

Санкт-Петербургский государственный университет

Математико-механический факультет

Математическое обеспечение и администрирование
информационных систем

Информационные системы и базы данных

Леонова Анна Васильевна

Оптимизация распределения сенсоров между
источниками сигналов

Выпускная квалификационная работа

Научный руководитель:

д. ф.-м. н., профессор Граничин О.Н.

Рецензент:

к. ф.-м. н., старший научный сотрудник

Иванский Ю.В.

Санкт-Петербург
2018

SAINT PETERSBURG STATE UNIVERSITY

Mathematics&Mechanics Faculty

Software and Administration of Information Systems

Information System and Data Bases

Leonova Anna

Optimization of Sensors Distribution among
the Signals' Sources

Graduation Project

Scientific supervisor:
Ph.D., Professor Oleg Granichin

Reviewer:
senior researcher Yury Ivanskiy

Saint Petersburg
2018

Содержание

Введение	4
Цель работы	6
1 Глава: Математическая модель	7
1.1 Постановка математической задачи	7
1.1.1 Модель наблюдения	7
1.1.2 Доверительные эллипсоиды	8
1.1.3 Функционал качества	9
1.2 Алгоритмы оптимизации	10
1.2.1 Метод Монте-Карло для расчета пересечения доверительных эллипсоидов	10
1.2.2 Полный перебор (Brute Force)	11
1.2.3 Алгоритм, основанный на решении LMI	11
2 Глава: Прототип информационно–аналитической системы для автоматизации процесса распределения целей между сенсорами	13
2.1 Общее описание	13
2.2 Функциональность	14
2.3 Структура базы данных	16
2.4 Визуализация	17
2.5 Реализация модуля решения задач LMI	17
3 Глава: Экспериментальные результаты	19
3.1 Анализ экспериментов	19
3.2 Нагрузочное тестирование	21
Заключение	22

Введение

Повышенный интерес привлекают сенсорные сети вследствие существенного прогресса интегрированных систем, средств беспроводной коммуникации и компонентов сенсорики в различных областях, в частности, при мониторинге и слежении за движением объектов. Практические применения эти задачи находят в таких приложениях, как интеллектуальное видеонаблюдение в местах большого скопления людей [1], отслеживание передвижений животных [2], управление воздушным движением [3], наблюдение за космическим пространством и оповещение о ситуации в нем [4].

Сенсорные сети могут использоваться для уменьшения неопределенностей в системе, связанных с неточностями измерительных приборов и воздействиями внешней среды, за счет комплексирования данных от разных источников. Более точное измеряемое значение можно получить при помощи совместного учета данных группы сенсоров. Однако при использовании большого количества датчиков возникают проблемы ограниченности ресурсов по пропускной способности, вычислительным возможностям и энергозатратности. При увеличении количества датчиков и объектов наблюдения возникает потребность в сокращении числа «активных» датчиков для уменьшения затрат на поддержание работоспособности сети. Таким образом, возникает задача распределения объектов слежения между сенсорами.

Создание и построение сложных адаптивных систем, использующих мультиагентные технологии, приобретают популярность и получают значительное развитие в последнее время. «Классическим подходом» всегда было выделение так называемого центрального узла. Сейчас при проектировании управляющих систем сложными объектами ориентируются на формирование большого множества групп независимых друг от друга агентов. В разных прикладных задачах демонстрируется достаточность гипотезы существования только локального «общения» агентов, то есть предположения, что каждый агент имеет взаимную связь с некоторым подмножеством агентов группы (набор соседей), а не взаимодействует со всеми. Посредством коллективных усилий агентами достигается их общая единая задача. Учитывая вышеизложенное, агенты действуют одинаково в некоторых крупных множествах при достаточно общих предположениях. При этом становится осуществимым замещение изначальной задачи в пространстве большой размерности на совокупность простых однотипных задач с одной целью, которая значительно проще, в пространстве размерности пропорциональной количеству таких крупных множеств, упомянутых ранее. В работах [5], [6], [7], [8], [9] подробно описывались такие системы.

Построение взаимодействия групп наблюдателей охватывает широкий пласт задач. Разделение заданий между участниками группы является важной составляющей традиционных задач робототехники. При решении сложных задач, декомпозиция которых осуществима, на практике наиболее рационально применение групп слежения. Таким образом, увеличение производительности системы, которая решает единую задачу, получается за счет оптимизации распределения на подгруппы слежения, которым назначены конкретные подзадачи. При этом сложность таких задач оптимизации экспоненциально растет в зависимости от количества подзаданий и мощности групп наблюдателей, поскольку в общем виде эти проблемы оптимизации принадлежат семейству трудоемких, переборных задач.

При решении задачи слежения одним из наиболее популярных алгоритмов является фильтр Калмана [10] [11], использующийся для наблюдения за разным количеством целей. Однако с увеличением количества наблюдаемых объектов возникают проблемы, вызванные большими коммуникационными и вычислительными нагрузками. Существуют некоторые решения этой проблемы [12], [13] на основе распределенной схемы фильтрации Калмана. Другие проблемы сенсорных сетей рассматривались в статьях [14], [15], [16], [17], в которых предлагались консенсусные алгоритмы. В статьях [15] и [14] исследовали гетерогенность и ограниченность пропускной способности сенсорных сетей соответственно и распределенные консенсусные алгоритмы, использующие модифицированную фильтрацию Калмана. А в статьях [16], [17] демонстрируются стратегии работы сенсорных сетей с асинхронной передачей сообщений и ограниченной энергозатратностью.

Всё это актуализирует задачу построения алгоритмов разбиения сенсоров на группы, желательно достаточно малые, для слежения за разными объектами. В работе [18] задача была достаточно строго математически сформулирована, и авторами статьи было предложено решение на основе использования метода линейных матричных неравенств (LMI), который может показывать лучшую эффективность по сравнению с методом полного перебора. Вышесказанное актуализирует цель выполненной работы. Похожая задача рассматривалась и в работе [19].

Цель работы

Целью работы является разработка информационно–аналитической системы для автоматизации процесса распределения целей между наблюдателями. Для достижения этой цели были сформулированы следующие задачи:

1. Разработка прототипа информационно–аналитической системы с возможностями визуализации задачи, настройки различных параметров.
2. Разработка формата хранения данных и структуры базы данных.
3. Сравнение алгоритмов оптимизации распределения объектов наблюдения: полного перебора (Brute Force) и основанного на LMI — посредством анализа результатов моделирования и нагрузочного тестирования.

1 Глава: Математическая модель

1.1 Постановка математической задачи

Рассмотрим математическую модель из [18], содержащую распределенную сеть из n сенсоров, расположенных в интересующей нас области, и m целей. Необходимо распределить цели между сенсорами таким образом, чтобы достаточно точно оценить траекторию движения целей группой наблюдателей. При распределении объектов наблюдения между датчиками (сенсорами) задача оценивания в общем виде получается избыточной. Можно учитывать показания только с тех сенсоров, которые дают наиболее существенный вклад в точность оценивания траектории цели, так как можно не учитывать данные с тех датчиков, которые дублируют информацию (такая ситуация возможна, например, при близком расположении сенсоров друг к другу). Таким образом можно понизить коммуникационные и вычислительные затраты. Следовательно, предлагается рассматривать задачу, используя формирования подгрупп слежения.

1.1.1 Модель наблюдения

Рассматривается множество $M = \{1, 2, \dots, m\}$ — набор целей с векторами состояния $\{x_t^i\}_{t=0,1,2,\dots}$, $x_t^i \in R^p$, $i \in M$ — траектория цели i , изменяющимися согласно следующему уравнению:

$$x_{t+1}^i = f^i(x_t^i) + w_t^i, \quad (1)$$

где f^i — некоторая функция динамики, $\{w_t^i\}$ — белый гауссовский шум с нулевым математическим ожиданием и ковариационной матрицей R_w^i : $Ew_t^i = 0$, $Ew_t^i(w_t^i)^T = R_w^i \leq \sigma_w^2 I_k$, для цели i . Здесь использованы следующие обозначения: E — символ математического ожидания, I — единичная матрица размерности $k \times k$, у которой на диагонали единицы, а остальные элементы нули, \leq — нестрогое неравенство для симметричных матриц, понимаемое в смысле неравенств для квадратичных форм.

Пусть $N = \{1, 2, \dots, n\}$ — набор датчиков, а $s_t^j \in R^d$ — состояние датчика j . В зоне видимости сенсоров находится вышеупомянутые m целей. Каждый датчик j предоставляет измерения $y_t^{i,j} \in R^d$, $i \in M$, $j \in N$, доступные этому сенсору j в момент времени t , соответствующие наблюдаемой траектории цели i и задающиеся согласно следующему уравнению:

$$y_t^{i,j} = \varphi^j(s_t^j, x_t^i) + v_t^{i,j}, \quad (2)$$

где φ^i — некоторая нелинейная дифференцируемая функция, $v_t^{i,j}$ — белый гауссовский шум с нулевым математическим ожиданием и ковариационной матрицей $Q_t^{i,j}$: $E v_t^{i,j} = 0$, $E v_t^{i,j}(v_t^{i,j})^T = Q_t^{i,j}$. Сенсоры могут хранить у себя в памяти, в том числе, и предшествующие l наблюдений

в моменты времени $t - 1, t - 2, \dots, t - l$.

Пусть $\mathbf{x}_t^i = [x_t^{i,1} x_t^{i,2}]^\top$ расположение наблюдаемых объектов и $\mathbf{s}_t^j = [s_t^{j,1} s_t^{j,2}]^\top$ расположение сенсоров на плоскости. При измерениях углов и дальностей до целей можно рассматривать функцию $\varphi(\cdot, \cdot)$ в виде двухмерного вектора, зависящего от векторов $\mathbf{s}_t^j, \mathbf{x}_t^i$ следующего вида:

$$\varphi(\mathbf{s}_t^j, \mathbf{x}_t^i) = \begin{bmatrix} \psi(\mathbf{s}_t^j, \mathbf{x}_t^i) \\ \rho(\mathbf{s}_t^j, \mathbf{x}_t^i) \end{bmatrix} \in \mathbb{R}^2,$$

где

$$\psi(\mathbf{s}_t^j, \mathbf{x}_t^i) = \operatorname{arctg} \left[\frac{x_t^{i,1} - s_t^{j,1}}{x_t^{i,2} - s_t^{j,2}} \right]$$

— угол азимута — угол между направлением на наблюдаемый объект и направлением на север;

$$\rho(\mathbf{s}_t^j, \mathbf{x}_t^i) = \sqrt{(x_t^{i,1} - s_t^{j,1})^2 + (x_t^{i,2} - s_t^{j,2})^2}$$

— расстояние от местоположения сенсора до объекта.

С каждой точкой наблюдения ассоциируется доверительный эллипсоид, отражающий ошибки по полученным измерениям. Более подробно доверительные эллипсоиды рассматриваются в следующем разделе.

1.1.2 Доверительные эллипсоиды

Рассмотрим наблюдение $y_t^{i,j}$, полученное с датчика j и соответствующее траектории цели i . Допустим, что обратная функция по второму аргументу $\varphi^{-1}(s_t^i, \cdot)$ существует и имеет вид

$$\varphi^{-1}(s_t^i, y_t^{i,j}) = s_t^j + \begin{bmatrix} y_t^{i,j,2} \sin y_t^{i,j,1} \\ y_t^{i,j,2} \cos y_t^{i,j,1} \end{bmatrix} = x_t^i + \xi_t^{i,j} = \eta^{i,j},$$

где $y_t^{i,j,1}$ и $y_t^{i,j,2}$ — первая и вторая координата вектора $y_t^{i,j}$, $\xi_t^{i,j}$ — независимые с нулевым средним $E\xi_t^{i,j} = 0$ и ковариацией $E\xi_t^{i,j}(\xi_t^{i,j})^\top = \Xi_t^{i,j}$.

Если матрица ковариации ошибок $v_t^{i,j}$ равна $Q^{i,j} = \begin{bmatrix} \sigma_\psi^2 & 0 \\ 0 & (y_t^{i,j,2} \sigma_\rho)^2 \end{bmatrix}$, то для ошибок $\xi_t^{i,j}$ матрица ковариации имеет вид

$$\Xi_t^{i,j} = R(y_t^{i,j,1}) \begin{bmatrix} (y_t^{i,j,2} \sigma_\psi)^2 & 0 \\ 0 & (y_t^{i,j,2} \sigma_\rho)^2 \end{bmatrix} R(y_t^{i,j,1})^\top,$$

где $R(\psi) = \begin{bmatrix} \sin \psi & -\cos \psi \\ \cos \psi & \sin \psi \end{bmatrix}$ — матрица поворота на угол ψ . Тогда при задании некоторого уровня достоверности $p \in [0, 1]$ доверительный эллипсоид, содержащий истинное значение параметра с вероятностью $1 - p$,

вокруг точки результата наблюдения $\eta^{i,j}$ сенсора j цели i , задается формулой:

$$\mathcal{E}^{i,j} = \{x^i : (x^i - \eta^{i,j})^\top (\Xi_t^{i,j})^{-1} (x^i - \eta^{i,j}) \leq \chi_{p,d}^2\}, \quad (3)$$

где $\chi_{p,d}^2$ — p -квантиль распределения χ^2 с d степенями свободы.

Любой эллипсоид можно задать следующими эквивалентными формулами:

$$\mathcal{E} = \{x \in \mathbb{R}^\nu : x = Rq + x_c, q \in \mathbb{R}^r, \|q\| < 1\}, \quad (4)$$

— как образ единичного шара при аффинном отображении, где $R \in \mathbb{R}^{\nu \times r}$ — прямоугольная матрица, x_c — центр эллипсоида;

$$\mathcal{E} = \{x \in \mathbb{R}^\nu : (x - x_c)^\top P^{-1} (x - x_c) \leq 1\}, P = RR^\top. \quad (5)$$

— как образ единичного шара при аффинном отображении, где матрица R максимального строчного ранга.

$$\mathcal{E} = \{x \in \mathbb{R}^\nu : (x - x_c)^\top P^{-1} (x - x_c) \leq 1\}, P = RR^\top \quad (6)$$

— через выпуклые квадратичные функции, где $A \in \mathbb{R}^{\nu \times \nu}$, $A \in \mathbb{R}^\nu$, $c \in \mathbb{R}$

1.1.3 Функционал качества

Рассмотрим два доверительных эллипсоида \mathcal{E}^{i,j_1^i} и \mathcal{E}^{i,j_2^i} . Вероятность того, что точка измерения лежит в пересечении $U_i^{\{j_1^i, j_2^i\}} = \mathcal{E}^{i,j_1^i} \cap \mathcal{E}^{i,j_2^i}$ равна $1 - 3p$ [19]. Допустим, что эллипсоидов не два, а значительно больше, тогда такое множество пересечения строить довольно сложно.

Обозначим $U_i^{\{j_1^i, \dots, j_{k_i}^i\}}$ — пересечение доверительных эллипсоидов, S — множество всех наборов $\{j_1^i, \dots, j_{k_i}^i\}$ при $i = 1, \dots, m$.

Для того, чтобы достаточно точно оценить траекторию движения цели i , целесообразно минимизировать объем пересечения $U_i^{\{j_1^i, \dots, j_{k_i}^i\}}$, но при этом учитывать ограничения на количество сенсоров, состоящих в группе слежения. В статье [18] обоснована задача минимизации:

$$\Phi(S) = \sum_{i \in M} (Vol(U_i^{\{j_1^i, \dots, j_{k_i}^i\}}) + \alpha |\{j_1^i, \dots, j_{k_i}^i\}|) \rightarrow \min, \quad (7)$$

где α — «стоимость» использования одного сенсора при наблюдении одной цели i , $|\cdot|$ — мощность множества (количество сенсоров, используемых при наблюдении за целью i ; то есть k_i).

Введем матрицу распределения ресурсов $G_t = [g_t^{i,j}]$, элементы которой принимают значения больше нуля, если сенсор j состоит в группе

слежения за целью i ; в противном случае – равны нулю.

Для решения подобных проблем предлагается воспользоваться «выпуклением» задачи, основанном на использовании специальных матричных норм, и получить разреженное субоптимальное решение. Математическая задача в этом случае сводится к минимизации суммы ненулевых компонент вектора, определяемого l_1 -нормой, минимум которой будет являться хорошей аппроксимацией разреженного решения. Тогда функционал качества примет вид:

$$\bar{\Phi}(G_t) = \sum_{i \in M} Vol(U_i^{\{j_1^i, \dots, j_{k_i}^i\}}) + \alpha \sum_{i \in M} \|G_t^{i \cdot}\|_1 \rightarrow \min_{G_t}, \quad (8)$$

где $\|\cdot\|_1$ – l_1 -«норма»: $\|x\|_1 = \sum_{i=1}^{\nu} |x_i|$, $G_t^{i \cdot}$ – множество сенсоров, оценивающих траекторию объекта i .

1.2 Алгоритмы оптимизации

1.2.1 Метод Монте-Карло для расчета пересечения доверительных эллипсоидов

Рандомизированные подходы стали широко использоваться с появлением метода статистического моделирования Монте-Карло (МК). Этот метод был предложен Метрополисом и Уламом [20] во время их работы над Манхэттенским проектом в Лос-Аламосе с фон Нейманом и Теллером. Метод МК предлагает простую схему для оценки математического ожидания функции, основанной на выборках ее значений для некоторых случайных аргументов (рандомизация). Первоначально рандомизированный подход использовался для аппроксимации многомерного интегрирования в транспортных уравнениях, возникающих в связи с проблемой понятия нейтрона в изотропной среде. Точнее, если следующую формулу сложно интегрировать аналитически

$$F = \int \dots \int_W f(w) dw, W \subset R^r,$$

то метод МК проще в практическом применении [21].

Пусть множество W является параллелепипедом с объемом $Vol(W)$, а функция f ограничена: $0 \leq f(w) \leq f_{max}$. Геометрический смысл интеграла представляет собой объем под графиком функции f .

Метод МК для этой задачи заключается в следующем:

1. Фиксируется положительное целое число T .
2. Выбирается из параллелепипедов $W \times [0, f_{max}] \subset R^{r+1}$ T выборок независимых одинаково распределенных случайных величин z_1, z_2, \dots, z_T .

3. Среди z_1, z_2, \dots, z_T подсчитывается число S тех z_i , последняя компонента которых не превосходит значения функции f в соответствующей точке, определяемой первыми r -координатами.
4. Вычисляется оценка

$$\hat{F} = \frac{S}{T} \text{Vol}(W) f_{max}$$

Для реализации метода находился ограничительный квадрат, содержащий пересечения эллипсоидов, в пространстве посредством поиска наименьшей большей полуоси эллипсоидов, которые участвовали в пересечении. Число «бросков» бралось равным 25 на 1×1 условных единиц (в терминах нашей системы 1 km^2).

1.2.2 Полный перебор (Brute Force)

Алгоритм полного перебора или brute force относится к классу методов поиска решения исчерпыванием всевозможных вариантов. Таким образом, проверяются все варианты распределения объектов между сенсорами с целью проанализировать эффективность тех или иных возможных распределений, а также найти экстремальные значения точности обнаружения целей. С практической точки зрения, алгоритм очень времязатратный и ресурсозатратный, но дает оптимальное решение. В дальнейшем он используется для сравнения результатов с другим более эффективным методом.

Для реализации этого алгоритма Brute Force последовательно вычисляются значения для всевозможных подмножеств множества сенсоров для каждой цели. После этого подсчитываются значения суммы объемов и значения функционала, определенного формулой (7), для всех перестановок получившихся значений на предыдущем этапе.

1.2.3 Алгоритм, основанный на решении LMI

Получающиеся множества доверительных эллипсоидов достаточно трудно хранить в памяти (ресурсозатратно) и подсчитывать объем. Типичным подходом в таких задачах является упрощение на основе построения аппроксимирующего множества $U_i^{j_1^i, \dots, j_{k_i}^i}$ эллипсоида $\hat{\epsilon}_i$. Будем считать, что множество пересечения U^{j_1, \dots, j_k} цели i возможно аппроксимировать эллипсоидом.

В теории робастного управления исследуются системы с неопределенностью. При рассмотрении задачи о подавлении органических внешних возмущений [22], [23] возникла l_1 оптимизация. Этот математический инструмент в совокупности с линейными матричными неравенствами

(LMI), появившимися впервые в таких работах, как [24], [25], оказался особо удобным в применении для ряда задач. В дальнейших исследованиях в этом направлении было установлено, что эти неравенства являются общим способом анализа и сведения в единое целое данных, добытых анализом, линейных систем [26]. С возникновением эффективного программного обеспечения, решающего LMI, этот инструмент стал вычислительно эффективным, что привело к более активному его использованию. В следующих работах [27, 28, 29, 30, 31, 32] рассматривалась и развивалась данная тема.

Алгоритм решает следующую задачу полуопределенного программирования [18]:

$$\text{minimize } \delta \quad (9)$$

при LMI-ограничениях

$$\begin{aligned} \forall i \in [1, m] \quad & g_t^{i,1} \geq 0, \dots, g_t^{i,n} \geq 0, \\ \forall i \in [1, m] \quad & \hat{A}_t^i > 0, \\ \forall i \in [1, m] \quad & \begin{bmatrix} \hat{A}_t^i & \hat{b}_t^i & 0 \\ (\hat{b}_t^i)^T & -1 & (\hat{b}_t^i)^T \\ 0 & \hat{b}_t^i & -\hat{A}_t^i \end{bmatrix} - \sum_{j=1}^n g_t^{i,j} \begin{bmatrix} A_t^{i,j} & b_t^{i,j} & 0 \\ (b_t^{i,j})^T & c_t^{i,j} & 0 \\ 0 & 0 & 0 \end{bmatrix} \leq 0, \\ & \sum_{i \in M} \log \det(\hat{A}_t^i)^{-1} + \alpha \sum_{i \in M} \|G_t^{i,\cdot}\|_1 \leq \delta. \end{aligned}$$

Известные составляющие задачи – это $A_t^{i,j}$, $b_t^{i,j}$, $c_t^{i,j}$, задающие по формуле (6) доверительные эллипсоиды, полученные по измерениям цели i с сенсора j ; α – «стоимость» включения сенсора в группу слежения.

Результатом решения задачи является матрица ресурсов G_t , которая отвечает за распределения целей между сенсорами; \hat{A}_t^i , \hat{b}_t^i , определяющие эллипсоиды пересечения для цели i формулой (6) (\hat{c}_t^i находятся по \hat{A}_t^i , \hat{b}_t^i).

Трактовать данную задачу можно таким образом – хотим минимизировать объемы эллипсоидов пересечения и количество участников в группах слежения.

2 Глава: Прототип информационно–аналитической системы для автоматизации процесса распределения целей между сенсорами

2.1 Общее описание

В ходе выполнения выпускной квалификационной работы был разработан прототип информационно–аналитической системы для автоматизации процесса распределения целей между сенсорами на языке Python, позволяющий облегчить работу пользователя по решению задачи и сравнению алгоритмов оптимизации распределения целей между сенсорами, а также позволяющая визуализировать состояния датчиков, наблюдаемых объектов и доверительные окрестности наблюдений сенсоров. Ниже представлена UML-диаграмма компонентов, созданная при помощи сервиса creately.com[33].

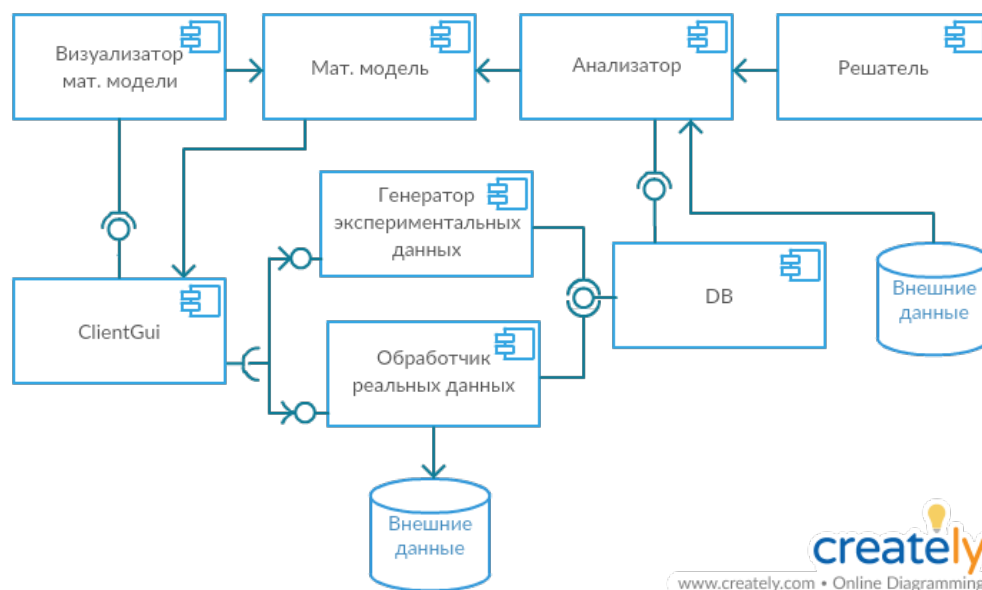


Рис. 1: UML-диаграмма компонентов.

Компонента «ClientGui» отвечает за взаимодействие с пользователем и предоставляет графический интерфейс системы, позволяющий пользователю задавать такие персональные настройки, как режим входных данных, выбор алгоритма решения задачи и его настройки, настройки визуализации.

Компонента «Визуализатор мат. модели» позволяет отображать математическую модель задачи, связанную с перемещением наблюдателей и целей, выполнением сенсорами наблюдений за объектами, траекторией движения объектов в режиме симуляции, траекторией (полученной сенсорами при наблюдении), вычислением доверительных эллипсоидов, эллипсоидами пересечения. Эта компонента служит для удобства восприятия задачи пользователем.

Компоненты «Мат.модель» и «Анализатор» отвечают за всю логику математической задачи: инициализацию агентов слежения и целей по предоставляемым данным, их перемещение, внесение рандомизации, слежение за объектами, вычисление доверительных эллипсов, вычисление распределения объектов, создание отчета результата.

Компонента «Обработчик реальных данных» отвечает за преобразование данных, предоставляемых системе, в формат требуемый системой.

Компонента «Генератор экспериментальных данных» отвечает за режим симуляции, а именно: за создание заданного количества агентов–наблюдателей и агентов–целей, за определение их модели поведения.

Компонента «Решатель» представляет собой алгоритмы решения задачи оптимизации.

Компонента «DB» отвечает за взаимодействие с хранилищем данных, в котором содержатся информация о сенсорах и целях, наблюдения, промежуточные данные и результаты работы алгоритмов.

Выбор языка обусловлен языка обуславливается поддержкой пакета, решающего задачи полуопределенного программирования, который требуется для реализации алгоритмов.

2.2 Функциональность

На рисунке 2 представлен графический интерфейс программы, демонстрирующий возможности настройки системы, располагающиеся на панели управления. Программа позволяет генерировать данные для модели наблюдения (режим симуляции) или задавать их через внешний источник информации. Выбор режима представлен в верхней части панели управления. Также возможно задания других параметров системы, влияющих на модель наблюдения – ошибок по измерениям углов и дальностей сенсоров в градусах и процентах соответственно, выбора режима работы системы – выбор алгоритма, режима сравнения, режима работы последовательно по каждому эксперименту.

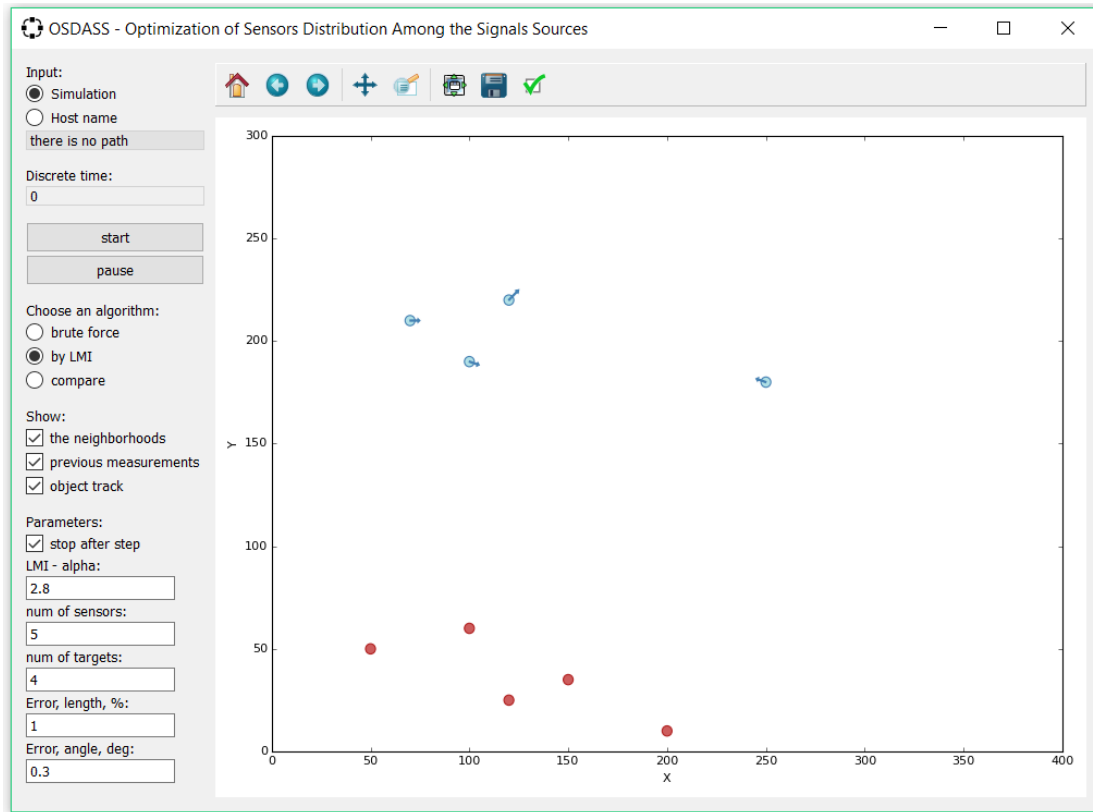


Рис. 2: Графический интерфейс системы.

Также для удобства пользователя есть средства навигации, расположенные на горизонтальной панели управления над элементом отображения визуализации задачи, для работы с ним (элементом отображения визуализации), позволяющие сохранять через диалоговое окно в задаваемой пользователем директории рисунок в задаваемом расширении, перемещать, изменять масштабы.

Результат работы алгоритмов предоставляется в отдельном окне и демонстрирует пользователю в виде отчета такие данные, как оптимальное распределение наборов сенсоров по целям в виде таблицы, дату и время, название алгоритма, значение суммы объемов, значение функционала качества относительно уровня достоверности (более подробно это значение описано в следующей главе). А также формируется отчет в виде текстового файла. Иллюстрированный пример результата работы алгоритма представлен на рисунке 3.

tracking group	
target 0	0, 1, 2, 3, 4
target 1	0, 2, 4
target 2	0, 1, 2, 3, 4
target 3	0, 1, 3, 4

Рис. 3: Пример результата работы системы.

2.3 Структура базы данных

В качестве хранилища сведений о сенсорах и наблюдаемых объектах, наблюдений, промежуточных данных и результатов работы алгоритмов оптимизации распределения целей была выбрана реляционная база данных MariaDB [34]. MariaDB с открытым исходным кодом и имеет поддержку интеграции в Python, что достаточно для реализуемого хранилища данных. Структура базы данных представлена на рисунке 4, выполненной с помощью сервиса creately.com[33].

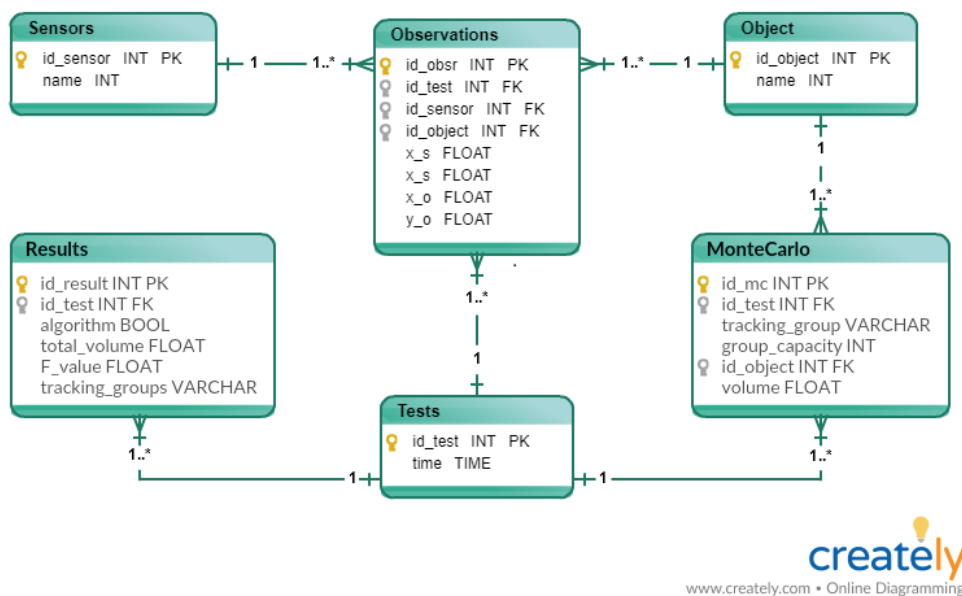


Рис. 4: Диаграмма базы данных.

В базе данных присутствует хранение истории «экспериментов», за-

дающихся атрибутом «время». Также происходит формирование отчета по результатам, содержащимся в хранилище, в виде текстового файла.

Архитектура была выбрана клиент–серверная.

2.4 Визуализация

Для рассмотрения берется «поле» размером $300 \times 400 \text{ km}^2$. В системе реализована навигация, позволяющая перемещать «поле», менять его масштаб посредством диалогового окна, кнопки приближения, кнопки перемещения или кнопкам возвращения к предыдущему масштабу и последующему, а также кнопки вернуть значение масштаба по умолчанию. Присутствует возможность сохранения рисунков «поля» с помощью диалогового окна выбора директории и атрибутов файла (наименование и расширение).

Система реализует настройку параметров «Show»: показать доверительные эллипсоиды, показать предыдущие наблюдения, показать траекторию цели для режима симуляции. Работа этих режимов продемонстрирована на рисунках 5.

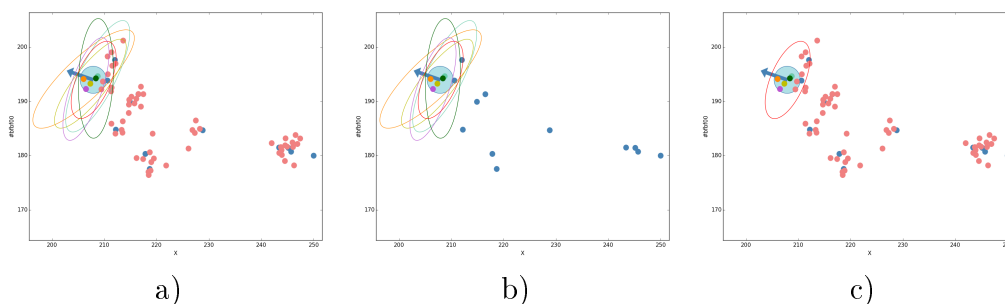


Рис. 5: Примеры работы параметров визуализации: а) все параметры выбраны, б) отключен показ истории наблюдений, в) отключен показ доверительных эллипсоидов.

2.5 Реализация модуля решения задач LMI

Для реализации данного модуля использовался пакет «cvxру» [35] языка программирования Python, как надстройка над другими решателями полуопределенного программирования такими, как например, «cvxopt», «mosek», «scs». В ходе разработки были попытки реализации системы на matlab, но из-за ряда недостатков этого языка программирования, в частности, в дальнейших перспективах невозможность создания интегрируемой библиотеки, использующей пакет «cvx», политики лицензирования matlab и др. – был выбран другой язык программирования.

При работе возникали трудности, связанные с решением проблем (например, сингулярность некоторой вспомогательной матрицы), для которых потребовалась настройка дополнительных параметров: увеличения количества шагов итерационного уточнения после решения системы ККТ и максимального количества итераций.

Для алгоритма, решающего LMI, было задано следующее значение параметра $\alpha = 1$. Составляющей результата работы алгоритма является матрица ресурсов G_t , и ниже представлен пример значения этой матрицы:

$$G_t = \begin{bmatrix} 4.39e - 01 & 6.53e - 01 & 2.94e - 07 & 8.92e - 07 & 5.09e - 07 \\ 3.56e - 01 & 7.31e - 01 & 3.29e - 07 & 4.08e - 06 & 8.95e - 07 \\ 2.51e - 01 & 5.21e - 01 & 3.71e - 07 & 1.51e - 01 & 1.71e - 01 \\ 4.72e - 07 & 5.57e - 01 & 7.21e - 07 & 1.24e - 06 & 5.63e - 01 \end{bmatrix},$$

где элемент $g_t^{i,j}$ соответствует вхождению сенсора j в группу слежения за целью i .

На практике элементы матрицы G_t отображают «вес» вклада сенсора в точность оценивания траектории и «абсолютных» нулей нет. Поэтому был введен некий порог «существенного» вклада. Этот порог в работе брался равным 0.001, ниже которого считается, что сенсор не участвует в наблюдении.

3 Глава: Экспериментальные результаты

3.1 Анализ экспериментов

Были проведены следующие эксперименты для $n = 5$ и $m = 4$, изображенного на рисунке 6.

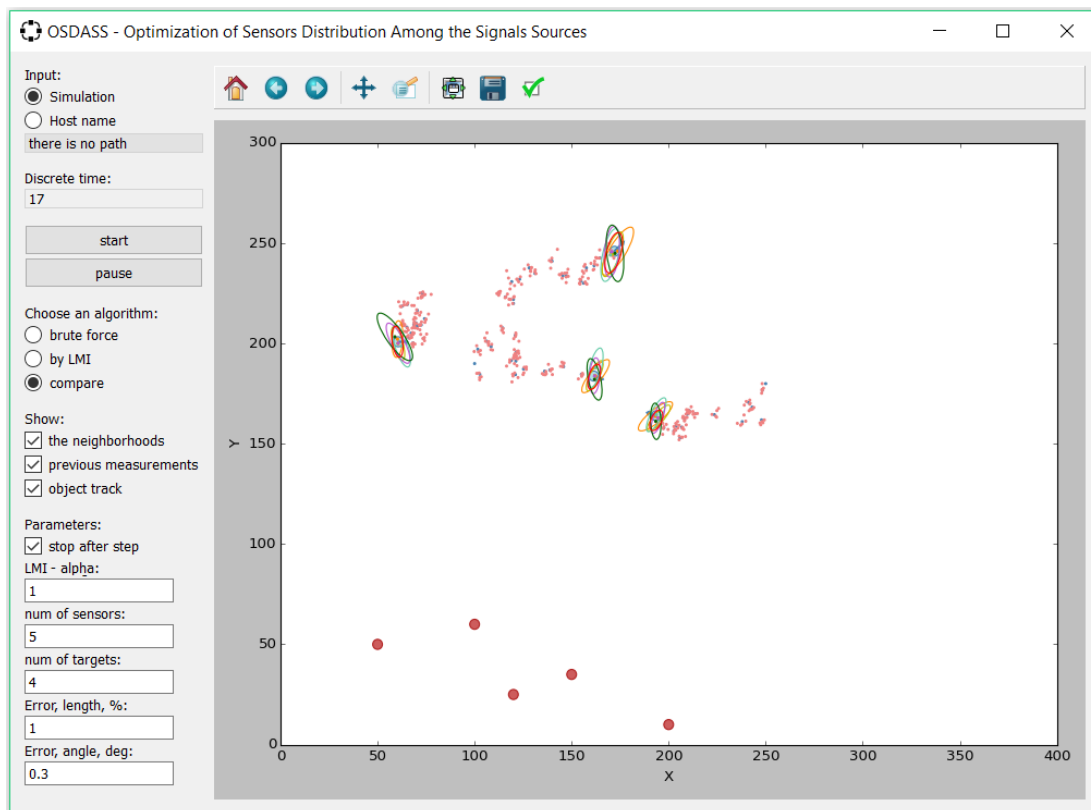


Рис. 6: Иллюстрация эксперимента.

Для получения результатов были подсчитаны площади реальных пересечений методом Монте-Карло для сравниваемых алгоритмов. Количество бросков бралось равным 25 на 1×1 условных единиц (в терминах нашей системы 1 km^2). Результат полного перебора представлен на рисунке 7.

Результат алгоритма, основанного на решении LMI, изображен на рисунке 8.

Сравнение результатов на рисунке 7 и 8 показывает, что метод перебора безусловно дает лучший ответ — сумма объемов пересечения равна

	tracking group
target 0	0, 1, 2, 3, 4
target 1	0, 2, 4
target 2	0, 1, 2, 3, 4
target 3	0, 1, 3, 4

Рис. 7: Результат Brute Force для рассматриваемого эксперимента.

	tracking group
target 0	0, 1
target 1	1, 4
target 2	1, 4
target 3	1, 4

Рис. 8: Результат алгоритма, основанного на LMI, для эксперимента.

163.3, а метод с LMI – на четверть хуже (этот показатель равен 217.7). Но видно, что количество сенсоров, которое используется для наблюдения не превышает двух для алгоритма на основе LMI. Стоит заметить, что в сравнении с наборами групп слежения результата полного перебора, которые (группы) состоят из максимального количества датчиков или почти максимального, группы наблюдения, полученные алгоритмом с использованием LMI, более привлекательны в смысле затрат на коммуникацию. Более того, при пересечении эллипсоидов вероятность принадлежности истинного значения пересечению становится меньше, чем у исходных доверительных эллипсоидов. При пересечении большего количества доверительных эллипсоидов кажется, что картина становится лучше (площадь области меньше). Но на самом деле сама область становится менее значимой с точки зрения доверительной вероятности, то есть уровень доверия становится ниже. Поэтому целесообразно сравнивать не показатели объемов, а показатели объемов, деленные на вероятность принадлежности истинного значения пересечению. Таким образом, значение объемов, нормированных на уровень доверия, существенно ниже

у метода на основе решения LMI, что дает значимый выигрыш по этому показателю.

В функционале (7) учитывается ограничение на сенсоры, количество которых мы хотим минимизировать, поэтому результаты демонстрируют выделение крайне расположенных сенсоров. В статье [19] используется слагаемое, ограничивающее нагрузку на один сенсор (то есть пытается минимизировать количество объектов, за которыми наблюдает сенсор), и тогда результаты получаются более сбалансированные.

3.2 Нагрузочное тестирование

Нижеследующая таблица представляет сравнение алгоритмов по времени работы на компьютере с параметрами: ОЗУ 8 ГБ, процессор Intel Core i7-7700HQ тактовой частотой 2.8 GHz процессора.

Таблица 1: Результаты нагрузочного тестирования

Серия	Кол-во сенсоров	Кол-во целей	Brute Force, с	by LMI, с
1	3	2	0.7062	0.4418
2	5	4	7.4825	1.1645
3	8	8	119.704	3.0956
4	16	16	>1200	18.2321

Для реализации алгоритмов была проведена серия «экспериментов». Каждая серия проводилась следующим образом. Было взято определенное количество сенсоров и целей, и вычислялись значения для серии «экспериментов». Для каждого теста по наблюдаемым данным было посчитано мат.ожидание по следующей формуле:

$$E = \frac{1}{h} * \sum_{i=1}^h X_i,$$

где E — мат. ожидание, X_i — времена, полученные при проведении «экспериментов».

Результаты сравнения показали несоизмеримые значения для алгоритма полного перебора и метода на основе LMI с увеличением значений n и m (количества сенсоров и целей).

Заключение

В ходе выполнения выпускной квалификационной работы были получены следующие результаты:

1. Реализован прототип системы для автоматизации процесса распределения целей между наблюдателями на языке Python, с возможностями графической интерпретации задачи и задания параметров как модели наблюдения, так и работы системы.
2. Создана структура хранения данных с использованием MariaDB базы данных, позволяющая хранить историю экспериментов.
3. Были реализованы алгоритмы оптимизации распределения целей между наблюдателями: Brute Force и алгоритм на основе решения LMI. С помощью реализованной системы был проведен анализ работы алгоритмов посредством сравнения результатов и нагрузочного тестирования. Сравнение показало лучшую эффективность решения на основе использования метода линейных матричных неравенств по сравнению с методом полного перебора.

Список литературы

- [1] Hanif A., Mansoor A. B., Imran A. S. Deep multi-view correspondence for identity-aware multi-target tracking / IEEE. In Proc. of 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017. P. 1–8.
- [2] Stochastic modeling and control for tracking the periodic movement of marine animals via AUVs / K. D. Smith, S.-C. Hsiung, C. White et al. / IEEE. In Proc. of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016. P. 3101–3107.
- [3] Thite A., Mishra A. Optimized multi-sensor multi-target tracking algorithm for air surveillance system / IEEE. In Proc. of 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016. P. 637–642.
- [4] Cooperative space object tracking using space-based optical sensors via consensus-based filters / B. Jia, K. D. Pham, E. Blasch et al. // IEEE Transactions on Aerospace and Electronic Systems. 2016. Vol. 52, no. 4. P. 1908–1936.
- [5] Сервис-ориентированное мультиагентное управление распределенными вычислениями / И. В. Бычков, Г. А. Опарин, А. Г. Феокистов [и др.] // Автоматика и телемеханика. 2015. № 11. С. 118–131.
- [6] Городецкий В.И., Бухвалов О.Л., Скобелев П.О. Современное состояние и перспективы промышленных применений многоагентных систем // Управление большими системами. 2017. № 66. С. 94–157.
- [7] Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов. М.: ФИЗМАТЛИТ, 2009. 280 с.
- [8] Rzevski G., Skobelev P. Managing Complexity. Wit Press, 2014. 216 p.
- [9] Wooldridge M. An introduction to Multi-agent Systems. John Wiley & Sons, 2009. 484 p.
- [10] Kalman R. E. A new approach to linear filtering and prediction problems // Journal of basic Engineering. 1960. Vol. 82, no. 1. P. 35–45.
- [11] Uhlmann J. K. Algorithms for multiple-target tracking // American Scientist. 1992. Vol. 80, no. 2. P. 128–141.
- [12] Olfati-Saber R. Distributed kalman filtering for sensor networks. In Proc. of 46th IEEE Conference on Decision and Control, 2007. P. 5492–5498.

- [13] Cattivelli F. S., Sayed A. H. Diffusion strategies for distributed Kalman filtering and smoothing // IEEE Transactions on automatic control. 2010. Vol. 55, no. 9. P. 2069–2084.
- [14] Olfati-Saber R., Sandell N. F. Distributed Tracking in Sensor Networks with Limited Sensing Range. In Proc. of American Control Conference, 2008. P. 3157–3162.
- [15] Consensus-based distributed estimation for target tracking in heterogeneous sensor networks / A. Petitti, D. Di Paola, A. Rizzo et al. In Proc. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011. P. 6648–6653.
- [16] Asynchronous consensus-based distributed target tracking / S. Giannini, A. Petitti, D. Di Paola et al. In Proc. 52th IEEE Conference on Decision and Control (CDC), 2013. P. 2006–2011.
- [17] Yang W., Shi H. Sensor selection schemes for consensus based distributed estimation over energy constrained wireless sensor networks // Neurocomputing. 2012. Vol. 87. P. 132–137.
- [18] Erofeeva V., Granichin O., Granichina O. Multi-Sensor Task Assignment Using Linear Matrix Inequalities in the Multiple Target Tracking Problem. In Proc. of the 18th IFAC Symposium on System Identification, 2018.
- [19] Ерофеева В.А. Оптимизация распределения целей между наблюдателями и оценивание состояний с помощью циклического подхода // Стохастическая оптимизация в информатике. 2018. Т. 52. С. 3–30.
- [20] Metropolis N., Ulam S. The Monte Carlo Method. Journal of the American Statistical Association. American Statistical Association, 1949. Vol. 44. P. 335–341.
- [21] Ермаков С.М., Жигляевский А.А. Математическая теория оптимального эксперимента. М.: Наука, 1975. 471 с.
- [22] Баранов А.Е., Граничин О.Н. Оптимальный регулятор линейного объектов с ограниченной помехой // Автоматика и телемеханика. 1984. № 5. С. 39–46.
- [23] Dahleh M., Pearson J. l^1 -optimal feedback controllers for MIMO discrete-time system // IEEE Transactions on Automatic Control. 1987. Vol. 32, no. 4. P. 314–322.
- [24] Якубович В.А. Метод матричных неравенств в теории устойчивости нелинейных регулируемых систем. I. Абсолютная устойчивость вынужденных колебаний // Автоматика и телемеханика. 1964. Т. 25, № 7. С. 1017–1029.

- [25] Якубович В.А. Метод матричных неравенств в теории устойчивости нелинейных регулируемых систем. II. 1965. Т. 26, № 4. С. 577–590.
- [26] Linear matrix inequalities in system and control theory / S. Boyd, L. El Ghaoui, E. Feron et al. Siam, 1994. Vol. 15.
- [27] Calafiore G., Polyak B. T. Stochastic algorithms for exact and approximate feasibility of robust LMIs. IEEE Transactions on Automatic Control, 2001. Vol. 46. P. 1755–1759.
- [28] Поляк Б.Т., Щербаков П.С. Техника D-разбиения при решении линейных матричных неравенств // Автоматика и телемеханика. 2006. № 11. С. 159–174.
- [29] Polyak B., Khlebnikov M., Shcherbakov P. An LMI approach to structured sparse feedback design in linear control systems. In Proc. 2013 European Control Conference (ECC), 2013. P. 833–838.
- [30] Поляк Б.Т., Хлебников М.В., Щербаков П.С. Управление линейными системами при внешних возмущениях: Техника линейных матричных неравенств. М.: ЛЕНАНД, 2014. 560 с.
- [31] Polyak B. T., Khlebnikov M. V., Shcherbakov P. S. Sparse feedback in linear control systems // Automation and Remote Control. 2014. Vol. 75, no. 12. P. 2099–2111.
- [32] Железнов К.О., Хлебников М.В. Синтез обратной связи для линейной системы управления с возмущением на входах и выходах: робастная постановка // Проблемы управления. 2017. Т. 3, № 0. С. 11–16.
- [33] Creately: Diagram Maker. URL: <https://creately.com> (online; accessed: 24.04.2018).
- [34] MariaDB. URL: <https://mariadb.com> (online; accessed: 05.03.2018).
- [35] CVXPY. URL: <http://www.cvxpy.org> (online; accessed: 25.12.2017).