

# Алгоритм обучения систем с дискретным управлением

Голиков К.А.

Разработан алгоритм обучения для задачи позиционирования систем с дискретным управлением, основанный на методе обобщения проб и ошибок, сохраняющихся в базе данных, с помощью глобальной интерполяции и градиентного спуска. Оптимизация алгоритма производится по критерию сокращения времени обучения (числа попыток). Алгоритм был протестирован на симуляторе для моделей систем, действующих на плоскости, двух разных типов: для мобильного робота с двумя ведущими гусеницами и для открытой кинематической цепи с вращательными и призматическими сочленениями.

**Ключевые слова:** позиционирование, алгоритм обучения, робот, интерполяция, аппроксимация.

Система работает в дискретном времени  $t$ . В каждый момент времени система находится в одном из своих состояний:

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

Состояние системы можно разбить на две составляющие: *статическое состояние*  $x(t)$ , известно с достаточной точностью, и *динамическое состояние*  $\dot{x}(t)$  – неизвестно.

В качестве наглядного изображения системы с дискретным управлением рассматриваем модели роботов разных конструкций, оперирующих на плоскости  $(x, y)$ .

*Статическое состояние системы* может быть описано позициями  $n$  точек на плоскости  $x(t) = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)$  и их массами  $m = (m_1, \dots, m_n)$ ,  $m_k = \text{const}$ ,  $k = \overline{1, n}$ . Кроме того, в системе имеются *приводы*, которые действуют на точки системы и обеспечивают изменение состояния системы во времени – движения системы. В позициях  $(n-1)$  точек системы закреплены по два привода, назовём эти точки *сочленениями*, а соответствующие сочленению два привода – *парой приводов-антагонистов сочленения*. Оставшаяся одна точка будет той, которой

нужно научиться управлять, она не имеет приводов, назовём её *особой точкой* робота (например, представляет собой центр захвата манипулятора или центр мобильного робота). Таким образом, в системе имеется  $m=2(n-1)$  приводов, ими можно управлять,  $(n-1)$  сочленение с координатами  $((x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1}))$  и особая точка  $(x_n, y_n)$ .

На каждый привод независимо можно подавать *дискретное управление*  $u_k(t) \in \{0, 1\}$ ,  $k = \overline{1, m}$ , которое означает, действует ли привод или выключен в текущий момент. Существует специальный вход  $reset \in \{0, 1\}$ , который мгновенно возвращает систему в известное *начальное положение* (абсолютно точно) из любого её состояния. Тогда общее управление системы записывается в виде:

$$u(t) = (u_1(t), u_2(t), \dots, u_m(t), reset) \in \{0, 1\}^{m+1}$$

*Динамическое состояние системы* – это скорости  $\mathbf{n}$  точек системы  $\dot{x}(t)$ , которые слагаются из нескольких составляющих: усилий приводов и остальных сил, действующих на робота, таких как инерция, кориолисовы/центробежные силы, гравитация и другие вне-модельные силы.

Пробой, попыткой, полным действием или *эпизодом обучения* будет фиксирование наблюдаемых состояний системы  $s(t)$  в заданный промежуток времени  $t_0 < t < t_1$  от известного начального состояния  $s_0$  в некоторое конечное  $s(t_1) = (x(t_1), 0)$ , при чём конечное состояние должно быть с нулевой динамической частью, т.е. установившимся (после конца эпизода предполагается использовать кнопку *reset*).

Данные эпизода формируют одну запись в *базе данных* –  $t_0 < t < t_1$

1. управление  $u(t)$
2. траектория движения особой точки системы  $x(t)$
3. смещения сочленений  $q(t)$
4. приближение динамики  $\dot{q}(t), \dot{x}(t)$

## ПОСТАНОВКА ЗАДАЧИ

Итак, есть система

$$s(t) = (x(t), \dot{x}(t)), \quad (s(t), u(t)) \rightarrow s(t+1)$$

в каждый момент  $t$  на её входы подаются *управляющие сигналы*  $u(t)$ , которые задаются бинарной кусочно-постоянной вектор-функцией, включают приводы, которые изменяют состояние системы  $s(t)$ . Частично

определить характеристики состояния системы в каждый момент  $t$  можно при помощи статической части – данных *обратной связи*  $x(t)$  – вещественнозначной вектор-функции, представляющей собой координаты точек в глобальной неподвижной системе отсчёта. Умышленно отказываемся от возможности аналитически построить точную или приближённую физическую модель системы, полагаемся только на опыты и обратную связь – это принципиальное ограничение нашего исследования.

#### ЗАДАЧИ:

1. За минимальное время произвести ценные действия с помощью приводов системы, отражающие возможности и принципы функционирования этой системы. Получить управления, хорошо подходящие для обобщения и предсказания поведения системы для решения поставленных задач.

*В рамках задачи позиционирования:* определить равномерную плотную решётку целевых точек, получить за минимальное время множество движений робота, покрывающих конечными установившимися положениями особой точки системы все целевые точки из начального положения с нужной точностью, отразить накопленный опыт в базе данных движений наиболее полным образом.

2. Выработать подход на основе интерполяции и градиентного спуска, позволяющий системе с накопленной базой действий производить новые действия, помещающие конечное положение особой точки системы в *недостигнутые ранее цели* с заданной точностью в области рабочего пространства, плотно исследованной на прошлом этапе. *Получить аппроксимацию функции*, отображающую принцип функционирования системы с учётом неизвестной части состояния системы по сохранённым в базе данных действиям.

*В рамках задачи позиционирования:* аппроксимацию функции обратной кинематики с учётом неизвестной динамики.

3. Обеспечить *в условиях изменяющегося динамического функционирования* системы (подмены действующих сил) *повторяемость* с заданной точностью ранее достигнутых целевых точек. В реальном времени обеспечить *коррекцию действия*, минимизирующую конечное отклонение от достигнутой ранее цели в плотно изученной области действия робота. Разработать быстрый метод обновления аппроксимации при получении новых данных, учитывающий уменьшение актуальности старого опыта.

Обучение системы отрабатывается в виртуальной среде, для задачи позиционирования роботов двух разных конструкций со своими особенностями. Эксперименты проводились с моделями реальных роботов с количеством приводов  $m$  от 4 до 10. Число переключений активных приводов  $k$  в управлениях эпизодов обучения находилось в пределах 4-8, а максимальная длительность эпизода  $N=10000$  тактов. Попытки делаем не очень длительные. Предполагаем, что если в  $u(t)$  общее число последовательностей единиц не велико, не превышает натурального  $k$  ( $k \ll N$ ), перемещение особой точки получится не очень сложным<sup>1</sup>.

Рассмотрим свойства первой задачи исследования: самым дорогим ресурсом при обучении робота является время реализации попытки, попыток нужно делать минимальное число, но при этом они должны быть максимально полезными для обучения. Это сделать сложно, потому что в неизученной области результат попытки и вид получаемого действия неизвестен.

*Цели позиционирования* при обучении на первом этапе выбираются произвольным равномерным образом в интересующей области, в которой будут ставиться задачи позиционирования при эксплуатации механизма. Целевая точка является центром небольшой окрестности, в соответствии с заданной наперед точностью позиционирования, если конечное установившееся положение особой точки системы находится в этой окрестности, говорим, что робот *попал в цель*. Пробуя попадать в конкретные цели, система в итоге учится попадать в любые точки.

Решением второй задачи данного исследования разработки алгоритма обучения будут два метода аппроксимации: *глобальная интерполяция случайными функциями 2-функционала* динамической части состояния системы и *приближение попытками к целевой точке от ближайшей известной в направлении градиента*. С результатами экспериментов можно ознакомиться в приведённых таблицах в конце данной работы.

Говорим, что система *повторила* действие, если конечное установившееся положение особой точки системы оказалось в допустимой окрестности целевой точки, вне зависимости от траектории, которую эта точка реально описала.

Для третьей задачи исследования сформулируем математическое представление изучаемых изменений и шумовых погрешностей в динамическом состоянии системы при эксплуатации робота для обновления алгоритма в будущем.

---

<sup>1</sup>Энерго-эффективные траектории – когда от положения покоя отклоняются только те сочленения, которые нужны, чтобы попасть в цель

## МОДЕЛЬ РОБОТА МАНИПУЛЯТОРА

1. *Особая точка* – центр захвата манипулятора. Критерий достижения цели – захват достиг цели (без учёта ориентации).
2. *Привод* – смыкающее либо размыкающее действие вращательного сочленения или выдвигающее либо сдвигающее действие призматического сочленения.
3.  $n = 3-5$  точек системы с массами  $m_1-m_n$ , 2-4 сочленения,  $m = 4-8$  приводов.

Модель движения идеализирована:

1. Без проскальзываний в сочленениях
  2. Без упругих связей и звеньев
  3. Без изгибания осей суставов
- и т.д.

Будем рассматривать плоские манипуляторы, т.е. захват может двигаться только на плоскости  $(x, y)$ .

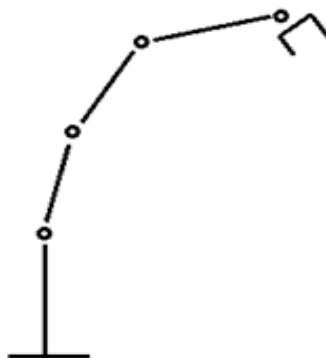


Рис. 1. Модель манипулятора

## ОСОБЕННОСТИ КИНЕМАТИКИ МАНИПУЛЯТОРОВ

Кинематика – геометрия движения, ветвь классической механики, описывает движение точек, тел и систем тел без рассмотрения причин их движения. В области кинематики роботов изучают взаимосвязь между конфигурациями сочленений  $q$  и точками декартового *оперативного* (рабочего) пространства  $x$ , пространства захвата, его возможные положения и ориентации. *Пространство сочленений*  $q = (q_1, \dots, q_n)$  – набор смещений в сочленениях различной структуры.

$$q \in S_q,$$

где  $S_q$  – разрешённые конфигурации, в реальных манипуляторах допускаются не все положения суставов, в механизме твёрдые элементы не могут проходить сквозь друг друга.

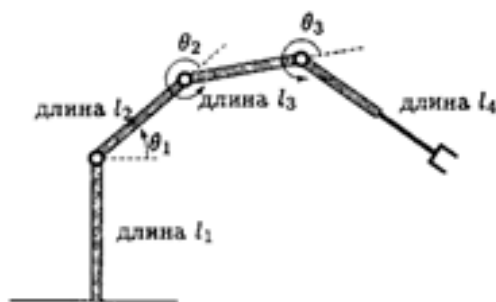


Рис. 2. Определение внутренних координат сочленений

Рассмотрим рисунок 2, начало координат рабочего пространства робота расположено в первом подвижном сочленении у основания кинематической цепи, углы вычисляются против часовой стрелки, начиная от условного продолжения предыдущего звена, а самый первый угол – начиная от оси  $x$  глобальных координат.

Задача прямой кинематики: при заданных положениях сочленений  $q$  определить положение центра захвата  $x$  в рабочем пространстве.

$$x = T q \tag{1}$$

Общая матрица прямого преобразования  $T$  из  $q$  в  $x$  имеет вид:

$$T = \left[ \begin{array}{c|c} {}^n_1R & {}^n_1P \\ \hline 0 & 1 \end{array} \right], \quad (2)$$

где  ${}^n_1R$  – матрица вращения ( $2 \times 2$ , а в 3D-пространстве  $3 \times 3$ ), задаёт ориентацию координат при переходе между системами отсчёта ближайших сочленений, здесь все повороты от захвата  $q^n$  к базе  $q^1$ ,  ${}^n_1P$  – столбец переноса положения начала системы отсчёта следующего сочленения относительно предыдущего, в нашем случае, учтены все смещения сочленений. Данный столбец даёт нам в общем виде формулы перевода из координат сочленений манипулятора в глобальные координаты положения захвата:

$${}^n_1P = f^n(l, \theta) = \begin{pmatrix} \sum_{i=1}^{n-1} l_{i+1} \cos(\sum_{j=1}^i \theta_j) \\ \sum_{i=1}^{n-1} l_{i+1} \sin(\sum_{j=1}^i \theta_j) \\ \sum_{i=1}^n \theta_i \end{pmatrix} \begin{matrix} -x \\ -y \\ \text{—ориентация} \end{matrix} \quad (3)$$

Параметры:

1. **Углы**  $\theta = (\theta_1, \dots, \theta_n)$  – поворотные соединения в порядке от основания к захвату (не все сочленения могут быть поворотными).  $\theta_1$  – ориентация сочленения у основания,  $\theta_n$  – ориентация захвата. Углы могут иметь строгие ограничения –  $\Theta_{k1} < \theta_k \leq \Theta_{k2}$ , диапазон измерения.
2. **Длины**  $= (l_{k1}, \dots, l_{kz})$  – некоторые сочленения могут быть призматическими и менять свою длину  $l_k$  в диапазоне от  $l_{k1}$  до  $l_{k2}$ .

Если функция  $f^n(l, \theta)$  даёт положение захвата и его ориентацию в глобальных координатах, то функции  $f^i(l, \theta), i = \overline{1, n}$  – положения промежуточных  $i$ -х точек-сочленений в глобальных координатах.

Т.е. формулы  $f^n(l, \theta)$  достаточно, чтобы управлять захватом манипулятора относительно координат сочленений. На практике обычно цели манипуляторам ставятся в оперативном пространстве, в координатах положения захвата. В этом случае приходится решать более сложную задачу обратной кинематики манипулятора.

## ОБРАТНАЯ КИНЕМАТИКА МАНИПУЛЯТОРОВ

Обратное отображение – это отображение из пространства сочленений в рабочее пространство робота:

$$q = f^{-1}(x)$$

Решения обратной кинематики:

1. Аналитические методы — решение, полученное напрямую в виде уравнения, предпочтительнее, но в общем виде неразрешимо для достаточно сложных реально существующих механизмов
2. Численные методы — сложно-вычислимы, неточны, являются приближениями, но это единственный практичный вариант решения в общем случае.

Распространённые методы численных решений обратной кинематики манипулятора:

1. Циклический координатный спуск (CCD – Cyclic Coordinate Descent)
2. Псевдо-обратный расширенный Якобиан

*Якобиан* или Матрица Якоби - инструмент, который широко используется в робототехнике и теории управления, определяет динамические отношения между двумя различными представлениями системы.

$$\dot{x} = \dot{T}(q) \dot{q}, \quad \dot{x} = J \dot{q}, \quad J = \dot{T},$$

*Якобиан* является матрицей частных производных, позволяет рассчитывать управление в силах (крутящих моментах) для обеспечения необходимых положения и скорости захвата. Может связывать не только скорости, но и ускорения

$$\ddot{x} = J(q) \dot{q} + J(q)\ddot{q}$$

Обратный Якобиан определяет изменения в  $q$  относительно  $x$ :

$$\dot{q} = J^{-1}\dot{x}$$

Манипулятор с пространством сочленений  $q$ , рабочим пространством  $x$ , кинематическим отображением  $f : q \rightarrow x$  *кинематически избыточен*, если  $\mathbf{dim} q > \mathbf{dim} x$ . Существует много путей в пространстве сочленений, соответствующих одному пути захвата в рабочем пространстве.

$\mathit{dim} q$  - размерность пространства сочленений, каждое сочленение плоского робота добавляет 1 размерность.

$\mathit{dim} x$  - размерность рабочего пространства, позиция и ориентация=2+1

Почти все реальные роботы манипуляторы кинематически избыточны - имеют большее количество сочленений, чем минимально требуется для выполнения их задач. Данное свойство вызывает проблемы при



выводе функции управления роботом - обратной кинематики, которая приобретает строгие ограничения применимости и неизбежно включает в себя кинематические особенности, а выполняемые траектории становятся «неповторяемы».

Точка называется *кинематической особенностью* робота, если Якобиан в ней имеет не полный ранг:  $\text{rang } J < \min(\text{dim } q, \text{dim } r)$ . Когда конфигурация манипулятора близка к кинематической особенности, для умеренных скоростей захвата требуются очень большие скорости соединений. Таким образом, нужно определять траектории для сочленений, избегающие с достаточным запасом особенностей.

Если пространство сочленений избыточно по сравнению с пространством захвата, т.е. система переопределена или слабо обусловлена, то Якобиан не является квадратной матрицей, не имеет обратной матрицы. Нет гарантий, что Якобиан обратим, вместо обратного Якобиана необходимо взять *псевдо-обратный*.

$$J^+ = (J^T J)^{-1} J^T$$

$J^T J$  по определению квадратный, т.е., возможно, *обратимый*.

Если Якобиан – это линейное приближение задачи, а псевдо-обратный Якобиан – приближение решения неразрешимой задачи, т.е. приближение приближения, почему же просто не использовать  $J^T$ , который легко вычислять, для нахождения смещения  $\dot{q}$ ? Потому что использование  $J^+$  ускоряет сходимость к цели.

*Виртуальными перемещениями* стремимся к правильному решению, совершая небольшие шаги, в сторону минимизации ошибки:

$$\text{Error} = |(I - J^+ J)\dot{x}|$$

## КОНКРЕТНАЯ МОДЕЛЬ ПЛОСКОГО МАНИПУЛЯТОРА

В модели руки, которую мы моделируем и тестируем есть максимум 4 сочленения: призматическое и три вращательных сочленения, обозначим:

$$q = (q_1, q_2, q_3, q_4) = (\phi, \psi, \theta, l_3)$$

Сочленения манипулятора имеют ограничения:

$$\begin{aligned} l_0 &\approx 0.07 & -58^\circ \leq \varphi \leq 47^\circ \\ l_1 &\approx 0.9575 & 56^\circ \leq \psi \leq 191^\circ \\ l_2 &\approx 0.8846 & -45^\circ \leq \theta \leq 55^\circ \\ l_3 &\approx 0 \div 0.4 \end{aligned}$$

Возможность отключать и подключать сочленения и приводы позволит нам более фактурно рассмотреть данную проблему обучения. Можно записать матрицы перехода к локальным координатам сочленений:

$$A_1 = \begin{pmatrix} 1 & 0 & l_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{—ключица}$$

$$A_2 = \begin{pmatrix} c_\phi & -s_\phi & l_2 c_\phi \\ s_\phi & c_\phi & l_2 s_\phi \\ 0 & 0 & 1 \end{pmatrix} \text{—плечо}$$

$$A_3 = \begin{pmatrix} c & -s_\psi & l_1 c \\ s & c & l_1 s \\ 0 & 0 & 1 \end{pmatrix} \text{—локоть}$$

$$A_4 = \begin{pmatrix} c_\theta & -s_\theta & l_0 c_\theta \\ s_\theta & c_\theta & l_0 s_\theta \\ 0 & 0 & 1 \end{pmatrix} \text{—запястье}$$

*Матрица прямого преобразования*<sup>2</sup> – перевод точки из координат сочленений  $q$  в координаты захвата  $x$ :

$$T = A_1 A_2 A_3 A_4$$

$$T = \begin{pmatrix} c_{\phi\psi\theta} & -s_{\phi\psi\theta} & l_0 c_{\phi\psi\theta} + l_1 c_{\phi\psi} + l_2 c_\phi + l_3 \\ s_{\phi\psi\theta} & c_{\phi\psi\theta} & l_0 s_{\phi\psi\theta} + l_1 s_{\phi\psi} + l_2 s_\phi \\ 0 & 0 & 1 \end{pmatrix}$$

<sup>2</sup>Сокращённые обозначения:  $c_\phi = \cos \phi$ ;  $s_\phi = \sin \phi$ ;  $c_{\phi\psi} = \cos(\phi + \psi)$ ;  $s_{\phi\psi} = \sin(\phi + \psi)$ ;  $c_{\phi\psi\theta} = \cos(\phi + \psi + \theta)$ ;  $s_{\phi\psi\theta} = \sin(\phi + \psi + \theta)$

## ВЕЗДЕХОД С ДВУМЯ ГУСЕНИЦАМИ

1. *Особая точка* — центр мобильного робота.
2. *Привод* — вращение гусеницы либо вперёд, либо назад.
3.  $n=3$  точки,  $2$  сочленения,  $m=4$  привода.

Модель движения идеализирована:

1. Без проскальзывания приводов
  2. Без изгибания осей приводов
  3. Приводы не сжимаются
- и т.д.

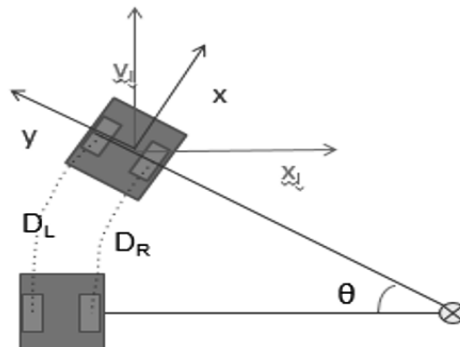
Не важно, колёса или гусеницы, главное, чтобы их можно вращать в обе стороны. Для устойчивости может быть добавлено несколько пассивных поддерживающих колёс.





Избыточность мобильного робота проявляется наиболее явно, кроме того, что есть бесконечное число траекторий проезда до целевой точки, находясь центром в целевой точке, приводы робота могут находиться в любом положении.



Рис. 3. Модель мобильного робота

## КИНЕМАТИКА МОБИЛЬНОГО РОБОТА



			
$V_L = +k,$ $V_R = 0$	$V_L = +k,$ $V_R = +k'$	$V_L = +k,$ $V_R = +k$	$V_L = +k,$ $V_R = -k$
кружится вокруг сочленения	кружится вокруг внешней точки	едет прямо	кружится во- круг центра

Кинематика мобильных роботов вычисляется от скорости и позиции робота. Влияние каждой гусеницы на скорость робота рассматривается отдельно. Поворот робота осуществляется с помощью разности управляющих воздействий. Преобразование усилия приводов в движение:

$$\Delta x = \frac{v}{\omega} \sin(\omega t); \Delta y = \frac{-v}{\omega} \cos(\omega t) + \frac{v}{\omega}$$

$$\Delta \theta = \frac{(V_L - V_R)t}{d}$$

#### ОБРАТНАЯ КИНЕМАТИКА МОБИЛЬНОГО РОБОТА

В данном конкретном случае получить аналитически обратную кинематику просто. Управление задаём в терминах углов вращения колёс:  $V_L = \dot{\Phi}_L r$  и  $V_R = \dot{\Phi}_R r$ .

$$(\dot{x}, \dot{y}, \dot{\theta}) = \left( \frac{\dot{\Phi}_L r + \dot{\Phi}_R r}{2}, 0, \frac{\dot{\Phi}_L r - \dot{\Phi}_R r}{2} \right)$$

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{r\dot{\Phi}_r + r\dot{\Phi}_l}{2} \\ 0 \\ \frac{r\dot{\Phi}_r - r\dot{\Phi}_l}{d} \end{pmatrix}$$

$$\dot{\xi}_I = T(\theta)\dot{\xi}_R$$

$$T^{-1}(\theta)\dot{\xi}_I = T^{-1}(\theta)T(\theta)\dot{\xi}_R$$

$$\dot{\xi}_R = T^{-1}(\theta)\dot{\xi}_I$$

$$T^{-1} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## ГОЛОНОМНАЯ КИНЕМАТИКА СИСТЕМЫ

В то время, как положение захвата манипулятора однозначно определяется углами сочленений, которые доступны в реальном времени, у мобильного робота значения датчиков просто возвращают обороты колёс, они должны быть интегрированы вместе со временем, что является источником большой неопределенности.

*Кинематика робота голономная*, если замкнутые траектории в пространстве сочленений  $q$  переводятся в замкнутые траектории в рабочем пространстве  $x$ .

*Кинематика неголономная система*, если замкнутые траектории в конфигурационном пространстве  $q$  могут не возвращать робота в начальное положение в рабочем пространстве  $x$ .

Манипулятор голономен, т.к. каждое положение сочленения соответствует единственному местоположению в пространстве. Мобильный робот, движущийся по рельсам тоже голономен.

Машинка или мобильный робот с гусеницами, которые могут вращаться с разной скоростью (differential-wheel robot) — неголономная система, потому что возвращение такого робота в начальное положение требует не только перемотки гусениц на нужное число оборотов назад, но ещё и правильного соотношения скоростей на них.

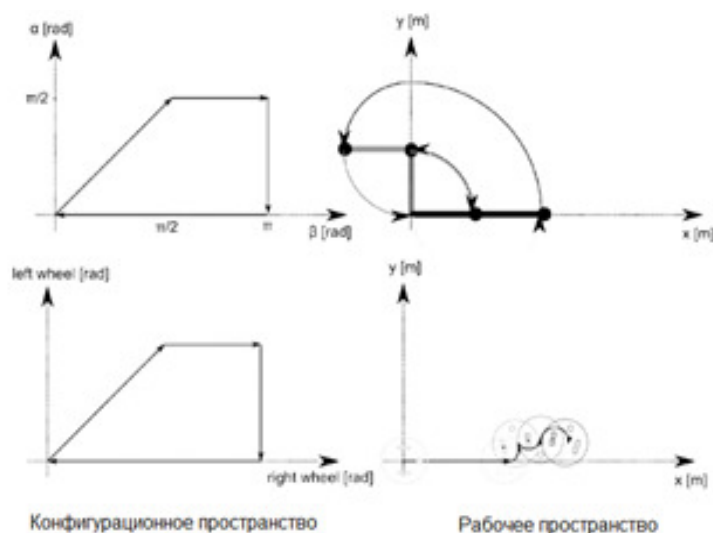


Рис. 4. Неголономная кинематика мобильного робота

На рисунке 5 показано, что мобильный робот сначала движется по прямой линии (обе гусеницы проворачиваются одинаковое число раз в секунду). Затем левая гусеница остаётся неподвижной, и только правая вращается вперёд. Потом только левая гусеница вращается назад. Наконец, правая гусеница вращается назад, приводя систему в изначальное состояние, но мобильный робот не вернулся к началу в координатах  $x$ .

Выполняя те же команды, на рисунке 5 видно, манипулятор с двумя вращательными сочленениями возвращается в исходное состояние.

Голономность или неголономность кинематики системы не должна влиять на качество работы алгоритма обучения и позиционирования. В том числе и эту зависимость предстоит изучить алгоритму.

## КООРДИНАТЫ СОЧЛЕНЕНИЯ И ПРИВОДА В МОДЕЛИ

Состояние системы  $s(t) = (x(t), x(\dot{t}))$  задано в неподвижных координатах рабочего пространства робота. Для робота это пространство является вычислимым, но не является непосредственно измеряемым. Доступные датчики, установленные в сочленениях, возвращают состояние приводов в локальных системах координат их сочленений. Информация с датчиков достаточно дискретизирована и точна. Функция вычисления статического состояния системы  $x$  хорошо изучена [3],[4],[5] по данным датчиков  $q$  (прямая кинематика) — простая и точная и не имеет особенностей:

$$f : q(t) \rightarrow x(t)$$

Обратная связь при управлении роботом может быть представлена в обоих этих пространствах  $q$  и  $x$ . Если представление  $x$  понятно и естественно, необходимо рассмотреть, что представляет из себя пространство  $q$ :

$$q(t) = ( q_1(t), \dots, q_{n-1}(t) )$$

где  $q_k(t)$  - это положения сочленений, они могут быть выражены в метрах или градусах поворота, в зависимости от сочленения.

Хоть конкретный график изменения положения сочленения во времени неизвестен, но можно предположить примерную функцию работы пары приводов-антагонистов сочленения. Самое основное - функция каждого привода гладкая, всюду существует первая и вторая производные, хотя, возможно, как мы покажем ниже, производные слева и справа могут не совпадать.

Более того, первый привод из пары имеет неубывающую функцию  $Q_{k_1}(t)$  в пространстве  $q$ , а второй — невозрастающую  $Q_{k_2}(t)$ .

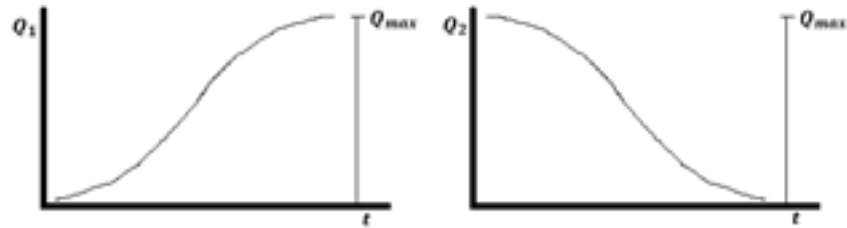


Рис. 5. Полное раскрытие и смыкание сочленения робота-манипулятора, показана функция действия приводов-антагонистов сочленения.

Тогда

$$\dot{q}(t) = (\dot{q}_1(t), \dots, \dot{q}_{n-1}(t)),$$

где  $\dot{q}(t)$  — это величины изменений положений каждого сочленения в заданный момент времени в локальных координатах соответствующего сочленения.

Влияния в каждый момент времени обоих приводов накладываются и общее смещение сочленения вычисляется

$$q_k(t) = q_k(t-1) + \dot{Q}_{k_1}(t) + \dot{Q}_{k_2}(t), \quad (4)$$

при этом за счёт возрастания одного и убывания второго, смещения приводов будут взаимнообратными.

$$\dot{q}_k(t) = \dot{Q}_{k_1}(t) + \dot{Q}_{k_2}(t) \quad (5)$$

Вообще говоря, величина смещения привода в одном направлении необязательно конечна. На рисунке показано, как себя ведут приводы в модели мобильного робота:

Можно видеть, что хотя на отметке  $Q_{max}$  гусеница вернулась в положение, с которого начиналось движение, смещение привода вперёд, а вместе с ним всего объекта, может продолжаться, что будет приводить к увеличению смещения в пространстве  $x$ . Можно бесконечно применять усилие в одном направлении (раскручивать колесо) и продолжать движение: будь то кружиться на одном месте или двигаться, пока не произойдёт столкновение с объектом среды. Величина смещения внутри одного привода в мобильном роботе неограничена.

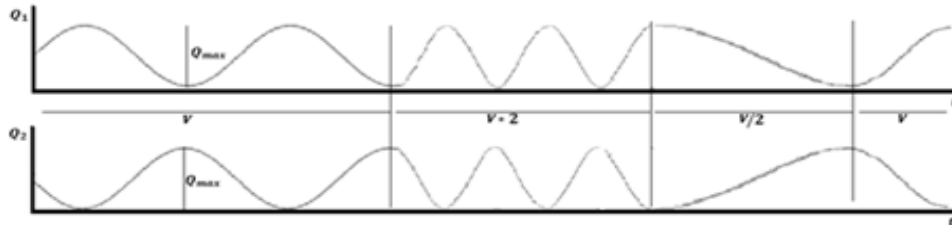


Рис. 6. Величина смещения от времени в приводах мобильного робота.

На том же рисунке 7 демонстрируется зависимость величины пройденного пути во времени для обоих приводов антагонистов, начинаем двигаться со скоростью  $v_k = \dot{Q}_{k_1}(t) = \dot{Q}_{k_2}(t)$ , затем ускоряемся до скорости  $2 \times v$ , замедляемся до скорости  $v/2$ , и в конце стабилизируем скорость на начальном уровне  $v$ .

Для робота манипулятора можно допустить, что сочленения прокручиваются целиком на  $360^\circ$ , не задевая звенья опоры, т.е. если устроены так, что в них нет конструктивных ограничителей.

В нашей модели манипулятора величина смещения внутри его приводов ограничена  $0 \leq q_k \leq Q_{k_{max}}$  (см. рис. 6), т.е. робот может раскрывать и смыкать сустав до каких-то пределов с обеих сторон. Это условие обобщает алгоритм обучения, мы стараемся в нём учесть разное поведение, которое может быть свойственно различным реальным системам.

По причине отсутствия информации о процессах, происходящих в приводах и сочленениях, нельзя полагаться на идентичность протекания процесса от попытки к попытке, от чего отсутствует повторяемость результата действия. Кроме того, во время движения на систему кроме усилий приводов действуют и другие силы, что нужно учитывать при генерации управления. Для симуляции такого поведения модели в внутреннюю функцию динамики системы будут введены *случайные отклонения*, научиться обрабатывать которые и есть конечная цель данного исследования.

Любое движение является в принципе *обратимым*, т.к. у каждого привода есть антагонист, смещающий  $q_k$  в противоположную сторону. Но обратное действие привода не гарантирует возвращения в изначальную точку. Теперь становится ясна причина наличия кнопки **reset**, она позволяет начинать любое действие с известного состояния системы.



## ПРОЦЕССЫ В ПРИВОДАХ ПРИ УПРАВЛЕНИИ

При задании действия в эпизоде *варьировать можно только время пуска привода относительно остальных, длительность активности каждого привода  $u_k(t)$*  (см. рис. 8). Длительность определяет на величину пройденного пути  $q_k(t)$  сочленением.

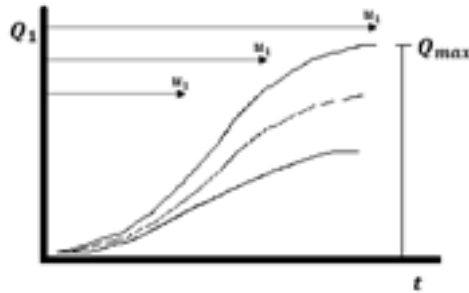


Рис. 7. Управление траекторией движения особой точки.

При работе приводы будут действовать по своим внутренним законам. Поскольку активация привода — не мгновенный процесс, существуют влияния инерции и иных сил на механизм системы, то для того, чтобы произошло видимое изменение известной части состояния системы  $x(t)$  в виде сдвигов точек системы, приводы должны быть активны в течение нескольких тактов времени подряд. Отбрасывая управляющие сигналы приводов, фактически не меняющие состояние системы, управление  $u(t)$  эпизода обучения можно представить, как кусочно-постоянную дискретную вектор-функцию, или как разряжённую матрицу размера  $N \times m$  с несколькими последовательностями подряд идущих единиц в столбцах, где  $m$  столбцов - число приводов, а  $N$  строк - максимальное число тактов для эпизода,  $(t_1 - t_0) < N$ , где  $t_0$  — начало эпизода, а  $t_1$  — конец эпизода. *Траектория движения особой точки системы* в течении эпизода есть положения  $n$ -й точки в каждый такт времени:

$$q_n(t_0, t_1) = \{q_n(t_0), q_n(t_0 + 1), \dots, q_n(t_i), \dots, q_n(t_1)\}, \quad (6)$$

$$t_0 < t_i < t_1$$

## ФУНКЦИЯ ПЕРЕМЕЩЕНИЯ СОЧЛЕНЕНИЯ

Используя (4) и (5) запишем в пространстве  $q$  формулу перемещения  $k$ -го сочленения в момент  $t$ , начиная эпизод с момента  $t_0$ :

$$q_k(t) = q_k(t_0) + \sum_{t=t_0}^t \dot{q}_k(t) = q_k(t_0) + \sum_{t=t_0}^t \left( \dot{Q}_{k1}(t) + \dot{Q}_{k2}(t) \right)$$

*Усилие привода* — одна из неизвестных функций, задаёт законы внутреннего функционирования системы, напрямую ей управлять нельзя. Обозначим смещение по усилию привода  $F_{kz} = F_{kz}(\dot{q}_k(t-1))$ ,  $k=\overline{1, n-1}$ ,  $z=1, 2$ . Усилие действует, когда управление  $u_{kz}(t)=1$  и зависит от скорости своего сочленения  $\dot{q}_k(t-1)$  в предыдущий момент времени. Тогда общий прирост смещения внутри привода за такт есть:

$$\dot{Q}_{kz}(t) = (-1)^z \left[ u_{kz}(t) F_{kz}(\dot{q}_k(t-1)) + I_{kz}(\dot{q}_k(t-1)) \right]$$

где  $F_{kz} \geq 0$  — смещение по усилию (Force) привода, передаваемое связанным точкам системы,  $I_{kz} \geq 0$  — смещение, вызванное инерцией (Inertia) после придания скорости.

Пример, как может выглядеть *функция усилия*:

$$F_{kz} = \dot{q}_k(t-1) + V,$$

где  $V = const$ .

Пример, как могут выглядеть *функции инерции*:

$$I_{kz} = \frac{\dot{q}_k(t-1)}{\sum_{j=1}^k m_j}$$

– для манипулятора, и

$$I_{kz} = \frac{\dot{q}_k(t-1)}{m_k + m_n}$$

– для вездехода, где  $m_j = const$ ,  $j = \overline{1, n}$ .

## ВЗАИМНЫЕ ВЛИЯНИЯ ПРИВОДОВ И СОЧЛЕНЕНИЙ

Как уже было сказано, любое движение имеет обратное, т.к. у каждого привода есть антагонист, смещающий  $q_k$  в противоположную сторону. В силу того, что разные по направлению силы могут быть активны одновременно, если они сравниваются по модулю, то они будут *компенсировать усилия друг друга*, плотно фиксируя сочленение.

В чем смысл отдельно выделять по сути два одинаковых состояния: *состояние покоя* сочленения, когда на её приводы-антагонисты не подаётся никакое управление  $u(0,0)$ , и *состояние взаимной блокировки*  $u(1,1)$ , когда на оба привода подаётся сигнал выработки усилия, если в обоих состояниях сочленение не движется?

В случае, когда на это сочленение не действуют и приводы иных сочленений, действительно, разницы нет. Но если рассмотреть случай, показанный на рисунке 9:

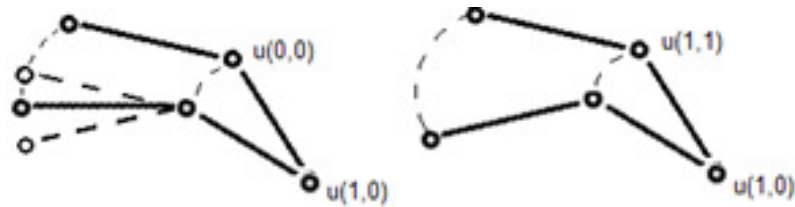


Рис. 8. (а) состояние покоя и (б) состояние взаимоблокировки локтя руки робота

Можно видеть, что в результате движений получаются разные положения захвата. В случае (а) левой части рисунка захват не доходит часть пути, относительно случая (б) в правой части. Пунктирные позиции показывают какое расстояние последнее звено не дошло – в нашей модели оно принято за половину.

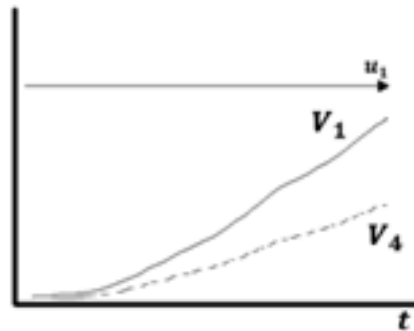


Рис. 9. Взаимное влияние сочленений. Закономерность скоростей

Т.е. сила, с которой сочленение влияет на звено следующее, за тем, которое связано непосредственно с этим сочленением, равна половине от исходной силы и противоположна по направлению. Сила, с которой со-

членение влияет на звено, через одно от непосредственно связанного с данным сочленением равна четверти исходной силы в обратную сторону. Рассмотрим рисунок 10, подаём управление на привод-смыкатель плеча, плечо начинает движение, развивая скорость полного усилия, и этим создаёт в покоящемся локте обратное усилие в половину величины, т.е. в приводе-размыкателе локтя имеем скорость  $-V_4=V_1/2$ . Добавим новый член в формулу усилия привода, который будет моделировать данное свойство:

$$\widetilde{\dot{Q}}_{kz}(t) = (-1)^z \left[ u_{kz}(t) F_{kz}(\dot{q}_k(t-1)) + I_{kz}(\dot{q}_k(t-1)) + \sum_{\substack{p=0 \\ p \neq k}}^n \frac{\dot{q}_p(t-1)}{(p-k)^2} \right]$$

Ещё раз смотрим на рисунок 9, часть (b) – сочленение фиксировано силами, не может свободно двигаться, таким образом, плечо перемещает всю руку, как единое звено. А в части (a), наоборот, в покоящемся сочленении есть свобода и неприкреплённый конец пытается сохранить своё положение при движении плеча, локоть раскрывается в противоположную сторону от движения плеча, вместо всего пути последнее звено проходит только половину.

Аналогичное эффект можно смоделировать для вездехода, рис. 11.

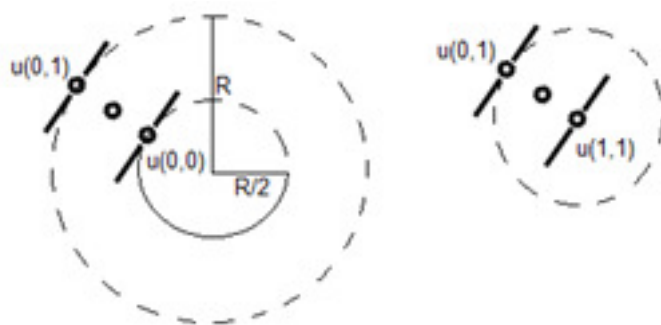


Рис. 10. (a) состояние покоя и (b) состояние взаимоблокировки левой гусеницы вездехода

## МОДЕЛЬ ОЖИДАЕМЫХ ИЗМЕНЕНИЙ В ДИНАМИЧЕСКОМ СОСТОЯНИИ СИСТЕМЫ ПРИ ЭКСПЛУАТАЦИИ ПОСЛЕ ОБУЧЕНИЯ

Допустим, что система обучена в предположительно статических условиях и теперь занимается решением ставящихся перед ней задач. В процессе работы механизма в среде функционирования могут происходить изменения условий *наблюдаемые* (перемещение в новое помещение, установка под углом, навешивание груза) и *ненаблюдаемые* (изменение вязкости рабочего тела пневматики и пр.). Это означает, что с течением времени внутренние законы действия системы могут "гладко" изменяться в небольших пределах. Ниже даём определение этим изменениям. Таким образом, от повторения к повторению конечное состояние эпизода может смещаться. Следовательно, запомненное в базе данных действие, успешно достигшее цели, на практике уже не всегда её достигает с нужной точностью. Динамическое состояние системы и изменения в нём задаются разностными дифференциальными уравнениями, проводя эксперименты с переобучением системы, уточнением управления важно выяснить границы применимости адаптации для разных видов уравнений.

### 1. Мгновенные изменения

Происходят быстро (порядка нескольких тактов), редко (1 эпизод из 1000), тем самым вносят шум, который нужно уметь *отфильтровывать*. Примеры таких изменений: порыв ветра, задевание недопустимого объекта в рабочем пространстве и др.

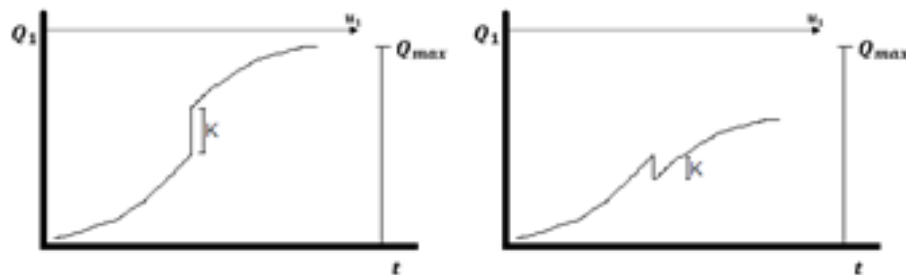


Рис. 11. Мгновенные изменения в модели системы

Для симуляции такого шума в закон функционирования привода добавим следующий член:

$$K(t_k) = \begin{cases} K, t \geq t_k \\ 0, t < t_k \end{cases}, \quad K = const$$

Запишем полученный закон для  $k$ -го привода:

$$\begin{aligned} \dot{\tilde{Q}}_{kz}(t) = & (-1)^z [u_{kz}(t) F_{kz}(\dot{q}_k(t-1)) + I_{kz}(\dot{q}_k(t-1)) + \\ & + \sum_{\substack{p=0 \\ p \neq k}}^n \frac{\dot{q}_p(t-1)}{(p-k)^2} + K(t_k)] \end{aligned} \quad (7)$$

## 2. Системные изменения

Происходят постепенно нарастающие изменения, которые действуют на работа длительное время, к ним нужно уметь *приспосабливаться* (нарастающие, в смысле, если новая функция динамики устоялась в течении 1000 тактов, постепенный возврат к исходной функции также является нарастающим изменением). Примеры системных изменений функционирования системы: атмосферное давление, влажность воздуха, температура помещения, износ подшипников и т.д.

$T$  тактов, достаточно много, для плавного перехода от одной функции динамического состояния системы к другой:

$$\begin{aligned} \dot{q}_k(t) & \rightarrow \left( (T-1)\dot{q}_k(t) + 1 \dot{\tilde{q}}_k(t) \right) / T \rightarrow \dots \\ & \rightarrow \left( (T-i)\dot{q}_k(t) + i \dot{\tilde{q}}_k(t) \right) / T \rightarrow \dots \\ & \rightarrow \left( 1 \dot{q}_k(t) + (T-1)\dot{\tilde{q}}_k(t) \right) / T \rightarrow \dot{\tilde{q}}_k(t), \end{aligned} \quad (8)$$

Происходят в виде постепенной замены функций усилия приводов, изменения конструкционных параметров (масс, направления гравитации и пр.). Значит, просто заменяем функцию динамики в каждый момент  $t$  на новую:

$$\begin{aligned} \dot{\tilde{Q}}_{kz}(t) = & (-1)^z \left[ u_{kz}(t) \check{F}_{kz} \left( \dot{\tilde{q}}_k(t)(t-1) \right) + \check{I}_{kz} \left( \dot{\tilde{q}}_k(t-1) \right) + \right. \\ & \left. + \sum_{\substack{p=0 \\ p \neq k}}^n \frac{\dot{q}_p(t-1)}{(p-k)^2} + K(t_k) \right] \end{aligned} \quad (9)$$

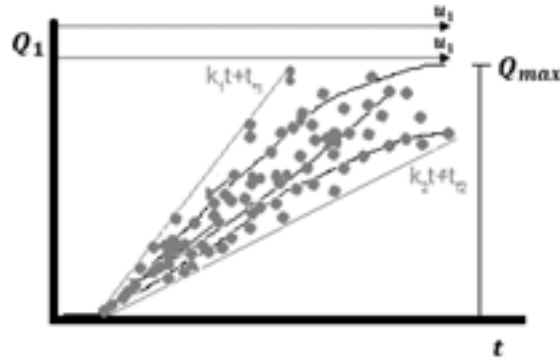


Рис. 12. Диапазон допустимых системных изменений в функциях усилия приводов

На рисунке ниже показаны границы, внутри которых могут быть выбраны функции усилия приводов:

Запишем формулу:

$$(k_{z2} \times t + t_{zf2}) \leq \dot{Q}_{k_z}(t) \leq (k_{z1} \times t + t_{zf1}), \quad (10)$$

где  $k_{z1} > k_{z2}$  – константы уклона,  $t_{zf1}, t_{zf2}$  – моменты времени, спустя которые усилие привода превосходит силу трения внутри сочленения. Аналогичный конус для противоположного убывающего привода, также расширяющийся с течением времени, только направленный вниз. Модель функционирует так, что чем меньше по длительности работает привод, тем меньше будет случайный разброс значений.

Чтобы более детально определить характеристики конкретной случайной функции из всех возможных нужно воспользоваться *теорией случайных процессов*. В рамках определённого здесь диапазона нужно задать *ансамбль реализаций случайного процесса* – правило всех реализаций этого процесса. Задать такое однопараметрическое семейство случайных величин (от параметра времени  $t$ ), заданных на одном и том же пространстве элементарных событий  $\Omega$ , мы можем достаточно произвольно, нужно только вспомнить, что для соответствующего привода функции в его семействе должны быть либо неубывающими, либо невозрастающими.

### 3. Граничные условия

В этом разделе представлены виды граничных возмущений на концах доступного хода привода, т.е. в первых тактах самого начала движения, а также при принудительной остановке системы в следствие некорректного управления – при ударах о границы свободного хода сочленений, о границы рабочей области. На рисунке 15 (а) показаны этапы работы одного сочленения целиком – от крайнего до крайнего момента, здесь внимание будет заострено на первой  $\Delta t_1$  и последней  $\Delta t_3$  частях движения сочленения.

3.1. Если не предотвращать столкновение робота с ограничениями (пре-

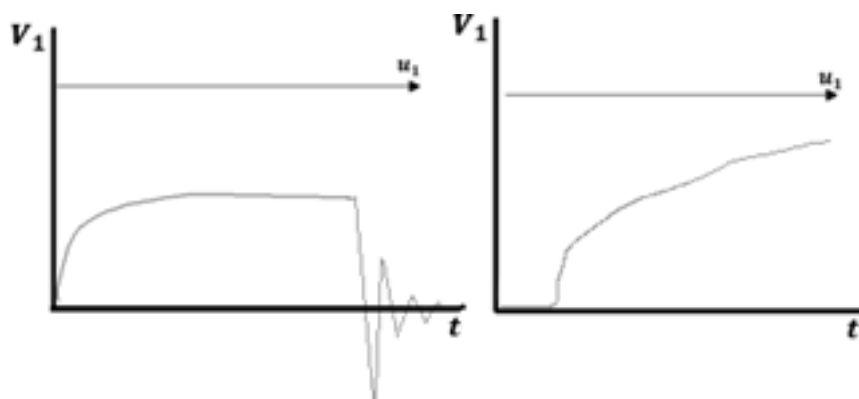


Рис. 13. Виды граничных изменений: (а) реакция смещения на столкновение, (б) преодоление трения на старте

делом движения сустава манипулятора или наезд на объект вездеходом), заблаговременно снизив скорость, можно получить сильные упругие колебания всей системы (рис. 16 (а)). Вид получаемых быстро затухающих колебаний можно описать следующей формулой (см. рис. 15 (b)):

$$f(x) = \frac{\sin x}{\sqrt{x}}$$

3.2. Колебания, возникающие на этапе *старта привода* (рис. 16 (b)), для преодоления трения внутри привода и сустава, или трения поверхности, требуется накопление импульса, за которым происходит моментальное ускорение особой точки системы и незначительная её раскачка. Располага



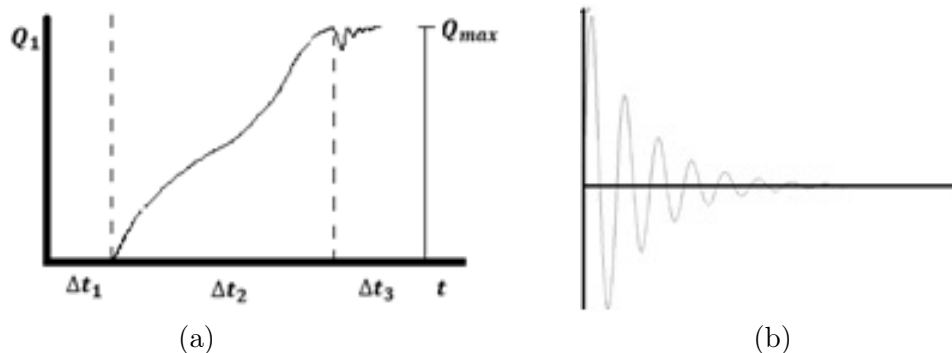


Рис. 14. (a) работа одного сочленения целиком, (b) функция колебаний столкновения

Ограничения суставов следует избегать с максимально возможным запасом при любой траектории пространства сочленений. В случае с проблемой на старте движения, можно понимать так, что максимальной эффективности позиционирования роботов с данными законами движения можно достичь на движениях средней длины относительно хода привода, т.е. мы принимаем во внимание тот факт, что не нужно располагать целевые точки очень близко к начальной позиции, чтобы получить нормальный результат позиционирования.

#### ОГРАНИЧЕНИЯ СКОРОСТИ И УСКОРЕНИЯ ПРИ УПРАВЛЕНИИ

Траектории суставов должны быть как можно более плавными (в реальном роботе грубые и резкие движения могут повредить механизм), и общее движение суставов должно быть не больше, чем необходимо для получения заданного движения захвата.

Даже идеальная модель системы неизбежно сталкивается с проблемами при использовании простого пропорционально-дифференцирующего управления в оперативном пространстве:

$$u_x = k_p(x^* - x) - k_v\dot{x}$$

где  $x$  и  $\dot{x}$  – позиция и скорость системы в оперативном пространстве,  $x^*$  – целевая позиция, а  $k_p$  и  $k_v$  – коэффициенты усиления пропорциональной и дифференцирующей составляющих.

Если так определить управляющий сигнал в пространстве  $x$  и преобразовать его в точные крутящие моменты сочленений  $q$ , то траектория достаточно длинного движения особой точки  $q_n(t)$  не будет прямой лини-

ей, т.к. либо двигатели не смогут фактически выдать нужные мгновенные крутящие моменты, либо будут недостаточны дискретизации управления или обратной связи. Некоторые из приводов связаны с меньшей массой, чем другие, в случае с кинематической цепью, и могут выдавать больший разброс по крутящим моментам в своём сочленении. Естественным представляется замедлить робота, ограничить скорость и ускорение для его сочленений. Введём желаемую скорость взяв её величину достаточно маленькой, чтобы крутящие моменты всех сочленений могли её обеспечить.

$$\begin{aligned}
 \tilde{x} &= (x - x^*) \\
 u_x &= -k_v \left( \dot{x} + \operatorname{sgn}(\tilde{x}) \max \left( V_{max}, \frac{k_p}{k_v} |\tilde{x}| \right) \right) \\
 u_x &= \begin{cases} -k_v (\dot{x} + \operatorname{sgn}(\tilde{x}) V_{max}) \\ -k_v \dot{x} + k_p \tilde{x} \end{cases}
 \end{aligned} \tag{11}$$

Остаётся сделать не одинаковое

### 1. Ограничение скорости

В случае с нашей моделью всё немного сложнее, потому что нет возможности задавать точные крутящие моменты на сочленения. Отсекать скорость можно только с некоторой задержкой, постфактум.

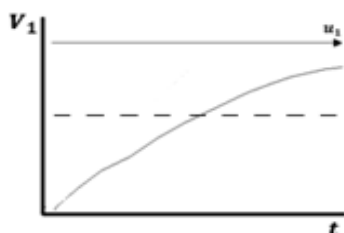


Рис. 15. Ограничение скорости

$$|\dot{q}_k(t)| < C_{kv}$$

где  $C_{kv} = \text{const}$

### 2. Ограничение ускорения

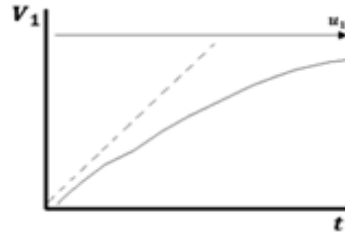


Рис. 16. Ограничение ускорения

Вопрос обстоит ещё сложнее с ограничением ускорений соединений роботов.

$$|\ddot{q}_k(t)| < C_{ka}$$

где  $C_{ka} = const$  – некоторая константа,

Упругие деформации создают колебательность в приводах при их активном разгоне и торможении, т.е. тогда, когда появляются ускорения [3]. Нежелательны колебания, возникающие *при торможении* привода, они затягивают процесс остановки робота, следовательно, снижают быстродействие системы. Колебания, возникающие *на этапе разгона* привода обычно успевают затухнуть к моменту его торможения и поэтому менее вредны. Однако, если длительность этапа движения с постоянной скоростью мала, оба эти типа колебаний могут наложиться, что сильнее затруднит процесс торможения.

## РЕЗУЛЬТАТЫ

Разработан алгоритм обучения систем с дискретным управлением в задаче позиционирования действовать своими приводами. Для первого и второго вопросов поставленных перед исследованием получены ответы, построены этапы алгоритма обучения, результаты сложности и времени обучения для двух моделей реальных устройств с нелинейными законами движения приведены в таблицах 1-4. В данном тексте третий вопрос исследования сформулирован как гипотеза для дальнейшей разработки, практические результаты ещё не получены.

Ответом на первый вопрос является двухэтапный алгоритм, который на первом этапе задействуя разные приводы и варьируя длительности их работы, даёт понимание разброса возможных состояний системы, направления работы приводов, локализует положение относительно робота интересующей области рабочего пространства в координатах управлений  $u(t)$ . Вторым этапом алгоритм создаёт плотную равномерную решётку конечных положений покрывающую все целевые точки интересующей области, подбор значений аргументов неизвестной функции, чтобы выдерживать равные дистанции между её значениями осуществляется методами Монте-Карло и МНК на основе действий, хранящихся в базе данных обучения.

Простые эксперименты показали, что случайная генерация бинарных матриц управлений за адекватное время не даёт такого же направленного эффекта, как данный двухэтапный алгоритм, по траекториям действий которого получается хорошая интерполяция: конечные положения и траектории равномерно распределены по интересующей области, без сгустков и пустот.

Сложность (в числе траекторий) алгоритма заметания решётки зависит от установленного шага (см. таблицы), чем меньше размер шага, тем выше сложность. Имея решётку конечных положений, дальше действовали двумя способами.

Первый способ – это приближаться к любым целевым точкам в изученной области попытками градиентного спуска. В среднем получается 8 попыток на 1 новую целевую точку. Выбираются несколько ближайших конечных точек к искомой и по ним строится промежуточное управление усреднением, приближающее к цели. Не всякие два управления подходят для получения промежуточного между ними, но если точек, рядом с целью достаточно много и дискретизация обратной связи и управления достаточно малы, алгоритм сходится к цели. Этот метод хорош тем, что легко применяет новый полученный опыт в процессе работы, а так-

же толерантен к шуму. Не требователен к вычислительным ресурсам, но имеет слишком большую сложность (число попыток). Алгоритм локальный, даже если учитывать обращение к базе данных, работает за константную асимптотику.

Второй способ – строить глобальную аппроксимацию случайными функциями[6] 2-функционала обратной кинематики системы. В среднем 1 попытка на новую целевую точку для данного метода, что очень хорошо. Но аппроксимация вычисляется для фиксированного набора произведённых траекторий. Пока не найден способ частичного пересчёта полной интерполяции (при добавлении каждой новой траектории нужно пересчитывать с нуля), очень высокие требования к оборудованию для пересчёта и квадратичная вычислительная сложность относительно записей в базе данных по памяти и CPU. Дegrадирует от зашумлённых данных.

*Сложностью* алгоритма назовём общее число движений системы при работе алгоритма обучения (изначальная категория, которую мы минимизируем). Метод оценки времени: если 1 такт дискретного времени = 1 миллисекунда (5 мкс), тогда 1 эпизод, максимальная длительность которого  $N$  тактов, в среднем длится  $N/10 \times (5 \text{ мкс}) \approx 0,5 \text{ с.}$ , ещё столько же нужно, чтобы вернуться в начальную точку. Получается, что одно позиционирование туда (эпизод) и обратно (reset) в среднем занимает 1 секунду. Всего целевых точек расположенных в интересующей области рабочего пространства робота 400 штук.

Таблица 1. Глобальная интерполяция для манипулятора

Этапы	Сложность	Шаг заметания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	255	71.43 мм	1	-	14 мин.
1.2	1352	8.93 мм	1	-	23 мин.
2	400	-	1	96%	7 мин.
Итого	2007	-	-	-	44 мин.

Далее есть намерение смешать описанные два подхода ко второму вопросу исследования, чтобы сгладить достоинствами одного недостатки другого. Применяя один алгоритм для разных систем показываем высокую абстрактность подхода, выявляем минимальные достаточные ограничения для успешного применения алгоритма для разных устройств.

Таблица 2. Глобальная интерполяция для вездехода

Этапы	Сложность	Шаг за-метания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	1359	18.75 мм	1	-	23 мин.
1.2	4288	5.62 мм	1	-	72 мин.
2	400	-	1	98%	7 мин.
Итого	6047	-	-	-	1.68 ч.

Таблица 3. Приближение попытками градиентного спуска манипулятора

Этапы	Сложность	Шаг за-метания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	255	71.43 мм	1	-	14 мин.
1.2	1352	8.93 мм	1	-	23 мин.
2	2468	-	7	100%	42 мин.
Итого	4075	-	-	-	1.19 ч.

Таблица 4. Приближение попытками градиентного спуска вездехода

Этапы	Сложность	Шаг за-метания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	1359	18.75 мм	1	-	23 мин.
1.2	4288	5.62 мм	1	-	72 мин.
2	3044	-	8	100%	51 мин.
Итого	8691	-	-	-	2.26 ч.

Продолжение исследования в направлении, которое можно описать так: память обширная, предоставляет возможности по построению адекватных предсказаний, но её данные со временем устаревают и перестают верно отображать обстановку вещей, а для того, чтобы переобучаться времени нет, уточнение действий должно производиться в реальном времени. После успешной фазы обучения происходит эксплуатация обученной системы, опыт постоянно дополняется. Так как записи действий сохраняются в базе данных навсегда, важно не запутаться в произведённых действиях с разными результатами и одной целевой точкой.

## Список литературы

- [1] Яблонский С.В. *Введение в дискретную математику*. — М.:Высшая школа, 2006.
- [2] Гасанов Э.Э., Кудрявцев В.Б. *Интеллектуальные системы. Теория хранения и поиска информации* — 2-е изд. — М.:Юрайт, 2017.
- [3] Юревич Е.И. *Основы робототехники*. — 2-е изд. — СПб.:БВХ-Петербург, 2005. — 416 с.: ил.
- [4] Зенкевич С.Л., Ющенко А.С. *Основы управления манипуляционными роботами*. М.:Изд-во МГТУ им. Н.Э.Баумана, 2004. — 480 с.
- [5] Кокс Д., Литтл Дж., О’Ши Д. *Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры*. Пер. с англ. — М.:Мир, 2000. — 687 с.
- [6] Бахвалов Ю.Н., Малыгин Л.Л., Черкасс П.С. *Многомерная интерполяция и аппроксимация на основе случайных функций* // Вестник Череповецкого ГУ, 2012. №2(39) [url: <https://cyberleninka.ru/article/v/metod-mashinnogo-obucheniya-na-osnove-algoritma-mnogomernoy-interpolyatsii-i-approksimatsii-sluchaynyh-funktsiy>]
- [7] Jan Peters, Stefan Schaal *Learning Operational Space Control* — University of Southern California, 2008. [url: [https://researchgate.net/publication/216053678\\_Learning\\_to\\_Control\\_in\\_Operational\\_Space](https://researchgate.net/publication/216053678_Learning_to_Control_in_Operational_Space)]
- [8] Travis DeWolf *IK and Operational space control* — University of Waterloo, 2013. [url: <https://studywolf.wordpress.com/site-index/>]
- [9] John Baillieul, D. P. Martin, Bruce R. Donald *Robotics. Resolution of kinematic redundancy*. Proceedings of symposia in applied mathematics, volume 41 — American Mathematical Soc., 1990. — 196 pp.
- [10] D. Lu *Kinematics of mobile robots* — University of Maryland, 2016. [url: <http://cfar.umd.edu/fer/cmsc498F-828K/lectures/Kinematics.pptx>]
- [11] Jason Clark *IK - Essential Math for Games Programmers*, 2005. [url: <https://essentialmath.com/InverseKinematics.pps>]
- [12] Welman, Chris, *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation* — Simon Fraser University. 1993. [url: <http://run.usc.edu/cs520-s15/ik/welman.pdf>]
- [13] Meredith, Michael and Maddock, Steve, *Real-Time Inverse Kinematics: The Return of the Jacobian* — University of Sheffield, 2006. [url: [https://staffwww.dcs.shef.ac.uk/people/S.Maddock/publications/MeredithMaddock2004\\_CS0406.pdf](https://staffwww.dcs.shef.ac.uk/people/S.Maddock/publications/MeredithMaddock2004_CS0406.pdf)]

**Learning algorithm of systems with discrete control**  
**Golikov K.A.**

The learning algorithm was developed for the problem of positioning systems with discrete control, it is based on a method of generalizing using a global interpolation and a gradient descent of trials and fails that stored in the database. The algorithm is optimized by the criterion of reducing the learning time (number of attempts). The algorithm was tested on a simulator for models of systems operating on a plane of two different types: for a mobile differential-drive robot and for an open kinematic chain with rotational and prismatic joints.

*Keywords:* positioning, learning algorithm, robot, interpolation, approximation.