



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М.В. ЛОМОНОСОВА  
Факультет вычислительной математики и кибернетики  
Кафедра системного анализа

**Айтеев Альнур Тулендыулы**

# **Параллельные вычисления в задачах управления**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Научный руководитель:**

к.ф.-м.н., доцент

И.В. Востриков

Москва, 2022

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Постановка задачи</b>	<b>3</b>
<b>3</b>	<b>Составление математической модели</b>	<b>4</b>
3.1	Изучение физической модели . . . . .	4
3.2	Расчет оптимального угла паруса . . . . .	6
3.3	Способы управления и виды движения . . . . .	8
3.4	Итоговая система . . . . .	9
<b>4</b>	<b>Теория динамического программирования</b>	<b>11</b>
4.1	Функция цены . . . . .	11
4.2	Множества достижимости и разрешимости. Их связь с функцией цены	12
4.3	Задача быстрогодействия . . . . .	13
<b>5</b>	<b>Численный алгоритм синтеза управления</b>	<b>15</b>
5.1	Расчет функции цены . . . . .	15
5.2	Синтез управления . . . . .	15
5.3	Расчет модифицированной функции цены . . . . .	18
<b>6</b>	<b>Вычисление функции цены и оптимального управления путём парал- лельных вычислений</b>	<b>19</b>
6.1	Основные понятия . . . . .	19
6.2	Вычисление всех $t_{ij}$ . . . . .	20
6.3	Вычисление функции цены $V(i, j)$ . . . . .	21
6.4	Вычисление оптимального пути для функции цены $V(i, j)$ . . . . .	21
6.5	Вычисление функции цены $V(i, j, p)$ . . . . .	21
6.6	Вычисление оптимального пути для функции цены $V(i, j, p)$ . . . . .	22
<b>7</b>	<b>Заключение</b>	<b>22</b>

# 1 Введение

Данная работа посвящена задаче управления движением для математической модели парусного судна и решение нахождения оптимального пути на ЭВМ используя параллельные методы. Решение задач управления зачастую связано с решением дифференциальных уравнений, что часто неразрешимо за полиномиальное время и может требовать от специалиста хороших знаний в области численных методов, математического моделирования и параллельного программирования. Кроме того, задачи управления возникают во многих областях человеческой деятельности и одним из способов решения таких задач является метод динамического программирования, разработанный Р.Беллманом, ([3]) в котором вводятся функции цены, являющиеся решениями соответствующих экстремальных задач ([4], [1], [8], [2]). Функция цены, как таковая, является решением дифференциального уравнения в частных производных, называемого уравнением Гамильтона - Якоби - Беллмана. Данный метод позволяет получать позиционное управление, а синтезирующая стратегия находится как множество управлений, на которых достигается экстремум в этом уравнении. Для сведения нелинейной модели парусного судна к линейной динамической модели вводится ряд допущений, благодаря чему введенный класс управлений допускает решение поставленной задачи методом динамического программирования, в котором искомая функция цены является решением дискретного уравнения Беллмана.

## 2 Постановка задачи

В данной работе мы поставили следующую задачу: вычислить оптимальное управление передвижения парусного судна из пункта А в пункт В за минимальное время с использованием параллельных методов. Данная задача будет разбита на следующие этапы:

- Составление математической модели движения парусного судна.
- Определение возможностей управления и его класса, введение ограничений.
- Изучение теории динамического программирования, задача которых состоит в решении подобных задач.
- Вычисление оптимального управления движением судна с помощью методов динамического программирования в рамках построенной модели с применением параллельных методов.

## 3 Составление математической модели

### 3.1 Изучение физической модели

Модель парусного судна составлена таким образом, что движение осуществляется за счёт ветра, действующего на парус. Отсутствуют какие-либо дополнительные силы, помогающие судну в движении. Парус является таким же двигателем для яхты, как гребной винт для моторного судна. Благодаря работе паруса с его помощью энергия ветра преобразуется в силу тяги, движущую парусник. Коэффициент полезного действия парусов и, как следствие, скорость судна на любом курсе относительно потока воздуха, прямо зависит от грамотно спроектированного парусного вооружения судна и профессионализма экипажа в эксплуатации средств для настройки и управления парусами. Данная модель рассматривает парус как обычный тканевой лист, без учёта аэродинамического удлинения и формы контура паруса. Особенность и влияние на движение разобраны в работах [6] и [7].

Рассмотрим силы, действующие на судно:

$$F_{tr} = k_1 v^2,$$

где  $F_{tr}$  — сила сопротивления воды, направленная противоположно движению,  $v$  — скорость судна,  $k_1$  — коэффициент трения/сопротивления воды.

$$F_d = k S v_{res}$$

где  $F_d$  — сила давления ветра на парус,  $S$  — площадь паруса,  $k$  — коэффициент полезного действия силы ветра,  $v_{res}$  — перпендикулярная составляющая скорости результирующего ветра.

Для данной модели применим второй закон Ньютона:

$$F_d + F_{tr} = ma \Rightarrow -k_1 v^2 + k S v_{res} = ma \quad (1)$$

Пусть сопротивление воды отсутствует ( $k_1 = 0$ ),  $m$  — масса судна, рассмотрим подробнее  $v_{res}$  — скорость ветра действующего на парус:

$$v_{res} = v_{\perp} - v \sin \theta$$

где  $\theta$  — угол между парусом и носом судна,

$$v_b = v_{\perp} + v_{\parallel}$$

где  $v_b$  — скорость истинного ветра,  $v_{\perp}$  — его составляющая, перпендикулярная парусу,  $v_{\parallel}$  — параллельная составляющая, не влияющая на движение.

**Определение 3.1.** *Истинный ветер - морской термин, обозначающий горизонтальный поток атмосферного воздуха над водной поверхностью, направление и скорость которого измеряются относительно стационарных морских или наземных объектов.*

Заметим, что:

$$v_{\perp} = v_b \sin \alpha$$

где  $\alpha$  - угол между парусом и ветром (см. рис. 1).

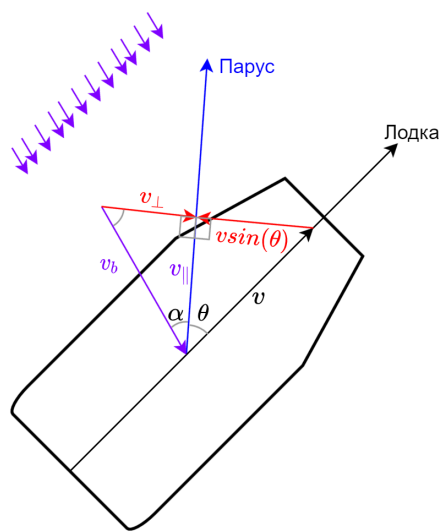


Рис. 1: Результирующий ветер.

Получим значение результирующего ветра:

$$v_{res} = v_b \sin \alpha - v \sin \theta$$

В данной модели нас будет интересовать ускорение по направлению движения. Так как парус редко направлен перпендикулярно движению, для получения нужного нам ускорения необходимо спроецировать уравнение (1) на направление движения судна. Получим в общем виде:

$$-k_1 v^2 + (v_b \sin \alpha - v \sin \theta) \sin \theta = ma \Rightarrow a = \frac{-k_1 v^2 + (v_b \sin \alpha - v \sin \theta) k S \sin \theta}{m}.$$

Теперь рассмотрим параметры, которые нам будут известны, и их связь с уже введенными ранее параметрами системы (см. рис. 2):

- $\alpha_v$  - угол ветра относительно общей неподвижной системы координат;

- $\theta$  - угол между парусом и носом;
- $\gamma$  - угол направления движения судна относительно оси  $Ox$  в общей системе координат.

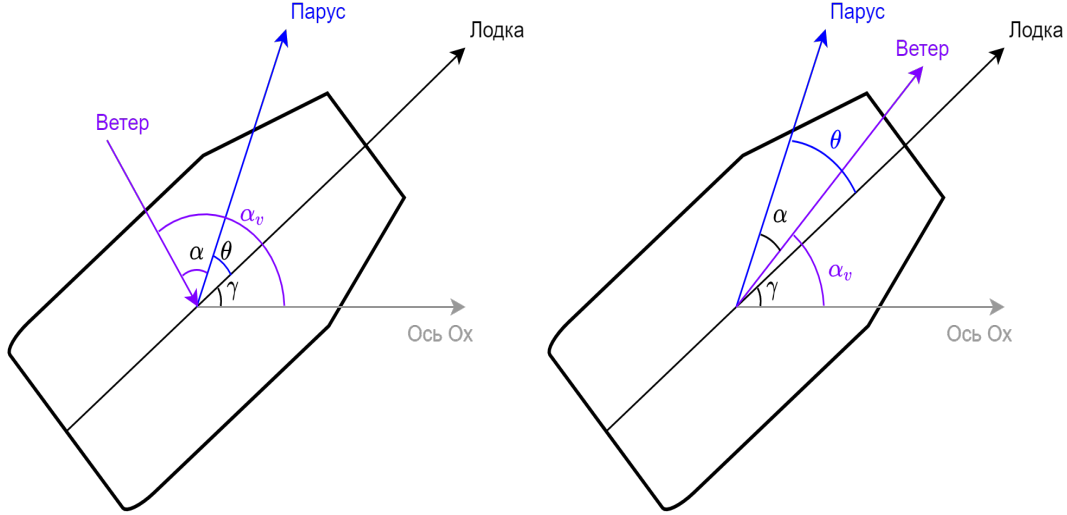


Рис. 2: Параметры модели.

Заметим, что угол  $\alpha$  непосредственно рассчитывается с помощью указанных выше параметров следующим образом (см. рис. 2):

$$\alpha = ||\alpha_v - \gamma| - \theta|$$

### 3.2 Расчет оптимального угла паруса

В данной работе мы хотим добиться движения с максимально возможной скоростью вдоль направления. Для этого необходимо рассчитать угол паруса, который позволит получить максимальное ускорение перед началом движения, с места, а значит и максимальную возможную скорость в данный момент времени. Пусть начальная скорость равна нулю, т.е.  $v = 0$ , и  $k_1 = 0$ . В дальнейшем сопротивлением воды мы пренебрежем, так как оно незначительно влияет на поведение движения судна.

Тогда получим:

$$a = \frac{(v_b \sin \alpha) k S \sin \theta}{m}$$

Для расчёта угла  $\theta$ , при котором будет достигаться максимум, введём вспомогательную функцию:

$$\begin{aligned} f(\theta) &= v_b \sin \alpha \sin \theta, \alpha = \theta - (\alpha_v - \gamma) \Rightarrow \\ f(\theta) &= v_b \sin (\theta - (\alpha_v - \gamma)) \sin \theta \end{aligned}$$

Приравняв ее производную к нулю, получим интересующие нас значения  $\theta$  :

$$\begin{aligned}
 f(\theta) &= v_b (\sin \theta \cos (\alpha_v - \gamma) - \cos \theta \sin (\alpha_v - \gamma)) \sin \theta = \\
 &= v_b \sin^2 \theta \cos (\alpha_v - \gamma) - v_b \cos \theta \sin \theta \sin (\alpha_v - \gamma) \Rightarrow \\
 f'(\theta) &= v_b (2 \sin \theta \cos \theta \cos (\alpha_v - \gamma) + \sin^2 \theta \sin (\alpha_v - \gamma) - \cos^2 \theta \sin (\alpha_v - \gamma)) = 0 \Rightarrow \\
 &\sin 2\theta \cos (\alpha_v - \gamma) + 2 \sin^2 \theta \sin (\alpha_v - \gamma) - \sin (\alpha_v - \gamma) = 0 \Rightarrow \\
 &\sin 2\theta \cos (\alpha_v - \gamma) + \sin (\alpha_v - \gamma) - \cos 2\theta \sin (\alpha_v - \gamma) - \sin (\alpha_v - \gamma) = 0 \Rightarrow \\
 &\sin 2\theta \cos (\alpha_v - \gamma) - \cos 2\theta \sin (\alpha_v - \gamma) = 0 \Rightarrow \sin 2\theta \cos (\alpha_v - \gamma) = \cos 2\theta \sin (\alpha_v - \gamma) \Rightarrow \\
 &\operatorname{tg} 2\theta = \operatorname{tg} (\alpha_v - \gamma) \Rightarrow \theta = \frac{\alpha_v - \gamma}{2} \pm \frac{\pi n}{2}, n \in \mathbb{Z}
 \end{aligned}$$

Как видно из значения  $\theta$ , оптимальное положение паруса такое, что он разбивает угол между направлением ветра и направлением движения судна ровно пополам. Отметим некоторые особенности:

- $\frac{\alpha_v - \gamma}{2} + \frac{\pi}{2}$ , появляется в случае, когда судно движется по ветру, т.к. (см. рис. 3)

$$\beta = \frac{\pi - (\alpha_v - \gamma)}{2} = \frac{\pi}{2} - \frac{\alpha_v - \gamma}{2} \Rightarrow \theta = \pi - \beta = \frac{\alpha_v - \gamma}{2} + \frac{\pi}{2}$$

- $\frac{\alpha_v - \gamma}{2} + \pi$  в силу симметрии аналогично случаю  $\frac{\alpha_v - \gamma}{2}$ , другие углы нас также не интересуют.

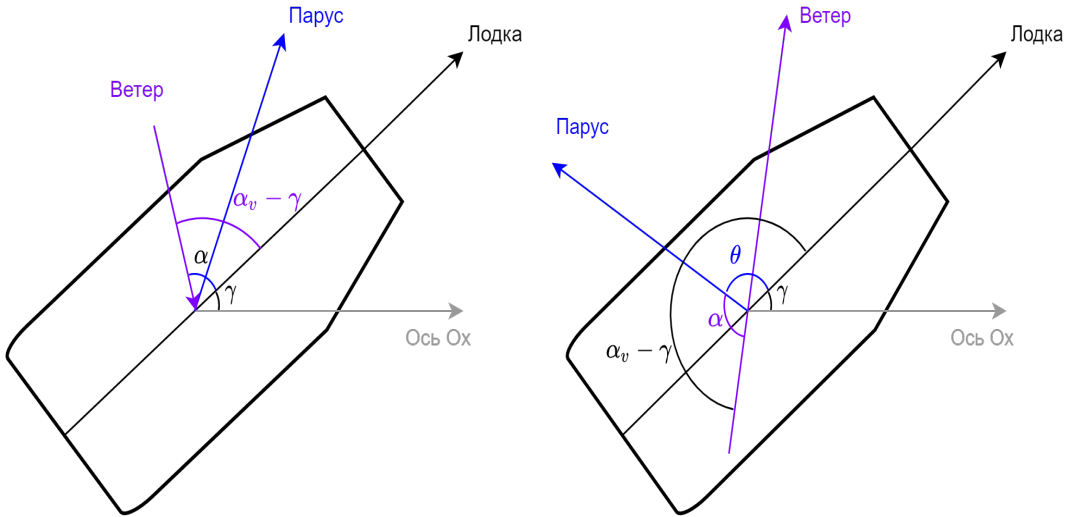


Рис. 3: Оптимальное положение паруса, для максимального ускорения.

В итоге получили два оптимальных положения паруса: в случае, когда судно идет по ветру  $-\frac{\alpha_v - \gamma}{2} + \frac{\pi}{2}$ , и  $\frac{\alpha_v - \gamma}{2}$  — против ветра (см. рис. 3).

Для подтверждения теории приведем замеры времени движения лодки между пунктами, лежащими неподалеку друг от друга. Расстояние между ними возьмем

небольшое, т.к. нас интересует моментальное ускорение. Параметры "по ветру":  $\alpha_v = \frac{\pi}{3}, \gamma = \frac{\pi}{3}$ ; "против ветра":  $\alpha_v = \frac{\pi}{2}, \gamma = 3.5\frac{\pi}{18}$ , остальные значения параметров одинаковы для всех испытаний и в данном случае не важны, поэтому уточняться не будут. Очевидно, что напрямик против ветра мы двигаться не можем, а строго по ветру - да. Из этих соображений в дальнейшем мы введем дополнительные ограничения на возможные углы движения судна относительно направления ветра.

### 3.3 Способы управления и виды движения

В предыдущих пунктах мы нашли оптимальный угол для достижения максимального моментального ускорения, но этот угол будет актуальным, если результирующий ветер, действующий на парус, постоянный, а это не так.

Таблица 1: Время движения на короткую дистанцию, с различными углами паруса.

Движение	$\theta$	Время
по ветру (не опт.)	$\frac{(\alpha_v - \gamma)}{2} + \frac{\pi}{2} - \frac{\pi}{18}$	5.9
по ветру (опт.)	$\frac{(\alpha_v - \gamma)}{2} + \frac{\pi}{2}$	5.6
против ветра (не опт.)	$\frac{(\alpha_v - \gamma)}{2} - \frac{\pi}{18}$	34.0
против ветра (опт.)	$\frac{(\alpha_v - \gamma)}{2}$	32.5
против ветра (не опт.)	$\frac{(\alpha_v - \gamma)}{2} + \frac{\pi}{18}$	34.8

Как только судно приобретает скорость, за счет его движения результирующий ветер, действующий на парус, меняет свое направление и значение. Это важно учитывать, если хотим передвигаться с максимальной скоростью. Именно поэтому мы вводим дополнительные пересчеты оптимального угла, с учетом изменения результирующего ветра каждые  $k$  секунд. Чем чаще мы корректируем угол, тем больше выигрыша в скорости мы получим, но исходя из физических возможностей, слишком часто мы этого делать не можем. Поэтому возьмём  $k = 20$ .

Разберём подробнее, как именно пересчитывается результирующий ветер, действующий на парус (см. рис. 4). После получения новых параметров оптимальный угол паруса поменяет свое положение на  $\angle BAC = \delta$ , (см. рис. 4). Чтобы вычислить его величину, дважды воспользуемся теоремой косинусов:

$$v_{\text{bnew}}^2 = v_b^2 + v^2 - 2v_b v \cos(\angle CBA), \angle CBA = \pi - (\alpha_v - \gamma);$$

$$\cos(\angle BAC) = \frac{v_b^2 + v_{\text{bnew}}^2 - v^2}{2v_b v_{\text{bnew}}} \Rightarrow \angle BAC = \delta = \arccos\left(\frac{v_b^2 + v_{\text{bnew}}^2 - v^2}{2v_b v_{\text{bnew}}}\right).$$

Как несложно заметить, при движении "по ветру"  $\theta$  будет увеличиваться на  $\delta$ , а "против ветра" уменьшаться на него же. Другими словами, парус будет стремиться к "направлению движения но не достигать его.

Приведем измерения, которые подтверждают, что пересчет оптимального угла дает выигрыш во времени: с пересчетом время = 391, без изменения угла время = 365.



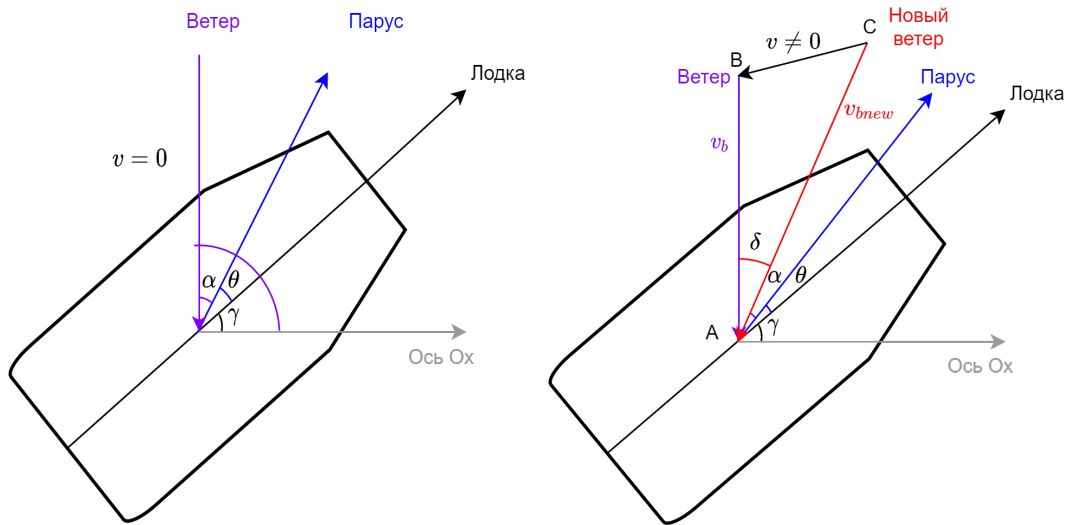


Рис. 4: Корректировка паруса по ходу движения, из-за изменения результирующего ветра.

Итак, с управлением паруса мы разобрались. Осталось оговорить, как часто и каким образом мы можем сменять направление движения судна  $\gamma$ . Смещением не по направлению движения мы пока пренебрегаем. Чтобы повернуть судно в нашей модели, его нужно остановить. Это означает, что в последующем движении мы опять тратим время на разгон с нуля и установку нужного угла паруса. Поэтому менять направление судна будем только при переходе от "уровня к уровню" т.е. при достижении нужного множества, просчитанного заранее геометрически. Тем самым управление углом  $\gamma$  просчитывается, исходя из геометрических соображений при расчете траектории движения. Каким именно образом рассчитывается управление углом  $\gamma$ , будет изложено в дальнейшем. Введем ограничения на это управление, а именно: двигаться в секторе 30 градусов относительно направления против истинного ветра - невозможно.

Получили два возможных типа управления и движения:

1. Движение по направлению  $\gamma$  с максимальным ускорением.
2. Поворот, смена направления движения  $\gamma_1 \rightarrow \gamma_2$ .

### 3.4 Итоговая система

Учитывая все выше сказанное, будем решать нашу задачу следующим образом. Разобъем расстояние между пунктами назначения на  $n$  уровней ( $n$  должно быть небольшим, расстояние между уровнями рассчитывается исходя из геометрических соображений, наличия препятствий, ограничений и погодных условий). Будем перемещаться от уровня к уровню по направлению  $\gamma_i$  с максимальным ускорением, меняя положение паруса (т.е. угол  $\theta$ ) для этого каждые  $k$  секунд.

Получим следующие системы, где  $i = \overline{1, n}$ .

Движение по направлению:

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = \frac{-k_1 x_{2i}^2 + (v_{bi} \sin \alpha(t) - x_{2i} \sin \theta_i(t)) k S \sin \theta_i(t)}{m} \\ \theta_i(t) - \text{кусочно постоянная функция} \\ \alpha(t) = ||\alpha_{vi} - \gamma_i| - \theta_i(t)| \\ t \in [t_{0i}, t_{1i}], t_{01} = 0. \end{cases}$$

Перемещение в общей (статической) системе координат:

$$\begin{cases} \dot{x}_i = x_{1i} \cos \gamma_i + x_{i-1} (t_{1(i-1)}) \\ \dot{y}_i = x_{1i} \sin \gamma_i + y_{i-1} (t_{1(i-1)}) \\ x_{11} = x_0, y_{11} = y_0 \end{cases}$$

Параметры управления рассчитываются из введенного ранее класса управлений (со всеми ограничениями).

Чтобы построить оптимальное управление углом  $\gamma$ , воспользуемся методами динамического программирования. Для этого сначала ознакомимся с возможными методами решения данной задачи в общем виде.

## 4 Теория динамического программирования

### 4.1 Функция цены

Рассмотрим задачу:

$$\begin{aligned}\dot{x} &= f(t, x, u), u \in U \\ x(t_0) &= x_0\end{aligned}$$

$U$  - множество кусочно-гладких, непрерывных слева функций. Требуется минимизировать функционал:

$$J(u(\cdot)) = \int_{t_0}^{t_1} L(t, x(t), u(t))dt + \phi(t_1, x(t_1))$$

где  $t_1$  — фиксировано.

Перейдем к рассмотрению принципа динамического программирования. Пусть  $t_0$  и  $x_0$  — параметры системы.

**Определение 4.1.** *Функцией цены называется функция*

$$V(t_0, x_0) = \inf_{u(\cdot) \in U} \left( \int_{t_0}^{t_1} L(t, x(t), u(t))dt + \phi(t_1, x(t_1)) \right)$$

Это соотношение называют основным уравнением динамического программирования. Будем обозначать  $V(t, x)$  как  $V(t, x | \phi(t_1, \cdot))$ , если  $V(t_1, x) = \phi(t_1, x)$ . Тогда справедливо соотношение:

$$V(t_0, x_0 | V(t_1, \cdot)) = V(t_0, x_0 | V(\sigma, \cdot | V(t_1, \cdot)))$$

называемое полугрупповым свойством для функции цены.

Пусть функция  $V(t, x)$  дифференцируема в точке  $(t, x)$ . Тогда, согласно [4] справедлива следующая система:

$$\begin{cases} \frac{\partial V}{\partial t}(t, x) + \inf_u \{ \langle \frac{\partial V}{\partial x}(t, x), f(t, x, u) \rangle + L(t, x, u) \} = 0 \\ V(t_1, x) = \phi(t_1, x) \end{cases}$$

Первое уравнение этой системы называется попятным уравнением Беллмана.

Введем обозначение:

$$U_t = \{u = u(t) : u(\cdot) \in U\}$$

**Лемма 4.0.1.** 1. Пусть  $V(t, x)$  дифференцируема в точке  $(t, x)$ . Тогда,

$$\frac{\partial V}{\partial t}(t, x) + \left\langle \frac{\partial V}{\partial x}(t, x), f(t, x, v) \right\rangle + L(t, x, v) \geq 0, \forall v \in U_t$$

2. Если дополнительно существует оптимальное управление  $u^*(\cdot)$  :

$$u^*(s) \rightarrow v^* \text{ при } s \rightarrow t, \text{ то}$$

**Теорема 4.1.** (о верификации). Пусть  $W(t, x)$  — гладкая функция, удовлетворяющая уравнению Беллмана и граничным условиям в некоторой области. Тогда в этой области

$$W(t, x) \leq V(t, x)$$

Если при этом существует такое управление  $u^*(\cdot)$ , что для соответствующей ему траектории  $x^*(\cdot)$  выполнено:

$$\begin{aligned} L(s, x^*(s), u^*(s)) + \left\langle \frac{\partial W}{\partial x}(s, x^*(s)), f(s, x^*(s), u^*(s)) \right\rangle = \\ = \min_{u \in U} \left\{ L(s, x^*(s), u) + \left\langle \frac{\partial W}{\partial x}(s, x^*(s)), f(s, x^*(s), u) \right\rangle \right\} \end{aligned}$$

то  $u^*(\cdot)$  — оптимальное управление, и  $W(t, x) = V(t, x)$ .

**Определение 4.2.** Управление  $u(\cdot, \cdot)$  как функция времени и фазовых переменных называется управлением в форме стратегии (стратегией).

**Определение 4.3.** Управление  $u(\cdot)$  как функция времени называется управлением в форме программы (программой).

**Следствие 4.1.1.**

$$u(t, x) = \arg \min_{u \in U} \left\{ \left\langle \frac{\partial V}{\partial x}, f(t, x, u) \right\rangle + L(t, x, u) \right\}$$

— оптимальная стратегия.

**Замечание 4.1.** Имея управление в форме стратегии  $u(t, x)$ , можно легко найти соответствующую программу, решая дифференциальное уравнение

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

относительно  $x(\cdot)$  и осуществляя подстановку  $u^*(t) = u(t, x(s))$ .

## 4.2 Множества достижимости и разрешимости. Их связь с функцией цены

Рассмотрим линейное уравнение:

$$\dot{x} = A(t)x(t) + B(t)u(t), t \in [t_0, t_1]$$

$$x(t) \in \mathbb{R}^n, A(t) \in \mathbb{R}^{n \times n}, u(t) \in \mathbb{R}^m, B(t) \in \mathbb{R}^{n \times m}, u(t) \in \mathcal{P}(t), \mathcal{P}(t) \in \text{conv } \mathbb{R}^m$$

**Определение 4.4.** Множеством достижимости в момент времени  $t$  из множества  $X^0$  называется следующее множество точек:

$$\mathcal{X}(t; t_0, X^0) = \{x \mid \exists u(s) \in \mathcal{P}(s), t_0 \leq s \leq t, x^0 \in X^0 : x(t; t_0, x^0) = x\}$$

Другими словами, множеством достижимости  $X(s)$  называется сечение пучка траекторий в момент времени  $s$ .

**Определение 4.5.** Множеством разрешимости во множество  $M$  (целевое множество) называется множество

$$\mathcal{W}(t; t_1, M) = \{x \mid \exists u(s) \in \mathcal{P}(s), t \leq s \leq t_1, x^1 \in M : x(t_1; t, x) = x^1\}$$

**Определение 4.6.** Многозначные отображения  $\mathcal{X}(\cdot)$ ,  $\mathcal{W}(\cdot)$  называются трубками достижимости и разрешимости соответственно. Рассмотрим функции цены:

$$\begin{aligned} V^1(t, x) &= \min_{u(\cdot)} \{d^2(x^0, X^0) \mid x(t; t_0, x^0) = x\}, \\ V^2(t, x) &= \min_{u(\cdot)} \{d^2(x^1, M) \mid x(t_1; t, x) = x^1\} \end{aligned}$$

**Утверждение 4.1.**

$$\begin{aligned} \mathcal{X}(t; t_0, X^0) &= \{x \mid V^1(t, x) \leq 0\} \\ \mathcal{W}(t; t_1, M) &= \{x \mid V^2(t, x) \leq 0\} \end{aligned}$$

Для того, чтобы задача нахождения управления была разрешима, необходимо, чтобы множество разрешимости в целевое множество соержжало стартовую позицию. Аналогично с множеством разрешимости из стартового множества, оно должно пересекаться с целевым множеством.

### 4.3 Задача быстрогодействия

Задача быстрогодействия заключается в том, чтобы за минимальное время попасть в заданную точку (в заданное множество).

$$\begin{cases} \dot{x} = Ax + Bu + f, u \in \mathcal{P} \\ x(t_0) = x^0 \\ x(\tau) = x \end{cases}$$

$$J(u) = \int_{t_0}^{\tau} dt \rightarrow \inf_{u(\cdot)}$$

Один из методов решения слудующий:

$$d^2(x(t_0), x^0) \rightarrow \inf_{u(\cdot)}$$

Функция цены данной задачи  $-V(\tau, x \mid t_0, x^0) = \min_{u(\cdot)} \{d^2(x(t_0), x^0) \mid x(\tau) = x\}$ .

Рассмотрим уравнение

$$V(\tau, x \mid t_0, x^0) = 0$$

и будем искать его минимальный корень  $\tau^* \geq t_0$ .

**Теорема 4.2.**  $\tau^*$  — минимальный корень уравнения (4.3)  $\Leftrightarrow \tau^*$  — оптимальное время перехода.

Также подробно изложены методы решения задач линейного предельного быстрого действия в работе [5].

## 5 Численный алгоритм синтеза управления

В нашей задаче будем рассматривать

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

где  $u(t) = [\gamma(t), \theta(t)]$ .

Рассмотрим следующий функционал:

$$J(u(\cdot)) = \int_{t_0}^{\tau} dt + \phi(\tau, x(\tau)) \rightarrow \min$$

где  $\tau$  — время попадания на границу целевого множества,  $\phi(\tau, x(\tau))$  — терминальные ограничения. Данный функционал минимизирует время перехода между заданными пунктами.

Изначально система, с помощью которой описывается движение нашего парусного судно была нелинейна, но после сделанных нами допущений, программного управления углом  $\theta$ , и введения ограничений, мы ее свели к линейной динамической модели, где остается вычислить управление углом  $\gamma$  при известных кусочно-постоянных остальных параметрах.

Для этого рассчитаем функцию цены и дискретно решим уравнение Беллмана.

### 5.1 Расчет функции цены

Будем рассчитывать функцию цены на сетке  $n \times m$ , где  $n$  — количество уровней, а  $m$  — количество точек на каждом уровне. Терминальные ограничения отвечают за попадание в нужную точку или ее окрестность. Чем больше значение этой функции, тем нежелательнее в конце движения оказаться именно в этой окрестности. Для вычисления функции цены, воспользуемся ее полегрупповым свойством.

$$V(n, j) = \phi(j),$$
$$V(i, j) = \min_{l \in \overline{1, m}} \{V(i+1, l) + t_{ij}\}, \quad i \in \overline{1, n-1}, \quad j \in \overline{1, m}$$

где  $t_{li}$  — время перехода из точки  $(i, j)$  в точку  $(i+1, l)$ .

### 5.2 Синтез управления

Теперь воспользуемся полученной функцией цены для определения оптимального управления, а именно угла  $\gamma_i$ , т.к. остальные углы рассчитываются по известными формулам, зависящим от  $\gamma_i$ .

Чтобы получить оптимальные  $\gamma_i$ , будем идти из стартовой точки  $(1, j_0)$ , перемещаясь по узлам сетки.  $j_{1opt} = j_0, j_{iopt}$  — положение на  $i$ -ом уровне при оптимальном управлении.

При перемещении с  $i$ -ого уровня на  $(i+1)$ -ый:

$$j_{(i+1)opt} = \arg \min_{j \in \overline{1, m}} (t_{i,j} + V(i+1, j))$$

где  $t_{i,j}$  — время перехода из  $(i, j_{iopt})$  в  $(i+1, j)$ .  
Пусть  $h$  — расстояние между уровнями. Тогда

$$\gamma_{iopt} = \arctg \left( \frac{h}{V(i+1, j_{(i+1)opt}) - V(i, j_{iopt})} \right)$$

Рассмотрим примеры. Пусть

$$n = 11, m = 21, k = 5, v = 5, \alpha_v = \frac{\pi}{2}, S = 10, m = 1000, h = 100$$

Как видно из результатов, оптимальных управлений получилось несколько. Все они дают одно и то же минимальное время  $t_{opt} = 644$ . Чтобы понять почему оптимальные углы определяются именно таким образом, для постоянного ветра можно посмотреть на график зависимости времени передвижения через один уровень от угла  $\gamma$ , (см. рис. (6)). Из-за симметрии при  $\alpha_v = \frac{\pi}{2}$ , на графике представлены углы от 20 до 50 градусов.

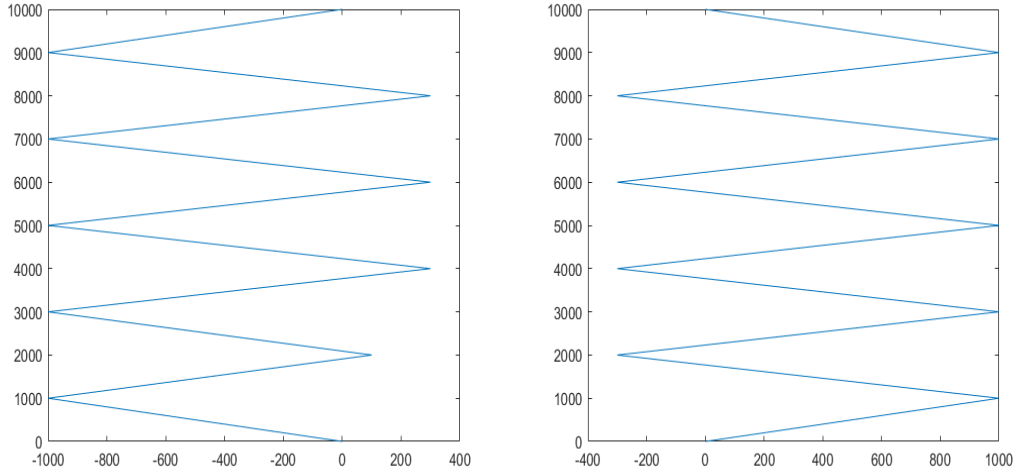


Рис. 5: Оптимальное управление в случае полной остановки судна для разворота

Можно заметить, что, когда управление не меняет угол  $\gamma$ , скорость можно не сбрасывать, это не учитывается при данном синтезе. Поэтому для этого случая функция цены даст гарантированную внешнюю оценку времени, а на практике, оптимальное время будет в разы меньше, в рассмотренном примере  $t_{opt 2} = 160$  (см. рис. 7).



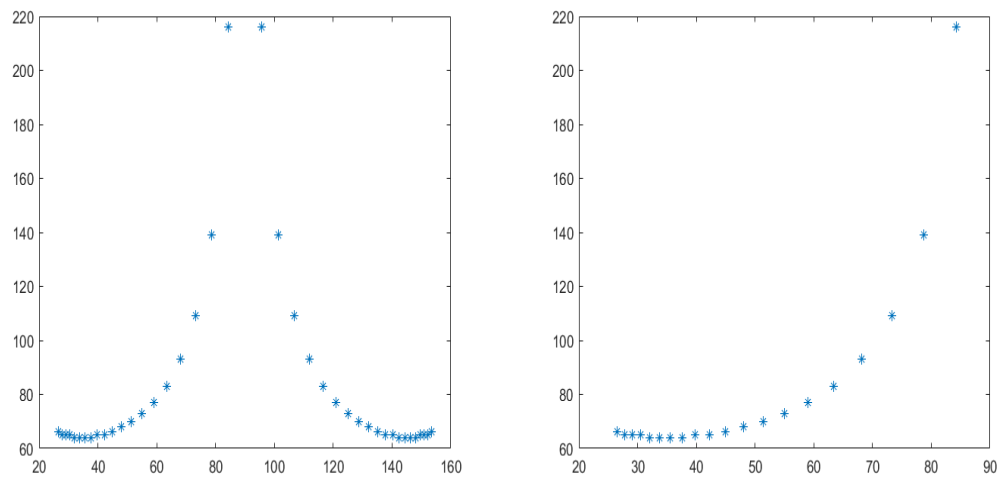


Рис. 6: Зависимость времени перехода между уровнем от  $\gamma$ .

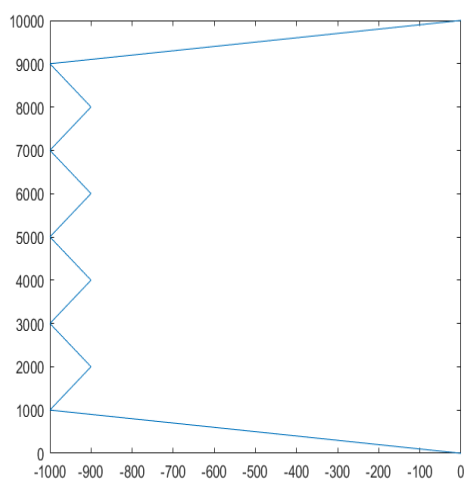


Рис. 7: Оптимальное управление, с учетом возможности не сбрасывать скорость.

### 5.3 Расчет модифицированной функции цены

В предыдущем пункте были указаны недостатки разобранной функции цены. С учетом большего количества сохраняемой и обрабатываемой информации, функция цены даст более точные результаты. Для хороших результатов при расчете функции цены надо учитывать время на поворот судна, разгон, остановку и настройку паруса. При этих условиях, для ее расчета нужно сохранять почти всю предысторию движения судна. Поэтому в данном разделе рассмотрим модифицированную функцию цены, учитывающую лишь часть предыстории, а именно, угол движения перед переключением  $\gamma$ . В этом случае потребуется в  $q$  раз больше памяти, и помимо граничных условий будет учтено время поворота судна.

Разобьем возможные углы поворота судна на  $q$  секторов. Если при переходе с одного уровня на другой судно поворачивает из  $k$ -ого сектора в  $l$ -ый, время на поворот будет рассчитываться через вспомогательную функцию:

$$\Gamma(k, l) = \beta |k - l|$$

В каждом узле сетки будет храниться  $q$  значений функций цены, чтобы минимизировать время с учетом разворотов судна.

Распишем новую функцию цены:

$$V(n, j, p) = \phi(j), \quad j \in \overline{1, m}, p \in \overline{1, q}$$

$$V(i, j, p) = \min_{p_f} \left\{ \Gamma(p, p_f) + \min_{j_f} (t_{i, j_f} + V(i + 1, j_f, p_f)) \right\}$$

где  $p$  - угол, под которым пришли в точку  $(i, j)$  с  $(i - 1)$ -ого уровня,  $p_f \in \overline{1, q}$ ,  $j_f$  - точки на  $(i + 1)$ -ом уровне относящиеся к  $p_f$  сектору,  $t_{i, j_f}$  - время перехода из точки  $(i, j)$  в точку  $(i + 1, j_f)$ . Расчет синтеза управления происходит аналогично предыдущему разделу.

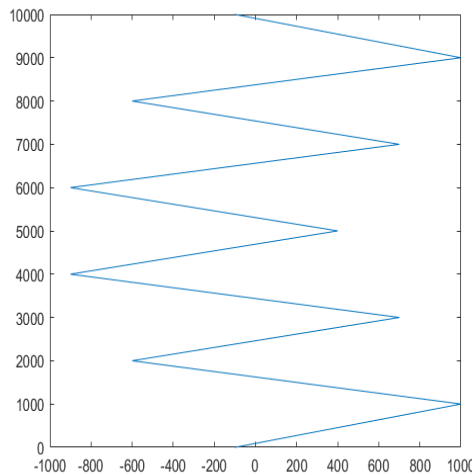


Рис. 8: Управление с учетом времени поворота при полной остановке.  $t_{opt} = 642$

## 6 Вычисление функции цены и оптимального управления путём параллельных вычислений

В основе архитектуры любого современного вычислительного устройства лежит два вида параллельной обработки данных: конвейерность и параллелизм. В условиях нашей задачи мы рассматриваем конвейер со стороны суперскалярной архитектуры, а значит, возложим эту ответственность на компилятор, как работу векторных процессорных инструкций MMX, SSE и AVX, в то время как параллелизм рассматриваем со стороны параллельного программирования на базе OpenMP. (использование потоков и контроллера для распределения работы между ядрами процессора) Каждый из методов эффективен наилучшим образом в конкретно поставленной задаче. К примеру, конвейер хорошо работает с последовательной обработке однородного пакета данных, например, сложение векторов размерности 512 бит, содержащие в себе числа двойной точности double, в то время как параллельная обработка эффективна с потоком независимых друг от друга данных, как в случае классического метода перемножения плотных матриц)

### 6.1 Основные понятия

Любая параллельная программа имеет свои характеристики, благодаря которым можно оценить эффективность распараллеливания. К таким характеристикам относятся:

- $f$  - доля последовательных операций в исходной программе ( $0 \leq f \leq 1$ ),
- $1 - f$  - доля распараллеленных операций в исходной программе ( $0 \leq f \leq 1$ ),
- $p$  - число процессоров (ядер, вычислительных узлов),
- $T_1$  - время работы программы на одном процессоре,
- $T_p$  - время работы программы на системе из  $p$  процессоров,
- $S = \frac{T_1}{T_p}$  - это ускорение работы программы при переходе с одного процессора на систему из  $p$  процессоров.
- $E = S/p$  - средняя доля времени выполнения параллельного алгоритма, в течение которого процессоры реально используются для решения задачи.
- $C = pT_p$  - стоимость вычислений
- $T_0 = pT_p - T_1$  - суммарные накладные расходы

Данные свойства формируют очень важный закон Амдала, который иллюстрирует ограничение роста производительности вычислительной системы с увеличением

количества вычислителей. Ограничение может быть математически представлено следующим образом:

$$S \leq \frac{1}{f + (1 - f)/p}$$

Следствием этого закона является факт того, что условные 10% последовательных операций гарантируют ускорение программы не более чем в 10 раз независимо от количества используемых процессоров. Кроме того, с увеличением количества процессоров растут суммарные накладные расходы, что значит эффективность распараллеливания в какой-то момент начинёт снижаться и использование большего количества ядер наносит вред производительности. Так же отметим, что если время выполнения алгоритма на одном процессоре будет фиксированным, то при увеличении числа процессоров эффективность распараллеливания как правило, уменьшается за счёт роста накладных расходов, но если же количество процессоров будет фиксированным, то эффективность распараллеливания можно увеличить, увеличивая сложность решаемой задачи.

Важной составляющей любого параллельного алгоритма является его масштабируемость - способность системы увеличивать свою производительность пропорционально дополнительным ресурсам. (обычно аппаратных) Данный показатель можно оценить через отношение прироста производительности системы к приросту используемых ей ресурсов. Чем ближе это отношение к единице, тем масштабируемость лучше. Кроме того, масштабируемость может быть вертикальной путём замены платформы на которой выполняется алгоритм на более производительный, так и горизонтальной за счёт добавления дополнительных программных и аппаратных средств.

## 6.2 Вычисление всех $t_{ij}$

Для нахождения функции цены с учетом и без времени поворота нужно вычислить время перехода от уровня  $i$  до  $i + 1$  с нулевой начальной скоростью. Так как водная гладь разбита на равномерную сетку, то всевозможные  $2m - 1$  путей перемещения между уровнями будут идентичными на каждом уровне, что создаёт ненужный объем вычислений. Следовательно, перед нахождением функции цены достаточно вычислить  $2m - 1$  путей для перехода между уровнями.

При последовательном вычислении требуется  $2m - 1$  независимых расчетов дифференциальных уравнений. Все они будут записаны в массив *gamma\_time* и очевидно что  $t_{il} \equiv \text{gamma\_time}(m - j + l)$ , где  $j$  - начальная точка, а  $l$  - точка куда корабль движется. В силу независимости данных не составит труда распараллелить нахождение массива *gamma\_time* создав любое число потоков. При вычислении одним потоком было затрачено 4.166885 секунд, в то время как 8 потоков справились за 1.363404 секунд. Прирост производительности составил приблизительно в 4 раза. Чем больше будут расстояния между точками, тем выше будет производительность.

### 6.3 Вычисление функции цены $V(i, j)$

Учитывая, что мы заранее просчитали всевозможные  $t_{ij}$ , есть возможность решить за кубическое время. Нужно провести  $m(m-1)(n-1)$  операций сравнения и сложения. Алгоритм нахождения функции цены имеет плохую масштабируемость в следствии жестких информационных связей между элементами матрицы  $V(i, j)$ . Единственная возможность распараллелить вычисления является использование барьеров для сохранения синхронизации. Вычисление функции цены начинается с  $n - 1$  уровня до 1 уровня поэлементно. Вычисление всех точек на уровне можно распределить между потоками и дать строгую команду не продолжать вычисление следующего уровня пока не будут вычислены все точки на текущем уровне. В случае малого количества точек данный метод может сильно проседать по производительности в сравнении с последовательным алгоритмом, так как большая часть вычислительных ресурсов будет уходить на распределение задач между ядрами процессора.

### 6.4 Вычисление оптимального пути для функции цены $V(i, j)$

В отличии от поиска функции цены поиск оптимального пути начинается со 2-го уровня по  $n$ -й уровень. Однако здесь так же присутствует жесткая связь между уровнями, что не позволяет распараллелить задачу на поиск оптимального пути в нескольких уровнях. Метод распараллеливания будет совпадать с методом вычисления функции цены: между потоками будет разделена работа поиска минимального элемента на уровне  $i$ , пока каждый поток на отдельном участке не найдет свой минимальный элемент будет стоять барьер на переход на следующий уровень. По аналогии с поиском функции цены при малом количестве точек или избыточном количестве потоков производительность незначительно отличаться от последовательного алгоритма.

Стоит отметить что в этом случае использование потоков может быть использована для более полезной цели: найти всевозможные оптимальные траектории судна. Мы уже знаем, что благодаря симметрии все эти пути будут эквивалентны по времени, однако это является правдой только в случае полного торможения судна перед поворотом. В любом случае очевидно что верхняя оценка количества всевозможных оптимальных путей равна  $2^n$ . При необходимости вычислить всевозможные пути использование параллельного программирования способно потенциально снизить время на расчеты пропорционально количеству используемых процессоров.

### 6.5 Вычисление функции цены $V(i, j, p)$

При учёте времени на поворот вычисления функции цены возрастает приблизительно в  $q * t$  раз, так как необходимо просчитать новое измерение функции цены и дважды найти минимальное значение функции для каждого элемента матрицы  $V(i, j, p)$ . По аналогии с двумерным аналогом жесткая связь матрицы сохраняется только между уровнями, что позволяет распараллелить расчет функции цены на каж-

дую точку уровня и поставить барьер до окончания обработки всех точек на текущем уровне.

## 6.6 Вычисление оптимального пути для функции цены $V(i, j, p)$

Распараллеливание построено по аналогии с функцией цены  $V(i, j)$ , с небольшим изменением того что распараллеливание идет в первую очередь по всем возможным секторам  $p$  и внутри каждого потока создается еще несколько потоков для вычисления минимального  $j$ .

## 7 Заключение

Таким образом:

- Составлена математическая модель движения парусного судна.
- Определен априорный класс управлений.
- Построен алгоритм вычисления оптимального синтеза управления движением для математической модели парусного судна.
- Написана компьютерная программа с применением параллельных методов, которая реализует построенный алгоритм перемещения и иллюстрирует движение парусного судна.

## Список литературы

- [1] А. В. Kurzhanski, P. Varaiya Dynamics and Control of Trajectory Tubes, Birkhäuser, 2014.
- [2] Куржанский А.Б. Управление и наблюдение в условиях неопределенности. М.: Наука, 1977.
- [3] Беллман Р. Динамическое программирование. М.: Изд-во иностр. лит., 1960, 400с.
- [4] Красовский Н.Н. Теория управления движением, М.:Наука, 1968
- [5] Лутманов С.В. Линеиные задачи оптимизации. Часть 2. Оптимальное управление линейными динамическими объектами. Учебное пособие, Пермь, 2005 .
- [6] Справочник по катерам, лодкам и моторам, под общей редакцией Новака Г. М., Ленинград, «Судостроение», 1982 .
- [7] Движение под парусом со скоростями быстрее ветра, «Катера и Яхты», 14, 1968.

- [8] On Some Nonstandard Dynamic Programming Problems of Control Theory, A. B. Kurzban, P. Varaiya Variational Methods and Applications /Eds. Giannessi F., Maugeri A., N. Y.: Kluwer, 2004. P. 613-627
- [9] Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. М.: Наука, 1976 .
- [10] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002. - 178 с.
- [11] Антонов А.С. Технологии параллельного программирования MPI и OpenMP: Учеб. пособие. Предисл.: В.А.Садовничий. – М.: Издательство Московского университета, 2012.