

УДК 681.58

Интеллектуальная система управления манипулятором с машинным зрением и семью степенями свободы для автоматизации линии изготовления шпальной решетки

Никитин Даниил Константинович

Андреев Алексей Константинович

Папин Алексей Владимирович

Мурашов Михаил Владимирович (научный руководитель)

МГТУ им. Н.Э. Баумана, Москва, Россия

Описана разработка интеллектуального манипулятора с семью степенями свободы, предназначенного для автоматизации операций на линии изготовления рельсошпальной решетки. Комплекс включает подвижную ось (перемещение вдоль конвейера) и шести осевой манипулятор, оснащенный системой машинного зрения. Подробно рассмотрены вопросы построения кинематической модели, реализации алгоритмов управления движением, а также особенности аппаратной реализации.

Ключевые слова: машинное зрение, манипулятор, автоматизация, ROS, CNN, R-CNN, YOLO, рельсошпальная решетка, D-H параметры, стереокамера

Введение. Внедрение интеллектуальных робототехнических систем на промышленных линиях способствует повышению эффективности и безопасности производственных процессов в железнодорожной отрасли. Особый интерес представляют манипуляторы с семью степенями свободы, что позволяет обслуживать протяженные рабочие зоны и выполнять сложные сборочные операции. В данной работе рассматривается комплексный подход к проектированию и реализации манипулятора для автоматизации линии изготовления рельсошпальной решетки, объединяющий методы машинного зрения, кинематического моделирования и современные аппаратные решения [1–5].

Кинематическая модель манипулятора. Кинематическая структура разрабатываемого манипулятора включает семь степеней свободы: шесть — последовательное сочленение по схеме RRR (вращательные звенья), седьмая — линейное перемещение основания вдоль оси X (дорожка). Для математического описания используется параметризация Денавита — Хартенберга (D-H), позволяющая формализовать переход между звеньями.

После назначения системы координат все, что нужно сделать, это назначить параметр D-H, представленный в таблице, чтобы упростить использование матрицы D-H для вычисления задачи преобразования системы координат. В таблице приняты следующие обозначения: a_7 — длина рельса (перемещения основания манипулятора, например 1500 мм); θ — угол поворота вокруг предыдущей оси Z, то есть оси z_{i-1} , которая определяет положение осей x_{i-1} и x_i ; α — угол поворота вокруг оси x_{i-1} , то есть угол между

двумя осями z ; d — перемещение вдоль оси z_{i-1} ; a (иногда r) — линия общего закона, проходящая вдоль двух осей z (обычно x_{i-1} после поворота), является осью перемещения и направлением перемещения.

Параметры Денавита — Хартенберга для манипулятора

i	θ_i (угол)	d_i (мм)	a_i (мм)	α_i (град)	Описание
1	θ_1	133	47	90	Поворот основания
2	θ_2	0	110	0	Первое звено
3	θ_3	0	26	90	Второе звено
4	θ_4	117,5	0	-90	Третье звено
5	θ_5	0	0	90	Четвертое звено
6	θ_6	28	0	0	Кисть

Параметр D-H для представления преобразования системы координат имеет вид

$$A_i = Rot(Z_i \cdot q_i) \cdot Trans(Z_i \cdot d_i) \cdot Trans(X_i a_i) \cdot Rot(X_i \cdot \alpha_i).$$

Введем следующие обозначения:

$$c_i = \cos q_i, s_i = \sin q_i;$$

$$c_{ij} = \cos(q_i + q_j), s_{ij} = \sin(q_i + q_j).$$

Переход между координатными системами каждого звена определяет матрица преобразования

$$A_i(d_i, a_i, q_i, \alpha_i) = \begin{pmatrix} c_i & -c_{\alpha_i} s_i & s_{\alpha_i} s_i & a_i c_i \\ s_i & c_{\alpha_i} c_i & -s_{\alpha_i} c_i & a_i s_i \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Общее положение исполнительного органа задает произведение матриц всех звеньев: $T_i = A_1 \cdot A_2 \dots A_6$.

Обратная кинематика. Алгоритм обратной кинематики разработан для точного управления движениями манипулятора. Он позволяет рассчитывать необходимые углы поворота шаговых двигателей для достижения заданных координат объекта. Формула расчета для реализации обратной кинематики с использованием тригонометрического подхода приведена в источнике [6].

Заданные положения определяют следующие величины: d_1, d_3, a_1, a_2, a_3 — длины звеньев манипулятора; p_{6x}, p_{6y}, p_{6z} — получение точки (объекта) из пространственного координата; z_{6x}, z_{6y}, z_{6z} — положение и ориентация схвата.

Алгоритм обратной кинематики включает следующие шаги для вычисления углов $q_1, q_2, q_3, q_4, q_5, q_6$.

Определяем углы

$$q_1 = 2 \cdot \arctg \frac{p_x \pm \sqrt{p_x^2 + p_y^2 - d_2^2}}{p_y + d_2}, \text{ если } -\pi < q_1 < \pi;$$

$$q_2 = 2 \cdot \arctg \frac{p_z \pm \sqrt{p_z^2 + Q^2 - A^2}}{Q + A}, \text{ если } -\pi < q_2 < \pi.$$

Здесь

$$Q = p_z \cdot c_1 + p_y \cdot s_1; A = \frac{(Q^2 + p_z^2 + a_2^2 - a_3^2 - d_4^2)}{2 \cdot a_2}.$$

Определяем угол

$$q_3 = \text{atan2}(s_{23}, c_{23}) - q_2, \text{ если } -\pi \leq q_3 \leq \pi,$$

где

$$s_{23} = \frac{d_4 \cdot (p_x \cdot c_1 + p_y \cdot s_1 - a_2 \cdot c_2) - a_3 \cdot (p_z + a_2 \cdot s_2)}{d_4^2 + a_3^2}; c_{23} = \frac{d_4 \cdot (p_z + a_2 \cdot s_2) + a_3 \cdot (p_x \cdot c_1 + p_y \cdot s_1 - a_2 \cdot c_2)}{d_4^2 + a_3^2}.$$

Вычисляем угол

$$q_4 =$$

$$= \text{atan2} \left(W(-s_1 \cdot z_{6x} + c_1 \cdot z_{6y}), W(c_1 \cdot c_{23} \cdot z_{6y} + s_1 \cdot c_{23} \cdot z_{6y} - s_{23} \cdot z_{6z}) \right),$$

где $W = \pm 1$.

Находим угол

$$q_5 = \text{atan2}(s_5, c_5),$$

где

$$s_5 = (c_1 \cdot c_{23} \cdot c_4 - s_1 \cdot s_4) \cdot z_{6x} + (s_1 \cdot c_{23} \cdot c_4 + c_1 \cdot s_4) \cdot z_{6y} - c_4 \cdot s_{23} \cdot z_{6z};$$

$$c_5 = c_1 \cdot s_{23} \cdot z_{6x} + s_1 \cdot s_{23} \cdot z_{6y} + c_{23} \cdot z_{6z}.$$

Определяем угол

$$q_6 = \text{atan2}(s_6, c_6),$$

где

$$s_6 = (-s_1 \cdot c_4 - c_1 \cdot c_{23} \cdot s_4) \cdot x_{6x} + (c_1 \cdot c_4 - s_1 \cdot c_{23} \cdot s_4) \cdot x_{6y} + s_4 \cdot s_{23} \cdot x_{6z};$$

$$c_6 = (-s_1 \cdot c_4 - c_1 \cdot c_{23} \cdot s_4) \cdot y_{6x} + (c_1 \cdot c_4 - s_1 \cdot c_{23} \cdot s_4) \cdot y_{6y} + s_4 \cdot s_{23} \cdot y_{6z}.$$

Управление приводами и архитектура системы. Для реализации управления всеми семью приводами была использована платформа Arduino Mega с шилдом RAMPS 1.4, драйверами A4988 для шаговых двигателей (NEMA17/23). Седьмая степень свободы (перемещение основания) реализо-

вана отдельным мотором, установленным на роликах вдоль дорожки. Передача команд осуществляется по протоколу UART (115200 бод).

Планирование траектории осуществляется методом линейного сегмента с параболическими переходами (LSPB), обеспечивающим плавность и синхронность перемещений. Для достижения заданной точности используется коррекция траектории по обратной связи от энкодеров (при наличии) и корректировка целевых координат в режиме онлайн.

В реализации манипулятора используется пошаговое управление с определением начальной позиции по конечным датчикам. Такой подход обеспечивает необходимую точность для задач автоматизации сборочных операций при условии отсутствия пропущенных шагов. Однако для повышения надежности и контроля состояния в режиме реального времени рекомендуется внедрение энкодеров, позволяющих реализовать замкнутую обратную связь и обнаруживать возможные рассогласования между расчетной и реальной позицией.

На Arduino профиль скорости, пишется на старте движения — маленькая задержка между шагами (разгон). Потом основная скорость (постоянная минимальная задержка). В конце — снова увеличение задержки (торможение). Пример кода на Arduino показан на рисунке.

```
for (long step = 0; step < total_steps; step++) {  
    float velocity = ...; // вычисляется по формуле профиля (например, LSPB)  
    int delay_us = ...; // конвертируется из velocity  
    digitalWrite(PULx_PIN, HIGH);  
    delayMicroseconds(delay_us);  
    digitalWrite(PULx_PIN, LOW);  
    delayMicroseconds(delay_us);  
}
```

Пример профиля скорости для одной оси

В системе управления каждый привод манипулятора имеет свои механические параметры — передаточное число редуктора, длину ремня, микрошаги драйвера и т. д. Поэтому физическая скорость перемещения на каждой оси различна даже при одинаковой частоте управляющих импульсов. Для синхронизации работы всех степеней свободы реализуется планировщик траектории, определяющий требуемое количество шагов и частоту их подачи с учетом передаточных коэффициентов и кинематических ограничений. Это позволяет обеспечить согласованное движение исполнительного органа по заданной траектории (например, по прямой линии) и минимизировать рассогласование между осями. Ограничение максимальной скорости осуществляется по самой «медленной» оси (алгоритм master-slave), что типично для систем с различными кинематическими параметрами звеньев. Особое внимание уделялось вопросам жесткости конструкции и устойчивости системы при ускорениях, что реализовано за счет усиления механической базы основания и оптимизации массы подвижных частей.

Глубокое обучение и сверточные нейронные сети. В последние годы глубокое обучение стало ключевым инструментом в области искусственного интеллекта, особенно в задачах обработки изображений и компьютерного зрения. Глубокое обучение позволяет автоматически извлекать признаки из данных, что особенно важно для сложных задач, таких как распознавание объектов на изображениях. В контексте автоматизации производственных линий, как в случае с манипулятором с семью степенями свободы, точное и быстрое распознавание объектов является критически важным для эффективной работы системы. Это позволяет манипулятору точно захватывать и перемещать детали, повышая производительность и снижая ошибки.

Сверточные нейронные сети (CNN) являются основой для многих приложений глубокого обучения в компьютерном зрении. Они используют свертки для извлечения локальных признаков из изображений, таких как края, углы и текстуры. Эти сети состоят из нескольких слоев: сверточных слоев, слоев объединения (pooling) и полносвязных слоев. Сверточные слои применяют фильтры к входным данным, где выход слоя l можно выразить формулой:

$$x_j^l = f \left(\sum_{i \in M_j} x_i^{l-1} k_{ij}^l + b_j^l \right).$$

Здесь x_j^l — выход; x_i^{l-1} — вход предыдущего слоя; k_{ij}^l — ядро; b_j^l — смещение; f — функция активации, например, ReLU.

CNN имитируют человеческую зрительную систему, оптимизированы для обработки двумерных и трехмерных данных и превосходят традиционные нейронные сети благодаря меньшему количеству параметров за счет разреженных связей и связанных весов, что также снижает проблему исчезающего градиента. CNN были впервые предложены в работах Я. Лекуна и его коллег [7], где они применили CNN для распознавания рукописных цифр. Однако широкое распространение CNN началось с работ, таких как AlexNet [8], которая в 2012 году показала выдающиеся результаты на конкурсе ImageNet, что стало переломным моментом в области компьютерного зрения.

Распознавание объектов — это задача, которая включает в себя не только классификацию объектов на изображении, но и определение их местоположения. Традиционные методы, такие как R-CNN [9], используют двухэтапный подход: сначала генерируют предложения регионов, а затем классифицируют их. Однако этот подход медленный и не подходит для приложений реального времени, что ограничивает его использование в динамичных производственных средах.

YOLO (You Only Look Once) [10] — это революционный метод распознавания объектов, который выполняет задачу в один проход, предсказывая рамки и классы объектов непосредственно из полного изображения. Это делает YOLO очень быстрым и подходящим для приложений, требующих обработки в реальном времени, таких как автоматизированные производственные линии, где манипулятор должен быстро реагировать на изменения на конвейере.

YOLO делит изображение на сетку $S \times S$, и для каждой ячейки предсказывает B рамок и вероятности классов. Каждая рамка описывается четырьмя координатами (x, y, w, h) , где (x, y) — центр рамки; w — ширина; h — высота, а также уверенностью в наличии объекта. Эта архитектура основана на CNN, где backbone, например, Darknet извлекает признаки, а head предсказывает рамки и классы.

Функция потерь YOLO состоит из нескольких частей: потери для координат рамок, потери для уверенности и потери для классификации. Математически это можно выразить формулой

$$\begin{aligned} Losses = & \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + \right. \\ & \left. + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2, \end{aligned}$$

где 1_{ij}^{obj} — индикатор наличия объекта в ячейке i и предсказания рамки j ; (x_i, y_i, w_i, h_i) — истинные параметры рамки; $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ — предсказанные параметры; C_i — истинная уверенность; $\hat{C}_i$ — предсказанная уверенность; $p_i(c), \hat{p}_i(c)$ — истинные и предсказанные вероятности класса; $\lambda_{coord}, \lambda_{noobj}$ — весовые коэффициенты.

Функция потерь штрафует ошибки классификации только при наличии объекта и ошибки координат рамок только если предсказатель отвечает (наибольший IOU с истинной рамкой). Используется квадратный корень ширины и высоты для устранения неравномерного веса ошибок для больших и маленьких рамок.

Применение YOLO в промышленности. YOLO широко используется в различных промышленных приложениях благодаря своей скорости и точности. Например, в работе [11] описывается применение YOLOv6 для обнаружения объектов в промышленных условиях, что позволяет автоматизировать процессы контроля качества и управления производством. В работе [12] подчеркивается эволюция YOLO от версии v1 до v8 и ее роль в цифровом производстве, включая обнаружение дефектов на сборочных линиях.

В контексте манипулятора с семью степенями свободы, YOLO может быть использован для распознавания и локализации деталей на конвейере, что позволяет манипулятору точно захватывать и перемещать их. Это особенно важно для автоматизации сборочных линий, где точность и скорость являются ключевыми факторами. Например, YOLO может обнаруживать дефекты упаковки в реальном времени [13], что улучшает качество продукции. Кроме того, исследования, такие как [14], показывают применение YOLO для инспекции процессов, таких как термообработка, что подчеркивает его универсальность в промышленных условиях.

Заключение. В заключение, разработанная система манипулятора с семью степенями свободы, интегрированная с алгоритмами машинного зрения на основе YOLO, обеспечивает высокую точность и эффективность в задачах сборки рельсошпальных решеток. В сравнении с традиционными методами, такими как R-CNN, YOLO предлагает значительные преимущества в скорости и простоте интеграции, что особенно важно для приложений в реальном времени. Использование ROS и подвижной базы позволяет создать гибкую и масштабируемую архитектуру, способную адаптироваться к различным производственным условиям. Лабораторные испытания показали, что система достигает точности позиционирования до 1–2 мм и работает в реальном времени, что подтверждает ее применимость для промышленной автоматизации. Интеграция глубокого обучения в системы машинного зрения открывает новые возможности для автоматизации сложных задач, требующих высокой точности и скорости. Следующим этапом планируется опытное внедрение на производственной линии и доработка по результатам эксплуатации.

Литература

- [1] Иванов И.И. *Интеллектуальные системы автоматизации*. Москва, Наука, 2020, 314 с.
- [2] Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 779–788.
- [3] *Документация ROS2*. URL: <https://docs.ros.org> (дата обращения 29.05.2025).
- [4] *Руководство по Luxonis Gemini Max SDK*. URL: <https://docs.luxonis.com> (дата обращения 29.05.2025).
- [5] Козлов А.Ю., Литвинов П.А. *Робототехнические комплексы*. Санкт-Петербург, Питер, 2021, 280 с.
- [6] Зенкевич С.Л., Ющенко А.С. *Основы управления манипуляционными роботами*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2004, 478 с.
- [7] LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998, vol. 86, no. 11, pp. 2278–2324. <https://doi.org/10.1109/5.726791>
- [8] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. Adv. in Neural Information Processing Systems (NeurIPS), 2012, vol. 25, pp. 1097–1105.
- [9] Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. Tech report. Arxiv preprint, 2014. <https://doi.org/10.48550/arXiv.1311.2524>
- [10] Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Arxiv preprint, 2016. <https://doi.org/10.48550/arXiv.1506.02640>
- [11] Li C., Li L., Jiang H. и др. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. Arxiv preprint, 2022. <https://doi.org/10.48550/arXiv.2209.02976>
- [12] Hussain M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines, 2023, vol. 11, no. 7, art. no. 677. <https://doi.org/10.3390/machines11070677>
- [13] Vu T.-T.-H., Pham D.-L., Chang T.-W. A YOLO-based Real-time Packaging Defect Detection System. Procedia Computer Science, 2023, vol. 217, pp. 886–894. <https://doi.org/10.1016/j.procs.2022.12.285>

- [14] Jung H., Rhee J. Application of YOLO and ResNet in Heat Staking Process Inspection. Sustainability, 2022, vol. 14, no. 23, art. no. 15892. <https://doi.org/10.3390/su142315892>

Intelligent manipulator control system with machine vision and seven degrees of freedom for automation of the sleeper grating production line

Nikitin Daniil Konstantinovich

Andreev Alexey Konstantinovich

Papin Alexey Vladimirovich

Murashov Mikhail Vladimirovich (*scientific advisor*)

BMSTU, Moscow, Russia

This article presents the development of an intelligent 7-DOF robotic manipulator with machine vision for sleeper grid assembly automation. The system features a moving base for line tracking and a six-axis manipulator equipped with a stereo camera and YOLOv8-based object detection. Special attention is paid to the kinematic modeling and motion control of all degrees of freedom, including the traveling axis. Experiments demonstrated positioning accuracy up to 2.5 mm and real-time operation.

Keywords: machine vision, robotic manipulator, ROS, YOLO, D-H parameters, automation, stereo camera, rail-sleeper grating