

## Отчёт о выполнении финального экзамена

1. На приложенных ниже снимках представлен вывод контроллера в скользящем режиме. Контроль  $\tau$  задаётся как обычный контроль обратной динамики с номинальными значениями параметров  $M(q)$ ,  $C(q, \dot{q})$  и  $g(q)$ :  $\tau = \hat{M}(q)v + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q)$ . Подставляя данный контроль в уравнение динамики, можно выразить  $\ddot{q}$ .

Уравнение скользящей плоскости задаётся как  $s = \left(\frac{d}{dt} + \lambda\right)\tilde{q} = \dot{\tilde{q}} + \lambda\tilde{q}$ .  $v$  задаётся как сумма  $v_s$  (скользящего фид-форварда) и  $v_n$  (номинального).  $v_n = \ddot{q}_d + \lambda\dot{\tilde{q}}$ . Таким образом  $v = v_n + v_s = v_s + \ddot{q}_d + \lambda\dot{\tilde{q}}$ .

2. В [репозитории](#) на Github находится код, реализующий робастное управление при помощи библиотек MuJoCo и Pinocchio.

Сначала к изменённой модели робота с изменёнными весами был применён обычный контроль, основанный на обратной динамике, который не смог привести систему к желаемому состоянию, сильно отклонившись от него. В папке logs/plots представлены графики ошибки, контроля и фазовый портрет всех для всех шарниров.

В свою очередь робастное управление привело систему к желаемому состоянию, но имело тряску в шарнирах, что в настоящем роботе может привести к серьёзным проблемам в двигателях вплоть до их поломки.

3. Тряска в шарнирах при применении робастного управления возникает из-за того, что контроллер в скользящем режиме ( $v_s$ ) при приближении к нулю начинает резко менять свой знак и значения, что приводит к резкому изменению значения контроля  $\tau$  на противоположное или сильно отличающееся. На практике это может привести к очень быстрой поломке двигателя из-за резких перепадов напряжения в моторах. Для исправления этой проблемы можно задать условие, что если норма вектора  $s$  меньше определённой границы  $\epsilon$  ( $\|s\| \leq \epsilon$ ), то норма заменяется на это значение  $\epsilon$ :

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

Чем больше это самое значение  $\epsilon$ , тем медленнее и менее точно будет сходиться к нулю ошибка положения, а если значение будет слишком низким, то тогда проблема с тряской сохранится. В папке logs/plots хранятся графики, показывающие разные параметры

системы при разных значениях  $\epsilon$ , а также в папке logs/videos можно найти записи работы системы для разных значений.