

# Экзаменационная работа

## Робастный контроллер

В случае управления манипулятором чаще всего неизвестны точные параметры системы. В таком случае можно применить робастное управление. Система, включающая демпфирование и кулоновское трение:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + D\dot{q} + F_c(\dot{q}) = \tau$$
$$M(q)\ddot{q} + (C(q, \dot{q}) + D)\dot{q} + g(q) + F_c(\dot{q}) = \tau$$

Систему можно попытаться линеаризовать в управлении при помощи оценок параметров:

$$u = \hat{M}(q) \cdot v + (\hat{C}(q, \dot{q}) + \hat{D})\dot{q} + \hat{g}(q) + \hat{F}_c(\dot{q})$$

Однако, правильнее посчитать, что демпфирование и кулоновское трение полностью неизвестны, так как не всегда есть возможность их оценить или измерить.

При подстановке такого управления в исходную систему мы получаем:

$$\ddot{q} = \underbrace{\tilde{M}^{-1}(q) \cdot (\tilde{C}(q, \dot{q})\dot{q} + \tilde{D}\dot{q} + \tilde{g}(q) + \tilde{F}_c(\dot{q}))}_{f(q, \dot{q})} + \underbrace{\tilde{M}^{-1}(q) \cdot \hat{M}(q)}_{B(q)} v$$

, где параметры с волной – разность оценки и истинного значения.

Произведя указанную замену, мы получаем уравнение вида:

$$\ddot{q} = f(q, \dot{q}) + B(q) \cdot v$$

Далее можно ввести скользящую поверхность. Так как в нашем случае система второго порядка, уравнение скользящей поверхности будет выглядеть следующим образом:

$$s = \dot{\tilde{q}} + L \cdot \tilde{q}$$

матрица  
коэффициентов

Теперь для определения управления необходимо воспользоваться сходимостью по Ляпунову. В качестве кандидата можно выбрать:

$$V = \frac{1}{2} \|s\|^2$$

Производная по времени:

$$\begin{aligned} \dot{V} &= s^T \dot{s} = s^T (\ddot{\tilde{q}} + L \dot{\tilde{q}}) = s^T (\ddot{q}_d - \ddot{q} + L \dot{\tilde{q}}) = \\ &= s^T (\ddot{q}_d - f(q; \dot{q}) - B(q) \cdot v + L \dot{\tilde{q}}) \end{aligned}$$

Чтобы производная была всегда меньше 0:

$$\ddot{q}_d - f(q; \dot{q}) - B(q)v + L \dot{\tilde{q}} = -k \frac{s}{\|s\|}$$

$$B(q)v = \ddot{q}_d - f(q; \dot{q}) + L \dot{\tilde{q}} + k \frac{s}{\|s\|}$$

Если же оценки достаточно близки, то:

$$B(q) \rightarrow I; \quad f(q; \dot{q}) \rightarrow 0$$

В таком случае:

$$v = \underbrace{\ddot{q}_d + L \dot{\tilde{q}}}_{v_n} + \underbrace{k s / \|s\|}_{v_s}$$

Из условия сходимости с определенной скоростью можно вывести условие, налагаемое на коэффициент:

$$\frac{1}{2} \frac{d}{dt} \|s\|^2 = s^T \dot{s} \leq -\eta \|s\|$$

$$\begin{aligned} \dot{s} &= \ddot{q} + L \dot{q} = v_n - \ddot{q} = v_n - \frac{d}{dt} f(q; \dot{q}) - B(q) \cdot (v_n + v_s) = \\ &= \underbrace{(I - B) \cdot v_n - f}_{w} - B v_s \end{aligned}$$

↓

$$s^T w - s^T B(q) v_s \leq \|s\| \|w\| - s^T B(q) v_s \leq -\eta \|s\|$$

$$v_s = \frac{k}{\sigma_{\max}} \hat{M}^{-1} \frac{s}{\|s\|}$$

↑  
максимальное сингулярное число матрицы

$$\|s\| \|w\| - s^T B(q) v_s \leq \|s\| \|w\| - \frac{k}{\sigma_{\max}} s^T M^{-1} s \leq \|s\| \|w\| - k \|s\| \leq -\eta \|s\|$$

$$\underline{k > \|w\| + \eta} \quad - \text{тогда сигнал } s \text{ будет сходиться}$$

Конечное управление представляет собой:

$$\begin{cases} u = \hat{M}(q) \cdot v + (\hat{C}(q; \dot{q}) + \hat{D}) \dot{q} + \hat{g}(q) + \hat{F}_c(\dot{q}) \\ v = \ddot{q}_d + L \dot{q} + v_s \\ v_s = \frac{s}{\|s\|} \\ s = \dot{q} + L \tilde{q} \end{cases} \quad s = \frac{\hat{M}^{-1}}{\sigma_{\max}}$$



Опять же, можно посчитать, что трение нам не известно, и не включать его в управление.

Главным недостатком такого управления является чаттеринг, который возникает из-за быстрой смены знака  $v_s$ . Чтобы избавиться от него можно вычислять  $v_s$  по системе:

$$v_s = \begin{cases} s \frac{s}{\|s\|}, & \|s\| > \epsilon \\ s \frac{s}{\epsilon}, & \|s\| \leq \epsilon \end{cases}$$

Тогда при правильно подобранном  $\epsilon$ , система не будет колебаться.

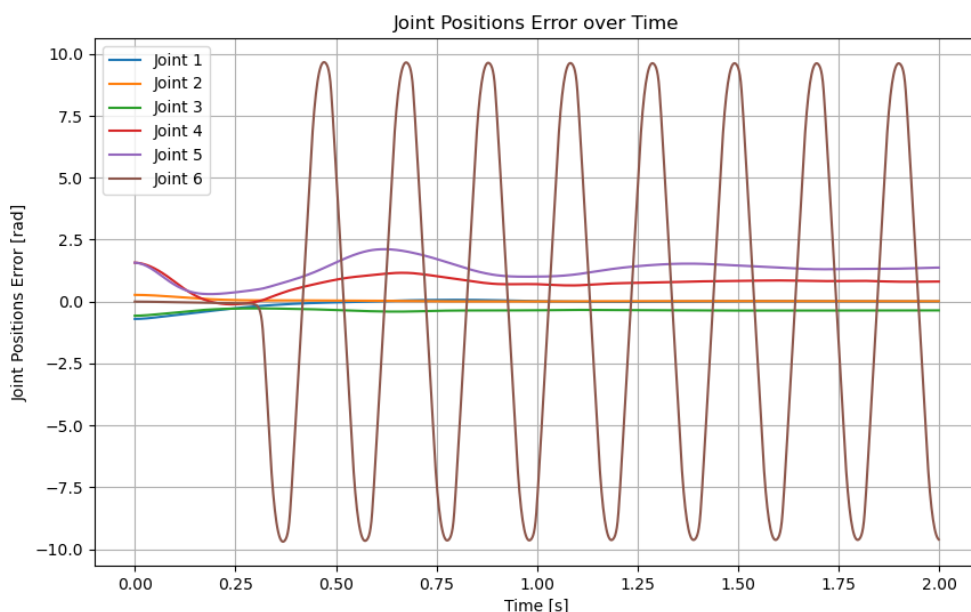
### Сравнение робастного контроллера с ПД

В python с использованием библиотеки Pinocchio и симулятора MuJoCo было написано робастное управление манипулятором, также был написан классический ПД контроллер.

В симуляторе было задано трение и масса эндефектора.

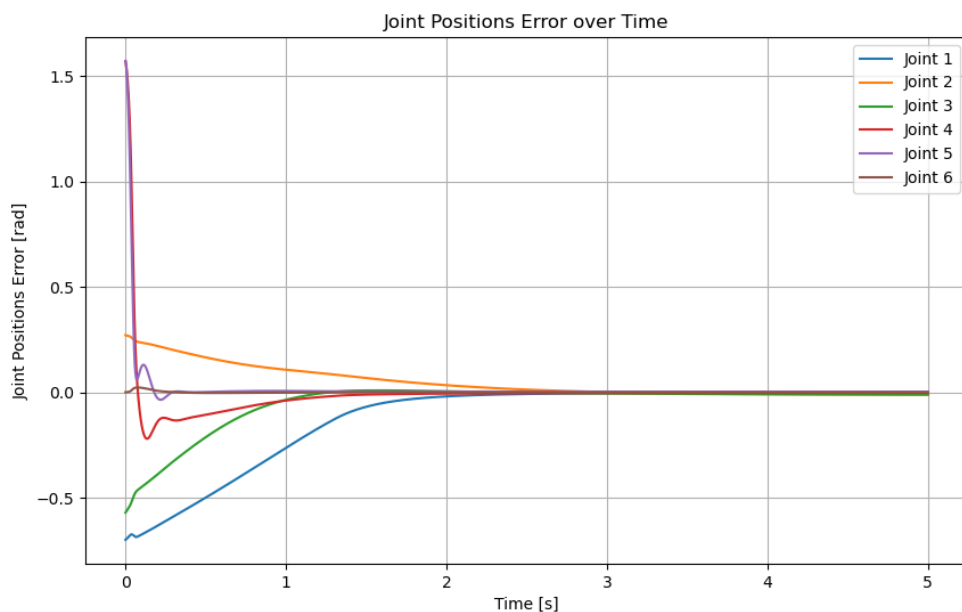
Исходный код, видео симуляции и графики представлены в репозитории.

Из результатов видно, что ПД контроллера совсем не справляется с управлением системой:



Ошибка не сходится к 0, а последний джоинт вообще совершает колебания с большой частотой.

При этом в случае робастного управления получается следующий результат:



Видно, что ошибка управления сходится в ноль.

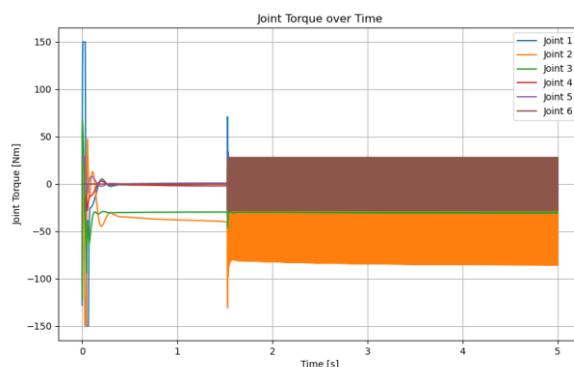
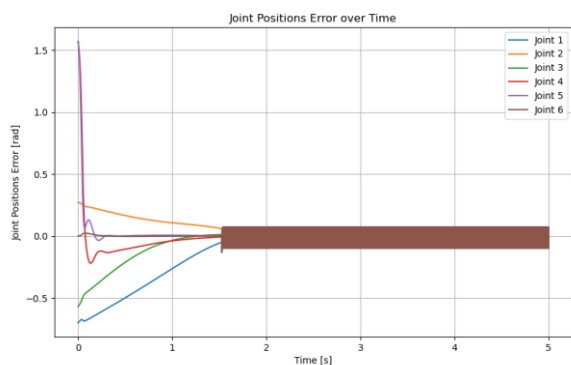
Параметры робастного контроллера подбирались вручную, согласно выведенному неравенству.

Параметр  $\varepsilon$  определялся таким образом, чтобы исключить чаттеринг.

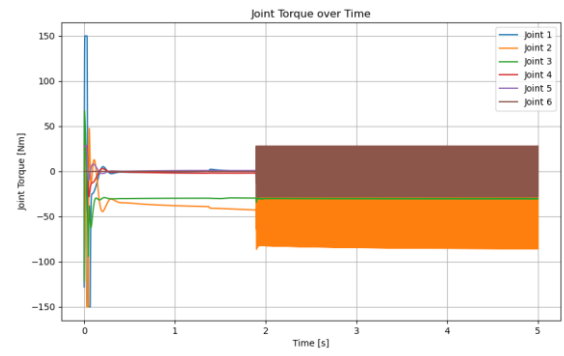
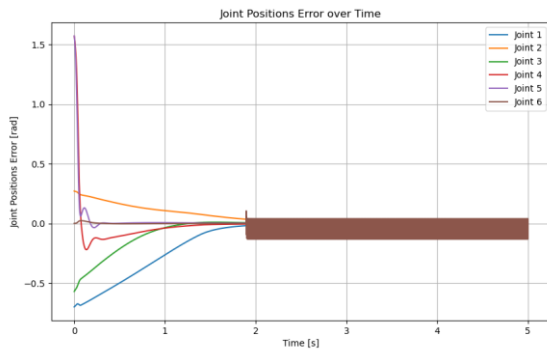
В моем случае пороговое значение, про котором пропадал чаттеринг -  $\|S\| = 55$ .

Графики ошибки и подаваемого управления для разных  $\varepsilon$ :

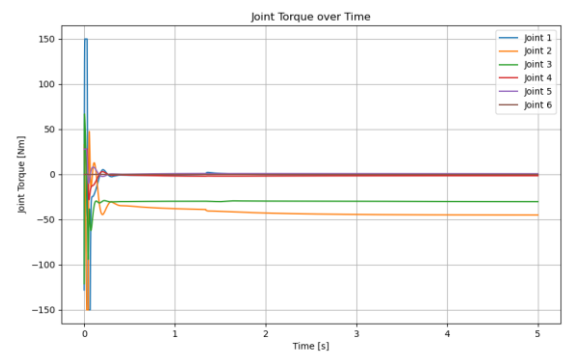
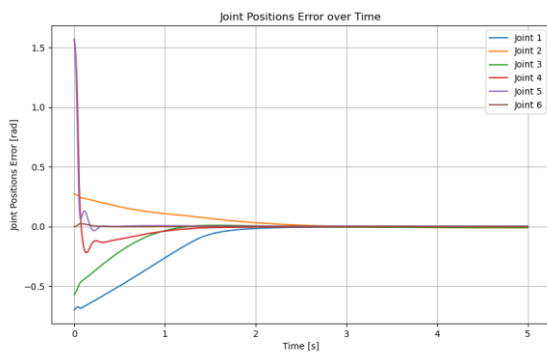
$$\varepsilon = 30$$



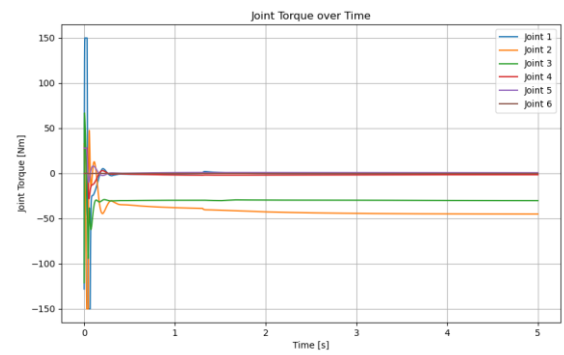
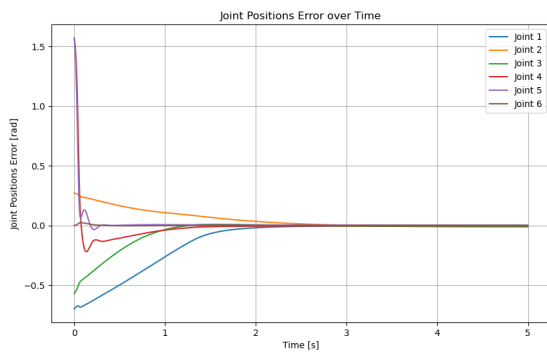
$$\varepsilon = 50$$



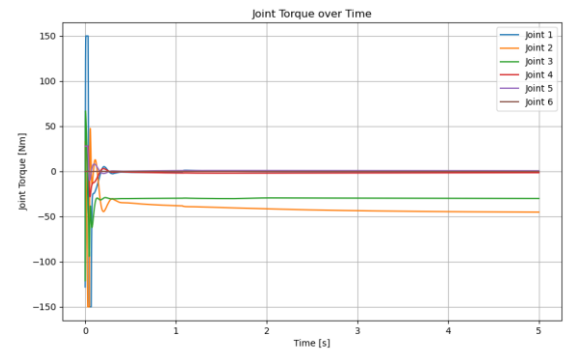
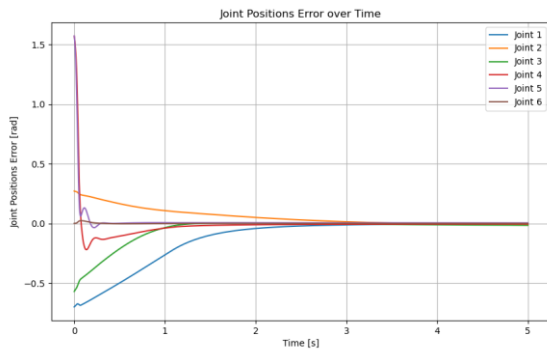
$$\varepsilon = 55$$



$$\varepsilon = 60$$



$$\varepsilon = 100$$



Вывод: При правильно подобранных параметрах робастный контроллер способен управлять системой с неизвестными точно параметрами, что выгодно отличает его от обычного ПД контроллера.