Постановка задачи поиска оптимального программного управления

Пусть поведение модели объекта управления описывается обыкновенным дифференциальным уравнением

$$\dot{x}(t) = f(x(t), u(t)) \, , \qquad (1)$$

где x – вектор состояния системы, $x = (x_1, \ldots, x_n)^T \in R^n$; u – вектор управления, $u = (u_1, ..., u_q)^T \in U \subseteq R^q$, U – некоторое заданное множество допустимых значений управления, определяемое прямым произведением отрезков $[a_1, b_1] \times ... \times [a_q, b_q]$; $t \in T = [0, t_f]$ – промежуток времени функционирования системы; момент окончания t_f заранее не задан; $f(x,u)$ – непрерывная вектор-функция; R^n – n-мерное евклидово пространство.

Начальное условие $x(0) = x_0$ задает начальное состояние системы.

Определим множество допустимых процессов $\boldsymbol{D}(x_0)$ как множество троек $d = (t_f, x(\cdot), u(\cdot))$, которые включают момент окончания процесса управления, траекторию $x(\cdot)$ и управление $u(\cdot)$ (где $\forall t \in T$: $x(t) \in R^n$, $u(t) \in U$, функции $x(\cdot)$ непрерывны и кусочно-дифференцируемы, а $u(\cdot)$ кусочно-непрерывны), удовлетворяющие уравнению (1) с заданным начальным условием.

На множестве $\boldsymbol{D}(x_0)$ определим функционал качества управления

$$I(d) = F(t_f, x(t_f)) \, , \qquad (2)$$

где $F(t_f, x)$ – заданная непрерывная функция.

Требуется найти такую тройку $d^* = (t_f*, x^*(\cdot), u^*(\cdot)) \in \boldsymbol{D}(x_0)$, что $I(d^*) = \min\limits_{d \in \boldsymbol{D}(x_0)} I(d)$.

Постановка задачи об управлении солнечным парусом

В задаче рассматривается модель плоского паруса, имеющего идеальную отражающую поверхность. Движение происходит благодаря зеркальной поверхности, создающей тягу при отражении падающего на поверхность солнечного светового излучения. Первые исследования в области применения солнечного излучения для межпланетных полетов связаны с опытами П.Н. Лебедева по обнаружению светового давления в начале XX века. Первое серьезное исследование данной проблемы принадлежит Ф.А. Цандеру. В своих работах он предлагал для создания тяги использовать тонкие зеркала, выполненные из алюминиевых листов.

Солнечный парус как двигатель создает тягу, используя поток энергии и электромагнитное излучения Солнца. При взаимодействии потока солнечного светового излучения с поверхностью паруса происходит изменение вектора количества движения потока фотонов.

Рассмотрим математическую модель движения центра масс космического аппарата под действием тяги в гравитационном поле:

$$M(t) \cdot \ddot{\vec{r}} = \vec{P} + M(t) \cdot \vec{g} + \vec{F}, \tag{3}$$

где $M(t)$ – масса аппарата, $\vec{r}$ –радиус-вектор в инерциальной системе координат, $\vec{P} = P\vec{e}$ – вектор тяги, $\vec{e}$ – единичный вектор направления тяги, $g(t, \vec{r})$ – вектор ускорения от гравитационных сил, t – время, $\vec{F}$ – вектор других внешних сил, действующих на аппарат, определяемый внешними условиями полета.

Модель движения солнечного паруса, описанная уравнением (1), содержит в себе учет гравитационных сил и сил давления солнечного излучения.

Масса космического аппарата является величиной постоянной, поскольку данная модель не учитывает износ солнечного паруса и изменение его площади, и складывается из двух составляющих: массы полезной нагрузки M_π и массы непосредственно паруса, задаваемой соотношением $M_S = S_0\rho\delta$. Подставляя полученные соотношения в уравнение движения, получаем:

$$\ddot{\vec{r}} = \frac{\wp_{\text{Å}} \cdot S_0}{S_0 \cdot \rho \cdot \delta + M_\pi} \cdot \left(\frac{R_\oplus}{R}\right)^2 \cdot (\vec{n} \cdot \vec{i})^2 \vec{n} + \vec{g},$$

где $\vec{n}$ – единичный вектор нормали к теневой стороне паруса, $\vec{i}$ –единичный вектор направления солнечных лучей, $(\vec{n} \cdot \vec{i})$ – косинус угла $\widehat{\vec{n}\,\vec{i}}$ установки паруса по отношению к солнечным лучам, $\wp_{\text{Å}}$ – солнечное давление на единичную площадку, установленную перпендикулярно к солнечным лучам, на орбите Земли $R_\oplus$, $R = R(r)$ – расстояние аппарата до Солнца (в гелиоцентрической системе координат $R = r$, ρ и δ - плотность и толщина материала паруса.

В уравнения движения вошли параметры паруса S_0, ρ, δ и M_π. Комбинацию этих параметров, содержащуюся в уравнении массы, можно заменить ускорением, которое сообщает парус космическому аппарату на расстоянии $R_\oplus$ от Солнца, будучи установленным перпендикулярно к солнечным лучам ($\vec{n} = \vec{i}$):

$$a_0 = \frac{\wp_{\text{Å}} \cdot S_0}{S_0 \cdot \rho \cdot \delta + M_\pi} \leq \frac{\wp_{\text{Å}}}{\rho \cdot \delta}$$

В конечном результате уравнение движения записывается следующим образом:

$$\ddot{\vec{r}} = a_0 \cdot \left(\frac{R_\oplus}{R}\right)^2 \cdot (\vec{n} \cdot \vec{i})^2 \vec{n} + \vec{g}$$

В данной модели ускорение a_0 для уравнения движения играет роль постоянного управляющего параметра. Единственная управляющая функция в этом уравнении – угол установки паруса (направление нормали к теневой стороне паруса), который может меняться в пределах:

$$-\frac{\pi}{2} \le (\overrightarrow{n(t)} \cdot \vec{i}) \le \frac{\pi}{2}$$
$$\left|\vec{n}\right| \equiv 1$$

При рассмотрении межорбитального участка перелета солнечного паруса, летящего вне сферы гравитационного воздействия планет, орбиты которых считаются компланарными с нулевым эксцентриситетом, движение космического аппарата происходит в центральном гравитационном поле Солнца под действием давления солнечного излучения. Полагая в уравнении движения $R = r$, $g = -g_0 r_0^2 r^{-3} r$ и обозначая через α угол установки паруса, составляемый нормалью к теневой стороне паруса и радиусом-вектором $\vec{r}$, совпадающим с направлением солнечных лучей $\vec{i} = \frac{\vec{r}}{r}$, получаем следующую систему уравнений плоского движения:

$$\begin{aligned}
&\dot{r}(t) = u(t),\\
&\dot{\theta}(t) = \frac{v(t)}{r(t)},\\
&\dot{u}(t) = \frac{v^2(t)}{r(t)} + \frac{r_0^2}{r^2(t)}(a_0 \cdot \cos^3\alpha - g_0),\\
&\dot{v}(t) = -\frac{u(t)v(t)}{r(t)} + a_0 \cdot \frac{r_0^2 \cdot \sin\alpha \cos^2\alpha}{r^2(t)},
\end{aligned}$$

где r, θ – радиальная и угловая позиции соответственно; u, v – радиальная и тангенциальная скорости, α – угол тангажа (переменная управления), a_0 – параметр яркости солнечного паруса, g_0 – гравитационное ускорение от Солнца на радиусе r_0. Схема модели космического аппарата с солнечным парусом изображена на рисунке 1.

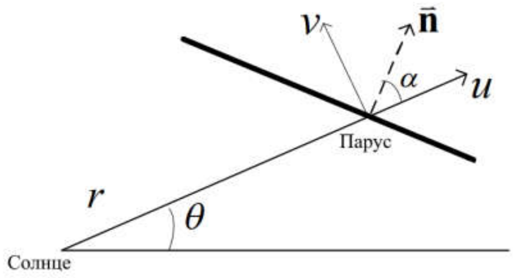

Рисунок 1 Схема объекта управления

Начальные условия определяются из положения солнечного паруса на гелиоцентрической орбите Земли.

$$\begin{aligned} r(0) &= 1, \\ \theta(0) &= 0, \\ u(0) &= 0, \\ v(0) &= 1. \end{aligned} \tag{4}$$

Конечные условия определяются в зависимости от гелиоцентрической орбиты выбранной целевой планеты и приведены в таблице 1.

Таблица 1. Конечные условия задачи перелета солнечного паруса

Целевая орбита планеты	r_f	u_f	$v_f = \frac{1}{\sqrt{r_f}}$
Меркурий	0,39	0	1,601281
Венера	0,72	0	1,178511
Марс	1,52	0	0,811107

Поскольку целью миссии является перелет между орбитами планет солнечной системы, на угловую позицию солнечного паруса в гелиоцентрической системе координат не накладываются ограничения.

Функционал качества управления:

$$I = \lambda_1 \frac{t_f}{86400} + \lambda_2 [r(t_f) - r_f]^2 + \lambda_3 [u(t_f) - u_f]^2 + \lambda_4 [v(t_f) - v_f]^2, \qquad (5)$$

где $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ весовые коэффициенты. Структура функционала соответствует реализации межпланетной миссии, заключающейся в переходе с орбиты Земли (ее задают начальные условия (4)) на орбиту Меркурия, Венеры или Марса (их определяют конечные условия, приведенные в таблице 1) за минимальное время. Первое слагаемое описывает число дней, затрачиваемое на перелет, а второе, третье и четвертое – точность выхода на заданную орбиту. Значения весовых коэффициентов характеризуют степень важности обеспечения требований к желаемой траектории. Для контроля степени выполнения конечных условий на каждой итерации контролируется величина невязки в виде модуля разности по всем координатам космического аппарата в конечный момент времени:

$$(|\, r(t_f) - r_f \,|, |\, u(t_f) - u_f \,|, |\, v(t_f) - v_f \,|).$$

Предлагается искать приближенное решение в виде функции насыщения, которая должна гарантировать выполнение заданных ограничений на управление параллелепипедного вида. Аргументы функции насыщения искать в виде линейной комбинации заданных базисных функций, которые используются в спектральном методе анализа и синтеза нелинейных систем управления. Поскольку момент окончания явно не задан, то возникает проблема определения семейства базисных функций. Поэтому применяется подход, связанный с введением новой независимой переменной $\tau \in [0,1]$, отрезок ее изменения делится на P подынтервалов одинаковой длины, равной $\frac{1}{P}$. Промежуток времени $[0, t_f]$ функционирования системы делится тоже на P подынтервалов переменной длины $h_{k+1} = t_{k+1} - t_k \geq 0$, $k = 0,1,\ldots,P-1$; $t_0 = 0$. При этом момент окончания процесса находится по формуле

$$t_f = \sum_{k=0}^{P-1} h_{k+1},$$

Моменту $\tau = 0$ соответствует $t = 0$, а моменту $\tau = 1$ момент окончания t_f. Используем соотношение, связывающее дифференциалы переменных t и τ.

$$dt = h_{k+1} P d\tau, \; t_k \le t < t_{k+1}, k = 0,1,...,P-1. \qquad (6)$$

Интегрируя по отдельному промежутку времени t, получаем $\int\limits_{t_k}^{t_{k+1}} dt = h_{k+1} P \int\limits_{\tau_k}^{\tau_{k+1}} d\tau$, $\underbrace{t_{k+1} - t_k}_{h_{k+1}} = h_{k+1} P(\tau_{k+1} - \tau_k)$. Отсюда следует, что шаг по переменной τ постоянный и равен $\frac{1}{P}$, так как $\tau_{k+1} - \tau_k = \frac{1}{P}$, $k = 0,1,...,P-1$; $\tau_0 = 0, \tau_P = 1$.

Найдем связь между переменными t и τ: $\int\limits_{t_k}^{t} dt = h_{k+1} P \int\limits_{\tau_k}^{\tau} d\tau$, $t - t_k = h_{k+1} P(\tau - \tau_k) = h_{k+1} P(\tau - \frac{k}{P}), \;\; k = 0,1,...,P-1$. Поскольку $t_k = \sum\limits_{j=0}^{k-1} h_{j+1}$, имеем

$$t = \sum_{j=0}^{k-1} h_{j+1} + h_{k+1} P(\tau - \frac{k}{P}), \;\; k = 0,1,...,P-1.$$

$$\frac{dx(\tau)}{d\tau} = h_{k+1} P f(x(\tau), u(\tau)), \quad \tau_k \le \tau < \tau_{k+1}; \; k = 0,1,...,P-1; \; x(0) = x_0. \qquad (7)$$

Искомое управление предлагается искать в виде функции насыщения sat, гарантирующей выполнение ограничений на управление:

$$u_j(\tau) = \mathrm{sat}\{g_j(\tau)\}, \; j \in \overline{1,q}; \qquad (8)$$

$$\mathrm{sat}\, g_j(\tau) = \begin{cases} a_j, \;\; g_j(\tau) \le a_j, \\ g_j(\tau), \; a_j < g_j(\tau) < b_j, \\ b_j, \;\; g_j(\tau) \ge b_j, \end{cases} \qquad \forall \tau \in [0,1]$$

$$g_j(\tau) = \sum_{i=0}^{P} c_i^j S(i, \tau).$$

В качестве базисной функций $S(i,\tau)$, можно использовать сплайны:

$S(i,\tau) = Sp*(\tau P - i),\ h = \frac{1}{P},\ i = 0,1,...,P$ - финитные функции, порожденные сплайнами, на отрезке $[0,1]$:

$$Sp*(t) = \begin{cases} 2^{p-1}(1+t)^p, t \in [-1,-\frac{1}{2}], \\ 1-2^{p-1}|t|^p, t \in [-\frac{1}{2},\frac{1}{2}], \\ 2^{p-1}(1-t)^p,\ t \in [\frac{1}{2},1], \\ 0, \qquad t \notin [-1,1], \end{cases}$$

где $Sp*(t)$ при $p=0$ задает кусочно-постоянный сплайн, при $p=1$ – кусочно-линейный (крышки), при $p=2$ – квадратичный, при $p=3$ – кубический.

МЕТОДИКА ПОДСЧЕТА ФУНКЦИОНАЛА

1. На каждой итерации выбранным метаэвристическим алгоритмом оптимизации генерируется расширенный вектор-столбец

$$(h_1,....,h_P|c_0^1,....,c_P^1|...|c_0^q,....,c_P^q)^T .$$

2. Интегрируя уравнение (7) с заданным начальным условием и управлением, вычисляемым по формуле (8), на отрезке $[0,1]$, получаем решение $x(\tau),\ \tau \in [0,1]$.

3. Подсчитать значение функционала $I = F(t_f, x(t_f)) = F(\sum_{k=0}^{P-1} h_{k+1}, x(1))$.

Алгоритм оптимизации, имитирующий поведение стаи серых волков

Метод серых волков (Grey Wolf Optimizer – GWO) имитирует охоту стаи серых волков за жертвой. Он относится к методам роевого интеллекта, в которых используется иерархия лидерства в стае и особый механизм охоты, заключающийся в отслеживании и приближении к жертве, ее последующем окружении и финальном нападении.

В начале работы метода случайным образом, используя предположение о равномерном распределении, на множестве допустимых решений D генерируется некоторый набор начальных точек (волков в стае): $I = \{x^j = (x_1^j, x_2^j, \dots, x_n^j)^T, j = 1, \dots, NP\} \subset D$, где x^j – вектор координат волка с номером j, NP – количество волков в стае. Поскольку в процессе охоты положение жертвы точно не известно вследствие ее постоянного движения (а в задаче оптимизации не известно положение точки экстремума), то члены стаи ориентируются на лидеров, полагая, что они обладают большей информацией о положении жертвы (точке экстремума).

В стае волков, где каждый волк характеризуется своей позицией в области допустимых решений, выбираются три последовательно лучших (α, β, γ) по величине целевой функции $f(x)$: $x^\alpha, x^\beta, x^\gamma$. Все волки в стае меняют свое положение с учетом сравнения своей текущей позиции с этими тремя наилучшими:

$$x^j(k+1) = \frac{x^{j,1}(k+1) + x^{j,2}(k+1) + x^{j,3}(k+1)}{3},$$

$$x^{j,1}(k+1) = x^{\alpha}(k) - A_{\alpha}^{j} \otimes D_{\alpha}^{j}(k),$$
$$x^{j,2}(k+1) = x^{\beta}(k) - A_{\beta}^{j} \otimes D_{\beta}^{j}(k),$$
$$x^{j,3}(k+1) = x^{\gamma}(k) - A_{\gamma}^{j} \otimes D_{\gamma}^{j}(k),$$

$$D_{\alpha}^{j}(k) = \left| C_{\alpha}^{j} \otimes x^{\alpha}(k) - x^{j}(k) \right|,$$
$$D_{\beta}^{j}(k) = \left| C_{\beta}^{j} \otimes x^{\beta}(k) - x^{j}(k) \right|,$$
$$D_{\gamma}^{j}(k) = \left| C_{\gamma}^{j} \otimes x^{\gamma}(k) - x^{j}(k) \right|,$$

где $\otimes$ – операция поэлементного произведения векторов по Адамару, k – номер итерации, $x^{j}(k+1), x^{j}(k)$ – следующее и текущее положения волков, $j = 1,...,NP$; $A_{\alpha}^{j}, A_{\beta}^{j}, A_{\gamma}^{j}$ – векторы, определяемые по правилу $A_{m}^{j} = 2a \otimes r_1 - a,\ m = \alpha, \beta, \lambda$; r_1 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$; a – вектор с одинаковыми компонентами, уменьшающимися линейно по закону $a_i = 2(1 - \frac{k}{K}),\ i = 1,...,n,$ K – максимальное число итераций; $C_{\alpha}^{j}, C_{\beta}^{j}, C_{\gamma}^{j}$ – векторы, определяемые по правилу $C_{m}^{j} = 2r_2, m = \alpha, \beta, \lambda$, r_2 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$. Имеется модификация, в которой $a_i = 2(1 - \frac{k^2}{K^2}),\ i = 1,...,n$. Программное обеспечение, подтвердило эффективность предложенного метода.

Алгоритм оптимизации, имитирующий поведение стаи горбатых китов

Метод горбатых китов (Whales Optimization Algorithm – WOA) имитирует охоту стаи горбатых китов за крилем или мелкой рыбой, используя уникальный способ преследования и окружения цели – движение из глубины к поверхности океана по спирали.

В начале работы метода случайным образом, используя предположение о равномерном распределении, на множестве допустимых решений D генерируется некоторый набор начальных точек (горбатых китов в стае): $I = \{x^j = (x_1^j, x_2^j, ..., x_n^j)^T, j = 1, ..., NP\} \subset D,$ где x^j – вектор координат кита с номером j, NP – количество горбатых китов в стае. Поскольку в процессе охоты положение жертвы точно неизвестно вследствие ее постоянного движения (в задаче оптимизации положение точки экстремума также неизвестно), то члены стаи ориентируются на лидера, полагая, что он обладает большей информацией о положении жертвы (точке экстремума).

В стае китов положение каждого кита характеризуется относительно лидера стаи. В процессе реализации охоты стая китов может, как исследовать множество допустимых решений в поисках косяка рыбы, так и окружать найденную добычу, постепенно приближаясь к жертве. Одним из возможных способов приближения является спиралевидное движение вокруг потенциальной добычи.

На каждой итерации для каждой особи ($j = 1, ..., NP$) генерируется случайное число p согласно равномерному закону распределения на отрезке $[0,1]$.

Если $p < 0,5$, то подсчитывается величина $\left|A^j\right|$, где A^j – вектор, определяемый по правилу $A^j = 2a \otimes r_1 - a;$ r_1 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$; a – вектор с одинаковыми компонентами, уменьшающимися линейно по закону

$a_i = 2(1 - \frac{k}{K}),\ i = 1,...,n,$ K – максимальное число итераций, знак $|\cdot|$ обозначает эвклидову норму вектора. Имеется модификация алгоритма, в которой координаты вектора a уменьшаются по квадратичному закону $a_i = 2(1 - \frac{k^2}{K^2}),\ i = 1,...,n$.

В случае, когда $|A^j| < 1$, используется формула, по которой новое положение кита определяется относительно расположения жертвы:

$$x^j(k+1) = x^*(k) - A^j \otimes D^j(k),$$

$$D^j(k) = \left| C^j \otimes x^*(k) - x^j(k) \right|,$$

где $x^*(k)$ – лучшая особь на текущей итерации, $\otimes$ – операция поэлементного произведения векторов по Адамару, k – номер итерации, $x^j(k+1), x^j(k)$ – следующее и текущее положения китов, $j = 1,..., NP$; C^j – вектор, определяемый по правилу $C^j = 2r_2$, r_2 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$. В этом случае реализуется процесс разработки найденного источника пищи.

Если $|A^j| \ge 1$, то реализуется исследование (поиск новых источников пищи) на множестве допустимых решений:

$$x^j(k+1) = x_{rand}(k) - A^j \otimes D^j(k),$$

$$D^j(k) = \left| C^j \otimes x_{rand}(k) - x^j(k) \right|,$$

где $x_{rand}(k)$ – случайно выбранная особь на текущей итерации.

Если $p \ge 0{,}5$, то новое положение кита определяется по формуле

$$x^j(k+1) = D^{*j}(k) \cdot e^{\beta l} \cdot \cos(2\pi l) + x^*(k),$$

$$D^{*j}(k) = \left| x^{*}(k) - x^{j}(k) \right|,$$

где l – случайная величина, распределенная равномерно на отрезке $[-1, 1]$, β – параметр логарифмической спирали.

Алгоритм оптимизации, имитирующий поведение стаи стрекоз

Алгоритм (Dragonfly Algorithm – DA) имитирует как статическое, так и динамическое поведение семейства стрекоз в природе, реализуя три принципа: разделение (уклонение одного индивидуума от другого внутри окрестности его расположения), выравнивание (в окрестности положения индивидуума скорости стрекоз полагаются одинаковыми), сплоченность (индивидуумы перемещаются к центру масс семейства). Конфигурация стаи стрекоз и направление их полета определяются задачей поиска пищи (стремлением к лидеру стаи), уклонением от хищника (наихудшего решения), памятью о предыдущем передвижении. Нахождение нового положения стрекозы (положения решения в множестве допустимых решений) производится либо под влиянием ближайших соседей в стае (если их число превышает единицу), либо посредством независимого перелета с использованием распределения Леви (в противном случае). Используя описанные принципы, стая стрекоз осуществляет разведку новых областей и разработку старых. Характерные черты поведения стрекоз используются при решении задачи поиска экстремума функции на множестве допустимых решений. Процедура поиска завершается при достижении заданного числа итераций.