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

$А\frac{20204-214(83)}{055(02)\ 7}БО$

ПРЕДИСЛОВИЕ

Настоящая брошюра содержит описание алгоритмов и программ решения некоторых классов линейных позиционных антагонистических дифференциальных игр двух лиц. В теоретическом плане представленные алгоритмы опираются на материал книг: Н.Н.Красовский, А.И.Субботин "Позиционные дифференциальные игры"; А.И. Субботин, А.Г.Ченцов "Оптимизация гарантии в задачах управления". Использовались также работы Л.С.Понтрягина, Б.Н.Пшеничного и других авторов.

Одной из типичных задач, относящихся к дифференциальным играм, является задача о приведении конфликтно-управляемой системы на заданное целевое множество M в заданный момент времени ϑ. Основная часть алгоритмов брошюры связана с решением именно этой задачи, более точно - с нахождением множества W всех начальных позиций (t_*, x_*), откуда первый игрок гарантирует приведение системы на M в момент ϑ при любом противодействии второго игрока. Множество W называется множеством позиционного поглощения или максимальным стабильным мостом.

В основе алгоритмов нахождения множества W лежит попятная процедура построения его сечений $W(t_i)$ с некоторым шагом по t. Рассматриваются только игры второго порядка, либо задачи, эквивалентные играм второго порядка. Множество M предполагается выпуклым. Специфика плоскости и свойство выпуклости позволили создать экономные алгоритмы счета.

Задаче нахождения множества W посвящены работы Н.Д. Боткина; М.А.Зарха; Е.А.Исаковой, Г.В.Логуновой, В.С.Пацко; А.М.Тарасьева, В.Н.Ушакова. Алгоритм и программа оценки погрешности численных построений сечений $W(t_i)$ приведены в работе Н.Д.Боткина и М.А.Зарха.

Работа В.П.Дятлова связана с построением последовательности программных итераций для решения дифференциальной игры с ограничением на допустимое число переключений второго игрока.

В работе В.Л.Туровой "Линейная дифференциальная игра качества" рассматривается задача на плоскости с нефиксированным моментом окончания. Цель первого игрока - привести управляемую систему в заранее заданную точку. Второй игрок препятствует этому. Описываются алгоритм и программа нахождения множества всех начальных состояний, откуда такое приведение возможно за конечное время.

Работа В.Л.Туровой "Вычисление цены игры и оптимального управления для одного класса дифференциальных игр" посвящена нахождению цены и оптимальной стратегии первого (минимизирующего) игрока в дифференциальной игре с фиксированным моментом окончания и интегро-терминальным функционалом платы. Программа позволяет решать задачи достаточно высокой размерности.

Н.Д.Боткин

ЧИСЛЕННОЕ ПОСТРОЕНИЕ СЕЧЕНИЙ МНОЖЕСТВА ПОЗИЦИОННОГО ПОГЛОЩЕНИЯ В ЛИНЕЙНОЙ ДИФФЕРЕНЦИАЛЬНОЙ ИГРЕ

I. Постановка задачи

Рассматривается линейная дифференциальная антагонистическая игра двух лиц [I]

$$\dot{x} = Ax + Bu + Cv\ , \quad x \in R^n \tag{I}$$

с фиксированным моментом окончания ϑ. Здесь x - фазовый вектор, A, B - постоянные матрицы соответствующих размерностей, C - постоянная матрица размерности $n \times 1$, u - управление первого игрока, стесненное в каждый момент t ограничением $u(t) \in \mu(t) P$. Управление v второго игрока скалярное, в каждый момент t оно удовлетворяет ограничению $|v(t)| \leq \nu(t)$. Множество P - выпуклый многогранник в конечномерном пространстве, скалярные функции $\mu(t)$ и $\nu(t)$ имеют вид

$$\mu(t) = a_1 t + a_2\ , \ \nu(t) = a_3 t + a_4\ .$$

Цель первого игрока - привести две выделенные координаты x_α, x_β фазового вектора x в момент ϑ на выпуклое целевое множество $M \subset R^2$, т.е. обеспечить включение

$$(x_\alpha(\vartheta), x_\beta(\vartheta))' \in M \tag{2}$$

Интересы второго игрока противоположны.

Исследуется задача о нахождении множества W всех начальных позиций (t^o, x^o), для каждой из которых первый игрок, используя позиционный способ управления, может обеспечить включение (2) вне зависимости от действий второго игрока. В соответствии с [I] будем называть W множеством позиционного поглощения для игры (I) или максимальным u-стабильным мостом. Множество $W(t) = \{ x \in R^n : (t,x) \in W\}$, $t \leq \vartheta$, назовем сечением W в момент t. В рассматриваемой задаче множество $W(t)$ совпадает с альтернированным интегралом [2, 3], построенным для игры (I) от множества M на отрезке $[t, \vartheta]$.

Пусть $X_{\alpha,\beta}(\vartheta, t)$ - матрица, составленная из α и β -ой строк фундаментальной матрицы Коши $X(\vartheta, t) = \exp A(\vartheta - t)$ системы $\dot{x} = Ax$. Сделав в (I) замену переменных $z(t) = X_{\alpha,\beta}(\vartheta, t)\, x(t)$, придем к эквивалентной дифференциальной игре второго порядка

$$\dot{z} = \mu(t) \cdot X_{\alpha,\beta}(\vartheta, t) B u + \nu(t) \cdot X_{\alpha,\beta}(\vartheta, t) C v, \; z \in R^2 \tag{3}$$

с моментом окончания ϑ и целевым множеством M. Ограничения на управления игроков имеют вид

$$u \in P, \quad |v| \leq 1 .$$

Обозначим через $\tilde{W}$ множество позиционного поглощения для игры (3). Оно связано с W соотношением [I,5]

$$W(t) = \{ x \in R^n : X_{\alpha,\beta}(\vartheta, t) x \in \tilde{W}(t) \} \quad , \; t \leq \vartheta .$$

В статье представлены алгоритм и написанная на его основе программа, которая выполняет переход от игры (I) к игре (3) и строит цепочку множеств $\tilde{W}(\vartheta - \Delta)$, $\tilde{W}(\vartheta - 2\Delta)$, $\ldots$, $\tilde{W}(\vartheta - k\Delta)$, $\ldots$. Здесь Δ – заданный шаг. В качестве основных исходных данных наряду с шагом Δ задаются матрицы A , B , C , вершины многогранника P , числа a_1 , a_2 , a_3 , a_4 . Целевое множество M подменяется выпуклым многоугольником M^*, вершины которого также задаются в качестве исходных данных. Предусмотрена возможность умножения M^* на коэффициент $\gamma > 0$, что соответствует замене целевого множества M в игре (3) на множество $\gamma \cdot M$.

2. Описание алгоритма

Положим

$$D(t) = \mu(t) X_{\alpha,\beta}(\vartheta, t) B ,$$

$$E(t) = \nu(t) X_{\alpha,\beta}(\vartheta, t) C .$$

Введем аппроксимирующую по отношению к (3) игру

$$\dot{z} = D^*(t) u + E^*(t) v \quad , \; z \in R^2 \quad , \qquad (4)$$

в которой момент окончания ϑ и ограничения на управления игроков – прежние, целевое множество M заменено много – угольником M^*, функции $D^*(t)$, $E^*(t)$ являются кусочно-постоянными аппроксимациями функций $D(t)$, $E(t)$. Они определены следующим образом. Полуось $t \leq \vartheta$ разобьем точками $t_0 = \vartheta$, $t_1 = \vartheta - \Delta$, ..., $t_k = \vartheta - k\Delta$, ... Каждый отрезок $[t_{k+1}, t_k]$ разделим дополнительно на m равных частей точками τ_k^j, $j = \overline{1, m+1}$, причем $\tau_k^1 = t_k$, $\tau_k^{m+1} = t_{k+1}$. Положим

$$D^*(t) = D(\tau_k^j); \quad t \in (\tau_k^{j+1}; \tau_k^j], \ j = \overline{1,m}, \ k = 0,1,...,$$

$$E^*(t) = \frac{1}{\Delta} \int_{t_{k+1}}^{t_k} E(\xi)\, d\xi, \quad t \in (t_{k+1}, t_k], \ k = 0,1,... .$$

Для аппроксимирующей игры (4) построим цепочку множеств

$$\tilde{W}^*(t_0), \tilde{W}^*(t_1), ..., \tilde{W}^*(t_k), \tilde{W}^*(t_{k+1}), ... ,$$

которые будут близки к множествам $\tilde{W}(t_0), \tilde{W}(t_1), ..., \tilde{W}(t_k)$, $\tilde{W}(t_{k+1})$,.... Здесь $\tilde{W}^*(t_0) = M^*$, $\tilde{W}^*(t_{k+1})$ – совокупность всех таких начальных точек $x^0 \in R^2$, для каж – дой из которых по любому постоянному управлению v второго игрока можно указать такое программное управление $u(\cdot)$ первого игрока, что система (4) под действием этих управле – ний из начальной позиции (t_{k+1}, x^0) попадет в момент t_k на множество $\tilde{W}^*(t_k)$. Известно [1,4], что все множества $\tilde{W}^*(t_k)$ – выпуклые многоугольники. Если $m = 1$, то функция $D^*(t)$ постоянна на каждом полу- интервале (t_{k+1}, t_k) и из [4] следует, что $\tilde{W}^*(t_k)$, $k = 0,1,2,...$, будут сечениями множества позиционного поглощения для игры (4).

Зафиксируем номер k и рассмотрим процедуру построения множества $\tilde{W}^*(t_{k+1})$ по множеству $\tilde{W}^*(t_k)$. Предположим, что в игре (4) второй игрок держит управление $v \equiv 0$. Пусть при этом Ω – множество всех таких точек x°, что из начальной позиции (t_{k+1}, x°) первый игрок может привести движение системы (4) в момент t_k на множество $\tilde{W}^*(t_k)$. Пусть γ^* – постоянное значение функции $E^*(t)$ на полуинтервале $(t_{k+1}, t_k]$. Справедливо соотношение

$$\tilde{W}^*(t_{k+1}) = (\Omega - \gamma^*\Delta) \cap (\Omega + \gamma^*\Delta). \qquad (5)$$

Приведем формулы, описывающие множество Ω. Положим $P^j = D(\tau_k^j) \cdot P$, $j = \overline{1,m}$. Множества P^j, $j=\overline{1,m}$, – выпуклые многоугольники из R^2. Пусть N^j – набор единичных внутренних нормалей к сторонам многоугольника P^j, L – набор единичных внешних нормалей к сторонам выпуклого многоугольника $\tilde{W}^*(t_k)$. Обозначим

$$N = \bigcup_{j=1}^{m} N^j, \quad H = L \cup N.$$

Имеем [I,5]

$$\Omega = \{ x \in R^2 : \max_{\ell \in H} [\ell' x - \varphi(\ell)] \leq 0 \}, \qquad (6)$$

где

$$\varphi(\ell) = \rho(\ell, \tilde{W}^*(t_k)) - \frac{\Delta}{m} \sum_{j=1}^{m} \min_{p \in P^j} \ell' p \quad ,$$

$\rho(\ell, \tilde{W}^*(t_k))$ – опорная функция множества $\tilde{W}^*(t_k)$. Поскольку функция φ выпукла, то она в силу (6) будет опорной функцией множества Ω. Граница Γ множества Ω задается формулой

$$\Gamma = \{ z \in R^2 : \max_{\ell \in H} [\ell' z - \varphi(\ell)] = 0 \} .$$

Для того чтобы эффективно использовать равенство (5), разобьем границу Γ на две части Γ_1 и Γ_2. С этой целью введем единичный вектор η, совпадающий по направлению с η^*. Если $\eta^* = 0$, то вектор η можно взять произвольным (в программе $\eta = (0,1)$, если $\eta^* = 0$). Будем считать, что вектора из набора $H = \{ h_1, h_2, \dots$ $\dots, h_r \}$ упорядочены по часовой стрелке (рис. I) и

$$h_1' \eta \geq 0 \ , \ h_2' \eta > 0 \ , \dots , h_s' \eta > 0 \ ,$$

$$h_{s+1}' \eta \leq 0 \ , \ h_{s+2}' \eta < 0 \ , \dots , h_r' \eta < 0 \ .$$

Этого всегда можно добиться перенумерацией векторов из набора H. Пусть

$$H_1 = \{ h_1 , \dots , h_s \} \ , \quad H_2 = \{ h_{s+1} , \dots , h_r \} \ .$$

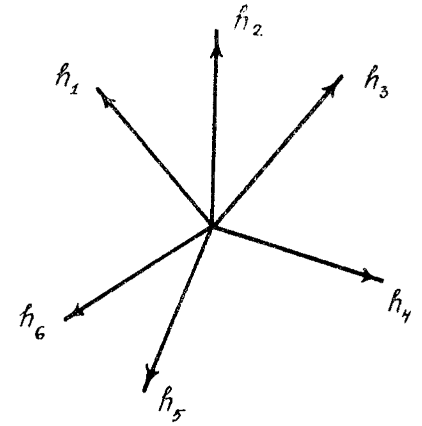


Рис. 1

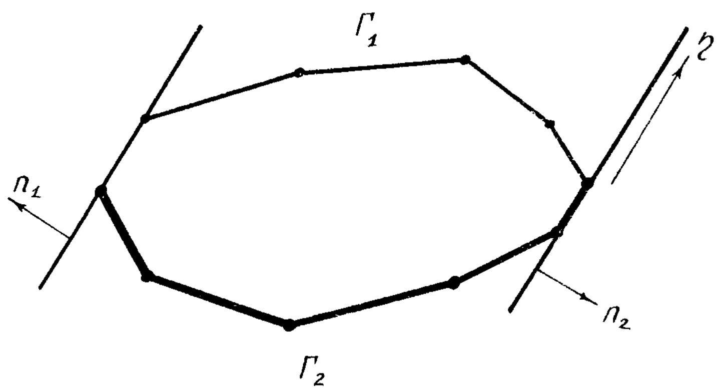


Рис. 2

Положим

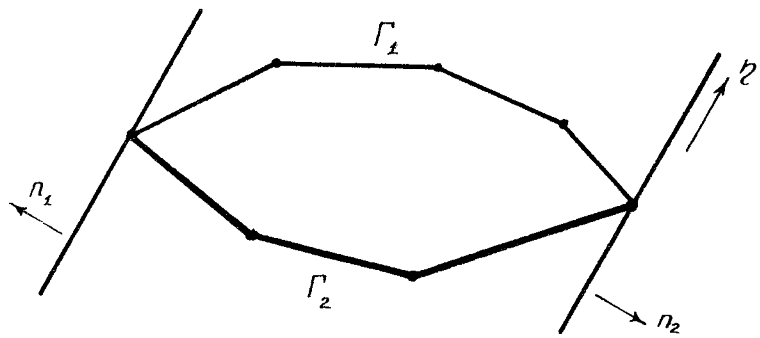
$$\Gamma_1 = \{ z \in \Gamma : \max_{\ell \in H_1} [\ell' z - \varphi(\ell)] = 0 \} ,$$

$$\Gamma_2 = \{ z \in \Gamma : \max_{\ell \in H_2} [\ell' z - \varphi(\ell)] = 0 \} .$$

Ломаные Γ_1 , Γ_2 схематично показаны на рис. 2,3. Ломаная Γ_2 выделена жирной линией. Линии, параллельные вектору η , являются опорными к многоугольнику Ω прямыми $n_1' z = \varphi(n_1)$, $n_2' z = \varphi(n_2)$. Здесь n_1 – вектор, полученный поворотом вектора η на угол $\pi/2$ против часовой стрелки, $n_2 = -n_1$. Для случая, изображенного на рис. 2, $h_1 = n_1$, $h_{s+1} = n_2$. На рис. 3 показан случай, когда $n_1 , n_2 \notin H$.

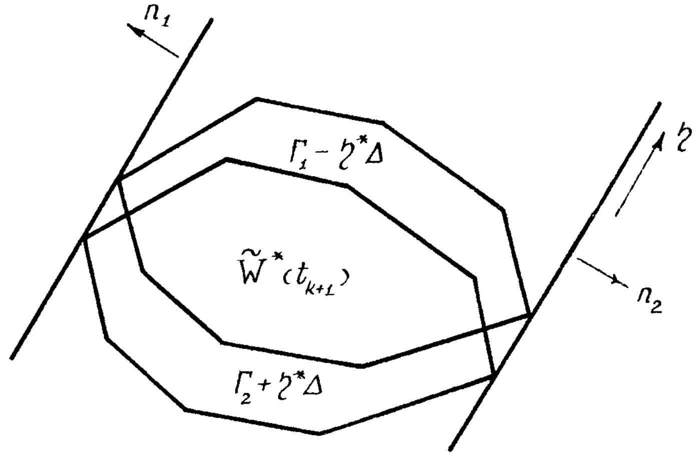
Из равенства (5) и определения ломаных Γ_1 , Γ_2 вытекает, что граница множества $\tilde{W}^*(t_{k+1})$ состоит из двух дуг, одна из которых является частью ломаной $\Gamma_1 - \eta^* \Delta$, другая – частью ломаной $\Gamma_2 + \eta^* \Delta$ (рис. 4). Таким образом, для отыскания множества $\tilde{W}^*(t_{k+1})$ надо построить ломаные $\Gamma_1 - \eta^* \Delta$, $\Gamma_2 + \eta^* \Delta$ и найти две точки их пересечения.

В целях упрощения алгоритма и увеличения его быстродействия в программе строятся не сами ломаные Γ_1 , Γ_2 , а некоторые близкие к ним ломаные $\bar{\Gamma}_1$, $\bar{\Gamma}_2$, совпадающие с Γ_1 , Γ_2 в случае, когда P – отрезок и $m = 1$. Таким образом, на каждом шаге попятной процедуры возникает погрешность за счет замены Γ_1 , Γ_2 на $\bar{\Gamma}_1$, $\bar{\Gamma}_2$.



Γ_1
η
n_1
Γ_2
n_2

Рис. 3



n_1
η
$\Gamma_1 - \eta^* \Delta$
$\tilde{W}^*(t_{k+1})$
n_2
$\Gamma_2 + \eta^* \Delta$

Рис. 4

Следовательно, программа строит, вообще говоря, не множества $\tilde{W}^*(t_j)$, $j=1,\dots$, а некоторые близкие к ним множества. Условимся использовать для них, однако, те же самые обозначения.

Опишем построение ломаных $\bar{\Gamma}_1$, $\bar{\Gamma}_2$. Пусть $x_1, x_2, \dots, x_\nu$ – вершины многоугольника $\tilde{W}^*(t_k)$, занумерованные в соответствии с обходом по часовой стрелке. Обозначим через ℓ_i внешнюю единичную нормаль к стороне $[x_i, x_{i+1}]$, $i=\overline{1,\nu-1}$, и через ℓ_ν – внешнюю нормаль к стороне $[x_\nu, x_1]$. Перенумеруем вершины многоугольника так, чтобы они оставались упорядоченными по часовой стрелке и выполнялись бы соотношения

$$\ell_1'\eta \geq 0,\ \ell_2'\eta > 0,\ \dots,\ \ell'_\omega\eta > 0,$$

$$\ell'_{\omega+1}\eta \leq 0,\ \ell'_{\omega+2}\eta < 0,\ \dots,\ \ell'_\nu\eta < 0.$$

Такую операцию будем называть упорядочением набора L относительно вектора η. Пусть ℓ – ненулевой вектор из R^2. Положим

$$\Phi^j(\ell) = \{p^\circ \in P^j:\ \ell' p^\circ = \min_{p\in P^j} \ell' p\},\ j=\overline{1,m}.$$

Множество $\Phi^j(\ell)$ представляет собой либо вершину многоугольника P^j, либо его ребро. Для любых непараллельных векторов $\ell, h \in R^2$ обозначим символом $\xi^j(\ell,h)$ точку из P^j, для которой

$$h'\xi^j(\ell,h) = \min_{\xi\in\Phi^j(\ell)} h'\xi.$$

В силу непараллельности векторов ℓ и h такая точка будет единственной. Определим на отрезке $[t_{k+1}, t_k]$ кусочно-постоянную функцию

$$\xi(\ell, h)(t) = \xi^j(\ell, h), \quad t \in (\tau_k^{j+1}, \tau_k^j], \; j = \overline{1,m},$$

$$\xi(\ell, h)(t_{k+1}) = \xi^m(\ell, h).$$

Рассмотрим дифференциальное уравнение

$$\dot{z} = \xi(t), \quad z \in R^2. \tag{7}$$

Его решение, отвечающее функции $\xi(\cdot)$ и начальному значению z^o в момент t^o, обозначим через $z(\cdot, t^o, z^o, \xi(\cdot))$. Из каждой вершины x_i, $i = \overline{2,\omega}$, выпустим по два решения уравнения (7), отличающихся выбором функции $\xi(\cdot)$. Для первого из них положим $\xi^{(1)}(t) = \xi(\ell_{i-1}, \ell_i)(t)$, для второго $\xi^{(2)}(t) = \xi(\ell_i, \ell_{i-1})(t)$. Обозначим

$$z_i^{(1)} = z(t_{k+1}, t_k, x_i, \xi^{(1)}(\cdot)), \quad z_i^{(2)} = z(t_{k+1}, t_k, x_i, \xi^{(2)}(\cdot)).$$

Перебрав все i от 2 до ω, получим набор точек $z_2^{(1)}, z_2^{(2)}, \ldots, z_\omega^{(1)}, z_\omega^{(2)}$. Отметим, что точки с одинаковыми нижними индексами могут совпадать. В этом случае одна из совпадающих точек отбрасывается.

Рассмотрим точку x_1. Если $\ell_1' \eta = 0$, то из точки x_1 выпустим одно решение уравнения (7) с $\xi(t) = \xi(\ell_1, -\eta)(t)$ и получим точку $z_1 = z(t_{k+1},$

$t_k, x_1, \xi(\cdot))$. Если $\ell_1' \eta > 0$, то из точки x_1 выпустим три решения уравнения (7), соответствующие функциям

$$\xi^{(1)}(t) = \xi(n_1, -\eta)(t), \ \xi^{(2)}(t) = \xi(n_1, \eta)(t), \ \xi^{(3)}(t) = \xi(\ell_1, -\eta)(t).$$

Получим три точки $z_1^{(1)}$, $z_1^{(2)}$, $z_1^{(3)}$ (некоторые из них или все могут совпадать).

Рассмотрим точку $x_{\omega+1}$. Из нее выпустим два решения уравнения (7). Первое при $\xi^{(1)}(t) = \xi(\ell_\omega, -\eta)(t)$, второе при $\xi^{(2)}(t) = \xi(n_2, \eta)(t)$. В результате получим две точки $z_{\omega+1}^{(1)}$, $z_{\omega+1}^{(2)}$, которые, возможно, совпадают.

Итак, если $\ell_1' \eta = 0$, то в качестве $\overline{\Gamma}_1$ берем ломаную $z_1, z_2^{(1)}, z_2^{(2)}, \ldots, z_{\omega+1}^{(1)}, z_{\omega+1}^{(2)}$. Если $\ell_1' \eta > 0$, то в качестве $\overline{\Gamma}_1$ берется ломаная $z_1^{(1)}, z_1^{(2)}, z_1^{(3)}, \ldots, z_{\omega+1}^{(1)}, z_{\omega+1}^{(2)}$. Для построения $\overline{\Gamma}_2$ достаточно заметить, что при замене вектора η на вектор $-\eta$ ломаные $\overline{\Gamma}_1$ и $\overline{\Gamma}_2$ меняются ролями.

Укажем некоторые свойства ломаных $\overline{\Gamma}_1$, $\overline{\Gamma}_2$:

1) концы ломаной $\overline{\Gamma}_1$ ($\overline{\Gamma}_2$) совпадают с концами Γ_1 (Γ_2),

2) каждая вершина $\overline{\Gamma}_1$ ($\overline{\Gamma}_2$) является вершиной Γ_1 (Γ_2),

3) вершина ломаной Γ_1 (Γ_2) является вершиной ломаной $\overline{\Gamma}_1$ ($\overline{\Gamma}_2$), когда она есть точка пересечения опорных к Ω прямых $h_i' z = \varphi(h_i)$, $h_{i+1}' z = \varphi(h_{i+1})$, где хотя бы один из векторов h_i, h_{i+1} входит в набор L, то есть является нормалью к одной из сторон многоугольника $\tilde{W}^*(t_k)$.

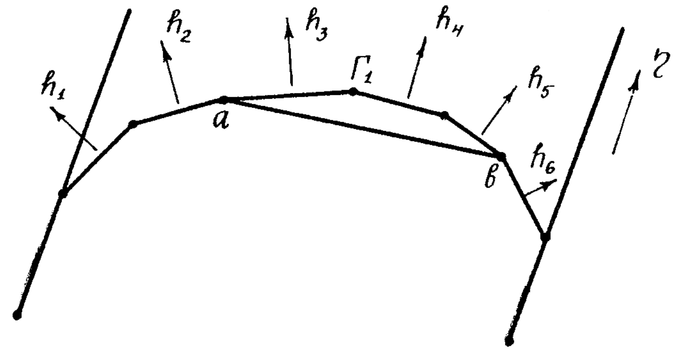
Рис. 5 поясняет свойства 1)-3) на примере линий $\overline{\Gamma}_1$,

Γ_1 . Ломаная $\bar{\Gamma}_1$ совпадает с Γ_1 всюду за исключением участка ab . Относительно векторов h_i , $i = \overline{1,6}$, показанных на этом рисунке, предполагается, что h_1 , h_2 , $h_6 \in L$, h_3 , h_4 , $h_5 \notin L$.

Из свойств I)-3) вытекает равенство $\bar{\Gamma}_i = \Gamma_i$, $i = 1,2$, если P - отрезок и $m = 1$. Действительно, в этом случае множество N состоит из двух противоположно направленных векторов, поэтому хотя бы один из соседних векторов h_i , $h_{i+1} \in H$ является вектором из L . Для общего случая свойства I)-3) показывают, что, если в L имеется много нормалей, то $\bar{\Gamma}_1$, $\bar{\Gamma}_2$ близки к Γ_1 , Γ_2 и даже могут совпадать с ними (для чего нужно, чтобы между любыми двумя соседними векторами из L находилось не более одного вектора из N). Если $\bar{\Gamma}_1$, $\bar{\Gamma}_2$ не совпадают с Γ_1 , Γ_2 , то множество, заключенное между $\bar{\Gamma}_1$, $\bar{\Gamma}_2$, вложено в множество, заключенное между Γ_1 , Γ_2 .

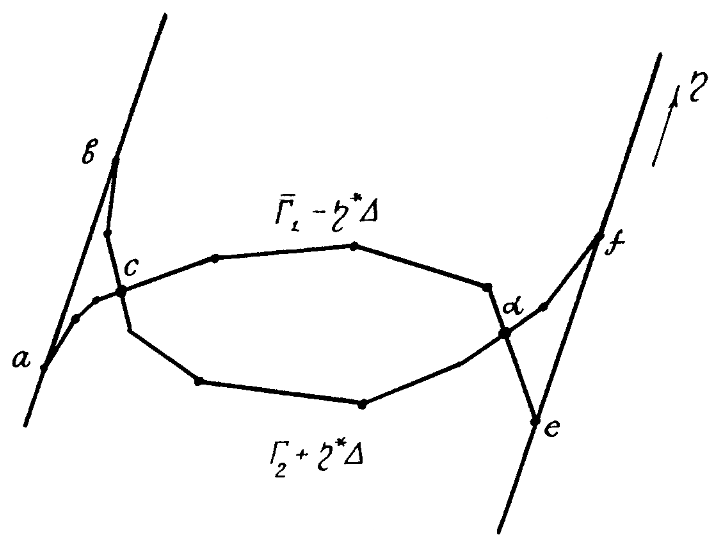
Скажем несколько слов о нахождении точек пересечения линий $\bar{\Gamma}_1 - \gamma^* \Delta$, $\bar{\Gamma}_2 + \gamma^* \Delta$. На рис. 6 показаны эти линии и точки их пересечения c , d . Точку c отыскиваем, двигаясь от точек a и b вправо по ломаным $\bar{\Gamma}_1 - \gamma^* \Delta$, $\bar{\Gamma}_2 + \gamma^* \Delta$. Точку d отыскиваем, двигаясь от точек e и f влево. Поскольку расстояние между точками a и b имеет порядок Δ , а шаг Δ обычно мал, то точка c находится после просмотра небольшого количества звеньев ломаных $\bar{\Gamma}_1 - \gamma^* \Delta$, $\bar{\Gamma}_2 + \gamma^* \Delta$. То же самое можно сказать об отыскании точки d .

Рассмотрим вопрос о вычислении матриц $D(t)$, $E(t)$, стоящих в (3) при u и v . Пусть $X_i(\vartheta, t)$ - i-ая строка матрицы $X(\vartheta, t)$. Положим $\psi_i(\tau) =$



h_1
h_2
h_3
h_4
h_5
h_6
Γ_1
a
в

Рис. 5



в
a
c
d
f
e
$\bar{\Gamma}_1 - \ell^* \Delta$
$\Gamma_2 + \ell^* \Delta$

Рис. 6

$= X_i(\vartheta, \vartheta-\tau)$. Функция $\psi_i(\cdot)$ является решением уравнения

$$\frac{d\psi_i}{d\tau} = \psi_i A \tag{8}$$

с начальным условием

$$\psi_i(0) = (0, \dots, 1, \dots, 0) ,$$

где единица стоит на i -ом месте. Это уравнение интегрируется при помощи стандартной программы с шагом $\Delta/(mq)$. Здесь q - целое число, задаваемое в качестве исходного данного. Напомним, что m - число частей, на которые дополнительно разбит каждый отрезок $[t_{k+1}, t_k]$. Таким образом, шаг дополнительного разбиения равен Δ/m , шаг интегрирования дифференциального уравнения (8) равен $\Delta/(mq)$.

3. Краткая характеристика программы

Программа написана на языке ФОРТРАН-ДУБНА для БЭСМ-6. Имя головной программы - *MOCT* , подпрограммы: *BB* , *ORDER* , *MOTION* , *INTRSE* , *MOVE* , *FUN* , *XA1* , *XA2* . Для интегрирования дифференциального уравнения (8) используется стандартная программа *INTSTP* из библиотеки стандартных программ ОИЯИ ДУБНА.

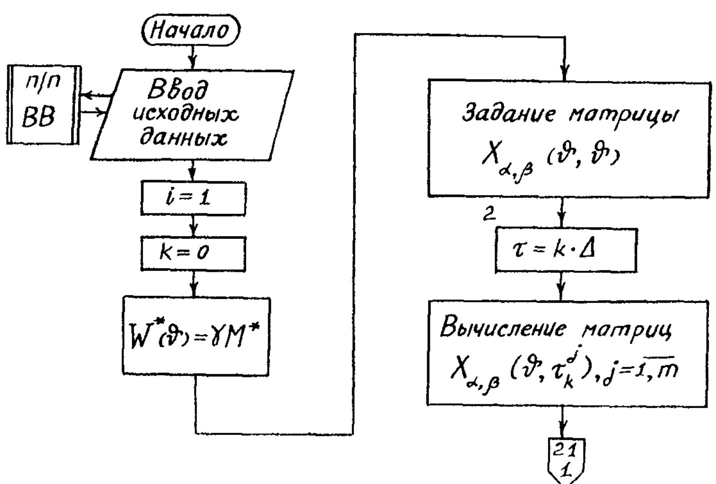
Программа *MOCT* предназначена для приближенного построения цепочки множеств $\tilde{W}(\vartheta-\Delta)$, $\tilde{W}(\vartheta-2\Delta), \dots$,

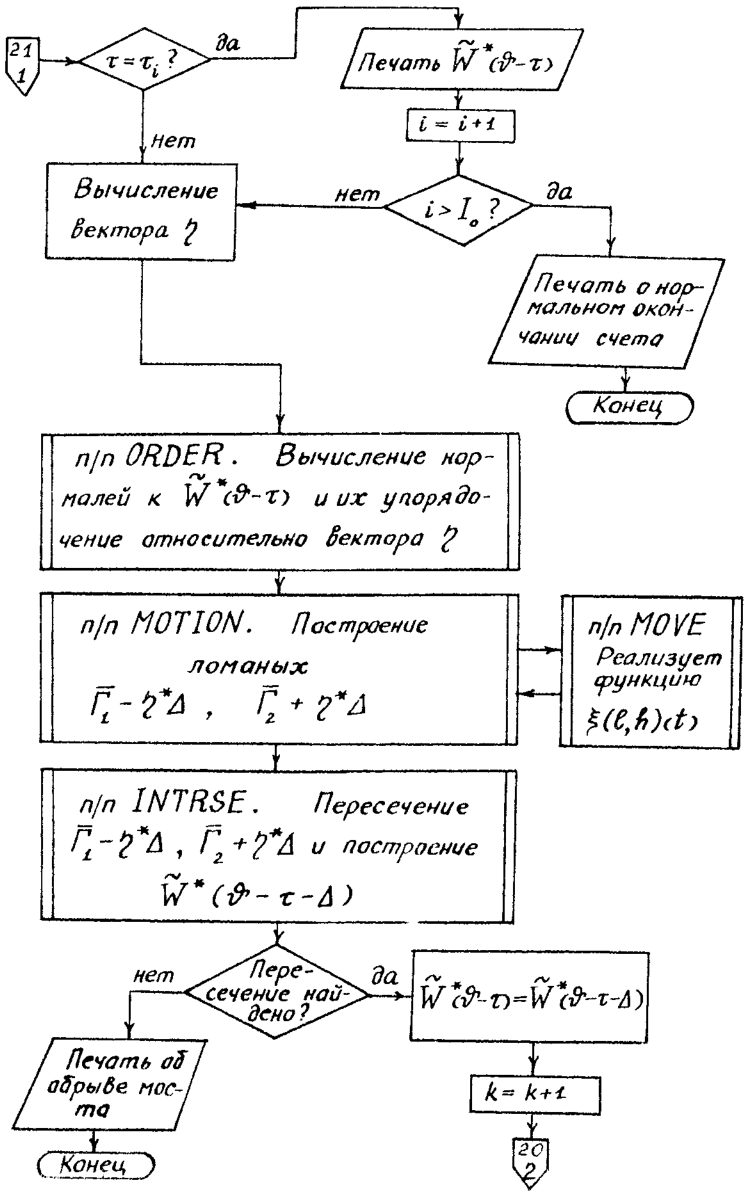
$\tilde{W}(\vartheta - k\Delta)$, ... , причем переход от исходной диффе - ренциальной игры (I) к эквивалентной игре (3) выполняется автоматически. Точность построений тем выше, чем меньше шаг Δ. При неизменном шаге Δ точность построений увеличивается с ростом числа m.

Если P - отрезок и $m=1$, то результатом счета являются сечения $\tilde{W}^*(t_k)$ множества позиционного поглощения для аппроксимирующей игры (4). Для оценки близости множеств $\tilde{W}^*(t_k)$ и $\tilde{W}(t_k)$ в этом случае можно воспользоваться методом из работы [6].

Блок - схема

Обозначения, применяющие в блок-схеме, взяты из предыдущих разделов. К ним добавлены следующие: $\tau = \vartheta - t$ - обратное время; τ_i, $i=\overline{1, I_o}$, - моменты обратного времени, в которые надо выводить информацию на печать; I_o - число этих моментов.




21
1
τ = τ_i ?
да
нет
Печать W̃*(ϑ-τ)
i = i+1
i > I_o ?
нет
да
Вычисление вектора η
Печать о нормальном окончании счета
Конец
п/п ORDER. Вычисление нормалей к W̃*(ϑ-τ) и их упорядочение относительно вектора η
п/п MOTION. Построение ломаных Γ̄₁ - η*Δ , Γ̄₂ + η*Δ
п/п MOVE Реализует функцию ξ(ℓ,h)(t)
п/п INTRSE. Пересечение Γ̄₁ - η*Δ , Γ̄₂ + η*Δ и построение W̃*(ϑ - τ - Δ)
Пересечение найдено?
нет
да
W̃*(ϑ-τ)=W̃*(ϑ-τ-Δ)
Печать об обрыве моста
Конец
k = k+1
20
2

Используемые переменные

N - порядок системы (I),

$N1$, $N2$ - номера α, β двух координат фазового вектора, которые надо привести на целевое множество M,

NTP - число I_0 моментов обратного времени, когда надо выдавать информацию на печать,

$TP(40)$ - массив моментов обратного времени, в которые надо выдавать информацию на печать,

$A(15,15)$ - матрица A системы (I),

$BE(15,10)$ - матрица B системы (I),

$CE(15)$ - матрица C системы (I),

TET - момент окончания игры ϑ,

$AKA1$, MU - коэффициенты a_1, a_2, определяющие функцию $\mu(t)$,

$AKA2$, HU - коэффициенты a_3, a_4, определяющие функцию $\nu(t)$,

NP - размерность управления u первого игрока,

KP - число вершин многогранника P,

$TOP(10,100)$ - двумерный массив, в котором содержатся вершины многогранника P, занумерованные в произвольном порядке. Каждый столбец этого массива соответствует одной из вершин,

$Y1(1500)$, $Y2(1500)$ - одномерные массивы, содержащие первые и вторые координаты вершин многоугольника $\tilde{W}^*(t_k)$ на k-ом шаге попятной процедуры,

QU - целая переменная, равная числу вершин многоугольника $\tilde{W}^*(t_k)$,

$SHAG$ – шаг Δ основного разбиения,

ASH – шаг дополнительного разбиения, $ASH=\Delta/m$,

$SHADY$ – шаг интегрирования дифференциального уравнения (8), $SHADY=\Delta/(mq)$,

II – число m – количество частей дополнительного разбиения,

IM – число q , определяющее $SHADY$,

TH – обратное время τ ,

$KOEF$ – множитель Y перед M^* (вещественная переменная).

Ввод данных и вывод результатов

Основная часть исходных данных вводится с перфокарт. Исключение составляет множество M^* , которое вводится посредством обращения к подпрограмме BB . Обращение имеет вид

```
CALL BB(Y1,Y2,QU,L).
```

Подпрограмма BB должна иметь следующую форму:

```
SUBROUTINE BB(X1,X2,M,L)
DIMENSION X1(M),X2(M)
```

Вычисление или ввод первых и вторых координат вершин многоугольника M^* и засылка их в массивы $X1$,

$X2$, присвоение переменной M числа вершин многоугольника M^* .

```
L=M/2
RETURN
END
```

Ее назначение – засылка в массивы $X1$, $X2$ первой и второй координат вершин многоугольника M^* . Необходимо, чтобы вершины M^* были расположены в массивах $X1$, $X2$ по часовой стрелке. Порядок ввода остальной информации указан в таблице. Там, где информация умещается на одной перфокарте, в графе "количество чисел на одной карте" стоит прочерк.

Результатами счета являются вершины многоугольников $\tilde{W}(\vartheta - \tau_i)$, где τ_i – заданные моменты обратного времени (массив TP). Для каждого момента τ_i информация на печать выдается в следующем виде:

I строка	$TAU = (\tau_i)$
2 строка	ЧИСЛО ТОЧЕК (число точек)
3 строка	$Y1(1)$, $Y2(1)$, ... , $Y1(5)$, $Y2(5)$
4 строка	$Y1(6)$, $Y2(6)$, ... , $Y1(10)$, $Y2(10)$
5 строка	$Y1(11)$, $Y2(11)$, ... и т.д.

Перед началом счета на печать выдаются исходные данные в порядке, указанном в таблице.

Особенности программы

Число вершин многоугольника $\tilde{W}^*(t_k)$ меняется с ростом k . Максимально допустимое число вершин определяется

Таблица

№ оператора READ	Вводимая информация	Количество чисел на одной перфокарте	Формат
I	N, N1, N2, NP, KP	-	5I3
2	II, IM	-	2I3
3	A (по строкам)	I0	10F7.3
4	BE (по строкам)	I0	10F7.3
5	CE	I0	10F7.3
6	TOP (по столбцам)	I0	10F7.3
7	AKA1, MU	-	2F7.3
8	AKA2, HU	-	2F7.3
9	KOEF	-	F7.3
I0	SHAG	-	F7.3
II	TET	-	F7.3
I2	NTP	-	I3
I3	TP	I0	10F7.3

размерностью массивов $Y1$, $Y2$, которая равна 1500. Если на k-ом шаге число вершин $\tilde{W}^*(t_k)$ окажется большим, чем 1500, то счет прерывается и на печать выдается сообщение

$$TAU=(\vartheta-t_k) \quad \text{ВЕРШИН} > 1500$$

В этом случае рекомендуется, если позволяет существо задачи, уменьшить число вершин целевого множества M^*, уменьшить число m или увеличить шаг Δ.

Может случиться, что на k-ом шаге попятной процедуры произошел обрыв множества позиционного поглощения, т.е. $\tilde{W}^*(t_k)=\phi$. Тогда на печать выдается сообщение

$$TAU=(\vartheta-t_k) \quad \text{МОСТ ОБОРВАЛСЯ}$$

Допускается в качестве целевого множества M^* брать отрезок. Точку в качестве M^* брать нельзя.

8. Тестовая задача

Для дифференциальной игры второго порядка

$$\dot{x}_1 = x_2 + v$$
$$\dot{x}_2 = u$$
$$|u|\leq 1, \ |v|\leq 1, \ x(\vartheta)\in\gamma\cdot M,$$

где M – симметричный относительно нуля отрезок на оси x_1 длины 2, вычислены для различных γ моменты обрыва множества позиционного поглощения. При вычислениях полагалось: $\Delta=0.05$, $m=5$, $q=1$. Для значений коэффициента γ равных 0.6, 0.8, 1, 1.1 получены следующие величины моментов обрыва: 0.7, 1.1, 1.9, 2.5 (время обратное).

ТЕКСТ ПРОГРАММЫ

```
      PROGRAM MOCT
      DIMENSION Y1(1500),
      Y2(1500),BY1(1500),
      BY2(1500),EL1(1500),
      EL2(1500),A(15,15),
      BE(15,10),CE(15),
      CTP1(15,20),CTP2(15,20)
      ,BCTP1(15),BCTP2(15),
      TOP(10,100),TP(40)
      INTEGER QU,C1,C2
      REAL MU,HU,KS1,KS2,KOEF
      EXTERNAL FUN
      COMMON/GRA/C1,C2,QU
      COMMON/A/A,N
      COMMON/KSI/AKS1,AKS2,
      SHAG,ASH,TH,TET
      COMMON/CTP/CTP1,CTP2,
      TOP,BE,NP,KP
      COMMON/OG/AKA1,MU
      COMMON/Y/Y1,Y2,BY1,BY2,
      EL1,EL2
      COMMON/EP/EPS,EPS1,EPS2
      ,EPS3
C     ВВОД ИСХОДНЫХ ДАННЫХ
      CALL BB(Y1,Y2,QU,L)
      RE'        N2,NP,KP
      F
      READ2,AKA2,HU
      READ2,KOEF
      READ2,SHAG
      READ2,TET
      READ1,NTP
      READ2,(TP(J),J=1,NTP)
C     ПЕЧАТЬ ИСХОДНЫХ ДАННЫХ
      PRINT1,N,N1,N2,NP,KP
      PRINT1,II,IM
      PRINT4,((A(I,J),J=1,N)
      ,I=1,N)
      PRINT4,((BE(J,K),J=1,N)
      ,K=1,NP)
      PRINT4,(CE(J),J=1,N)
      PRINT4,((TOP(I,J),I=1
      ,NP),J=1,KP)
      PRINT4,AKA1,MU
      PRINT4,AKA2,HU
      PRINT4,KOEF
      PRINT4,SHAG
      PRINT4,TET
      PRINT1,NTP
      PRINT4,(TP(J),J=1,NTP)
C     ПОДГОТОВКА ДАННЫХ
      DO 191 I=1,QU
      Y1(I)·      '1(I)
  191 Y2(J        'I)
```

На стр. 27 - 37 идёт текст программы на Фортране.

Л и т е р а т у р а

I. Красовский Н.Н., Субботин А.И. Позиционные дифференциальные игры. – М.: Наука, 1974. – 456 с.

2. Понтрягин Л.С. О линейных дифференциальных играх. 2. – Докл. АН СССР, 1967, т. 175, № 4, с. 764-766.

3. Понтрягин Л.С. Линейные дифференциальные игры преследования. – Матем. сборник, 1980, т. 112, вып. 3, с. 307-330.

4. Пшеничный Б.Н., Сагайдак М.И. О дифференциальных играх с фиксированным временем. – Кибернетика, 1970, № 2, с.54-63

5. Субботин А.И., Ченцов А.Г. Оптимизация гарантии в задачах управления. – М.: Наука, 1981. – 288 с.

6. Botkin N.D. Evaluation of numerical construction error in differential game with fixed terminal time.-Problems of Control and Information Theory,1982,Vol.11,N 4,pp.283--295.