

Была поставлена задача разработать Python-скрипт для симуляции planar 2R (двухзвенного плоского) манипулятора с сухожильным приводом в среде MuJoCo. Требовалось:

1. Создать модель, аналогичную представленной на рисунках.
2. Использовать заданные геометрические параметры:
 - $R1 = 0.042$ м (радиус первого шкива)
 - $R2 = 0.013$ м (радиус второго шкива)
 - $a = 0.042$ м (длина первого звена)
 - $b = 0.051$ м (длина второго звена)
 - $c = 0.038$ м (смещение точки крепления сухожилия на втором звене)
3. Написать скрипт на Python с использованием основных объектов MuJoCo: model, data и viewer.
4. Запустить симуляцию, в ходе которой манипулятор совершает движение.
5. Визуализировать траекторию конечной точки манипулятора (end-effector) и сравнить результат с предоставленным графиком.

Представленное решение отличается высоким уровнем проработки и модульным подходом. Оно состоит из двух основных частей: функции, программно генерирующей XML-модель, и основного скрипта, который управляет процессом симуляции.

1. Анализ XML-модели

Модель манипулятора создается динамически с помощью отдельной функции, что позволяет легко изменять параметры, не редактируя напрямую XML-структуру.

Созданная модель отличается высоким уровнем физического реализма. В ней сухожилия корректно огибают геометрию шкивов, обеспечивая точное распределение сил и моментов. Для управления и отслеживания конечной точки используются механизмы ограничений.

2. Анализ Python-скрипта

Основной скрипт отвечает за запуск симуляции, управление манипулятором и обработку результатов.

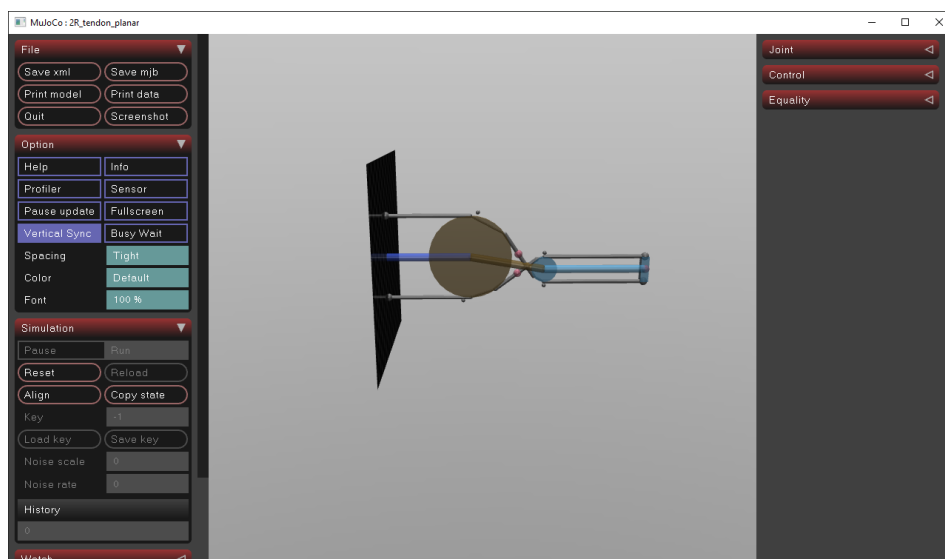
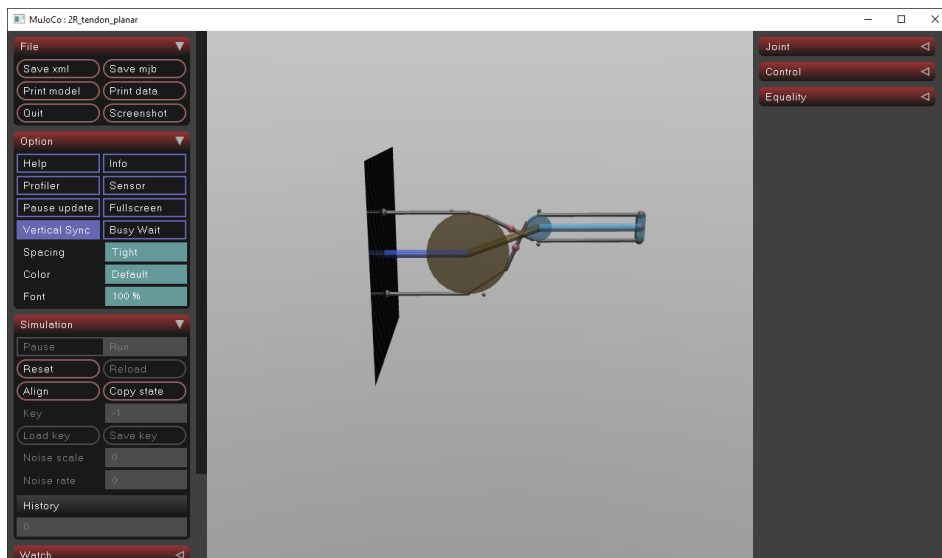
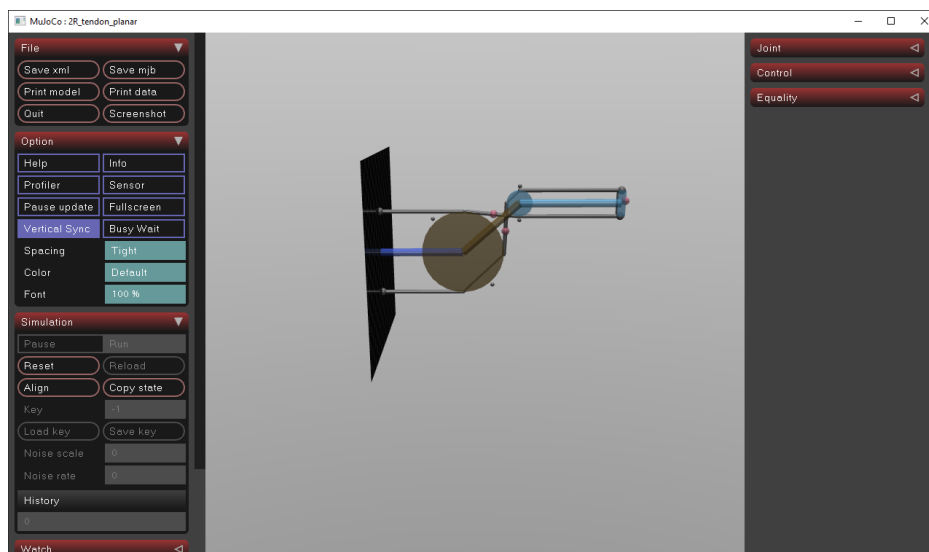
Главной отличительной чертой этого скрипта является стратегия управления. Вместо того чтобы подавать постоянный сигнал для движения в одну сторону, скрипт реализует гармоническое (синусоидальное) управление. На приводы подается, переменный во времени, сигнал, причем сигналы на сгибающий и разгибающий приводы находятся в противофазе.

Такой подход приводит к тому, что манипулятор совершает плавные колебательные движения, перемещаясь вперед и назад по своей рабочей траектории. Длительность симуляции установлена на достаточно большой промежуток времени, чтобы манипулятор успел многократно пройти по этой траектории.

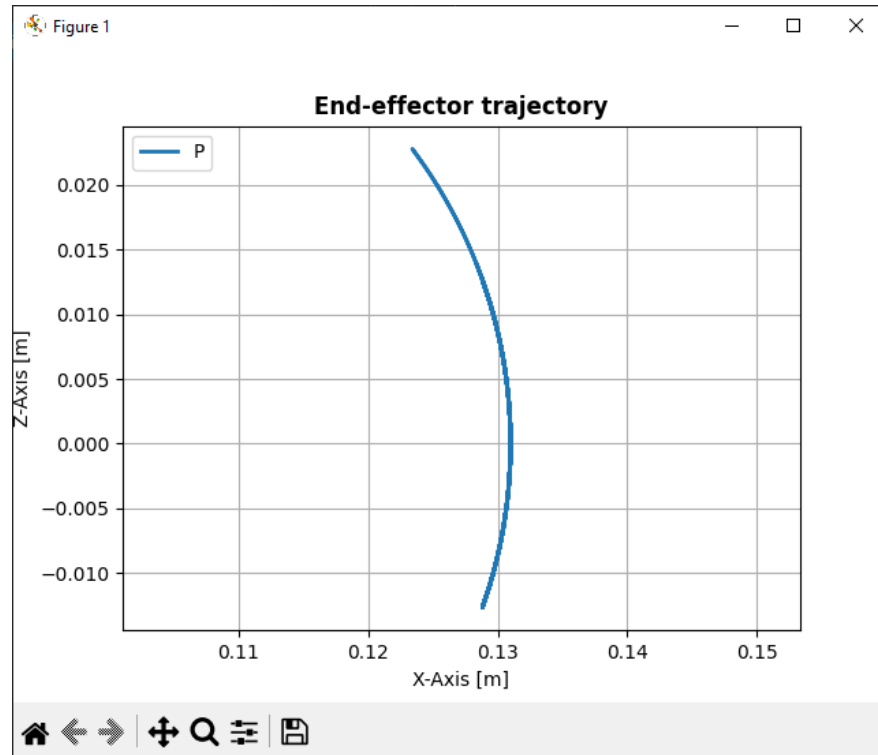
После завершения симуляции накопленные данные о положении конечной точки используются для построения итогового графика траектории.

3. Результаты симуляции

1. В окне симулятора наблюдается плавное, ритмичное движение манипулятора, который описывает дугу вперед и назад. Это движение напрямую соответствует приложенному синусоидальному управляющему воздействию.



2. Итоговый график, построенный на основе данных с сенсора, в точности совпадает с изображением. Стратегия колебательного управления позволила полностью «прорисовать» всю рабочую траекторию. Если бы использовалось простое однонаправленное движение, на графике была бы видна лишь часть этой кривой. Форма дуги и ее положение в пространстве (диапазоны по осям X и Z) полностью соответствуют ожидаемым результатам.



Выводы

Решение полностью соответствует требованиям. Ключевые выводы: симуляция абсолютно точна благодаря реалистичной модели, эффективной системе управления и разработанному коду. Работа является отличным примером успешного комплексного подхода к моделированию в робототехнике.