1} - \frac{\partial d_{11}}{\partial q_2} = \frac{L_2^2 m_2}{4} \sin(2q_2) + L_1 L_2 m_2 \sin(q_2), \\ c_{121} &= \frac{\partial d_{12}}{\partial q_1} + \frac{\partial d_{11}}{\partial q_2} - \frac{\partial d_{12}}{\partial q_1} = -\frac{L_2^2 m_2}{4} \sin(2q_2) - L_1 L_2 m_2 \sin(q_2), \\ c_{122} &= \frac{\partial d_{22}}{\partial q_1} + \frac{\partial d_{21}}{\partial q_2} - \frac{\partial d_{12}}{\partial q_2} = 0, \\ c_{211} &= \frac{\partial d_{11}}{\partial q_2} + \frac{\partial d_{12}}{\partial q_1} - \frac{\partial d_{21}}{\partial q_1} = c_{121}, \\ c_{212} &= \frac{\partial d_{21}}{\partial q_2} + \frac{\partial d_{22}}{\partial q_1} - \frac{\partial d_{21}}{\partial q_2} = 0, \\ c_{221} &= \frac{\partial d_{12}}{\partial q_2} + \frac{\partial d_{12}}{\partial q_2} - \frac{\partial d_{22}}{\partial q_1} = 0, \\ c_{222} &= \frac{\partial d_{22}}{\partial q_2} + \frac{\partial d_{22}}{\partial q_2} - \frac{\partial d_{22}}{\partial q_2} = 0. \end{aligned}$$

În final obținem matricea

$$\begin{aligned} C(q, \dot{q}) &= \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{L_2^2 m_2}{8} s_{2q_2} \dot{q}_2 - \frac{1}{2} L_1 L_2 m_2 s_{q_2} \dot{q}_2 & -\frac{L_2^2 m_2}{8} s_{2q_2} \dot{q}_1 - \frac{1}{2} L_1 L_2 m_2 s_{q_2} \dot{q}_1 \\ \frac{L_2^2 m_2}{8} s_{2q_2} \dot{q}_1 + \frac{1}{2} L_1 L_2 m_2 s_{q_2} \dot{q}_1 & 0 \end{bmatrix}. \end{aligned} \quad (3.30)$$

²⁷Deoarece termenii care nu sunt pe diagonala principală a matricelor de inerție sunt neglijați în practică, considerăm doar momentele principale de inerție corespunzătoare fiecărei axe de rotație $\{I_x, I_y, I_z\}$. A se avea în vedere că momentele de inerție se exprimă în sistemele de coordonate locale, atașate obiectelor în cauză.

Energia potențială se obține prin înmulțirea masei elementului cu accelerația gravitațională și înălțimea centrului de greutate:

$$P_1 = m_1 g \frac{L_1}{2} \cos(q_1), P_2 = m_2 g \left(L_1 \cos(q_1) + \frac{L_2}{2} \cos(q_1) \cos(q_2) \right), P = P_1 + P_2. \quad (3.31)$$

Vectorul de gravitație se determină folosind (3.16):

$$G(q) = \begin{bmatrix} -\frac{m_1 g L_1 + 2m_2 g L_1}{2} \sin(q_1) - \frac{m_2 g L_2}{2} \sin(q_1) \cos(q_2) \\ -\frac{m_2 g L_2}{2} \cos(q_1) \sin(q_2) \end{bmatrix}. \quad (3.32)$$

Odată obținute matricele D , C și G , modelul dinamic (3.13) pentru partea mecanică a brațului robotic este complet.

3.3 Modelarea elementelor de execuție

Până acum am discutat în general despre roboți (brațe robotice) ca procese pur mecanice. Dar un astfel de proces trebuie completat cu elemente de execuție (actuatoare) și elemente de măsură (senzori). Dinamica senzorilor este de multe ori ignorată ²⁸, iar caracteristica intrare-ieșire se consideră liniară. Dinamica actualelor este însă de multe ori important de luat în considerare ²⁹.

Cuplele unui braț robotic pot să fie acționate de actuatoare electromecanice, hidraulice sau pneumatice ³⁰. Cele mai des întâlnite actuatoare pentru brațe robotice sunt motoarele electrice de curent continuu [13]-Cap 4.

Considerăm mai departe că fiecare cuplă a brațului robotic este acționată de un motor electric de curent continuu. Schema electrică a motorului este reprezentată în Figura 3.3. Din legile lui Kirchhoff rezultă ecuația tensiunilor electrice din circuit:

$$v = R_a i + L_a \frac{di}{dt} + v_b \quad (3.33)$$

unde i este curentul prin armătură, v_i este tensiunea de comandă (prin armătură), R_a este rezistența armăturii, iar L_a este inductanța armăturii. v_b este tensiunea back EMF (indusă de forța electromotoare) și este proporțională cu viteza motorului:

$$v_b = K_b \omega. \quad (3.34)$$

Cuplul τ dat de motor depinde de curentul prin armătură:

$$\tau_m = K_m i, \quad (3.35)$$

²⁸Se consideră că dinamica senzorilor este mult mai rapidă decât dinamica procesului.

²⁹O referință recomandată pentru mai multe detalii despre senzori și actuatoare este [9].

³⁰Pentru detalii referitoare la acționări pneumatice și hidraulice a se consulta [16] și [3].

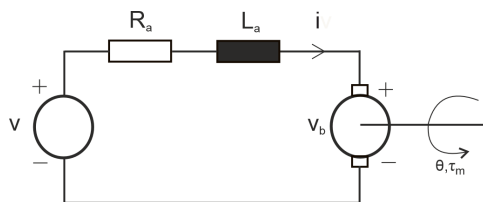


Figura 3.3: Schema electrică a unui motor de curent continuu

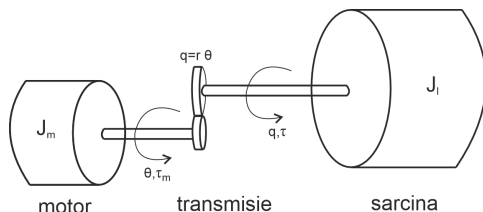


Figura 3.4: Schema mecanică a motorului care acționează o cuplă (sarcina este dată de legătura cuplei)

unde K_m este constanta de cuplu. Ecuația pentru partea mecanică rezultă din legea a 2-a a lui Newton:

$$J_m \frac{d^2\theta}{dt^2} + B_m \frac{d\theta}{dt} + r\tau = \tau_m, \quad (3.36)$$

unde J_m este momentul de inerție al motorului ³¹, iar B_m este coeficientul de frecare vâscoasă. Raportul de transmisie r pentru angrenajul cu roți este definit ca

$$r = \frac{q}{\theta} \leq 1, \quad (3.37)$$

în care q este variabila cuplei acționată de motor. Figura 3.4 prezintă schema mecanică a motorului. Sarcina este dată de legătura robotului atașată cuplei, și este caracterizată de un cuplu de sarcină τ (datorită mișcării legăturii) și o inerție de sarcină J_l .

Din ecuațiile (3.33)-(3.37) rezultă modelul matematic pentru motorul de curent continuu controlat în armătură, pentru fiecare cuplă i a robotului ³²:

$$J_{mi}\ddot{\theta}_i + B_i\dot{\theta}_i + r_i\tau_i = (K_{mi}/R_{ai})v_i, \quad i = 1, \dots, n \quad (3.38)$$

unde θ_i este poziția unghiulară a rotorului corespunzător motorului i . Coeficientul B_i este dat de expresia

$$B_i = B_{mi} + K_{bi} \frac{K_{mi}}{R_{ai}}.$$

³¹Considerăm că include și inerția transmisiei.

³²Am neglijat inductanța armăturii: $L_{ai} \approx 0$.

Ecuatiile pentru toate motoarele care acționează un braț robotic cu n cuple pot fi grupate în forma matriceală:

$$J_m \ddot{\theta} + B \dot{\theta} + R\tau = K_m v, \quad (3.39)$$

unde

$$J_m = \text{diag}\{J_{mi}\}, \quad B = \text{diag}\{B_i\}, \quad R = \text{diag}\{r_i\}$$

$$K_m = \text{diag}\{K_{mi}/R_{ai}\}, v = [v_1 \dots v_n]^T, \theta = [\theta_1 \dots \theta_n]^T, \tau = [\tau_1 \dots \tau_n]^T, i = 1, \dots, n.$$

Exercițiul 15. Considerăm motorul de curent continuu folosit pentru acționarea cuplelor robotului industrial PUMA 560, cu următorii parametri [7]: $R_m = 5.1 \Omega$, $K_b = 0.22 V s/\text{rad}$, $K_m = 0.228 N m/A$, $B_m = 817 \cdot 10^{-6} N m s/\text{rad}$, $J_m = 200 \cdot 10^{-6} kg m^2$, $r = 1/107$, $v_{max} = 0.9 V$. Să se proiecteze un regulator de tip PD pentru controlul poziției, pe baza funcției de transfer a motorului (se ignoră cuplul de sarcină - $\tau = 0$).

Soluție. La modelul (3.38) aplicăm transformata Laplace pentru condiții inițiale zero ($\theta(0) = 0, \dot{\theta}(0) = 0$):

$$J_m \theta(s) s^2 + B \theta(s) s = (K_m/R_a) v(s).$$

Funcția de transfer se calculează ca

$$H(s) = \frac{\theta(s)}{v(s)} = \frac{K_m/R_a}{J_m s^2 + B s},$$

pe care o putem aduce la forma standard

$$H(s) = \frac{K}{s(Ts + 1)},$$

unde K este amplificarea procesului iar T este constanta de timp. Pentru parametrii numerici menționați mai sus, din calcul rezultă $T = 0.0189$ și $K = 4.217$. Regulatorul PD are funcția de transfer

$$H_{PD} = K_p + K_d s = K_p (T_d s + 1),$$

unde $T_d = K_d/K_p$. Dacă dorim să compensăm inerția (dinamica) procesului, putem alege $T_d = T$. Funcția de transfer în buclă închisă se poate calcula ca

$$H_{cl} = \frac{H H_{PD}}{1 + H H_{PD}} = \frac{1}{T_{cl} s + 1},$$

cu constanta de timp $T_{cl} = 1/(K K_p)$. Timpul de răspuns pentru acest sistem de ordinul întâi este $t_r = 4 T_{cl}$. Deci parametrul K_p îl putem obține impunând un anumit timp de reglare t_r pentru răspunsul în buclă închisă.

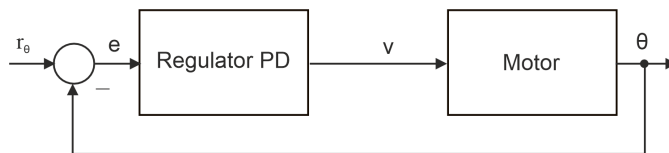


Figura 3.5: Structura de conducere pentru controlul poziției motorului

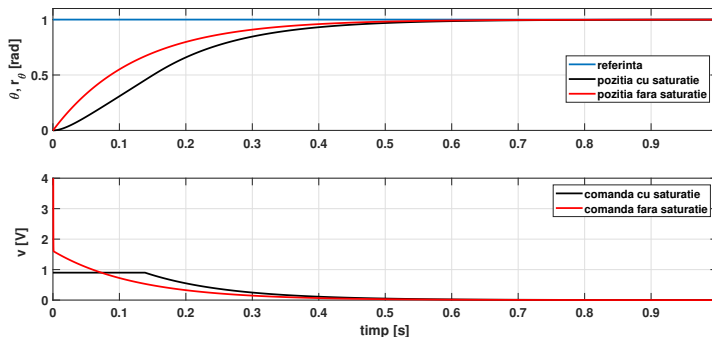


Figura 3.6: Rezultate de simulare cu regulatorul PD pentru controlul poziției motorului

Considerând de exemplu un timp de reglare de $t_r = 0.5$ s, rezultă $K_p = 18.9708$, iar apoi din calcul rezultă $K_d = 0.3585$.

Figura 3.5 arată structura de conducere a sistemului nostru, în care r_θ este referința de poziție. Figura 3.6 prezintă răspunsul sistemului la o referință treaptă unitară la momentul $t = 0$ pentru condiții inițiale nule (partea de sus), precum și comanda v aplicată motorului (partea de jos). În practică comanda aplicată motorului este saturată (v_{max}) pentru a proteja motorul, ca urmare sunt ilustrate două cazuri în figură: atunci când avem saturație pe comandă și atunci când o neglijăm. Se poate observa că în ambele cazuri răspunsul sistemului este cu eroare staționară zero și timp de reglare de 0.5 secunde, doar că atunci când nu avem saturație avem o tensiune de comandă foarte mare la pornire. Intrarea comenzii în saturație pentru o perioadă mare de timp poate să ne afecteze performanțele de reglare. Soluția practică cea mai simplă în acest caz este de a face regulatorul mai lent.

□

3.4 Modelul neliniar al unui robot

Modelul dinamic (3.13) al robotului ca sistem mecanic a inclus până acum inerția, gravitația, forțele centripete și efectul de Coriolis. În practică este uneori necesar să se țină cont și de alte forțe, precum forța de frecare, jocuri

sau forțe elastice. În lucrarea de față vom considera că este suficient să adăugăm forța de frecare la nivelul cuplelor robotului.

Forța de frecare la nivelul unei cuple i a robotului este de multe ori modelată ca o expresie neliniară de forma [30]:

$$F(\dot{q}_i) = F_v \dot{q} + F_c \operatorname{sgn}(\dot{q}_i), \quad (3.40)$$

unde F_v este coeficientul frecării vâscoase, iar F_c este coeficientul frecării de tip Coulomb. Frecarea vâscoasă mai este numită și frecare dinamică și este aproximată de multe ori ca fiind liniară. Frecarea statică (în jurul vitezei zero), cum este frecarea de tip Coulomb, deși este importantă în practică, de multe ori este ignorată în etapa de analiză și proiectare a unui regulator, deoarece este dificil de modelat exact, și complică mult regulatorul rezultat din proiectare ³³.

Considerăm mai departe doar forța de frecare dinamică, care se poate scrie sub forma matriceală pentru toate cuplele n ale unui robot

$$F(\dot{q}) = \begin{bmatrix} F_{b1} & 0 & \dots \\ \vdots & \ddots & \\ 0 & & F_{bn} \end{bmatrix} \dot{q} = F_b \dot{q}, \quad (3.41)$$

unde coeficienții $b_1 - b_n$ sunt pozitivi și se determină de obicei experimental. Modelul dinamic al robotului devine astfel

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau. \quad (3.42)$$

Mai departe încorporăm și modelul motoarelor care acționează cuplele. Astfel, dacă folosim relația (3.37) pentru a înlocui θ cu q în (3.39), îl izolăm pe τ și îl substituim în (3.42) obținem

$$(J_m + R^2 D(q))\ddot{q} + (B + R^2 C(q, \dot{q}))\dot{q} + R^2 F(\dot{q}) + R^2 G(q) = R K_m v, \quad (3.43)$$

care se poate scrie mai compact printr-o schimbare de notații:

$$D_p(q)\ddot{q} + C_p(q, \dot{q})\dot{q} + F_p(\dot{q}) + G_p(q) = K_r v. \quad (3.44)$$

Acesta este modelul neliniar pentru brațul robotic ca proces electro-mecanic pe care dorim să îl controlăm, care are ca intrări tensiunile de comandă ale motoarelor v_i , iar ca iesiri măsurate unghiurile cuplelor q_i .

³³Un exemplu de abordare în vederea identificării experimentale a parametrilor forțelor de frecare pentru un robot de tip UR5 se poate găsi în [27].

Partea II

Controlul unui robot

În partea a II-a ne vom axa pe tot ce ține de proiectarea structurii de conducere a unui robot. Controlul unui braț robotic se face de cele mai multe ori în spațiul cuplelor, iar sarcinile pe care efectorul final al robotului trebuie să le opereze sunt de obicei precizate în spațiul cartezian. Pentru a face legătura dintre cele două tipuri de obiective urmărite, presupunem structura de principiu din Figura 3.7. În această structură avem un regulator proiectat astfel încât referința de poziție q_d să urmărească poziția măsurată a cuplelor q ³⁴. La nivelul referinței, avem un model geometric invers, determinat analitic sau numeric, care ne calculează online referința pentru cuple q_d din referința la nivelul poziției efectorului final p_d . Referința p_d ar putea fi de exemplu generată de un planificator de traiectorie adițional. Fiindcă eroarea de urmărire la nivelul cuplelor $e_q = q_d - q$ poate să nu fie suficient de relevantă raportat la sarcina pe care robotul trebuie să o efectueze, este folosit și un model geometric direct care transformă traiectoria măsurată a robotului din spațiul cuplelor în spațiul cartezian ($q \rightarrow p$). Astfel se poate calcula ulterior și eroarea de urmărire la nivel de traiectorie a efectorului final $e_p = p_d - p$.

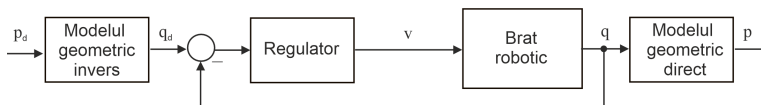


Figura 3.7: Structura de control a unui robot în spațiul cuplelor

³⁴În cazul în care regulatorul se bazează și pe informații de viteză sau accelerație, structura se completează cu semnale adiționale.

Capitolul 4

Controlul liniar

În acest capitol vom arăta cum se poate proiecta o structură de conducere liniară pentru un braț robotic. În prima secțiune descriem cum se poate aduce modelul dinamic al robotului, dezvoltat în capitolul anterior, la o formă liniară. În secțiunea a doua prezentăm cea mai simplă abordare în controlul roboților - controlul independent în cuple, în care presupunem regulatoare cu reacții după ieșire (poziții măsurate pentru fiecare cuplă) într-o abordare *SISO* (single input - single output) din Teoria sistemelor. În ultima secțiune vom arăta cum se poate proiecta un regulator direct pentru toate cuplele robotului, abordarea MIMO (multiple input - multiple output), pe baza unei abordări în spațiul stărilor. Ambele abordări vor fi ilustrate prin exemple și rezultate de simulări.

4.1 Determinarea modelului liniarizat

În această secțiune vom descrie două modalități de a ajunge la modelul liniar al unui braț robotic, pornind de la modelul neliniar (3.44). Prima metodă se bazează pe aproximări succesive, ținând cont de interpretarea fizică a termenilor care apar în modelul robotului. A doua abordare este mai sistematică, și se referă la metoda generală de liniarizare a unui sistem neliniar din Teoria sistemelor.

4.1.1 Liniarizarea prin aproximări

Pentru început, rescriem modelul dinamic (3.44) la nivel de fiecare cuplă independentă ([1]):

$$(J_{mi} + r_i^2 d_{ii})\ddot{q}_i + B_i \dot{q}_i + r_i F_{bi} = (r_i K_{mi} / R_{ai}) v_i - r_i^2 w_i, \quad i = 1, \dots, n, \quad (4.1)$$

unde d_{ii} include doar termenii diagonali constanți ai matricei $D(q)$, în timp ce termenul w_i reprezintă o perturbație care include toți termenii i rămași

ai matricei $D(q)$, precum și restul componentelor i ale matricelor $C(q, \dot{q})$ și $G(q)$ ³⁵. Este important de remarcat faptul că pentru un raport de transmisie r_i mic (subunitar), și dacă termenii care țin cont de forțele de rotație în $C(q, \dot{q})$ nu sunt foarte mari, atunci dinamica procesului se poate aproxima ca un sistem de **n ecuații liniare de ordinul 2**.

Considerăm mai departe cazul când același tip de motor este folosit pentru acționarea fiecărei cuple. Astfel, prin incorporarea termenului de frecare F_{bi} în perturbația w , putem rescrie modelul (4.1) ca ³⁶

$$\underbrace{(J_m + r^2 d_{ii})}_{J_{pi}} \ddot{q}_i + B_i \dot{q}_i = \underbrace{(r K_{mi} / R_{ai})}_{K_i} v_i - w_{pi}, \quad i = 1, \dots, n. \quad (4.2)$$

Exercițiul 16. Considerăm robotul cu două cuple de rotație din Figura 3.1, și parametrii robotului: $L_1 = 0.095$ m, $L_2 = 0.1$ m, $m_1 = 0.095$ kg, $m_2 = 0.37$ kg, $g = 9.81$ m/s², $I_{1x} = 2.27 \cdot 10^{-2}$ kg m², $I_{2y} = 2.27 \cdot 10^{-2}$ kg m², $F_{b1} = 0.24$, $F_{b2} = 0.16$, $r = 0.1$, $K_m = \text{diag}\{[0.1 \ 0.1]\}$. Neglijăm frecarea internă $F_{mi} = 0$, amortizarea $B_i = 0$, și momentul de inerție $J_{mi} = 0$ la nivelul fiecărui motor. Să se calculeze modelul liniarizat folosind aproximația (4.2).

Soluție. Ținând cont de neglijarea dinamicii motorului, înlocuind valorile parametrilor în relația (4.2) obținem:

$$0.1^2 d_{11} \ddot{q}_1 = 0.1^2 v_1 - w_{p1}, \quad 0.1^2 d_{22} \ddot{q}_1 = 0.1^2 v_2 - w_{p2}.$$

Efectuând calculul pentru a determina valoarea termenilor constanți din d_{11} și d_{22} rezultă în final:

$$J_{p1} = 2.63 \cdot 10^{-4}, \quad J_{p2} = 2.36 \cdot 10^{-4}, \quad K_1 = K_2 = 0.01. \quad (4.3)$$

□

4.1.2 Metoda generală de liniarizare

Începem prin a defini mărimile de stare ale sistemului (3.44) ca fiind pozițiile și vitezele cuplelor: $x_1 = q$ și $x_2 = \dot{q}$. Vectorul de stare este $x = [x_1^T \ x_2^T]^T$, iar intrarea sistemului este $u = v$. Ca urmare, putem rescrie modelul neliniar al sistemului sub forma standard neliniară în spațiul stărilor

$$\dot{x} = f(x, u) = \begin{bmatrix} x_2 \\ D_p^{-1}(x_1)(-C_p(x_1, x_2)x_2 - F_p(x_2) - G(x_1) + K_p u) \end{bmatrix} \quad (4.4)$$

³⁵Se poate spune că efectul neliniarităților este ascuns în perturbația w .

³⁶B și K devin scalari.

Considerăm că sistemul are punctul de echilibru x_e pentru comanda de echilibru u_e , adică $f(x_e, u_e) = 0$. Ipoteza de bază pentru liniarizare este că sistemul operează în vecinătatea acestui punct de echilibru. Dacă dezvoltăm în serie Taylor de ordinul întâi obținem

$$\dot{x} = \frac{\partial f}{\partial x}|_{(x^e, u^e)} \cdot (x - x_e) + \frac{\partial f}{\partial u}|_{(x^e, u^e)} \cdot (u - u_e), \quad (4.5)$$

din care rezultă **sistemul liniarizat**

$$\dot{x}_l = Ax_l + Bu_l, \quad (4.6)$$

unde $x_l = x - x_e$ și $u_l = u - u_e$. În particular, dacă punctul de echilibru este $x_e = 0$ pentru comanda de echilibru $u_e = 0$, atunci sistemul liniarizat este

$$\dot{x} = Ax + Bu. \quad (4.7)$$

Următorul exercițiu exemplifică liniarizarea pentru cazul unui sistem mecanic cu un singur grad de libertate.

Exercițiul 17. Considerăm pendulul mecanic suspendat din Figura 4.1. Brațul pendulului are lungime l iar la capăt are o bilă de masă m . Pendulul poate efectua o mișcare de rotație în plan, cu unghiul q . Ca intrare a sistemului avem cuplul aplicat τ . Din legea a 2-a a lui Newton putem scrie ecuația de mișcare

$$ml^2\ddot{q}(t) = \tau(t) - mgl \sin q(t) - bl\dot{q}(t), \quad (4.8)$$

în care în membrul drept avem cuplul aplicat, cuplul datorită gravitației și cuplul datorită forței de frecare. Să se determine modelul liniarizat.

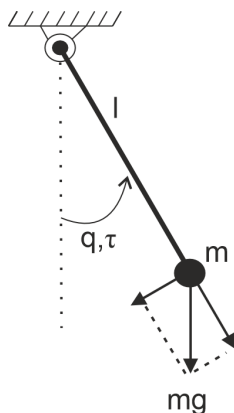


Figura 4.1: Pendul mecanic

Soluție. Adoptăm mărimile de stare $x_1 = q$ și $x_2 = \dot{q}$. Vectorul de stare este $x = [x_1 \ x_2]^T$, iar intrarea este $u = \tau$. Modelul neliniar în spațiul stărilor este

$$\dot{x} = f(x, u) = \begin{bmatrix} f_1(x, u) \\ f_2(x, u) \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin(x_1) - \frac{b}{ml} x_2 + \frac{1}{ml^2} u \end{bmatrix}.$$

Ne interesează dinamica în jurul punctului de echilibru $x_e = 0$ cu $u_e = 0$. Matricile A și B din modelul liniarizat (4.7) se calculează ca

$$A = \left. \frac{\partial f}{\partial x} \right|_{(x^e, u^e)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \Bigg|_{(x^e, u^e)} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{b}{ml} \end{bmatrix},$$

$$B = \left. \frac{\partial f}{\partial u} \right|_{(x^e, u^e)} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix} \Bigg|_{(x^e, u^e)} = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}.$$

□

Pentru roboți cu mai mult de două cuple, calculul derivatelor pentru liniarizare devine destul de elaborat, iar în practică se apelează deseori la calcul simbolic.

Exercițiul 18. Considerăm robotul cu 2 grade de libertate din Figura 3.1. Parametrii robotului sunt: $L_1 = 0.095$ m, $L_2 = 0.1$ m, $m_1 = 0.095$ kg, $m_2 = 0.37$ kg, $g = 9.81$ m/s², $I_{1x} = 2.27 \cdot 10^{-2}$ kg m², $I_{2y} = 2.27 \cdot 10^{-2}$ kg m², $F_{b1} = 0.24$, $F_{b2} = 0.16$, $r = 1$, $K_m = \text{diag}\{[1 \ 1]\}$. Neglijăm frecarea internă $F_{mi} = 0$, amortizarea $B = 0$, și momentul de inerție $J_m = 0$ la nivelul fiecărui motor. Să se calculeze modelul liniarizat prin calcul simbolic în MATLAB³⁷, considerând punctul de echilibru $x_e = 0$ pentru comanda $u_e = 0$.

4.2 Controlul liniar prin reacție după ieșire

Controlul prin reacție după ieșire în abordarea SISO, numit în literatură **control independent în cuple**, este o strategie clasică de control, în care este proiectat câte un regulator pentru fiecare cuplă individuală (e.g. Figura 4.2 prezintă structura de conducere pentru un robot cu două cuple). Efectul celorlalte cuple este considerat ca o perturbație pe care regulatorul trebuie să o rejeteze.

Dorim astfel să proiectăm câte un regulator liniar pentru fiecare cuplă i , care să asigure urmărirea referinței q_{di} pentru fiecare poziție q_i , și să

³⁷Pentru calculul derivatelor din (4.5) se poate folosi funcția *jacobian*, iar pentru evaluarea derivatelor în punctul de echilibru funcția *eval*.

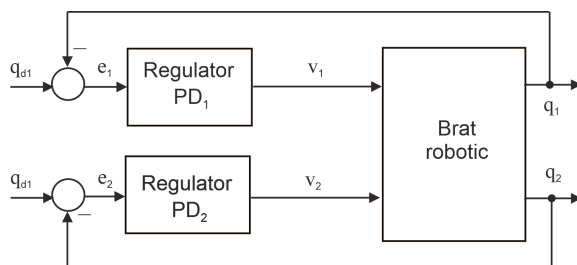


Figura 4.2: Control independent în cuple cu regulatoare PD pentru un robot cu două cuple

rejechteze perturbația w_{pi} . Fiindcă regulatorul este proiectat pe baza modelului liniarizat, garanțiile de stabilitate discutate aici se referă la stabilitatea locală în vecinătatea punctului de liniarizare.³⁸

În rezolvarea problemei de control, considerăm pentru început **regulatoare standard de tip PD** (proporțional derivativ)

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i, \quad (4.9)$$

unde e_i reprezintă eroarea de urmărire, definită ca $e_i = q_{di} - q_i$, iar derivata erorii este $\dot{e}_i = \dot{q}_{di} - \dot{q}_i$.

Dacă presupunem că referința este constantă pe porțiuni, adică $\dot{q}_{di} = 0$, și înlocuim (4.9) în (4.2), obținem

$$J_{pi}\ddot{q}_i(t) + (B_i + K_i K_{di})\dot{q}_i(t) + K_i K_{pi}q_i(t) = K_i K_{pi}q_{di}(t) - w_{pi}(t), \quad i = 1, \dots, n. \quad (4.10)$$

Aplicând transformata Laplace pentru condiții inițiale nule obținem funcția de transfer care face legătura dintre ieșirile q_i și intrările de tip referință q_{di} și perturbație w_{pi} :

$$q_i(s) = \frac{K_i K_{pi}}{J_{pi}s^2 + (B_i + K_i K_{di})s + K_i K_{pi}} q_{di}(s) - \frac{1}{J_{pi}s^2 + (B_i + K_i K_{di})s + K_i K_{pi}} w_{pi}(s), \quad i = 1, \dots, n. \quad (4.11)$$

Ecuatiile caracteristice pentru fiecare cuplă i sunt

$$J_{pi}s^2 + (B_i + K_i K_{di})s + K_i K_{pi} = 0,$$

care se pot rescrie ca

$$s^2 + \frac{B_i + K_i K_{di}}{J_{pi}}s + \frac{K_i K_{pi}}{J_{pi}} = 0.$$

³⁸Pentru o discuție referitoare la stabilitatea globală a sistemului cu astfel de regulatoare liniare, pe baza de funcții Lyapunov, a se vedea [13]-Cap 8.

Considerând forma standard a ecuației de ordinul 2

$$s^2 + 2\zeta_i\omega_{ni}s + \omega_{ni}^2 = 0,$$

putem deduce parametrii regulatorului în funcție de factorul de amortizare ζ și pulsația naturală ω_n :

$$K_{pi} = \frac{J_{pi}\omega_{ni}^2}{K_i}, \quad K_{di} = \frac{2\zeta_i\omega_{ni}J_{pi} - B_i}{K_i}. \quad (4.12)$$

Deși există în literatură relații care ne dau legătura dintre ζ și ω_n cu supra-reglajul M_p și timpul de răspuns t_r pentru un sistem de ordinul doi [15], aici vom adopta o abordare empirică. Deoarece dorim ca răspunsul sistemului să fie fără supra-reglaj, adică practic nu dorim oscilații, putem alege ζ la valoarea 1 (răspuns supra-amortizat). ω_n ne spune cât de rapid să răspundă sistemul, deci alegem o valoare cât mai mare, dar având grijă ca să nu intre comanda v_i în saturație.

Strategia de control prezentată mai sus se dovedește de cele mai multe ori foarte eficientă în practică. Totuși uneori nu puteam garanta o eroare staționară zero, iar atunci trebuie să extindem structura la **reglatoarele PID** (proporțional integrator derivativ)

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i + K_{ii} \int_0^t e_i dt, \quad (4.13)$$

în care avem deja trei parametri de proiectat pentru fiecare regulator.

Deoarece eroarea staționară este dată de termenul gravitațional, o alternativă la regulatorul PID este **regulatorul PD cu un termen gravitațional suplimentar** [1]

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i + G_{pi}(q). \quad (4.14)$$

În acest caz regulatorul devine neliniar datorită termenului $G_{pi}(q)$, iar comenzile nu mai sunt decuplate. Eroarea staționară este însă dependentă de acuratețea cu care știm componenta gravitațională a modelului.

Exercițiul 19. Considerăm robotul din Figura 3.1 liniarizat prin aproximări în Exercițiul 16. Tensiunea de comandă v_i este limitată în domeniul $[-1.18, 1.18] V$.

- a) Să se proiecteze o structură de conducere independentă în cuple folosind regulatoarele PD.
- b) Să se testeze structura de conducere pe modelul neliniar al robotului în MATLAB/Simulink pentru semnale de referință treaptă. Interpretați rezultatele în termeni de supra-reglaj, timp de reglare și eroare staționară.

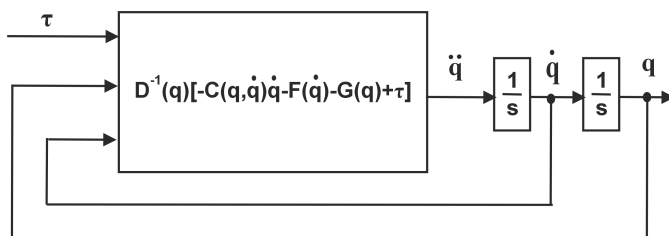


Figura 4.3: Schema de principiu pentru partea mecanică a robotului

Soluție. Proiectăm două regulatoare PD de forma

$$v_1 = K_{d1}\dot{e}_1 + K_{p1}e_1,$$

$$v_2 = K_{d2}\dot{e}_2 + K_{p2}e_2,$$

folosind relația (4.12). Alegem factorul de amortizare $\zeta = 1$ astfel încât să avem un răspuns aperiodic (fără oscilații). Începem cu o valoare pentru pulsația naturală $\omega_n = \omega_{n1} = \omega_{n2}$ de 0.1, pe care o tot creștem prin încercări succesive până când răspunsul sistemului la treaptă este suficient de rapid, iar comanda nu intră în saturație pentru un interval de timp foarte mare. În final ajungem la valoarea $\omega_n = 12$ care ne oferă performanțe suficient de bune. Parametrii reguletoarelor PD sunt: $K_{p1} = 3.7805, K_{p2} = 3.402, K_{d1} = 0.6301, K_{d2} = 0.567$.

Mai departe implementăm structura de conducere din Figura 4.2 în Simulink. Modelul neliniar al procesului poate fi implementat pornind de la schema de principiu din Figura 4.3, care prezintă doar partea mecanică a robotului (fără motor). Pe lângă cele două blocuri de tip integrator, blocul care include toate expresiile neliniare din model se poate implementa printr-un bloc de tip *MATLAB function*. Rezultatele de simulare sunt prezentate în Figura 4.4. Se poate observa că răspunsul nu are suprareglaj, timpul de răspuns este sub o secundă, însă avem o eroare staționară datorită neliniarității procesului. Din Figura 4.5 se poate observa că mărimea de comandă intra în saturație pentru o scurtă perioadă de timp (aproximativ 0.2 sec), ceea ce sugerează că nu putem face răspunsul sistemului mult mai rapid.

□

Exercițiul 20. Să se proiecteze o structură de conducere independentă în cuple folosind regulatoare PD cu gravitație pentru robotul din Figura 3.1, liniarizat prin aproximări în Exercițiul 16, și cu tensiunea de comandă v_i limitată în domeniul $[-1.18, 1.18] V$. Să se testeze structura de conducere prin simulări cu modelul neliniar al procesului:

- a) pentru semnale de referință treaptă, și să se interpreteze rezultatele în termeni de suprareglaj, timp de reglare și eroare staționară;

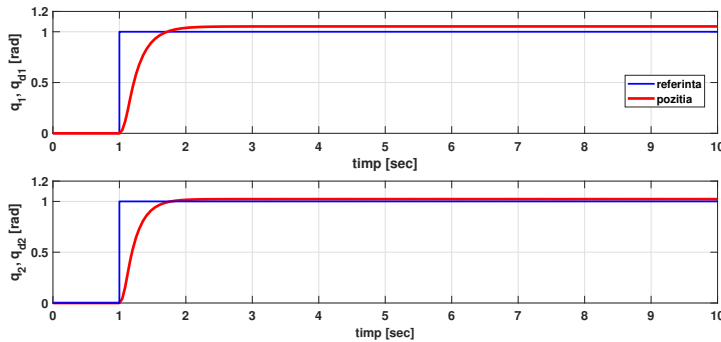


Figura 4.4: Rezultate de simulare pentru controlul independent în cuple cu reglatoare PD - răspunsul la referință treaptă

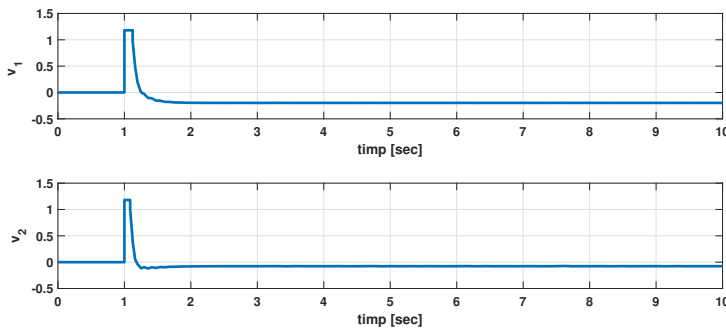


Figura 4.5: Rezultate de simulare pentru controlul independent în cuple cu reglatoare PD - comanda

b) pentru semnale de referință sinusoidale, respectiv să se interpreteze rezultatele.

4.3 Controlul liniar prin reacție după stare

Abordarea SISO prin reacție după ieșire de tip control independent în cuple prezentată anterior funcționează bine atunci când mișcarea robotului este relativ lentă, când raportul de transmisie este mare, astfel încât efectele neliniare de cuplaj între legăturile robotului se pot neglija. În cazul motoarelor moderne și mai performante, care sunt conduse direct în cuplu, efectul cuplajelor este semnificativ mai mare. Suplimentar, atunci când apare și o componentă elastică la nivelul cuplelor, reglatoarele de tip PD pentru fiecare cuplă se dovedesc neperformante [31]-Cap. 6. O soluție de control mai avansată în acest caz o reprezintă **controlul prin reacție după stare**

³⁹, care ne permite impunerea unor performanțe mai stricte pentru sistemul în buclă închisă.

Pentru a putea proiecta un regulator cu reacție după stare, trebuie îndeplinite două condiții a priori: i) sistemul trebuie să fie controlabil, ii) toate stările trebuie să fie măsurabile. Începem cu condiția de **controlabilitate**. Considerăm modelul robotului obținut prin liniarizare de forma

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (4.15)$$

cu $x \in R^n$ și $u \in R^m$, n este ordinul sistemului, iar m numărul de intrări. Se definește matricea de controlabilitate

$$M_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B]. \quad (4.16)$$

Sistemul este controlabil dacă $\text{rang}(M_c) = n$.

A doua condiție este ca toate **stările** x_1, x_2, \dots, x_n să fie **măsurabile**. O relaxare a acestei condiții este ca stările nemăsurabile să fie măcar **observabile**, astfel încât să poată fi estimate cu un observator (estimator) de stare ⁴⁰.

4.3.1 Problema de stabilizare

Pentru regulatorul cu reacție după stare, începem cu **problema de stabilizare**: dorim să proiectăm o lege de reglare de forma $u(t) = -K \cdot x(t)$ care să stabilizeze sistemul. Structura de reglare este cea din Figura 4.6. Se pornește de la performanțele specificate prin polii impuși pentru sistemul în buclă închisă $p_i = [p_{i1} \ p_{i2} \ \dots \ p_{in}]^T$. Polinomul caracteristic impus este

$$\mu_i(s) = s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0 = (s - p_{i1})(s - p_{i2})\dots(s - p_{in}). \quad (4.17)$$

Regulatorul după stare are expresia generală

$$u = -Kx = -[K_1 \ K_2 \ \dots \ K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = -K_1x_1 - K_2x_2 - \dots - K_nx_n, \quad (4.18)$$

iar sistemul în buclă închisă devine

$$\dot{x} = Ax + B(-Kx) = \underbrace{(A - BK)}_{A_{cl}} x \quad (4.19)$$

³⁹Pentru mai multe detalii referitoare la modelare, control și estimare în spațiul stărilor a se vedea [10],[15],[28].

⁴⁰A se vedea secțiunea 4.3.3.

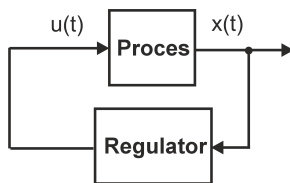


Figura 4.6: Structura de reglare după stare - stabilizare

Polii sistemului în circuit închis sunt soluțiile polinomului caracteristic

$$\mu_{cl}(s) = \det(sI_n - A_{cl}) = (s - p_{cl1})(s - p_{cl2}) \dots (s - p_{cln}).$$

Egalând polinoamele μ_i și μ_{cl} coeficient cu coeficient obținem un sistem de n ecuații cu n necunoscute, de unde rezultă soluția unică K ⁴¹.

Ne mai rămâne să răspundem la următoarea întrebare: cum alegem polii p_i pe care să îi impunem sistemului în circuit închis astfel încât să obținem performanțele dorite? Alegerea polilor nu este o problemă trivială, și se complică cu cât crește ordinul sistemului. În practică, ne putem ghida după câteva reguli sau indicații:

- Alegem polii reali negativi astfel încât sistemul să fie stabil, iar răspunsul sistemului să nu fie oscilant.
- Polii sistemului în buclă închisă se aleg *la stânga* polilor procesului, astfel încât sistemul în buclă închisă să fie mai rapid decât procesul.
- Polii impuși nu pot fi mutați prea mult spre stânga din considerente de realizabilitate practică - saturație comandă, pas de eșantionare⁴², efecte neliniare de frecvență mare, zgomot de măsură, etc.
- O posibilă abordare este impunerea polilor dominanți⁴³ conform polinomului $s^2 + 2\zeta\omega_n s + \omega_n^2$, în funcție de suprareglaj și timpul de reglare, iar restul polilor se adoptă mai la stânga pe axa reală.
- O alta abordare ar fi o configurație Butterworth a polilor în planul complex⁴⁴.

⁴¹În MATLAB soluția K se poate calcula cu funcția *place*, care presupune un algoritm mai bine condiționat numeric pentru sisteme de ordin mai mare.

⁴²Toată discuția din această carte este în timp continuu. În practică, implementarea reguletoarelor pe echipamente numerice presupune operația de discretizare (a se vedea [15]-Cap 8, [10]-Cap 3.3).

⁴³Polii dominanți pentru un sistem în timp continuu sunt polii cei mai aproape de axa imaginară a planului complex.

⁴⁴Pentru mai multe informații despre filtre Butterworth și configurația Butterworth a polilor a se vedea [11]-Cap. 8.7.

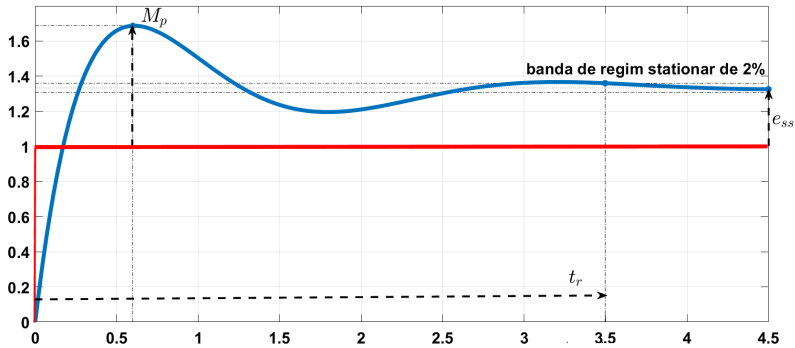


Figura 4.7: Indicatori de performanță pentru răspunsul treaptă al unui sistem: suprareglaj (M_p), timp de reglare (t_r) și eroare staționară (e_{ss})

4.3.2 Problema de urmărire

De multe ori în practică nu este suficient ca să garantăm că sistemul nostru este stabil, ci dorim să ne asigurăm și că ieșirea procesului $y(t)$ ne urmărește o anumită intrare de referință $r(t)$. Aceasta reprezintă **problema de urmărire**, care presupune de obicei indicatori de performanță ⁴⁵ precum (vezi de exemplu Figura 4.7):

- eroarea staționară, definită ca diferența dintre ieșirea și intrarea în regim staționar $e_{ss} = r_{ss} - y_{ss}$,
- timpul de reglare t_r , definit ca timpul în care ieșirea intră într-o bandă de $\epsilon\%$ fața de valoarea finală,
- supareglaj M_p , care reprezintă valoarea maximă cu care ieșirea depășește referința împărțit la valoarea finală (în procente).

De obicei se dorește o eroare staționară zero, un suprareglaj de maxim 10% și un timp de reglare cât mai mic (cu ϵ ales 1% sau 2%).

Începem prin a adăuga la modelul procesului în spațiul stărilor și ecuația de ieșire:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (4.20)$$

cu $x \in R^n, u \in R^m$ și $y \in R^p$. Vom prezenta mai departe două soluții mai des întâlnite pentru problema de urmărire: i) reglare după stare cu precompensare, ii) reglare după stare cu integrator.

⁴⁵Pentru mai multe detalii a se vedea [15].

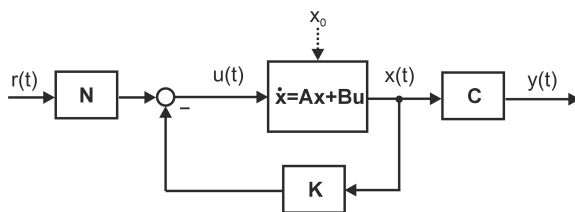


Figura 4.8: Reglare după stare cu precompensator - urmărire

O variantă simplă de a asigura urmărirea unei referințe este să adăugăm un *precompensator* pentru semnalul de referință (Figura 4.8), astfel încât legea de reglare (4.18) devine

$$u(t) = -Kx(t) + Nr(t). \quad (4.21)$$

În buclă închisă obținem

$$\dot{x}(t) = (A - BK)x(t) + BNr(t) \quad (4.22)$$

$$y(t) = Cx(t) \quad (4.23)$$

Considerăm compensatorul K ca fiind proiectat astfel încât să impună valorile proprii (polii) dorite matricei $A_{cl} = A - BK$. Precompensatorul N îl putem proiecta din condițiile de regim staționar:

$$0 = (A - BK)x_{ss} + BNr_{ss}, \quad y_{ss} = Cx_{ss}.$$

Rezultă relația intrare-ieșire în regim staționar

$$y_{ss} = -C(A - BK)^{-1}BNr_{ss},$$

la care dacă impunem condiția de eroare staționară nulă, adică $y_{ss} = r_{ss}$, obținem relația de calcul a precompensatorului:

$$N = -[C(A - BK)^{-1}B]^{-1}. \quad (4.24)$$

Principala limitare a legii de reglare (4.21) este lipsa robusteții la incertitudini parametriche și perturbații persistente. Ca urmare, uneori se apelează la o lege de reglare după stare cu componentă integratoare - Figura 4.9.

Începem prin a introduce o stare suplimentară $x_i \in R^p$ datorită componentei integratoare: $\dot{x}_i(t) = e(t) = r(t) - y(t) = r(t) - Cx(t)$. Modelul intra-stare-ieșire extins este ⁴⁶

$$\underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix}}_{\hat{x}_e(t)} = \underbrace{\begin{bmatrix} A & 0_{n \times p} \\ -C & 0_{p \times p} \end{bmatrix}}_{A_e} \underbrace{\begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}}_{x_e(t)} + \underbrace{\begin{bmatrix} B \\ 0_{p \times m} \end{bmatrix}}_{B_u} u(t) + \underbrace{\begin{bmatrix} 0_{n \times p} \\ I_{p \times p} \end{bmatrix}}_{B_r} r(t). \quad (4.25)$$

⁴⁶ $0_{p \times p}$ și $I_{p \times p}$ reprezintă matricea zero, respectiv matricea unitate de dimensiuni $p \times p$.

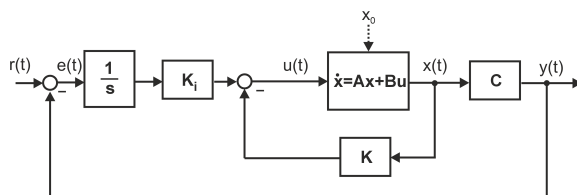


Figura 4.9: Reglare după stare cu integrator - urmărire

Adoptăm legea de reglare

$$u(t) = -Kx(t) + K_i x_i(t) = -[K \quad -K_i] \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} = -K_e \cdot x_e(t), \quad (4.26)$$

iar modelul în buclă închisă devine

$$\dot{x}_e(t) = (A_e - B_u K_e)x_e(t) + B_r r(t).$$

Deci problema se reduce la alocarea valorilor proprii ai matricei $A_e - B_u K_e$ prin proiectarea lui K_e .

Ca și comparație între cele două abordări, este important de observat că deși reglarea după stare cu precompensare este mai simplă de proiectat și nu implică adăugarea de noi stări la model, nu este robustă la incertitudini de modelare sau perturbații persistente. În schimb, reglarea după stare cu componentă integratoare este robustă la anumite categorii de incertitudini de modelare sau perturbații persistente.

Exercițiul 21. Pentru pendulul din Figura 4.1, modelat în cadrul Exercițiului 17, considerăm că putem măsura atât poziția unghiulară q , cât și viteza unghiulară \dot{q} . Considerăm valorile parametrilor $m = 0.4 \text{ kg}$, $l = 0.6 \text{ m}$, $g = 0.98 \text{ m/s}^2$, $b = 0.1 \text{ N s}^{-2}/\text{m}$. Să se proiecteze un regulator cu reacție după stare cu precompensare pentru urmărirea unei referințe de poziție de tip treaptă, cu amplitudini de $\pm 0.5 \text{ rad}$. Să se testeze soluția de control pe modelul neliniar în MATLAB/Simulink.

Soluție. Definim mărimile de stare $x_1 = q$ și $x_2 = \dot{q}$, mărimea de comandă $u = \tau$ și ieșirea de urmărit $y = q$. Matricile sistemului liniarizat, după înlocuirea valorilor numerice ale parametrilor sunt ⁴⁷

$$A = \begin{bmatrix} 0 & 1 \\ -16.33 & -0.41 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 6.94 \end{bmatrix}, C = [1 \quad 0]. \quad (4.27)$$

Polii procesului se pot determina calculând valorile proprii ale matricei $A: p_{1,2} = -0.20 \pm 4.03i$. Se poate verifica că sistemul respectă condiția

⁴⁷Prezentăm valorilor doar cu 2 zecimale, dar în calculul din MATLAB se lucrează cu o precizie de 5 zecimale.

de controlabilitate. Impunem poli doriti pentru sistemul în buclă închisă $p_{i1} = -10$ și $p_{i2} = -11$. Deci dorim să impunem polinomul caracteristic $\mu_i(s) = s^2 + 21s + 110$. Pentru legea de control $u = -k_1 x_1 - k_2 x_2 = -Kx$ polinomul caracteristic pentru sistemul în buclă închisă este $\mu_i(s) = s^2 + (\frac{5}{12} + \frac{125}{18}k_2)s + (\frac{49}{3} + \frac{125}{18}k_1)$. Egalând cele două polinoame și rezolvând sistemul cu două ecuații și două necunoscute rezultă $k_1 = 13.48$ și $k_2 = 2.96$. Alternativ, calculul se putea face apelând funcția *place* în MATLAB. Mai departe extindem legea de control la problema de urmărire $u = -Kx + Nr$, unde r este referința de poziție unghiulară. Din calcul, folosind relația (4.24), rezultă $N = 15.84$. Rezultatele de simulare, pornind de la condiții inițiale nule, sunt prezentate în Figura 4.10. Se poate observa că referința r urmărește ieșirea y cu eroarea staționară zero și fără suprareglaj. Timpul de reglare se poate îmbunătăți prin impunerea unor poli mai rapizi (valori reale negative mai departe de axa imaginară), dar asta ar fi cu prețul unor comenzi mai mari (a se vedea evoluția comenzii din partea de jos a Figurii 4.10). □

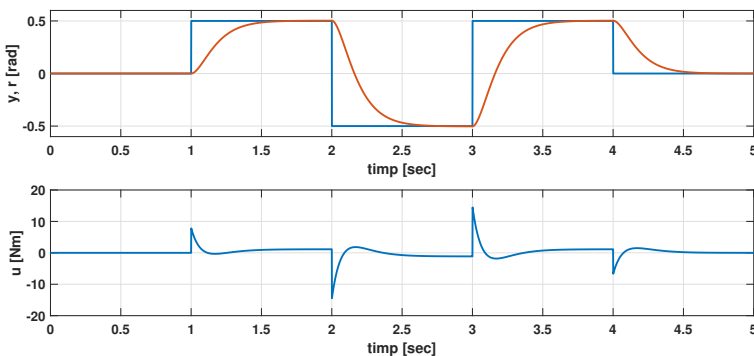


Figura 4.10: Rezultate de simulare pentru controlul poziției unui pendul

Exercițiul 22. Considerăm robotul din Figura 3.1, liniarizat în cadrul Exercițiului 18. Să se proiecteze și testeze o structură de conducere pentru problema de urmărire parcurgând următorii pași:

- Să se proiecteze un regulator cu reacție după stare cu precompensare pentru urmărire.
- Să se testeze structura de conducere prin simulări cu modelul neliniar al procesului pentru semnale de referință treaptă.
- Să se interpreteze rezultatele în termeni de suprareglaj, timp de reglare și eroare staționară.

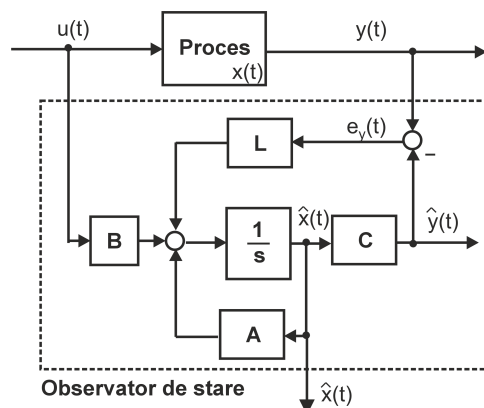


Figura 4.11: Observator de stare de tip Luenberger

4.3.3 Estimarea stărilor

De multe ori în practică nu pot fi măsurate toate stările sistemului, sau senzorii necesari pentru a măsura toate stările ar presupune un cost prea mare ⁴⁸. Deși ipoteza ca toate stările să fie măsurabile nu este respectată, o alternativă constă în proiectarea unor structuri adiționale cu rol în estimarea stărilor nemăsurabile \hat{x} (astfel încât $\hat{x} \xrightarrow{t \rightarrow \infty} x$). O astfel de structură se numește **observator (estimator) de stare**.

Un sistem de forma (4.20) este observabil, adică stările pot fi estimate din informațiile intrare-ieșire, dacă matricea de observabilitate

$$M_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.28)$$

are rangul egal cu ordinul sistemului ($\text{rang}(M_o) = n$).

Pentru sisteme liniare deterministice ⁴⁹, cel mai des întâlnit tip de observator de stare este observatorul de tip Luenberger, format dintr-o copie a modelului procesului și o reacție de corecție - Figura 4.11. Structura de tip observator de stare pornește de la o copie a modelului procesului

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) \\ \hat{y}(t) &= C\hat{x}(t). \end{aligned}$$

Deoarece în practică noi nu cunoaștem exact condițiile inițiale ale procesului $x(0) = x_0$, rezultă că $x_0 \neq \hat{x}_0$. Pentru ca $\hat{x}(t) \rightarrow x(t)$ pe măsura

⁴⁸În unele aplicații aparte problema de estimare este și mai critică: e.g. sistemele biomedicale, unde este de dorit ca senzorii să nu fie de natura invazivă.

⁴⁹Varianta pentru sisteme stohastice este dată de estimatorul (filtrul) Kalman.

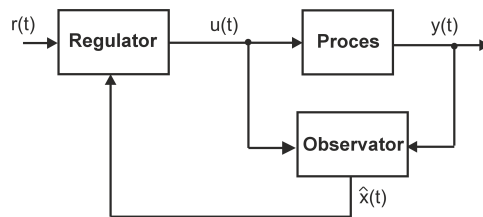


Figura 4.12: Structura de conducere cu regulator cu reacție după stare și observator de stare

ce $t \rightarrow \infty$, trebuie să adăugăm o reacție de corecție. Ca urmare, ecuațiile observatorului sunt

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad (4.29)$$

$$\hat{y}(t) = C\hat{x}(t), \quad (4.30)$$

unde $L = [l_1 \ l_2 \ \dots \ l_n]^T$ este amplificarea de corecție a observatorului. Dacă definim eroare de estimare ca $e_x(t) = x(t) - \hat{x}(t)$, se poate demonstra că $\dot{e}_x(t) = (A - LC)e_x(t)$. Sistemul care descrie dinamica erorii este stabil, adică $e_x(t) \rightarrow 0$ pe măsură ce $t \rightarrow \infty$, dacă toate valorile proprii ale matricei $A - LC$ sunt în semiplanul stâng. Deci problema proiectării lui L se poate reduce la una de alocare de poli - polii observatorului.

Ca regulă generală, observatorul trebuie să fie mai rapid decât sistemul în buclă închisă (regulator+proces). Altfel spus, dacă polii sistemului în buclă închisă impuși prin reacție după stare p_i trebuie să fie la stânga polilor procesului p , polii observatorului p_o trebuie să fie la stânga acestor poli impuși (p_i). Calculul amplificării L se poate face analitic, egalând polinomul caracteristic impus $\mu_{imp}^o(s) = (s - p_{o1})(s - p_{o2})\dots(s - p_{on})$ cu polinomul caracteristic în buclă închisă $\mu_{cl}^o(s) = \det(sI_n - (A - LC))$ și rezolvând sistemul algebric care rezultă ⁵⁰.

Odată proiectat, observatorul de stare se poate integra cu oricare din structurile cu reacție după stare descrise anterior, în care starea procesului x se înlocuiește cu starea estimată a procesului \hat{x} - Figura 4.12.

Exercițiul 23. Pentru pendulul din Figura 4.1, modelat în cadrul Exercițiului 17, considerăm că putem măsura doar poziția unghiulară q . Valorile parametrilor sunt aceleași ca și în Exercițiul 21. Să se proiecteze și testeze o structură de conducere cu regulator cu reacție după stare și observator de stare pentru problema de urmărire.

Soluție. Definim mărimile de stare $x_1 = q$ și $x_2 = \dot{q}$, mărimea de comandă $u = \tau$ și ieșirea de urmărit $y = q$. Matricile sistemului liniarizat

⁵⁰ L se poate calcula și numeric folosind aceeași funcție *place* în MATLAB.

sunt date în (4.27). Se poate verifica că sistemul este observabil, deci putem estima starea x_2 . Deoarece dorim să testăm structura de conducere pe sistemul neliniar, pentru care model liniar folosit pentru proiectarea regulatorului și a observatorului este doar o aproximare, apelăm la structura cu reglare după stare și integrator. Matricile sistemului extins (4.25) cu starea x_i sunt

$$A_e = \begin{bmatrix} 0 & 1 & 0 \\ -16.33 & -0.41 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ 6.94 \\ 0 \end{bmatrix}, \quad B_r = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Impunem polii pentru sistemul extins în buclă închisă, cu legea de reglare (4.26): $p_{i1} = -10$, $p_{i2} = -11$, $p_{i3} = -3$. Apelând la funcția *place* în MATLAB obținem $K = [22.56 \ 3.39 \ -47.52]$. De asemenea impunem polii observatorului $p_{o1} = -15$ și $p_{o2} = -16$, pentru care rezultă amplificarea $L = [30.58 \ 210.92]^T$. Implementând structura din Figura 4.12 în MATLAB/Simulink, pentru referințe treaptă de amplitudini $\pm 0.5 \text{ rad}$, obținem rezultatele de simulare din Figura 4.13. Se poate observa că avem eroare staționară zero și cu suprareglaj zero, în condiții de incertitudini de modelare. Timpul de reglare este însă ușor mai mare față de varianta cu precompensare și fără observator de stare.

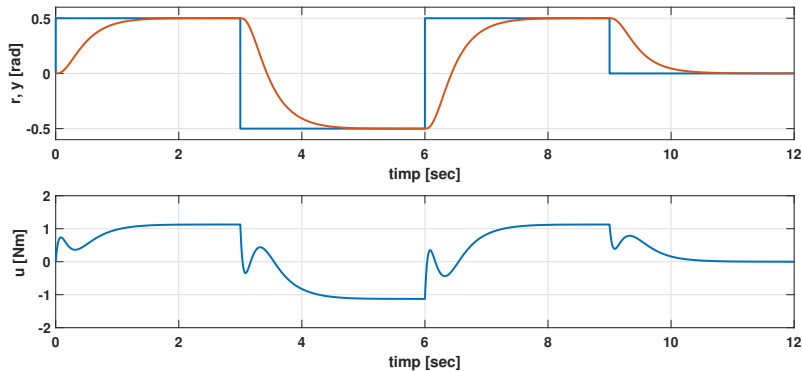


Figura 4.13: Rezultate de simulare pentru structura de conducere cu regulator cu reacție după stare cu integrator și observator de stare

□

Capitolul 5

Generarea traiectoriei

În acest capitol ne vom ocupa de planificarea (determinarea) traiectoriei de mișcare a roboților (*trajectory planning*). Prin traiectorie înțelegem o succesiune de poziții determinate la anumite momente de timp. De obicei succesiunea de puncte este furnizată de utilizator (în cel mai simplu caz un punct inițial și unul final) sau de către un planificator al traseului ⁵¹.

Traietoria este calculată de obicei prin interpolare ca o funcție netedă de timp $r(t)$, uneori și cu derivatele de ordinul 1 și 2 netede (vitezele și accelerațiile), și reprezintă referința transmisă structurii de conducere a robotului (Figura 5.1). În funcție de spațiul în care se face interpolarea, putem vorbi de o determinare a **traiectoriei în spațiul cuplelor** sau în **spațiul cartezian**. Următoarele două subcapitole vor trata în detaliu ambele cazuri.

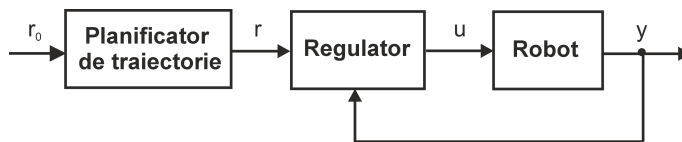


Figura 5.1: Structura de conducere cu planificator de traiectorie pentru referință

5.1 Traietorii în spațiul cuplelor

Considerăm situația în care avem o succesiune de poziții în spațiul cartezian $\{p_i\}$ (poziții ale efectorului final al robotului raportate la sistemul de referință), din care obținem o succesiune de poziții în spațiul cuplelor $\{q_i\}$

⁵¹La nivel planificării traseului (*path planning*) se ține cont de restricții în spațiul de mișcare, ocolire obstacole, etc.

folosind modelul geometric invers al robotului. Ca ipoteză simplificatoare, considerăm că timpul necesar pentru parcurgerea unui segment $\{i, i + 1\}$ este același pentru toate cuplele.

Vom prezenta inițial cazul când avem doar două poziții - poziția inițială și poziția finală, urmând ca să extindem ulterior discuția la mai multe puncte. Fără a reduce generalitatea, restrângem discuția pentru cazul unei singure cuple; traiectoria pentru restul cuplelor se poate determina într-o manieră similară.

5.1.1 Traiectoria între două puncte în spațiul cuplelor

Considerăm poziția inițială $q(t_0) = q_0$ și poziția finală $q(t_f) = q_f$, unde $t_f - t_0$ ne dă intervalul de timp în care trebuie să aibă loc mișcarea. Cea mai simplă abordare constă în *interpolarea liniară*, pe baza ecuației dreptei ce unește cele două puncte:

$$q(t) = q(t_0) + \frac{q_f - q_0}{t_f - t_0}(t - t_0) = a_0 + a_1 t. \quad (5.1)$$

Mișcarea în acest caz ar avea loc cu viteză constantă, dar problema este că nu avem control asupra vitezelor și accelerațiilor de pornire și oprire.

Să presupunem că vrem să specificăm și viteza inițială $v(t_0) = v_0$ și viteza finală $v(t_f) = v_f$ ⁵². Cele 4 restricții (2 de poziții și 2 de viteze) ne impun folosirea unui *polinom de ordinul cel puțin 3 pentru interpolare*, care are 4 parametri independenți:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3. \quad (5.2)$$

Prin derivare putem obține funcția pentru viteza dorită ca

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2. \quad (5.3)$$

Aplicând cele 4 restricții de poziție și viteză, obținem un sistem de 4 ecuații

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \quad (5.4)$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (5.5)$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 \quad (5.6)$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2. \quad (5.7)$$

Sub forma matriceală sistemul se scrie ca

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix}. \quad (5.8)$$

⁵²Abordarea se bazează pe [31] - cap 5.

Parametrii polinomului $a_0 - a_3$ se pot determina ca soluție unică a sistemului de mai sus.

Dacă pe lângă specificațiile de poziții și viteze mai vrem să adăugăm și o accelerație inițială $\alpha(t_0) = \alpha_0$ și finală $\alpha(t_f) = \alpha_f$, atunci trebuie să efectuăm *interpolare folosind un polinom de ordinul 5*:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5, \quad (5.9)$$

unde viteza dorită este

$$\dot{q}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4, \quad (5.10)$$

iar accelerația dorită

$$\ddot{q}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3. \quad (5.11)$$

Aplicând cele 6 restricții de poziție, viteză și accelerație obținem sistemul de ecuații

$$q_0 = a_0 + a_1t_0 + a_2t_0^2 + a_3t_0^3 + a_4t_0^4 + a_5t_0^5 \quad (5.12)$$

$$q_f = a_f + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \quad (5.13)$$

$$v_0 = a_1 + 2a_2t_0 + 3a_3t_0^2 + 4a_4t_0^3 + 5a_5t_0^4 \quad (5.14)$$

$$v_f = a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4 \quad (5.15)$$

$$\alpha_0 = 2a_2 + 6a_3t_0 + 12a_4t_0^2 + 20a_5t_0^3 \quad (5.16)$$

$$\alpha_f = 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3, \quad (5.17)$$

care sub forma matriceală devine

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ \alpha_0 \\ q_f \\ v_f \\ \alpha_f \end{bmatrix}, \quad (5.18)$$

iar parametrii $a_0 - a_5$ se obțin ca soluție unică.

5.1.2 Traectoria prin mai multe puncte în spațiul cuplelor

Abordarea din subsecțiunea anterioară se poate extinde pentru mai multe puncte. Să considerăm de exemplu cazul în care avem 3 puncte la 3 momente de timp: $q(t_0) = q_0$, $q(t_1) = q_1$ și $q(t_f) = q_f$. Vom considera cazul când

putem specifica vitezele dorite în fiecare din cele 3 puncte ⁵³: $v(t_0) = v_0$, $v(t_1) = v_1$ și $v(t_f) = v_f$.

Mai departe vom adopta două polinoame de ordinul 3, câte unul pentru fiecare segment de timp. Astfel definim primul polinom

$$q(t) = f_1(t) = a_{10} + a_{11}(t - t_0) + a_{12}(t - t_0)^2 + a_{13}(t - t_0)^3 \quad (5.19)$$

pentru intervalul de timp $t \in [t_0, t_1]$, iar al doilea polinom

$$q(t) = f_2(t) = a_{20} + a_{21}(t - t_1) + a_{22}(t - t_1)^2 + a_{23}(t - t_1)^3 \quad (5.20)$$

pentru intervalul de timp $t \in [t_1, t_f]$. Pe baza celor 6 restricții de poziție și viteză aplicate la nivelul celor două polinoame putem determina un sistem cu 8 ecuații și 8 necunoscute, din care rezultă coeficienții a_{ij} ($i = \overline{1, 2}, j = \overline{0, 3}$) ai polinoamelor ca soluție unică. Pentru a asigura continuitatea vitezei și accelerației la comutarea între polinoame vom adăuga condițiile ⁵⁴

$$\begin{aligned} q(t_1) &= f_1(t_1) = f_2(t_1) = q_1, \\ \dot{q}(t_1) &= \dot{f}_1(t_1) = \dot{f}_2(t_1) = v_1, \\ \ddot{q}(t_1) &= \ddot{f}_1(t_1) = \ddot{f}_2(t_1) = 0. \end{aligned} \quad (5.21)$$

Exercițiul 24. Vom considera robotul cu 5 cuple de rotație din Figura 5.2. Fiecare variabilă a cuplei reprezintă un unghi $q_i = \theta_i$, $i = \{1, 2, 3, 4, 5\}$. În ceea ce urmează, presupunem că modelul geometric direct și invers au fost determinate a priori folosind Robotics Toolbox. Parametrii robotului sunt: $L_0 = 0.06$ m, $L_1 = 0.03$ m, $L_2 = 0.15$ m, $L_3 = 0.185$ m, $L_4 = 0.1$ m. Să se determine traiectoria în spațiul cuplelor între două poziții, pentru o mișcare doar la nivelul cuplei q_2 a robotului: $q_2(t_0) = -10^\circ$ și $q_2(t_f) = 20^\circ$, la momentele de timp $t_0 = 0$ și $t_f = 1$ sec. Suplimentar, să se adopte viteze zero la cele două poziții $v_2(t_0) = v_2(t_f) = 0$.

Soluție. Alegem un polinom de ordinul 3 de forma (5.2) pentru descrierea mișcării, în care coeficienții îi obținem ca soluție a ecuației:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -10 \\ 0 \\ 20 \\ 0 \end{bmatrix}. \quad (5.22)$$

Rezultă că traiectoria cuplei 2 între cele două puncte poate fi descrisă prin funcția

$$q_2(t) = -10 + 90t^2 - 60t^3, \quad (5.23)$$

⁵³A se vedea [8] - pp. 207 pentru detalii legate de modul în care se pot determina vitezele prin puncte în practică.

⁵⁴Considerăm accelerația zero la momentul comutării.

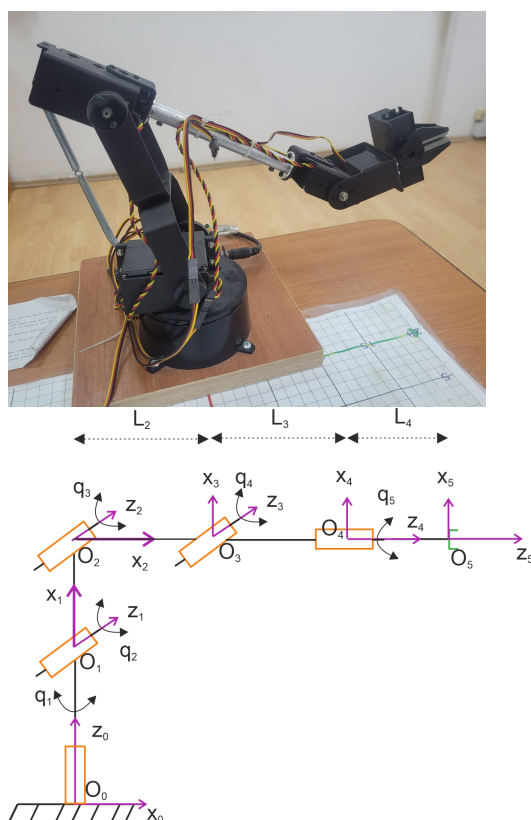


Figura 5.2: Robotul Lynxmotion cu 5 cuple de rotație și reprezentarea schematică

viteza este descrisă de expresia

$$\dot{q}_2(t) = 180t - 180t^2, \quad (5.24)$$

iar accelerația

$$\ddot{q}_2(t) = 180 - 360t. \quad (5.25)$$

Reprezentarea grafică a celor 3 traiectorii este ilustrată în Figura 5.3. Limitările unei astfel de traiectorii în practică, atunci când avem mai multe poziții succesive, este faptul că accelerația ar prezenta discontinuități (salturi), ceea ce ar afecta mișcarea robotului.

□

Exercițiul 25. Considerăm mișcarea robotului (Figura 5.2) din punctul $p(t_0) = [x_0 \ y_0 \ z_0]^T = [0.25 \ 0.1 \ 0.3]^T m$ în punctul $p(t_f) = [x_f \ y_f \ z_f]^T = [0.15 \ 0.2 \ 0.4]^T m$. Pentru simplitate, adoptăm momentele de timp $t_0 = 0$ și $t_f = 1$ sec. Cerințe în MATLAB:

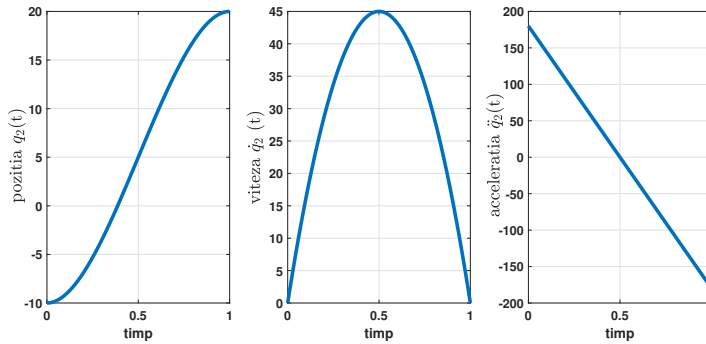


Figura 5.3: Traiectoriile pentru poziție, viteză și accelerație pentru cupla q_2

- Să se determine poziția inițială q_0 și finală q_f în spațiul cuplelor folosind modelul geometric invers.
- Să se determine traiectoria $q(t)$ între punctele q_0 și q_f folosind un polinom de ordinul 5, considerând viteze și accelerații zero la cele două capete $\dot{q}(t_0) = \dot{q}(t_f) = \ddot{q}(t_0) = \ddot{q}(t_f) = 0$.
- Să se afișeze traiectoriile atât în spațiul cuplelor cât și în spațiul cartezian.
- Să se compare rezultatele cu cele obținute cu funcția `traj` din *Robotics Toolbox* [6].

5.2 Traectorii în spațiul cartezian

Construirea traiectoriei în spațiul cuplelor este relativ ușor din punct de vedere al calculului numeric și nu prezintă probleme de singularitate, dar ne conduce în final la o mișcare a efectorului final complicată⁵⁵ și greu de anticipat. Din acest motiv în practică se preferă de multe ori construirea traiectoriei în spațiul cartezian. Acest lucru este mai solicitant din punct de vedere computațional deoarece modelul geometric invers trebuie aplicat pentru întreaga traiectorie generată (nu doar pentru setul de poziții inițiale ca și în cazul traiectoriei în spațiul cuplelor).

Vom considera cazul când traiectoria poate fi construită din linii drepte între pozițiile specificate inițial în spațiul cartezian⁵⁶. Adoptăm un profil trapezoidal al vitezei, care crește și scade ca o rampă în apropierea pozițiilor specificate inițial, iar între poziții este constantă. Restrângem discuția la

⁵⁵De obicei nu este o linie dreaptă între puncte.

⁵⁶Este cea mai des întâlnită abordare în practică atunci când e vorba de o traiectorie în spațiul cartezian.

doar două poziții pe axa X a efectorului final. Abordarea este similară și pentru celelalte două axe Y și Z, și se poate extinde ulterior pentru mai multe puncte⁵⁷.

Începem prin a defini cele două puncte între care vrem să fie mișcarea pe axa X: $x(t_0) = x_0$ și $x(t_f) = x_f$. Pentru simplitate, vom considera viteze zero în cele două puncte $\dot{x}(t_0) = \dot{x}(t_f) = 0$. Mai departe vom împărți intervalul de timp $t_0 - t_f$ în 3 segmente: (t_0, t_b) , $(t_b, t_f - t_b)$ și $(t_f - t_b, t_f)$.

Pentru intervalele de timp (t_0, t_b) și $(t_f - t_b, t_f)$ vom adopta funcții polinomiale de ordinul 2 pentru traiectorie

$$x(t) = \underline{a}_0 + \underline{a}_1(t - t_0) + \underline{a}_2(t - t_0)^2, \quad t \in [t_0, t_b], \quad (5.26)$$

$$x(t) = \bar{a}_0 + \bar{a}_1(t - (t_f - t_b)) + \bar{a}_2(t - (t_f - t_b))^2, \quad t \in [t_f - t_b, t_f]. \quad (5.27)$$

Vitezele rezultante pentru aceste două intervale de timp sunt

$$\dot{x}(t) = \underline{a}_1 + 2\underline{a}_2(t - t_0), \quad t \in [t_0, t_b], \quad (5.28)$$

$$\dot{x}(t) = \bar{a}_1 + 2\bar{a}_2(t - (t_f - t_b)), \quad t \in [t_f - t_b, t_f], \quad (5.29)$$

iar accelerațiile

$$\ddot{x}(t) = 2\underline{a}_2, \quad t \in [t_0, t_b], \quad (5.30)$$

$$\ddot{x}(t) = 2\bar{a}_2, \quad t \in [t_f - t_b, t_f]. \quad (5.31)$$

Pentru a determina coeficienții polinomului (5.26), aplicăm restricția $\dot{x}(t_0) = 0$ în (5.28), de unde rezultă că $\underline{a}_1 = 0$. Din condiția $x(t_0) = x_0$ aplicată în (5.26) rezultă că $\underline{a}_0 = x_0$. La momentul t_b impunem o viteză constantă v_b în (5.28), de unde rezultă $v_b = \dot{x}(t_b) = 2\underline{a}_2(t_b - t_0)$, deci $\underline{a}_2 = \frac{v_b}{2(t_b - t_0)}$.

Într-o manieră similară se pot determina și coeficienții polinomului (5.27): $\bar{a}_2 = -\frac{v_b}{2t_b}$, $\bar{a}_1 = v_b$, $\bar{a}_0 = x_f - \frac{v_b t_b}{2}$.

Pentru intervalul din mijloc $(t_b, t_f - t_b)$ vom defini traiectoria ca o linie dreaptă

$$x(t) = x(t_b) + v_b(t - t_b), \quad t \in [t_b, t_f - t_b]. \quad (5.32)$$

Din considerente de simetrie avem $x(\frac{t_0+t_f}{2}) = \frac{x_0+x_f}{2}$. Deci putem scrie

$$\frac{x_0 + x_f}{2} = x(t_b) + v_b(\frac{t_f + t_0}{2} - t_b).$$

Rezultă că $x(t_b) = \frac{x_0+x_f}{2} - v_b(\frac{t_f+t_0}{2} - t_b)$. Mai departe impunem condiția ca la momentul de timp t_b polinomul (5.26) și dreapta (5.32) să aibă aceeași valoare:

$$x_0 + \frac{v_b(t_b - t_0)}{2} = \frac{x_0 + x_f}{2} - v_b(\frac{t_f + t_0}{2} - t_b), \quad (5.33)$$

⁵⁷Dacă dorim să ținem cont și de orientare, atunci pe lângă cele 3 poziții $\{x, y, z\}$ mai trebuie să adăugăm 3 unghiuri de rotație în jurul acestor axe - a se vedea [8] - pp. 217 pentru detalii.

de unde rezultă că $t_b = \frac{x_0 - x_f + v_b t_f}{v_b}$. Dacă ținem cont de definiția momentului de comutație t_b că $t_0 < t_b < \frac{t_f + t_0}{2}$ atunci putem calcula restricția pentru viteza v_b astfel încât mișcarea să fie fezabilă:

$$\frac{x_f - x_0}{t_f - t_0} < v_b < \frac{x_f - x_0}{\frac{t_f + t_0}{2} - t_0}, \quad \text{dacă } v_b > 0, \quad (5.34)$$

$$\frac{x_f - x_0}{\frac{t_f + t_0}{2} - t_0} < v_b < \frac{x_f - x_0}{t_f - t_0}, \quad \text{dacă } v_b < 0. \quad (5.35)$$

În final obținem traiectoria între cele două poziții ca o funcție definită pe 3 intervale:

$$x(t) = \begin{cases} \underline{a}_0 + \underline{a}_1(t - t_0) + \underline{a}_2(t - t_0)^2, & t \in [t_0, t_b), \\ x(t_b) + v_b(t - t_b), & t \in [t_b, t_f - t_b), \\ \bar{a}_0 + \bar{a}_1(t - (t_f - t_b)) + \bar{a}_2(t - (t_f - t_b))^2, & t \in [t_f - t_b, t_f]. \end{cases} \quad (5.36)$$

Este important de avut în vedere că în practică traiectoria generată în spațiul cartezian poate întâmpina 3 tipuri de probleme:

- anumite puncte intermediare nu pot fi atinse de robot (probleme de singularitate),
- în urma aplicării modelului geometric invers pot rezulta viteze foarte mari la nivelul anumitor cuple,
- la punctul final se poate ajunge pe mai multe căi - soluții multiple.

Exercițiul 26. *Considerăm cazul când vrem să generăm o traiectorie între două puncte, de-a lungul axei X a efectorului final al unui robot (raportată la sistemul de referință fix). Poziția inițială este $x(t_0) = x_0 = 1m$, iar cea finală $x(t_f) = x_f = 0.5m$. Ca momente de timp, pentru simplitate alegem $t_0 = 0$ și $t_f = 1$ sec. Să se determine o traiectorie $x(t)$ între cele două puncte x_0 și x_f , în care viteza să aibă profilul trapezoidal, folosind ecuația (5.36).*

Soluție. Începem prin a alege viteza v_b la mijlocul intervalului dat de restricția (5.35), deci $v_b = -0.75m/sec$. De aici putem calcula momentul de comutație de la segmentul 1 la segmentul 2: $t_b = 0.3$. Se pot calcula mai departe coeficienții din ecuația (5.36): $\underline{a}_0 = 1$, $\underline{a}_1 = 0$, $\underline{a}_2 = -1.25$, $\bar{a}_0 = 0.6125$, $\bar{a}_1 = -0.75$, $\bar{a}_2 = 1.25$. Figura 5.4 ilustrează grafic traiectoria $x(t)$ obținută, împreună cu profilul vitezei și accelerației. Se poate observa că traiectoria poziției este netedă, viteza are profilul trapezoidal, iar accelerația este constantă pe porțiuni. \square

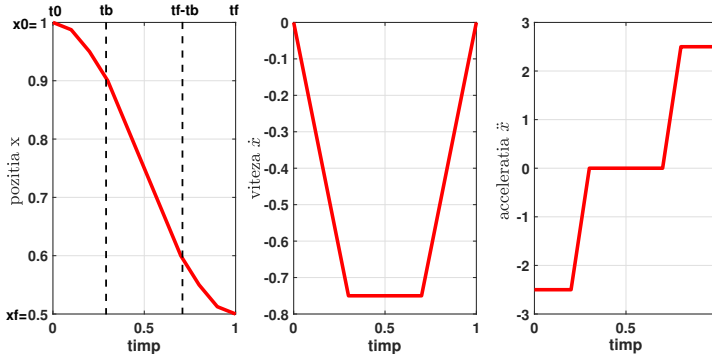


Figura 5.4: Traiectoriile pentru poziția, viteză și accelerația pentru axa X

Exercițiul 27. Considerăm mișcarea robotului (Figura 5.2) din punctul $p(t_0) = [x_0 \ y_0 \ z_0]^T = [0.25 \ 0.1 \ 0.3]^T m$ în punctul $p(t_f) = [x_f \ y_f \ z_f]^T = [0.15 \ 0.2 \ 0.4]^T m$. Pentru simplitate, adoptăm momentele de timp $t_0 = 0$ și $t_f = 1$ sec. Cerințe în MATLAB:

- Să se determine traiectoria $p(t) = [x(t) \ y(t) \ z(t)]^T$ între punctele $p(t_0)$ și $p(t_f)$ pentru un profil trapezoidal al vitezei.
- Să se determine traiectoria echivalentă în spațiul cuplelor $q(t)$ folosind modelul geometric invers.
- Să se afișeze traiectoriile atât în spațiul cuplelor cât și în spațiul cartezian.
- Să se compare rezultatele cu cele obținute cu funcția `ctrj` din Robotics Toolbox [6].

Capitolul 6

Controlul neliniar

Strategiile de control liniar din Capitolul 4 funcționează pe baza ipotezei că robotul operează în vecinătatea punctului de echilibru în care s-a efectuat liniarizarea. Pentru creșterea performanțelor în cazul mișcărilor ample și rapide ale robotului, caz în care această ipoteză nu se mai respectă, este necesar să ținem cont de modelul neliniar al robotului în proiectarea regulatorului. Așadar, pentru a obține garanții globale de stabilitate, de multe ori trebuie să proiectăm o structură de conducere neliniară. În acest capitol vom discuta despre cea mai des utilizată metodă de control neliniar pentru roboți - controlul neliniar decuplat sau metoda cuplului calculat ⁵⁸ ([31], [13], [1]).

Considerăm un braț robotic cu n grade de libertate care are modelul dinamic de forma

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau. \quad (6.1)$$

Neglijăm dinamica elementelor de execuție și considerăm că robotul poate fi comandat direct în cuplu. Astfel, procesul neliniar dat de brațul robotic are ca intrări cuplurile de comandă date de motoare τ_i , iar ca ieșiri măsurate unghiurile cuplelor q_i , cu $i = 1, \dots, n$.

Controlul neliniar decuplat este o strategie de control modernă care se poate folosi pentru creșterea performanțelor de reglare, fiind un caz particular al strategiei mai generale intitulată *feedback linearization* ([1],[31], [21]). Structura de control neliniar decuplat presupune o buclă interioară și una exterioară, precum în Figura 6.1. Bucla interioară se bazează pe modelul dinamic invers și realizează decuplarea neliniară, care mai este numită liniarizare dinamică. În cazul ideal, bucla interioară devine un sistem liniar invariant în timp de tip MIMO (multiple input multiple output) decuplat. Astfel, pentru bucla exterioară se poate alege orice metodă de control clasic (PID, reacție după stare, etc).

⁵⁸În engleză *Computer torque control*.

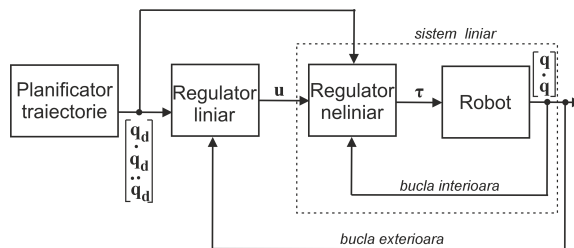


Figura 6.1: Control neliniar decuplat - structura de principiu

Pentru proiectarea buclei interioare, începem prin a introduce schimbarea de notație $V(q, \dot{q}) = C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q)$, astfel încât să putem rescrie modelul robotului într-o formă mai compactă:

$$D(q)\ddot{q} + V(q, \dot{q}) = \tau. \quad (6.2)$$

Mai departe, adoptăm următoarea lege de reglare pentru bucla interioară:

$$\tau = D(q)(\ddot{q}_d - u) + V(q, \dot{q}) \quad (6.3)$$

Dacă definim eroarea de urmărirea ca

$$e = q_d - q, \quad (6.4)$$

unde q_d este referința de poziție a cuplelor, iar q poziția cuplelor măsurată, atunci $\dot{e} = \dot{q}_d - \dot{q}$ și $\ddot{e} = \ddot{q}_d - \ddot{q}$. Înlocuind cuplul din (6.3) în modelul (6.2) și folosind definiția (6.4) putem obține:

$$\ddot{e} = u. \quad (6.5)$$

Astfel bucla internă poate fi privită ca un proces dublu integrator din perspectiva buclei externe. Acest model cu n dublu integratoare (un dublu integrator pentru fiecare canal intrare-ieșire/cuplă) poate fi scris în spațiul stărilor ca:

$$\dot{x} = Ax + Bu \quad (6.6)$$

unde

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, A = \begin{bmatrix} 0_n & I_n \\ 0_n & 0_n \end{bmatrix}, B = \begin{bmatrix} 0_n \\ I_n \end{bmatrix} \quad (6.7)$$

În continuare, pe baza acestui model, vom considera pentru bucla externă un regulator cu reacție după stare cu componentă integratoare:

$$u = -Kx + K_i e \quad (6.8)$$

unde ϵ este ieșirea integratorului regulatorului ($\dot{\epsilon} = e$) - a se vedea Figura 6.2⁵⁹. Sistemul nostru este controlabil, deci amplificarea K se poate obține prin alocare de poli, pornind de la polinomul caracteristic impus pentru sistemul în buclă închisă

$$\mu_{imp}(s) = s^{2n} + a_{2n-1}s^{2n-1} + a_{2n-2}s^{2n-2} + a_{2n-3}s^{2n-3} + \dots + a_1s + a_0 = (s - p_1)(s - p_2)\dots(s - p_{2n}). \quad (6.9)$$

Amplificarea integratorului K_i se poate determina prin alocare de poli pentru sistemul extins cu stările integratorului, sau se poate determina empiric din simulări/experimente astfel încât să se obțină performanțe de reglare cât mai bune, etc. În final, structura detaliată de conducere este ilustrată în Figura 6.2. A se observa ca metoda necesită viteza și accelerația referinței, ca urmare este nevoie de un planificator de traiectorie care să ne asigure că referințele au o traiectorie netedă în timp.

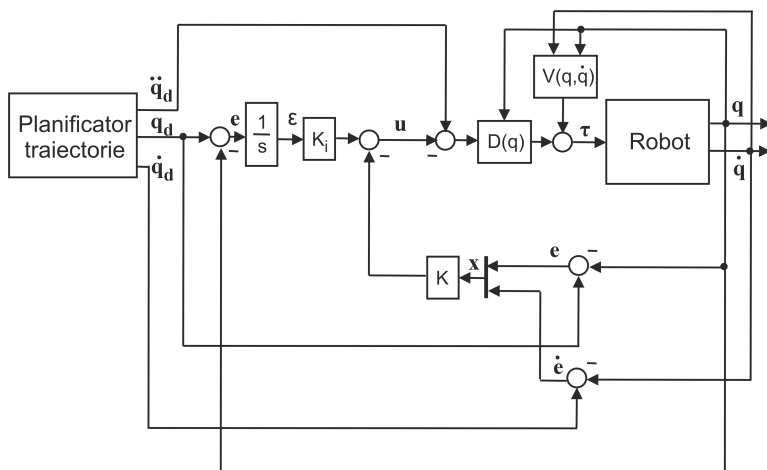


Figura 6.2: Control neliniar decuplat - structura detaliată

Metoda presupune un efort de calcul (numeric) semnificativ, iar performanțele sunt foarte dependente de acuratețea cu care cunoaștem modelul procesului. Cu cât diferențele dintre model și sistemul real sunt mai mari, performanțele se degradează semnificativ. În literatura de specialitate sunt numeroase extensii ale acestei metode astfel încât să asigure și robustețea. Multe abordări presupun modificarea legii de reglare (6.8) astfel încât să conțină un termen extra pentru robustețe ($u = -Kx + K_i\epsilon + u_{rob}$), dintre care cea mai cunoscută este extensia cu un regulator în regim modal alunecător (a se vedea [13]-Cap. 8.6.2 sau [1]-Cap. 5).

⁵⁹În cazul ideal de proces dublu integrator, componenta integratoare a regulatorului nu mai este necesară. În practică însă, datorită incertitudinilor de modelare, bucla internă nu se comportă ca un dublu integrator, iar astfel componenta integratoare a regulatorului poate aduce un plus de robustețe.

Exercițiul 28. Considerăm robotul din Figura 3.1, cu parametrii din Exercițiul 18. Considerăm că robotul poate fi controlat direct în cuplu, în domeniul $[-1.18, 1.18]$ Nm. Să se proiecteze o structură de control neliniară decuplată. Să se testeze structura de conducere prin simulări pentru semnale de referință sinusoidale. Să se interpreteze rezultatele.

Soluție. Considerăm pentru bucla interioară legea de reglare (6.3), iar pentru bucla exterioară (6.8) (vezi Figura 6.1). Proiectăm regulatorul cu reacție după stare K pentru procesul dublu integrator (6.5) cu $n = 2$ prin alocare de poli. Considerând polii impuși pentru sistemul în buclă închisă $[-4 -4 -9 -9]$ ⁶⁰, se poate obține prin calcul

$$K = \begin{bmatrix} 36 & 0 & 13 & 0 \\ 0 & 36 & 0 & 13 \end{bmatrix}.$$

Amplificarea integratorului K_i este determinată empiric în simulări, pornind de la o valoare inițială de 0.1. În final, obținem performanțe satisfăcătoare pentru $K_i = \text{diag}([0.6 \ 0.6])$.

Considerăm scenariul cu referință sinusoidală pentru ambele cuple: $q_{d1}(t) = q_{d2}(t) = \sin(t)$. Figura 6.3 ilustrează erorile de urmărire ($e_1 = q_{d1} - q_1$, $e_2 = q_{d2} - q_2$) pentru cele două cuple folosind controlul neliniar decuplat, comparativ cu controlul independent în cuple de tip PD cu gravitație. Deoarece referința presupune un regim permanent (nu se ajunge la regim staționar), din motive de cauzalitate eroarea de urmărire nu poate să fie niciodată zero. Se poate observa că eroare maximă pentru controlul neliniar decuplat este de aproximativ 0.01 rad (0.5 deg), iar pentru controlul independent în cuple eroarea maximă este mai mare - aproximativ 0.05 rad (2.8 grade). □

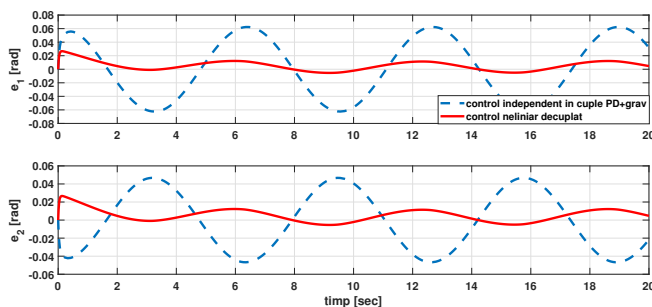


Figura 6.3: Comparație erori de urmărire: control neliniar decuplat vs. control independent în cuple

⁶⁰Poli au fost aleși astfel încât răspunsul sistemului să fie fără oscilații, suficient de rapid, dar evitând totodată saturația comenzii.

Capitolul 7

Control și estimare pentru roboți mobili

În capitolele anterioare ne-am referit predominant la roboți cu bază fixă complet acționați (numărul de grade de libertate este egal cu numărul de intrări de comandă). În acest capitol vom arăta că tehnicile de modelare și control pot fi folosite, cu unele adaptări, și pentru roboții mobili, care sunt de multe ori subacționați (numărul de grade de libertate este mai mare decât numărul de intrări de comandă). Pentru a nu complica expunerea foarte mult și din considerente de spațiu, ne vom limita la controlul liniar pe baza modelelor liniarizate prin metoda generală. Problema de estimare a stărilor devine mai critică în astfel de aplicații, deoarece aproape niciodată nu avem acces la toate stările prin măsurători.

Vom continua discuția cu ajutorul a două exemple foarte des întâlnite în aplicații (comerciale, educaționale, cercetare, etc.): un robot terestru cu roți de tip segway (Figura 7.1), respectiv un robot aerian (dronă) de tip quadcopter (Figura 7.2).

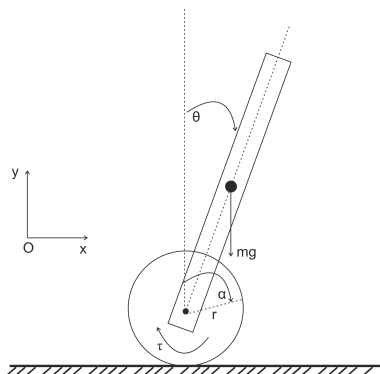


Figura 7.1: Robot pe roți de tip segway

Începem cu partea de **modelare matematică**. Pentru robotul mobil de tip segway schițat în Figura 7.1, definim unghiul de rotație în planul vertical θ și unghiul de rotație a roților α . Cuplul de comandă τ acționează la nivelul roților prin intermediul unor motoare de curent continuu ⁶¹. Robotul se deplasează în planul XOY , iar deplasarea pe axa X este dată de relația $x = r\alpha$ ⁶². Robotul este format din două părți mecanice, roțile și corpul propriu-zis, și poate fi considerat ca un pendul pe roți. Modelul dinamic al robotului se poate determina pe baza ecuațiilor Euler-Lagrange ([4], [2])⁶³

$$\begin{bmatrix} J_p & m_p l r \cos(\theta) \\ m_p l r \cos(\theta) & J_w + m_w r^2 + m_p r^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} -m_p g l \sin(\theta) \\ -m_p l r \dot{\theta}^2 \sin(\theta) \end{bmatrix} = \begin{bmatrix} -\tau \\ \tau \end{bmatrix}, \quad (7.1)$$

unde J_p și J_w reprezintă momentele de inerție ale corpului și ale roții, m_p și m_w se referă la masa corpului și a roții, l este distanța de la pivot la centrul de greutate, r este raza roții, iar g este accelerația gravitațională.

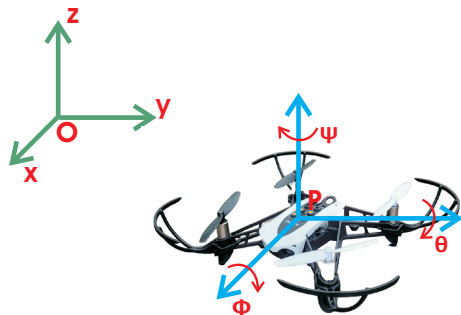


Figura 7.2: Drona Parrot Mambo

Pentru drona din Figura 7.2, definim mai întâi vectorul care conține poziția $p = [x \ y \ z]^T$ și orientarea $\eta = [\phi \ \theta \ \psi]^T$ (unghiurile Euler ϕ -roll, θ -pitch, ψ -yaw) în spațiul cartezian (sistemul de coordonate fix). Modelul

⁶¹Neglijăm dinamica motorului.

⁶²Ignorăm partea de orientare, în care robotul și-ar schimba direcția de mers prin rotirea celor două roți la viteze diferite.

⁶³Se neglijează forțele de frecare, și se consideră un contact ideal cu suprafața, astfel încât nu pot să aibă loc alunecări.

dinamic se poate determina folosind formalismul Euler-Lagrange ([19]): ⁶⁴

$$\left\{ \begin{array}{l} \ddot{x} = [c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)] \frac{U_{coll}}{m} \\ \ddot{y} = [c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)] \frac{U_{coll}}{m} \\ \ddot{z} = -g + c(\phi)c(\theta) \frac{U_{coll}}{m} \\ \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = J^{-1}(\eta) \left(\begin{bmatrix} U_{\phi} \\ U_{\theta} \\ U_{\psi} \end{bmatrix} - C(\eta, \dot{\eta})\dot{\eta} \right) \end{array} \right. \quad (7.2)$$

unde Jacobianul este dat de expresia

$$J(\eta) = \begin{bmatrix} I_x & 0 & -I_x s(\theta) \\ 0 & I_y c^2(\phi) + I_z s^2(\phi) & (I_y - I_z) c(\phi) s(\phi) c(\theta) \\ -I_x s(\theta) & (I_y - I_z) c(\phi) s(\phi) c(\theta) & I_x s^2(\theta) + I_y s^2(\phi) c^2(\theta) + I_z c^2(\phi) c^2(\theta) \end{bmatrix} \quad (7.3)$$

iar matricea de Coriolis este

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad (7.4)$$

cu termenii

$$\begin{aligned} c_{11} &= 0 \\ c_{12} &= (I_y - I_z)(\dot{\theta}c(\phi)s(\phi) + \dot{\psi}s^2(\phi)c(\theta)) + (I_z - I_y)\dot{\psi}c^2(\phi)c(\theta) - I_x\dot{\psi}c(\theta) \\ c_{13} &= (I_z - I_y)\dot{\psi}c^2(\theta)s(\phi)c(\phi) \\ c_{21} &= (I_z - I_y)(\dot{\theta}s(\phi)c(\phi) + \dot{\psi}s^2(\phi)c(\theta)) + (I_y - I_z)\dot{\psi}c^2(\phi)c(\theta) + I_x\dot{\psi}c(\theta) \\ c_{22} &= (I_z - I_y)\dot{\phi}c(\phi)s(\phi) \\ c_{23} &= -I_x\dot{\psi}s(\theta)c(\theta) + I_y\dot{\psi}s^2(\phi)s(\theta)c(\theta) + I_z\dot{\psi}c^2(\phi)s(\theta)c(\theta) \\ c_{31} &= (I_y - I_z)\dot{\psi}c^2(\theta)s(\phi)c(\phi) - I_x\dot{\theta}c(\theta) \\ c_{32} &= (I_z - I_y)(\dot{\theta}c(\phi)s(\phi)s(\theta) + \dot{\phi}s^2(\phi)c(\theta)) + (I_y - I_z)\dot{\phi}c^2(\phi)c(\theta) + \\ & \quad I_x\dot{\psi}s(\theta)c(\theta) - I_y\dot{\psi}s^2(\phi)s(\theta)c(\theta) - I_z\dot{\psi}c^2(\phi)s(\theta)c(\theta) \\ c_{33} &= (I_y - I_z)\dot{\phi}c^2(\theta)s(\phi)c(\phi) - I_y\dot{\theta}s^2(\phi)s(\theta)c(\theta) - I_z\dot{\theta}c^2(\phi)s(\theta)c(\theta) + \\ & \quad I_x\dot{\theta}s(\theta)c(\theta). \end{aligned}$$

⁶⁴ $s(\cdot)$ și $c(\cdot)$ sunt notații prescurtate pentru $\sin(\cdot)$ și $\cos(\cdot)$

Intrările de comandă sunt cuplurile pe cele trei axe de rotație U_ϕ, U_θ, U_ψ , și forța colectivă U_{coll} care influențează altitudinea ⁶⁵. Parametrii sunt momentele de inerție $\{I_x, I_y, I_z\}$, masa dronei m , accelerația gravitațională g .

Ambele modele pot fi rescrise sub forma generală neliniară în spațiul stărilor

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (7.5)$$

iar apoi liniarizate într-un punct de echilibru \mathbf{x}_e pentru o comandă de echilibru \mathbf{u}_e , astfel încât să ajungem la forma standard liniară în spațiul stărilor

$$\dot{\mathbf{x}}_l = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{(\mathbf{x}^e, \mathbf{u}^e)} \cdot \mathbf{x}_l + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}|_{(\mathbf{x}^e, \mathbf{u}^e)} \cdot \mathbf{u}_l = \mathbf{A}\mathbf{x}_l + \mathbf{B}\mathbf{u}_l, \quad (7.6)$$

unde $\mathbf{x}_l = \mathbf{x} - \mathbf{x}^e$ și $\mathbf{u}_l = \mathbf{u} - \mathbf{u}^e$.

Exercițiul 29. Considerăm drona din Figura 7.2, cu modelul dinamic (7.2) și parametrii: $m = 0.063 \text{ kg}$, $I_x = 0.5829 \cdot 10^{-4} \text{ kg m}^2$, $I_y = 0.7169 \cdot 10^{-4} \text{ kg m}^2$, $I_z = 1.000 \cdot 10^{-4} \text{ kg m}^2$, $g = 9.8 \text{ m/s}^2$. Să se determine modelul liniarizat.

Soluție. În regimul de funcționare de tip planare ⁶⁶, modelul poate fi mult simplificat considerând că pentru fiecare unghi α putem face aproximația $\sin(\alpha) = \alpha$ și $\cos(\alpha) = 1$, și de asemenea avem $U_{coll} = mg + \Delta U_{coll}$ [29]. Modelul simplificat devine:

$$\begin{cases} \ddot{x} = \theta g \\ \ddot{y} = -\phi g \\ \ddot{z} = \frac{\Delta U_{coll}}{m} \end{cases}, \quad \begin{cases} \ddot{\phi} = \frac{U_\phi}{I_x} \\ \ddot{\theta} = \frac{U_\theta}{I_y} \\ \ddot{\psi} = \frac{U_\psi}{I_z} \end{cases}. \quad (7.7)$$

Acestea sugerează că un control de tip liniar ar putea să fie suficient atunci când drona funcționează în regimuri apropiate de cel de planare.

Ca o abordare alternativă, putem rescrie modelul dronei în spațiul stărilor ca în (7.5), unde vectorul de stare este $\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ și vectorul de intrare $\mathbf{u} = [U_{coll}, U_\phi, U_\theta, U_\psi]^T$. Considerăm mai departe punctul de echilibru în regim de planare $\mathbf{x}^e = [x^e, y^e, z^e, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, cu intrările $\mathbf{u}^e = [U_{coll}^e, 0, 0, 0]^T$. Modelul liniarizat poate fi obținut ca (7.6),

⁶⁵Dacă se neglijează dinamica motoarelor, există o relație algebrică cunoscută între vitezele motoarelor și aceste cupluri și forțe [23],[26].

⁶⁶În engleză *hovering*.

iar expresiile matricelor A și B rezultate din calcul sunt

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix}.$$

□

Exercițiul 30. Considerăm robotul de tip segway din Figura 7.1, cu modelul (7.1), și parametri⁶⁷: $J_p = 0.16 \text{ Nm}$, $m_p = 2.94 \text{ kg}$, $l = 0.168 \text{ m}$, $m_w = 0.46 \text{ kg}$, $r = 0.0515 \text{ m}$, $J_w = 0.00045 \text{ Nm}$, $g = 9.81 \text{ m/s}^2$. Să se determine modelul liniarizat considerând ca vector de stare $\mathbf{x} = [\theta \ \alpha \ \dot{\theta} \ \dot{\alpha}]^T$ și intrarea $u = \tau$.

Înainte de a trece la partea de control, ne punem problema accesibilității stărilor sistemului. Pentru roboții mobili din Figurile 7.2 și 7.1, dar și în general la majoritatea roboților mobili, nu putem măsura toate stările. Ca urmare trebuie să proiectăm mai întâi un **observator de stare** pentru a estima stările nemăsurabile. Considerăm scenariul când roboții sunt folosiți în spații închise, unde avem posibilitatea de a măsura pozițiile liniare și unghiulare prin intermediul unui sistem de camere (e.g. OptiTrack [18])⁶⁸. Deci ne rămâne ca să estimăm vitezele printr-un observator de tip Luenberger pe baza modelului liniarizat:

$$\dot{\hat{\mathbf{x}}}_l = A\hat{\mathbf{x}}_l + B\mathbf{u}_l + L(\mathbf{y}_l - C_p\hat{\mathbf{x}}_l), \quad (7.8)$$

unde $\hat{\mathbf{x}}_l$ este starea estimată, $\mathbf{y}_l = \mathbf{y}_p - \mathbf{y}_p^e$, cu ieșirea măsurată definită ca $\mathbf{y}_p = C_p\mathbf{x}$.

⁶⁷Parametrii sunt luați din [4].

⁶⁸În spații deschise, poziția unei drone ar putea fi măsurată prin GPS, iar la robotul de tip segway poziția poate fi determinată din distanța parcursă de roți - pornind de la unghiul motoarelor (măsurat de exemplu printr-un traductor rotativ). Totuși problema de estimare este în general mai complicată, neavând de obicei acces la măsurători în termeni de orientare sau poziție unghiulară, decât la viteze sau accelerații prin intermediul unui accelerometru sau giroscop.

Odată ce avem toate stările accesibile din măsurători sau estimări, putem proiecta un regulator cu reacție după stare pentru stabilizare și urmărire. Considerăm următoarea **lege de reglare** pentru cei doi roboți mobili

$$\mathbf{u} = \mathbf{u}_{fb} + \mathbf{u}_{ff}, \quad (7.9)$$

formată dintr-o componentă cu reacție (feedback) după eroarea dintre stările estimate și referințele pentru stări

$$\mathbf{u}_{fb} = -K\mathbf{e}_x = -K(\hat{\mathbf{x}}_l - \mathbf{x}_r), \quad (7.10)$$

și o componentă în buclă deschisă (feedforward) care depinde de semnale exogene ca perturbații măsurabile \mathbf{d} sau referințe \mathbf{r} .

Exercițiul 31. Considerăm drona din Figura 7.2, cu modelul neliniar (7.2), și cu modelul liniarizat și parametrii din Exercițiul 29. Vectorul ieșirilor măsurabile este $\mathbf{y}_p = C_p\mathbf{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T$ cu $C_p = [I_6 \ 0_{6 \times 6}]$. Să se proiecteze un regulator de forma (7.9) împreună cu un observator de stare de forma (7.8) care să rezolve problema de stabilizare și urmărire a procesului. Să se testeze structura de conducere proiectată pe modelul neliniar în simulări, pentru referința $\mathbf{x}_r = [r_x \ r_y \ r_z \ \mathbf{0}_{1 \times 9}]^T$, cu $r_x = \sin(0.1t)$, $r_y = \cos(0.1t)$, $r_z = 2$. Ca și condiții inițiale, considerăm că drona este deja în regim de planare la o altitudine de 1m.

Soluție. Proiectăm mai întâi regulatorul cu reacție după stare din (7.9) prin alocare de poli, considerând stările direct accesibile (ignorăm observatorul), referința zero $\mathbf{x}_r = \mathbf{0}$, iar componenta $\mathbf{u}_{ff} = \mathbf{0}$ (adică pentru legea de reglare simplificată $\mathbf{u} = -K\mathbf{x}_l$). Impunem poli pentru sistemul liniar în buclă închisă $p_i = [-2 \ -3 \ -4 \ -5 \ -6 \ -7 \ -8 \ -8 \ -10 \ -11 \ -12 \ -13]$. Apelând funcția $place(A, B, p_i)$ obținem regulatorul cu reacție după stare K care ne stabilizează sistemul, adică ne aduce stările \mathbf{x}_l în punctul $\mathbf{0}$. Dacă regulatorul primește la intrare eroarea de urmărire \mathbf{e}_x (cazul în care referința \mathbf{x}_r este nenulă și $\mathbf{u} = -K\mathbf{e}_x$), va aduce această eroare la zero, rezolvând problema de urmărire⁶⁹. Fiindcă sistemul este neliniar, de multe ori este necesar în practică și o componentă de *feedforward*. Considerăm cazul simplu în care vrem să compensăm doar forța gravitațională necesară pentru regimul de planare, deci adoptăm $\mathbf{u}_{ff} = [mg \ 0 \ 0]^T$. Mai departe proiectăm observatorul de stare de forma (7.8) tot prin alocare de poli, impunând poli mai rapizi decât cei impuși de regulator. Adoptăm poli observatorului $p_o = [-500 \ -19 \ -28 \ -32 \ -400 \ -600 \ -10 \ -41 \ -35 \ -200 \ -7 \ -9]$, iar apelând funcția $place(A', C', p_o)'$ obținem amplificarea L a observatorului. Acuma putem testa (7.9) împreună cu observatorul în simulări.

⁶⁹Am presupus că referința variază lent în timp, sau este constantă pe porțiuni.

Scenariul de simulare presupune ca drona să parcurgă un cerc în planul XY de rază $1m$, iar la un moment dat să urce în altitudine de la $1m$ la $2m$. În implementare considerăm un pas de eșantionare de $1ms$. Deși modelul și simularea noastră nu surprinde partea de decolare, avem totuși un interval de inițializare, deoarece modelul este neliniar și avem nevoie de timp pentru convergența observatorului. Amplificările regulatorului sau ale observatorului nu pot să fie foarte mari, deoarece comanda poate să intre în saturație, sau pur și simplu anumite componente neliniare ale sistemului de frecvența mai mare pot destabiliza sistemul. Ca urmare, prin această structură de conducere liniară nu putem control drona decât pentru mișcări relativ lente. Mai departe, adoptăm următorul profil al referințelor pe cele trei axe, considerând o perioadă de inițializare de 5 sec :

$$\begin{aligned} r_x &= \{0 \text{ dacă } t \leq 5, \quad \sin(0.1(t - 5)) \text{ dacă } t > 5\} \\ r_y &= \{0 \text{ dacă } t \leq 5, \quad \cos(0.1(t - 5)) \text{ dacă } t > 5\} \\ r_z &= \{1 \text{ dacă } t \leq 20, \quad 2 \text{ dacă } t > 20\}. \end{aligned}$$

Figura 7.3 arată evoluția în timp a pozițiilor măsurate $\{x, y, z\}$ raportată la referințele $\{r_x, r_y, r_z\}$, iar Figura 7.4 arată urmărirea cercului în planul XY . Se poate observa perioada de inițializare de 5 secunde, referințele pe x și y se activează la momentul $t = 5\text{ sec}$, iar referința treaptă pe z la momentul $t = 20\text{ sec}$. Drona urmărește satisfăcător referințele, deși se poate observa în planul XY că cercul nu este parcurs perfect (apar mici erori de urmărire), deoarece traiectoria de referință se modifică în continuu (viteza referinței nu este niciodată zero, ca urmare nu avem niciodată regim staționar) și drona nu este suficient de rapidă. Aceste erori depind de viteza de parcurgere impusă de referință, dar pot fi făcute mai mici prin creșterea amplificărilor regulatorului.

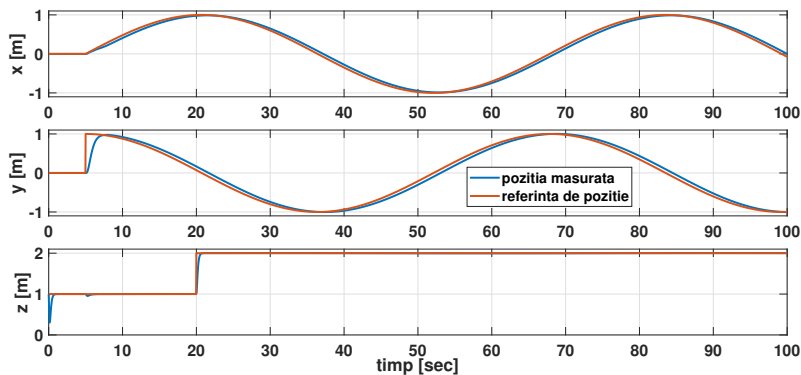


Figura 7.3: Control liniar cu observator pentru o dronă - urmărirea celor trei referințe de poziție

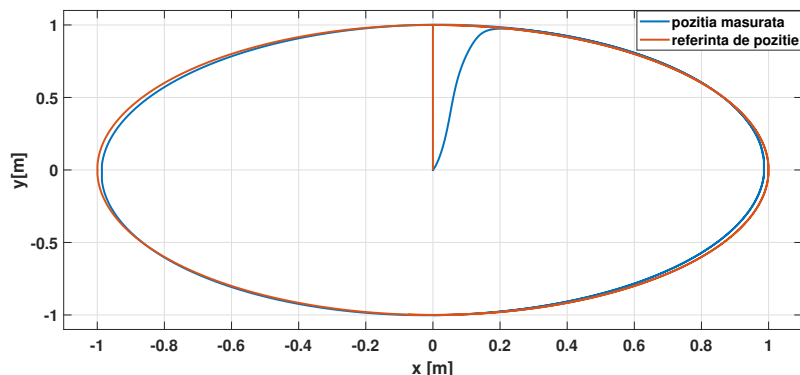


Figura 7.4: Control linear cu observator pentru o dronă - urmărirea referințelor în planul XY

□

Exercițiul 32. Considerăm robotul pe roți de tip segway din Figura 7.1, cu modelul nelinier (7.1), și cu modelul liniarizat și parametrii din Exercițiul 30. Vectorul ieșirilor măsurabile este $\mathbf{y}_p = C_p \mathbf{x} = [\theta \ \alpha]^T$ cu $C_p = [I_2 \ 0_2]$. Să se proiecteze un regulator de forma (7.9) împreună cu un observator de stare de forma (7.8) care să ne rezolve problema de stabilizare și urmărirea a procesului. Să se testeze structura de conducere proiectată pe modelul nelinier în simulări, pentru referința $\mathbf{x}_r = [r_\theta \ \mathbf{0}_{1 \times 3}]^T$, cu $r_\theta = \{0 \text{ dacă } t \in [0, 5), (t - 5) \text{ dacă } t \in [5, 25), -2(t - 25) + 20 \text{ dacă } t \in [25, 45), (t - 45) - 20 \text{ dacă } t \in [45, 65), 0 \text{ dacă } t \in [65, 80)\}$ (timpul este în secunde). Considerăm condiții inițiale zero pentru robot, adică robotul este în repaus în poziție verticală.

Structura de conducere discutată mai sus are la bază proiectarea unui regulator cu reacție după stare prin alocare de poli, în care performanțele sunt impuse de modul în care sunt aleși polii. O abordare alternativă foarte populară este proiectarea unui regulator optimal de tip LQR⁷⁰ [15], în care amplificarea K este obținută prin minimizarea unei funcții obiectiv de forma

$$J = \int_0^\infty (\mathbf{x}_l^T Q \mathbf{x}_l + \mathbf{u}_l^T R \mathbf{u}_l) dt, \quad (7.11)$$

unde Q și R sunt matrici diagonale, reprezentând ponderile pentru stări și pentru mărimile de comandă. Principialul avantaj pentru sisteme de tip MIMO este că prin aceste ponderi putem influența direct anumite stări individuale sau limita anumitor mărimi de comandă. De asemenea, componenta în buclă deschisă \mathbf{u}_{ff} poate fi dezvoltată astfel încât să compenseze și anumite forțe de frecare, sau alte perturbații măsurabile.

⁷⁰Linear quadratic regulator.

Extensia la un control de tip neliniar pentru roboții mobili de tipul celor prezentați în acest capitol nu este trivială, deoarece avem de a face cu sisteme subacționate. Metoda de decuplare neliniară prezentată în Capitolul 6 se poate aplica doar la sisteme complet acționate. Totuși, există extensii ale acestei metode și pentru sisteme subacționate - a se vedea *partial feedback linearization* [5]. Pentru partea de estimare, extensii neliniare sunt observatorul Luenberger neliniar (a se vedea Capitolul Observers în [14]) sau filtrul Kalman extins [15], care ambele se folosesc de modelul neliniar al sistemului. Avantajul unei abordări neliniare este că permite obținerea unor performanțe superioare chiar și pentru mișcări ample și rapide ale roboților, oferind garanții globale sau cvasi-globale de stabilitate. Prețul este complexitatea crescută a proiectării și depanării, dar și necesitatea unui sistem hardware cu putere de calcul semnificativ sporită.

Bibliografie

- [1] Frank L. Lewis, Darren M. Dawson, Chaouki T. Abdallah, *Robot manipulator control: Theory and practice*, CRC Press, 2003.
- [2] C.-E. Framing , R. Hedinger , E.S. Iglesias , F.-J. Heßeler , D. Abel, *Edubal: An open balancing robot platform for teaching control and system theory*, IFAC-PapersOnLine **53** (2020), no. 2, 17168–17173.
- [3] Peter Beater, *Pneumatic drives*, Springer, 2007.
- [4] J. Akesson , A. Blomdell , R. Braun, *Design and control of yaip—an inverted pendulum on two wheels robot*, IEEE International Conference on Control Applications, 2006, pp. 2178–2183.
- [5] Amal Choukchou-Braham, Brahim Cherki, Mohamed Djemaï, Krishna Busawon, *Analysis and control of underactuated mechanical systems*, Springer Science & Business Media, 2013.
- [6] Peter Corke, *Robotics, vision and control: Fundamental algorithms in matlab*, Springer, 2017.
- [7] Peter I. Corke, *Visual control of robots: High-performance visual servoing*, Mechatronics (vol 2) Research Studies Press (John Wiley), 1996.
- [8] John J. Craig, *Introduction to robotics: Mechanics and control*, Pearson, 2005.
- [9] Clarence W. de Silva, *Mechatronics: a foundation course*, CRC Press, 2010.
- [10] Toma-Leonida Dragomir, *Elemente de teoria sistemelor*, Editura Politehnica, 2004.
- [11] Glyn James , Phil Dyke, *Advanced modern engineering mathematics*, Pearson, 2018.
- [12] ———, *Modern engineering mathematics*, Pearson, 2020.

- [13] Bruno Siciliano, Oussama Khatib (ed.), *Springer handbook of robotics, Editia a 2-a*, Springer, 2016.
- [14] William S. Levine (ed.), *The control handbook (vol. 3) - control systems advanced methods*, CRC Press, 2011.
- [15] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, *Feedback control of dynamic systems*, Pearson, 2020.
- [16] Noah D. Manring , Roger C. Fales, *Hydraulic control systems*, John Wiley & Sons, 2020.
- [17] J. Denavit, R.S. Hartenberg, *A kinematic notation for lower pair mechanisms*, Applied Mechanics **22** (1955), 215–221.
- [18] H. Lacheray , J. Desouza-Coelho J. S. Furtado , H. H. Liu , G. Lai, *Comparative analysis of optitrack motion capture systems*, Advances in Motion Sensing and Control for Robotic Applications (2019), 15–31.
- [19] Mathe Antal Koppany, *Nonlinear control for commercial drones in autonomous railway maintenance*, Teză de doctorat, Universitatea Tehnică din Cluj-Napoca, 2016.
- [20] D. Banabic , M.M. Radulescu , C. Ratiu L. Deacu, *Tehnica hidraulicii proporționale*, Dacia, 1989.
- [21] Jean-Jacques Slotine , Weiping Li, *Applied nonlinear control*, Pearson, 1991.
- [22] Daniel Liberzon, *Calculus of variations and optimal control theory: A concise introduction*, Princeton University Press, 2012.
- [23] J.A. Marshall , W. Sun , A. L’Afflitto, *A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems*, Annual Reviews in control **52** (2021), 390–427.
- [24] Cosmin Marcu, *Sisteme de conducere a roboților*, Lucrări de laborator, Universitatea Tehnică din Cluj-Napoca, 2017.
- [25] Zoltán Nagy, *Hardware and controller design for mechanical systems - inverted pendulum and robot arm*, Lucrare de dizertație, Universitatea Tehnică din Cluj-Napoca, 2017.
- [26] B.J Emran , H. Najjaran, *A review of quadrotor: An underactuated mechanical system*, Annual Reviews in Control **46** (2018), 165–180.

- [27] I. Ulici , A. Codrean , T. Natsakis, *Ros-based toolbox for motor parameter identification of robotic manipulators*, IFAC Conference on Intelligent Control and Automation Sciences (ICONS), 2022, pp. 136–141.
- [28] Katsuhiko Ogata, *Modern control engineering*, Pearson, 2009.
- [29] B. Theys , G. Dimitriadis , T. Andrienne , P. Hendrick , J. De Schutter, *Wind tunnel testing of a vtol mav propeller in tilted operating mode*, IEEE International Conference on Unmanned Aircraft Systems (ICUAS), 2014, pp. 1064–1072.
- [30] A. Jaritz , M. W. Spong, *An experimental comparison of robust control algorithms on a direct drive manipulator*, IEEE Transactions on Control Systems Technology **4(6)** (1996), 627–640.
- [31] Mark W. Spong, Seth Hutchinson, M. Vidyasagar, *Robot modeling and control*, Wiley, 2006.
- [32] Șerban Onciu, *Controlul unui braț robotic de tip ur5*, Lucrare de licență, Universitatea Tehnică din Cluj-Napoca, 2017.