УДК 519.853, 004.85, 517.977.5

СОВМЕСТНОЕ ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ И МЕТАЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ОПТИМИЗАЦИИ ДЛЯ СИНТЕЗА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ СПУТНИКОМ

А.В. Пантелеев

Московский авиационный институт (национальный исследовательский университет)

Россия, 125993, Москва, Волоколамское шоссе, д. 4, А-80, ГСП-3

E-mail: avpanteleev@inbox.ru

В.Н. Пановский

Московский авиационный институт (национальный исследовательский университет)

Россия, 125993, Москва, Волоколамское шоссе, д. 4, А-80, ГСП-3

E-mail: panovskiy.v@yandex.ru

Ключевые слова: оптимизация, метаэвристические алгоритмы, нелинейное программирование, оптимальное управление, машинное обучение, open-source, OSOL Extremum.

Аннотация: В данной работе представлен двухэтапный подход для решения задачи синтеза управления спутником с обратной связью. На первом этапе формулируется задача поиска оптимального программного управления, которая решается для различных начальных условий путем преобразования к задаче нелинейного программирования и последующего применения метаэвристических алгоритмов оптимизации. Таким образом, в результате первого этапа находятся управления и соответствующие им траектории для различных начальных условий из заданного множества. На втором этапе полученная информация агрегируется и используется для построения регрессии, которая ищется как решение задачи обучения с учителем. Полученная функция используется в качестве аппроксимации данных, полученных на первом этапе, для синтеза требуемого управления с обратной связью.

1. Введение

В теории управления большая часть исследований проводится в области применения нейронных сетей в качестве аппроксимации управления [1, 2]. В большинстве случаев в данных работах предпринимаются попытки решить вычислительно сложную задачу напрямую без приобретения априорной информации об оптимальных или хотя бы субоптимальных управлениях.

Предлагается комбинированный двухэтапный подход, который использует идеи и алгоритмы теории оптимального управления и базовые техники машинного обучения. Основной идеей первого этапа является формулирование задачи поиска оптимального программного управления, которая впоследствии преобразуется к задаче нелинейного программирования, решение которой можно найти с помощью метаэвристических алгоритмов оптимизации. Данная процедура повторяется для разных начальных условий. Результатом данного этапа является набор оптимальных программных управлений и соответствующих им траекторий. Полученный набор данных используется на втором этапе для формулировки задачи обучения с учителем.

Выбор именно такого подхода обоснован наличием большого числа эффективных методов оптимального программного управления [3–5]и большого числа алгоритмов оптимизации и построения регрессии [6, 7].

Данная работа может быть рассмотрена как апробация open-source программного комплекса [10], разрабатываемого авторами, для решения задачи синтеза оптимального управления с полной обратной связью для некоторого заданного множества начальных состояний.

2. Решение задачи поиска субоптимального управления с обратной связью

2.1. Этап 1: Поиск оптимальных программных управлений

Основной целью первого этапа является сбор априорной информации о поведении системы в совокупности с информацией о субоптимальных управлениях.

Рассмотрим решение следующей задачи оптимального управления [4, 5, 8]:

$$
\begin{aligned}
&\dot{x}(t) = f(t, x(t), u(t)),\\
&x(t_0) = x_0 \in \Omega,\\
&\Gamma_i(x(t_1)) = 0, i = 1, ..., l,\\
&I = \int_{t_0}^{t_1} f^0(t, x(t), u(t))\,dt + F(x(t_1)) \to \min,
\end{aligned}
\tag{1}
$$

где $t \in T = [t_0; t_1]$ – непрерывное время функционирования системы, $x \in \mathbb{R}^n$ – вектор состояния, $u \in U \subset \mathbb{R}^q$ – вектор управления, U – множество допустимых значений управления, $f(t, x, u) = (f_1(t, x, u), ..., f_n(t, x, u))^{\mathrm{T}}$ – непрерывная вектор-функция, $\Omega \in \mathbb{R}^n$ – множество возможных начальных состояний, $f(t, x, u)$ и $F(x)$ - заданные непрерывные функции.

Для решения задачи поиска оптимального программного управления предлагается использовать параметрическую форму задания управления, с помощью которой задачу можно свести к задаче параметрической оптимизации. Таким образом, получаемое управление будет субоптимальным.

Для применения метаэвристических алгоритмов оптимизации (они могут быть любого типа: эволюционные, биоинспирированные, методы роевого интеллекта, мультиагентные, меметические и др. [6]) необходимо параметризовать управление, например, в виде кусочно-постоянных, кусочно-линейных функций, разложения по ортогональной системе базисных функций и т.д. Здесь искомое программное управление ищется в виде кусочно-линейных функций: $u_j(t) = \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i} \cdot a_{j,i} + \frac{t-\tau_i}{\tau_{i+1}-\tau_i} \cdot a_{j,i+1}$, $i = 1, ..., n$, где $\tau_i = t_0 + i \cdot \frac{t_1-t_0}{n}$ и $\tau_i \leqslant t < \tau_{i+1}$, $j = 1, ..., q$, которые задаются вектором коэффициентов $a_j = (a_{j,0}, a_{j,1}, ..., a_{j,n})^{\mathrm{T}}$, $j = 1, ..., q$.

Таким образом, решаемая задача (1) сводится к проблеме поиска вектора $a^* = \left(a_1^{\mathrm{T}}, ..., a_q^{\mathrm{T}}\right)^{\mathrm{T}}$, минимизирующего критерий качества управления для заданного начального состояния из множества Ω. На первом этапе для всех начальных условий $x^k(t_0) = x_0^k \in \Omega, k = 1, ..., K$, где K – количество различных начальных состояний, находятся векторы $a^{*,k}$ и, как следствие, субоптимальные управления $\tilde{u}^{*,k}(t), k = 1, ..., K$. С этой целью предлагается использовать свободно распространяемую библиотеку алгоритмов оптимизации OSOL Extremum [10].

2.2. Этап 2: Формирование регрессии

2.2.1. Подготовка данных. Имея набор субоптимальных управлений $\left\{\tilde{u}^{*,k}(t)\right\}_{k=1}^K$, можно получить набор соответствующих им траекторий $\left\{\tilde{x}^{*,k}(t)\right\}_{k=1}^K$. Этой информации достаточно для построения набора данных, который предлагается формировать в виде таблицы 1.

Таблица 1. Шаблон набора данных

k	t	$\tilde{x}_1^{*,k}(t)$	$\cdots$	$\tilde{x}_n^{*,k}(t)$	$\tilde{u}_1^{*,k}(t)$	$\cdots$	$\tilde{u}_q^{*,k}(t)$
1	t_1^k	$\tilde{x}_1^{*,1}(t_1^k)$	$\cdots$	$\tilde{x}_n^{*,1}(t_1^k)$	$\tilde{u}_1^{*,1}(t_1^k)$	$\cdots$	$\tilde{u}_q^{*,1}(t_1^k)$
1	t_2^k	$\tilde{x}_1^{*,1}(t_2^k)$	$\cdots$	$\tilde{x}_n^{*,1}(t_2^k)$	$\tilde{u}_1^{*,1}(t_2^k)$	$\cdots$	$\tilde{u}_q^{*,1}(t_2^k)$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
K	t_1^k	$\tilde{x}_1^{*,K}(t_1^k)$	$\cdots$	$\tilde{x}_n^{*,K}(t_1^k)$	$\tilde{u}_1^{*,K}(t_1^k)$	$\cdots$	$\tilde{u}_q^{*,K}(t_1^k)$
K	t_2^k	$\tilde{x}_1^{*,K}(t_2^k)$	$\cdots$	$\tilde{x}_n^{*,K}(t_2^k)$	$\tilde{u}_1^{*,K}(t_2^k)$	$\cdots$	$\tilde{u}_q^{*,K}(t_2^k)$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$

2.2.2. Формирование регрессии. Регрессия является классическим примером задачи обучения с учителем [7]. В рамках данного исследования найденные на предыдущем этапе управления могут быть рассмотрены как желаемое значение регрессора для вектора входных параметров – времени и вектора состояния. Другими словами, требуемое управление с обратной связью может быть представлено следующим образом:

$$\tilde{u}_j^{*,k}\left(t_i^k, \tilde{x}^{*,k}\left(t_i^k\right)\right) = \tilde{u}_j^{*,k}\left(t_i^k\right), j = 1, ..., q, k = 1, ..., K, \tag{2}$$

где $t_i^k, i = 1, 2, 3, ...$ – моменты времени из отрезка $[t_0; t_1]$.

Приведенное соотношение показывает, как генерировать управляющие воздействия для ранее использованных входных параметров, но неясно, что делать для остальных значений, который могут возникать в будущем. Алгоритмы машинного обучения имеют свойство обобщать данные, используя конечный набор наблюдений [7], поэтому их применение является естественным.

Для построения регрессии предлагается использовать хорошо изученный и часто применяемый алгоритм случайного леса [7], который реализован в библиотеке Scikit-Learn [11].

В результате обучения вырабатывается правило, согласно которому по данным о текущем времени и векторе состояния генерируется управление, т.е. $u(t) = u(t, x(t))$, что означает реализацию принципа обратной связи.

3. Задача оптимального управления спутником

3.1. Постановка задачи

Рассмотрим задачу гашения вращательного движения спутника, описанную в [12]. После преобразования к безразмерным величинам система дифференциальных уравнений, описывающая движение твердого тела относительно центра масс, выглядит следующим образом:

$$\dot{p} = u_1/6, \dot{q} = u_2 - 0.2 \cdot p \cdot r, \dot{r} = 0.2 \cdot (u_3 + p \cdot q), p(0) = p_0, q(0) = q_0, r(0) = r_0. \tag{3}$$

Ограничения на управление представлены в виде интервалов: $U = [-500; 500] \times [-500; 500] \times [-500; 500]$. Начальные значения p_0, q_0, r_0 берутся из множества $\{-25, -20, ..., -5, 0, 5, ..., 20, 25\}$, что приводит к $11^3 = 1331$ возможным комбинациям. Время функционирования системы: $t_0 = 0, t_1 = 1$. Требуемое конечное состояние системы: $p(1) = q(1) = r(1) = 0$, функционал качества управления: $I = \int_0^1 \sum_{i=1}^3 |u_i(t)|\, dt + R \cdot [p^2(1) + q^2(1) + r^2(1)]$, где $R = 10$ – параметр штрафа.

В качестве основного инструмента для решения задачи параметрической оптимизации был использован интервальный метод взрывов [5].

3.2. Итоговый результат

Рассмотрим несколько конфигураций начальных состояний $x_0 \in \Omega = [-25; 25] \times [-25; 25] \times [-25; 25]$, которые не были использованы на первом этапе.

Случай 1. Начальное состояние: $p_0 = 24, q_0 = 16, r_0 = 16$. Достигнутое конечное состояние: $p(t_1) = -0.067, q(t_1) = -0.065, r(t_1) = -0.094$.

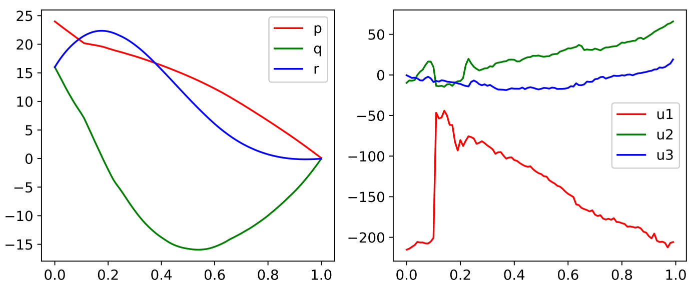

Рис. 1. Случай 1: траектории и управления

Случай 2. Начальное состояние: $p_0 = -9, q_0 = 13, r_0 = -24$. Достигнутое конечное состояние: $p(t_1) = 0.096, q(t_1) = -0.008, r(t_1) = -0.019$.

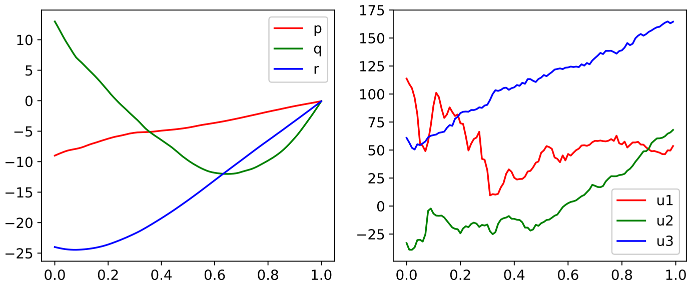

Рис. 2. Случай 2: траектории и управления

Аналогичные результаты получаются при других начальных состояниях из Ω.

По приведенным на рис. 1 и рис. 2 графикам видно, что предлагаемый подход имеет потенциал решения проблемы синтеза оптимального управления с обратной связью. Очевидным недостатком является негладкая форма получаемого управления.

4. Заключение

В ходе данной работы эффективность разработанного программного комплекса OSOL Extremum была апробирована путем решения около 1300 задач поиска оптимального программного управления, а предложенный подход был успешно применен для решения прикладной задачи гашения вращательного движения спутника.

Дальнейшие исследования планируется проводить по нескольким направлениям: определение потери относительно критерия эффективности и обеспечение более гладкой формы управления, генерируемого регрессией.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (16-07-00419 А).

Список литературы

1. Shin J.-H., Jun H.-B., Kim J.-G. Dynamic Control of Intelligent Parking Guidance Using Neural Network Predictive Control // Computers & Industrial Engineering. 2018. No. 120. P. 15-30.
2. Zhang G., Deng Y. and Zhang W. Robust Neural Path-Following Control for Underactuated Ships with the DVS Obstacles Avoidance Guidance // Ocean Engineering. 2017. No. 143. P. 198-208.
3. Kuo Y.-L., Wu T.-L. Open-Loop and Closed-Loop Attitude Dynamics and Controls of Miniature Spacecraft Using Pseudowheels // Computers and Mathematics with Applications. 2012. No. 64. P. 1282-1290.
4. Panteleev A.V., Metlitskaya D.V. Using the Method of Artificial Immune Systems to Seek the Suboptimal Program Control of Deterministic Systems // Automation and Remote Control. 2014. Vol. 75, No. 11. P. 1922-1935.
5. Panteleev A.V., Panovskiy V.N., Korotkova T.I. Interval Explosion Search Algorithm and Its Application to Hypersonic Aircraft Modelling and Motion Optimization Problems // Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software. 2016. No. 9. P. 55-67.
6. Brownlee J. Clever Algorithms: Nature-Inspired Programming Recipes. North Carolina, Morrisville: LuLu, 423 p.
7. Herbrich R., Graepel T. Machine Learning: An Algorithmic Perspective. FL: CRC Press, 2015. 430 p.
8. Пантелеев А.В., Бортаковский А.С. Теория управления в примерах и задачах. М.: ИНФРА-М, 2016. 583 с.
9. Пантелеев А.В., Скавинская Д.В., Алешина Е.А. Метаэвристические алгоритмы поиска оптимального программного управления. М.: ИНФРА-М, 2016. 396 с.
10. Panovskiy V.N. Open-Source Optimization Library - Extremum. https://github.com/wol4aravio/OSOL.Extremum
11. Hackeling G. Mastering Machine Learning with scikit-learn. Birmingham: Packt Publishing, 2014. 221 p.
12. Крылов И.А. Численное решение задачи об оптимальной стабилизации спутника // Вычислительная математика и математическая физика. 1968. Т. 8, № 1. С. 284-291.