МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра методов оптимального управления

ДЕЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

Отчёт по практике

ЛОЗОВСКИЙ
Иван Иванович
студента 4 курса,
специальность «прикладная математика»

Научный руководитель:
канд. физ.-мат. наук
доцент Н.М. Дмитрук

Минск, 2021

ОГЛАВЛЕНИЕ

ГЛАВА 1

ТЕОРИЯ

Задача оптимального управления для группы q взаимосвязанных линейных нестационарных объектов управления имеет вид:

$$\sum_{i\in I} c_i' x_i(t_f) \to \min_u, \tag{1.1}$$

$$\dot{x}_i = A_i(t)x_i + \sum_{j\in I_i} A_{ij}(t)x_j + B_i(t)u_i,$$

$$x_i(t_0) = x_{i,0},\ i \in I,$$

$$\sum_{i\in I} H_i x_i(t_f) \geqslant g,$$

$$u_i(t) \in U_i, \quad t \in T, \quad i \in I.$$

В *децентрализованном* случае каждый объект управления имеет собственный регулятор. Этот локальный регулятор для каждого i-го объекта по точному состоянию $x_i^*(\tau),\ \tau \in T_h$, своего объекта и состояниям $x_j^*(\tau - h),\ \tau \in T_h,\ j \in I_i$, остальных объектов в предыдущий момент времени строит локальное управляющее воздействие $u_i^*(t),\ t \in [\tau,\ \tau + h),\ \tau \in T_h$, которое используется до тех пор, пока не будет измерено и обработано следующее состояние $x_i^*(\tau + h)$.

Для определения управления, которое будет использоваться регулятором на первом шаге, будем считать, что имеется единый центр, который до начала процесса управления способен по априорной информации вычислить оптимальную программу.

Вспомогательная задача оптимального управления для i-го регулятора для момента τ аналогично [3] имеет вид:

$$\sum_{k\in I} c_k' x_k(t_f) \to \min_{u_i}, \tag{1.2}$$

$$\dot{x}_i = A_i(t)x_i + \sum_{j\in I_i} A_{ij}(t)x_j + B_i(t)u_i, \quad x_i(\tau) = x_i^*(\tau)$$

$$\dot{x}_k = A_k(t)x_k + \sum_{j\in I_k} A_{kj}(t)x_j + f_k^0(t|\tau), \quad x_k(\tau) = x_k^0(\tau|\tau - h), \quad k \in I_i,$$

$$\sum_{k\in I} H_k x_k(t_f) \geqslant g(\tau), \quad u_i(t) \in U_i, \quad t \in T(\tau),$$

где

$$f_k^0(t|\tau) = B_k(t) u_k^0(t|\tau - h), \quad t \in T(\tau), \tau \in T_h \backslash t_0, \quad k \in I_i;$$

$$g(\tau) = \sum_{k\in I} H_k x_k^0(t_f|\tau - h) + \left[g - \sum_{k\in I} H_k x_k^0(t_f|\tau - h) \right] /q.$$

Алгоритм построения реализации оптимальной обратной связи $u_i^*(t),\ t \in T,\ i \in I$ для случая децентрализованного оптимального управления имеет вид:

1. Найти оптимальную программу $u^0(t),\ t \in T$, задачи (1.1).

2. Задать значение оптимальной обратной связи для позиции $(t_0, x^*(t_0))$:

   $$u^0(t_0,\ x^*(t_0)) = u^0(t_0),$$

   и подать на объект управления управляющее воздействие

   $$u_i^*(t) \equiv u_i^*(t_0) := u_i^0(t_0,\ x^*(t_0)), \ \ t \in [t_0,\ t_0 + h).$$

3. Измерить $x_i^*(t_0 + h)$ и вычислить $x_k^0(t_0 + h),\ k \in I_i$.

4. Положить $\tau = t_0 + h,\ x_k(\tau) = x_k^0(\tau),\ k \in I_i,\ x_i(\tau) = x_i^*(\tau)$.

5. Найти оптимальную программу $u_i^0(t|\tau,\ x_i^*(\tau),\ x^*(\tau - h)),\ t \in [\tau,\ t_f]$, задачи (1.2).

6. Задать значение оптимальной обратной связи для позиции $(\tau, x(\tau))$:

   $$u_i^0(\tau, x(\tau)) = u_i^0(\tau|\tau,\ x_i^*(\tau),\ x^*(\tau - h)),$$

   и подать на объект управления управляющее воздействие

   $$u_i^*(t) \equiv u_i^*(\tau) := u_i^0(\tau, x(\tau)), \ \ t \in [\tau,\ \tau + h).$$

7. В момент $\tau + h$ измерить $x_i^*(\tau + h)$ и вычислить $x_k^0(\tau + h),\ k \in I_i$, положить $\tau := \tau + h$, при $\tau < t_f$ вернуться к шагу 5.

Основная трудоемкость реализации данного алгоритма приходится на шаг 5, на котором необходимо быстро (за время, не превосходящее h) решать задачу оптимального управления (1.2).

Пусть $u^0(\tau|\tau,\ x^*(\tau),\ x^*(\tau-h)) = (u_i^0(\tau|\tau,\ x_i^*(\tau),\ x^*(\tau-h)),\ i\in I),\ t\in T(\tau)$ — управляющее воздействие, составленное из оптимальных програм задач (1.2), $x(t|x^*(\tau),u^0(\cdot)),\ t\in T(\tau),$ — порождаемая им траектория группы с начальными условиями $x_i(\tau)=x_i^*(\tau),\ i\in I.$

В качестве обоснования применимости указанной схемы и алгоритма для решения децентрализованной задачи (1.1) используется утверждение([2], с. 599):

Теорема 1.1 Для любого $\tau \in T_h$ управляющее воздействие $u^0(t|\tau,\ x_i^*(\tau),\ x^*(\tau-h)),\ t\in T(\tau)$, переводит группу объектов управления из состояния $x^*(\tau)$ на терминальное множество, то есть $x(t_f|x^*(\tau),u^0(\cdot))\in X_f$.

Используя элементы функциональной формы [1], задачу (1.2) можно представить в виде:

$$\sum_{s\in T_h(\tau)} c_i'(s)u_i(s)\to\min,$$

$$\sum_{s\in T_h(\tau)} D_i(s)u_i(s)\geqslant \tilde g_i(\tau),$$

$$u_{i,*}\leqslant u_i(s)\leqslant u_i^*,\quad s\in T_h(\tau).$$

Здесь $T_h(\tau)=T_h\cap[\tau,t_f]$;

$$c_i'(s)=\int\limits_s^{s+h}\psi_i'(t)B_i(t)dt,\quad D_i(s)=\int\limits_s^{s+h}\Phi_i(t)B_i(t)dt,\quad s\in T_h(\tau);$$

$$\dot\psi_i'=-\psi_i'A_i(t)-\sum_{j\in I_i}\psi_j'A_{ji}(t),\quad \psi_i(t_f)=c_i,\quad i\in I;$$

$$\tilde g_i(\tau)=-\left[g+\sum_{k\in I}g_k(\tau-h)\right]/q+g_i(\tau-h)+\Phi_i(\tau)\left(x_i^0(\tau|\tau-h)-x_i^*(\tau)\right)-$$

$$-\Phi_i(\tau-h)x_i^*(\tau-h)-D_i(\tau-h)u_i^*(\tau-h),$$

$$\Phi_k(t)=\sum_{j\in I}H_jF_{jk}(t_f,t),\quad F_{jk}(t_f,t)=(E^{A(t_f-t)})_{jk}.$$

ГЛАВА 2

ПРАКТИКА

В качестве примера рассмотрим задачу оптимального управления для объекта, состоящего из трех взаимосвязанных систем и описываемого линейными уравнениями с аддитивными возмущениями w:

$$\begin{aligned} \ddot{z}_1 &= -2kz_1 + kz_2 + u_1 + w_1, \\ \ddot{z}_2 &= -2kz_2 + kz_1 + kz_3 + u_2 + w_2, \\ \ddot{z}_3 &= -2kz_3 + kz_2 + u_3 + w_3. \end{aligned} \tag{2.1}$$

Система (2.1) описывает систему трех масс ($q = 3$, массы для простоты выбраны равными 1), соединенных одинаковыми пружинами жесткости k. Управления приложены к каждой из масс.

Управляющее воздействия ограничены по модулю

$$|u_i(t)| \leq L,\ t \in [t_0, t_f],\ i \in I = \{1,\ 2,\ 3\},$$

и в момент времени t_f требуется перевести объект (2.1) на терминальное множество, задаваемое ограничениями

$$|z_i(t_f)| \leq d_1,\ |\dot{z}_i(t_f)| \leq d_2, i \in I,$$

а критерий качества имеет вид:

$$\int\limits_{t_0}^{t_f} \sum_{i \in I} |u_i(t)|\, dt \to \min .$$

Параметры задачи выберем следующим образом:

$$k = 10,\ L = 1,\ t_0 = 0,\ t_f = 4.7;$$

$$d_i = 0.1,\ \dot{z}_i(0) = 1,\ z_i(0) = 1,\ i \in I.$$

Пусть также $N = 100$, тогда период квантования равен $h = \dfrac{1}{N} = \dfrac{1}{100}$.

Будем считать, что в конкретном рассматриваемом процессе управления движение объекта описывается дифференциальными уравнениями (2.1) с аддитивными возмущениями:

$$\begin{aligned} \ddot{z}_1^{\,*} &= -2kz_1^* + kz_2^* + u_1^* + w_1^*, \\ \ddot{z}_2^{\,*} &= -2kz_2^* + kz_1^* + kz_3^* + u_2^* + w_2^*, \\ \ddot{z}_3^{\,*} &= -2kz_3^* + kz_2^* + u_3^* + w_3^*, \end{aligned} \tag{2.2}$$

где возмущение задается следующим образом

$$w^*(t) = \begin{pmatrix} 0.1\,\cos t \\ \sin 2t \\ 0.7\,\cos 2t \end{pmatrix}, \quad t \in [t_0, t_f].$$

Отметим, что, хоть в объекте (2.2) и присутствует возмущение w^*, оптимальная обратная связь будет определяться по детерминированной задаче. Такой подход соответствует классическому. Иными словами, для отыскания оптимальной обратной связи будем использовать математическую модель (2.1).

Сформулируем задачу оптимального управления для системы дифференциальных уравнений первого порядка. Произведем замену переменных:

$$\begin{aligned} x_1 &= z_1, \qquad & x_3 &= z_2, \qquad & x_5 &= z_3, \\ x_2 &= \dot{z}_1, \qquad & x_4 &= \dot{z}_2, \qquad & x_6 &= \dot{z}_3. \end{aligned}$$

Здесь $n = 6$ — размерность состояния x; $r = q = 3$ — размерность управления; $m = 2n$ — число ограничений на состояния $x_i,\ i = 1...n$.

Тогда видно, что задача оптимального управления группой (2.1) имеет вид:

$$\int\limits_{t_0}^{t_f} \sum_{i \in I} |u_i(t)|\, dt \to \min, \tag{2.3}$$

$$\dot{x}_i = A_i x_i + \sum_{j \in I_i} A_{ij} x_j + B_i u_i,$$

$$x_i(t_0) = x_{i,0},\ i \in I,$$

$$\sum_{i \in I} H_i x_i(t_f) \leqslant g,$$

$$u_i(t) \in U_i, \quad t \in T, \quad i \in I.$$

где

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -2k & 0 & k & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ k & 0 & -2k & 0 & k & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & k & 0 & -2k & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; x_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$H = \begin{pmatrix} H_1 & H_2 & H_3 \end{pmatrix},$$

$$H_1 = \begin{pmatrix} H_p \\ 0_{8\times 2} \end{pmatrix}, H_2 = \begin{pmatrix} 0_{4\times 2} \\ H_p \\ 0_{4\times 2} \end{pmatrix}, H_1 = \begin{pmatrix} 0_{8\times 2} \\ H_p \end{pmatrix}, H_p = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix},$$

$$g = 0.1 \cdot \mathbb{1}_m,$$

где $\mathbb{1}_s$ — вектор из единиц размерности s. Также стоит отметить, что

$$B_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Из теории следует, что задача оптимального управления для i-го регулятора для момента τ имеет вид:

$$\int\limits_{t_0}^{t_f} \sum_{i \in I} |u_i(t)| \, dt \to \min, \tag{2.4}$$

$$\dot{x}_i = A_i(t)x_i + \sum_{j \in I_i} A_{ij}(t)x_j + B_i(t)u_i, \quad x_i(\tau) = x_i^*(\tau)$$

$$\dot{x}_k = A_k(t)x_k + \sum_{j \in I_k} A_{kj}(t)x_j + f_k^0(t|\tau), \quad x_k(\tau) = x_k^0(\tau|\tau - h), \quad k \in I_i,$$

$$\sum_{k \in I} H_k x_k(t_f) \leqslant g(\tau), \quad u_i(t) \in U_i, \quad t \in T(\tau),$$

где

$$f_k^0(t|\tau) = B_k(t)u_k^0(t|\tau - h), \quad t \in T(\tau), \tau \in T_h \backslash t_0, \quad k \in I_i;$$

$$g(\tau) = \sum_{k \in I} H_k x_k^0(t_f|\tau - h) + \left[g - \sum_{k \in I} H_k x_k^0(t_f|\tau - h) \right] / q.$$

Если применить замену $u_i(s) = z_i(s) - v_i(s), \quad s \in T_h(\tau)$, изменить знаки так, чтобы они соответсвовали тем, что в теории, и перевести задачу в функциональную форму, то она, согласно теории, примет следующий вид:

$$\sum_{s \in T_h(\tau)} z_i(s) + v_i(s) \to \min, \tag{2.5}$$

$$\sum_{s \in T_h(\tau)} D_i(s)(z_i(s) - v_i(s)) \geqslant \tilde{g}_i(\tau),$$

$$0 \leqslant z_i(s) \leqslant 1, \quad 0 \leqslant v_i(s) \leqslant 1, \quad s \in T_h(\tau),$$

где

$$D_i(s) = \int\limits_s^{s+h} \Phi_i(t) B_i(t) dt, \quad s \in T_h(\tau),$$

$$\tilde{g}_i(\tau) = -\left[g + \sum_{k \in I} g_k(\tau - h)\right]/q + g_i(\tau - h) + \Phi_i(\tau)\left(x_i^0(\tau|\tau - h) - x_i^*(\tau)\right) -$$

$$-\Phi_i(\tau - h)x_i^*(\tau - h) - D_i(\tau - h)u_i^*(\tau - h),$$

$$g_k(\tau) = \Phi_k(\tau)x_k^*(\tau) + \int\limits_\tau^{t_f} \Phi_k(t)B_k(t)u_k^0(t|\tau)dt.$$

Функциональную форму (2.5) также можно записать в виде, который в дальнейшем будет использоваться:

$$\sum_{s \in T_h(\tau)} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} z_i(s) \\ v_i(s) \end{pmatrix} \to \min, \tag{2.6}$$

$$\sum_{s \in T_h(\tau)} \begin{pmatrix} D_i(s) & -D_i(s) \end{pmatrix} \begin{pmatrix} z_i(s) \\ v_i(s) \end{pmatrix} \geqslant \tilde{g}_i(\tau),$$

$$0 \leqslant z_i(s) \leqslant 1, \quad 0 \leqslant v_i(s) \leqslant 1, \quad s \in T_h(\tau).$$

Для записи левой части ограничений задачи (2.6) в матричном виде в функции

```
function res = Form_Ale_dec(k)
    D_values = zeros(m, 1, N_initial);%12x1x100
    for i = 1:N_initial
        D_values(:,:,i) = Get_D_i(k,t_0 + i*h - h);
    end
```

```
    res = [];
    for i = 1:N_initial
        res = [res  D_values(:,:,i) ...
            -D_values(:,:,i)];
    end
end
```

формируется матрица `Ale_i`, которая будет иметь вид:

$$\left(D_i(t_0) \ \ -D_i(t_0) \ \ \ldots \ \ D_i(t_f - h) \ \ -D_i(t_f - h)\right). \tag{2.7}$$

Матрица `Ale_i` будет являться одним из параметров стандартной процедуры linprog для решения задач линейного программирования.

В функции

```
function u = P_dec(i, tau, g_wave_dec)
    N = round((t_f - tau)/h);%new N for new  tau
    c = ones(2*N, 1);
    ub = L*ones(2*N, 1);
    startBlockNumber = N_initial - N + 1; % ...
      startBlockNumber...N_initial -- N numbers
    Ale_dec_i = ...
      GetLayer(i,q,Ale_dec(:,2*(startBlockNumber ...
      - 1) + 1 : 2*N_initial));
    Opt=optimset('TolFun',1e-9,'TolX',1e-9);
    z_and_v = linprog(c,(-1)*Ale_dec_i,...
            (-1)*g_wave_dec,[],[],zeros(2*N, ...
              1),ub,Opt);
    zv = reshape(z_and_v, 2, N);
    u  = zv(1,:) - zv(2,:); % 1 /times N
end
```

определяется оптимальная программа для времени (`tau`).

Результатом стандартной процедуры linprog будет вектор `z_and_v`, который можно записать в виде:

$$\begin{pmatrix} z_i(\tau) \\ v_i(\tau) \\ z_i(\tau+h) \\ v_i(\tau+h) \\ \vdots \\ z_i(t_f-h) \\ v_i(t_f-h) \end{pmatrix}.$$

В конце метода вектор преобразуется в позиционное решение `u`.

В функции `P_dec` используется метод `GetLayer(i,q, matrix)`:

```
% if the source has 21 lines
% GetLayer(2,3, source) returns lines 8-14 ...
    (middle layer)
function res = GetLayer(layerNum, layersAmount, ...
    source)
    [ysize,] = size(source);
    layerHeight = ysize/layersAmount;
    from = (layerNum - 1)*layerHeight + 1;
    to = from + layerHeight - 1;
    res = source(from:to,:);
end
```

Данный метод вернет i-ый "слой"матрицы из q слоёв. Он часто используется в коде.

Функция `P_dec` вычисляется каждым регулятор на каждом временном отрезке. Для этого регулятор перед ее вызовом должен посчитать величины $g_k(\tau-h), k \in I, \tilde{g}_i(\tau), x_i(\tau|\tau-h)$. Ниже представлен код по их вычислению, то есть для каждой из указанных.

Вычисление величин $g_k(\tau-h), k \in I$:

```
function g_k = Get_g_k(k, tau, x, u0)
    N = round((t_f - tau)/h);%new N for new  tau ...
        (and also (t_f - t_0)/h = N_initial)
    sum = zeros(m, 1);%12x1
    %get N corresp tau
    N_start = round((tau - t_0)/h) + 1;
    d_h_values = GetLayer(k,q,d_h_matrix(:, ...
        N_start : N_initial));
```

```
    for i = 1:N
        sum = sum + d_h_values(:,i) * u0(i);
    end
    g_k = Get_Phi_k(k, tau)*x + sum;
end
```

Тут в переменной `d_h_matrix` содержатся величины $D_i(s), s \in T_h, i \in I$. Переменная вычисляется до начала работы регуляторов.

Вычисление величин $\tilde{g}_i(\tau)$:

```
function res = Get_g_wave_dec(i, tau, ...
   g_values_prev, x0i, X_dec, U_dec)%
    N = round((tau - t_0)/h) + 1;
    sum = zeros(m,1);
    for j=1:q
        sum = sum + g_values_prev(:,j);
    end
    res = (g_new - sum)/q + g_values_prev(:,i) + ...
        Get_Phi_k(i,tau) * ...
        (x0i - GetLayer(i,q,X_dec(:,N))) - ...
        Get_Phi_k(i,tau - ...
          h)*GetLayer(i,q,X_dec(:,N - 1)) - ...
        Get_D_i(i,tau - h)*U_dec(i,N - 1);
end
```

Тут `g_values_prev` $\equiv \big(g_1(\tau-h) \quad g_2(\tau-h) \quad g_3(\tau-h)\big)$, `x0i` $\equiv x_i(\tau|\tau-h)$.

Вычисление величин $x_i(\tau|\tau-h)$:

```
function state = guessNextState(tau, curState, u)
    t_begin = tau;
    t_end = tau + h;
    x0 = curState;
    begin_end_steps = trajectoryNoW(x0, t_begin, ...
      t_end, u);
    state = begin_end_steps(:,2);
end

function x = trajectoryNoW(x0, t_begin, t_end, u)
    N = round((t_end - t_begin)/h);
    x = zeros(n, N);
    x(:,1) = x0;
```

```
    for j = 1:N
        curr = t_begin + (j-1)*h;
        next = t_begin + j*h;
        x(:, j+1) = F(h) * x(:,j) + integral(@(t) ...
          F(next - t)*b,curr, next, ...
          'ArrayValued', ...
          true,'RelTol',0,'AbsTol',1e-12)*u(:, j);
    end
end
```

Для вычисления этих величин применяется формула Коши (возмущения не учитываются, потому что они неизвестны регулятору).

Таким образом, задачу решает следующий код:

```
    [U_0, X_0] = ...
      GetProgramSolutionAndTrajectory(); ...
      %size(U_0) = 3x100%size(X_0) = 6x101
    U_dec = U_0;
    X_dec = X_0;
    guessedState = guessNextState(t_0, x_0, ...
      U_0(:,1));
    g_values_prev = zeros(m, q);
    for i = 1:q
        g_values_prev(:,i) = Get_g_k(i, t_0, ...
          GetLayer(i,q,X_dec(:, 1)) , ...
          GetLayer(i,q,U_dec));
    end
    g_values = g_values_prev;%just init somehow
    tau = t_0 + h;
    for kk = 2:N_initial
        X_dec(:, kk) = getNextState(tau - ...
          h,X_dec(:, kk - 1),U_dec(:, kk - 1));
        for i = 1:q
            x0i = GetLayer(i,q,guessedState);
            g_wave_dec = Get_g_wave_dec(i, tau, ...
              g_values_prev, x0i, X_dec, U_dec);
            U_dec(i, kk : N_initial) = P_dec(i, ...
              tau, g_wave_dec);
            g_values(:,i) = Get_g_k(i, tau, ...
              GetLayer(i,q,X_dec(:, kk)) , ...
```

```
                U_dec(i,kk:N_initial));
        end
        g_values_prev = g_values;
        guessedState = guessNextState(tau, ...
            X_dec(:,kk), U_dec(:,kk));
        tau = tau + h;
    end
    X_dec(:, N_initial + 1) = ...
        getNextState(tau,X_dec(:,N_initial ...
        ),U_dec(:, N_initial));
```

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Балашевич, Н. В. Численные методы программной и позиционной оптимизации линейных систем управления /Н. В. Балашевич, Р. Габасов, Ф. М. Кириллова // Журнал вычислительной математики и математической физики. – 2000. – Т. 40, № 6. – С. 838-859.

2 Габасов, Р. Оптимальное децентрализованное управление группой динамических объектов / Р. Габасов, Н.М. Дмитрук, Ф.М. Кириллова // Журнал вычислительной математики и математической физики. – 2008. –Т. 48, № 4. – С. 593-609.

3 Габасов, Р. Оптимальное децентрализованное управление группой динамических объектов / Р. Габасов, Н.М. Дмитрук, Ф.М. Кириллова // Журнал вычислительной математики и математической физики. – 2008. –Т. 48, № 4. – С. 58-74.