ISSN 2658-5014 (Print) СИИТ ISSN 2686-7044 (Online)
http://siit.ugatu.su 2025. Т. 7, № 4 (23)
СИСТЕМНАЯ ИНЖЕНЕРИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
Научная статья

УДК 004.8 DOI 10.54708/2658-5014-SIIT-2025-no4-p3 EDN TBEBLS

TRANSPORT-BY-THROWING – РОБОТОТЕХНИЧЕСКИЙ ПЕРЕБРОС ПРЕДМЕТОВ: АЛГОРИТМ ПРОГНОЗИРОВАНИЯ ТРАЕКТОРИИ

К. В. Миронов

Аннотация. Перемещение предметов перебросом — это перспективный способ робототехнической транспортировки деталей в гибких производственных системах. Эта статья, являющаяся продолжением предшествующих работ автора, посвящена разработке алгоритма прогнозирования траектории бросаемого предмета. Обсуждается задача построения полезной модели для прогнозирования траектории брошенного объекта. Разработка и первоначальная проверка предложенных идей осуществляются с использованием упрощенного моделирования движения и результатов экспериментов по метанию. Дается краткое введение в среду моделирования. Для прогнозирования применяется и разрабатывается метод двух ближайших соседей (2NN). Для повышения эффективности и скорости алгоритма предлагается привязка преобразований координат. Исследуется прогнозирование траектории на основе метода kNN. Также предлагается сравнение текущей траектории только с небольшим подмножеством всего набора данных.

Ключевые слова: робототехническая система; транспортировка; переброс; захват; статистическая оценка баллистических кривых; трекинг движущихся объектов; алгоритм ближайших соседей.

Цитирование: Миронов К. В. Transport-by-Throwing – робототехнический переброс предметов: алгоритм прогнозирования траектории // СИИТ. 2025. Т. 7, № 4 (23). С. 3–28. EDN TBEBLS.

Введение

Эта статья является продолжением работ [Мир24а, Мир24б и Мир25а] и посвящена разработке алгоритма прогнозирования траектории бросаемого предмета. Здесь обсуждается задача построения полезной модели для прогнозирования траектории брошенного объекта. Разработка и первоначальная проверка предложенных идей осуществляются с использованием упрощенного моделирования движения и результатов экспериментов по метанию. В разделе 1 дается краткое введение в среду моделирования. Для прогнозирования применяется и разрабатывается метод двух ближайших соседей (2NN). Для повышения эффективности и скорости алгоритма предлагается привязка преобразований координат (раздел 2). Операция прогнозирования траектории kNN исследуется в разделе 3. Также предлагается сравнение текущей траектории только с небольшим подмножеством всего набора данных (раздел 4).

В соответствии с определением задачи, приведенным в [Мир24а], предсказатель имеет следующие входные данные: эталон оценочных измерений текущей траектории брошенного объекта

$$X_c = \begin{pmatrix} x_{c1}(0) \\ x_{c2}(0) \\ x_{c3}(0) \end{pmatrix}, \begin{pmatrix} x_{c1}(1) \\ x_{c2}(1) \\ x_{c3}(1) \end{pmatrix}, \begin{pmatrix} x_{c1}(2) \\ x_{c2}(2) \\ x_{c3}(2) \end{pmatrix}, \dots, \begin{pmatrix} x_{c1}(t) \\ x_{c2}(t) \\ x_{c3}(t) \end{pmatrix}$$

и большой, но конечный набор предыдущих траекторий, измеренных как в области измерения, так и в области захвата

$$L = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_m, Y_m)\}.$$

Каждое измерение положения $X(t)$ состоит из временного индекса $t = 1, 2, \dots, n$ и измеренных значений координат $x_{c1}(t)$, $x_{c2}(t)$, $x_{c3}(t)$ в декартовой системе координат, связанной с оптическим центром левой камеры. Это положение системы координат устанавливается с помощью операции стереотриангуляции[1]. Для каждой траектории k из набора данных, соответствующей

[1] Camera Calibration Toolbox for Matlab. URL: http://www.vision.caltech.edu/bouguetj/calib_doc.

каждому моменту времени t после выпуска, когда было произведено измерение, у нас имеется соответствующая точка $X_k(t)$. Предполагается, что измерения с одинаковым временным индексом t были сделаны в один и тот же промежуток времени после выпуска для всех траекторий. Задача предсказателя состоит в том, чтобы оценить значение Y_C с использованием этого входного сигнала. Как отмечалось в [Мир24б], основным методом решения этой задачи является алгоритм k ближайших соседей.

1. Средства первоначальной проверки

Цель этого раздела – описать средства, используемые в следующих разделах для первоначальной проверки алгоритма прогнозирования. Эта проверка проводится для того, чтобы полностью исключить бесполезные методы и теоретически сравнить основные идеи. Первоначальная проверка основана на двух вариантах: моделирование полета объекта на основе упрощенных моделей движения и применение алгоритмов к полученным наборам данных из работы [Мир25а]. Упрощенное моделирование полета используется для проверки работоспособности алгоритма в целом, то есть в идеальных условиях, а применение к реальным полетным данным должно показать практическую применимость концепций.

Физические модели полета снаряда либо сложны, либо неточны [Мир24б]. Моделирование, основанное на упрощенных моделях, помогает определить, являются ли концепции полностью бесполезными, но, чтобы быть уверенным в их использовании в экспериментах, необходимы реальные данные. Моделирование полета основано на модели гравитационного сопротивления, выраженной дифференциальным уравнением [Мир24б, формула (10)]. Поскольку это уравнение не может быть решено аналитически, движение объекта вычисляется итеративно, применяя следующую процедуру.

Первая итерация устанавливается точкой выпуска. Имитируемое бросающее устройство отклоняется в скорости высвобождения v в различных измерениях. Предполагалось, что эти отклонения имеют случайные ошибки с нормальным распределением вокруг номинальных значений со стандартными отклонениями 0.1 м/с для значений скорости в каждом направлении, что выглядит более оптимистично, чем 0.16 и 0.33 м/с, измеренные для метательного устройства [Мир25а, табл. 1]. Такое предположение допускается, поскольку моделирование идеально и предназначено для проверки алгоритма.

Как только объект брошен, новые значения положения и скорости объекта

$$X = \begin{pmatrix} x_1(i) \\ x_2(i) \\ x_3(i) \end{pmatrix}, \qquad v = \begin{pmatrix} v_1(i) \\ v_2(i) \\ v_3(i) \end{pmatrix}$$

вычисляются с помощью следующей операции:

$$\begin{pmatrix} a_1(i) \\ a_2(i) \\ a_3(i) \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix} - k \begin{pmatrix} v_1(i-1) \\ v_2(i-1) \\ v_3(i-1) \end{pmatrix} \sqrt{v_1^2(i-1) + v_2^2(i-1) + v_3^2(i-1)}\,, \tag{1}$$

$$\begin{pmatrix} v_1(i) \\ v_2(i) \\ v_3(i) \end{pmatrix} = \begin{pmatrix} v_1(i-1) \\ v_2(i-1) \\ v_3(i-1) \end{pmatrix} + \begin{pmatrix} a_1(i) \\ a_2(i) \\ a_3(i) \end{pmatrix} \Delta t\,, \tag{2}$$

$$\begin{pmatrix} x_1(i) \\ x_2(i) \\ x_3(i) \end{pmatrix} = \begin{pmatrix} x_1(i-1) \\ x_2(i-1) \\ x_3(i-1) \end{pmatrix} + \frac{1}{2} \left(\begin{pmatrix} v_1(i-1) \\ v_2(i-1) \\ v_3(i-1) \end{pmatrix} + \begin{pmatrix} v_1(i) \\ v_2(i) \\ v_3(i) \end{pmatrix} \right) \Delta t\,. \tag{3}$$

Laminar Air Flow – Welcome to the Tubus Bauer webpage. URL: http://www.tubusbauer.de/laminar-air-flow.html.

Эта операция представляет собой итеративное решение дифференциального уравнения [Мир24б, формула (10)]. Здесь $a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ – вектор ускорения объекта, i – количество итераций, k – коэффициент лобового сопротивления, Δt – разница во времени между двумя итерациями. X выражается в системе координат, где первое измерение коллинеарно направлению силы тяжести, а третье измерение коллинеарно номинальному направлению броска. Значение k было рассчитано в соответствии с [Мир24б, уравнение (6)] с использованием значений коэффициентов сопротивления для теннисного мяча в стандартных условиях [Ala10]. Значение Δt равно 1 мс (для лучшей интерпретируемости). Применение модели с этими значениями Δt и k существенно не отличается (не более чем на один миллиметр) от результатов [Tut13]. Смоделированная частота кадров наблюдателя была установлена на 100 кадров в секунду. Это значение близко к 110 кадрам в секунду реальной системы наблюдения. Таким образом, для простоты и интерпретируемости лучше всего использовать целое число миллисекунд в качестве межкадрового периода. Поскольку шаг пересчета положения составляет 1 мс, измеряется каждое 10-е вычисленное положение.

Эта модель позволяет оценить алгоритм абсолютному наблюдателю. Если есть необходимость имитировать ошибочного наблюдателя, ошибки могут быть случайным образом распределены вокруг значений, сгенерированных симулятором, с заданным значением стандартного отклонения. Моделирование процесса в основном применялось для проверки операции прогнозирования kNN (раздел 3) и распределения подмножеств (раздел 4). Последовательность преобразований координат (раздел 2) была подтверждена на основе наборов данных, полученных в ходе экспериментов по метанию [Мир25а].

2. Преобразования координат

Прогнозы основаны на сравнении текущей траектории с траекториями из базы данных. Цель такого сравнения состоит в том, чтобы найти траекторию, которая имеет максимально схожую форму с текущей. Однако пространственные координаты X_c объекта в системе координат камеры (эти положения возвращаются операцией стереотриангуляции) зависят не только от формы траектории, но также от взаимного положения камеры, объекта и горизонтального направления броска. Как показано в разделе 3, разностной метрикой для траекторий является среднее евклидово расстояние между соответствующими точками. Если стартовые точки для двух разных бросков имеют определенное евклидово расстояние d друг от друга, разница между ними будет почти равна $d_0 + d$, где d_0 – разница в гипотетическом случае, когда эти траектории могут иметь одинаковые стартовые точки. Если две траектории одинаковой формы имеют одну и ту же стартовую точку и различные горизонтальные направления броска, расстояние между координатами соответствующих точек будет увеличиваться со временем. Даже если будет обнаружено сходство формы этих траекторий, прогноз kNN, основанный на координатах камеры, будет ошибочным. Эти эффекты проиллюстрированы на рис. 1.

Можно настроить условия броска таким образом, чтобы связь между бросающим устройством и камерами была постоянной. Также возможно свести к минимуму отклонение горизонтального направления броска, которое не может быть полностью устранено из-за отклонения устройства (подраздел 2.3). Такой подход требует сбора больших баз данных, чтобы учитывать различные возможные горизонтальные направления броска и поддерживать одну и ту же настройку на этапе обучения и функционирования. Фактически, для каждой новой конфигурации системы требуется, чтобы траектории изучались заново, что снижает простоту реконфигурации системы, являющуюся одним из основных преимуществ TbT [Мир25а, разд. 2].

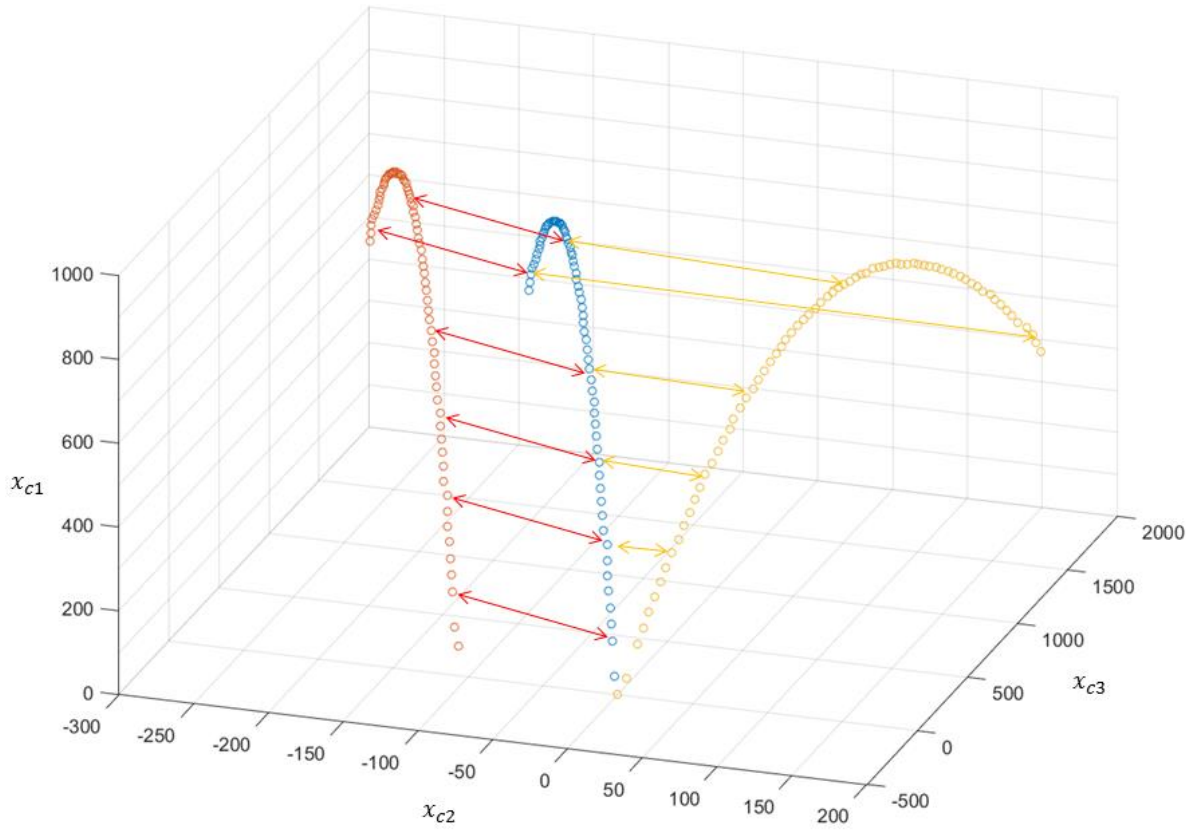

Рис. 1 Евклидово расстояние между точками различных траекторий с одинаковой временной меткой в координатах камеры зависит от положения точки запуска. Все три построенные траектории имеют одинаковую форму, но красная имеет другую точку старта, а желтая – другой азимут броска, – Евклидово расстояние велико

Целью эталона преобразования координат является предоставление системы координат для хранения и прогнозирования траектории, которая позволяет сравнивать их только на основе свойств формы (независимо от азимута броска и пространственного положения точки запуска) и правильного прогноза траектории на основе метода kNN.

2.1. Краткий обзор

Справка состоит из трех преобразований. Прежде всего, определяются координаты

$$X_g = \begin{pmatrix} x_{g1} \\ x_{g2} \\ x_{g3} \end{pmatrix},$$

связанные с гравитацией. Координатная ось x_{g1} коллинеарна направлению силы тяжести. Цель этого преобразования – локализовать гравитацию в одном пространственном измерении и упростить дальнейшие вычисления (подраздел 2.2). Кроме того, координаты, связанные с гравитацией, проецируются на 2D-плоскость полета (Plane-of-Flight, POF) $X_p = \begin{pmatrix} x_{p1} \\ x_{p2} \end{pmatrix}$ (подраздел 2.3). Преобразование обеспечивает неизменность координат относительно горизонтального направления броска. Наконец, одна из измеренных точек траектории выбирается в качестве нулевой точки для координат $X_z = \begin{pmatrix} x_{z1} \\ x_{z2} \end{pmatrix}$ (подраздел 2.4). При вычитании координат нулевой точки обеспечивается инвариантность траектории относительно взаимного положения метательного устройства и системы камер. Прогнозирование координат объекта производится в системе координат нулевой точки. Результаты прогнозирования затем могут быть

преобразованы обратно в систему координат, связанную с камерой, или в систему координат, связанную с роботом, что полезно для определения движения захвата. Эти обратные преобразования обсуждаются в подразделе 2.5. Все преобразования координат основаны на проективной геометрии. Основные принципы проективной геометрии кратко описаны ниже.

Преобразование координат из трехмерной системы координат α в трехмерную систему координат b выполняется путем умножения матрицы преобразования на координаты точки в однородной системе координат:

$$\begin{pmatrix} x_{b1} \\ x_{b2} \\ x_{b3} \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & x_{01} \\ p_{21} & p_{22} & p_{23} & x_{02} \\ p_{31} & p_{32} & p_{33} & x_{03} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{a1} \\ x_{a2} \\ x_{a3} \\ 1 \end{pmatrix} \tag{4}$$

или в более компактной версии:

$$X_b = P_{AB}\, X_a, \tag{5}$$

где $X_0 = \begin{pmatrix} x_{01} \\ x_{02} \\ x_{03} \end{pmatrix}$ – положение начала координат для системы координат α в системе b; $P_{\text{rot}} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix}$ – матрица вращения. Векторы

$$P_1 = \begin{pmatrix} p_{11} \\ p_{21} \\ p_{31} \end{pmatrix}, \qquad P_2 = \begin{pmatrix} p_{12} \\ p_{22} \\ p_{32} \end{pmatrix}, \qquad P_3 = \begin{pmatrix} p_{13} \\ p_{23} \\ p_{33} \end{pmatrix}$$

равны нормальным векторам, коллинеарным направлению осей координат из системы координат α в систему координат b. Если известны значения двух векторов из $\{P_1, P_2, P_3\}$, то этого достаточно для определения третьего, поскольку он равен их перекрестному произведению:

$$P_3 = P_1 \times P_2. \tag{6}$$

Это утверждение подтверждается следующими факторами. Если векторный триплет $\{P_1, P_2, P_3\}$ правильный, то направление P_3 коллинеарно произведению P_1 и P_2. Длина вектора произведения равна

$$|P_1 \times P_2| = |P_1||P_2| \sin\frac{\pi}{2} = 1 \times 1 \times 1 = 1 = |P_3|. \tag{7}$$

Обратное преобразование координат из A в B достигается путем решения (4) в виде системы линейных уравнений с неизвестными $\{x_{a1}, x_{a2}, x_{a3}\}$. Другой способ вычисления X_a из X_b и P_{AB} X – умножение обратного значения P_{AB} на X_b:

$$X_a = P_{AB}^{-1}\, X_b\,. \tag{8}$$

Другими словами, матрица перехода для обратного преобразования вычисляется как обратная базовой матрице перехода:

$$P_{BA} = P_{AB}^{-1}\,. \tag{9}$$

Преобразование координат на основе матрицы обратного перехода в дальнейших подразделах встречается чаще, чем на основе матрицы прямого перехода. Причина этого заключается в том, что новая система координат чаще определяется положением ее начала координат и осей координат в старой системе координат, а не наоборот.

При определении новой системы координат часто возникает задача подгонки плоскости или определения плоскости. В подразделе 2.2 это представлено как определение горизонтальной плоскости, основанной на векторе нормали, а в подразделе 2.3 – как плоскость, соответствующая измеренным координатам. Стандартное уравнение, выражающее плоскость в трехмерном пространстве, выглядит следующим образом:

$$Ax_1 + Bx_2 + Cx_3 + D = 0\,. \tag{10}$$

Здесь A, B, C, D – коэффициенты параметров плоскости. Очевидно, что их значения могут различаться в выражении одной и той же плоскости, в то время как пропорция

$$A : B : C : D = \text{const}.$$

Здесь и ниже рассматривается случай, когда $A^2 + B^2 + C^2 = 1$. Вектор

$$P = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

перпендикулярен плоскости цели. Если это вектор нормали к плоскости ($A^2 + B^2 + C^2 = 1$), уравнение может быть преобразовано в следующее:

$$Ax_1 + Bx_2 + Cx_3 - Ax_{01} - Bx_{02} - Cx_{03} = 0. \quad (11)$$

Здесь $X_0 = (x_1(0); x_2(0); x_3(0))$ – определенная точка целевой плоскости, для которой D из уравнения (10) равно $-Ax_1(X_0) - Bx_2(X_0) - Cx_3(X_0)$. Это выражение наиболее полезно для преобразований систем координат. Оно позволяет определять плоскость на основе известных значений вектора нормали P и начальной точки X_0, лежащей на плоскости.

Другой способ – выразить значение одной координаты как зависящее от двух других координат:

$$x_2 = ax_1 + bx_3 + c\,, \quad (12)$$

где $\alpha = A/B$; $b = C/B$; $c = \frac{-Ax_{01} - Bx_{02} - Bx_{03}}{B}$. Это выражение плоскостей используется в наборе инструментов MATLAB для подгонки кривых. Переход к стандартной форме уравнения плоскости может быть осуществлен путем определения значений $\{A, B, C, D\}$ с использованием следующих формул:

$$B = \frac{-1}{\sqrt{a^2+b^2+1}},\ A = \frac{a}{\sqrt{a^2+b^2+1}},\ C = \frac{b}{\sqrt{a^2+b^2+1}},\ D = \frac{c}{\sqrt{a^2+b^2+1}}. \quad (13)$$

,

Выражение (11) может быть определено с помощью этих значений.

2.2. Система координат, связанная с гравитацией

Многие из ранее упомянутых уравнений, например (1)–(3), предполагают, что первое измерение системы координат коллинеарно силе тяжести, а третье коллинеарно горизонтальному направлению движения объекта. Расчет физических моделей движения в такой системе проще, поскольку нет необходимости учитывать гравитацию как значительную силу во всех измерениях, кроме первого. Для текущего алгоритма основная причина определения системы координат силы тяжести заключается в том, что проще отличить плоскость полета (раздел 2.3) от нее, а не от системы координат камеры.

В экспериментальной установке направление силы тяжести определяется на основе тяжелого груза, подвешенного на длинном гвозде. Если на гвозде отмечены две удаленные точки, tt_1 и tt_2, направление силы тяжести может быть определено как вектор от верхней точки к нижней точке. Чтобы определить начало координат гравитационной системы координат, на световом барьере, установленном на метательном устройстве, выбираются две точки L_1 и L_2. Начало координат системы координат находится посередине между ними, так что оно находится рядом с точкой запуска мяча. x_3 и x_2 лежат в плоскости, перпендикулярной x_1. Направление x_2 задается проекцией L_1L_2 на эту плоскость. В этом случае x_3 находится вблизи номинального направления броска.

Иллюстрация этого определения приведена на рис. 2.

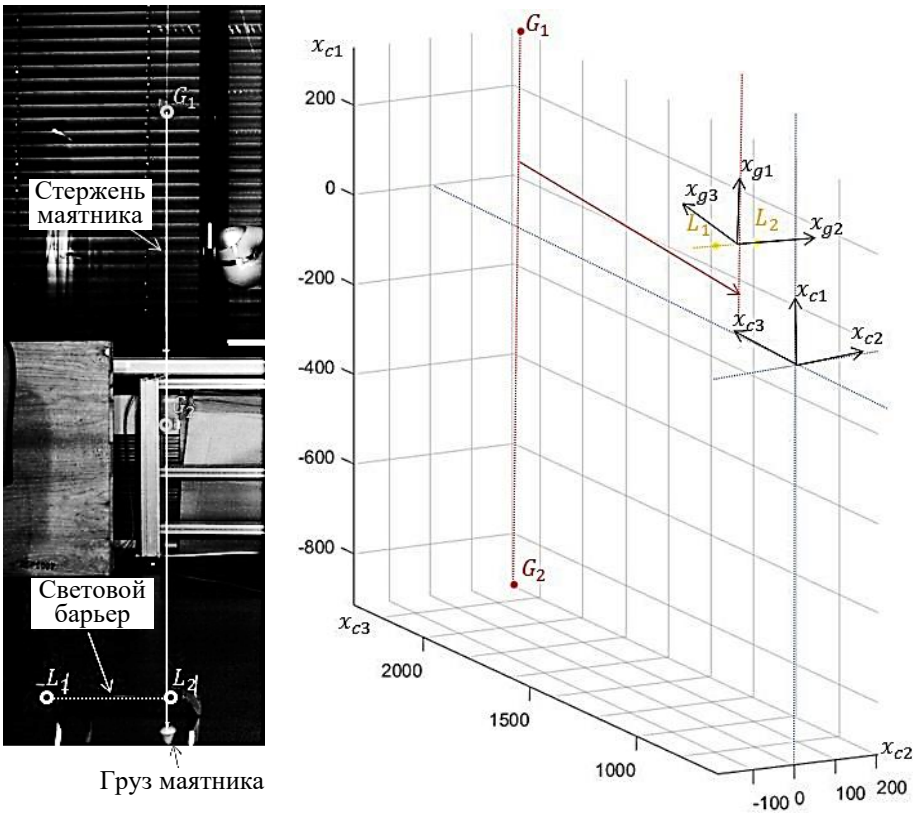

Рис. 2 Определение гравитационных систем координат на основе изображений с камер. Маятник с тяжелым грузом подвешен в поле зрения камеры. Когда маятник статичен, направление гвоздя коллинеарно силе тяжести. Вектор силы тяжести определяется на основе двух красных точек, G_1 и G_2, на стержне. Начало координат системы координат определяется на основе двух желтых точек L_1 и L_2 на световом барьере. На левой стороне показана часть калибровочного изображения с выделенными опорными точками G_1, G_2, L_1 и L_2. В правой части рисунка эти точки проецируются в систему координат 3D-камеры, и показано положение системы координат гравитации

Гравитационная система координат определяется на основе знания 4 точек $\{G_1, G_2, L_1, L_2\}$ следующим образом. Исходная точка X_0 определяется как средняя точка на отрезке L_1L_2

$$X_0 = \frac{1}{2}(L_1 + L_2). \tag{14}$$

Координатная ось x_1 определяется как нормализованный вектор G_2G_1

$$P_1 = \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \frac{G_1 - G_2}{|G_1 - G_2|}. \tag{15}$$

Горизонтальная плоскость может быть определена с использованием уравнения (11) на основе параметров, полученных в соответствии с (14) и (15). Координатная ось x_2 определяется как нормализованная проекция вектора L_1L_2 на горизонтальную плоскость, которая выражается (11). Поскольку начало координат системы координат лежит на L_1L_2, достаточно определить проекцию точки L_1

$$\Lambda = \begin{pmatrix} x_{\Lambda 1} \\ x_{\Lambda 2} \\ x_{\Lambda 3} \end{pmatrix}.$$

Это определение производится в два этапа. На первом этапе определяется линия, нормальная к плоскости (то есть коллинеарная нормали к плоскости) и включающая точку L_1.

Каноническое линейное уравнение в пространстве имеет вид системы из двух уравнений, выраженных одной формулой с двумя знаками равенства:

$$\frac{x_1 - x_{01}}{A} = \frac{x_2 - x_{02}}{B} = \frac{x_3 - x_{03}}{C}. \quad (16)$$

На втором этапе решается система уравнения (16) и уравнения плоскости (11), чтобы определить точку, в которой линия пересекает плоскость

$$\frac{1}{A}x_{\Delta 1} - \frac{1}{B}x_{\Delta 2} + 0x_{\Delta 3} = \frac{x_{01}}{A} - \frac{x_{02}}{B}, \quad 0x_{\Delta 1} + \frac{1}{B}x_{\Delta 2} - \frac{1}{C}x_{\Delta 3} = \frac{x_{02}}{B} - \frac{x_{03}}{C}, \quad (17)$$

$$Ax_{\Delta 1} + Bx_{\Delta 2} + Cx_{\Delta 3} = x_{01} + x_{02} + x_{03}.$$

Координатная ось x_3 определяется как перекрестное произведение x_1 и x_2.

2.3. 2D-представление для 3D-координат

Когда объект находится в воздухе, сила тяжести направлена вниз, а сопротивление воздуха направлено назад. Следовательно, если эффект Магнуса и асимметрия тела не оказывают большого влияния (для невращающихся тел, таких как параллелепипеды), можно предположить, что общая сила, влияющая на брошенный объект, лежит в вертикальной плоскости. Это так называемая плоскость полета (Plane-of-Flight, PoF), которая определяется двумя векторами: гравитацией – g и горизонтальной проекцией скорости объекта – v_{hor}. Если все силы, воздействующие на тело, лежат в такой плоскости и вектор скорости также лежит в этой плоскости, то будущее движение объекта под воздействием этих сил также будет происходить в этой плоскости. Другими словами, траектория тела в данном случае плоская. Это обстоятельство привело к идее расчета траектории в двумерной системе координат, связанной с PoF. Эта трансформация имеет следующие потенциальные преимущества:

Это обеспечивает неизменность горизонтального направления броска. Прогноз ближайших соседей основан на поиске в базе данных траекторий, аналогичных текущей. Когда сравнение двух бросков с разным горизонтальным направлением выполняется в 3D-системе координат, связанной с камерой или гравитацией, сходство их форм не может быть обнаружено из-за больших расстояний между точками одной и той же временной метки. В плоских координатах траектории аналогичной формы распознаются как аналогичные.

Уменьшение размерности пространства означает уменьшение порядка вычислений. Решаемая задача в такой системе потенциально более проста и более устойчива к ошибкам.

Подгонка плоскости к данным измерений позволяет обнаруживать отклонения при измерении траектории. Если точки измерены точно, они будут хорошо соответствовать PoF. Точки с плохой подгонкой считаются выбросами и не включаются в дальнейшие расчеты.

Наиболее полезным представлением плоскости для задачи подгонки является выражение (12), поскольку используются только три параметра (вместо шести параметров, как в (11) и четырех параметров в (10)), и значения этих параметров уникальны и независимы друг от друга (вместо (10), где значения параметров могут изменяться, если соотношение между ними является постоянным). Следовательно, подгонка плоскости здесь представляет собой поиск в пространстве 3D-параметров. Задача подгонки PoF может стать еще проще, если мы предположим, что PoF является вертикальным в исходной системе координат. Например, если точка $Q = \begin{pmatrix} x_{q1} \\ x_{q2} \\ x_{q3} \end{pmatrix}$ лежит на PoF, то точка $Q' = \begin{pmatrix} q \\ x_{q2} \\ x_{q3} \end{pmatrix}$ будет лежать на PoF для любого реального q значение α в уравнении (12) равно нулю, а плоскость представлена следующим уравнением:

$$x_2 = bx_3 + c. \quad (18)$$

В этом случае задача подгонки представляет собой поиск в 2D-пространстве параметров. Фактически, это линия, вписывающаяся в плоскость, выраженную через x_3 и x_2. Здесь параметр b равен касательной угла α между PoF и осью x_3, в то время как параметр c представляет точку на оси x_2, где он пересекает PoF (рис. 3).

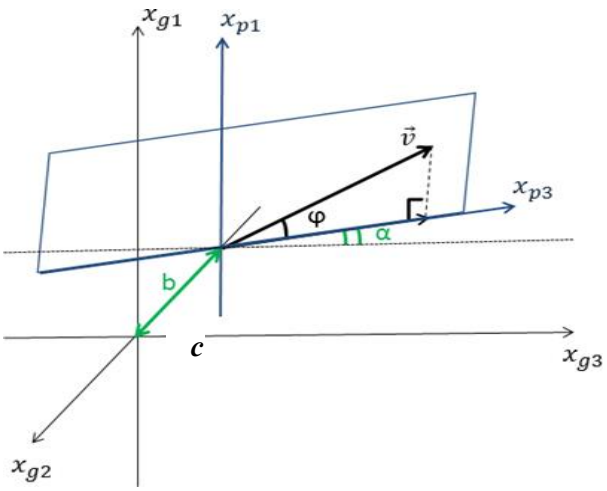

Рис. 3 Плоскость полета в системе координат, связанной с гравитацией

Это упрощение является основной мотивацией при определении координат, связанных с гравитацией, в подразделе 2.2, поскольку в координатах, связанных с гравитацией, PoF является вертикальным. Если мы оперируем координатами объекта в системе координат камеры, направление гравитационной силы, соответствующее направлению x_1 в предыдущих описанных расчетах, неясно, как и точка запуска, соответствующая основанию системы координат. Определение параметров плоскости полета в данном случае является задачей подгонки плоскости в 3D-пространстве. На рис. 3 показано, как эта плоскость расположена в гравитационной системе координат. Легко видеть, что если положение PoF определяется углом α, который находится между горизонтальной проекцией скорости объекта и осью x_1, то это азимут броска. Система координат PoF определяется осью x_1 и осью x_4, которая коллинеарна направлению броска:

$$x_{g1}(t) = x_{p1}(t), \quad x_{g2}(t) = (x_{p3}(t) - c)\cos\alpha\,, \; x_{g3}(t) = (x_{p3}(t) - c)\sin\alpha\,. \tag{19}$$

Если точка лежит в PoF и ее 3D-координаты известны, преобразовать ее в координаты PoF не составляет проблемы:

$$x_{p3}(t) = \frac{x_{g3}}{\cos\alpha} + c. \tag{20}$$

Проблема этого подхода заключается в том, что для преобразования необходимо знать α. На самом деле, значение самого угла не используется в уравнениях преобразования. Необходимыми параметрами являются $\sin\alpha$ и $\cos\alpha$. Тангенс α может быть взят в качестве параметра b из уравнения (18), а $\sin\alpha$ и $\cos\alpha$ могут быть рассчитаны по касательной с использованием стандартных тригонометрических уравнений:

$$\cos\alpha = (\tan^2\alpha + 1)^{-\frac{1}{2}} = \frac{1}{\sqrt{b^2+1}}\,, \tag{21}$$

$$\sin\alpha = \tan\alpha\cos\alpha = \frac{b}{\sqrt{b^2+1}}\,. \tag{22}$$

Пусть $q_1 = \frac{1}{\cos\alpha}$ и $q_2 = \sin\alpha$ являются промежуточными константами, используемыми в процессе преобразования координат из 3D в PoF. Тогда

$$q_2 = \sqrt{b^2+1}\,, \; x_{p3} = q_1\big(x_{g3} - c\big), \; x_{p1} = x_{g1}. \tag{23}$$

Процесс преобразования координат обратно из PoF в 3D выглядит следующим образом:

$$q_2 = \frac{b}{q_1}\,, \; x_{g3}(t) = \frac{x_{p3}(t)}{q_1}, \; x_{g2}(t) = cx_4(t), \; x_{g1} = x_{p1}. \tag{24}$$

Чтобы сделать эталон преобразования однородным, координаты PoF объекта считаются 3D-вектором. Значения координат хранятся в x_{p1} (высота) и x_{p3} (расстояние). Координата x_{p2} является фиктивной. В операции прогнозирования траектории предполагается, что ее значение каждый раз равно нулю. Если процесс преобразования выполняется в матричной форме (то есть соответствует уравнению (4)), параметры матрицы преобразования определяются следующим образом.

Исходная точка X_0 определяется как точка пересечения между осью PoF и осью x_{g2}:

$$X_0 = \begin{pmatrix} 0 \\ c \\ 0 \end{pmatrix}. \tag{25}$$

Координатная ось x_{p1} равна x_{g1}.

$$P_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{26}$$

Координатная ось x_2 определяется как коллинеарная проекции PoF на горизонтальной плоскости:

$$P_2 = \begin{pmatrix} 0 \\ \sin\alpha \\ \cos\alpha \end{pmatrix}. \tag{27}$$

Координатная ось x_3 определяется как перекрестное произведение x_1 и x_2.

Значение x_2 равно нулю для всех точек, лежащих на PoF. Для других точек значение этой координаты равно расстоянию от PoF. Из-за ошибок наблюдения полученные значения координат обычно не лежат в одной плоскости, и анализ x_{p2} позволяет определить качество подгонки.

Ранее в этом подразделе предполагалось, что значения α и c уже известны. Однако в действительности эти значения определяются отклонением метательного устройства. Поскольку нет средств для непосредственного измерения этого отклонения, значения α и c следует оценивать путем подгонки плоскости к данным измерений. Эта оценка может быть осуществлена с помощью различных статистических методов. Метод оценки параметров PoF должен обладать следующими свойствами [Mir15]:

Точность подгонки: Определение точности является сложной задачей, поскольку нет достоверных данных о координатах объекта. Если точки будут переведены в PoF, а затем обратно в 3D, их координаты будут отличаться от исходных значений. Трудно сказать, является ли это отклонение результатом неточности алгоритма или исправления ошибок измерения. Согласно [Pon15], в позиционировании сфер отсутствует существенная составляющая систематической ошибки, и отклонения являются случайными по внешнему виду. Нормальный внешний вид разницы (например, растущая разница со временем) означает, что значения a и c оцениваются с ошибками.

Надежность: Алгоритм должен работать правильно, даже если есть несколько точек с совершенно неверными координатами.

Стабильность: Во время полета объекта значения α и c пересчитываются после каждого нового полученного кадра. Если механизм оценки PoF вероятен, они не должны сильно изменяться после каждого пересчета, но должны быть скорректированы до определенного значения. Оценочные значения параметров должны быть обязательно одинаковыми, когда для оценки используется вся траектория $\{X_g(1), X_g(2), ..., X_g(n)\}$ и когда известна только часть входных данных $\{X_g(1), X_g(2), ..., X_g(m)\}$. Это требование связано с обстоятельствами прогнозирования. Предсказателю нужно только сравнить текущую траекторию, измеренную в начальной части, и траектории из базы данных, которые полностью отслеживаются.

Производительность: Поскольку приложение работает в режиме реального времени, все вычисления должны быть выполнены в течение определенного периода времени.

Для определения α и b было применено несколько методов оценки: наименьшие квадраты (least squares – LS), надежные наименьшие квадраты (random sample consensus – RLS), RANSAC, расчет среднего значения и расчет медианы. Обычно реализация оценки LS заключается в процедуре [Hla12]. Вектор измерения X зависит от вектора состояния θ, как показано в следующем уравнении:

$$X = H\theta + U, \tag{28}$$

где H – матрица, определяющая соотношение между X и θ, а U – шум измерения (то есть погрешность измерения). H имеет размер $m \times n$, где m – количество измеренных значений в X, а n – количество параметров, описывающих состояние системы в θ. Если значение X известно, оценка состояния системы Θ методом наименьших квадратов может быть получена с использованием следующего уравнения [Hla12]:

$$\theta = H^{\#}X\,, \tag{29}$$

где $H^{\#}$ обозначает псевдоинверсию матрицы H.

Чтобы выразить полиномиальные зависимости, матричное уравнение преобразуется в другое представление:

$$P\,\theta = I, \tag{30}$$

где P – матрица измерения координат $m \times n$ (m – количество измерений; n – размерность пространства в случае линейной зависимости); θ – вектор полиномиальных коэффициентов, а I – вектор единиц размера m. В случае подгонки линии в координатной плоскости $x_{g2}Ox_{g3}$ это уравнение имеет следующий вид:

$$\begin{pmatrix} x_{g2}(1) & x_{g3}(1) \\ x_{g2}(2) & x_{g3}(2) \\ \vdots & \vdots \\ x_{g2}(m) & x_{g3}(m) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \tag{31}$$

Уравнение целевой линии выглядит следующим образом:

$$Ax_1 + Bx_2 = 1. \tag{32}$$

Его можно легко преобразовать в зависимость x_2 от x_3

$$x_1 = bx_2 + c, \tag{33}$$

где $b = A^{-1}$; $c = -B\,b$.

Оценка методом наименьших квадратов A и B получается с использованием операции псевдоинверсии:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} x_{g2}(1) & x_{g3}(1) \\ x_{g2}(2) & x_{g3}(2) \\ \vdots & \vdots \\ x_{g2}(m) & x_{g3}(m) \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{34}$$

LS показал высокую скорость (выполнение за 1–2 мс даже в такой относительно медленной среде, как MATLAB) и достаточную стабильность. Он работает достаточно точно на траекториях, которые не содержат выбросов. Ошибки, по-видимому, случайным образом распределены вокруг нуля и в большинстве случаев не превышают нескольких миллиметров. Применение LS к полученным данным показало, что LS не является надежным. Отклонения могут привести к ошибкам до нескольких десятков сантиметров.

Применение RLS[2] улучшает устойчивость к выбросам. Была применена надежная оценка методом наименьших квадратов с помощью набора инструментов для подгонки кривой в MATLAB. Вероятность того, что он смог подогнать точки к линии в двумерной системе

[2] Curve fitting toolbox – MATLAB. URL: http://www.mathworks.com/products/curvefitting.

координат, была высокой. Значение x_2 для точек на траектории обычно не превышает 1 мм. Работа функции подгонки MATLAB довольно медленная, для вычисления значений α и b требуется около 20 мс. Эта производительность недостаточна, поскольку частота кадров камеры составляет 110 кадров в секунду, что означает, что получение информации о прогнозе после каждого нового кадра весь цикл обработки данных (позиционирование объекта на обоих изображениях, стереотриангуляция, преобразование в координаты PoF, прогнозирование и преобразование обратно в 3D) не должно превышать 9 мс. Конечно, более низкоуровневая реализация увеличивает скорость.

Оценка PoF RANSAC представляет собой применение обобщенного метода RANSAC к задаче подбора линий [Mir15]. Две точки выбираются случайным образом из траектории. Информации из двух точек достаточно, чтобы получить правильное решение (34). Данное решение дает гипотетический результат. После этого производится проверка, насколько хорошо точки прилегают к этой плоскости. Если подгонка удачна, значит, гипотеза доказана. В противном случае используется другая пара точек. Реализация RANSAC оказалась надежней, чем LS, точность была довольно хорошей.

RANSAC выбирает точки случайным образом, позволяя быстро обрабатывать даже большие объемы данных. При прогнозировании траектории этот объем относительно невелик (менее 100 кадров). Поэтому предлагается детерминированный способ выбора пар точек, поскольку в данном случае он не требует больших вычислительных затрат [Mir15]. Пусть $\{X(1), X(2), ..., X(m)\}$ будет частью траектории, доступной в текущий момент (после m или $m + 1$ кадров, здесь m четно). Берется $m/2$ пары точек: $\{X(1), X(m/2 + 1)\}$, $\{X(2), X(m/2 + 2)\}$, ..., $\{X(m/2), X(m)\}$. Это позволяет выдвинуть $m/2$ гипотезы о положении PoF (то есть значениях a и b). Простейшим методом определения PoF в такой ситуации является вычисление средних значений α и b. Несколько неожиданно, точность и стабильность такой оценки не хуже, чем для LS. Для повышения надежности производится двойная оценка. После первого прогона точки с плохим прилеганием к плоскости удаляются из набора. Затем оценка повторяется. Надежность этого алгоритма не хуже, чем для RLS; однако он также не может работать с траекториями, где частота выбросов высока (систематическая ошибка достигает нескольких сантиметров).

Если вместо средних значений используются медианные значения α и b, надежность оценки повышается. Медиана более надежна с выбросами, потому что в отсортированном списке значений они будут либо в первой, либо в последней позиции, а средняя часть списка состоит из выбросов. Поэтому в качестве метода оценки выбран расчет медианы. Он точен в том смысле, что ошибки, по-видимому, случайным образом распределены вокруг нулевого значения и обычно составляют менее 1 мм. Только в нескольких случаях разница исключала 2 мм. Он также надежен в том смысле, что точность остается неизменной, даже если количество выбросов составляет около 20 % от всего набора данных, и стабилен, поскольку значения a и c не изменяются более чем на вторую цифру в зависимости от $X(1 : n)$ или только $X(1 : m)$). Это также быстрее, чем любой другой метод. Выполнение в MATLAB занимает около 1 мс для обработки траектории, состоящей из 80 точек [Mir15].

Сравнение точности для шести предложенных методов показано в табл. 1.

Таблица 1

Стандартное отклонение при подгонке плоскости с использованием различных предложенных методов

Метод	$\sigma(x_{p2})$, мм	$\sigma(\alpha)$, 10^{-3} рад	$\sigma(c)$, мм
LS	6.50	218.8	365.1
RLS	0.95	3.3	1.3
RANSAC	0.93	3.7	2.0
Среднее	3.67	4.1	2.0
Медиана	0.85	3.2	1.1

Стандартное отклонение для установленных точек от плоскости было измерено на основе четвертого набора данных из [Мир25а, табл. 2]. Он включает в себя 111 траекторий по 100 точек каждая, то есть всего 11 100 точек. Стандартные отклонения для шести предложенных методов оценки перечислены в таблице. В левом столбце указано стандартное отклонение x_2 (то есть расстояние от точек до установленной плоскости). Этот параметр позволяет оценить точность преобразования PoF для фиксированного объема данных. Параметры в двух столбцах справа позволяют оценить стабильность подгонки. Они показывают стандартные отклонения для параметров PoF (α и c), оцененных с различным количеством доступных кадров. Количество кадров было изменено следующим образом. На первой итерации для оценки использовались первые 40 кадров. Во втором случае был использован 41. На последней итерации была взята информация со всех 100 кадров. Из данных, приведенных в таблице, следует, что медианная оценка является наиболее точным методом.

Точность и стабильность медианы оценки также доказывают, что модель PoF верна. Нет значительной силы, направленной в сторону, влияющей на брошенный объект. Если горизонтальная проекция траектории не является прямой линией, ошибки будут зависеть от времени, а параметры PoF будут варьироваться в зависимости от m. Поскольку средняя посадка не включает такой эффект, влияние боковой силы считается незначительным. Эти результаты показывают, что преобразование PoF справедливо для сферических объектов, брошенных линейным пусковым устройством [Mir15].

Значение α демонстрирует отклонения метательного устройства с точки зрения горизонтального направления броска. Гистограмма отклонений α от среднего значения показана на рис. 4. Поскольку значение угла невелико, оно почти равно значению тангенса. Обратите внимание, что угол 30×10^{-3} рад является значимым. Это соответствует смещению на 6 см на расстоянии 2 м.

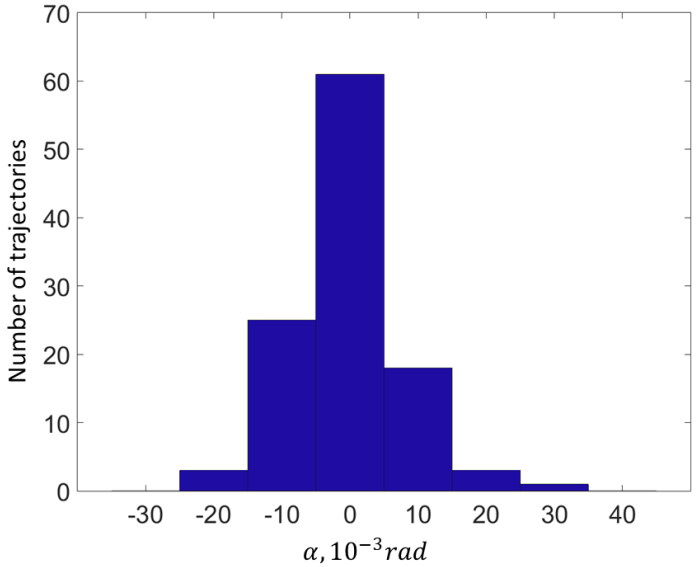

Рис. 4 Гистограмма значений α для траекторий из набора данных

2.4. Инвариантность к точке выпуска

PoF обеспечивает независимость сравнения траекторий с направлением броска. Еще одним аспектом является зависимость сравнения от положения стартовой точки в системе координат. Исходная точка систем координат, связанных с гравитацией и PoF, определяется таким образом, чтобы они были близки к точке в пространстве, где шар пересекает световой барьер. Когда система камер срабатывает, начинается процесс отслеживания. Другими словами, положение z такое, что координаты шара для всех траекторий равны этой точке. В целях

интерпретируемости практично, чтобы координата этой общей нулевой точки была равна $\begin{pmatrix}0\\0\\0\end{pmatrix}$, то есть чтобы нулевая точка была равна исходной точке системы координат. Исходная точка систем координат, связанная с гравитацией и PoF, определяется таким образом, чтобы они были близки к точке в пространстве, где мяч пересекает световой барьер. Когда система камер срабатывает, начинается процесс отслеживания. Другими словами, положение шара при $t = 0$ равно $\begin{pmatrix}0\\0\\0\end{pmatrix}$, и $z = 1$. В действительности этого не происходит. Положение объекта в первом кадре отличается от нуля на несколько миллиметров. Неверная оценка точки запуска может привести к неверным результатам при сравнении траекторий. Идея состоит в том, что можно перевести траекторию в новую систему координат, где нулевая точка более вероятна, чем в X_g или X_p. Для этого в качестве нулевой точки выбирается определенная точка траектории, и ее координаты вычитаются из следующих точек траектории. Другими словами, преобразование координат состоит из вычитания координат нулевой точки.

$$\begin{aligned} x_{z1}(i) &= x_{p1}(i+z) - x_{p1}(z)\,, \\ x_{z2}(i) &= 0\,, \\ x_{z3}(i) &= x_{p3}(i+z) - x_{p3}(z)\,. \end{aligned} \tag{35}$$

где z_1, z_2 и z_3 – координаты объекта в системе координат нулевой точки. Операция вычитания удалена из второй строки, так как мы имеем дело с координатами 2D PoF, а вторая координата не используется в дальнейших вычислениях. Однако он сохранен для обеспечения совместимости с матрицами 3D-преобразования.

Формула преобразования координат матрицы в этом случае будет иметь следующий вид:

$$\begin{pmatrix} x_{z1}(i) \\ x_{z2}(i) \\ x_{z3}(i) \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -x_{p3}(z) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -x_{p3}(z) \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{p1}(i+z) \\ x_{p2}(i+z) \\ x_{p3}(i+z) \\ 1 \end{pmatrix}. \tag{36}$$

Вопрос в том, как правильно выбрать нулевую точку. Важно найти такое значение z, при котором координаты центра мяча измеряются с достаточной точностью. С другой стороны, нулевая точка должна находиться в начале траектории. Сравнение траекторий менее ошибочно, когда между ними существует большая разница, которая должна быть выше возможных ошибок оценки. Разница между измеряемыми точками больше, если они находятся далеко от нулевой точки. Поэтому лучше, если нулевая точка находится рядом с точкой запуска. Однако визуальный анализ из [Мир24б, подразд. 3.1] показал, что измерение положения объекта не очень точное в первых пяти кадрах. Из-за этого 7-й кадр видеоряда предлагается в качестве нулевой точки.

Геометрическая взаимосвязь между системами координат, связанными с гравитацией, PoFи нулевой точкой, показана на рис. 5.

2.5. Обратное преобразование и система координат робота

Преобразования координат, описанные в предыдущих разделах, направлены на повышение эффективности алгоритма прогнозирования. Операция прогнозирования выполняется в системе координат нулевой точки, и результат этой операции также выражается в этой системе. Однако контроллер робота имеет дело с мировыми координатами, поэтому требуется обратное преобразование координат результатов прогнозирования.

В предыдущих разделах были определены следующие матрицы преобразования:

- P_{cg} из системы, связанной с камерой, в систему, связанную с гравитацией (14)–(17);
- P_{gp} из системы, связанной с гравитацией, в систему, связанную с PoF (25)–(27);
- P_{pz} из системы, связанной с PoF, в систему, связанную с нулевой точкой (36).

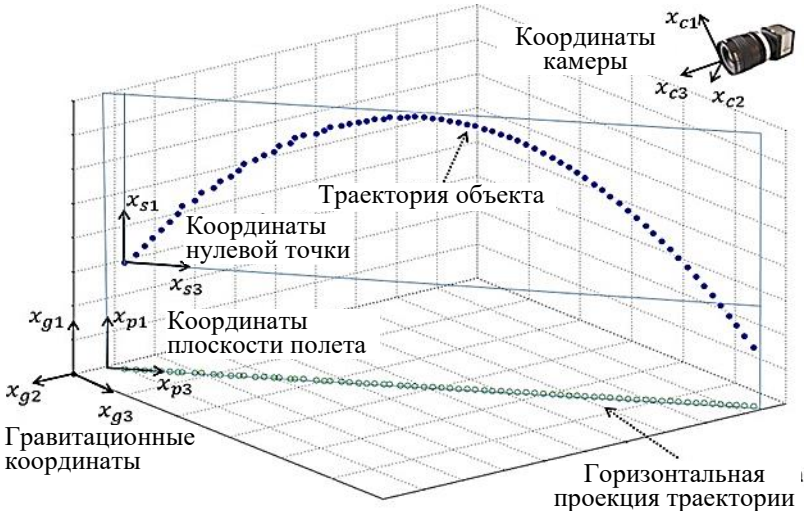

Рис. 5 Гистограмма значений α для траекторий из набора данных

Преобразования координат производятся путем умножения обратных значений этих матриц на координаты. Это означает, что обратное преобразование производится путем умножения этих матриц на координаты объекта:

$$\begin{pmatrix} x_{c1}(i) \\ x_{c2}(i) \\ x_{c3}(i) \end{pmatrix} = P_{cg}P_{gp}P_{pz}\begin{pmatrix} x_{z1}(i-z) \\ x_{z2}(i-z) \\ x_{z3}(i-z) \end{pmatrix}. \quad (37)$$

Контроллер робота использует определенную систему координат, подключенную к основанию манипулятора робота. Матрица P_{rc} соединяет эту систему с системой координат камеры и может быть получена во время дополнительной калибровки. Уравнение для преобразования конечных координат в роботизированную систему имеет следующий вид:

$$\begin{pmatrix} x_{r1}(i) \\ x_{r2}(i) \\ x_{r3}(i) \end{pmatrix} = P_{rc}P_{cg}P_{gp}P_{pz}\begin{pmatrix} x_{c1}(i-z) \\ x_{c2}(i-z) \\ x_{c3}(i-z) \end{pmatrix}, \quad (38)$$

где $\begin{pmatrix} x_{r1}(i) \\ x_{r2}(i) \\ x_{r3}(i) \end{pmatrix}$ – вектор координат объекта в системе координат, связанной с роботом. Как P_{cg}, так и P_{rc} определяются до броска. Это позволяет определить матрицу для перехода от гравитации к системе координат робота:

$$P_{rg} = P_{rc}P_{cg}\ . \quad (39)$$

Преобразование координат из системы, связанной с нулевой точкой, в систему, связанную с роботом, в этом случае будет выглядеть следующим образом:

$$\begin{pmatrix} x_{r1}(i) \\ x_{r2}(i) \\ x_{r3}(i) \end{pmatrix} = P_{rg}P_{gp}P_{pz}\begin{pmatrix} x_{c1}(i-z) \\ x_{c2}(i-z) \\ x_{c3}(i-z) \end{pmatrix}. \quad (40)$$

3. Предсказатель

Принцип kNN предложен в качестве базового метода прогнозирования траектории в [Мир24б, раздел 5.4]. В этом разделе показано, как этот метод адаптирован к этой задаче. Подраздел 3.1 описывает расчет привязки будущих положений объекта на основе соответствующих положений двух ближайших соседей, которые уже известны. В подразделе 3.2 рассматривается вопрос о том, как определить, какие траектории являются ближайшими соседями из базы данных.

3.1. Предсказатель

Рассмотрим простое применение kNN для прогнозирования временных рядов. Пусть X_C – измеренная часть текущей траектории, а набор пар ввода-вывода $L = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_m, Y_m)\}$ – набор данных предыдущих выборок траекторий. Y_c – это часть траектории, которую необходимо предсказать. Предсказатель сравнивает X_C с входными данными из набора данных и ищет пример X_s, который наиболее похож или «близок» к X_C . Конечная часть ближайшей траектории Y_s принимается за прогнозируемую часть текущей траектории. Это прогноз 1 ближайшего соседа:

$$s = \arg\min_{i=1}^{m} |X_c - X_i|, \quad Y_c = Y_s. \tag{41}$$

Вопрос о том, как определить сходство траекторий, рассматривается в следующем подразделе. Для этого подраздела предположим, что существует механизм, позволяющий проводить такое определение. Рассмотрим прогнозирование на основе двух ближайших соседей. Из набора данных берутся две траектории, наиболее похожие на текущую траекторию, и прогноз рассчитывается как среднее значение их Y:

$$Y_c = \frac{Y_1 + Y_2}{2}. \tag{42}$$

Эта операция может быть легко адаптирована к произвольному числу k ближайших траекторий, которые принимаются:

$$Y_c = \frac{1}{k}\sum_{j=1}^{k} Y_j \,. \tag{43}$$

Таким образом, простой предсказатель kNN вычисляет выходное значение как среднее значение выходных данных k примеров, взятых из набора данных, где входные данные являются ближайшими к текущему входу.

Взвешенный kNN является модифицированной версией подхода. Цель этой модификации состоит в том, чтобы учитывать не только тот факт, что «сосед находится рядом», но и числовое расстояние между текущим примером и его соседями. Выходное значение предиктора в таком случае рассчитывается по следующей формуле:

$$Y_c = \sum_{j=1}^{k} w_j Y_j. \tag{44}$$

Здесь $\sum_{j=1}^{n} w_j = 1$. Значения $\{w_1, w_2, ..., w_k\}$ определяются относительно расстояния от соответствующего соседа до текущего примера. Метод взвешивания траекторий для случая двух соседей показан ниже.

На рис. 6 показан график для трех траекторий в системе координат с нулевой точкой.

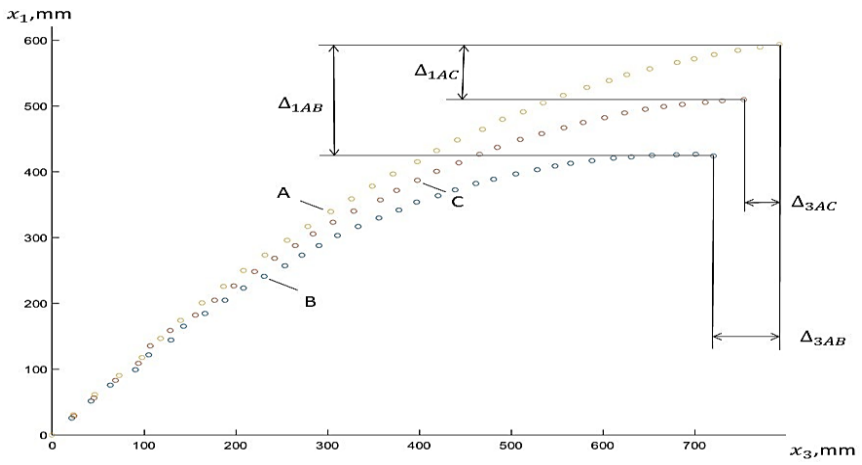

Рис. 6 Расстояния между соответствующими точками на текущей траектории *C* и траекториями *A* и *B* из набора данных

Здесь C – текущая траектория, A и B – ее ближайшие соседи, взятые из набора данных. Известна траектория области измерения X_A, X_B, траектория области захвата Y_A, Y_B для A и B и траектория области измерения X_C для C. Предполагается, что в любой момент времени объект с траектории A находится выше и дальше, чем объект с траектории C, а объект с траектории C выше и дальше, чем объект с траектории B. Другие случаи рассмотрены в подразделе 3.2 и в разделе 4.

В случае прогнозирования на основе 2 ближайших соседей, когда прогнозируется 1 координата объекта C, известны соответствующие (то есть измеренные в тот же период после броска) координаты A и B. Можно рассчитать расстояния между этими тремя точками для каждого t в обоих измерениях:

$$\begin{aligned} \Delta_{1AC}(t) &= x_{1A}(t) - x_{1C}(t), \\ \Delta_{1AB}(t) &= x_{1A}(t) - x_{1B}(t), \\ \Delta_{3AC}(t) &= x_{3A}(t) - x_{3C}(t), \\ \Delta_{3AB}(t) &= x_{3A}(t) - x_{3B}(t) \end{aligned} \tag{45}$$

пропорции

$$d_1(t) = \frac{\Delta_{1AC}(t)}{\Delta_{1AB}(t)}\ ,\quad d_2(t) = \frac{\Delta_{2AC}(t)}{\Delta_{2AB}(t)}\,. \tag{46}$$

Вектор пропорции $D(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix}$ рассчитывается для каждого доступного измерения, а затем оценивается его среднее значение $D = \{\widehat{d_x}, \widehat{d_z}\}$:

$$D = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \frac{1}{n}\sum_{t=1}^{n} \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix}. \tag{47}$$

Теперь, если известны результаты измерений для Y_1 и Y_2 и определены значения координат для Y_c, то для любого t в зоне ловли можно использовать следующую формулу

$$x_{\hat{1}C}(t) = x_{1B}(t) + \hat{d}_1\big(x_{1A}(t) - x_{1B}(t)\big),\quad x_{\hat{3}A}(t) = x_{3B}(t) + \hat{d}_2\big(x_{3A}(t) - x_{3B}(t)\big). \tag{48}$$

Это уравнение может быть преобразовано в шаблонное уравнение предсказания WKNN 2.22 следующим образом:

$$x_{\hat{1}C}(t) = \hat{d}_1 x_{1A}(t) + \big(1 - \hat{d}_1\big)\big(x_{1B}(t)\big),\quad x_{\hat{3}A}(t) = \big(1 - \hat{d}_2\big)\, x_{3A}(t) + \hat{d}_2\big(x_{3B}(t)\big). \tag{49}$$

Если вместо WKNN используется простой kNN, оба веса в этой формуле заменяются на ½:

$$x_{\hat{1}C}(t) = \frac{1}{2}x_{1A}(t) + \frac{1}{2}\big(x_{1B}(t)\big),\quad x_{\hat{3}A}(t) = \frac{1}{2}x_{3A}(t) + \frac{1}{2}\big(x_{3B}(t)\big). \tag{50}$$

Если C лежит между B и A, значения d_1 и d_2 лежат между 0 и 1. Следовательно, веса соседей также лежат между 0 и 1.

Процесс расчета будущей позиции на основе двух известных соседей с использованием уравнений (45), (46), (49) или уравнения (50) определяется здесь как операция прогнозирования k-NN (k-NN Forecasting Operation – KFO). Входные данные для KFO включают X_C , X_A, Y_A, X_B, Y_B. Оценка Y_c является результатом операции.

3.2. Поиск ближайших соседей

KFO не включает определение того, каковы траектории в L (X_A; Y_A) и (X_B; Y_B). Полная операция прогнозирования состоит из определения ближайших соседей (X_A; Y_A) и (X_B; Y_B) среди ряда траекторий в наборе данных L (операция поиска k-NN, KSO) и последующего применения KFO к ним. Сходство траекторий здесь определяется как сходство координат объекта в точках с одинаковой отметкой времени. Следовательно, евклидово расстояние между соответствующими точками показывает, как далеко траектории лежат друг от друга. Евклидово расстояние вычисляется по уравнению, которое хорошо известно из школьной геометрии:

$$d_{ij} = d_i(t) = \sqrt{(x_{1i}(t) - x_{1c}(t))^2 + (x_{3i}(t) - x_{3c}(t))^2}\ . \quad (51)$$

Здесь i – индекс траектории в наборе данных; $\begin{pmatrix} x_{1c} \\ x_{2c} \\ x_{3c} \end{pmatrix}$ – вектор координат текущей траектории; $\begin{pmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{pmatrix}$ – соответствующий вектор от траектории, которая рассматривается как возможный сосед. Значение j описано в следующем абзаце. Δi определяется как средняя евклидова разница между соответствующими точками:

$$\Delta_i = \frac{1}{n}\sum_{j=1}^{n}\left|d_{ij}\right|\ . \quad (52)$$

Здесь n – количество точек, используемых для расчета расстояния; j – индекс этих точек в списке. В качестве ближайшего соседа выбирается траектория с минимальным средним расстоянием:

$$s = \arg\left|\min_{i=1}\frac{1}{n}\sum_{t=1}^{n}|d_i(t)|\right|\ . \quad (53)$$

Здесь s – индекс ближайшего соседа в базе данных траекторий. Задача вычисления расстояния здесь проще, чем во многих других задачах классификации, регрессии и прогнозирования, где используется k-NN, поскольку все используемые измерения имеют одинаковый масштаб.

Обратите внимание, что j в уравнении (52) не равно отметке времени t измерения. Временные метки показывают количество кадров, оставшихся после запуска в момент текущего измерения. j не имеет такого значения. Точки для оценки расстояния могут быть взяты со всей доступной части траектории, но нет требования, чтобы использовались все доступные кадры. Например, если некоторые кадры потеряны наблюдателем, они не используются для расчета расстояния. Пусть список использованных кадров (UFL) представляет собой набор временных меток для кадров, которые используются при расчете расстояния. Если UFL $U = \{24, 26, 27\}$, это означает, что расстояние между траекториями будет рассчитано следующим образом:

$$\Delta_i = \frac{1}{3}(d_i(24) + d_i(26) + d_i(27)).$$

Отображение от t до j в этом случае выглядит следующим образом:

$$(t = 24) \rightarrow (j = 1),\ \ (t = 26) \rightarrow (j = 2),\ \ (t = 27) \rightarrow (j = 3).$$

Обратите внимание, что правильное сравнение расстояний между двумя парами траекторий возможно только в том случае, если UFL для расчета этих расстояний были одинаковыми. Расстояние между соответствующими точками X_c (t) и X_i (t) растет с ростом t, и для одной и той же пары траекторий в общем случае расстояние, рассчитанное с $U = \{31, 32, 33\}$, будет больше, чем расстояние, рассчитанное с $U = \{1, 2, 3\}$. Это свойство доказано экспериментами по метанию. На рис. 7 приведена гистограмма таких расстояний.

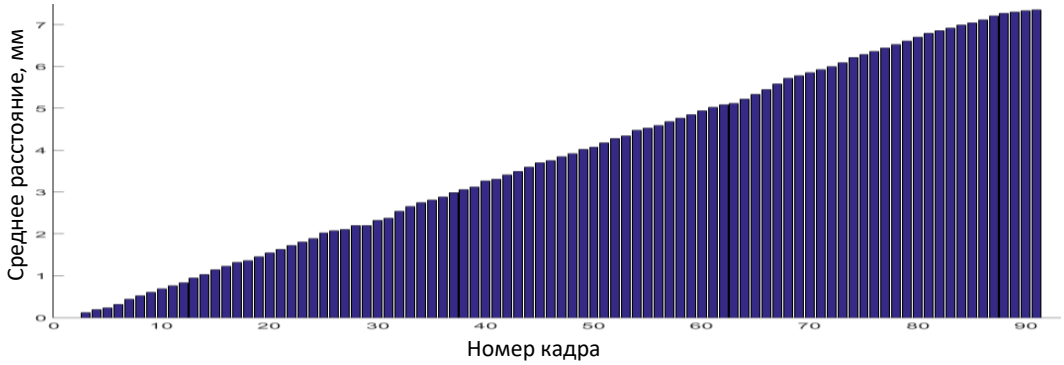

Рис. 7 Среднее расстояние между точками из базы данных в соответствии с их номером кадра

На рис. 7 представлены средние расстояния между точками с соответствующим номером кадра. Эти значения были получены на основе 111 траекторий из набора [Мир25а, разд. 1.4]. Можно видеть, что среднее расстояние растет почти линейно с увеличением числа кадров.

Какая часть доступных кадров должна быть вставлена в UFL? С одной стороны, чем больше элементов используется для поиска, тем лучше для качества оценки. С другой стороны, точность оценки пропорций растет с увеличением временных меток; расстояния становятся выше. Поэтому UFL в различных версиях алгоритма имеет следующий вид:

$$U = l,\ l+1,\ l+2, \ldots, n. \tag{54}$$

Это временная метка последнего доступного кадра в текущий момент, l – определенное число (очевидно, $l \in [k, n]$).

4. Распределение подмножества соседей

В предыдущих подразделах рассматривается оценка алгоритма при некоторых конкретных допущениях. Основные предположения заключаются в следующем:

- возможность определения ближайших соседей таким образом, чтобы текущая траектория лежала между ними в обоих измерениях в течение всей продолжительности полета;
- стабильность пропорций d_1 и d_2 из уравнения (46) во времени.

Правило о том, что текущая траектория должна лежать между двумя ближайшими соседями в обоих измерениях, может быть просто добавлено в KSO. Для этой цели поиск более высокого и более низкого ближайшего соседа выполняется отдельно. Простая эвристика для поиска заключается в следующем. Для каждой траектории X_i для каждой временной метки проверяется, находится ли объект выше объекта с текущей траектории X_c. Если $x_{1c}(t) > x_{1i}(t)$ для большинства временных меток, траектория X_i добавляется в список нижних траекторий.

Стабильность пропорций d_1 и d_2 во времени не является очевидной. Моделирование процесса показало, что в типичном случае (то есть когда из набора берутся три случайные траектории) эти пропорции нестабильны, и объект между двумя другими в начале полета может быть выше или ниже друг друга в конце (то есть траектории пересекаются, рис. 8).

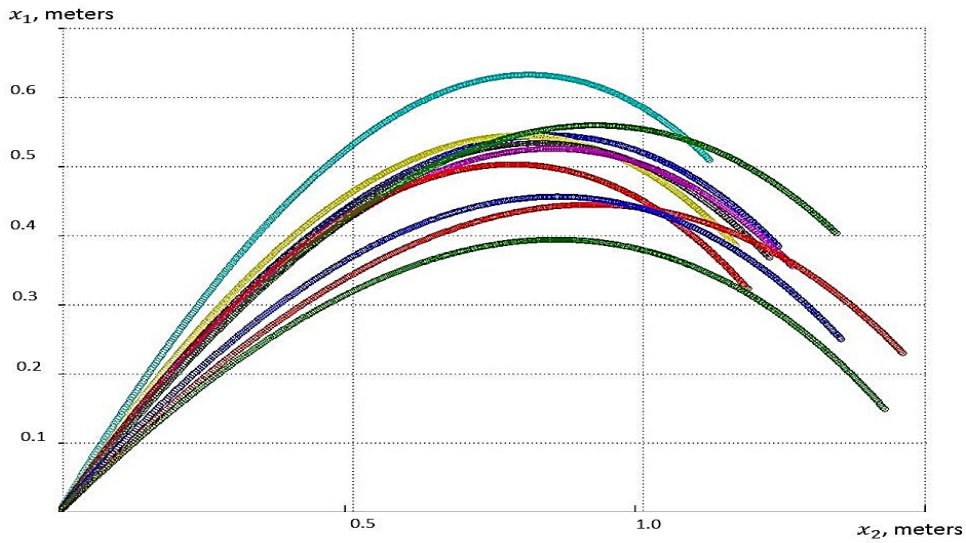

Рис. 8 Проекции траекторий в плоскости полета, случайно сгенерированные в среде моделирования. Видно, что траектории пересекаются

Различные результаты можно увидеть, когда в моделировании производится конкретная настройка параметров броска. Скорость броска может быть задана в декартовых координатах или в полярных координатах. В обеих областях скорость запуска состоит из двух параметров: значения v и высоты φ в полярной области или горизонтальной проекции v_3 и вертикальной

проекции v_1 в декартовой области. Форма траектории может быть определена обоими параметрами любой области. В типичном случае (см. рис. 8) моделирования значения параметров запуска были заданы как случайные, обычно распределенные вокруг номинальных значений. Стандартные отклонения этих параметров были установлены на одинаковое значение (0.1 м/с как для горизонтальной, так и для вертикальной составляющих скорости). Это приводит к пересечению траекторий, которые можно увидеть на рис. 8.

Можно увидеть другую ситуацию, когда стандартное отклонение для одного параметра запуска установлено намного меньшим, чем для другого. Примеры графиков для десяти траекторий, сгенерированных такой асимметричной настройкой параметров запуска, представлены на рис. 9: *а* – параметры заданы в полярной области, стандартное отклонение скорости запуска установлено на 0.01 м/с, а стандартное отклонение высоты броска установлено на 0.1 рад; *б* – параметры заданы в полярной области; стандартное отклонение для скорости запуска – 0.1 м/с, а стандартное отклонение для высоты броска – 0.01рад; *в* – параметры установлены в декартовой области; стандартное отклонение для вертикальной скорости – 0.01 м/с, а стандартное отклонение для горизонтальной скорости – 0.1 м/с; *г* – параметры установлены в декартовой области; стандартное отклонение для вертикальной скорости – 0.1 м/с, а стандартное отклонение для горизонтальной скорости – 0.01 м/с.

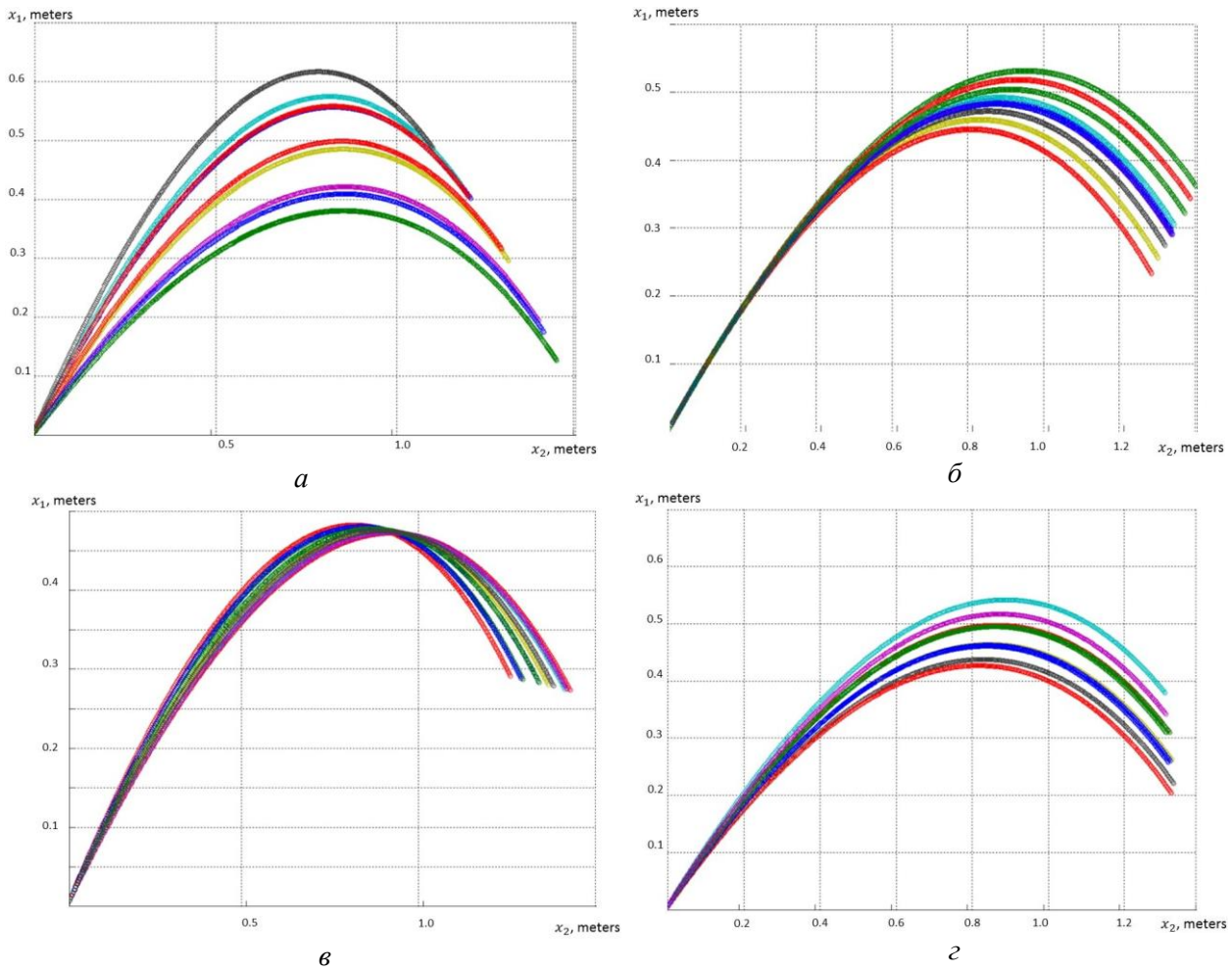

Рис. 9 Проекции плоскости полета траекторий, принадлежащих одному кластеру:
а – относительно v; *б* – относительно φ; *в* – относительно v_1; *г* – относительно v_3

Можно видеть, что эти графики имеют более однородный внешний вид, чем случайные графики на рис. 8. Траектории каждого из этих четырех графиков похожи друг на друга.

Например, последние точки траекторий на рис. 9, *г* имеют почти одинаковое значение горизонтальной координаты.

Для точности предсказателя необходимо задать следующие два вопроса: 1) Являются ли пропорции d_1 и d_2 для траекторий, сгенерированных с помощью этой настройки, стабильными во времени или нет? 2) Если одна траектория лежит между двумя другими в начале, будет ли она также лежать между ними в конце? Численные эксперименты с имитационным моделированием позволили ответить на эти вопросы со следующими результатами:

1. Скорость броска устанавливается в полярной области с высоким отклонением угла и низким отклонением значения: пропорция остается стабильной и изменяется только во втором порядке. Если в начале одна траектория находится между двумя другими в обоих измерениях, то в конце она будет находиться между обоими измерениями.

2. Скорость броска устанавливается в полярной области с низким отклонением угла и высоким отклонением значения: отклонения пропорций, а также пересечение траектории происходят в самом начале полета. После первых 60 миллисекунд пропорции становятся стабильными, и больше никаких пересечений не обнаруживается.

3. Скорость броска задается в декартовой области с низким отклонением вертикальной проекции и высоким отклонением горизонтальной проекции: пропорция нестабильна, и траектории часто пересекаются в обоих измерениях.

4. Скорость броска устанавливается в декартовой области с высоким отклонением вертикальной проекции и низким отклонением горизонтальной проекции: для вертикального измерения пропорция сохраняется стабильной, и траектории не пересекаются. В горизонтальном измерении это требование не выполняется. Однако более важно, чтобы соответствующие координаты в горизонтальном измерении были почти одинаковыми для всех траекторий.

Величины v, φ и v_l изменяются во времени, поэтому нельзя сказать, что «для этой траектории $v = 5.6$ и для этой траектории $v = 5.4$». Значение этих параметров в конкретный момент времени является единственным значением, которое может быть обсуждено. В связи с этим трудно оценить значение этих параметров на основе ошибочных измерений. С другой стороны, горизонтальная скорость не сильно меняется со временем. Среднее значение этого параметра может быть оценено в результате деления $x_3(t)$ на t.

Основными преимуществами k-NN по сравнению с другими методами, основанными на выборке (например, нейронными сетями и машинами опорных векторов), являются простота реализации и хорошая совместимость. Предсказатель принимает решения на основе конкретных примеров. Основные недостатки связаны с тем, что обучающая выборка не модифицируется до более упрощенной модели, но все они хранятся в базе данных. Когда применяется прогнозирование k-NN, необходимо сравнить текущую траекторию со всеми траекториями из набора данных. Это приводит к большим затратам памяти и времени на k-NN с огромными базами данных. Скорость обработки очень важна для задач прогнозирования снарядов, критичных по времени. Следовательно, сравнение текущей траектории с большим количеством выборок маловероятно.

Предложение этого подхода основано на сравнении текущей траектории с относительно небольшим подмножеством, а не со всеми траекториями из набора данных. Это позволяет уменьшить объем вычислений, поэтому даже большие базы данных могут быть относительно быстро обработаны. Наиболее важным аспектом этого предложения является то, что предположение о стабильности пропорций верно для траекторий внутри подмножества. Это достигается за счет того, что один из параметров скорости почти одинаков для всех траекторий в подмножестве. Другими словами, основной принцип распределения подмножества заключается в том, что текущая траектория сравнивается только с траекториями из набора данных таким образом, чтобы значение определенного параметра скорости p для этих траекторий было почти равно соответствующему параметру текущей траектории. Какой параметр следует взять за основу для такого распределения? Рассмотрим перечисленные выше параметры как потенциальный p:

1. *Значение скорости и высота в полярной области*: Оба эти параметра маловероятны, поскольку их трудно оценить. Световые барьеры обеспечивают измерение проекции скорости в направлении, перпендикулярном им, но это не значение скорости. Точное измерение значения скорости и высоты сразу после броска трудно достичь, так как точность стереопозиционирования недостаточна для этой задачи [Мир25а, разд. 3.3].

2. *Вертикальная проекция скорости*: Этот параметр маловероятен, так как он не обеспечивает стабильность пропорций.

3. *Горизонтальная проекция скорости*: Этот параметр обеспечивает пропорцию с высокой стабильностью для вертикальных координат, и его относительно легко оценить. Горизонтальная скорость существенно не меняется во времени, поэтому расчет средней горизонтальной скорости по прошествии относительно длительного периода времени может быть использован для оценки горизонтальной скорости в момент броска. Еще одним преимуществом использования горизонтальной скорости в качестве параметра сортировки является то, что все траектории имеют почти одинаковое значение x_3 в соответствующий момент времени. Поэтому, оценивая значение горизонтальной скорости, мы можем примерно оценить время, когда мяч достигнет рабочего пространства захвата. Процесс оценки горизонтальной скорости описан ниже.

Согласно определению средней горизонтальной скорости, она оценивается как результат деления горизонтальной траектории объекта на время, оставшееся после броска. В системе координат с нулевой точкой это соответствует делению координаты x_3 на время t, оставшееся после нулевой точки:

$$\widehat{v_3(t)} = \frac{x_3(t)}{t} = \frac{x_3(t)}{n\tau}\,. \tag{55}$$

Здесь τ – продолжительность межкадрового периода; n – количество оставшихся кадров.

Абсолютные погрешности при стереопозиционировании могут достигать нескольких миллиметров [Мир25а, разд. 3.3]; однако для оценки скорости относительные погрешности имеют решающее значение. Если значение x_3 равно 10 мм, а абсолютная погрешность составляет 1 мм, то относительная погрешность, очевидно, составляет около 10 %. Если абсолютная погрешность одинакова, и x_3 равно 100 мм, то относительная погрешность составит около 1 %. Этот пример показывает, что точность оценки скорости повышается с ростом значения x_3. Данная операция не должна быть выполнена слишком поздно. Системе требуется некоторое время для выполнения прогноза и выполнения движения захвата. В базовой экспериментальной установке кадры с номерами 35 по 39 использовались для расчета средней горизонтальной скорости [Mir15]. Использование этих кадров позволяет оценить горизонтальную скорость, что видно на рис. 10.

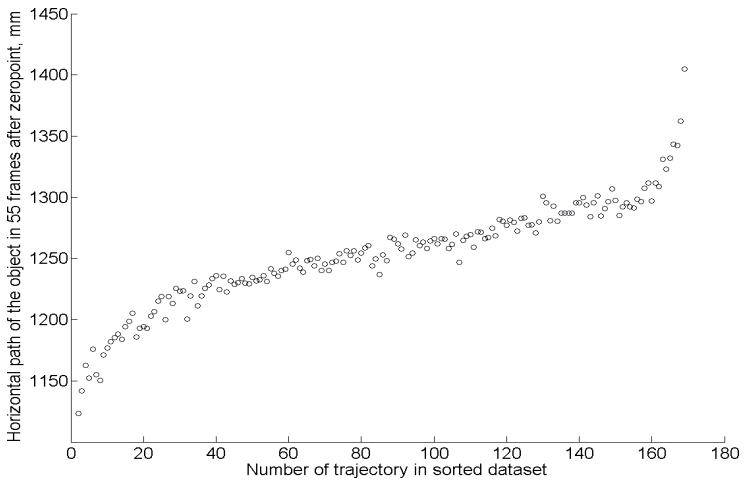

Рис. 10 Горизонтальная координата мяча через 0.5 с после броска для всех траекторий в зависимости от порядкового номера в отсортированном наборе данных [Mir15]

На этом рисунке представлены результаты сортировки базы данных траекторий по отношению к v_3. Траектории с низкими значениями v_3 были помещены в начало списка, в то время как траектории с высокими значениями были помещены в конец. На рисунке значение горизонтальной координаты для кадра с номером 55 нанесено на график относительно порядкового номера траектории в отсортированном наборе данных. Видно, что для соседних траекторий значение x_3 отличается не более чем на 2 см, по крайней мере, для траекторий с номерами от 20 до 160. Траектории в начале и в конце списка отличаются больше из-за эффекта границы. Это траектории, захваченные с исключительными значениями силы броска, и они используются только для обучения.

Операция сортировки набора данных по v_3, описанная в предыдущем разделе, применяется к базе данных траекторий на этапе обучения. Когда алгоритм предсказывает текущую траекторию, он работает с уже отсортированной базой данных. Он оценивает значение v_3 и сравнивает его только с траекториями из набора данных, которые имеют аналогичное значение v_3. Структура алгоритмов обучения и прогнозирования описана в разделе 5.

Как упоминалось выше, пропорция d_3 не поддерживается стабильной во времени для наборов данных с аналогичной горизонтальной скоростью. Однако из-за одинаковых значений $x_3(t)$ для всех элементов набора данных принцип взвешенных ближайших соседей может быть заменен простым принципом ближайшего соседа:

$$x_{\hat{C}3}(t) = \frac{x_{A3}(t)+x_{B3}(t)}{2}. \tag{56}$$

Если предположить, что значения $x_{C3}(t)$, $x_{A3}(t)$ и $x_{B3}(t)$ почти одинаковы в любой момент, а расстояние между соответствующими точками на траекториях минимально, то эта операция позволяет точно прогнозировать будущие координаты в измерениях x_3. Эксперименты показали, что простое предсказание ближайших соседей является более точным, чем взвешенные ближайшие соседи в целом.

Сравнение текущей траектории с небольшим подмножеством всего набора данных сокращает время вычислений T. Если KSO обрабатывает весь набор данных, то T упрощенным способом может быть определено как

$$T = M\tau_d + \tau_{KFO}\,, \tag{57}$$

где M – общее количество траекторий в наборе данных; τ_d – время для вычисления расстояния между двумя траекториями и его сравнения с текущим наименьшим расстоянием (то есть $n\tau_d$ – время применения KSO); τ_{KFO} – время применения KFO.

При использовании распределения подмножества оно становится равным

$$T = \tau_{\text{all}} + m\tau_d + \tau_{\text{KFO}}\,, \tag{58}$$

где τ_{all} – время выделения подмножества, а m – количество примеров в подмножестве. Очевидно, что распределение подмножеств полезно, если оно значительно сокращает временные затраты:

$$\tau_{\text{all}} + m\tau_d \ll M\tau_d\,, \tag{59}$$

$$\tau_{\text{all}} \ll (M-m)\tau_d\,. \tag{60}$$

Количество траекторий в кластере влияет на точность и скорость прогнозирования. Наилучшие результаты были достигнуты, когда размер кластера был равен примерно 1/8 всего набора данных.

5. Краткое описание предиктора

В этом разделе кратко излагается разработка алгоритмов предиктора, обсуждавшегося ранее. Фактически, предиктор состоит из двух алгоритмов: алгоритма обучения и алгоритма прогнозирования. Алгоритм прогнозирования состоит из операций, которые выполняются для

прогнозирования текущей траектории. Он вызывается после получения каждого нового кадра. Алгоритм обучения состоит из операций, выполняемых с набором данных траекторий до начала работы системы. Он направлен на сокращение вычислений, выполняемых во время прогнозирования. Поскольку прогноз выполняется в режиме реального времени, все вычисления, которые могут быть выполнены до запуска системы, выполняются до запуска системы. В этом разделе не описывается процесс извлечения 3D-координат объекта из видеопотока. Это уже исследовалось в [Мир25а]. Здесь предполагается, что предсказатель получает последовательность измеренных координат объекта в системе координат камеры с начала до последнего доступного кадра в качестве входных данных, а алгоритм обучения получает набор траекторий, наблюдаемых в системе координат камеры.

На этапе обучения решаются две основные задачи: перевести все выборочные траектории в универсальную систему координат с нулевой точкой и отсортировать их по расчетной горизонтальной скорости. Сортировка траекторий направлена на ускорение процедуры распределения кластеров на этапе прогнозирования. Структура алгоритма показана на рис. 11.

Рис. 11 Структура алгоритма обучения

Прежде всего, для каждой траектории в наборе выполняется привязка преобразований координат. Каждая траектория последовательно переводится в систему координат силы тяжести (см. раздел 2.2), систему координат плоскости полета (см. раздел 2.3) и систему координат нулевой точки (см. раздел 2.4). После построения графика траектории в системе нулевой точки оценивается значение средней горизонтальной скорости (см. раздел 4). После этого все траектории в наборе данных сортируются по средней горизонтальной скорости. В итоговой базе данных одна траектория обозначается i, v_i, X_i, где i – порядковый номер траектории в наборе, v_i – значение расчетной средней горизонтальной скорости, а X_i – матрица, показывающая динамику координат объекта во времени.

Структура предиктора показана на рис. 12. Для улучшения визуализации функциональные блоки помещаются в поля, соответствующие системе координат, в которой они используются.

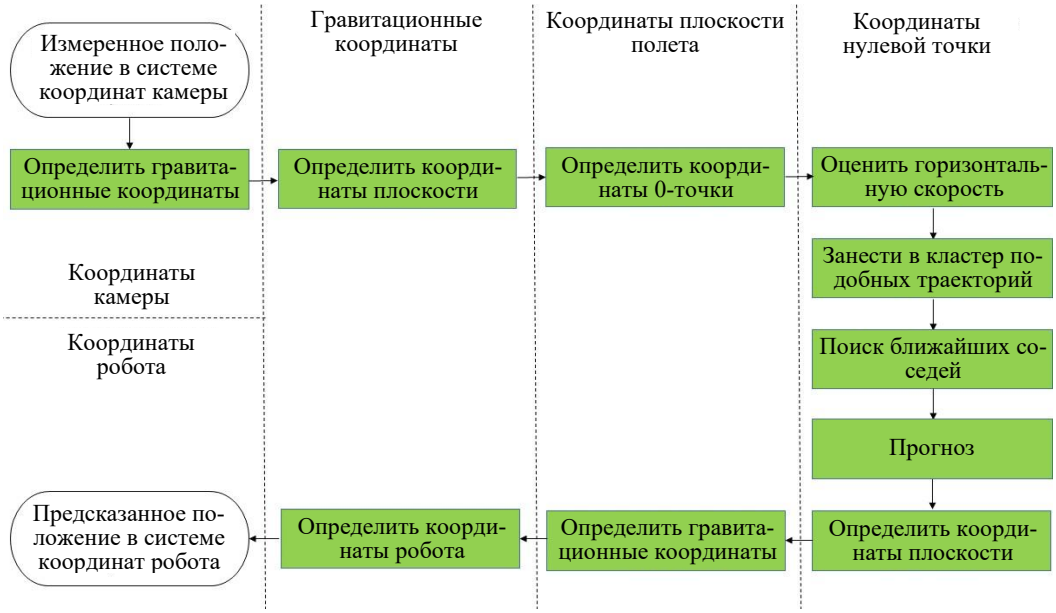

Рис. 12 Структура алгоритма прогнозирования

Прежде всего выполняется привязка преобразований координат к измеряемому участку траектории. Затем оценивается средняя горизонтальная скорость. Те же операции выполняются с каждой траекторией выборки на этапе обучения. После этого из набора данных выделяется кластер похожих траекторий. Для этого находят выборочную траекторию X_j со значением расчетной средней горизонтальной скорости, наиболее близким к соответствующему параметру текущей траектории. Затем диапазон выборочных траекторий вокруг X_j с порядковыми номерами от $j - q$ до $j + q$ помещается в кластер аналогичных траекторий. После выделения кластера выполняются операции поиска ближайших соседей внутри кластера (раздел 3.2) и прогнозирования (раздел 3.1). Вторая операция возвращает предсказанную привязку положения объекта в системе координат нулевой точки. Эта ссылка переводится обратно в гравитационную систему координат, а затем в систему координат робота (раздел 2.5). Предсказанные координаты объекта в системе координат робота выводятся на вывод предсказателя. Эти данные затем используются для определения движения захвата.

Заключение

Результатом исследования является новый алгоритм прогнозирования траектории брошенного сферического тела. Основные различия между этим алгоритмом и известными предикторами заключаются в следующем:

1. Точная физическая модель полета не требуется. Предсказатель работает исключительно на основе предыдущих примеров.

2. Метод не требует массивной параллельной обработки, мгновенной обратной связи и использования специального оборудования. Все вычисления производятся на стандартном процессорном блоке за достаточное количество времени.

3. Метод позволяет сравнивать текущую траекторию с большим количеством траекторий, хранящихся в базе данных, и все эти траектории учитываются без больших вычислительных затрат.

4. Предсказатель не зависит от положения точки запуска, горизонтального направления броска и местоположения наблюдателя.

Таким образом, доказана теоретическая применимость метода k ближайших соседей в задаче прогнозирования траектории для системы транспортировки объектов путем броска и ловли.

Благодарности

Автор выражает признательность профессору Дитмару Дитриху (Dietmar Dietrich) и коллеге Мартину Понграцу (M. Pongratz) – лидерам исследовательского проекта «Транспортировка бросанием» в Техническом университете Вены, а также профессорам Л. Р. Черняховской, В. В. Васильеву, М. Б. Гузаирову, Р. А. Мунасыпову и Г. Р. Шахмаметовой (Уфимский университет науки и технологий) за полезные замечания и поддержку.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

[Ala10] Alam F. Chowdhury, et al. A comparative study of golf ball aerodynamics // Australasian Fluid Mechanics Conference. Auckland. New Zealand. December 2010.

[Hla12] Hlawatsch F. Parameter Estimation Methods: Lecture Notes, Grafisches Zentrum HTU GmbH. Vienna. Austria. March 2012.

[Mir15] Mironov K. V., Vladimirova I. V., Pongratz M. Processing and forecasting the trajectory of a thrown object measured by the stereo vision system // IFAC-PapersOnLine. Vol. 48. No. 11. Pp. 28–25. June 2015. DOI: 10.1016/j.ifacol.2015.09.155.

[Pon15] Pongratz M., Mironov K. V. Accuracy of positioning spherical objects with stereo camera system // IEEE International Conference on Industrial Technology. Seville. Spain. Pp. 1608–1612. March 2015. DOI: 10.1109/ICIT.2015.7125326.

[Мир24а] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: обсуждение научно-технической задачи // СИИТ. 2024. Т. 6. № 1(16). С. 43–53. EDN: QGFZBW.

[Мир24б] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: обзор используемых методов // СИИТ. 2024. Т. 6. № 3(18). С. 3–48. EDN: FUUPEN.

[Мир25а] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: алгоритм прогнозирования траектории // СИИТ. 2025. Т. 7. № 2(20). С. 3–29. EDN: QGFZBW.

METADATA | МЕТАДАННЫЕ

Поступила в редакцию 26 января 2025 г.

Title: Transport-by-throwing – robotic throwing of objects: Trajectory prediction algorithm.

Abstract: Throwing is a promising method for robotic transportation of parts in flexible manufacturing systems. This paper, which is a continuation of the author's previous works, is devoted to the development of an algorithm for predicting the trajectory of a thrown object. The problem of constructing a useful model for predicting the trajectory of a thrown object is discussed. The development and initial verification of the proposed ideas are carried out using simplified motion modeling and the results of throwing experiments. A brief introduction to environmental modeling is given. The two nearest neighbors (2NN) method is used and developed for forecasting. To improve the efficiency and speed of the algorithm, coordinate transformation binding is proposed. Trajectory prediction based on the kNN method is investigated. It is also proposed to compare the current trajectory with only a small subset of the entire data set.

Key words: robotic system; transportation; transfer; capture; statistical evaluation of ballistic curves; tracking of moving objects; nearest neighbor algorithm.

Cite: Mironov K. V. **"**Transport-by-throwing – robotic throwing of objects: Trajectory prediction algorithm" // SIIT. 2025. Vol. 7, no. 4 (23), pp. 3–28. EDN TBEBLS.

Language: Russian.

Об авторе | About the author

МИРОНОВ Константин Валерьевич
Уфимский университет науки и технологий, Россия.
Доц. каф. вычислительной техники и защиты информации. Дипл. спец. по защите инф-и (Уфимск. гос. авиац. техн. ун-т, 2012). PhD (Техн. ун-т Вены, 2016). Иссл. в обл. робототехники, применения ИИ в техн. системах.
E-mail: mironovconst@gmail.com
ORCID: 0000-0002-4828-1345

MIRONOV Konstantin Valeryevich
Ufa University of Science and Technology, Russia.
Assoc. Prof., Department of Computer Science and Information Security. Information Security Specialist (Ufa State Aviation Tech. Univ., 2012). PhD (Vienna University of Technology, 2016). Research in the field of robotics and intelligent control.
E-mail: mironovconst@gmail.com
ORCID: 0000-0002-4828-1345