6. Controlul în Robotică

Modelul unei drone

Figura 6.1 ilustrează anatomia unui sistem de control. Starea curentă $\mathbf{x}(t)$ a dronei este calculată de îndată ce datele sunt citite de la senzorii dronei (e.g. sonar, cameră, senzori inerțial, etc.). Această stare este mai departe comparată cu starea dorită $\mathbf{x}_0$, operație urmată de calculul unei valori de control $\mathbf{u}$, valoare ce ar trebui să aducă starea dronei mai aproape de starea dorită $\mathbf{x}_0$.

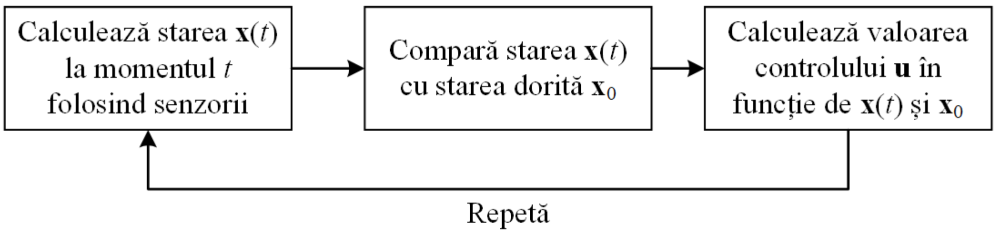

Fig. 6.1 Diagrama bloc al unui sistem de control. În cazul unei drone, ciclu de control se repetă cu o frecvență de $500Hz$.

Așa cum este prezentat în Figura 6.2, dorim ca din diferite stări (poziții) ale dronei, să ajungem în starea (poziția) dorită $\mathbf{x}_0$. Algoritmul utilizat în calculul valorii de control se numește *regulator*, *controller*, sau *lege re reglare*.

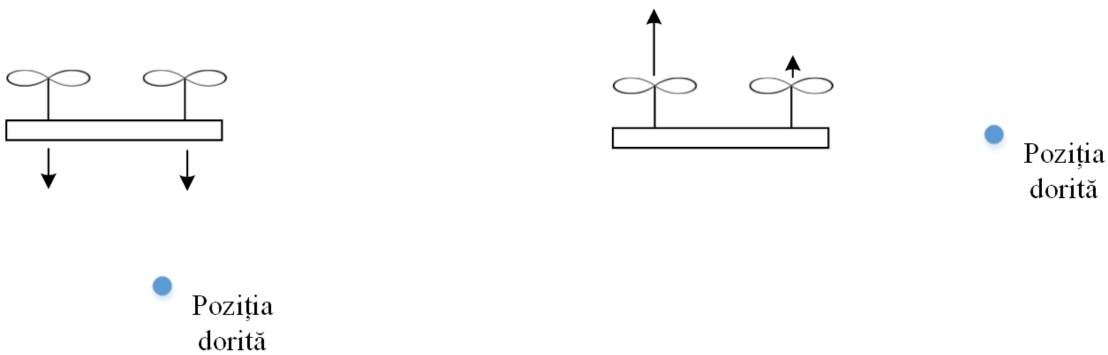

Fig. 6.2 .

Una dintre cele mai simple legi de reglare este reglarea proporțională:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_0 + K(\mathbf{x} - \mathbf{x}_0) \tag{6.1}$$

unde $\mathbf{u}_0$ este valoarea de control necesară să mențină drona pe loc în zbor, $\mathbf{x}_0$ este starea dorită (locul unde dorim ca drona să stea pe loc în zbor), iar K este o matrice pe care o vom defini. Dacă robotul este deja în starea dorită ($\mathbf{x} = \mathbf{x}_0$), atunci regulatorul va aplica doar valoarea $\mathbf{u}_0$ ca si valoare de control. Dacă $\mathbf{x} \neq \mathbf{x}_0$, atunci vom aplica o valoare de control care este proporțională cu deviația stării $\mathbf{x}$ față de starea dorită $\mathbf{x}_0$.

6.1 Punctul fix (punctul de echilibru)

Un punct fix, sau punct de echilibru, pentru sistemul $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ este o stare $\mathbf{x}_0$ în care:

$$\dot{\mathbf{x}} = f(\mathbf{x}_0, \mathbf{u}_0) = 0, \quad \text{pentru un anumit } \mathbf{u}_0 \tag{6.2}$$

Zburatul pe loc al dronei satisface această condiție. Intuitiv, un punct fix $\mathbf{x}_0$ este o stare în care drona rămâne atunci când se aplică controlul $\mathbf{u}_0$. Pentru dronă, acest punct fix asociat cu zburatul pe loc este:

$$\mathbf{x}_0 = \begin{bmatrix} x_0 \\ y_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{și} \mathbf{u}_0 = \begin{bmatrix} mg \\ 0 \end{bmatrix} \tag{6.3}$$

Nu este posibil să avem un punct fix $\mathbf{x}_0 = \begin{bmatrix} x_0 & y_0 & 0 & \dot{x}_0 & \dot{y}_0 & \dot{\theta}_0 \end{bmatrix}^T$ în care $\dot{x}_0 \neq 0$, $\dot{y}_0 \neq 0$, sau $\dot{\theta}_0 \neq 0$.

6.2 Controlul Proporțional-Derivativ (PD)

Fie drona 2D din Figura 6.3, constrânsă să se deplaseze doar pe axa Y.

Dinamica este dată de ecuația:

$$\ddot{y} = \frac{u_1}{m} - g \tag{6.4}$$

respectiv starea:

$$\mathbf{x} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \tag{6.5}$$

Valoarea de control u_1 este propulsia totală produsă de elice.

Dorim să stabilizăm sistemul în jurul stării:

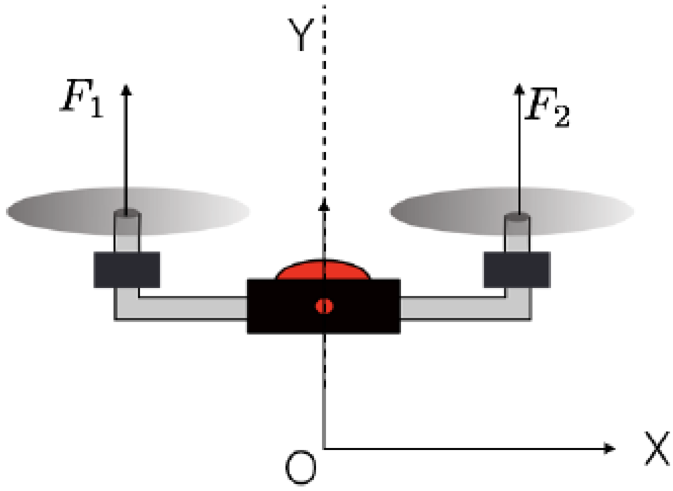

Fig. 6.3 Dronă 2D contrânsă să se deplaseze doar pe axa Y.

$$\mathbf{x}_0 = \begin{bmatrix} y_0 \\ 0 \end{bmatrix} \tag{6.6}$$

cu $u_0 = mg$. Acesta este un punct fix, deoarece $\ddot{y} = \frac{mg}{m} - g = 0$.

Pentru a stabili sistemul la valoarea $\mathbf{x}_0$, putem alege valoarea lui $\mathbf{u}$:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_0 + k_P(y - y_0) \tag{6.7}$$

Sistemul în buclă închisă este astfel:

$$\ddot{y} = \frac{mg}{m} - g + \frac{k_P}{m}(y - y_0) = \frac{k_P(y - y_0)}{m} \tag{6.8}$$

unde $k_P < 0$ este o constantă aleasă de noi.

Fie $\bar{y} = y - y_0$, $\dot{\bar{y}} = \dot{y}$ și $\ddot{\bar{y}} = \ddot{y}$. Atunci:

$$\ddot{\bar{y}} = \frac{k_P}{m}(y - y_0) = \frac{k_P}{m}\bar{y} \tag{6.9}$$

Acest controller nu ar reuși să stabilizeze sistemul, deoarece folosește doar o componentă proporțională (k_P) în calculul valorii de control. O altă opțiune este să folosim și o componentă derivativă:

$$u(\mathbf{x}) = u_0 + \frac{k_P}{m}(y - y_0) + k_D\dot{y} \tag{6.10}$$

unde $k_D < 0$ este o altă constantă aleasă de noi. Astfel:

$$\ddot{\bar{y}} = \frac{k_P}{m}\bar{y} + \frac{k_D}{m}\dot{\bar{y}} \tag{6.11}$$

Ecuația anterioară este un exemplu de controller **Proporțional-Derivativ (PD)**, unde

k_P se numește factor de amplificare proporțional, iar k_D factor de amplificare derivativ.

Controllerul are forma:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_0 + K(\mathbf{x} - \mathbf{x}_0) = \mathbf{u}_0 + \begin{bmatrix} k_P & k_D \end{bmatrix} \begin{bmatrix} y - y_0 \\ \dot{y} \end{bmatrix} \tag{6.12}$$

Matricea K poartă denumirea de **matrice de amplificare**.

În acest exemplu, orice valoarea a factorilor k_P și k_D mai mică decât zero va rezulta în stabilitate asimptotică globală.

6.3 Linear Quadratic Regulator (LQR)

Metoda LQR este direct aplicabilă sistemelor liniare. În cazul sistemelor neliniare, putem liniariza modelul în jurul unui punct fix (de exemplu punctul pentru zburatul pe loc), urmată de aplicarea metodei pentru sistemul liniarizat:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \approx \dot{\mathbf{x}} = A(\mathbf{x} - \mathbf{x}_0) + B(\mathbf{u} - \mathbf{u}_0) \;\rightarrow\; \text{control bazat pe } A, B \tag{6.13}$$

Linear Quadratic Regulator (LQR) este o metodă ce are două obiective simultane:

1. Stabilizarea sistemului liniar, în cazul în care acesta este stabilizabil,

2. Optimizarea unui criteriu de performanță.

Fie sistemul liniar:

$$\dot{\mathbf{x}} = A(\mathbf{x} - \mathbf{x}_0) + B(\mathbf{u} - \mathbf{u}_0) \tag{6.14}$$

unde $\mathbf{x} \in \Re^n$, $\mathbf{u} \in \Re^m$, iar $\mathbf{x}_0$ este un punct fix. Pentru a simplifica notațiile, vom considera $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$ și $\bar{\mathbf{u}} = \mathbf{u} - \mathbf{u}_0$. Astfel, ecuația precendentă devine:

$$\dot{\mathbf{x}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}} \tag{6.15}$$

Exemple de criterii de performanță pot fi:

- minimizarea deviației stării de la starea dorită $\mathbf{x}_0$
- minimizarea deviației variabilei de control de la valoarea sa nominală $\mathbf{u}_0$

Combinarea acestor două criterii rezultă în funcția de cost:

$$J(\bar{\mathbf{x}}(0)) = \int_0^\infty \left[\bar{\mathbf{x}}(t)^T\bar{\mathbf{x}}(t) + \bar{\mathbf{u}}(t)^T\bar{\mathbf{u}}(t)\right] dt \tag{6.16}$$

unde $\bar{\mathbf{x}}(t)^T\bar{\mathbf{x}}(t) = ||\bar{\mathbf{x}}(t)||^2$ și $\bar{\mathbf{u}}(t)^T\bar{\mathbf{u}}(t) = ||\bar{\mathbf{u}}(t)||^2$ sunt două valori scalare. $\bar{\mathbf{x}}(t)$ este starea la timpul t, luând în considerare că am pornit de la starea $\bar{\mathbf{x}}(0)$.

În funcția de cost de mai sus penalizăm stările și variabilele de control în mod egal. În cazul în care dorim să penalizăm anumite stari sau semnale de control decât altele, putem introduce următoarele matrici de ponderare:

$$J(\bar{\mathbf{x}}(0)) = \int_0^\infty \Big[\bar{\mathbf{x}}(t)^T Q \bar{\mathbf{x}}(t) + \bar{\mathbf{u}}(t)^T R \bar{\mathbf{u}}(t)\Big] dt \tag{6.17}$$

unde $Q \in \Re^{n\times n}$ și $Q \in \Re^{m\times m}$ sunt matrici predefinite ce indică importanța anumitor stări sau variabile de control. Aceste matrici trebuie să fie *pozitiv definite* și *simetrice*. O matrice simetrică Q este pozitiv definită dacă $\mathbf{x}^T Q \mathbf{x} > 0$, pentru toate $\mathbf{x}$. Aceasta este echivalent cu valorile proprii pozitive ale lui Q. În practică, vom alege Q și R să fie *diagonale*.

Obiectivul nostru este să găsim o modalitate de mapare a stărilor la variabilele de control, în așa fel încât sa minimizăm funcția de cost. Soluția pentru aceasta este controllerul *liniar*:

$$\bar{\mathbf{u}}(\bar{\mathbf{x}}) = K^* \bar{\mathbf{x}} \tag{6.18}$$

unde K^* reprezintă soluția matricei de amplificare. Valorile lui K^* se obțin în doi pași:

- **Pasul 1**: Rezolvarea următoarei ecuații matriceale, pentru a găsi matricea $S \in \Re^{n\times n}$:

$$Q - SBR^{-1}B^T S + SA + A^T S = 0 \tag{6.19}$$

  S este singura necunoscută din această ecuație, denumită *Ecuația Riccatti*, având soluția S^*.

- **Pasul 2**: Calculează:

$$K^* = -R^{-1}B^T S^* \tag{6.20}$$

  Algoritmul de control final este:

$$\bar{\mathbf{u}}(\bar{\mathbf{x}}) = K^* \bar{\mathbf{x}} \tag{6.21}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_0 + K^*(\mathbf{x} - \mathbf{x}_0) \tag{6.22}$$

Ecuația Riccatti poate fi rezolvată utilizând metode numerice. În Python sau Matlab, rezolvarea este oferită de funcția *lqr*, ce poate fi apelată:

$$[K^*, S^*] = lqr(A, B, Q, R) \tag{6.23}$$

Python și Matlab folosesc o convenție diferită în funcția *lqr*, și anume calculeză cu semn negativ valoarealui K: $\bar{\mathbf{u}}(\bar{\mathbf{x}}) = -K^* \bar{\mathbf{x}}$.

Interpretarea lui S^*: Soluția S^* ne oferă costul optim:

$$J^*(\bar{\mathbf{x}}(0)) = \bar{\mathbf{x}}(0)^T S^* \bar{\mathbf{x}}(0) \tag{6.24}$$

unde $J^*(\bar{\mathbf{x}}(0))$ este costul introdus de aplicarea lui $K^*\bar{\mathbf{x}}$ atunci când pornim din $\bar{\mathbf{x}}(0)$.

Stabilitatea: Algoritmul de control stabilizează (asimptotic global) sistemul către punctul fix $\bar{\mathbf{x}} = 0$ ($\mathbf{x} = \mathbf{x}_0$). Astfel, sistemul dinamic $\dot{\mathbf{x}} = A(\mathbf{x} - \mathbf{x}_0) + B(\mathbf{u} - \mathbf{u}_0)$ este stabilizat la punctul fix $\mathbf{x}_0$.

Algoritmul LQR reușește să rezolve două probleme:

- minimizează un criteriu de performanță, destinat minimizării deviațiilor lui $\mathbf{x}$ de la $\mathbf{x}_0$ și ale lui $\mathbf{u}$ de la $\mathbf{u}_0$.
- rezultatul final este un sistem în buclă închisă stabil asimptotic global:

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}} = A\bar{\mathbf{x}} + BK\bar{\mathbf{x}} = (A + BK)\bar{\mathbf{x}} \tag{6.25}$$

Alegerea lui Q și R: În teorie, orice valoare pozitiv semidefinită pentru Q și R ar trebui să facă sistemul stabil. În realitate, acest lucru este greu de realizat, deoarece:

1. sistemul este în realitate neliniar,
2. modelul dinamic nu este perfect,
3. starea sistemului ar trebui estimată perfect pentru a putea utiliza acest controller.

Prin experimente, putem însă să alegem valorile pentru Q și R în așa fel încât să obținem comportamentul dorit al sistemului. Spre exemplu, dacă observăm ca sistemul (drona) nu se stabilizează pe axa Z, putem modifica a treia valoarea de pe diagonala matricei Q, aceasta fiind asociată variabilei de stare ce corespunde poziției pe axa Z din vectorul de stare ($[x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]$).

Bibliografie

[1] K. Hauser, *Robotic Systems*. Illinois, IL, USA: university of illinois urbana-champaign, 2023. [Online]. Available: https://motion.cs.illinois.edu/RoboticSystems/

[2] C. Powers, D. Mellinger and V. Kumar, *Quadrotor Kinematics and Dynamics*. Dordrecht: Springer Netherlands, 2015, pp. 307–328. [Online]. Available: https://doi.org/10.1007/978-90-481-9707-1_71