МЕТОД ПОСЛЕДОВАТЕЛЬНОГО КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ ДЛЯ РЕШЕНИЯ ДИСКРЕТНЫХ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

А. О. Гацко

Белорусский государственный университет, г. Минск;
fpm.gacko@bsu.by;
науч. рук. – Н. М. Дмитрук, канд. физ.-мат. наук, доц.

В работе рассматривается задача оптимального управления (ОУ) нелинейной дискретной системой. Поскольку задача может решаться как задача нелинейного программирования, к ней применимы методы последовательного квадратичного программирования (Sequential Quadratic Programming – SQP). В работе исследуются особенности применения SQP и структура квадратичной подзадачи, возникающей в связи с рассматриваемой задачей ОУ. Актуальность работы обусловлена широким спектром приложений задач ОУ, в частности, часто возникающей необходимостью решать такие задачи в режиме реального времени.

Ключевые слова: Оптимальное управление; дискретные системы; последовательное квадратичное программирование; линейно-квадратичная подзадача.

В работе рассматривается задача ОУ дискретной стационарной нелинейной системой вида

$$x_{k+1} = f(x_k, u_k),\ k = \overline{0, N-1}, \tag{1}$$

где $x_k \in \mathbb{R}^n$ – состояние, $u_k \in \mathbb{R}^r$ – управление в дискретный момент k. На состояния и управления накладываются ограничения:

1) смешанные: $h(x_k, u_k) \le 0,\ k = \overline{0, N-1}$;

2) граничные: $r_0(x_0) + r_N(x_N) = 0$;

3) терминальные $h_N(x_N) \le 0$.

Задача ОУ [1, гл.8] состоит в нахождении такого (оптимального) управления, которое порождает траекторию системы (1), вместе с ней удовлетворяет всем ограничениям и минимизирует заданный критерий качества, например, типа Больца

$$\sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N).$$

Относительно функций $f: \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$, $h: \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^m$, $r: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^p$, $L: \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}, E: \mathbb{R}^n \to \mathbb{R}$ предполагается, что они дважды непрерывно дифференцируемы.

Задачи ОУ, даже для дискретных систем (1), как правило, достаточно сложны и требуют численного решения. При этом их удобно представлять как задачи нелинейного программирования [1, гл.6]:

$$\min_{x_0,u_0,x_1,u_1,\ldots,u_{N-1},x_N} \sum_{k=0}^{N-1} L(x_k,u_k)+E(x_N),$$
$$x_{k+1}-f(x_k,u_k)=0, k=\overline{0,N-1}, \tag{2}$$
$$h(x_k,u_k)\le 0, k=\overline{0,N-1},$$
$$r_0(x_0)+r_N(x_N)=0, h_N(x_N)\le 0.$$

Для решения задачи (2) различают два подхода:

1. *Параллельный*: переменными оптимизации выступают как управления $u_k, k=\overline{0,N-1}$, так и состояния $x_k, k=\overline{0,N}$;

2. *Последовательный* подход, при котором переменными оптимизации являются только управления $u_k, k=\overline{0,N-1}$.

Параллельный подход, фактически решает задачу (2), с указанными в ней переменными оптимизации. Эта задача имеет большую размерность и ярко выраженную структуру, которая задана ограничениями-равенствами, порождаемыми динамикой системы (1). Для решения (2) целесообразно применение методов, которые учитывают эту структуру.

Последовательный подход основан на том, что состояния $x_k, k=\overline{1,N}$, можно исключить из задачи (2), поскольку они выражаются через начальное состояние x_0 и управления $\mathbf{u}=\{u_0,...,u_{N-1}\}$:

$$x_0(x_0,\mathbf{u})=x_0,\; x_{k+1}(x_0,\mathbf{u})=f(x_k(x_0,\mathbf{u}),u_k), k=\overline{0,N-1},$$

Тогда задачу (2) можно переписать в виде так называемой функциональной формы задачи ОУ

$$\min_{x_0,u_0,u_1,\ldots,u_{N-1}} \sum_{k=0}^{N-1} L(x_k(x_0,\mathbf{u}),u_k)+E(x_N(x_0,\mathbf{u})),$$
$$h(x_k(x_0,\mathbf{u}),u_k)\le 0, k=\overline{0,N-1}, \tag{3}$$
$$r_0(x_0)+r_N(x_N(x_0,\mathbf{u}))=0, h_N(x_N(x_0,\mathbf{u}))\le 0.$$

Задача (3) снова может быть решена методами нелинейного программирования, но она не имеет столь выраженной структуры, как в (2). Заметим, что в последовательном подходе уравнение системы (1) выполняется точно, в то время как в параллельном подходе даже по завершении итераций равенство (1) выполняется приближенно.

Далее рассмотрим один из популярных методов нелинейного программирования – SQP [2], и как этот метод учитывает структуру задачи (2) в параллельном подходе.

Запишем (2) в виде общей задачи нелинейного программирования. Для этого соберем все переменные в вектор $\omega=(x_0,u_0,x_1,u_1,\ldots,u_{N-1},x_N)$, целевую функцию в $F(\omega)$, все равенства в функцию $G(\omega)$ и все неравенства в функцию $H(\omega)$:

$$\min_{\omega} F(\omega), G(\omega)=0, H(\omega)\le 0.$$

Функция Лагранжа для этой задачи нелинейного программирования

$$\mathcal{L}(\omega,\lambda,\mu)=F(\omega)+\lambda^T G(\omega)+\mu^T H(\omega),$$

для задачи (2) может быть представлена в виде суммы

$$\mathcal{L}(\omega,\lambda,\mu)=\sum_{k=0}^{N}\mathcal{L}_k(\omega_k,\lambda,\mu),$$

где локальные функции $\mathcal{L}_k$ для (2) определяются следующим образом:

$$\mathcal{L}_0(\omega_0,\lambda,\mu)=L_0(x_0,u_0)+\lambda_1^T f_0(x_0,u_0)+\mu_0^T h_0(x_0,u_0)+\lambda_r^T r_0(x_0),$$

$$\mathcal{L}_k(\omega_k,\lambda,\mu)=L_k(x_k,u_k)+\lambda_{k+1}^T f_k(x_k,u_k)-\lambda_k^T x_k+\mu_k^T h_k(x_k,u_k),\ k=\overline{1,N-1},$$

$$\mathcal{L}_N(\omega_N,\lambda,\mu)=E_N(x_N)-\lambda_N^T x_N+\mu_N^T h_N(x_N)+\lambda_r^T r_N(x_N).$$

Заметим, что множители Лагранжа (λ,μ) появляются во всех $\mathcal{L}_k$, а прямые переменные $\omega_k=(x_k,u_k)$, $k=\overline{1,N-1}$, $\omega_N=x_N$ в этих функциях разделены. Поэтому гессиан $\nabla^2_{\omega}\mathcal{L}$ является блочно-диагональной матрицей – все «смешанные» производные второго порядка по ω равны нулю:

$$\frac{\partial^2\mathcal{L}}{\partial\omega_i\partial\omega_j}(\omega,\lambda,\mu)=0,\ \text{ при } i\ne j.$$

Пусть $\omega=(x,\lambda,\mu)$ – текущая итерация метода SQP (индекс итерации опустим для упрощения выкладок). На итерации SQP решает квадратичную подзадачу [2] относительно переменных $\Delta\omega=(\Delta x_0,\Delta u_0,\dots,\Delta x_N)$:

$$\begin{aligned}&\min_{\Delta\omega}\nabla F(\omega)^T\Delta\omega+\frac{1}{2}\Delta\omega^T\nabla^2_{\omega}\mathcal{L}(\omega,\lambda,\mu)\Delta\omega,\\&G(\omega)+\nabla G(\omega)^T\Delta\omega=0,\\&H(\omega)+\nabla H(\omega)^T\Delta\omega\le 0.\end{aligned}\tag{4}$$

Новая итерация SQP строится по правилу $\bar{\omega}=\omega+\Delta\omega$. Алгоритм завершает работу при $\|\Delta\omega\|\le\varepsilon$, где $\varepsilon>0$ – выбранная точность.

Задачу (4) запишем в терминах исходной задачи ОУ (2). Для этого введем некоторые обозначения. Диагональные блоки гессиана функции Лагранжа на текущей итерации обозначим

$$Q_k=\nabla^2_{\omega_k}\mathcal{L}(\omega,\lambda,\mu), k=\overline{1,N}.$$

Введем обозначения для градиентов функций, входящих в критерий качества:

$$g_k = \nabla_{(x,u)} L(x,u), k=\overline{1,N},\ g_N = \nabla_x E(x_N).$$

В линеаризации динамической системы (1) обозначим матрицы

$$A_k = \frac{\partial f(x_k,u_k)}{\partial x}, B_k = \frac{\partial f(x_k,u_k)}{\partial u}, k=\overline{1,N-1},$$

и невязки уравнений динамической системы (1)

$$a_k = f(x_k,u_k) - x_{k+1},\ \ k=\overline{1,N-1}.$$

Наконец, для ограничений введем обозначения:

$$R_0 = \frac{\partial x_0(x_0)}{\partial x},\ \ R_N = \frac{\partial r_N(x_N)}{\partial x},\ H_k = \frac{\partial h_k(x_k,u_k)}{\partial (x_k,u_k)}, H_N = \frac{\partial h_N(x_N)}{\partial x},$$

$$r = r_0(x_0) + r_N(x_N),\ \ h_k = h_k(x_k,u_k),\ \ h_N = h_N(x_N).$$

С учетом введенных обозначений, задача квадратичного программирования (4) в терминах задачи ОУ (2) принимает вид:

$$\begin{aligned}
&\min_{\Delta x_0, \Delta u_0, \dots, \Delta x_N} \sum_{k=0}^{N-1} \left\{ \frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}^T Q_k \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + g_k^T \begin{bmatrix} \Delta x_N \\ \Delta u_N \end{bmatrix} \right\} + \frac{1}{2} \Delta x_N^T Q_N \Delta x_N + \Delta g_N^T \Delta x_N, \\
&A_k \Delta x_k + B_k \Delta u_k - \Delta x_{k+1} + a_k = 0,\ \ k = 0, \dots, N-1, \\
&h_k + H_k \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} \le 0, k = \overline{0, N-1}, \\
&r + R_0 \Delta x_0 + R_N \Delta x_N = 0, h_k + H_k \Delta x_N \le 0.
\end{aligned} \tag{5}$$

Полученная задача (5) снова является задачей ОУ. Первая группа ограничений-равенств – линейная нестационарная динамическая система:

$$\Delta x_{k+1} = A_k \Delta x_k + B_k \Delta u_k + a_k = 0,\ \ k = 0, \dots, N-1,$$

также линейны все ограничения. Если все блоки гессиана Q_k положительно определены, то (5) – выпуклая линейно-квадратичная задача и может быть решена методами квадратичного программирования [2]. Например, эффективным численным методом решения задачи квадратичного программирования (5), учитывающим разреженность ограничений-равенств в силу динамики, разреженный прямо-двойственный метод внутренней точки, реализованный в пакете OOQP.

Библиографические ссылки

1. Методы оптимизации : учеб. пособие / В. В. Альсевич [и др.]. Мн., 2011.
2. *Nocedal J., Wright S. J.* Numerical Optimization. New York, 2006.
3. *Boyd S., Vandenberghe L.* Convex optimization. Cambridge, 2004.