Сидоренко Данил Дмитриевич

Санкт-Петербургский политехнический университет Петра Великого

ОПТИМИЗАЦИЯ РЕШЕНИЯ ОЗК НА МИКРОКОНТРОЛЛЕРАХ STM32F1 ДЛЯ РУЧНОГО УПРАВЛЕНИЯ ПЯТИОСЕВЫМ МАНИПУЛЯТОРОМ.

Аннотация. Оптимизация алгоритмов управления манипуляторами в ручном режиме необходима для предотвращения задержек в управлении при использовании дистанционного управления. Необходимость применения помехоустойчивых методов передачи данных, а также расстояние передачи играют большую роль в задержке управления оператором. В данной статье рассматривается классический метод решения ОЗК с помощью формализма Денавита-Хартенберга, а также приводится два варианта его оптимизации, что должно положительным образом сказать на сокращении задержки управления и повысить удобство управления манипулятором в ручном режиме на большой дистанции. В качестве первого варианта оптимизации рассмотрено применение геометрического описания кинематических связей между шарнирами манипулятора. В качестве второго варианта рассмотрено применение заранее просчитанных функциональных массивов вместо использования тригонометрических функций, а также переход только на целочисленные операции.

Ключевые слова. Робототехника, манипулятор, обратная задача кинематики, STM32, формализм Денавита-Хартенберга, числа с плавающей точкой, ручное управление манипулятором.

Ведение. Бортовые системы управления мобильными манипуляторами с ручным управлением не редко разрабатываются для микропроцессоров, не обладающих большой вычислительной мощностью. В связи с этим появляется необходимость оптимизации алгоритмов управления. Наиболее затратная для вычислений часть системы управления является решение обратной задачи кинематики при позиционировании рабочего органа манипулятора. Существует несколько методов решения ОЗК для манипуляторов. В данной статье представлено сравнение двух методов решения ОЗК: с помощью формализма ДХ и через геометрическое представление кинематических связей, так же предложен метод оптимизации решения ОЗК и приведено сравнение предложенного метода оптимизации.

Применение формализма Денавита-Хартенберга. Манипулятор управляется в ручном режиме управления с помощью задающих рукояток. Оператор наклоном ЗР задает скорость перемещения ЗУ по одной из доступных координат. [2, 4] В качестве доступных координат выступают 3 декартовые оси x, y b z, а также вращение вокруг осей x и y. В приведенной системе управления в манипуляторе отсутствует возможности поворота ЗУ вокруг оси z. В технике принято вращение вокруг продольной оси нарывать креном, а вокруг поперечной тангажом. Так как изменение ориентирующих координат не требуют контроля скорости, то за контроль скорости изменения углов тангажа и крена отвечают трехпозиционные переключатели, которые позволяют менять значения ориентирующих координат с линейной скоростью. Все решения ОЗК опираются на базовую систему координат, которая жестко связана с основанием манипулятора. Схема расположения БСК, системы координат звеньев и системы координат ЗУ приведена на рисунке 1.

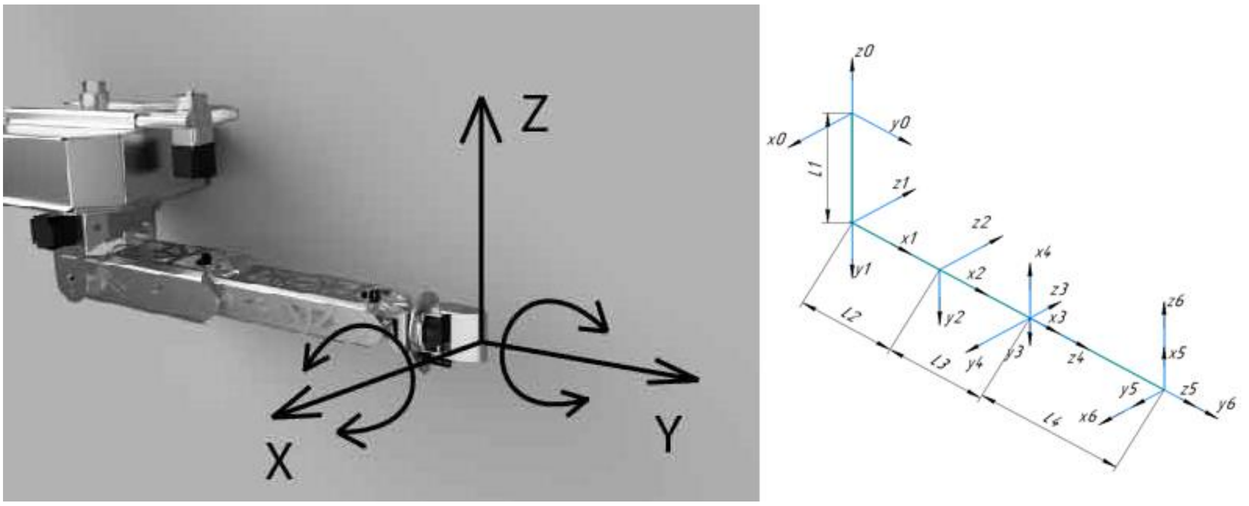

Рисунок 1 – Системы координат манипулятора

На рисунке 1 хБ, уБ, zБ – базовая система координат; х6, у6, z6 – система координат ЗУ; Li – длины соответствующих звеньев.

Выбор БЗК и систем координат звеньев был выполнен исходя из формализма Денавита-Хартенберга. [1, 6, 7] Оси вращения кинематических пар соответствуют осям zi соответствующих звеньев. При этом на рисунке 1 нумерация звеньев начинается с 0, а также пятая и шестая системы координат находятся на конце ЗУ и не учувствуют в передаче моментов вращения на звенья. Формализм ДХ подразумевает возможность решения как прямой, так и обратной кинематических задач путем решения матричного уравнения

$$T_0^6 = A_1A_2A_3A_4A_5A_6 \qquad (1)$$

Где A_i – матрица переноса соответствующего звена, а T_0^6 – матрица положения перехода то БСК к системе координат ЗУ. Матрицы A_i имеют вид

$$A_i = \begin{matrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & \sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{matrix} \qquad (2)$$

В формуле (2) a_i, и α_i являются параметрами Денавита-Хартенберга, а d_i – длина соответствующего звена. Присоединительной угловой координатой звена является параметр θ_i.

Матрица T_0^6 имеет вид:

$$T_0^6 = \begin{matrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 0 \end{matrix} \qquad (3)$$

где n_i, o_i, a_i – нормали системы ЗУ, а p_i – вектор положения координат захватного устройства относительно базовой системы координат

Как видно из формулы (3) для представления ориентирующих координат формализм ДХ использует 9 параметров. Оператор управляет двумя ориентационными углами. Данное несоответствие приводит к необходимости перевода ориентационных координат, задаваемых оператором, в форму матрицы T_0^6. Операция перевода координат требует вычисления сравнимого количества тригонометрических функций с операцией нахождения нового положения манипулятора, согласно заданным скоростям. Так же перевод углов крена и тангажа в матрицу перехода не целесообразен из-за конструктивного отсутствия в манипуляторе возможности вращать ЗУ относительно оси z.

Применение геометрического подхода. Для сокращения количества выполняемых операций применим геометрический подход к решению ОЗК. В данном методе кинематические связи описывается с помощью геометрических законов. Геометрический подход позволяет сократить число неизвестных в уравнениях для ОЗК тем, что значение угла θ_5 и может быть использовано без преобразований. Для данного метода требуется рассмотреть геометрические взаимосвязи углов и линейных размеров, представленных на рисунке 2.

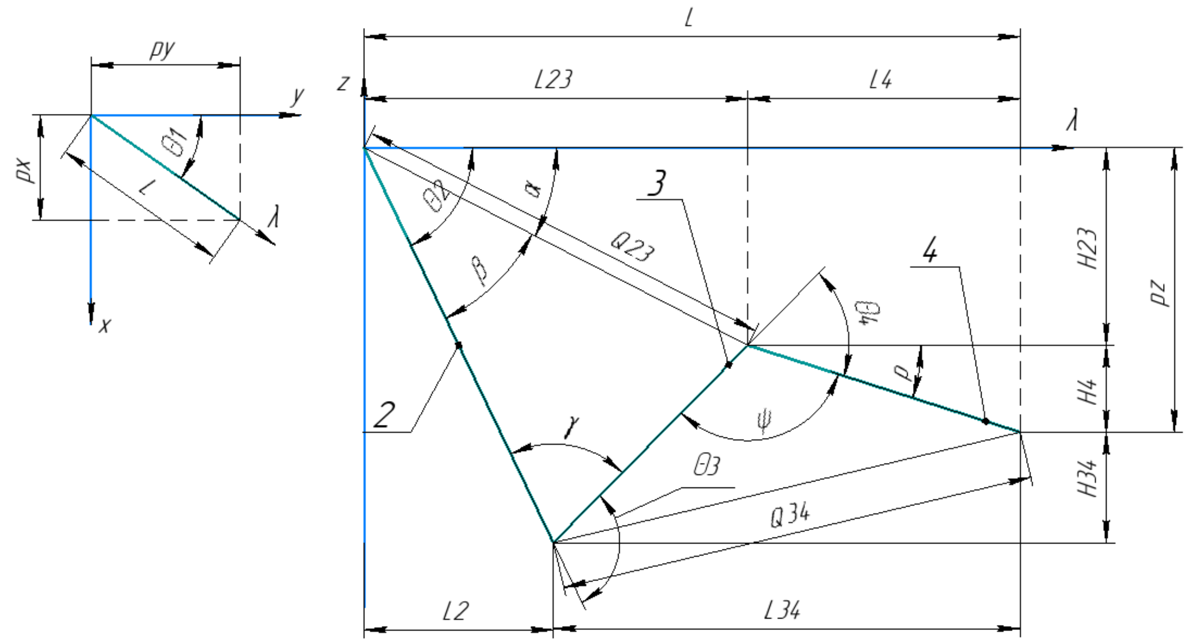

Рисунок 2 – Геометрическое описание манипулятора

Так же для сокращения вычислений БСК перенесена в точку сочленения первого и второго звеньев.

Пример расчета одной из обобщенных координат приведен в формулах 4 - 9.

$$L = \sqrt{x^2 + y^2} \tag{4}$$

$$H_{34} = l_2 \cdot \cos(\theta_2) - z \tag{5}$$

$$L_{34} = L - l_2 \cdot \sin(\theta_2) \tag{6}$$

$$Q_{34} = \sqrt{H_{34}^2 + L_{34}^2} \tag{7}$$

$$\varphi = acos\left(\frac{l_3^2 + l_4^2 - Q_{34}^2}{2 \cdot l_3 \cdot l_4}\right) \tag{8}$$

$$\theta_4 = \pi - \varphi \tag{9}$$

Высечение тригонометрических функций и операции с числами с плавающей точкой требуют затрат большого количества вычислительных ресурсов, так как микроконтроллеры STM32F1 [8] не обладают аппаратной поддержкой плавающей точки. Использование ряда Тейлора для оптимизации работы тригонометрических функций невозможно, так как для корректной работы алгоритма необходимо описать функции косинуса и синуса в диапазоне от 0 до pi с точностью не ниже чем 0,072 градуса. Данная точность продиктована дискретностью работы шаговых двигателей. Достижение заявленной точности в заданном диапазоне возможно только при использовании третьего члена в ряду Тейлора, что приводит к необходимости многократного умножения чисел с плавающей точкой и никак не ускоряет работу алгоритма.

Применение функциональных массивов. Следующий метод оптимизации решения алгоритма ОЗК призван избавиться от операций с числами с плавающей точкой и тригонометрический преобразований. Данный метод основан на использовании заранее подготовленного массива данных с тригонометрическими преобразованиями. Так же в данном методе применяются другие единицы измерения длин и градусов, чтобы избавиться от операций с числами с плавающей точкой. Применимы размерности угловых координат с дискретностью сопоставимой с заданной точностью изменения угловых координат. В примере для удобства расчетов дискретность угловых координат выше и составляет 0,01 градуса. Для более оптимального использования памяти микроконтроллера допускается снижение дискретности измерений до 0,072 градусов. Для баланса между точностью вычисления тригонометрических функций и занимаемой памятью максимальное значение синуса равно UINT16_MAX (65535). Применяются только беззнаковые значения для хранения тригонометрических функций, так как знак синуса и косинуса определяется непосредственно в теле алгоритма ОЗК, что позволяет значительно сократить количество хранимых данных и увеличить точность определения тригонометрических функций. В памяти микроконтроллера храниться два функциональных массива – массив косинуса со значениями от 0 до 65535, соответствующими углам от 0 до 90 градусов состоящий из 9000 значений; массив арккосинусов со значениями от 0 до 9000 состоящий из 6553 значений. В ходе экспериментов выяснилось, что хранить весть спектр значений арккосинуса в 65535 значений излишне и достаточно выбрать только каждое десятое значение, что значительно сокращает потребляемую память микроконтроллера и не сказывается на заданной точности. Схема геометрического представления используемых преобразований представлена на рисунке 3. Заштрихованной частью показана та часть, значения которой храниться в памяти микроконтроллера, все остальные значения находятся благодаря четности и симметричности тригонометрических функций.

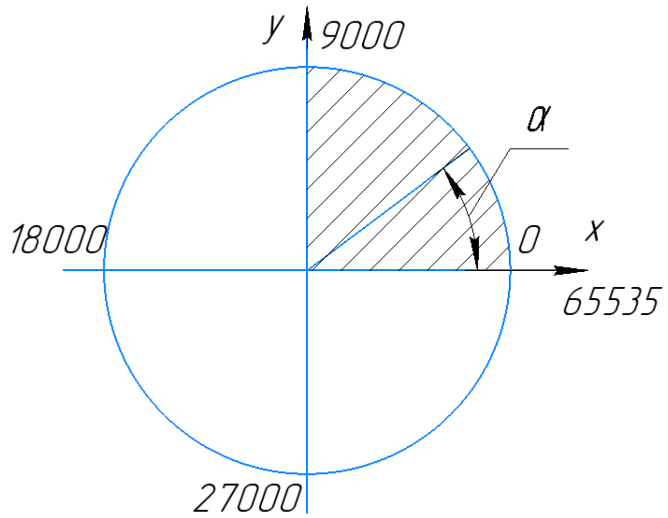

Рисунок 3 – Схема тригонометрических преобразований

При использовании данного метода исчезает необходимость вычисления тригонометрических функций, так как их значения берутся напрямую из массива. Так же благодаря использованию только целочисленных преобразований операции умножения и деления стали занимать меньше времени. Минусом данного метода является большой объем занимаемой памяти. При данных размерностях функциональных массивов они суммарно занимают 31106 байт, что составляет 96% Flash памяти на 32Кб микроконтроллерах. Использование дискретности в 0,072 градуса приведет к снижению затрат памяти в 2 раза, что позволит использовать микроконтроллеры с объемом памяти в 32Кб.

Использование данного метода приводит к необходимости дополнительной обработки знаков и четности тригонометрических функций. В качестве примера обработки значений ниже приведен фрагмент вычисления угла θ_1 с рисунка 3, где значение элемента массива s с индексом 0 соответствует углу θ_1.

```
if (x > 0)
    s[0] = -Acos((y * y + L * L - x * x)/(2 * y * L));
else
    s[0] = Acos((y * y + L * L - x * x)/(2 * y * L));
```

Сравнение методов решения. Тестирование скорости работы вышеописанных алгоритмов производилось на микроконтроллере STM32F103T6 с тактовой частотой 8,0 МГц и нулевым уровнем оптимизации. В качестве тестового задания использовалось последовательное решение ОЗК для окружности в пространстве по 360 точкам. Диаграмма времени работы описанных алгоритмов представлена на рисунке 4, где MYIK – алгоритм, использующий функциональные массивы; IK – алгоритм, использующий геометрическое представление кинематических связей; IKDH – алгоритм, использующий формализм Денавита-Хартенберга.

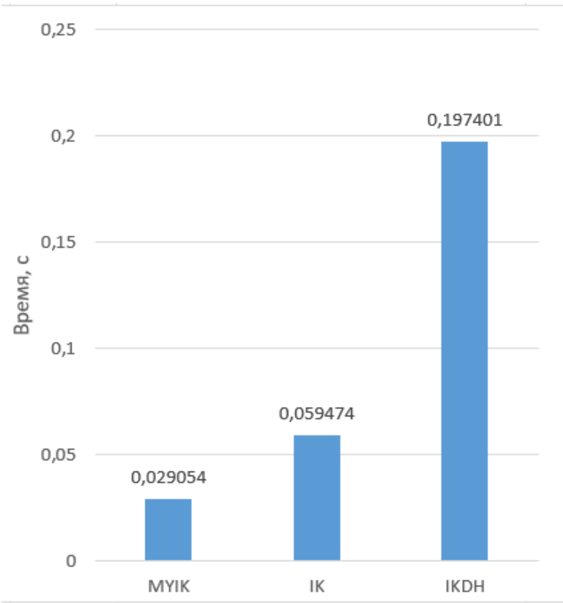

Рисунок 4 – Диаграмма сравнения решений

Как видно из рисунка 4 использование геометрического подхода дает выигрыш почти в 4 раза, а использование функциональных массивов дает преимущество еще в 2 раза.

Выводы. Использование функциональных массивов приводит к уменьшению времени работы ОЗК еще в 2 раза по сравнению с геометрическим подходом. Но в то же время использование функциональных массивов приводит к необходимости хранения большого количества данных, что исключает возможность использования одного микроконтроллера для осуществления всего необходимого для манипулятора функционала. Использование готовых библиотек протоколов связи, таких как CAN и UART накладывает ограничение на объем памяти доступный для хранения функциональных массивов. В некоторых случаях гораздо важнее объем памяти, занимаемый программой, чем уменьшение времени работы в 2 раза, и в таких случаях целесообразнее использовать геометрический метод.

Литература

1. Guo, D., Ju, H., Yao, Y., Ling, F., & Li, T. (2009). Efficient Algorithms for the Kinematics and Path Planning of Manipulator. 2009 International Conference on Artificial Intelligence and Computational Intelligence.

2. Kondak, K., Huber, F., Schwarzbach, M., Laiacker, M., Sommer, D., Bejar, M., & Ollero, A. (2014). Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. 2014 IEEE International Conference on Robotics and Automation (ICRA).

3. From, P. J., Duindam, V., Pettersen, K. Y., Gravdahl, J. T., & Sastry, S. (2010). Singularity-free dynamic equations of vehicle–manipulator systems. Simulation Modelling Practice and Theory, 18(6), 712–731.

4. Wuthier, D., Kominiak, D., Kanellakis, C., Andrikopoulos, G., Fumagalli, M., Schipper, G., & Nikolakopoulos, G. (2016). On the design, modeling and control of a novel compact aerial manipulator. 2016 24th Mediterranean Conference on Control and Automation (MED).

5. Love, L. J., Jansen, J. F., & Pin, F. G. (2004). On the modeling of robots operating on ships. IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004.

6. Шахинпур М. Курс робототехники / Шахинпур М. Пер с англ – М. Мир, 1990 – 572 с, ил.

7. Matlab Robotic System Toolbox User`s Guide, 2019 – 294 с.

8. STM32VLDISCOVERY – Режим доступа: https://www.st.com/en/evaluation-tools/stm32vldiscovery.html