Laborator 6 (SPER)

Planificarea mișcării în formație

22 martie 2022

Cuprins

compilat la: 22/03/2022, 01:02

Scopul lucrării

Discutăm despre arhitectura "leader-followers" ce permite să controlăm comportamentul unei echipe de agenți. Traiectoria liderului este obținută prin aplicarea noțiunii de câmp potențial iar urmăritorii sunt menținuți în formație printr-o comandă de tipul "formation control" ce penalizează devieri față de pozițiile și vitezele de echilibru.

1 Noțiuni teoretice

Problema planificării pentru o echipă de agenți este complexă deoarece:

- complexitatea de calcul crește cu numărul de agenți implicați;
- apar dificultăți specifice (alocare de sarcini, evitare de coliziuni între agenți);
- apar constrângeri structurale (de comunicație, de rază de observație).

O soluție populară este implementarea unei metode de tipul *leader-followers*:

lider este agentul ce are ca sarcină generarea/urmărirea unei traiectorii ce permite atingerea destinației;

urmăritori au legi de comandă simple ce permit urmărirea liderului și menținerea unei formații (de obicei, cu structură predefinită).

Până acum, traiectoria liderului a rezultat din proceduri precum cele descrise în Laboratorul 3 (traiectoria este generată offline iar apoi este urmărită online) sau în Laboratorul 5 (traiectoria este calculată online pe baza informației curente). Considerăm un sistem[1] "integrator", caracterizat de poziție ($p \in \mathbb{R}^n$) și viteză ($v \in \mathbb{R}^n$), acestea fiind starea, respectiv comanda sistemului:

$$\dot{p} = v. \tag{1}$$

Definim un *câmp potențial* ce conține:

componente repulsive Pentru o colecție de obstacole cu centre c_i aplicăm o lege proporțională cu distanța față de acestea (repulsie radială):

$$\phi\left(\|p - c_i\|\right) = \frac{a_1}{a_2 + \|p - c_i\|}, \quad \phi\left(\|p - c_i\|\right) = \begin{cases} \dfrac{a_1}{a_2}, & p = c_i, \\ 0, & \|p - c_i\| \to \infty, \end{cases} \tag{2}$$

componentă atractivă Pentru a direcționa agentul către destinație, penalizăm distanța față de acesta (liniar, exponențial, etc.):

$$\psi\left(\|p - p_d\|\right) = \|p - p_d\|. \tag{3}$$

[1] La următorul nivel de complexitate considerăm modelele "uniciclu" sau "mașină Dubins".

Câmpul potențial total este format din suma acestor componente:

$$P(x) = \psi\left(\|p - p_d\|\right) + \sum_i \phi\left(\|p - c_i\|\right), \tag{4}$$

iar comanda aplicată sistemului este (în cazul cel mai simplu), proporțională cu gradientul câmpului potențial în punctul ce corespunde poziției curente:

$$u = -\nabla P(p) = -\left(\frac{p - p_d}{\|p - p_d\|} + \sum_i \frac{p - c_i}{\|p - c_i\|}\right). \tag{5}$$

Pentru a asigura o formație avem la îndemână mai multe metode (ce depind de natura dinamicii agentului), de senzorii acestuia (poziționare relativă sau absolută, rază de acoperire) și de graful de comunicație între agenți.

Considerăm N agenți "dublu-integrator", caracterizați de stare (compusă din poziție – $p \in \mathbb{R}^n$ și viteză – $v \in \mathbb{R}^n$) și comandă (accelerație – $u \in \mathbb{R}$):

$$\dot{p}_i = v_i, \quad v_i = u_i. \tag{6}$$

Fiecare agent are o listă de vecini $\mathcal{N}_i$ de care este legat prin ponderi $\omega_{ij} \geq 0$ (valoarea zero apare dacă agenții i, j nu sunt vecini).

Relațiile de adiacență (graful de interconexiuni al agenților) sunt caracterizate de "matricea Laplaciană":

$$L = \left[\ell_{ij}\right]_{i,j=1:N}, \text{ cu } \ell_{ij} = \begin{cases} \sum\limits_{k \in \mathcal{N}_i} \omega_{ik}, & \text{dacă } i = j, \\ -\omega_{ij}, & \text{dacă } i \neq j. \end{cases} \tag{7}$$

Pentru definirea unei formații se impun relații între poziții și viteze relative față de vecini:

$$p_j - p_i = p_j^\star - p_i^\star, \quad v_j - v_i = v_j^\star - v_i^\star. \tag{8}$$

Acestea sunt respectate prin alegerea unor legi de comandă de forma

$$\begin{aligned} u_i = &- k_p \sum_{j \in \mathcal{N}_i} \omega_{ij}(p_i - p_j - p_i^\star + p_j^\star) \\ &- k_v \sum_{j \in \mathcal{N}_i} \omega_{ij}(v_i - v_j - v_i^\star + v_j^\star), \end{aligned} \tag{9}$$

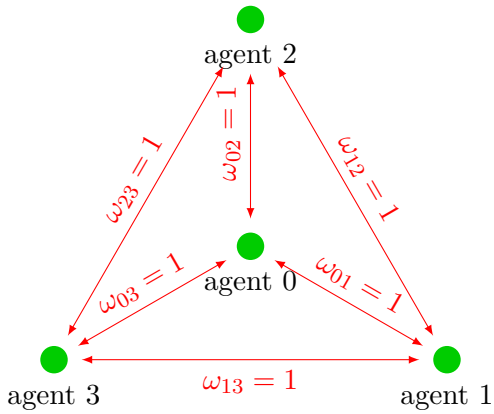
unde $k_p > 0$, $k_v > 0$.
Evoluția erorilor de poziție și respectiv viteză, $e_p = p^\star - p$, $e_v = v^\star - v$ este guvernată de dinamica

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_v \end{bmatrix} = \begin{bmatrix} 0 & I_{nN} \\ -k_p(L \otimes I_n) & -k_v(L \otimes I_n) \end{bmatrix} \begin{bmatrix} e_p \\ e_v \end{bmatrix}, \tag{10}$$

unde $'\otimes'$ denotă produsul Kronecker.

2 Implementare a unui algoritm de menținere a formației

Considerăm 1+3 agenți pe care dorim să ii aducem într-o configurație triunghi: în centru este poziționat liderul (agentul 0) iar urmăritorii (agenții 1, 2 și 3) se poziționează relativ față de acesta și îi urmăresc acestuia viteza, după cum este schițat în continuare:



Laplacianul grafului de mai sus, cf. definiției (7) este

$$L=\begin{bmatrix} 4 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}, \text{cu } \mathcal{N}_1=\{0,2,3\},\ \mathcal{N}_2=\{0,1,3\},\ \mathcal{N}_3=\{0,1,2\}. \quad (11)$$

iar pozițiile și vitezele ce definesc traiectoria sunt:

$$p_0=\Big(0,0\Big),\ p_1=\Big(\frac{\sqrt{3}}{2},-\frac{1}{2}\Big),\ p_2=\Big(0,1\Big),\ p_3=\Big(-\frac{\sqrt{3}}{2},-\frac{1}{2}\Big), \quad (12a)$$

$$v_0=\Big(0,0\Big),\ v_1=v_2=v_3=\Big(0,0\Big). \quad (12b)$$

Având aceste elemente construim acum comenzile (9) ce se aplică agenților:

```
control=control-kp*(agent[i-1]['position'][-1,:]-agent[j-1]['position'][-1,:]-p[i,:]+p[j,:])-kv*(agent[i-1]['velocity'][-1,:]-agent[j-1]['velocity'][-1,:]-v[i,:]+v[j,:])
```

Aceasta ne permite ulterior să incrementăm[2] starea sistemului (poziție și viteză):

```
agent[i-1]['position']=np.vstack((agent[i-1]['position'], agent[i-1]['position'][-1,:]+T*agent[i-1]['velocity'][-1,:]))
agent[i-1]['velocity']=np.vstack((agent[i-1]['velocity'], agent[i-1]['velocity'][-1,:]+T*control))
```

[2]Deși în (6) considerăm dinamica în timp continuu, în cod implementăm varianta discretizată.

3 Exerciții propuse

Exerciţiul 1. Codul asociat acestui laborator implementează doar problema de menținere a formației ("formation control") presupunând că liderul este în repaus. Modificați codul astfel încât liderul să urmărească o traiectorie arbitrară.

Exerciţiul 2. Modificați codul (inclusiv varianta obținută în exercițiul anterior) pentru cazul mașinii Dubins a cărei dinamică este dată de ecuațiile:

$$\begin{cases} \dot{x} &= u_V \cos\phi, \\ \dot{y} &= u_V \sin\phi, \\ \dot{\phi} &= \frac{u_V}{L} \tan u_\phi. \end{cases}$$

Exerciţiul 3 (Tema 2 – 15p). Rezolvați următoarele cerințe:

i) Pentru o colecție dată de obstacole (definite de centrele lor) și o destinație, construiți și ilustrați câmpul potențial definit în (4); **[5p]**

ii) Plecând dintr-un punct inițial aleator, aplicați comanda (5) sistemului (1). Ilustrați traiectoria rezultată; **[5p]**

iii) Considerați un grup de agenți cărora le aplicați comanda de la punctul anterior dar unde, în plus, adăugați un termen de evitare a coliziunii:

$$\sum_{i \neq j} \phi\left(\|p_i - p_j\| - d_{ij}\right) = \sum_{i \neq j} \frac{a_1}{a_2 + \|p_i - p_j\| - d_{ij}}.$$

Acest termen menține distanța dintre agenții i și j la valoarea $d_{ij} > 0$. Ilustrați traiectoriile rezultate. **[5p]**