АКАДЕМИЯ НАУК СССР
УРАЛЬСКИЙ НАУЧНЫЙ ЦЕНТР
ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ

АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ ИГР

(Материалы по математическому обеспечению ЭВМ)

Свердловск 1984

АКАДЕМИЯ НАУК СССР
УРАЛЬСКИЙ НАУЧНЫЙ ЦЕНТР
ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ

АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ ИГР

(материалы по математическому обеспечению ЭВМ)

Свердловск, 1984

УДК 519.9

Алгоритмы и программы решения линейных дифференциальных игр (материалы по математическому обеспечению ЭВМ). Свердловск: УНЦ АН СССР, 1984.

Брошюра содержит набор алгоритмов и программ, предназначенных для решения некоторых типов линейных дифференциальных игр.

Материал рассчитан на вычислителей, инженеров и научных работников, интересующихся численными методами теории управления и теории дифференциальных игр.

Ответственные редакторы -
доктор физ.-мат.наук А.И.Субботин
кандидат физ.-мат.наук В.С.Пацко

$A\frac{20204-214(83)}{055(02)\ 7}$ БО

М.А.Зарх

ПОСТРОЕНИЕ МНОЖЕСТВА ПОЗИЦИОННОГО ПОГЛОЩЕНИЯ В ДИФФЕРЕНЦИАЛЬНОЙ ИГРЕ ВТОРОГО ПОРЯДКА

I. Постановка задачи

Рассмотрим линейную антагонистическую дифференциальную игру двух лиц

$$\dot{x} = A(v^{(1)})x + Ev^{(1)} + Bu + Cv^{(2)} + F \qquad (I)$$

$$x \in R^2, \; v^{(1)} \in D, \; u \in P, \; v^{(2)} \in Q, \; x(\vartheta) \in M$$

с фиксированным моментом окончания ϑ и фазовыми ограничениями $\Phi(t)$. Здесь u – управляющий параметр первого игрока, $v^{(1)}$ и $v^{(2)}$ – управляющие параметры второго игрока. Множества P, Q – выпуклые многогранники из R^p и R^{q_2}, множество D – дискретный набор векторов $q^{(1)}, q^{(2)}, \ldots, q^{(m)}$ из R^{q_1}. При любом $t \leq \vartheta$ множество $\Phi(t)$ – выпуклый многоугольник в R^2.

Цель первого игрока – привести движение системы (I) в момент ϑ на выпуклое множество M с соблюдением фазовых ограничений. Интересы второго игрока противоположны. Обозначим через W множество всех начальных позиций, из которых первый игрок успешно решает свою задачу при помощи позиционного способа управления. Следуя [2, 3], назовем W множеством позиционного поглощения, а множество $W(t) = \{x \in R^2 : (t,x) \in W\}$ – сечением W в момент t.

Требуется построить при помощи ЭВМ набор сечений $W(t_i)$ для моментов времени $t_i = \vartheta - i\Delta$, $i = \overline{1,n}$, из заданного промежутка $[t_*, \vartheta]$, $t_n = t_*$. Основными входными данными описываемой программы, помимо шага $\Delta > 0$ и моментов t_*, ϑ, являются: функции $v^{(1)} \to A(v^{(1)})$, $t \to \Phi(t)$, реализуемые в виде программ, матрицы B, C, E, вектор F, множества P, Q, D, M.

2. Описание алгоритма

Построение сечений $W(t_i)$ – рекуррентный процесс: для нахождения $W(t_{i+1})$ необходимо, чтобы было построено $W(t_i)$. При этом $W(t_0) = W(\vartheta) = \Phi(\vartheta) \cap M$. Ниже описывается алгоритм построения $W(t_{i+1})$ на основе $W(t_i)$.

Зафиксируем $v^{(1)} = q^{(k)}$ и обозначим через $\bar{F}_k$ вектор $Eq^{(k)} + F$. Рассмотрим дифференциальную игру

$$\dot{x} = A(q^{(k)})x + Bu + Cv + \bar{F}_k \qquad (2)$$

$$u \in P, \quad v \in Q, \quad x(t_i) \in W(t_i)$$

с фиксированным моментом окончания t_i и без фазовых ограничений. Сделаем замену

$$z(t) = X^{(k)}(t_i, t)x(t) + \int_t^{t_i} X^{(k)}(t_i, \tau)\bar{F}_k \, d\tau$$

($X^{(k)}(t_i, t)$ - фундаментальная матрица Коши системы $\dot{x} = A(q^{(k)})x$) и перейдем к дифференциальной игре

$$\dot{z} = u + v$$

$$u \in P_i^{(k)}(t) = X^{(k)}(t_i, t)BP, \quad v \in Q_i^{(k)}(t) = X^{(k)}(t_i, t)CQ \qquad (3)$$

$$z(t_i) \in W(t_i)$$

Пусть $\tilde{W}^{(k)}(t_{i+1})$ - сечение множества позиционного поглощения для игры (3). Множество $\tilde{W}^{(k)}(t_{i+1})$ строится при помощи подпрограммы, описанной в работе [1] настоящего сборника. Сечение множества позиционного поглощения $W^{(k)}(t_{i+1})$ для игры (2) находится по формуле

$$W^{(k)}(t_{i+1}) = [X^{(k)}(t_i, t_{i+1})]^{-1}(\tilde{W}^{(k)}(t_{i+1}) - \int_{t_{i+1}}^{t_i} X^{(k)}(t_i, \tau)\bar{F}_k\, d\tau)$$

Сечение $W(t_{i+1})$ находим по формуле

$$W(t_{i+1}) = \Phi(t_{i+1}) \cap (\bigcap_{k=1}^{m} W^{(k)}(t_{i+1}))$$

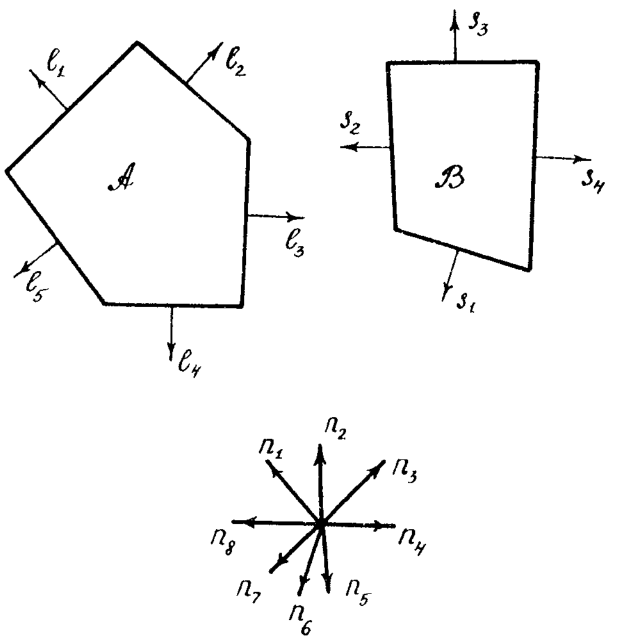
При численном построении множеств $\tilde{W}^{(k)}(t_{i+1})$ игра (3) подменяется кусочно-постоянной по времени с шагом h аппроксимирующей игрой [1] . Сечения $\tilde{W}^{(k)}(t_{i+1})$, а стало быть и $W^{(k)}(t_{i+1})$ при этом получаются выпуклыми многоугольниками.

Таким образом, для построения $\{W(t_i)\}$ необходимо решить следующую задачу. Даны выпуклые многоугольники $\mathcal{A}$ и $\mathcal{B}$ из R^2 с m_1 и m_2 вершинами. Требуется найти выпуклый многоугольники $\mathcal{A} \cap \mathcal{B}$.

Пусть многоугольники $\mathcal{A}$ и $\mathcal{B}$ заданы вершинами a_i , $i=\overline{1,m_1}$ и b_j , $j=\overline{1,m_2}$, упорядоченными по часовой стрелке. Пусть ℓ_i , $i=\overline{1,m_1}$ и s_j , $j=\overline{1,m_2}$ – упорядоченные наборы единичных внешних нормалей к сторонам выпуклых многоугольников $\mathcal{A}$ и $\mathcal{B}$ (рис. I). Будем говорить, что вектор e лежит левее (правее) вектора d , если поворот вектора e до сонаправленности с d по наименьшему углу осуществляется по часовой стрелке (рис. 2) (против часовой стрелки).

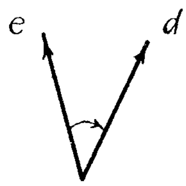
Определим упорядоченный по часовой стрелке набор $\{n_k\}$ всех единичных внешних нормалей, снятых с многоугольников $\mathcal{A}$ и $\mathcal{B}$ (рис. 1). Будем предполагать, что в случае совпадения каких-либо двух нормалей, они представлены в наборе одним вектором. Таким образом, общее число $\bar{m}$ векторов в наборе удовлетворяет неравенству $\bar{m} \leq m_1 + m_2$, и равенство реализуется лишь в том случае, когда среди нормалей ℓ_i , s_j нет совпадающих.

Найдем сначала такой вектор s_r , что ℓ_1 левее s_r и нет других векторов s_j , лежащих левее s_r и правее ℓ_1 . Перенумеруем набор $\{s_j\}$: $\bar{s}_1 = s_r$, $\bar{s}_2 = s_{r+1}, \ldots,$ $\bar{s}_{m_2-r+1} = s_{m_2}$, $\bar{s}_{m_2-r+2} = s_1$, $\ldots$, $\bar{s}_{m_2} = s_{r-1}$. Допустим, что проделано m шагов и составлен упорядоченный набор n_k , $k=1,2,\ldots,m$, содержащий все векторы из совокупности $\ell_1, \ldots, \ell_{i-1}$, $\bar{s}_1, \ldots, \bar{s}_{j-1}$. Рассмотрим пару векторов ℓ_i , $\bar{s}_j$. Если ℓ_i левее $\bar{s}_j$, то



ℓ_1
ℓ_2
ℓ_3
ℓ_4
ℓ_5
$\mathcal{A}$
s_1
s_2
s_3
s_4
$\mathcal{B}$
n_1
n_2
n_3
n_4
n_5
n_6
n_7
n_8

Рис. 1



e
d

Рис. 2

полагаем $n_{m+1} = \ell_i$ и на следующем шаге рассмотрим пару $\ell_{i+1}, \bar{s}_j$. Если ℓ_i и $\bar{s}_j$ совпадают, то $n_{m+1} = \ell_i$ и на следующем шаге рассматривается пара ℓ_{i+1}, $\bar{s}_{j+1}$. В случае, когда ℓ_i правее $\bar{s}_j$, полагаем $n_{m+1} = \bar{s}_j$ и переходим к рассмотрению пары $\ell_i, \bar{s}_{j+1}$. На первом шаге рассматривается пара $\ell_1, \bar{s}_1$. Действуя по описанному выше правилу, придем к одной из двух ситуаций: а) выполнено m шагов и на следующем шаге выбрана пара ℓ_{m_1}, $\bar{s}_j$, причем ℓ_{m_1} левее $\bar{s}_j$; в этом случае положим $n_{m+1} = \ell_{m_1}$ $n_{m+2} = \bar{s}_j, \dots, n_{\bar{m}} = \bar{s}_{m_2}$; в) выполнено m шагов, на следующем шаге выбрана пара $\ell_i, \bar{s}_{m_2}$ и $\bar{s}_{m_2}$ левее ℓ_i; в этом случае положим $n_{m+1} = \bar{s}_{m_2}$, $n_{m+2} = \ell_i, \dots,$ $n_{\bar{m}} = \ell_{m_1}$. Таким образом, получим набор n_k, $k = \overline{1,\bar{m}}$.

Каждому $k = \overline{1,\bar{m}}$ поставим в соответствие число γ_k, представляющее из себя минимум из значений двух опорных функций $\rho(n_k, \mathcal{A})$, $\rho(n_k, \mathcal{B})$ на векторе n_k. Зададим на плоскости кусочно-линейную функцию f. Для любого $n \in R^2$ однозначно определим числа $k \in \{1, 2, \dots, \bar{m}\}$, $\alpha \geq 0$, $\beta \geq 0$, такие, что $n = \alpha n_k + \beta n_{k+1}$. Положим $f(n) = \alpha \gamma_k + \beta \gamma_{k+1}$. Опорная функция пересечения $\mathcal{A} \cap \mathcal{B}$ есть выпуклая оболочка $co\, f$ введенной функции.

Алгоритм овыпукления функции заключается в последовательном изменении функции f, которое производится на каждом шаге в результате проверки функции на локальную выпуклость в окрестности очередного вектора из набора. На первом шаге рассматриваются векторы $n_0 = n_{\bar{m}}$, n_1, n_2 вместе со значениями функции f на этих векторах. Анализируя взаимное расположение на плоскости прямых $n_i' x = f(n_i)$, $i = 0,1,2$ соответствующих этим векторам и значениям функции, делаем заключение о поведении функции в окрестности вектора n_1 . Если функция f выпукла в окрестности n_1 , то на следующем шаге рассматриваем векторы n_1 , n_2 , n_3 . При этом функцию f оставляем без изменения. Если функция не выпукла в окрестности вектора n_1 , и векторы n_0 , n_1 , n_2 не принадлежат одной полуплоскости (т.е. n_0 правее n_2), то $\mathcal{A} \cap \mathcal{B} = \varnothing$. В случае, когда функция не выпукла в окрестности вектора n_1 и векторы n_0 , n_1 , n_2 принадлежат одной полуплоскости, то n_1 из набора выбрасывается, а функция f заменяется на непрерывную функцию, совпадающую с f вне конуса, порожденного векторами n_0 , n_2 , и линейную в этом конусе. На следующем шаге проверяем локальную выпуклость новой функции на тройке векторов n_0 , n_2 , n_3 . Действуя по описанному выше правилу, после $\bar{m}$ -го шага получим новый набор векторов $\tilde{n}_i$, $i = \overline{1,\tilde{m}}$, и новую кусочно-линейную функцию $\tilde{f}$, заданную своими значениями на векторах $\tilde{n}_i$. Если функции f и $\tilde{f}$ не совпадают, т.е. хотя бы при одной локальной проверке имела место невыпуклость, то повторяем описанную выше последовательность шагов, взяв в качестве исходной полученную функцию $\tilde{f}$. Так действуем до тех пор пока в очередной последовательности шагов не получим локальную выпуклость функции на каждом шаге. При этом найденная функция $f^* = co\, f$ есть опорная функция пересечения $\mathcal{A} \cap \mathcal{B}$.

3. Описание программы

Переменные, используемые в программе

$IV1$ – размерность вектора управления $v^{(1)}$,
MB – размерность вектора управления u,
MC – размерность вектора управления $v^{(2)}$,
KB – количество вершин многогранника P,
KC – количество вершин многогранника Q,
$NV1$ – количество векторов в множестве D,

$C11(10)$ – одномерный массив, содержащий первую строку матрицы E,

$C12(10)$ – одномерный массив, содержащий вторую строку матрицы E,

$B(2,10)$ – двумерный массив, в котором находится матрица B системы (I),

$C(2,10)$ – двумерный массив, в котором находится матрица C системы (I),

$FF(2)$ – одномерный массив, содержащий свободный член F системы (I),

$V1(5,20)$ – двумерный массив, содержащий координаты векторов из набора D. Координатами J-го вектора из D являются числа $V1(1,J), \dots, V1(IV1,J)$

$P(5,20)$ – двумерный массив, содержащий координаты вершин многогранника P. Координатами J-ой вершины являются числа $P(1,J), \dots, P(MB,J)$.

$Q(5,20)$ – двумерный массив, содержащий координаты вершин многогранника Q ,

$PM1(1000)$ – одномерный массив первых координат вершин многоугольника M , упорядоченных по часовой стрелке,

$PM2(1000)$ – одномерный массив вторых координат многоугольника M ,

NPM – количество вершин M ,

$DLINA$ – момент окончания игры ϑ ,

DEL – число Δ ,

HH – шаг h ,

$PIC(50)$ – одномерный массив моментов обратного времени. Сечения, соответствующие этим моментам выдаются на печать,

$IWMAX$ – количество выводимых на печать сечений.

В процессе выполнения программы вычисляются:

TAY – текущее обратное время, равное $\vartheta - t$,

$GB1(1000)$ – одномерный массив первых координат вершин многоугольника $W(t_i)$,

$GB2(1000)$ – одномерный массив вторых координат вершин многоугольника $W(t_i)$,

MBG – количество вершин многоугольника $W(t_i)$.

На печать выводятся MBG , $GB1$, $GB2$ в моменты обратного времени $PIC(1)$, $PIC(2)$, ... , $FIC(IWMAX)$. Если все эти сечения непусты, то после вывода последнего из них печатается

ORDINARY FINISH T = ...

Если оказывается, что в некоторый момент сечение множества позиционного поглощения пусто, то после вывода последнего непустого сечения, соответствующего набору PIC, печатается одна из следующих записей:

$$INTERSECTION\ \ WITH\ \ FO(T)\ IS\ EMPTY\ T=\ldots,$$

если при $t_i = T$ пересечение $\Phi(t_{i+1}) \cap W^{(1)}(t_{i+1})$ пусто;

$$EMPTINESS\ \ OF\ \ BRIDGE\ \ N=\ldots\ T=\ldots$$

если $\tilde{W}^{(N)}(t_{i+1}) = \emptyset$;

$$EMPTINESS\ \ OF\ INTERSECTION$$

$$WITH\ \ BRIDGE\ \ N=\ldots\ \ T=\ldots$$

если $\Phi(t_{i+1}) \cap W^{(1)}(t_{i+1}) \cap \ldots \cap W^{(N)}(t_{i+1}) = \emptyset$

Порядок ввода данных

Числовой материал вводится с перфокарт в следующем порядке:

1. $IV1$, $NV1$, MB, KB, MC, KC - по формату $6I3$,
2. $C11$ - по формату $10F7.3$,
3. $C12$ - по формату $10F7.3$,
4. B - по формату $10F7.3$. Элементы матрицы B пробиваются на перфокартах по строкам, т.е. в следующем порядке $B(1,1), B(1,2), \ldots, B(1,MB), B(2,1), B(2,2), \ldots, B(2,MB),$
5. $C2$ - по строкам, по формату $10F7.3$,

6. FF – по формату $2F7.3$,
7. $V1$ – по строкам, по формату $10F7.3$,
8. P – по строкам, по формату $10F7.3$,
9. Q – по строкам, по формату $10F7.3$,
10. $DLINA$ – по формату $F7.3$,
11. DEL – по формату $F7.3$,
12. HH – по формату $F7.3$,
13. $IWMAX$ – по формату $I3$,
14. PIC – по формату $10F7.3$.

Целевое множество задается при помощи подпрограммы $PAZB$. Обращение к ней имеет вид

```
CALL PAZB(PM1,PM2,NPM)
```

Последовательность операторов подпрограммы $PAZB$ должна быть следующей:

```
SUBROUTINE PAZB(PM1,PM2,NPM)
DIMENSION PM1(300),PM2(300)
. . .
RETURN
END
```

В ходе выполнения операторов, обозначенных тремя точками, в массивы $PM1$, $PM2$ должны быть занесены первые и вторые координаты вершин многоугольника M . Вершины M должны быть упорядочены по часовой стрелке. Целая переменная NPM должна стать равной числу вершин M .

Функция $v^{(1)} \to A(v^{(1)})$ реализована в виде подпрограммы $AOTV$, обращение к которой имеет вид

```
CALL AOTV(IV1,SV,A)
```

Подпрограмма $AOTV$ должна иметь следующую форму

```
SUBROUTINE AOTV(N,V,A)
DIMENSION V(5), A(2,2)
. . .
RETURN
END
```

Здесь

N – входной параметр, размерность вектора $v^{(1)}$,

V – входной параметр, массив координат вектора $v^{(1)}$,

A – двумерный массив, в который в результате выполнения операторов, обозначенных тремя точками, должна быть занесена матрица $A(v^{(1)})$.

Отображение $t \to \Phi(t)$, которое каждому моменту t ставит в соответствие выпуклый многоугольник на плоскости, реализовано посредством подпрограммы FO, обращение к которой имеет вид

```
CALL FO (TAУ, F01, F02, NFO)
```

Подпрограмма должна быть написана в следующей форме

```
SUBROUTINE FO(TAY,F01,F02,NFO)
DIMENSION F01(1000),F02(1000)
. . .
RETURN
END
```

Здесь

TAY – входной параметр, обратное время,

$F01$, $F02$, NFO – выходные параметры.

В результате выполнения операторов, обозначенных тремя точками, в массивы $F01$, $F02$ должны быть занесены первые и вторые координаты вершин многоугольника $\Phi(\vartheta - TAY)$, упорядоченных по часовой стрелке. Число NFO должно равняться количеству вершин многоугольника $\Phi(\vartheta - TAY)$.

Подпрограммы

В процессе работы головная программа P обращается к подпрограммам $AOTV$, FO , $CLOI$, PBM . О первых двух уже было сказано ранее. Поясним назначение остальных и дадим их краткое описание.

Подпрограмма $CLOI$ строит сечение множества позиционного поглощения для игры (3). Обращение к ней осуществляется посредством выполнения оператора

```
CALL CLOI
```

Подпрограмма не имеет параметров, она обменивается информа - цией с головной программой через общие блоки памяти. Перед обращением к $CL01$ в массивах $GB1$, $GB2$ должны находиться первые и вторые координаты вершин целевого множе-ства для игры (3). Целая переменная MG должна равняться числу вершин. После работы подпрограммы $CL01$ в массивах $GB1$, $GB2$ находятся первые и вторые координаты вершин построенного сечения множества позиционного поглощения. Пере-менная MG равна количеству этих вершин.

Подпрограмма PBM находит пересечение двух выпуклых многоугольников $\mathcal{A}$, $\mathcal{B}$. Обращение к подпрограмме PBM имеет вид

$$CALL\ PBM(X1,X2,N,Y1,Y2,M,P1,P2,NP)$$

Входные параметры: $X1$, $X2$ - первые и вторые координаты вершин выпуклого многоугольника $\mathcal{A}$, упорядоченных по ча-совой стрелке; N - количество вершин $\mathcal{A}$; $Y1$, $Y2$ - первые и вторые координаты упорядоченных по часовой стрелке вершин выпуклого многоугольника $\mathcal{B}$, M - количество вершин $\mathcal{B}$. Выходные параметры: $P1$, $P2$ - первые и вто-рые координаты вершин многоугольника $\mathcal{A} \cap \mathcal{B}$, NP - коли-чество вершин $\mathcal{A} \cap \mathcal{B}$.

В процессе выполнения подпрограмма PBM обращается к подпрограмме COF при помощи оператора

$$CALL\ COF(A1,A2,P,B1,B2,Q,NA,NB)$$

Подпрограмма строит выпуклую оболочку кусочно-линейной функ -

ции. f . Входные параметры $A1$, $A2$ – первые и вторые координаты векторов, задающих разбиение плоскости на конусы линейности функции f , переменная NA – количество векторов. Массив P содержит значения функции f на этих векторах. Выходные параметры: массивы $B1$, $B2$, Q и число NB определяют функцию $f^* = co f$.

Подпрограмма COF обращается к подпрограмме $CHECK$ при помощи оператора

```
CALL CHECK(A1,A2,B1,B2,C1,C2,A,B,C,IB)
```

Подпрограмма проверяет условие выпуклости функции на тройке векторов $(A1,A2)$, $(B1,B2)$, $(C1,C2)$. Числа A B , C – соответствующие этим векторам значения функции. Если условие выпуклости выполняется, то $IB = 0$, в противном случае $IB = 1$.

Особенности программы

1. Множество $W(t_i)$ в программе считается пустым не только в тех случаях, когда оно действительно пусто, но и в тех, когда оно является точкой.

2. В программе $CLOI$ задается целый параметр MDL . Если на каком-то шаге оказывается, что число всех нормалей, снятых с многоугольников $W(t_i)$, $P(t_i)$, $Q(t_i)$ превышает MDL , то многоугольник $W(t_i)$ заменяется для последующих вычислений на приближенный с меньшим числом сторон.

3. При обращении к подпрограмме PBM общее количество нормалей, пересекаемых многоугольников не должно превосходить

1000.

Контрольный пример

Дифференциальная игра второго порядка

$$\dot{x}_1 = x_2 + v$$

$$\dot{x}_2 = -\alpha x_1 + u$$

$$|u| \leq 1, \ |v| \leq 0.2, \ \alpha \in \{0.1; 0.2\}, \ \vartheta = 8,$$

$$M = \{(x_1, x_2) : |x_1| \leq 1, \ |x_2| \leq 1\},$$

$$\Phi(t) = \{(x_1, x_2) : |x_2| \leq 1.5\},$$

(параметром α распоряжается второй игрок) во введенных обозначениях будут иметь вид

$$\dot{x} = A(v^{(1)})x + Bu + Cv^{(2)} + Ev^{(1)} + F$$

$$u \in P, \ v^{(2)} \in Q, \ v^{(1)} \in D, \ x(\vartheta) \in M,$$

где

$$A(v^{(1)}) = \begin{bmatrix} 0 & 1 \\ -v^{(1)} & 0 \end{bmatrix}, \ E = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ B = C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$F = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ D = \{0.1, 0.2\},$$

$$P = \{(x_1, x_2) : x_1 = 0, \ |x_2| \leq 1\},$$

$$Q = \{(x_1, x_2) : |x_1| \leq 0.2, \ x_2 = 0\}, \ M = \{(x_1, x_2) :$$

$$|x_1| \leq 1, \ |x_2| \leq 1\}, \ \vartheta = 8, \ \Phi(t) = \{(x_1, x_2) : |x_2| \leq 1{,}5\}.$$

Требуется построить сечения $W(t_i)$ множества позиционного поглощения для моментов $t_1 = 7.8$, $t_2 = 6$, $t_3 = 5$, $t_4 = 3$, $t_5 = 1$. Для построения множеств положим $\Delta = 0,1$. На рис.3

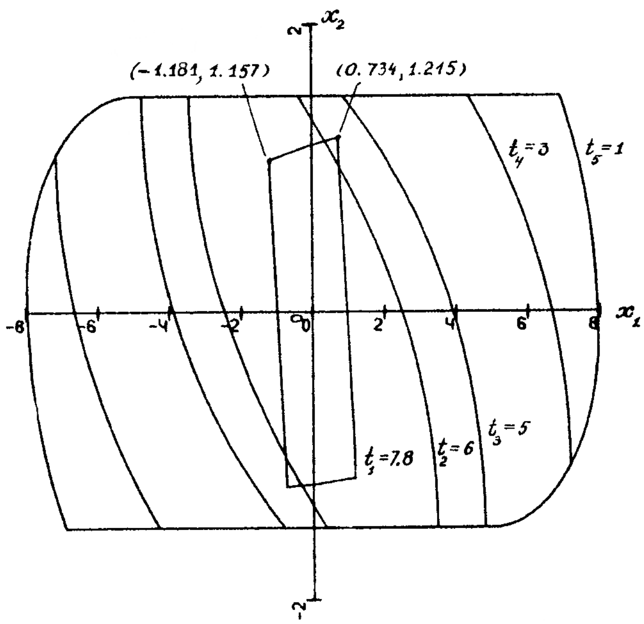
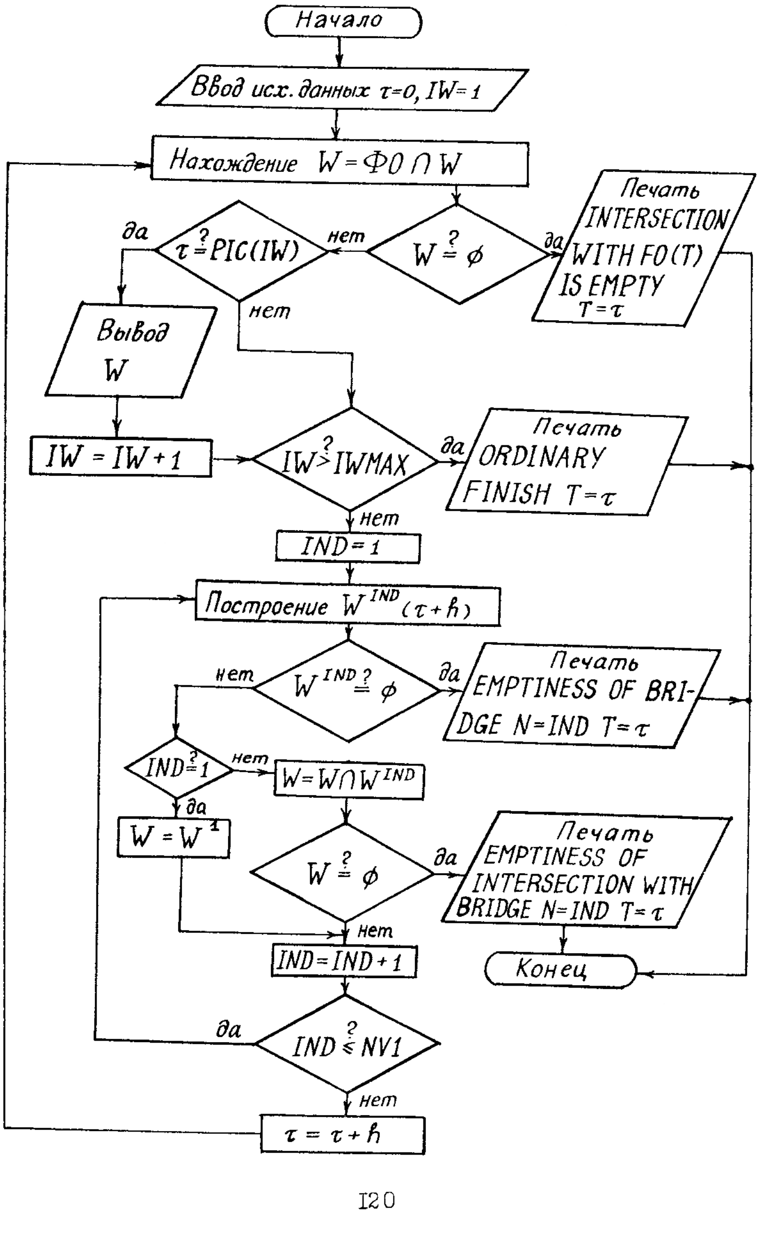


Рис. 3.

изображены сечения $W(t_i)$. Приведены также координаты характерных угловых точек.

Блок-схема



Начало
Ввод исх. данных τ=0, IW=1
Нахождение W=Φ0 ∩ W
W ≟ ∅
да
Печать INTERSECTION WITH FO(T) IS EMPTY T=τ
нет
τ ≟ PIC(IW)
да
Вывод W
нет
IW=IW+1
IW >? IWMAX
да
Печать ORDINARY FINISH T=τ
нет
IND=1
Построение W^IND (τ+h)
W^IND ≟ ∅
да
Печать EMPTINESS OF BRI-DGE N=IND T=τ
нет
IND ≟ 1
нет
W=W∩W^IND
да
W=W^1
W ≟ ∅
да
Печать EMPTINESS OF INTERSECTION WITH BRIDGE N=IND T=τ
нет
IND=IND+1
Конец
IND ≤? NV1
да
нет
τ=τ+h

ТЕКСТ ПРОГРАММЫ

```
      PROGRAM P
      DIMENSION A(10,10),
     C11(10),C12(10),
     C2(2,10),B(2,10),F(2)
      DIMENSION V1(5,20),
     Q(5,20),P(5,20),SV(5)
      DIMENSION FO1(1000),
     FO2(1000),PM1(1000),
     PM2(1000)
      DIMENSION GB1(1000),
     GB2(1000),FF(2)
      DIMENSION BG1(1000),
     BG2(1000)
      DIMENSION PIC(50)
      COMMON/AP/A/CICT/B,C2,
     FF,P,Q,MB,MC,KB,KC,MG
      COMMON/OP/DEL,HH
      COMMON/DF/IKT,JKT,N
      COMMON/TT/TT1,TT2,TT3,
     TT4
      COMMON/G/GB1,GB2
      COMMON/HOP/FO1,FO2,
     F1(1000)
      COMMON/PAB/RN1(1000),
     RN2(1000),AF(1000)
      COMMON/ICT/ISTOP
      COMMON//TCLOI,LOP
C ВВОД ДАННЫХ
      READ1,IV1,NV1,MB,KB,
     MC,KC
      READ2,(C11(I),I=1,IV1)
      READ2,(C12(I),I=1,IV1)
      READ2,((B(I,J),J=1,MB)
     ,I=1,2)
      READ2,((C2(I,J),J=1,MC
     ),I=1,2)
      READ2,(FF(I),I=1,2)
      READ2,((V1(I,J),J=1,
     NV1),I=1,IV1)
      READ2,((P(I,J),J=1,KB)
     ,I=1,MB)
      READ2,((Q(I,J),J=1,KC)
     ,I=1,MC)
      READ2,DLINA
      READ2,DEL
      READ2,HH
      READ1,IWMAX
      READ2,(PIC(J),J=1,
     IWMAX)
    1 FORMAT(6I3)
    2 FORMAT(10F7.2)
      TAY=0.
      CALL PAZB(PM1,PM2,NPM)
    7 CALL FO(TAY,FO1,FO2,
     NFO)
      CALL PBM(FO1,FO2,NFO,
     PM1,PM2,NPM,BG1,BG2,MBG)
      IF(ISTOP.EQ.1)GOTO120
      IF(TAY.LE.PIC(IW)-1.
     E-5)GOTO10
      PRINT5,TAY,CE
      PRINT6,MBG
      PRINT7,(BG1(J1),BG2(J1)
     ,J1=1,MBG)
    5 FORMAT(1X,3HTAY,F7.3,5X
     ,2HCE,F7.3)
    6 FORMAT(4X,I5)
    7 FORMAT(5(4X,F7.3,2X,
     F7.3))
```

На страницах 121 - 126 идёт текст программы на Фортране

```
50 RETURN
   END
   SUBROUTINE CHECK(A1 ,A2,
   B1,B2,C1,C2,A,B,C,IBYB)
   COMMON/ICT/ICTOP
   IBYB=0
   ICTOP=0
   D=A1*B2-A2*B1
   D1=A*B2-B*A2
   D2=A1*B-B1*A
   T1=D1/D
   T2=D2/D
   R=T1*C1+T2*C2
   IF(R+1.E-6.LE.C)GOTO10
   IBYB=1
   R=A1*C2-A2*C1
   IF(R+1.E-6.GE.0.)ICTOP
   =1
10 RETURN
   END
```

Л и т е р а т у р а

I. Исакова Е.А., Логунова Г.В., Пацко В.С. Построение стабильных мостов в линейной дифференциальной игре с фиксированным моментом окончания. – Наст. сборник.

2. Красовский Н.Н., Субботин А.И. Позиционные дифференциальные игры. – М.: Наука, I974. – 456 с.

3. Субботин А.И., Ченцов А.Г. Оптимизация гарантии в задачах управления. – М.: Наука, I98I. – 288 с.