Математические заметки СВФУ
Январь—март, 2019. Том 26, № 1

УДК 517.977.58

УЛУЧШЕНИЕ СХОДИМОСТИ МЕТОДА НЕЙШТАДТА — ИТОНА

В. Г. Старов

Аннотация. Нейштадт и Итон разработали метод и численный алгоритм решения задачи оптимального быстродействия. В этом методе необходимо решать дискретное уравнение для нахождения начального значения сопряженной системы. В ходе решения этого уравнения на каждом шаге нужно находить величину приращения начального значения сопряженной системы. Это достигается путем выбора длины шага. При этом начальное значение сопряженной системы должно удовлетворять определенным условиям. Затем решается система дифференциальных уравнений, где в качестве начальных данных используются полученные значения сопряженной системы и определяется траектория. Таким образом, на каждом шаге решения дискретного уравнения необходимо использовать итерационный процесс. Доказана сходимость метода. При этом нужно отметить слабую сходимость метода. Для улучшения сходимости предлагается следующий подход. Делается несколько шагов метода Нейштадта — Итона. Полученные начальные значения сопряженной системы аппроксимируются и экстраполируются на определенном интервале. Затем снова делаем несколько шагов метода Нейштадта — Итона и при этом используем экстраполированные значения начальных значений сопряженной системы. Таким образом, предлагается «скользящая» аппроксимация найденных решений дискретного уравнения для каждой компоненты с помощью алгебраической функции с последующей ее экстраполяцией. Экстраполяция позволяет значительно уменьшить вычислительные затраты.

DOI: 10.25587/SVFU.2019.101.27248

Ключевые слова: аппроксимация, экстраполяция, итерация, оптимальное управление, сопряженная система.

1. Постановка задачи и краткое изложение метода Нейштадта — Итона

Задачи линейного быстродействия представляют значительный практический интерес. Решением этих задач занимались многие исследователи, которые разработали различные методы (см., например, [1–15]). Эти методы численные, поэтому для них актуально уменьшение трудоемкости (времени счета). Настоящая работа является продолжением исследований, изложенных в [16].

Пусть управляемая система описывается линейным дифференциальным уравнением

$$\dot{x} = Ax + Bu, \quad x(t_0) = x_0. \tag{1}$$

Здесь x — n-мерный вектор фазового состояния, A и B — матрицы размеров $n \times n$ и $n \times m$ соответственно, u — m-мерный вектор управления, компоненты которого принадлежат классу кусочно непрерывных функций и подчинены ограничениям $|u_j| \leq M_j$, $M_j > 0$, $j = \overline{1,m}$. Предполагается, что линейная система полностью управляема и переводима в начало координат ограниченным управлением, т. е. x_0 принадлежит области управляемости V.

Задача. *Найти управление u, переводящее систему* (1) *из начального состояния $x(t_0)$ в начало координат $x(t_k) = 0$ за минимальное время $T = t_k - t_0$.*

Приближенное решение этой задачи может быть получено методом Нейштадта — Итона. В [2] приводится описание этого метода. Согласно [2] функция Гамильтона H принимает в данном случае следующий вид:

$$H(\psi, x, u) = \psi(Ax + Bu),$$

а сопряженная система уравнений принимает вид

$$\dot{\psi} = -A'\psi. \tag{2}$$

Пусть $p = (p_1, \dots, p_n)$ — отличный от нуля вектор. Обозначив через $\psi(t,p)$ решение уравнения (2) с начальным условием $\psi(0,p) = p$, а через $u(t,p)$ — управление, соответствущее функции $\psi(t,p)$, в силу условия максимума имеем

$$\psi(\tau)Bu(\tau) = \max_{u \in U} \psi(\tau)Bu.$$

Это управление будем рассматривать на отрезке $0 \leq t \leq T$. Через $x(t)$ обозначим траекторию, соответствующую управлению $u(t,p)$ и удовлетворяющую конечному условию $x(T) = 0$. Начальную точку $x(0)$ этой траектории обозначим через $\xi_T(p) = \big(\xi_T^1(p), \dots, \xi_T^n(p)\big)$. Траектория $x(t)$ — оптимальная траектория, переводящая точку $x_0 = \xi_T(p)$ в начало координат за время T. Выпишем решение дифференциального уравнения (1) в момент T, учитывая, что $x(T) = 0$:

$$\sum_{i=1}^{n} \varphi_i(T) \left[\xi_T^i(p) + \int\limits_0^T (\psi^i(\tau)Bu(\tau,p))d\tau \right] = 0.$$

Выражения в квадратных скобках обращаются в нуль в силу линейной независимости векторов $\varphi_1(T), \dots, \varphi_n(T)$, поэтому

$$\xi_T^i(p) = -\int\limits_0^T (\psi^i(\tau)Bu(\tau,p))d\tau, \quad i = 1, 2, \dots, n. \tag{3}$$

Таким образом, задавая n-мерный (отличный от нуля) вектор p, получаем точку $\xi_T(p)$, определяемую формулой (3).

Метод Нейштадта — Итона позволяет найти вектор p, который дает решение задачи (1). Опишем вкратце алгоритм вычисления вектора p.

Вводится в рассмотрение функция

$$f(t,p) = p(x_0 - \xi_t(p)) \tag{4}$$

и показывается, что она непрерывна по t, p. При любом фиксированном p она непрерывно возрастает и существует единственное t, $0 \leq t \leq t_k$, для которого $f(t,p) = 0$. Обозначим это значение t через $F(p)$.

Рассматривается дифференциальное уравнение

$$\frac{dp}{d\tau} = -[x_0 - \xi_{F(p)}(p)],$$

здесь $\xi_{F(p)}(p)$ — решение уравнения (3) в момент времени $t = F(p)$.

Доказывается, что решение $p(\tau)$ при $\tau \longrightarrow \infty$ дает решение задачи (1).

Итон разработал итерационный процесс для нахождения решения этого уравнения. Он заменил дифференциальное уравнение разностным

$$\frac{p_{k+1} - p_k}{\Delta\tau_k} = -[x_0 - \xi_{F(p)}(p)], \quad \text{или} \quad p_{k+1} = p_k - \Delta\tau_k[x_0 - \xi_{F(p)}(p)],$$

далее проводится нормирование, т. е.

$$p_{k+1} = \frac{q_{k+1}}{|q_{k+1}|}, \quad \text{где} \quad q_{k+1} = p_k - \Delta\tau_k[x_0 - \xi_{F(p)}(p)].$$

В окончательном виде алгоритм расчета p имеет следующий вид. Положим

$$q_{k+1}^{(m)} = p_k - 2^{-m}(x_0 - \xi_{F(p_k)}(p_k)) \tag{5}$$

и выберем наименьшее из неотрицательных целых чисел $m = 0, 1, 2, 3, \ldots$, для которых вектор $p_{k+1} = \frac{q_{k+1}^m}{|q_{k+1}^m|}$ удовлетворяет неравенству

$$p_{k+1}(x_0 - \xi_{F(p_k)}(p_{k+1})) < -2^{-(m+1)}|(x_0 - \xi_{F(p_k)}(p_k))|^2. \tag{6}$$

В качестве p_0 предлагается принять вектор $-x_0/|x_0|$.

В [4] отмечается слабая сходимость метода.

2. Модификация метода

Значение m, которое удовлетворяет неравенству (6), обозначим через m_r. Делаем несколько итераций алгоритма Итона и определяем максимальное изменение среди значений компонент вектора p на k-й и $(k-1)$-й итерациях, $k = 1, 2, 3, \ldots$. Если это изменение меньше текущей компоненты по абсолютной величине, то запоминаем значение m_r на k-й итерации. Обозначим его через m_h, а номер итерации обозначим через k_h. Далее делаем l итераций, где $l = n$, а n — размерность вектора x в системе (1). При этом m_r определяется на итерациях $k_h + 1$, $k_h + 2, \ldots$. Запоминаем значения m_r на каждой итерации. Суммируем их и определяем среднее значение:

$$m_{sr} = \sum_{k=k_h+1}^{k_h+l} m_{rk}/l.$$

Затем делаем следующие l итераций, где в качестве m_h берется целая часть среднего значения, полученного на предыдущих l итерациях. На этих итерациях снова определяем m_{sr}. Таким образом, в качестве m_h берем «скользящее» среднее, полученное на предыдущих l итерациях.

Делаем $2 \times l$ итераций, начиная с $k_h + 1$. На всех этих итерациях суммируем $\Delta\tau_k$, которые удовлетворяют неравенству (6). Определяем суммарное время

$$D\tau_s = \sum_{k=k_h+1}^{k_h+2\times l} \Delta\tau_k.$$

Начиная с итерации $k_h + 1 + l$ и по $k_h + 2 \times l$-ю запоминаем значения всех компонент вектора p на каждой итерации, а также времена τ_k. Обозначим через τ_q время на $k_h + 2 \times l$ итераций. Отметим, что числа $F(p_k)$ образуют монотонно возрастающую последовательность, сходящуюся к числу T — времени оптимального быстродействия для задачи (1). Запоминаем время, полученное на $k_h + 2 \times l$ итераций. Обозначим его через T_q.

Представим каждую компоненту вектора p как заданную функцию $g_i(\tau_k)$, $i = \overline{1, n}$, проходящую через запомненные значения в моменты времени τ_k. Предлагается аппроксимировать уравнения этих функций для каждой компоненты алгебраической зависимостью (полиномом), а затем эти полиномы экстраполировать. Интервал экстраполяции $\mathscr{L}_v$ будем увеличивать на величину $\Delta\tau = D\tau_s \times n$, где n — размерность вектора x в системе (1). Уменьшать интервал экстраполяции будем способом половинного деления, т. е. $\mathscr{L}_d = D\tau_s/d$, где $d = 2, 4, 8, \ldots$. Обозначим экстраполированные значения через p_w и будем подставлять их в формулу (4). Таким образом осуществляется «скользящая» аппроксимация и экстраполяция решений уравнения (5). После каждой итерации будем определять норму отклонения $\epsilon_t = |x_0 - \xi_{F(p_k)}(p_k)|$ и сравнивать ее с заданной ϵ_k. Если $\epsilon_t < \epsilon_k$, то решение найдено. Введем также логические переменные n_ϕ и n_e.

2.1. Квадратичная аппроксимация функций. При точечной квадратичной аппроксимации в качестве меры отклонения полинома

$$Q_c(z) = a_0 + a_1 z + \ldots, + a_c z^c$$

от заданной функции $y = f(z)$ на множестве точек $z_0, z_1 \ldots, z_l$ принимаем величину

$$S = \sum_{j=1}^{l} [Q_c(z_j) - f(z_j)]^2,$$

называемую *квадратичным отклонением.* Для построения аппроксимирующего полинома требуется подобрать коэффициенты $a_0, a_1 \ldots, a_c$ так, чтобы величина S была наименьшей.

Для решения задачи аппроксимации нужно воспользоваться общим приемом дифференциального исчисления. А именно, найдем частные производные от величины S по всем переменным $a_0, a_1 \ldots, a_c$. Приравнивая эти частные переменные к нулю, получим для определения неизвестных $a_0, a_1 \ldots, a_c$ систему $c + 1$ уравнений с $c + 1$ неизвестными.

Для простоты в качестве полинома взята линейная модель

$$y = a_0 + a_1 z. \tag{7}$$

Коэффициенты a_0 и a_1 определяются из системы уравнений

$$a_0 s_0 + a_1 s_1 = t_0,$$
$$a_0 s_1 + a_1 s_2 = t_1,$$

где

$$s_0 = \sum_{j=1}^{l} 1, \quad s_1 = \sum_{j=1}^{l} z_j, \quad s_2 = \sum_{j=1}^{l} z_j^2, \quad t_0 = \sum_{j=1}^{l} y_j, \quad t_1 = \sum_{j=1}^{l} z_j y_j,$$

l — количество точек.

2.2. Алгоритм модифицированного метода. Разработана следующая схема алгоритма решения задачи (1).

Задаем вектор $p_0 = -x_0/|x_0|$. Определяем номер итерации k_h и величину m_h. Также даем n_e значение нуль и n_ϕ — нуль.

Шаг 1. Делаем $2 \times l$ итераций, начиная с $k_h + 1$. Определяем величину временного интервала $D\tau_s$, а также время τ_q. Запоминаем значения всех компонент вектора p на соответствующих итерациях, а также времена τ_k и T_q.

Шаг 2. Для каждой компоненты вектора p определяем аппроксимирующий полином по формуле (7).

Шаг 3. Интервалам $\mathscr{L}_v$ и $\mathscr{L}_d$ присваиваем значение $D\tau_s$. Экстраполируем все полиномы на величину интервала $\mathscr{L}_v$ от времени τ_q, т. е. определяем вектор $p_w = p(\tau_q + \mathscr{L}_v)$.

Шаг 4. Подставляем экстраполированные значения вектора p_w в формулу (4) и определяем $F(p_w)$.

Шаг 5. Если n_ϕ меньше единицы, то переходим к шагу 6, иначе к шагу 9.

Шаг 6. Если $F(p_w)$ больше T_q, то переходим к шагу 7, иначе к шагу 8.

Шаг 7. T_q присваиваем значение $F(p_w)$ и запоминаем. Запоминаем вектор p_w. Увеличиваем n_e на единицу, увеличиваем интервал $\mathscr{L}_v$ на величину $\Delta\tau$ определяем новое значение вектора $p_w = p(\tau_q + \mathscr{L}_v)$ и переходим к шагу 4.

Шаг 8. Если n_e меньше единицы, то переходим к шагу 9. Если n_e больше нуля, то вектору p_w и времени T_q присваиваем значения, полученные на шаге 7. Номер итерации k_h увеличиваем на величину $2 \times l$ и переходим к шагу 10.

Шаг 9. Если $F(p_w)$ больше T_q, то присваиваем T_q значение $F(p_w)$, номер итерации k_h увеличиваем на величину $2 \times l$ и переходим к шагу 10. Если $F(p_w)$ меньше T_q, то уменьшаем интервал $\mathscr{L}_d$ и определяем вектор $p_w = p(\tau_q + \mathscr{L}_d)$. Увеличиваем n_ϕ на единицу и переходим к шагу 4.

Шаг 10. Если $\epsilon_t < \epsilon_k$, то решение найдено, иначе даем n_e и n_ϕ значение нуль и переходим к шагу 1.

3. Численное моделирование

Для подтверждения эффективности предлагаемой модификации метода были проведены численные исследования. Они проводились на системах линейных дифференциальных уравнений 4-го порядка.

Под эффективностью понимается отношение времени суммарного интегрирования на всех итерациях модифицированного алгоритма к суммарному времени интегрирования методом Нейштадта — Итона в процентах.

В приводимых примерах величины $b = 4$, $M = 5$. В таблицах величины x_{01}, x_{02}, x_{03}, x_{04} — начальные значения фазовых координат системы (1).

Из начальных данных составляем различные комбинации, которые отличаются друг от друга нулевыми значениями и их расположением. Ненулевые начальные данные изменяются в интервале $1 \div 20$.

Введем обозначения:

N_{var} — количество изменений начальных данных в данной комбинации (варианты); для каждого варианта определяется величина ϵ_k как результат счета методом Нейштадта — Итона;

$T_{\Sigma N}$ — суммарное время интегрирования на всех итерациях метода Нейштадта — Итона для одного варианта начальных данных;

$T_{\Sigma M}$ — суммарное время интегрирования на всех итерациях модифицированного метода для одного варианта начальных данных;

$$R_k = \left(\left(\sum_{g=1}^{N_{var}} \frac{T^g_{\Sigma M}}{T^g_{\Sigma N}} \times 100 \right) / N_{var} \right)$$

— средняя эффективность комбинации в %.

Пример 1

$$\begin{array}{lll}
\dot{x_1} = x_2, & & x_1(t_0) = x_{10}, \\
\dot{x_2} = x_3, & & x_2(t_0) = x_{20}, \\
\dot{x_3} = x_4, & & x_3(t_0) = x_{30}, \\
\dot{x_4} = bu, & |u| \le M, & x_4(t_0) = x_{40}.
\end{array}$$

x_0				N_{var}	R_k
0.0	0.0	0.0	x_{40}	10	3.766
x_{10}	x_{20}	0.0	0.0	14	7.242
x_{10}	0.0	0.0	x_{40}	14	9.967
x_{10}	0.0	0.0	0.0	13	8.559
0.0	x_{20}	x_{30}	0.0	14	5.675
0.0	x_{20}	0.0	0.0	12	6.591
0.0	0.0	x_{30}	x_{40}	14	6.995
0.0	0.0	x_{30}	0.0	12	7.585
x_{10}	x_{20}	x_{30}	x_{40}	16	4.897

Пример 2

$$
\begin{aligned}
&\dot{x_1} = x_2, && x_1(t_0) = x_{10},\\
&\dot{x_2} = x_3, && x_2(t_0) = x_{20},\\
&\dot{x_3} = x_4, && x_3(t_0) = x_{30},\\
&\dot{x_4} = a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 + bu, && x_4(t_0) = x_{40},
\end{aligned}
$$

$$\lambda_{12} = -0.11 \pm 1.268, \quad \lambda_{34} = -0.55 \pm 0.077, \quad |u| \leq M.$$

x_0				N_{var}	R_k
0.0	0.0	0.0	x_{40}	13	6.358
x_{10}	x_{20}	0.0	0.0	11	8.690
x_{10}	0.0	0.0	x_{40}	11	8.279
x_{10}	0.0	0.0	0.0	12	8.417
0.0	x_{20}	x_{30}	0.0	14	7.307
0.0	x_{20}	0.0	0.0	12	7.409
0.0	0.0	x_{30}	x_{40}	12	6.429
0.0	0.0	x_{30}	0.0	12	8.583
x_{10}	x_{20}	x_{30}	x_{40}	11	7.363

Пример 3

$$\begin{aligned}
&\dot{x_1} = x_2, && x_1(t_0) = x_{10},\\
&\dot{x_2} = x_3, && x_2(t_0) = x_{20},\\
&\dot{x_3} = x_4, && x_3(t_0) = x_{30},\\
&\dot{x_4} = a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 + bu, && x_4(t_0) = x_{40}
\end{aligned}$$

$$\lambda_{12} = -0.045 \pm 1.118, \quad \lambda_3 = -0.635, \quad \lambda_4 = -0.126, \quad |u| \le M.$$

x_0				N_{var}	R_k
0.0	0.0	0.0	x_{40}	10	6.120
x_{10}	x_{20}	0.0	0.0	14	9.335
x_{10}	0.0	0.0	x_{40}	14	9.838
x_{10}	0.0	0.0	0.0	13	8.914
0.0	x_{20}	x_{30}	0.0	15	9,102
0.0	x_{20}	0.0	0.0	15	7.691
0.0	0.0	x_{30}	x_{40}	14	8.318
0.0	0.0	x_{30}	0.0	12	7.976
x_{10}	x_{20}	x_{30}	x_{40}	13	7.095

4. Заключение

Из приведенных примеров видно, что R_k меньше 10 %, т. е. вычислительные затраты в среднем уменьшены более чем на порядок по сравнению с методом Нейштадта — Итона.

Это свидетельствует об эффективности предложенного алгоритма для решения задачи линейного быстродействия для системы (1).

ЛИТЕРАТУРА

1. *Александров В. М.* Приближенное решение задачи линейного быстродействия // Автоматика и телемеханика. 1998. № 12. С. 3–13.
2. *Болтянский В. Г.* Математические методы оптимального управления. М.: Наука, 1969.
3. *Тятюшкин А. И.* Численные методы и программные средства оптимизации управляемых систем. Новосибирск: Наука, 1992.
4. *Федоренко Р. П.* Приближенное решение задач оптимального управления. М.: Наука, 1978.
5. *Шевченко Г. В.* Численный алгоритм решения линейной задачи оптимального быстродействия // Журн. вычисл. математики и мат. физики. 2002. Т. 42, № 8. С. 1184–1196.
6. *Александров В. М.* Последовательный синтез оптимального по быстродействию управления // Журн. вычисл. математики и мат. физики. 1999. Т. 39, № 9. С. 1464–1478.
7. *Александров В. М.* Сходимость метода последовательного синтеза оптимального по быстродействию управления // Журн. вычисл. математики и мат. физики. 1999. Т. 39, № 10. С. 1650–1661.

8. *Александров В. М.* Численный метод решения задачи линейного быстродействия // Журн. вычисл. математики и мат. физики. 1998. Т. 38, № 6. С. 918–931.
9. *Александров В. М.* Решение задач оптимального управления на основе метода квазиоптимального управления // Тр. Ин-та математики СО АН СССР. Т. 10. Модели и методы оптимизации. Новосибирск: Наука, 1988. С. 18–54.
10. *Жолудев А. И., Тятюшкин А. И., Эринчек Н. М.* Численные методы оптимизации управляемых процессов // Изв. АН СССР. Тех. кибернетика. 1989. № 4. С. 14–31.
11. *Пшеничный Б. Н., Соболенко Л. А.* Ускоренный метод решения задачи линейного быстродействия // Журн. вычисл. математики и мат. физики. 1968. Т. 8, № 6. С. 1343–1351.
12. *Дубовицкий А. Я., Рубцов В. А.* Линейные быстродействия // Журн. вычисл. математики и мат. физики. 1970. Т. 10, № 1. С. 216–223.
13. *Киселев Ю. Н.* Быстросходящиеся алгоритмы для линейного оптимального быстродействия // Кибернетика. 1990. Т. 62, № 6. С. 47–57.
14. *Срочко В. А.* Итерационные методы решения задач оптимального управления. М.: Физматлит, 2000.
15. *Дюркович Е.* Численный метод решения линейных задач быстродействия с оценкой точности // Докл. АН. 1982. Т. 265, № 4. С. 793–797.
16. *Старов В. Г.* Модификация метода Нейштадта — Итона // Мат. заметки СВФУ. 2014. Т. 21, № 3. С. 107–112.

Поступила в редакцию 19 сентября 2018 г.
После доработки 20 октября 2018 г.
Принята к печати 13 ноября 2018 г.

Старов Валентин Григорьевич
Институт математики им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4, Новосибирск 630090

Математические заметки СВФУ
Январь—март, 2019. Том 26, № 1

UDC 517.977.58

IMPROVEMENT OF NEUSTADT—EATON'S METHOD CONVERGENCE

V. G. Starov

Abstract: Neustadt and Eaton developed a method and an iterative algorithm for solving time-optimal control problems. According to the method, it is necessary to solve a discrete equation to find the initial value of the adjoint system. At each step of the algorithm it is needed to find the increment of the initial value of the adjoint system. This is achieved by choosing the step length, while the initial value must satisfy certain conditions. Then the system of differential equations is solved, where the obtained values are used as initial data and the trajectory is determined. Thus, the iterative process must be used on each step of solving the discrete equation. The convergence of this method was previously proved. However, the convergence is weak. To improve it, the following approach is proposed. After a few steps of Neustadt–Eaton's method the obtained initial values of the adjoint system are approximated and extrapolated at a certain interval. Then again we take a few steps of Neustadt–Eaton's method and use the extrapolated initial values of the adjoint system. Thus we propose "sliding" approximation of the discrete equation solutions for each component using an algebraic function followed by extrapolation. It is shown that such extrapolation significantly reduces computational costs.

DOI: 10.25587/SVFU.2019.101.27248

Keywords: approximation, extrapolation, iteration, optimal control, adjoint system.

REFERENCES

1. *Aleksandrov V. M.*, "An approximate solution of the linear time-optimality problem," Autom. Remote Control, **59**, No. 12, 1699–1707 (1999).
2. *Boltyanskii V. G.*, Mathematical Methods of Optimal Control, Holt, Rinehart and Winston (1971).
3. *Tyatyushkin A. I.*, Numerical Methods and Software for Controlled Systems Optimization [in Russian], Nauka, Novosibirsk (1992).
4. *Fedorenko R. P.*, Approximate Solution to Optimal Control Problems [in Russian], Nauka, Moscow (1978).
5. *Shevchenko G. V.*, "Numerical algorithm for a linear time-optimal control problem," Comput. Math. Math. Phys., **42**, No. 8, 1123–1134 (2002).
6. *Aleksandrov V. M.*, "Sequential synthesis of time-optimal control," Comput. Math. Math. Phys., **39**, No. 9, 1402–1415 (1999).
7. *Aleksandrov V. M.*, "Convergence of the method of sequential synthesis of time-optimal control," Comput. Math. Math. Phys., **39**, No. 10, 1582–1593 (1999).
8. *Aleksandrov V. M.*, "A numerical method for solving a linear time-optimal control problem," Comput. Math. Math. Phys., **38**, No. 6, 881–893 (1998).
9. *Aleksandrov V. M.*, "Solution of optimal control problems on the basis of the quasi-optimal control method [in Russian]," Tr. Inst. Mat., **10**, pp. 18–54, Nauka, Novosibirsk (1988).
10. *Zholudev A. I., Tyatyushkin A. I., and Ehrinchek N. M.*, "Numerical optimization methods for controlled processes [in Russian]," Izv. Akad. Nauk, Tekh. Kibern., No. 4, 14–31 (1989).

11. *Pshenichnyj B. N. and Sobolenko L. A.*, “Accelerated method for solving time-optimal linear control problem [in Russian],” Zh. Vychisl. Mat. Mat. Fiz., **8**, No. 6, 1345–1352 (1968).
12. *Dubovitskii A. Ya. and Rubtsov V. A.*, ”Linear time-minimal problems,” U. S. S. R. Comput. Math. Math. Phys., **8**, No. 5, 1–18 (1971).
13. *Kiselev Yu. N.*, “Fast-convergence algorithms for linear time-optimal control,” Cybernetics, **26**, No. 6, 848–869 (1990).
14. *Srochko V. A.*, Iterative Methods for Solving Optimal Control Problems [in Russian], FIZMATLIT, Moscow (2000).
15. *Dyurkovich E.*, “A numerical method for solving time-optimal problems, with an estimate of accuracy,” Sov. Math., Dokl., **26**, 132–136 (1982).
16. *Starov V. G.*, “Neustadt–Eaton’s method modification [in Russian],” Mat. Zametki SVFU, **21**, No. 3, 107–112 (2014).

Submitted September 19, 2018
Revised October 20, 2018
Accepted November 13, 2018

Valentin G. Starov
Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, Novosibirsk 630090, Russia