Серия «Математика»
2014. Т. 8. С. 164—177

Онлайн-доступ к журналу:
http://isu.ru/izvestia

И З В Е С Т И Я
Иркутского государственного университета

УДК 517.977

Численные методы расчета оптимального по быстродействию управления *

А. И. Тятюшкин
Институт динамики систем и теории управления СО РАН

Аннотация. В статье предложены алгоритмы для численного решения задачи оптимального быстродействия в линейных системах. Предлагаемый набор алгоритмов позволяет организовать многометодную технологию для расчета оптимального по быстродействию управления в задачах с различными особенностями. В соответствии с этой технологией решение задачи находится многометодным алгоритмом, состоящим из последовательности шагов разных методов, подключаемых к процессу оптимизации с целью его ускорения. Такая технология позволяет учитывать особенности решаемой задачи на всех стадиях ее решения и повышает эффективность поиска оптимального управления.

Ключевые слова: метод выпуклых оболочек, множество достижимости, задача быстродействия, минимаксный алгоритм, линейная управляемая система.

Введение

Задачи оптимального быстродействия составляют важный класс прикладных задач оптимального управления. К ним относятся, например, следующие задачи из динамики полета: приведение движущихся объектов в заданное положение, ориентация и стабилизация летательных аппаратов, подъем ракеты на максимальную высоту и другие задачи управления движением. Наряду с нелинейной моделью объекта управления на стадии проектирования или на малом временном отрезке нередко используется линейная модель. При этом многие задачи терминального управления формируются в виде задачи минимизации квадратичной функции, которая служит мерой уклонения конечного

* Работа выполнена при частичной финансовой поддержке Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ (НШ-5007.2014.9).

состояния от некоторого заданного состояния. Последовательность таких задач возникает и при решении задачи быстродействия прямыми методами. Поэтому построению эффективных вычислительных алгоритмов для решения этой задачи уделяется большое внимание.

Для численного решения задачи быстродействия при наличии ограничений только на управляющие функции здесь описывается алгоритмическое обеспечение, основанное на математических методах [1; 2; 5; 6; 10] и позволяющее реализовать многометодную технологию [7; 12] поиска оптимального по быстродействию управления. Задачи с ограничениями на фазовые координаты обычно решаются посредством дискретизации системы с формированием задачи линейного программирования, для решения которой затем используются специальные методы, учитывающие структуру матрицы основных ограничений [4].

1. Минимизация выпуклой функции конечного состояния

Будем рассматривать задачи оптимального управления в линейных системах

$$\dot{x} = A(t)x + B(t)u(t) + f(t), \quad x(t_0) = x^0, \tag{1.1}$$

где $x(t) \in R^n$, $u(t) \in R^r$, $A(t)$, $B(t)$, $f(t)$ — $n \times n$, $n \times r$, $n \times 1$-матрицы с кусочно-непрерывными элементами.

В теории линейных управляемых систем значительную роль играет переходная матрица [1] $\Phi(t, \tau)$, называемая также матрицей Грина. Она удовлетворяет следующему матричному дифференциальному уравнению:

$$\frac{d}{d\tau}\Phi(t, \tau) = -\Phi(t, \tau)A(\tau), \quad \Phi(t, t) = E. \tag{1.2}$$

С помощью этой матрицы решение (1.1) записывается по формуле Коши

$$x(t) = \Phi(t, t_0)x^0 + \int\limits_{t_0}^{t} \Phi(t, \tau)\left[B(\tau)u(\tau) + f(\tau)\right] d\tau.$$

Введя обозначение $c = \Phi(t_1, t_0)x^0 + \int\limits_{t_0}^{t_1} \Phi(t_1, \tau)f(\tau)\, d\tau$, вектор состояния в момент $t = t_1$ представим в виде

$$x(t_1) = c + \int\limits_{t_0}^{t_1} \Phi(t_1, \tau)B(\tau)u(\tau)\, d\tau. \tag{1.3}$$

Из (1.3) следует, что вектор c является состоянием системы (1.1) в момент $t = t_1$, в которое она переводится с помощью нулевого управления $(u(t) \equiv 0)$.

Множество конечных состояний $x(t_1)$, соответствующее заданному классу допустимых управлений, называется множеством достижимости системы (1.1).

Важнейшим свойством линейных систем, которое позволяет добиться существенного ускорения численных методов оптимизации, является выпуклость множества достижимости. С использованием этого свойства многие алгоритмы дают возможность найти новое приближение для управления с помощью решения простой задачи выпуклого программирования. В отличие от нелинейных систем алгоритмы решения данной задачи не требуют дополнительных интегрирований системы (1.1).

Пусть на правых концах траекторий системы (1.1) определена непрерывно дифференцируемая выпуклая скалярная функция $\varphi(x(t_1))$ и поставлена задача

$$\varphi(x(t_1)) \to \min . \tag{1.4}$$

Наиболее распространенный подход к решению задачи (1.1), (1.4) состоит в применении градиентных методов

$$u^{k+1}(t) = u^k(t) + \alpha_k B'(t)\psi^k(t), \quad k = 0, 1, \dots ,$$

где $\psi(t)$ — решение задачи Коши

$$\dot{\psi}(t) = -A'(t)\psi(t), \quad \psi(t_1) = -\nabla\varphi(x(t_1)), \tag{1.5}$$

α_k находится, например, из условия наискорейшего спуска.

Опишем вычислительную технологию решения задачи (1.1), (1.4) с ограничениями на управление

$$\alpha_j(t) \le u_j(t) \le \beta_j(t), \quad j = \overline{1, r}, \quad t \in [t_0, t_1], \tag{1.6}$$

методом условного градиента. В отличие от традиционной схемы метода, в которой точки множества достижимости получаются интегрированием системы (1.1), построим алгоритм, в котором эти точки генерируются с помощью базисных решений сопряженной системы.

Решением линейной вспомогательной задачи $\nabla\varphi\left(x^k(t_1)\right)' x \to \min$, определяемой на каждом шаге метода, будет управление

$$\bar{u}_j^k(t) = \begin{cases} \alpha_j(t), & \text{если} \quad b_j(t)'\psi^k(t) < 0, \\ \beta_j(t), & \text{если} \quad b_j(t)'\psi^k(t) \ge 0, \quad j = \overline{1, r}, \end{cases} \tag{1.7}$$

где $\psi^k(t)$ — решение задачи Коши (1.5) при $\psi(t_1) = -\nabla\varphi\left(x^k(t_1)\right)$. Итерация метода условного градиента описывается формулами

$$u^{k+1}(t) = u^k(t) + \alpha_k \left[\bar{u}^k(t) - u^k(t)\right],$$

$$\alpha_k = \arg \min_{0 \le \alpha \le 1} \phi\left(x^k(t_1) + \alpha\left[\bar{x}^k(t_1) - x^k(t_1)\right]\right). \tag{1.8}$$

Для получения точек $x^k(t_1)$ и $\bar{x}^k(t_1)$ необходимо интегрировать систему (1.1) при управлениях $u^k(t)$ и $\bar{u}^k(t)$. Используя покоординатное представление точки $x(t_1)$:

$$x_i(t_1) = c_i + \int\limits_{t_0}^{t_1} \psi^i(t)' B(t) u(t)\, dt, \quad i = \overline{1, n},$$

можно исключить операцию интегрирования системы (1.1), заменив ее интегрированием более простой системы. Для этого необходимо предварительно найти решения (1.5) $\psi^i(t)$, при начальных условиях — единичных векторах ортонормированного базиса: $\psi^i(t_1) = e^i$, $i = \overline{1, n}$; с помощью этих решений посчитать и запомнить в узлах интегрирования строки $n \times r$-матрицы $M(t)$, равные $\psi^i(t)' B(t)$, $i = \overline{1, n}$. Проинтегрировав систему (1.1) при $u(t) \equiv 0$, вычислить вектор c.

Пусть теперь задано допустимое управление $u^k(t)$. Тогда итерация (1.8) состоит из следующих операций:

1) решается задача Коши $\dot{x}^k = M(t) u^k(t)$, $x^k(t_0) = c$, в результате чего найдется вектор $x^k(t_1)$;

2) решается задача Коши (1.5), и по формулам (1.7) вычисляется управление $\bar{u}^k(t)$;

3) решается задача Коши $\dot{\bar{x}}^k = M(t)\, \bar{u}^k(t)$, $\bar{x}^k(t_0) = c$, после чего будем иметь $\bar{x}^k(t_1)$;

4) решается задача одномерного поиска

$$\varphi\left(x^k(t_1) + \alpha\left[\bar{x}^k(t_1) - x^k(t_1)\right]\right) \to \min_{0 \le \alpha \le 1};$$

5) по формуле (1.8) строится новое приближение.

Начиная со второй итерации операцию 1) можно не выполнять, так как значение $x^{k+1}(t_1)$ будет найдено при выполнении операции 4). Однако в целях сохранения устойчивости процесса вектор $x^{k+1}(t_1)$ через определенное число итераций следует получить с помощью операции 1). Итерационный процесс (1.8) прекращается при выполнении неравенства

$$\varphi\left(x^k(t_1)\right) - \varphi\left(x^k(t_1) + \alpha_k\left[\bar{x}^k(t_1) - x^k(t_1)\right]\right) \le \varepsilon$$

или необходимого условия оптимальности

$$-\int\limits_{t_0}^{t_1} B'(t) \psi^k(t) \left[u^k(t) - \bar{u}^k(t)\right] dt \le \varepsilon_1,$$

где $\psi^k(t)$ — решение задачи Коши (1.5). Значение левой части второго неравенства удобно находить при выполнении операции 2). Значения ε и

ε_1 определяют точность решения задачи (1.1), (1.5), (1.6) и должны задаваться с учетом погрешностей счета величин, стоящих в левых частях неравенств. Поскольку на практике задать "достижимое" значение ε_1 не всегда удается, во избежание зацикливания обычно применяется первое неравенство, выполнение которого свидетельствует о прекращении сходимости релаксационного процесса.

Недостаток данного алгоритма состоит в том, что его реализация требует получения и хранения базисного набора решений сопряженной системы, составляющего фундаментальную матрицу. Однако если размерность задачи такова, что имеющийся объем памяти позволяет реализовать описанный алгоритм, то такой подход будет вполне оправданным, поскольку задачи Коши, решаемые в п. 1) и 3), значительно проще, чем задачи Коши (1.1), а число итераций, выполняемых методом условного градиента для получения решения с заданной точностью, обычно бывает достаточно большим. Если необходимый объем памяти $rnN + \nu n > V$, где V — допустимый объем, $\nu \geq 2$, то можно реализовать традиционный вариант метода условного градиента, в котором вместо операций 1) и 3) будет интегрироваться система (1.1) при управлениях $u^k(t)$ и $\bar{u}^k(t)$ соответственно. Тогда потребуется всего $rN + \nu n$ слов памяти. Трудоемкость итерации эквивалентна интегрированию $2n$ скалярных уравнений.

1.1. Минимизация нормы уклонения от заданного состояния

Рассмотрим задачу, в которой вместо (1.4) будет минимизироваться функция, измеряющая уклонение конечного состояния системы от заданного состояния x^T:

$$\left\|x(t_1) - x^T\right\| \to \min, \tag{1.9}$$

где $x(t_1)$ — решение системы (1.1) при $t = t_1$, соответствующее управлению из класса допустимых (1.6). Решение этой задачи, очевидно, можно получить, применяя в приведенном выше алгоритме вместо операции одномерного поиска 4) следующую формулу:

$$\alpha_k = \begin{cases} \alpha^* = \dfrac{(x^T - x^k(t_1))'(\bar{x}^k(t_1) - x^k(t_1))}{\left\|\bar{x}^k(t_1) - x^k(t_1)\right\|^2}, & \text{если} \quad \alpha^* \leq 1; \\ 1, \quad \text{если} \quad \alpha^* > 1. \end{cases} \tag{1.10}$$

Известно, что метод условного градиента обладает медленной сходимостью, поэтому в его развитие предложены алгоритмы квадратичного программирования [9, 11], которые позволяют получить решение задачи (1.9) за существенно меньшее число итераций.

1.2. Метод выпуклых оболочек

Основная идея рассматриваемого алгоритма состоит в том, что для уменьшения целевой функции (1.9) максимально используются результаты наиболее трудоемкой операции — решения задачи Коши, полученные на нескольких итерациях. Вместо выпуклой оболочки двух точек x^k и $\bar{x}^k$, на которой отыскивается новое приближение в методе условного градиента, здесь применяется выпуклая оболочка s, $2 \leq s \leq n+1$, точек множества достижимости. Для поиска минимума функции (1.9) на выпуклой оболочке s точек решается простейшая задача квадратичного программирования. Приближение, полученное в результате решения этой задачи, значительно лучше приближения, полученного на отрезке методом условного градиента.

Алгоритм начинает работу с допустимого управления u^1, и на его первой итерации, как и в методе условного градиента, новое приближение строится в виде выпуклой комбинации u^1 и управления (1.7), вычисленного на решении системы (1.5) при

$$\psi(t_1) = g^1 = \left(x^1(t_1) - x^T\right) / \left\|x^1(t_1) - x^T\right\| .$$

Параметр α_k вычисляется по формуле (1.10).

Если $0 < \alpha_k < 1$, то управления u^1 и $u^2 = \bar{u}^1$, а также соответствующие им точки $x^1 = x^1(t_1)$ и $x^2 = \bar{x}^1$ дополняются управлением (1.7), найденным при

$$\psi(t_1) = g^2 = \frac{x^1 + \alpha_k(x^2 - x^1) - x^T}{\|x^1 + \alpha_k(x^2 - x^1) - x^T\|},$$

и соответствующей ему точкой $x^3 = \bar{x}(t_1)$. Следующее приближение отыскивается на выпуклой оболочке трех точек, содержащей отрезок, на котором ищется новое приближение методом условного градиента. Следовательно, приращение целевой функции (1.9) по модулю будет не меньше чем приращение, достигаемое методом условного градиента. Практика показывает, что метод выпуклых оболочек позволяет найти решение задачи (1.9) за существенно меньшее машинное время.

Опишем ν-ю итерацию алгоритма. Пусть на его итерациях получены управления $u^1, u^2, \ldots, u^k$, $k \leq n+1$, и им соответствующие точки x^i, $i = \overline{1,k}$, с помощью которых представляется приближение

$$u^\nu = \sum_{i=1}^{k} \beta_i u^i \quad \text{и} \quad x^\nu = \sum_{i=1}^{k} \beta_i x^i,$$

где $\beta_i \geq 0$, $i = \overline{1,k}$, $\sum\limits_{i=1}^{k} \beta_i = 1$. Тогда шаг алгоритма состоит из следующих операций:

1) на плоскости, проходящей через точки x^i, $i=\overline{1,k}$, отыскивается ближайшая к x^T точка y. Для этого решается задача безусловной минимизации

$$\left\| \left[x^k + \sum_{i=1}^{k-1} \alpha_i (x^i - x^k) \right] - x^T \right\| \to \min,$$

решение которой удовлетворяет линейной системе алгебраических уравнений

$$\sum_{j=1}^{k-1} \alpha_j (x^i - x^k)'(x^j - x^k) + (x^i - x^k)'(x^k - x^T) = 0, \quad i = \overline{1, k-1}. \quad (1.11)$$

С помощью решения данной системы и $\alpha_k = 1 - \sum\limits_{i=1}^{k} \alpha_i$ легко строится искомая точка на плоскости: $y = \sum\limits_{i=1}^{k} \alpha_i x^i$;

2) если среди α_i, $i=\overline{1,k}$, имеются отрицательные и, следовательно, точка y не принадлежит выпуклой оболочке точек x^i, $i=\overline{1,k}$, то на отрезке, содержащем точки y и x^ν, отыскивается точка пересечения отрезка с гранью выпуклой оболочки: $z = y + \sigma(x^\nu - y)$, где $\sigma = \max\limits_{\alpha_i<0} \{\alpha_i \,/\, (\alpha_i - \beta_i)\} = \alpha_{i_0} \,/\, (\alpha_{i_0} - \beta_{i_0}))$, или $z = \sum\limits_{i=1}^{k} \gamma_i x^i$, $\gamma_i = \alpha_i + \sigma(\beta_i - \alpha_i)$. Очевидно, $\gamma_{i_0} = 0$, поэтому точка x^{i_0} и управление u^{i_0} исключаются из дальнейшего рассмотрения;

3) если число оставшихся точек $s \le 2$, то $k = k-1$ и повторяется операция 1), а при наличии $\alpha_i < 0$ — операция 2). При $s = 1$ выполняется операция 5);

4) если после выполнения операции 1) имеем $\alpha_i > 0$, $i = \overline{1,s}$, $s \le k$, то точка y — решение задачи на выпуклой оболочке точек x^i, $i=\overline{1,k}$, а управление, приводящее в эту точку, найдется по формуле

$$u(t) = \sum_{i=1}^{s} \alpha_i u^i(t), \quad t \in [t_0, t_1]; \quad (1.12)$$

5) оставшиеся точки x^i, $i=\overline{1,s}$, дополняются новой точкой $x^{s+1} = \bar{x}(t_1)$, соответствующей управлению (1.7), посчитанному на решении задачи Коши (1.5) при $\psi(t_1) = (y - x^T) \,/\, \left\|y - x^T\right\|$, и повторяются операции 1)–5) при $k = s+1$. Процесс прекращается при выполнении неравенства

$$\left\|y - x^T\right\|^2 - \left(g^{s+1}\right)' \left(\bar{x}(t_1) - x^T\right) \le \varepsilon,$$

которое следует проверять после окончания операции 5) (ε — заданная точность решения задачи (1.9)). Найденная при этом точка y является проекцией x^T на множество достижимости: $y = \arg\min \left\|x - x^T\right\|$.

Управление (1.12), приводящее систему (1.1) в точку y, будет оптимальным для задачи (1.9).

Трудоемкость итерации описанного алгоритма незначительно больше, чем у метода условного градиента, поскольку вычисление проекции точки x^T на выпуклую оболочку заданных точек не содержит трудоемких операций интегрирования систем. Между тем число внешних итераций, выполняющих интегрирование прямой и сопряженной систем, существенно сокращается, и, следовательно, алгоритм требует значительно меньше времени центрального процессора для решения задачи (1.9) с заданной точностью.

Требования алгоритма к памяти определяются в основном числом управлений u^i, $i = \overline{1,s}$, и их размерностью r. Учитывая релейную структуру управлений (1.7), для хранения их в памяти достаточно запомнить лишь точки переключения каждой управляющей функции и ее начальных значений: $u^i_j(t_0)$, $j = \overline{1,r}$, $i = \overline{1,s}$, что требует $(n+2)$ слов памяти. Кроме того, необходим объем памяти для хранения управления (1.12), точек x^i, $i = \overline{1,s}$, $s \le n+1$, а также векторов, с помощью которых реализуются решения линейной системы алгебраических уравнений и численное интегрирование. Таким образом, необходимый объем памяти составляет $rN + \mu n + (n+1)n + (n+2)r$, $\mu \ge 5$, слов.

1.3. Алгоритм, основанный на теореме о минимаксе

Для решения задачи (1.9) с ограничениями на управление (1.6) построим алгоритм, основанный на теоремах об отделимости и о минимаксе [3]. Если множество достижимости $X(l)$ строго выпукло и содержит точку x^T, то данный алгоритм позволяет получить управление, которое с минимальной интенсивностью $l^* \le l$ переводит систему (1.1) в точку x^T, и следовательно, на этом управлении функционал (1.9) достигает абсолютного минимума. В случае, когда $x^T \notin X(l)$, данный алгоритм найдет управление, которое переводит систему (1.1) из x^0 в точку множества $X(l)$, ближайшую к точке x^T.

Пусть при $u(t) \equiv 0$ решена задача Коши (1.1), найден вектор $c = x(t_1)$ и задано первое приближение

$$g^1 = \left(x(t_1) - x^T\right) / \left\|x(t_1) - x^T\right\| .$$

Тогда $\lambda(0, g^1) = g^{1'}(x(t_1) - x^T) > 0$ и итерация алгоритма будет состоять из следующих операций:

1) при заданном l вычисляется значение функции

$$\lambda(l, g^1) = g^{1'}(x(g^1) - x^T);$$

2) если $\lambda(l, g^1) < 0$, то находится $l_1 < l$, при котором $\lambda(l_1, g^1) = 0$:

$$l_1 = l\ \frac{\lambda(0, g^1)}{\lambda(0, g^1) - \lambda(l, g^1)},$$

полагаем $k = 1$;

3) Выполнив операцию максимизации, найдем вектор $g^{k+1} = g(\alpha_k)$ такой, что $\lambda(l_k, g^{k+1}) > 0$;

4) находится новое значение

$$l_{k+1} = \frac{g^{k'}(c - x^T)}{g^{k'}(c - x^T) - \lambda(l_k, g^k)}\ l_k,$$

для которого $\lambda(l_{k+1}, g^{k+1}) = 0$;

5) если $l_{k+1} < l$ и $l_{k+1} - l_k > \varepsilon_l$, то повторяются операции 3) и 4). При $l_{k+1} - l_k \le \varepsilon_l$ решение задачи прекращается, и минимальная интенсивность управления полагается равной l_{k+1};

6) если $l_{k+1} > l$ и, следовательно, $\lambda(l, g^k) > 0$, то процесс расширения множества достижимости $X(l)$ прекращается и в дальнейшем при заданном l повторяются итерации основного алгоритма до выполнения неравенства

$$\left\| x(g^k) - x^T \right\| - \lambda(l, g^k) \le \varepsilon. \tag{1.13}$$

Управление, с которым на последней итерации найдена точка $x(g^k)$, будет оптимальным.

Сходимость данного алгоритма имеет место лишь в случае строго выпуклого множества $X(l)$. Поэтому в общем случае возможно зацикливание итерационного процесса. Чтобы избежать зацикливания, следует ввести дополнительное условие прекращения процесса

$$\lambda(l, g^{k+1}) - \lambda(l, g^k) \le \varepsilon_\lambda,$$

где ε_λ — оценка погрешностей вычисления функции $\lambda(l, g)$. Если это условие выполняется при нарушении неравенства (1.13), то данный алгоритм не может обеспечить решение задачи (1.9) и для дальнейшего поиска оптимального управления необходимо использовать изложенные выше алгоритмы.

2. Алгоритмы решения задачи быстродействия

Рассмотрим задачу линейного быстродействия: перевод системы (1.1) из точки x^0 в точку x^T при ограничениях на управление (1.6) за наименьшее время.

2.1. Минимаксный алгоритм

Один из первых алгоритмов [10], предложенных для решения задачи линейного быстродействия, основан на теореме об отделимости выпуклых множеств. По своей конструкции он близок к методу минимизации интенсивности управления [2; 6]. Вместо наименьшего значения параметра $l = l^*$ алгоритмы быстродействия отыскивают минимальный отрезок времени $t^* - t_0$, за который переводима система (1.1) из x^0 в x^T при заданных ограничениях на управление. Предполагая управляемость системы (1.1), построим алгоритм, который генерирует сходящуюся возрастающую последовательность моментов времени $t_1, t_2, \ldots$ с двумя основными свойствами:

1) вдоль последовательности $\{t_k\}$ множество достижимости

$$X(t_k) = \Big\{ x \in R^n : \ x = c + \int\limits_{t_0}^{t_k} \Phi(t_k, t) B(t) u(t) dt,$$

$$|u_j(t)| \le l, \ \ j = \overline{1, r}, \ \ t \in [t_0, t_k] \Big\}$$

расширяется ($X(t_k) \subset X(t_{k+1})$),

2) в момент $t^* = \lim\limits_{k \to \infty} t_k$ происходит первое касание точки x^T.

Для определения значений t_k, $k = 1, 2, \ldots$, будем применять следующее условие:

$$\psi(t_k)' \left(x(t_k) - x^T \right) = 0,$$

проверка которого требует одновременного интегрирования систем (1.1) и (1.5) в прямом времени.

При каждом фиксированном значении t_k корректировка вектора g — нормали опорной плоскости — осуществляется посредством решения задачи $\lambda(g, t_k) \to \max\limits_{\|g\| \le 1}$. Иными словами, полагается

$$g^{k+1} = \arg \max_{\|g\| \le 1} \lambda(g, t_k).$$

Введем множества

$$G_+^k = \{ g \in R^n : \ \ \lambda(g, t_k) > 0, \ \ \|g\| = 1 \},$$

$$G_0^k = \{ g \in R^n : \ \ \lambda(g, t_k) = 0, \ \ \|g\| = 1 \}.$$

Пусть $g^0 \in G_+^0$. Тогда k-й шаг минимаксного алгоритма состоит из следующих операций:

1) при управлении (1.7) в прямом времени интегрируется система

$$\dot{x} = A(t)x + \sum_{j=1}^{r} b_j(t) \bar{u}_j(t) + f(t), \quad x(t_k) = \bar{x}(t_k) = x(g_k, t_k),$$

$$\dot{\psi} = -A(t)'\psi, \quad \psi(t_k) = g^k, \quad t \ge t_k, \tag{2.1}$$

до первого корня t_{k+1} функции $\psi(t)'(x(t) - x^T)$:

$$\psi(t_{k+1})'(x(t_{k+1}) - x^T) = 0;$$

2) при фиксированном t_{k+1}, начиная с $g = \psi(t_{k+1})$ выполняется несколько шагов градиентного метода. В результате находится $g^{k+1} \in G_{+}^{k+1}$.

Операции 1), 2) повторяются, пока градиентный метод обеспечивает выполнение неравенства $\lambda(g^{k+1}, t_{k+1}) > 0$, а интегрирование системы (2.1) позволяет получить $t_{k+1} > t_k$. Для управляемой системы данный итерационный процесс сходится к (g^*, t^*):

$$\lambda(g^*, t^*) = \min_{x(t^*) \in X(t^*)} \left\| x(t^*) - x^T \right\| = 0, \quad \lambda(g^*, t^*) = \max_{g \in G_{+}^{k}} \lambda(g, t^*).$$

При этом управление (1.7), посчитанное на решении системы (1.5) при $\psi(t_1) = g$, будет оптимальным по быстродействию.

Недостатками минимаксного алгоритма являются, с одной стороны, необходимость интегрирования сопряженной системы в прямом времени и, с другой — гарантированная вычислительная устойчивость только для строго выпуклых множеств достижимости.

Трудоемкость итерации алгоритма оценивается трудоемкостью интегрирования $2n(\nu + 1)$ скалярных уравнений на отрезке $[t_0, t_k]$, где ν — число пересчетов $\lambda(g, t_k)$, при вычислении параметра α_k. Требуемый объем памяти — γn слов, $\gamma \leq 5$.

2.2. Алгоритм выпуклых оболочек

Рассмотрим алгоритм, в котором вместо градиентных процедур для попадания в область G_{+}^{k} применяется итерационная процедура из раздела 1. Тогда в качестве вектора g берется приближенное решение задачи (1.9):

$$g^k = \left(x^k(t_k) - x^T \right) \Big/ \left\| x^k(t_k) - x^T \right\| . \qquad (2.2)$$

Поскольку $\max\limits_{\|g\| \leq 1} \lambda(g, t_k) = \min\limits_{x \in X(t_k)} \left\| x - x^T \right\|$, при $x^k(t_k) \neq x^T$ найдется такой номер k, для которого g^k вида (2.2) будет принадлежать G_{+}^{k}. В противном случае $x^T \in X(t_k)$ и при $g^k \in G_0^k$ вычисляется оптимальное управление (1.7).

Пусть $g^0 = (x^0 - x^T) \,/\, \left\| x^0 - x^T \right\|$, $k = 0$. Тогда итерация алгоритма состоит из следующих операций:

1) начиная с $x(t_k) = \bar{x}(t_k)$, $\psi(t_k) = g^k$ интегрируется $2n$-мерная система (2.1) от t_k до ближайшего к нему корня t_{k+1} уравнения

$$\psi(t)'(x(t) - x^T) = 0;$$

2) задается входная информация для алгоритма выпуклых оболочек из разд. 1.2: $x^1 = \bar{x}(t_k)$, $u^1 = \bar{u}(t)$ — управление (1.7), приводящее (1.1) в $\bar{x}(t_k)$.

Затем при $\psi(t_k) = (x^1 - x^T) \; / \left\| x^1 - x^T \right\|$ интегрируется система (1.5) и по формуле (1.7) считается управление $u^2 = u(t)$. Решение системы, соответствующее $u^2(t)$, при $t = t_1$ берется в качестве x^2;

3) начиная с $k = 2$ реализуются итерации алгоритма выпуклых оболочек из разд. 1.2, пока не выполнится неравенство

$$\left(g^{k+1}\right)' (\bar{x}(t_k) - x^T) = \lambda(g^{k+}, t_k) > 0,$$

где $g^{k+1} = (y^\nu - x^T) \; / \left\| y^\nu - x^T \right\|$, y^ν — полученное приближение для задачи (1.9). Затем k меняется на $k+1$ и повторяются операции 1)–3).

Сходимость этого процесса следует из сходимости метода условного градиента и имеет место для произвольного выпуклого множества.

Итерационный процесс прекращается при выполнении неравенства $\left\| y^\nu(t_k) - x^T \right\| < \varepsilon$. Тогда оптимальное управление считается по формуле (1.12).

Решение практических задач показывает также, что данный алгоритм требует меньше времени для получения оптимального по быстродействию управления, переводящего систему (1.1) из x^0 в x^T с заданной точностью. Одним из недостатков алгоритма является неустойчивость операции прямого интегрирования сопряженной системы при получении значений t_k, $k = 1, 2, \ldots$. Для некоторых систем этот недостаток может оказаться существенным и требует специальных способов интегрирования системы (2.1).

Заключение

Получение численного решения задачи оптимального управления при наименьших затратах вычислительных ресурсов особенно актуально в управляемых системах реального времени, например в бортовых системах управления летательными аппаратами. Путем многократного решения задачи оптимального программного управления в некоторых случаях можно получить решение задачи синтеза оптимального управления. Так, с помощью оптимальных по быстродействию программных управлений, полученных для разных начальных состояний, в одной задаче преследования [8] построена линия переключения управления в фазовом пространстве.

Список литературы

1. Андреев Ю. Н. Управление конечномерными линейными объектами / Ю. Н. Андреев. – М. : Наука, 1976. – 424 с.

2. Васильев О.В. К численному решению задач линейного быстродействия / О. В. Васильев, А. И. Тятюшкин // Дифференц. и интегр. уравнения. — Иркутск : Изд-во Иркут. ун-та, 1973. – Вып. 2. – С. 57–69.
3. Габасов Р. Качественная теория оптимальных процессов / Р. Габасов, Ф. М. Кириллова. – М. : Наука, 1971. – 508 с.
4. Габасов Р. Конструктивные методы оптимизации. Ч. 1. Линейные задачи / Р. Габасов, Ф. М. Кириллова, А. И. Тятюшкин. – Минск : Университетское, 1984. – 214 с.
5. Калман Р. Очерки по математической теории систем / Р. Калман, П. Фалб, М. Арбиб. – М.: Мир, 1971. – 400 с.
6. Красовский Н. Н. Теория управления движением / Н. Н. Красовский. – М. : Наука, 1968. – 476 с.
7. Тятюшкин А. И. Многометодная технология оптимизации управляемых систем / А. И. Тятюшкин. – Новосибирск : Наука, 2006. – 343 с.
8. Тятюшкин А. И. Численное исследование свойств оптимального управления в одной задаче преследования / А. И. Тятюшкин, Б. Е. Федунов // Изв. РАН, ТиСУ. – 2005. – № 3. – С. 104–113.
9. Barr R. O. An efficient computational procedure for a generalized quadratic programming problem // J. SIAM Control. — 1969. — Vol. 7, N 3. — P. 415–429.
10. Eaton J. H. An iterative solution to time-optimal control // J. Math. Anal. and Appl. – 1962. – Vol. 5, N 2. – P. 329-344.
11. Gibert E. G. An iterative procedure for computing the minimum of a quadratic from on a convex set // J. SIAM Control. – 1966. – Vol. 4, N 1. – P. 61–80.
12. Tyatyushkin A. I. A multimethod technique for solving optimal control problem // Optimization Letters. – 2012. – N 7. – P. 1335–1347.

A. I. Tyatyushkin

Numerical Methods for Calculation of Time-Optimal Control

Abstract. The article is devoted to algorithms for calculation of time-optimal control in the linear systems. Proposed set of algorithms make possible apply multi-method technique to solving time-optimal control problems with different singularity. Corresponding to this technology the solution is found by a multimethods' algorithm consisting of a sequence of steps of different methods applied to the optimization process in order to accelerate it. Such a technology allows to consider some particularities of the problem of interest at all stages of its solution and to improve the efficiency of optimal control search.

Keywords: convex hull method, attainability set, time-optimal problem, minimax algorithm, linear controlled system.

References

1. Andreev N. Manage finite linear objects (in Russian). Moscow, Nauka 1976, 424 p.
2. Vasil'ev O.V., Tyatyushkin A.I. On the numerical solution of linear time-optimal control (in Russian). *Differential and Integral Equations*. Irkutsk, University Press, 1973, issue 2, pp. 57–69.
3. Gabasov R., Kirillova F.M. Qualitative theory of optimal processes (in Russian). Moscow, Nauka, 1971, 508 p.

4. Gabasov R., Kirillova F.M., Tyatyushkin A.I. Constructive Optimization Methods. Part 1: Linear problems (in Russian). Minsk, 1984, 214 p.
5. Kalman R., Falb P., Arbib M. Essays on mathematical systems theory. New York, Wiley, 1971 . 400 p.
6. Krasovskii N.N. Theory of motion control (in Russian). Moscow, Nauka, 1968, 476 p.
7. Tyatyushkin A.I., Fedunov B.E. Numerical study of the properties of optimal control in a pursuit problem (in Russian). *Math. RAS, Tisza.* 2005, no. 3, pp. 104–113.
8. Tyatyushkin A.I. Multimethod optimization technology controlled systems (in Russian). Novosibirsk, Nauka, 2006, 343 p.
9. Barr R.O. An efficient computational procedure for a generalized quadratic programming problem. *J.SIAM Control*, 1969, vol. 7, no. 3, pp. 415–429.
10. Eaton J.H. An iterative solution to time-optimal control. *J. Math. Anal. and Appl.*, 1962, vol. 5, no. 2, p. 329–344.
11. Gibert E.G. An iterative procedure for computing the minimum of a quadratic from on a convex set. *J. SIAM Control*, 1966, vol. 4, no. 1, pp. 61–80.
12. Tyatyushkin A.I. A multimethod technique for solving optimal control problem. *Optimization Letters*, 2012, vol. 7, pp. 1335–1347.

Tyatyushkin Alexander Ivanovich, Doctor of Sciences (Physics and Mathematics), Professor, Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS), 134, Lermontov st., Irkutsk, 664033,
(e-mail: tjat@icc.ru)