

УДК 004:658.5  
DOI 10.17513/snt.40251

## МЕТОД ОПРЕДЕЛЕНИЯ ТРАЕКТОРИИ РОБОТА-МАНИПУЛЯТОРА В РОБОТИЗИРОВАННЫХ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСАХ С УЧЕТОМ ВРЕМЕННЫХ ОГРАНИЧЕНИЙ

Холопов В.А., Макаров М.А., Кладов А.С.

ФГБОУ «МИРЭА – Российский технологический университет», Москва,  
e-mail: makarov\_m@mirea.ru

В статье представлен метод динамического определения траектории для промышленного робота-манипулятора, работающего в условиях динамического производственного окружения, включающего как статические объекты, так и движущиеся препятствия. Цель исследования – разработка метода определения траектории робота-манипулятора для работы в условиях временных ограничений и динамических препятствий. В рамках исследования описывается модель рабочего пространства, основанная на воксельном представлении, позволяющая оценивать доступные для манипулятора области с учетом его габаритов, кинематических ограничений и динамически меняющейся среды. Рассматривается формирование графа свободного пространства с учетом временных ограничений, где каждым узлом графа присваиваются временные метки, отражающие доступность и безопасный момент времени для нахождения манипулятора в соответствующей области. В процессе работы был разработан алгоритм поиска оптимальной траектории, адаптированный к трехмерному пространству и влиянию временных ограничений. Алгоритм позволяет при необходимости вводить режим ожидания или выбирать обходные маршруты, сохраняя при этом безопасность и избегая столкновений с движущимися объектами. Исследование имеет практическое применение и может быть использовано для переналадки роботизированных технологических комплексов в условиях многономенклатурного мелкосерийного производства. Были проведены виртуальные эксперименты в различных сценариях работы комплекса в среде Unity. Сравнительный анализ показывает, что предложенный метод обеспечивает повышение эффективности и снижение аварийных ситуаций с учетом изменений в производственной среде.

**Ключевые слова:** роботизированные технологические комплексы, робот-манипулятор, планирование траектории, временные ограничения, минимизация коллизий

## A METHOD FOR DETERMINING THE ROBOT MANIPULATOR TRAJECTORY IN ROBOTIZED TECHNOLOGICAL COMPLEXES WITH CONSIDERATION OF TIME CONSTRAINTS

Kholopov V.A., Makarov M.A., Kladov A.S.

MIREA – Russian Technological University, Moscow, e-mail: makarov\_m@mirea.ru

This article presents a method for dynamically determining the trajectory of an industrial robot manipulator operating in a dynamic production environment, which includes both static objects and moving obstacles. The purpose of the study is to develop a method for determining the trajectory of a robot manipulator for operation under time constraints and dynamic obstacles. The research describes a workspace model based on a voxel representation, enabling the assessment of accessible areas for the manipulator, considering its dimensions, kinematic constraints, and the dynamically changing environment. The formation of a free-space graph with time constraints is considered, where each node of the graph is assigned a time stamp reflecting when it is safe for the manipulator to be in that particular area. During the course of the study, a path search algorithm adapted to three-dimensional space and time constraints was developed. The algorithm allows for the introduction of waiting modes or selecting alternate routes as necessary, thereby maintaining safety and preventing collisions with moving objects. The research findings have practical applications and can be utilized for the reconfiguration of robotized technological complexes in conditions of multi-assortment, small-batch production. Virtual experiments were conducted under various scenarios using the Unity environment. A comparative analysis shows that the proposed method improves efficiency and reduces emergency situations, taking into account changes in the production environment.

**Keywords:** robotized technological complexes, robot manipulator, trajectory planning, time constraints, collision minimization

### Введение

Современное производство характеризуется высокой степенью автоматизации и широким использованием роботизированных технологических комплексов (РТК), направленных на повышение производительности и качества продукции. Робот-манипулятор в составе РТК является ключевым и наиболее подвижным элементом, выполняющим задачи перемещения заго-

товок и деталей между различными технологическими участками. В условиях многономенклатурного мелкосерийного производства особое значение приобретает синхронизация действий робота-манипулятора с остальными компонентами РТК. Одной из ключевых задач в таких условиях является определение осуществимой траектории движения робота-манипулятора с учетом временных ограничений, динамики компо-

нентов системы и возможных препятствий в рабочей зоне. С увеличением сложности производственных задач и необходимости быстрого реагирования на изменения номенклатуры и объемов выпуска продукции становится все более актуальной разработка методов, позволяющих роботу-манипулятору планировать свои действия с учетом временных ограничений системы. Недостаточная синхронизация и отсутствие учета временных ограничений могут приводить к простоям и выводу из строя оборудования, а также к снижению общей эффективности комплекса.

Существующие методы планирования траекторий часто не учитывают в полной мере динамические изменения в РТК. Это обуславливает необходимость разработки новых подходов, которые позволят интегрировать временные ограничения в процесс планирования и обеспечить синхронную работу всех компонентов РТК.

В последние годы было проведено множество исследований, направленных на решение задач планирования траектории в динамических средах. Существующие подходы можно разделить на несколько категорий согласно используемым алгоритмам и методам. Алгоритмы, основанные на выборке, такие как Rapidly-Exploring Random Tree (RRT), широко используются для планирования пути в динамических средах. Эти методы позволяют эффективно исследовать пространство состояний и находить пути обхода препятствий. В работе [1] предложено улучшение традиционного алгоритма RRT для повышения эффективности и вероятности успешного планирования в присутствии динамических препятствий. Также используются гибридные методы, комбинирующие выборку с учетом конфигураций и ограничений области выборки, что позволяет улучшить планирование в узких пространствах [2]. Другой популярный подход основан на использовании дорожных карт (roadmaps). В исследованиях [3, 4] предлагается метод планирования движения на основе заранее вычисленной дорожной карты статической части окружения с учетом динамических препятствий в реальном времени.

Для построения траекторий используются методы, оптимизирующие определенные критерии, такие как: время, энергопотребление и плавность движения. В работах [5, 6] планирование траектории формулируется как задача оптимизации с учетом кинематических и динамических ограничений манипуляторов. Оптимизация позволяет получать гладкие и оптимальные по времени траектории, но часто не учи-

тывает динамические изменения в окружении или необходимость синхронизации с внешними процессами. Также существуют методы, направленные на использование избыточности манипуляторов для удовлетворения множественных ограничений. В исследовании [7] предложена стратегия управления-планирования, которая комбинирует управление и планирование для корректировки конфигураций манипулятора в реальном времени, эффективно обрабатывая ограничения.

Реактивные методы ориентированы на быструю адаптацию к изменениям в окружении на основе данных, полученных от датчиков. В работе [8] представлен реактивный подход к планированию пути для манипуляторов в динамических средах в реальном времени, позволяющий роботам реагировать в реальном времени на появление препятствий и корректировать траекторию «на лету». Адаптивные методы включают глобальное планирование с локальной адаптацией. В исследовании [9] предложен алгоритм, который использует потенциальные поля и оптимизационные техники для адаптации траектории манипулятора в динамической среде, оптимизируя ее в реальном времени.

Важным критерием планирования траекторий является обеспечение безопасности в условиях взаимодействия «человек – робот». В работе [10] разработан алгоритм предотвращения столкновений в реальном времени для промышленных манипуляторов, вводящий инвариантное безопасное множество для определения множества безопасных состояний. Оптимизационный подход позволяет минимизировать отклонение от желаемой траектории при обеспечении безопасности. В исследовании [11] представлен подход, основанный на использовании систем машинного зрения для активного предотвращения столкновений в условиях совместной работы человека и робота. Описанная в статье система динамически корректирует поведение робота на основе данных с датчиков, обеспечивая безопасность операторов.

Некоторые исследования сосредоточены на использовании симуляций для планирования траекторий. В работе [12] демонстрируется, как моделирование может помочь в планировании траекторий с учетом динамики и кинематики робота. Подобные методы предназначены для первоначального планирования, но они не используются для работы в динамически изменяющихся средах. Несмотря на разнообразие методов планирования траекторий для роботизированных манипуляторов, большинство

из них не учитывают интеграцию временных ограничений и синхронизацию с другими компонентами системы. Реактивные и адаптивные методы обеспечивают быструю реакцию на изменения в окружении, но часто не рассматривают необходимость выполнения действий в заданные временные интервалы или координации с другими элементами РТК. Оптимизационные методы позволяют получать траектории, оптимальные по времени или другим критериям, но обычно предполагают статичность окружения и не учитывают динамические препятствия. Методы, основанные на дорожных картах и выборке, эффективны для поиска путей в сложных пространственных конфигурациях, однако также могут не учитывать синхронизацию с другими компонентами систем, что критично при работе роботов-манипуляторов в составе РТК.

С учетом выявленных недостатков существующих алгоритмов и методов возникает необходимость в разработке метода планирования траектории робота-манипулятора, который интегрирует временные ограничения и обеспечивает синхронизацию с другими компонентами РТК в динамических средах. Такой метод должен обеспечивать реальное время планирования и перепланирования траекторий с учетом предсказуемых и непредсказуемых динамических препятствий.

**Целью исследования** является разработка метода определения траектории робота-манипулятора для работы в условиях временных ограничений и динамических препятствий.

#### Материалы и методы исследования

Предлагаемый метод основан на интеграции алгоритмов планирования траекторий с учетом временных ограничений, данных от системы машинного зрения

и информации о состоянии оборудования. Он предполагает создание актуальной модели рабочего пространства, включающей статические и динамические объекты, применение адаптированного алгоритма  $A^*$ , учитывающего временные ограничения и позволяющего планировать траектории в реальном времени с минимизацией риска коллизий посредством обмена информацией о текущем состоянии компонентов РТК и планах работы. В реальных условиях в РТК робот-манипулятор при планировании и перепланировании траекторий должен учитывать не только статическую геометрию рабочей зоны, но и временные ограничения, возникающие по разным причинам. Такие ограничения можно разделить на 4 категории.

1. Временные окна доступности оборудования.

2. Синхронизация с подвижными компонентами РТК.

3. Техническое обслуживание и перерывы.

4. Задержки и непредвиденные ситуации.

Перечисленные категории включают в себя циклы работы элементов РТК, когда они недоступны для взаимодействия с роботом-манипулятором (категория 1), учитывают работу подвижных компонентов РТК (категория 2) и технические перерывы (категория 3), а также внеплановые ситуации, требующие динамического планирования траекторий (категория 4).

Для обеспечения точного взаимодействия робота-манипулятора с объектами и компонентами РТК используется единая система координат. Пространство рабочего окружения представляется в виде регулярной трехмерной сетки (воксельной модели). Каждый воксель характеризуется координатами и состоянием (свободен или занят). Кубический воксель описывается множеством точек:

$$\left\{ (x, y, z, s) \mid a - \frac{s}{2} \leq x \leq a + \frac{s}{2}, b - \frac{s}{2} \leq y \leq b + \frac{s}{2}, c - \frac{s}{2} \leq z \leq c + \frac{s}{2} \right\}, \quad (1)$$

где  $(a, b, c)$  – координаты центра вокселя,  $s$  – длина стороны вокселя.

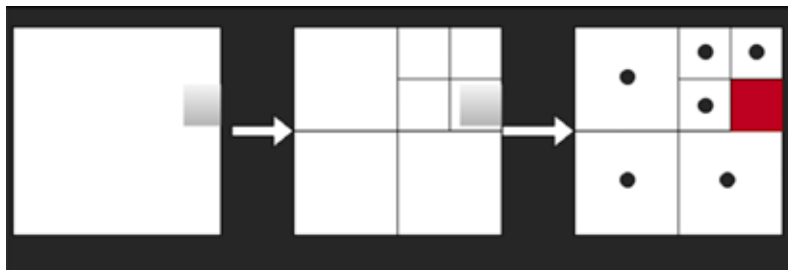


Рис. 1. Локальное уточнение

Изначально сетка формируется с некоторым базовым разрешением, где часть вокселей может быть занята (пересекаться с геометрией загруженной модели окружения). На занятых вокселях выполняется локальное уточнение сетки путем деления на ячейки меньшего размера, вплоть до достижения дочерними ячейками размера установленной точности (предела уточнения). Рекурсивному делению подлежат только те ячейки, которые после очередной итерации уточнения оказались занятыми (рис. 1). После достижения предела уточнения ячейки, оставшиеся занятыми, удаляются из сетки.

После выполнения алгоритма создания рабочего пространства полученная совокупность ячеек принимается как неравномерная воксельная структура, определяющая «безопасную зону». Под «безопасной зоной» понимается множество свободных ячеек, в которых робот при учете своих габаритов и кинематических ограничений может находиться без коллизий. Однако авторами отмечается, что предлагаемая «безопасная зона» – это аппроксимация, поскольку воксельная дискретизация неизбежно вносит упрощения. Поэтому некоторые потенциально достижимые состояния могут быть исключены из дальнейших вычислений алгоритмом. Для компенсации этого метода предполагается динамическое перепланирование и учет времени ожидания.

Для решения задачи поиска пути формируется граф свободного пространства, где узлами являются центры свободных ячеек. Ребра графа задают возможные перемещения между соседними узлами. Соседство между узлами определяется исходя из их геометрической близости, как правило, с учетом направления движения робота-манипулятора. Для неоднородной воксельной сетки соседство определяется между ячейками одинакового уровня разбиения или с учетом перехода между уровнями уточнения, при этом критерий соседства

может быть также статически задан с учетом евклидовой метрики или адаптирован к конкретным условиям.

Чтобы учесть временные ограничения и избежать коллизий, каждому узлу графа присваивают дополнительную временную метку доступности узла  $t(n)$ , отражающую момент времени, когда робот может безопасно находиться в данном узле. Для статически всегда свободных узлов  $t(n)$  может быть фиксированной или неограниченной по времени. Для узлов, доступных лишь в определенные промежутки времени (например, из-за движущихся препятствий),  $t(n)$  обновляется при перепланировании. Если в узле появилось препятствие, узел не удаляется из графа, обновляются данные о его доступности – теперь робот может находиться там лишь после определенного времени ожидания. Таким образом, динамические объекты трактуются как временные ограничения на узлы графа.

Для нахождения оптимальной траектории робота-манипулятора используется алгоритм поиска пути  $A^*$ , адаптированный для работы в трехмерном пространстве и учитывающий временные ограничения. Алгоритм  $A^*$  сочетает стоимость достижения текущего узла  $g(n)$  и эвристическую оценку оставшейся стоимости  $h(n)$ :

$$f(n) = g(n) + h(n) + w(n), \quad (2)$$

где  $f(n)$  – общая оценка стоимости пути через узел  $n$ ;

$g(n)$  – накопленная стоимость пути до узла  $n$  с учетом времени прохождения и расстояния;

$h(n)$  – эвристическая оценка оставшейся стоимости пути до цели;

$w(n)$  – дополнительная стоимость, отражающая недоступность узла  $n$  в момент прихода робота.

В качестве эвристической функции  $h(n)$  используется евклидова метрика, вычисляющая расстояние между текущим и целевым узлами:

$$h(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2 + (z_n - z_{goal})^2}, \quad (3)$$

где  $(x_n, y_n, z_n)$  – координаты текущего узла,  $(x_{goal}, y_{goal}, z_{goal})$  – координаты целевого узла.

Функция  $w(n)$  вводится для учета временных параметров и может быть определена следующим образом:

$$w(n) = \begin{cases} \infty, & \text{если узел } n \text{ недоступен в момент } t(n), \\ 0, & \text{в противном случае.} \end{cases} \quad (4)$$

где  $t(n) = t_{start} + g(n)$ , где  $t_{start}$  – время начала движения робота.

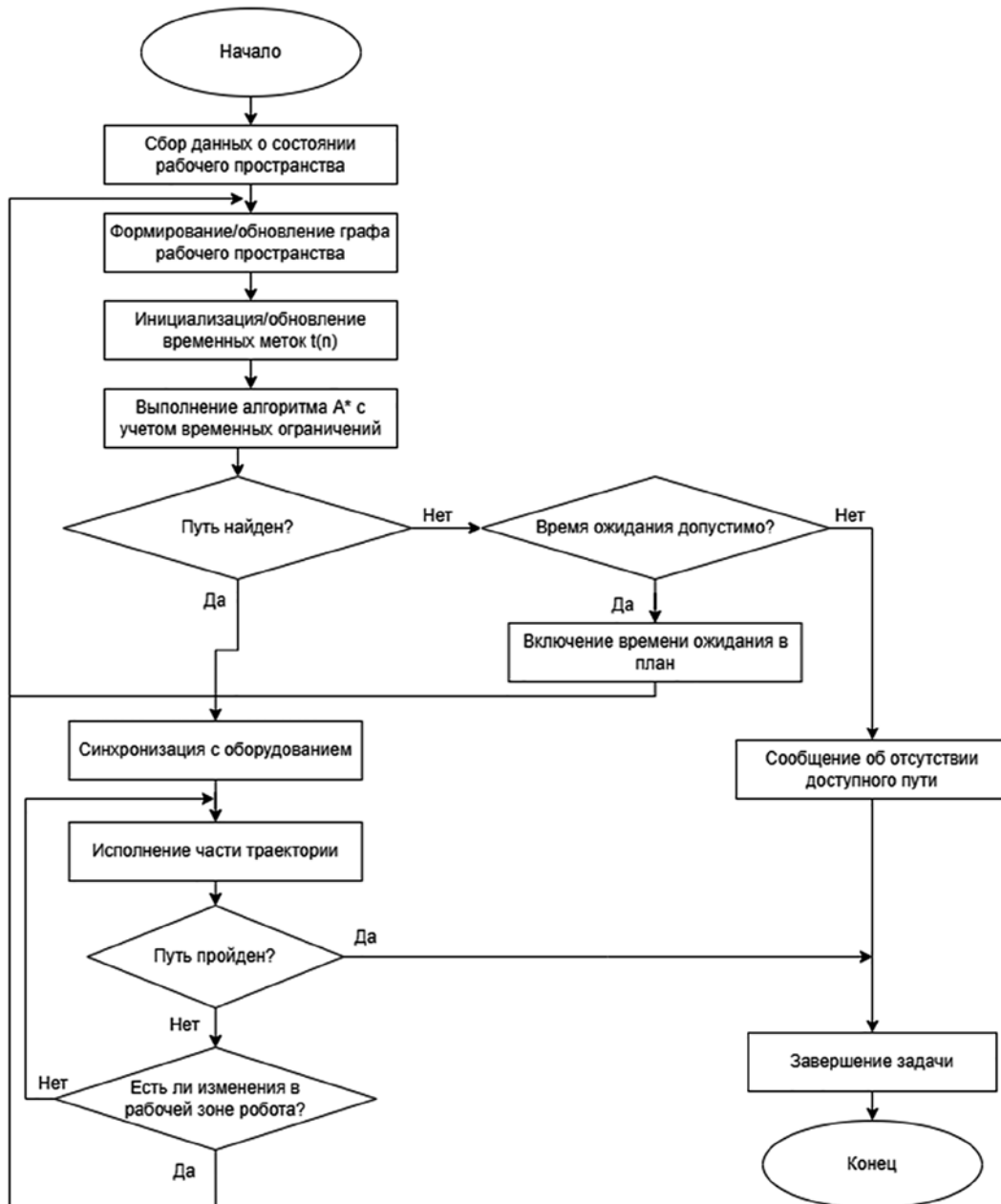


Рис.2. Алгоритм работы метода планирования траектории робота-манипулятора

Это означает, что, если узел  $n$  недоступен в момент времени  $t(n)$  из-за занятости другим объектом или временных ограничений, стоимость перехода через него становится бесконечной, и алгоритм избегает этого пути. Если же узел станет доступен только спустя некоторое время, алгоритм учтет возможность ожидания в безопасном положении до момента, когда узел станет доступен. Если ожидание превышает допустимые пределы или ситуация блокирует путь навсегда, алгоритм сигнализирует об отсутствии решения. В случаях, когда целевой узел или путь к нему становятся до-

ступными только спустя определенное время, алгоритм предусматривает возможность ожидания. Вместо немедленного возврата ошибки при недоступности пути алгоритм включает в план траекторию с учетом времени ожидания в узлах, где это необходимо. Робот может задержаться в безопасном положении до момента, когда путь станет доступным. Если текущее место ожидания в перспективе станет небезопасным, алгоритм предпримет попытку найти другую точку для ожидания или альтернативный путь. Если недоступность узла или пути является постоянной или время ожидания

превышает допустимые пределы, алгоритм сообщает об отсутствии доступного пути к целевому узлу, запрашивая вмешательство оператора.

Для обеспечения согласованной работы с другими компонентами РТК реализуется обмен информацией. Система машинного зрения непрерывно отслеживает изменения в рабочем пространстве и обновляет данные о положении динамических объектов в режиме реального времени. Робот-манипулятор получает данные о состоянии оборудования, такие как:

- статус станка с ЧПУ (занят, свободен, время завершения текущей операции);
- состояние конвейерной линии, включая положение и скорость движения заготовок;
- плановые операции и техническое обслуживание, влияющие на доступность оборудования;
- данные от систем машинного зрения о положении и движении динамических объектов.

Эта информация используется для обновления временных меток  $t(n)$  узлов графа и позволяет роботу корректировать свой план движения в соответствии с текущей ситуацией. При появлении новых препятствий или изменении состояния оборудования происходит обновление графа и, при необходимости, перепланирование траектории.

На рисунке 2 представлен обобщенный вид алгоритма планирования траектории робота-манипулятора.

Для проверки и демонстрации эффективности предложенного метода была разработана имитационная модель в среде Unity.

Выбор Unity обусловлен ее возможностями по визуализации, гибкости разработки и наличию встроенного физического движка, позволяющего симулировать динамику системы с высокой степенью реалистичности.

Экспериментальная модель представляет собой виртуальный РТК, включающий следующие компоненты:

- робот-манипулятор с шестью степенями свободы, способный выполнять задачи перемещения заготовок и деталей в трехмерном пространстве;
- два станка с числовым программным управлением (ЧПУ), расположенных в рабочей зоне робота и выполняющих обработку деталей;
- конвейерная линия для подачи и отвода заготовок, синхронизированная с работой робота и станков;
- система машинного зрения, обеспечивающая получение актуальной информации о положении объектов и состоянии рабочей зоны.

В модели реализован адаптированный алгоритм поиска пути  $A^*$ . Алгоритм работы метода планирования траектории робота-манипулятора (рис. 2) учитывает временные метки  $t(n)$  узлов, а также эвристическую оценку расстояния до цели. При обнаружении недоступности узлов алгоритм может рассмотреть ожидание, если это допустимо. В противном случае будет проанализирована возможность перепланирования или вывода сообщения о недоступности пути. Основные функции, необходимые для реализации метода по заданному алгоритму, представлены на листинге (листинг).

```
// Инициализация графа свободного пространства
Graph graph = BuildFreeSpaceGraph(voxelGrid);
// Инициализация временных меток узлов
foreach (Node node in graph.Nodes)
{
    node.TimeStamp = GetNodeAvailabilityTime(node);
}
// Функция поиска оптимального пути
List<Node> FindOptimalPath(Node startNode, Node goalNode)
{
    // Очередь с приоритетом для открытых узлов
    PriorityQueue<Node> openSet = new PriorityQueue<Node>();
    HashSet<Node> closedSet = new HashSet<Node>();

    startNode.G = 0;
    startNode.F = Heuristic(startNode, goalNode);
    openSet.Enqueue(startNode);

    while (openSet.Count > 0)
    {
        Node current = openSet.Dequeue();
        if (current == goalNode)
            return ReconstructPath(current);

        closedSet.Add(current);

        foreach (Node neighbor in current.Neighbors)
        {
```

```

        if (closedSet.Contains(neighbor))
            continue;

        float tentativeG = current.G + GetTransitionCost(current, neighbor);

        // Проверка временных ограничений и доступности узла
        float arrivalTime = t_start + tentativeG;
        if (arrivalTime < neighbor.TimeStamp)
        {
            // Узел недоступен в момент прибытия
            // Возможно, стоит рассмотреть ожидание или пропустить узел
            if (!CanWaitAtNode(current, neighbor))
                continue;
            // Добавить ожидание, если это допустимо
            tentativeG += ComputWaitingTime(arrivalTime, neighbor.
TimeStamp);
        }

        if (tentativeG < neighbor.G)
        {
            neighbor.Previous = current;
            neighbor.G = tentativeG;
            neighbor.F = neighbor.G + Heuristic(neighbor, goalNode) +
GetTimePenalty(neighbor);
            if (!openSet.Contains(neighbor))
                openSet.Enqueue(neighbor);
        }
    }
}

return null; // Путь не найден или превышено время ожидания
}
// Функция эвристики
float Heuristic(Node node, Node goal)
{
    return Vector3.Distance(node.Position, goal.Position);
}
// Стоимость перехода между соседними узлами
float GetTransitionCost(Node fromNode, Node toNode)
{
    return Vector3.Distance(fromNode.Position, toNode.Position) / robotSpeed;
}
// Штраф за временные ограничения
float GetTimePenalty(Node node)
{
    if (IsNodeUnavailable(node, node.G))
        return float.PositiveInfinity;
    else
        return 0;
}
// Проверка недоступности узла в момент прибытия
bool IsNodeUnavailable(Node node, float arrivalTime)
{
    return arrivalTime < node.TimeStamp;
}

```

Листинг. Реализация алгоритма метода планирования траектории на языке C#

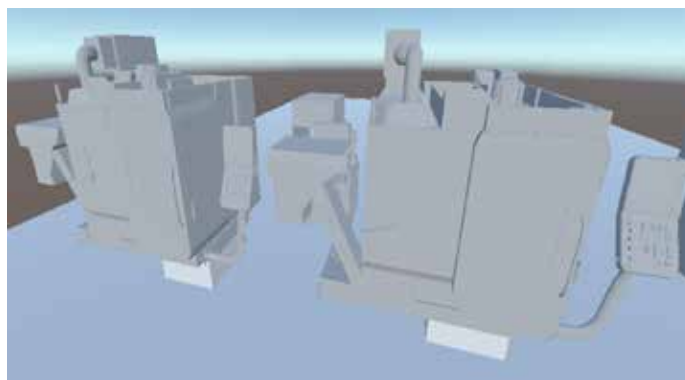


Рис.3. Моделирование рабочего пространства РТК на движке Unity

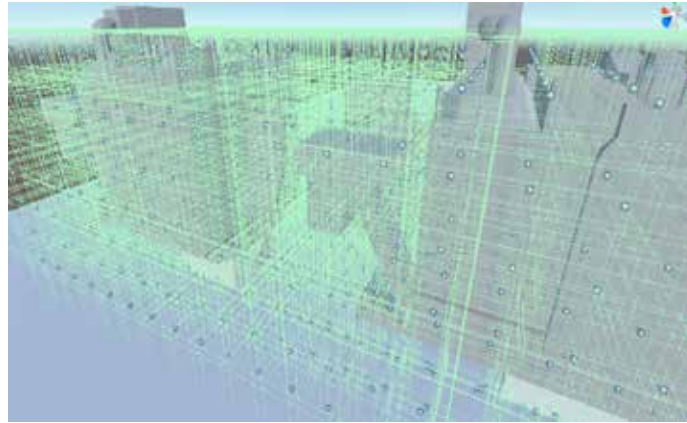


Рис. 4. Формирование регулярной воксельной сетки на модели РТК

Моделирование рабочего пространства представлено на рисунке 3, а визуализация формирования графов – на виртуальной модели на рисунке 4.

#### Результаты исследования и их обсуждение

Для апробации предложенного алгоритма были проведены виртуальные эксперименты с использованием разработанной модели РТК. Целью экспериментов было оценить эффективность предложенного метода по следующим показателям:

- время цикла обработки одной заготовки с момента начала операции до ее завершения роботом;
- число зарегистрированных коллизий или потенциально опасных ситуаций, при которых расстояние между роботом и препятствием снижалось ниже установленного порога, требующего немедленного вмешательства в управляющее действие;
- время простоя оборудования, определяемого как процент от общего времени, в течение которого робот или другие компоненты РТК не участвуют в выполнении задач;
- общая производительность системы, вычисляемая как количество обработанных заготовок или объем выполненных операций за единицу времени.

В состав РТК входили: робот-манипулятор с шестью степенями свободы, два станка с ЧПУ, расположенные в рабочей зоне робота, а также конвейерная линия для подачи и отвода заготовок. Система машинного зрения обеспечивала получение актуальной информации о положении динамических объектов, включая перемещения других роботов или периодическое открытие/закрытие дверей станков.

Было определено два экспериментальных сценария для предлагаемого метода планирования траектории робота-манипу-

лятора, отражающих различные производственные ситуации.

В первом сценарии рабочее пространство и оборудование оставались неизменными, без непредвиденных ситуаций. Временные метки узлов были статическими, поскольку отсутствовали динамические препятствия. В таких условиях предложенный метод совпадал с классическим А\* по структуре планирования, но благодаря учету базовых временных параметров, таких как время достижения целей и возможность ожидания в узлах, можно было незначительно улучшить согласованность работы робота-манипулятора с оборудованием. Время цикла обработки заготовки в среднем сократилось примерно на 3% по сравнению с базовым решением. Легкая адаптация временных меток позволила уменьшить время простоя оборудования и полностью избежать потенциальных коллизий, которые могли возникать при неточном согласовании с операционным расписанием оборудования.

Во втором сценарии в рабочей зоне периодически появлялись дополнительные динамические объекты, например открывающаяся и закрывающаяся дверь станка или другой работающий робот-манипулятор. В таких условиях предлагаемый метод учитывал динамические обновления временных меток и при необходимости вводил ожидание либо выбирал обходной путь, чтобы избежать столкновений. Такой подход позволил сократить время цикла на 5%, при этом полностью исключив коллизии. Для базового А\* в этом сценарии фиксировалось в среднем 2 ситуации, когда расстояние между роботом и препятствием становилось критически малым, что приводило к аварийной остановке. В результате снизился и уровень простоя оборудования за счет более эффективной адаптации робота к изменениям. Результаты приведены в таблице.



## Сравнение результатов классического и адаптированного A\* в двух сценариях

Показатель	Сценарий 1 (статичный)	Сценарий 2 (динамический)
Время цикла без учета временных параметров, с	120	140
Время цикла с учетом временных параметров, с	116	133
Сокращение времени цикла, %	3	5
Количество опасных ситуаций (классический A*)	0	2
Количество опасных ситуаций (с использованием предлагаемого метода)	0	0
Время простоя оборудования без учета временных параметров, %	15	18
Время простоя оборудования с учетом временных параметров, %	12	13

### Заключение

Проведенные исследования и экспериментальные результаты подтвердили эффективность предложенного метода определения траектории робота-манипулятора с учетом временных ограничений и динамики среды. Метод позволяет синхронизировать действия робота с другими компонентами РТК, минимизировать коллизии и повысить общую производительность комплекса. Разработанная модель в среде Unity продемонстрировала способность метода адаптироваться к различным производственным ситуациям, включая появление динамических препятствий и изменение состояния оборудования. В дальнейшем планируются проведение испытаний на реальном оборудовании для подтверждения результатов моделирования, а также исследование возможности применения методов машинного обучения для улучшения прогнозирования состояния системы и дальнейшего повышения адаптивности и эффективности метода.

### Список литературы

1. Wei Kun, Ren Bingyin. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm // *Sensors*. 2018. Vol. 18. Is. 2. DOI: 10.3390/s18020571.
2. Chen Hanlin, Zang Xizhe, Zhu Yanhe, Zhao Jie. Hybrid Sampling-Based Path Planning for Mobile Manipulators Performing Pick and Place Tasks in Narrow Spaces // *Applied Sciences*. 2024. Vol. 14. Is. 22. DOI: 10.3390/app142210313.
3. MohananM.G., Salgaonkar A. Robotic Motion Planning in Dynamic Environments and its Applications // *International Journal of Robotics and Control Systems*. 2022. Vol. 2. Is. 4. P. 666-691. DOI: 10.31763/ijrcs.v2i4.816.
4. Cefalo Massimo, Oriolo Giuseppe. A general framework for task-constrained motion planning with moving obstacles // *Robotica*. 2019. Vol. 37. Is. 3. P. 575–598. DOI: 10.1017/S0263574718001182.
5. Pham Hung, Pham Quang-Cuong. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. // *arXiv: 1707.07239*. 2017. DOI: 10.48550/arXiv.1707.07239.
6. Duan Honggang, Z. Rongmin, Yu Fei, Gao Jun, Chen Yuan. Optimal Trajectory Planning for Glass-Handling Robot Based on Execution Time Acceleration and Jerk // *Journal of Robotics*. 2016. Vol. 2016. Is. 1. P. 9329131. DOI: 10.1155/2016/9329131.
7. Barbalata C., Vasudevan R., Johnson-Roberson M. A constrained control-planning strategy for redundant manipulators. // *arXiv:1810.03945*. 2018. DOI: 10.48550/arXiv.1810.03945.
8. Sandakalum Thushara, Ang Marcelo H. Motion Planning for Mobile Manipulators—A Systematic Review // *Machines*. 2022. Vol. 10. Is. 2. DOI: 10.3390/machines10020097.
9. Jonathan Michaux, Patrick Holmes, Bohao Zhang, Che Chen, Baiyue Wang, Shrey Sahgal, Tiancheng Zhang, Sidhartha Dey, Shreyas Kousik, Ram Vasudevan. Can't Touch This: Real-Time, Safe Motion Planning and Control for Manipulators Under Uncertainty // *arXiv: 2301.13308v2*. 2023. DOI: 10.48550/arXiv.2301.13308.
10. Bedada, Wendwosen Palli, Gianluca. Real Time Collision Avoidance with GPU-Computed Distance Maps // *arXiv: 2407.02363v1*. 2024. DOI: 10.48550/arXiv.2407.02363.
11. Roni-Jussi Halme, Minna Lanz, Joni Kämäräinen, Roel Pieters, Jyrki Latokartano, Antti Hietanen. Review of vision-based safety systems for human-robot collaboration // *Procedia CIRP*. 2018. Vol. 72. P. 111-116. DOI: 10.1016/j.procir.2018.03.043.
12. Yang F., Wang C., Cadena C., Hutter M. iPlanner: Imperative Path Planning. // *arXiv: 2302.11434*. 2023. DOI: 10.48550/arXiv.2302.11434.