Politechnika
Śląska

Sprawozdanie z przedmiotu
Metody Optymalizacji

Temat: Metody obliczeniowe optymalizacji dynamicznej z ograniczeniami

Data wykonania sprawozdania: 21.05.2023

Sprawozdanie przygotowali:
Dawid Kania
Oliwier Cieślik
AiR S2-I/Rob

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z metodami obliczeniowymi optymalizacji dynamicznej poprzez znalezienie optymalnych rozwiązań do podanych przez prowadzących problemów.

Minimalizacja przy pomocy programu

W tej części ćwiczenia należy napisać program minimalizujący wskaźnik jakości $\mathrm{I}(\mathrm{x},\mathrm{u})$ dla układu dyskretnego opisanego równaniem stanu (2):

$$\mathrm{I}(\mathrm{x},\mathrm{u}) = \sum_{\mathrm{i}=0}^{6} (\mathrm{x}_\mathrm{i}^2 + 1{,}5\mathrm{u}_\mathrm{i}^2) \tag{1}$$

$$\mathrm{x}_{\mathrm{i}+1} = \mathrm{x}_\mathrm{i} + 3\mathrm{u}_\mathrm{i} \tag{2}$$

Ograniczenia sterowania i stanu układu w poszczególnych chwilach czasu:

$$\mathrm{x}_0 = 15 \tag{3}$$

$$\mathrm{x}_7 = 70 \tag{4}$$

$$\mathrm{x}_3 = 40 \tag{5}$$

$$|\mathrm{u}_\mathrm{i}| \leq 5 \tag{6}$$

$$\mathrm{u}_3 = 3 \tag{7}$$

$$\varepsilon = 0{,}1 \tag{8}$$

Ograniczenie z równania (6) dla uproszczenia można podzielić na dwa osobne równania postaci $\mathrm{h}_\mathrm{j}(\mathrm{x},\mathrm{u}) < \mathrm{a}_\mathrm{j}$:

$$\mathrm{u}_\mathrm{i} \leq 5 \tag{9}$$

$$-\mathrm{u}_\mathrm{i} \leq 5 \tag{10}$$

Zmodyfikowany funkcjonał kary:

$$\begin{aligned}\bar{I}(\mathrm{x},\mathrm{u}) = \sum_{\mathrm{i}=0}^{6} (\mathrm{x}_\mathrm{i}^2 + 1{,}5\mathrm{u}_\mathrm{i}^2) + t_1 * 0.5 * (\mathrm{x}_7 - 70)^2 \\ + t_2 * 0.5 * (\mathrm{x}_3 - 40)^2 + \\ + t_3 * 0.5 * (u_3 - 3)^2 + \sum_{\mathrm{i}=0}^{6} [t_4(u_i - 5) * \max(\{0, u_i - 5\}))] + \\ + \sum_{\mathrm{i}=0}^{6} [t_5(-u_i - 5) * \max(\{0, -u_i - 5\})]\end{aligned} \tag{11}$$

Rozwiązanie problemu

W celu rozwiązania problemu zostały dobrane następujące parametry:

$$t = [t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5] = [2 \quad 2 \quad 2 \quad 2 \quad 2] \tag{12}$$

$$\alpha = 0{,}5 \tag{13}$$

$$\beta = 2 \tag{14}$$

$$c = 1 \tag{15}$$

Dla danego problemu można określić horyzont sterowania $\mathrm{N} = 7$.

Po wykonaniu obliczeń uzyskano poniższe wartości:

i	0	1	2	3	4	5	6	7
x	15	16,1672014	25,09117	40,00517	48,88648	40,00142	55,02674	70,01671
u	0,389067	2,97465772	4,971332	2,960435	-2,96169	5,008441	4,996654	-

Kod źródłowy

```
% Autorzy:
% Dawid Kania
% Oliwier Cieślik


clear all; close all; clc


x0 = 15; % stan poczatkowy
t = [2 2 2 2 2]; % początkowe wspolczynniki kary
N = 7; % horyzont sterowania
I = @(x,u) sum( x.^2 + 1.5 * u.^2, "all"); % wskaznik jakosci

% ograniczenia
R = {
    @(x,u,v) 0.5*( x(end) - v ).^2
    @(x,u,v) 0.5*( x(3+1) - v ).^2
    @(x,u,v) sum( (u-v).*max(0, (u-v)) ,"all")
    @(x,u,v) sum( (-u-v).*max(0, (-u-v)) ,"all")
    @(x,u,v) 0.5*( u(3+1) - v ).^2
};

a = { [70], [40], ones(N,1) * 5,  ones(N,1) * 5, [3] }
v = a
c = 1

epsilon = 0.1
beta = 2
alfa = 0.5


for nr_iteracji = 1:1000
    % krok 4

    % sterowanie jako wartość startowa funkcji fminsearch
    u = ones(N,1);
    u_optymalne = u;

    % obliczanie optymalnego sterowania
    u_optymalne = fminsearch(@(u) I_calc(u, I, R, t, x0, v), u_optymalne);
    x_optymalne = obliczanie_x(u_optymalne, x0);
    I_optymalne = I_calc(u, I, R, t, x0, v);

    % krok 5 wyznaczenie 'r'
```

```
    r = {
        x_optymalne(end) - a{1}
        x_optymalne(3+1) - a{2} % tu byl blad
        max(0, u_optymalne - a{3})
        max(0, -u_optymalne - a{4})
        u_optymalne(3+1) - a{5}
    };

    % krok 6 - liczenie gamma
    vra = [];
    for w=1:length(r)
        vra = [vra; v{w} + r{w} - a{w} ];
    end

    gamma = sqrt(sum(vra.^2, "all"))


    % krok 7
    if(gamma < epsilon)
        break
    end

    % krok 9
    if(gamma < c)
        c = alfa * c
        for w = 1:length(v )
            v{w} = a{w} - r{w};
        end
    end

    % krok 11
    if(gamma >= c)
        t = beta * t;
        for w = 1:length(v )
            v{w} = a{w} - (1/beta) * r{w};
        end
    end

end


% funkcja obliczajaca stan x na bazie sterowania u
function x = obliczanie_x(u, x0)

    % równanie stanu
    x_plus_1 = @(x,u) x + 3*u;

    % obliczenie x
    x = [x0];
```

```
    for i = [1:length(u)]
        x = [x; x_plus_1(x(i), u(i))];
    end
end

%obliczanie zmodyfikowanego wskaznika jakosci
function I_m = I_calc(u, I, R, t, x0, v)

    x = obliczanie_x(u, x0);
    x_n = x(1:end-1);

    % obliczenie I_m
    I_m = I(x_n,u);
    for j = 1:length(R)
        I_m = I_m + t(j)*R{j}(x,u,v{j});
    end

end
```