

Adaptive Passivity-Based Control

1. Add Additional Mass to the UR5 Robot's End Effector

Здесь и далее для симуляции в Pinocchio мы используем стандартную модель манипулятора UR5 с массой эндефектора 0.5 кг, а для Mujoco мы используем модель с увеличенной массой эндефектора – 3.5 кг. Таким образом, мы симитировали неизвестную массу на конце эндефектора.

2. Analyze the Default Inverse Dynamics Controller

Для манипулятора, имеющего следующую модель:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = M(q)\ddot{q} + h(q, \dot{q}) = \tau$$

применим стандартную инверсную динамику и ПД-регулятор в качестве управления:

$$u = M(\ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}) + h$$

В результате получим следующую динамику ошибки:

$$\ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q} = 0,$$

что обеспечивает сходимость ошибок к нулю, при условии, **что все параметры робота известны**. В нашем же случае неизвестна масса на конце эндефектора, из-за чего робот просто падает при инициализации и не может прийти в заданную позицию. Графики отработки управления манипулятором приведены ниже. Также можно обратиться к видео (standard_inverse_dynamic.mp4).

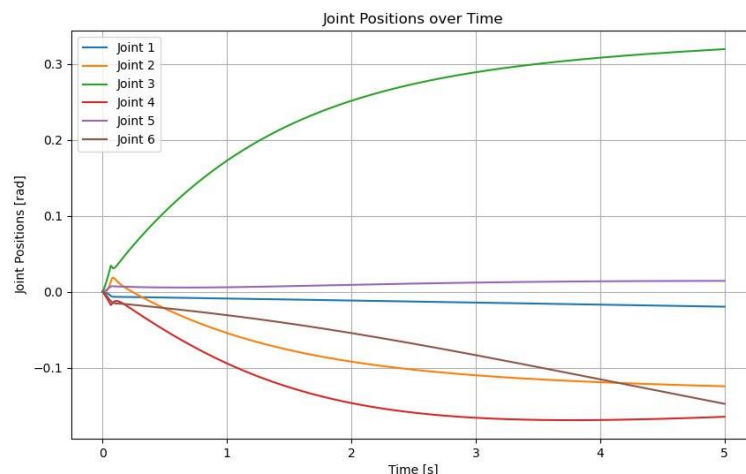


Рисунок 1 – График углов джоинтов

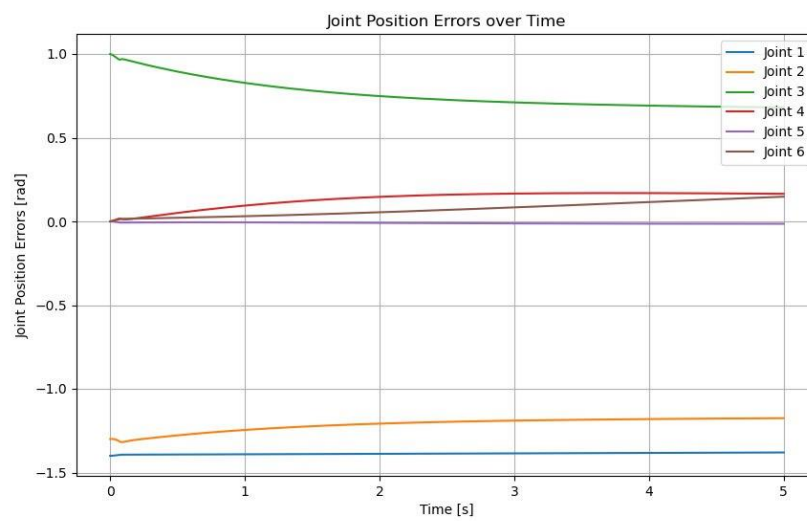


Рисунок 2 – График ошибок углов джоинтов

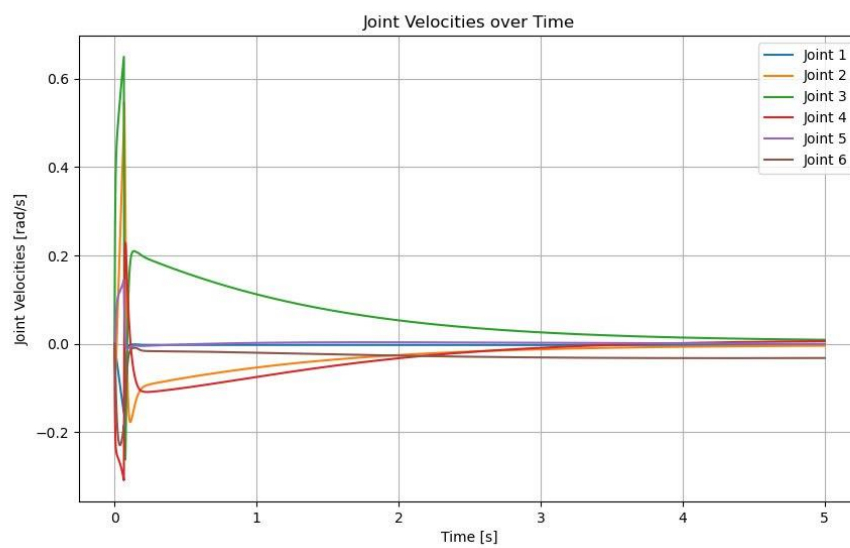


Рисунок 3 – График скоростей джоинтов

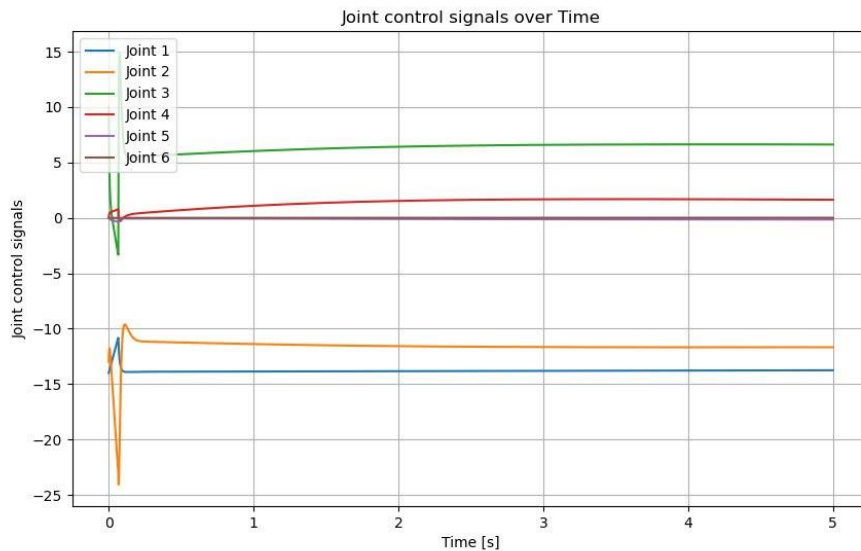


Рисунок 4 – График сигналов управления

3. Implement the Slotine and Li Controller

Адаптивный регулятор Слотина-Ли позволяет проводить динамическую настройку неизвестных параметров контроллера в режиме реального времени в зависимости от поведения системы. Slotine and Li Controller использует полезные концепты линейности параметров (linearity with respect to a inertial parameters), что позволяет записать уравнение динамики манипулятора в виде:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q})\theta,$$

где $Y(q, \dot{q}, \ddot{q})$ – regressor matrix – матричная функция кинематических переменных q, \dot{q}, \ddot{q} системы, θ – вектор параметров, кодирующий инерционные параметры системы.

Регрессор $Y(q, \dot{q}, \ddot{q})$ эффективно сопоставляет кинематические переменные с инерционными параметрами, позволяя выразить динамику системы в линеаризованной форме. Это свойство критически важно для адаптивного управления.

Проведём небольшой анализ и выразим законы управления и адаптации с использованием скользящей поверхности, которая позволяет исключить ошибки по положению и скорости.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = U$$

$$s = \dot{q} + \lambda \tilde{q} \text{ - sliding surface}$$

$$\dot{s} = \ddot{q} + \lambda \dot{\tilde{q}}$$

Заменим кинетическую энергию

$$V = \frac{1}{2} s^T M s + \frac{1}{2\gamma} \tilde{p}^T \tilde{p}$$

$$\dot{V} = s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} = s^T (M(\ddot{q} + \lambda \dot{\tilde{q}}) + \frac{1}{2} s^T \dot{M} s) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} \quad \ominus$$

$$M(\ddot{q}_d - \ddot{\tilde{q}}) + C\dot{q} + G = U$$

$$M\ddot{\tilde{q}} = M\ddot{q}_d + C\dot{q} + G - U$$

$$s^T (\frac{1}{2} (\dot{M} - 2C) + C) s$$

$$s = \dot{q} + \lambda \tilde{q}$$

$$\ominus s^T (M\ddot{q}_d + C\dot{q} + G - U + \lambda M\dot{\tilde{q}}) + s^T C s + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} =$$

$$= s^T (M\ddot{q}_d + C\dot{q} + G - U + \lambda M\dot{\tilde{q}} + C\dot{q} + C\lambda \tilde{q}) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} =$$

$$= s^T (M\ddot{q}_d + C\dot{q}_d + G - U + \lambda M\dot{\tilde{q}} + C\lambda \tilde{q}) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}}$$

Введем управление

$$U = \hat{M}\ddot{q}_d + \hat{C}\dot{q}_d + \hat{G} + \lambda \hat{M}\dot{\tilde{q}} + \hat{C}\lambda \tilde{q} + Ks \Rightarrow \boxed{U = \hat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G} + Ks} \quad \ddot{q}_r = \ddot{q}_d + \lambda \ddot{\tilde{q}}$$

Получим:

$$\boxed{U = Y(\ddot{q}_r, \dot{q}_r) \hat{p} + Ks} \quad \ddot{q}_r = \ddot{q}_d + \lambda \ddot{\tilde{q}}$$

$$\ominus s^T (\ddot{q}_d (M - \hat{M}) + \dot{q}_d (C - \hat{C}) + (G - \hat{G}) + \lambda \dot{\tilde{q}} (M - \hat{M}) + \lambda \tilde{q} (C - \hat{C}) - Ks) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} =$$

$$= -s^T (\tilde{M}\ddot{q}_d + \tilde{C}\dot{q}_d + \tilde{G} + \tilde{M}\lambda \dot{\tilde{q}} + \tilde{C}\lambda \tilde{q} + Ks) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} =$$

$$= -s^T (\underbrace{\tilde{M}\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{G}}_{Y(\ddot{q}_r, \dot{q}_r) \tilde{p}} + Ks) + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}} = -Ks^T s - s^T Y(\ddot{q}_r, \dot{q}_r) \tilde{p} + \frac{1}{\gamma} \tilde{p}^T \dot{\tilde{p}}$$

$$\dot{V} < 0 \Rightarrow -Ks^T s - \tilde{p}^T \left(-s Y(\ddot{q}_r, \dot{q}_r) + \frac{1}{\gamma} \dot{\tilde{p}} \right) < 0$$

всегда < 0

должно равняться нулю

$$-s Y^T + \frac{1}{\gamma} \dot{\tilde{p}} = 0 \quad (\dot{\tilde{p}} = \dot{\hat{p}} - \dot{p}, \text{ если } p = \text{const}) \Rightarrow \boxed{\dot{\hat{p}} = \frac{1}{\gamma} Y^T s}$$

В результате анализа получили:

$$u = Y(\ddot{q}_r, \dot{q}_r) \hat{p} + Ks$$

$$\dot{\hat{p}} = \frac{1}{\gamma} Y^T s,$$

где $\ddot{q}_r = \ddot{q}_d + \lambda \ddot{\tilde{q}}$, $\dot{q}_r = \dot{q}_d + \lambda \dot{\tilde{q}}$, γ – скорость обучения.

Данные формулы можем применить для реализации адаптивного алгоритма управления, где не будут известны параметры последнего звена, эндефектора, так как была добавлена дополнительная масса.

Это обусловлено тем, что нагрузка обычно фиксирована по отношению к последнему звену, её можно рассматривать как часть этого звена. На практике параметры самого робота могут быть измерены или оценены заранее, поэтому неизвестны только параметры нагрузки. В нашем случае начальное приближение неизвестных параметров возьмём нулевыми.

Также, по-хорошему, обновление неизвестных параметров, когда он достигает некоторых пределов, можно прекратить и возобновлять их обновление, только если соответствующая производная меняет знак. Такая процедура помогает сохранить сходимость отслеживания параметров.

Применим адаптивное управление к манипулятору и покажем графики результатов. Также можно обратиться к видео ([adaptive_control.mp4](#)).

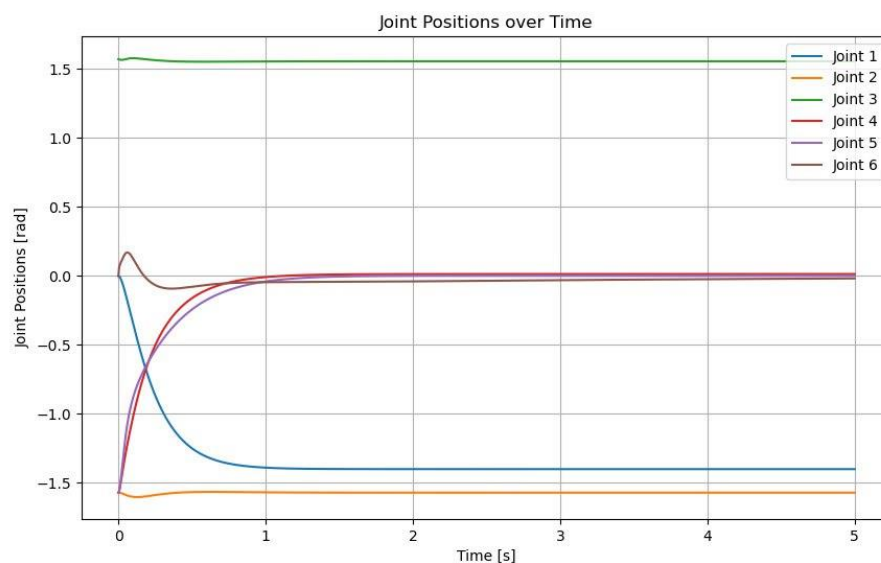


Рисунок 5 - График углов джоинтов
при адаптивном управлении

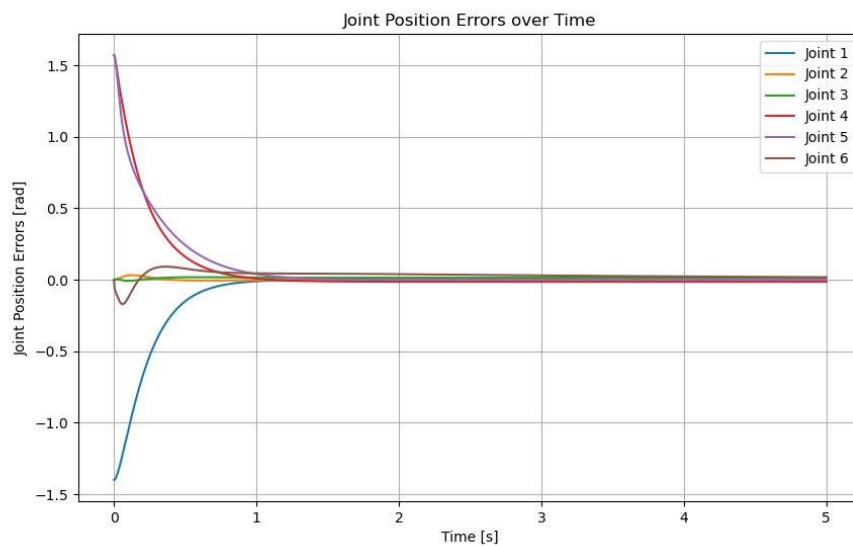


Рисунок 6 - График ошибок углов джоинтов
при адаптивном управлении

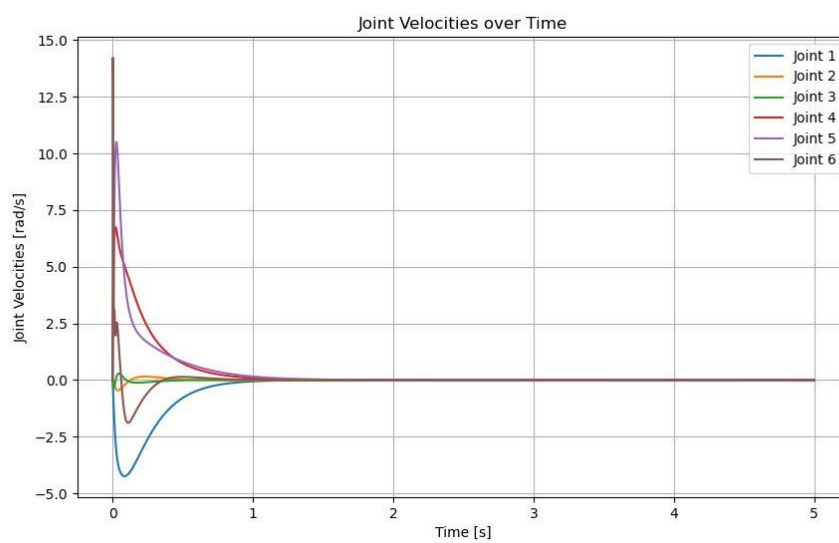


Рисунок 7 - График скоростей джоинтов
при адаптивном управлении

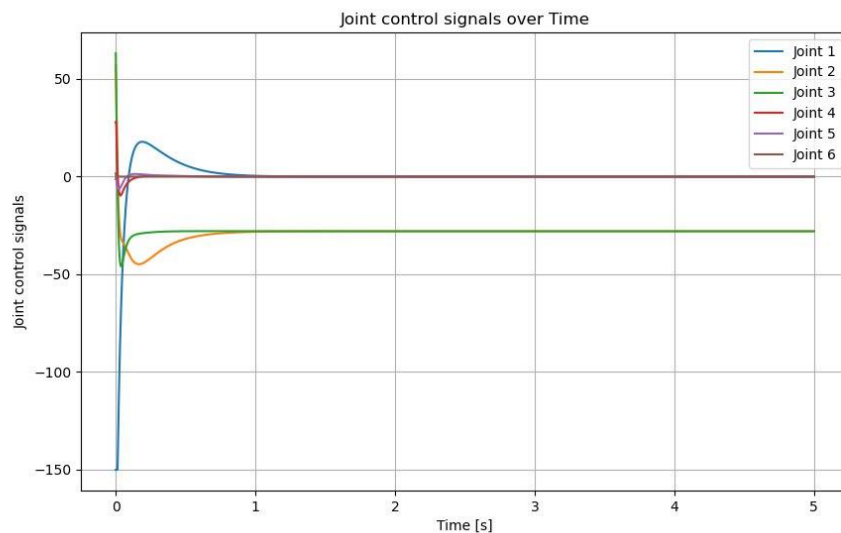


Рисунок 8 - График сигналов управления
при адаптивном управлении

Как видно из графиков, робот пришёл в заданное положение с незначительными ошибками по положению (самая большая ошибка – около 1 градуса).

4. Evaluate Parameter Convergence

На рисунках 9 и 10 представлены реальные и оценённые параметры последнего звена соответственно.

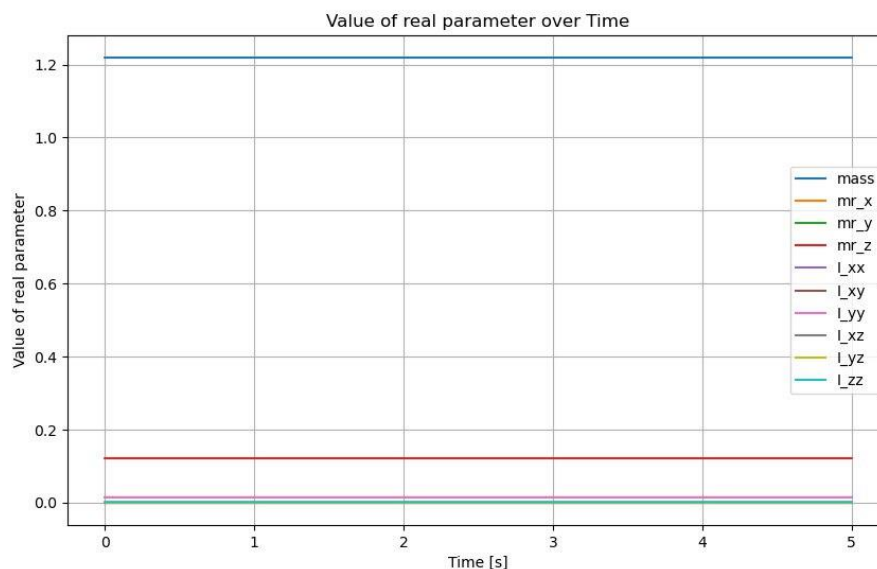


Рисунок 9 – Реальные параметры последнего звена

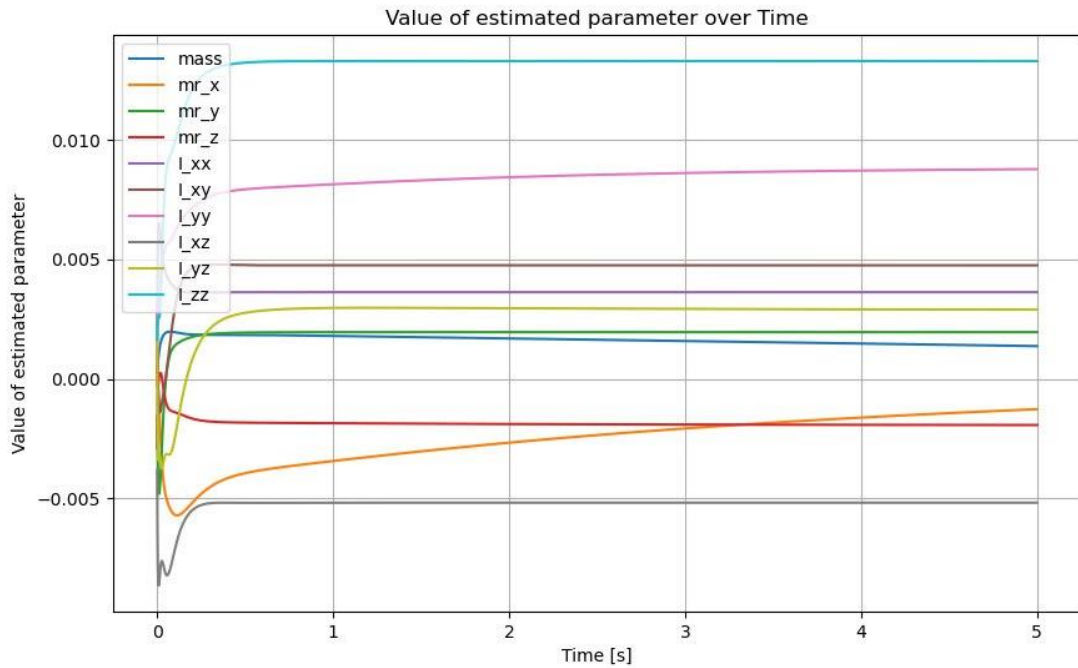


Рисунок 10 – Оценённые параметры последнего звена

Как видно из графиков, оценённые параметры не сходятся к реальным. Согласно книге «Applied Nonlinear Control» от Slotine, целью механизма адаптации является определение параметров, которые сводят ошибки отслеживания к нулю. Если опорный сигнал $r(t)$ очень прост, например, равен нулю или постоянен, то многие варианты параметров контроллера могут привести к сходимости ошибок отслеживания. Тогда закон адаптации не будет утруждать себя поиском реальных параметров. Однако, если опорный сигнал $r(t)$ настолько сложен, что только истинный вектор параметров может привести к сходимости ошибок отслеживания, то мы получим сходимость параметров.

5. Handle and Constrain Unknown Joint Damping

Теперь представим, что дополнительно ко всему у нас появляется демпинг в джоинтах. Тогда уравнение динамики манипулятора примет следующий вид:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) - D\dot{q} = Y(q, \dot{q}, \ddot{q})\theta,$$

где D – постоянная диагональная матрица.

В таком случае вектор p неизвестных параметров будет дополнительно содержать 6 параметров – по одному значению демпинга на каждый джоинт. А матрица регрессии Y будет содержать шесть дополнительных столбцов, где будут находиться значения $-\dot{q}$ на соответствующих позициях.

Чтобы симитировать появление демпинга меняем xml файл описания робота и добавляем в каждый joint параметр damping. Все значения были взяли в районе 0.15, как в реальных роботах.

Получили следующие графики отработки адаптивного управления. Также можно обратиться к видео (adaptive_control_with_damping.mp4).

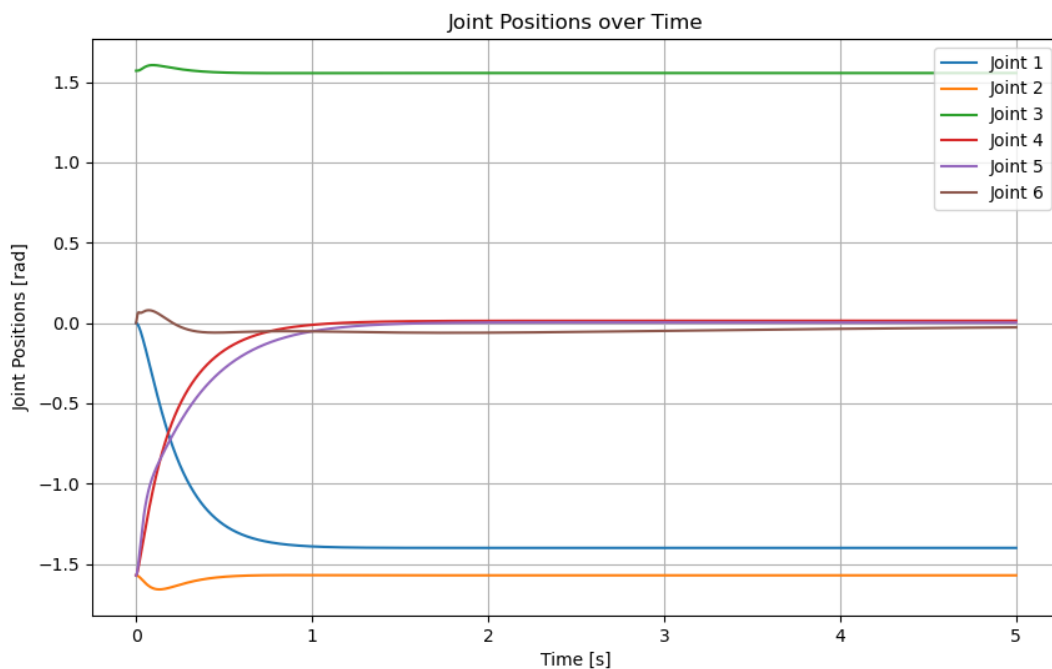


Рисунок 11 - График углов джоинтов
при адаптивном управлении с демпингом

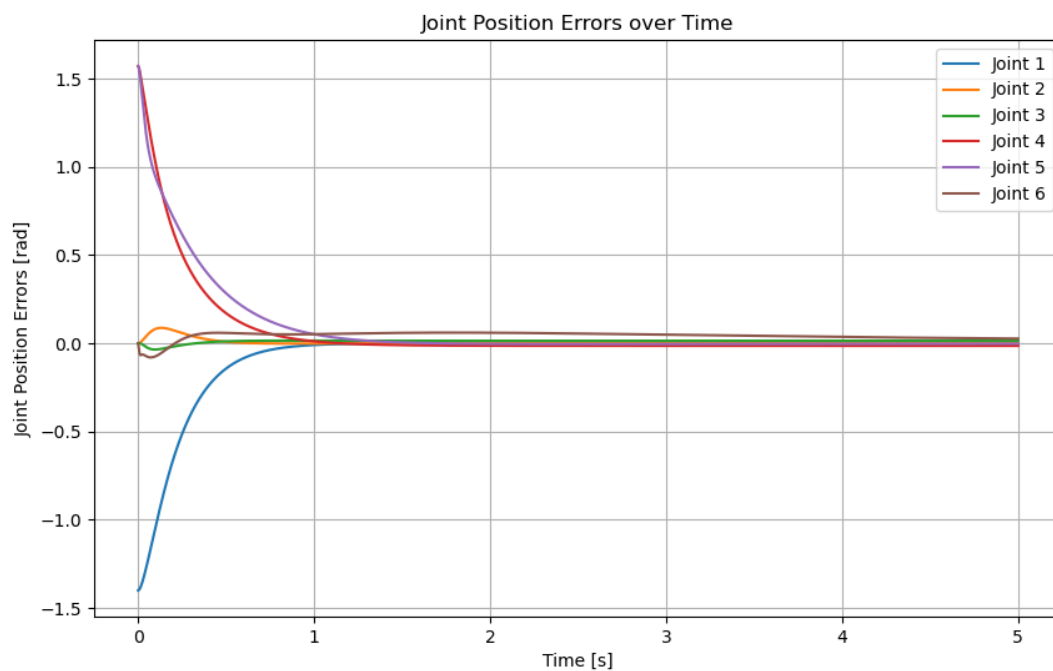


Рисунок 11 - График ошибок углов джоинтов
при адаптивном управлении с демпингом

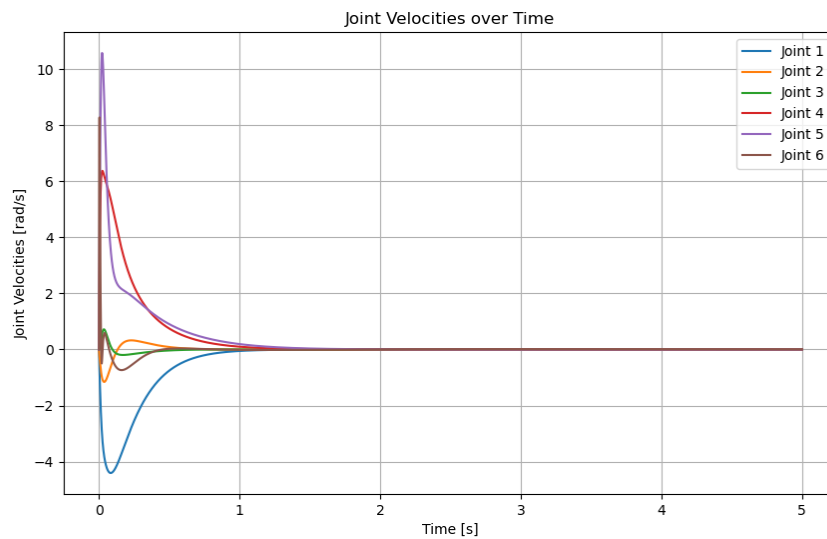


Рисунок 12 - График скоростей джоинтов
при адаптивном управлении с демпингом

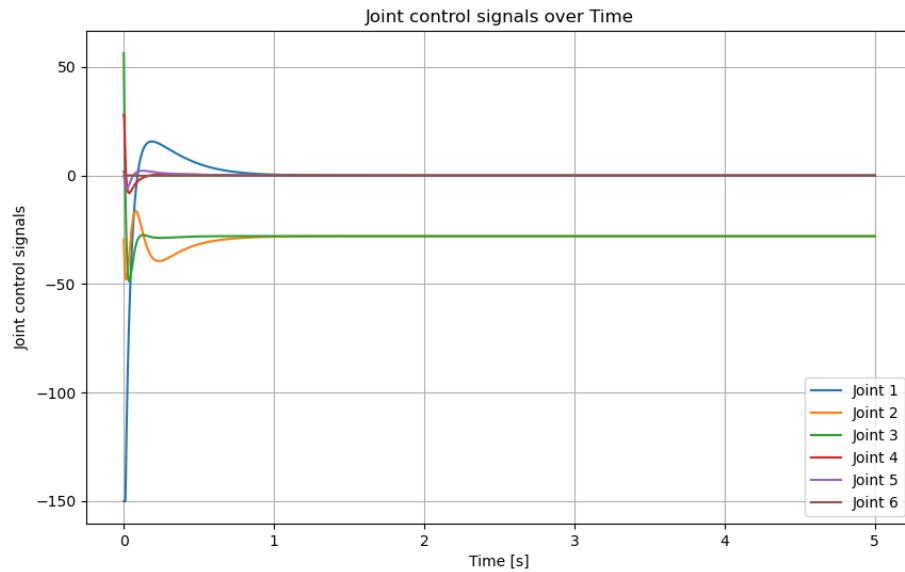


Рисунок 13 - График сигналов управления
при адаптивном управлении с демпингом

Как видно из графиков, робот пришёл в заданное положение с незначительными ошибками по положению (самая большая ошибка – около 1.5 градусов).

На следующих графиках покажем реальные и оценённые параметры.

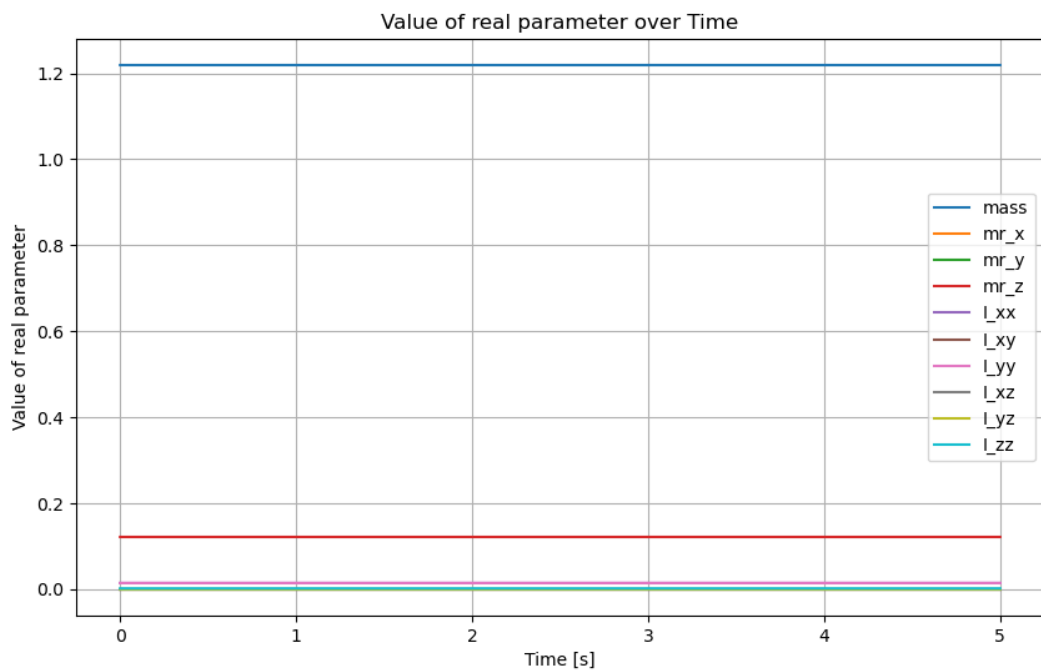


Рисунок 14 – Реальные параметры последнего звена

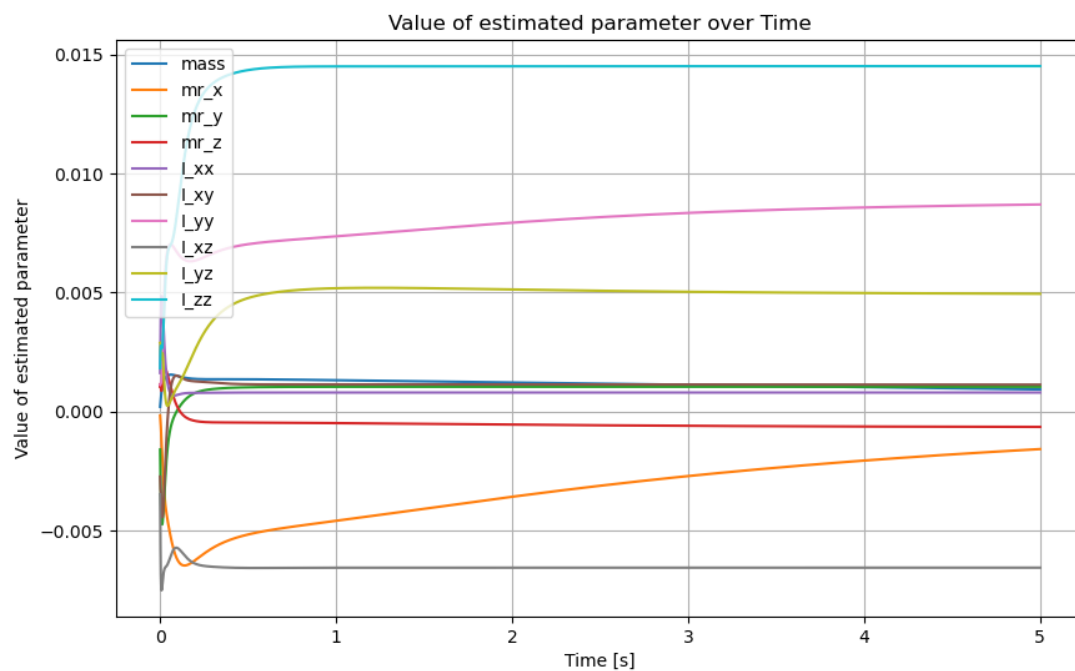


Рисунок 15 – Оценённые параметры последнего звена

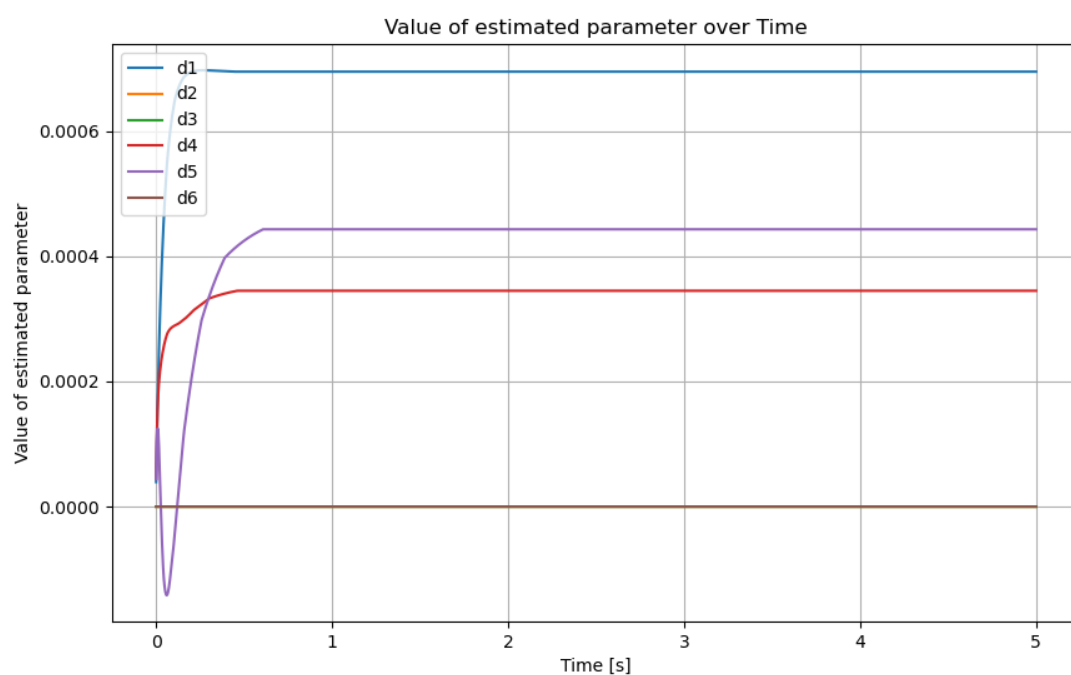


Рисунок 16 – Оценённые параметры демпинга в джоинтах робота