

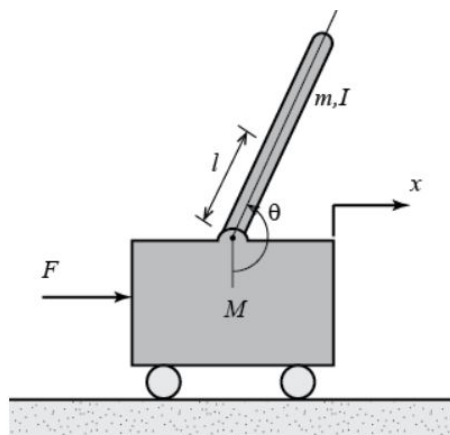
# Optimizare

## Laborator 6: Control optimal. Metoda bariera

Acest laborator incepe cu o scurta prezentare a sistemului CVX, acesta preocupandu-se cu rezolvarea problemelor convexe. Dupa aceasta sectiune formulam o problema de control optimal pentru pendulul invers, in care am sarit peste pasul de modelare, nefiind obiectul cursului. Rezolvam problema cu doua metode invatate la curs si anume metoda barierei si cea a gradientului proiectat, iar solutiile obtinute sunt comparate cu cele gasite cu CVX si functiile oferite de Matlab. In cea de a treia sectiune, prezentam problema lantului suspendat pentru care am implementat gradientul proiectat si cel conditionat. In aceeasi maniera ca si in sectiunea precedenta, comparam cu solutiile produse de CVX si functiile Matlab.

### 1 Control optimal

Sistemul din acest exemplu consta dintr-un pendul inversat montat pe un carucior motorizat. Popularitatea sistemului deriva in parte din faptul ca este instabila fara control, adica pendulul va cadea pur și simplu daca nu este mișcat caruciorul pentru a-l echilibra. In plus, dinamica sistemului este neliniară. Obiectivul sistemului de control este de a echilibra pendulul inversat prin aplicarea unei forțe pe caruciorul de care pendulul este atasat. Un exemplu din lumea reala care se refera direct la acest sistem de pendul inversat este controlul atitudinii unei rachete rapel la decolare.



la noi o sa fie phi, nu teta

Notam:

- $M$  - masa caruciorului-ului = 0,5 kg

- m - masa pendulului = 0,2 kg
- b - coeficientul de frecare pentru carucior = 0,1 N/m/sec
- l - lungimea până la centrul de masă al pendulului = 0,3 m
- I - momentul masic de inerție a pendulului = 0,006 kg m<sup>2</sup>
- F - forța aplicată caruciorului
- x - coordonata de poziție a caruciorului
- $\phi$  - unghiul pendulului

Utilizând legile lui Newton și liniarizând ecuațiile dinamicii găsim următoarea reprezentare pe stare a sistemului:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ ml \end{bmatrix} u \Leftrightarrow \dot{z}_{t+1} = Az_t + Bu_t$$

Pentru acest sistem formulăm în continuare problema de control optimal pe orizont finit N pentru urmărirea unei referințe date.

Pentru aceasta considerăm costuri de etapă:  $\ell_t^z(z_t) = 1/2 \|z_t - z_t^{ref}\|_{Q_t}^2$ ,  $\ell_t^u(u_t) = 1/2 \|u_t - u_t^{ref}\|_{R_t}^2$ , unde  $Q_t \succeq 0$  și  $R_t \succ 0$ . (Notatie:  $\|x - x^{ref}\|_Q^2 = (x - x^{ref})^T Q (x - x^{ref})$ )

De asemenea considerând dimensiunile caruciorului trebuie impusă o restricționată forță aplicată asupra acestuia. Astfel avem ca:  $-3 \leq u \leq 3$  **=> constrangere pe intrare**

Deci avem următoare formulare a problemei de control:

$$\begin{aligned} \min_{z_t, u_t} \quad & \sum_{t=1}^N \ell_t^z(z_t) + \sum_{t=0}^{N-1} \ell_t^u(u_t) \\ \text{s.l.:} \quad & z_{t+1} = A_t z_t + B_t u_t \quad \forall t = 0, \dots, N-1, \quad z_0 \text{ dat.} \\ & -3 \leq u_t \leq 3 \end{aligned}$$

funcția obiectiv este patratică  
constrângeri liniare  
=> ducem problema în forma QP

**facem QP cu dense NPC (eliminarea stărilor)**

Observăm că variabila de decizie pentru problema de mai sus este:

$$x = [u_0^T \quad z_1^T \quad \dots \quad u_{N-1}^T \quad z_N^T]^T \in \mathbb{R}^{N(n_z+n_u)}$$

Pentru simplitate vom considera în cele ce urmează ca orizontul de predicție N = 3.

Putem să restrângem variabila și să o scriem doar în funcție de intrări, utilizându-ne de dinamica sistemului:

$$\begin{aligned} z_1 &= Az_0 + Bu_0 \\ z_2 &= Az_1 + Bu_1 = A^2z_0 + ABu_0 + Bu_1 \\ z_3 &= Az_2 + Bu_2 = A^3z_0 + A^2Bu_0 + ABu_1 + Bu_2 \end{aligned}$$

z - vector de stare  
vector de stare în funcție de intrări

Eliminand starile obtinem ca variabila de decizie:

$$x = \begin{bmatrix} u_0^T & u_1^T & u_2^T \end{bmatrix}^T$$

De asemenea notand  $\bar{z} = [z_1^T \ z_2^T \ z_3^T]^T$  putem scrie ecuatie de mai sus in urmatoarea forma compacta

$$\bar{z} = \overline{AB}x + A_p z_0$$

forma asta va fi stricata (dense NPC)

unde:

$$\overline{AB} = \begin{bmatrix} B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{bmatrix}, A_p = \begin{bmatrix} A \\ A^2 \\ A^3 \end{bmatrix}$$

Prin prisma celor de mai sus, putem rescrie functia cost pentru stare:

$$\begin{aligned} \frac{1}{2} \sum_{j=1}^3 \|z_j - z_j^{\text{ref}}\|_{Q_j}^2 &= \frac{1}{2} \|\bar{z} - \bar{z}^{\text{ref}}\|_{\bar{Q}}^2 = \frac{1}{2} \|\overline{AB}x + A_p z_0 - \bar{z}^{\text{ref}}\|_{\bar{Q}}^2 \\ &= \frac{1}{2} x^T \overline{AB}^T \bar{Q} \overline{AB} x + \left( z_0^T A_p^T \bar{Q} \overline{AB} - (\bar{z}^{\text{ref}})^T \bar{Q} \overline{AB} \right) x \end{aligned}$$

unde  $\bar{Q} = \text{diag}(Q_1, \dots, Q_3)$  si  $\bar{z}^{\text{ref}} = [z_1^{\text{ref}} \dots z_3^{\text{ref}}]^T$

Procedand in aceasi maniera si pentru intrare, gasim:

$$\begin{aligned} \sum_{j=0}^2 \|u(j) - u^{\text{ref}}(j)\|_{R_j}^2 &= \|x - \bar{x}^{\text{ref}}\|_{\bar{R}}^2 \\ &= \frac{1}{2} x^T \bar{R} x - (\bar{x}^{\text{ref}})^T \bar{R} x + \frac{1}{2} (\bar{x}^{\text{ref}})^T \bar{R} \bar{x}^{\text{ref}} \end{aligned}$$

unde  $\bar{R} = \text{diag}(R_0, \dots, R_2)$  si  $\bar{x}^{\text{ref}} = [x_0^{\text{ref}} \dots u_2^{\text{ref}}]$

In cele de urma putem scrie functia obiectiv ca:  $\frac{1}{2} x^T H x + h^T x$ , unde  $H = \bar{R} + \overline{AB}^T \bar{Q} \overline{AB}$  si  $h = \overline{AB}^T \bar{Q} A_p z_0 - \overline{AB}^T \bar{Q} \bar{z}^{\text{ref}} - \bar{R} \bar{x}^{\text{ref}}$

Eliminand starile, am mai ramas doar cu constrangerea pe intrare:  $-3 \text{ones}(3, 1) \leq x \leq 3 \text{ones}(3, 1)$  cum pun constrangerea  $-3 \leq u_i \leq 3$  sub forma  $Cx \leq d$ ?

In concluzie avem de rezolvat urmatoarea problema QP:

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T H x + h^T x \\ \text{s.t.} \quad & Cx \leq d \end{aligned}$$

$-3 * [1 \leq x \leq 3]^T \Rightarrow I * x \leq 3 * [1 \ 1 \ 1]^T$   
 $\Rightarrow C = \begin{bmatrix} I \\ -I \end{bmatrix}$   
 $d = 3 * [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$

A se observa ca reformularea problemei cu metoda eliminarea starilor obtinem o problema de optimizare care are doar constrangeri de inegalitate.

Amintim pentru rezolvarea acestei probleme doi algoritmi si vom compara rezultatele acestora cu ce ne va furniza cvx-ul si functia quadprog.

### • Metoda bariera

Fie:  $\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad g(x) \leq 0, Ax = b.$

unde  $f, g$  sunt functii convexe. Formulam problema bariera:

m - nu inseamna masa pendului

$$\min_{x \in \mathbb{R}^n} f(x) - \tau \sum_{i=1}^m \log(-g_i(x)) \quad \text{s.t.} \quad Ax = b. \quad (1)$$

x, punctul initial, strict fezabil = sa respecte toate constrangerile

Alegem  $x_0$  strict fezabil,  $\tau_0 > 0, \sigma < 1$  si  $\epsilon > 0$ .

Atata timp cat  $m\tau_k \geq \epsilon$  repeta:

problema fara constrangeri

1. Calculeaza

metoda gradient/ metoda newton (astazi MG)

$$x_{k+1} = x(\tau_k) \quad (2)$$

pornind din punctul initial  $x_k$  ("warm start"); sunt in bucla exteriora, calc  $x_{k+1}$ , calc  $x_k$ , si ma intorc iar la mg, dar nu de la 0, de la  $x_k$

2. Descreste parametrul  $\tau_{k+1} = \sigma\tau_k$ .

'warm start' => folosim ce am gasit iteratia anterioara

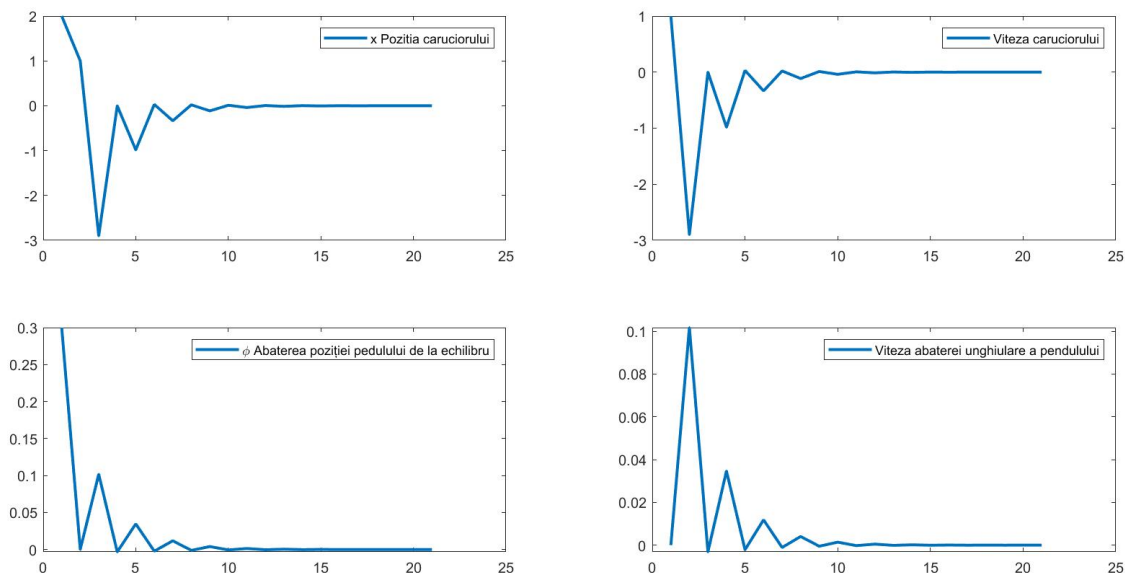
$\Rightarrow$  Pentru a determina  $x_{k+1}$  din expresia 2 rezolvam (e.g. Metoda gradient, Metoda Newton) reformularea cu bariera logaritmica cu parametrul  $\tau_k$  1, pornind din punctul initial  $x_k$ ;

$\Rightarrow$  Dupa  $k$  iteratii avem  $f(x_k) - f^* \leq m\tau_0\sigma^k$ .

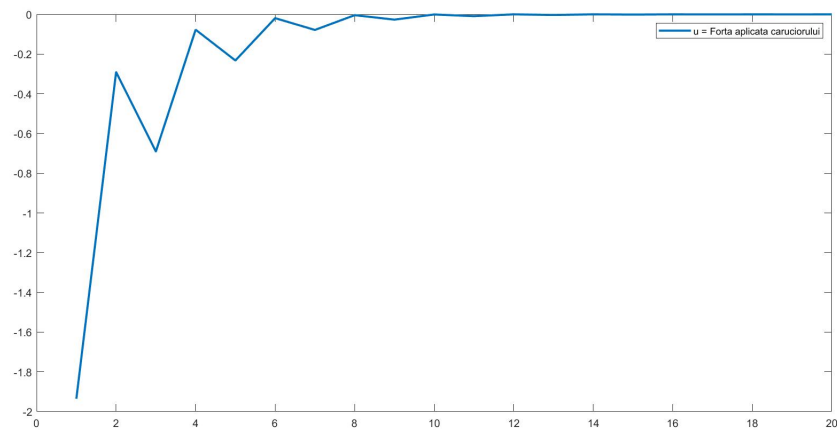
## Rezultate numerice

Luam pentru costul functiei:  $Q = I_4$ , dar 10 pe pozitiile (1,1) si (3,3) si  $R = 1$ . Alegem starea initiala:  $z_0 = [2; 1; 0.3; 0]$ ,  $\tau = 1$ ,  $\sigma = 0.6$ , si  $\epsilon = 0.0001$ .

Graficul variabilei de stare a sistemului



Graficul intrarii



## 2 Cerinta

Inlocuiti in template-ul atasat laboratorului, functia quadprog din matlab cu:

- (3p)Metoda bariera folosind updatarea metodei gradient
- (1p)Versiunea cvx

## References

- [1] R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3
- [2] Modelarea pundulului invers pe un carucior