Редактирование статьи

he_projectile Никогда не публиковалось

Ещё один PID-контроллер.Теория

В сегодняшней статье расскажу, как на стенде измеряется угол, чем обеспечивается защита от дурака, и как мне помог ChatGPT.

Итак, рабочая часть стенда представляет собой луч: неуравновешенную балку с мотором и датчиком. Когда луч находится в исходном состоянии — висит вертикально вниз. Тяга пропеллера создаёт момент силы, вращающий луч против часовой стрелки. Примем это направление за положительное, ограниченное углом 170°. В этом диапазоне сила тяжести всегда вращает луч в отрицательном направлении.

Измерение угла

Первостепенная задача — измерение угла отклонения луча от исходной позиции. Затем нужно проверить, что луч вращается в вертикальной плоскости для предотвращения включения стенда в неправильном положении или выключения мотора при заваливании. Первую задачу можно решить при помощи энкодера на оси, а вторую — административно. Но у квадрокоптера нет энкодеров, а аварийные ситуации случаются произвольно. Поэтому буду использовать инерциальный датчик MPU6050 в модуле GY-521.

В одной из статей я описал способ определения ориентации инерциального датчика. Ориентация представляется в виде матрицы 3×3, в которой столбцы соответствуют осям датчика, а строки — проекциям на север, запад и вертикаль.
Казалось бы, достаточно прикрепить датчик к балке, посмотреть направление оси $\vec{z}$, запомнить направление $\vec{y}$, после поворота посчитать угол между прежним и текущим направлением. Но такой вариант доступен только при полной уверенности в том, что ось датчика $\vec{z}$ параллельна оси луча, а я не стану гарантировать это.

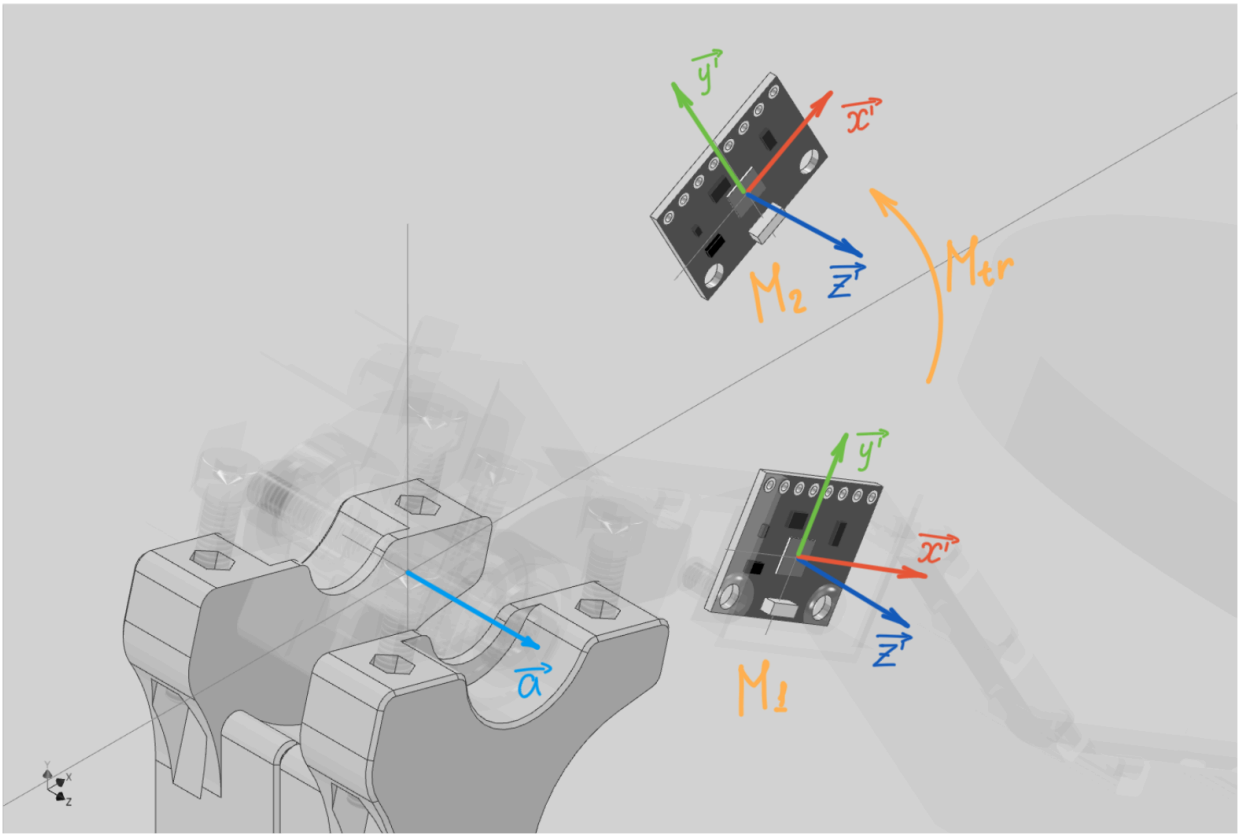

Рисунок 1. Поворот датчика вокруг оси

Поэтому надо обобщить обе операции для случайного закрепления датчика. Как и раньше, алгоритм прототипируется в матлабе. МК обрабатывает данные датчика, вычисляет матрицу и отправляет её компьютеру по USB-CDC в виде массива из 9 байт. При запуске стенда луч неподвижнен. Программа запоминает изначальную ориентацию датчика (M_1). Матрица вращается вокруг вектора $\vec{a}$, и получается матрица M_2 . Этот вектор и является осью вращения, надо вычислить его. Что мы знаем об $\vec{a}$? Его проекции на M_1 и M_2 одинаковы, при этом матрицы связаны матрицей перехода:

$$M_{tr} = M_2 \cdot M_1^{-1} \qquad (1)$$

То есть, при применении к $\vec{a}$ линейного оператора M_{tr} снова выходит $\vec{a}$. Получается, что $\vec{a}$ является собственным вектором для матрицы перехода. Так как размерность матрицы 3, то и собственных векторов тоже 3. Как выбрать нужный? Сперва надо посчитать собственные числа M_{tr} . Собственное число - это скаляр, который растягивает собственный вектор. При повороте не происходит растяжения вектора-оси, или же происходит растяжение с коэффициентом 1. Значит, нужно выбрать вектор, для которого $\lambda = 1$.

У этого решения есть несколько нюансов. Во-первых, погрешности измерений, расчётов и неровность механики приводят к тому, что M_{tr} не в полной мере является ортогональной матрицей, и $\lambda \neq 1$. Поэтому приходится выбирать число, наиболее близкое к 1. Во-вторых, вычисление собственных чисел матрицы — трудоёмкая задача как для МК, так и для программиста.
Прежде чем кинуться в написание сишных функций, я обратился за советом к ChatGPT и не пожалел.

Совет 1. Транспонировать матрицу вместо вычисления обратной. Для ортогональной матрицы, к которой относится моя матрица ориентации, $M^{-1} = M^T$.
Транспонировать матрицу проще, и это снизит нагрузку на работу МК.

Совет 2. Использовать след матрицы поворота для вычисления угла.

$$tr(M_{tr}) = M_{tr(11)} + M_{tr(22)} + M_{tr(33)} = 1 + 2cos(\psi) \qquad (2)$$

$$\psi = acos\left(\frac{tr(M_{tr}) - 1}{2}\right), \qquad (3)$$

где ψ - угол поворота матрицы поворота.

Из-за погрешностей измерений и вычислений значение $tr(M_{tr})$ может выходить за границы $[-1; 1]$, что вызовет ошибку вычисления арккосинуса, поэтому необходимо проверять это условие и обрезать лишнее. Кто-то скажет, что это заметание проблемы под ковёр, и я соглашусь. Но так как в МК сложно реализовать перехват исключений, то я решил проверять ортогональность и матрицы ориентации отдельно.
Данный метод возвращает θ при $0 \leq \theta < \pi$ и $2\pi - \theta$ при $\pi \leq \theta < 2\pi$, однако это не проблема для моего стенда, так как отклонение луча ограничено.

Совет 3. Извлекать ось вращения из антисмимметричной части матрицы поворота.

$$\overrightarrow{a} = \frac{1}{2sin(\psi)} \cdot \begin{bmatrix} M_{tr(32)} - M_{tr(23)} \\ M_{tr(13)} - M_{tr(31)} \\ M_{tr(21)} - M_{tr(12)} \end{bmatrix} \qquad (4)$$

Таким образом, вычисления угла и оси стали быстрыми и непринуждёнными. При запуске стенда необходимо перевести луч из нижнего положения в положение повыше. Из полученных данных будет вычислен вектор оси. Угол между осью и вертикалью

должен иметь значение, близкое к $\pi/2$. Сильное отклонение приведёт к переходу стенда в аварийное состояние и отключению мотора до перезапуска программы.

Расчёт параметров модели

Прежде чем угробить стенд жёсткими режимами работы, хотелось бы поиграть в математическое моделирование. Может быть, я получу полезный практический опыт или даже готовые коэффициенты PID. Для этого составлю уравнение динамики стенда, напишу PID-реулатор для управления тягой, абстрагируюсь от задержки измерения угла и изменения тяги, инерции пропеллера.С

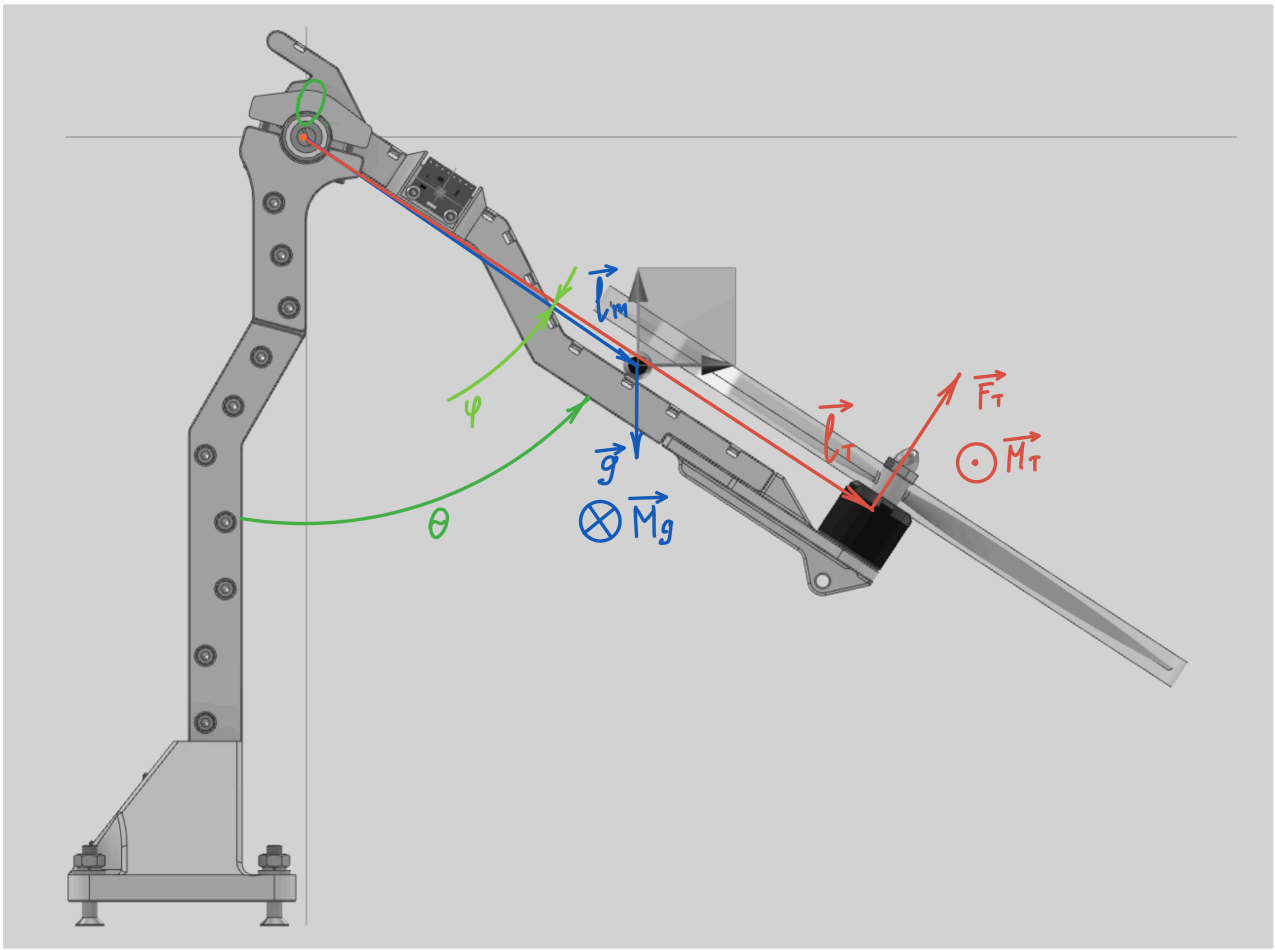

Рисунок 2. Схема действующих сил

Сила тяжести, действующая на луч и приложенная к центру масс луча, создаёт момент, равный

$$\overrightarrow{M_g} = \overrightarrow{l_g} \times m\overrightarrow{g}$$

или

$$M_g = m \cdot g \cdot l_g \cdot sin(\theta - \varphi), \qquad (5)$$

где m — масса луча, l_g — расстояние от оси O до центра масс луча, θ — угол отклонения луча, φ — угол между лучом и вектором до центра масс.

Так как отклонение луча ограничено механически 170°, направление момента не меняется.
Уравнение статики для этой системы выглядит так:

$$\overrightarrow{M_g} + \overrightarrow{M_t} = 0$$

или

$$0 = F_T \cdot l_T - m \cdot g \cdot l_m \cdot sin(\theta - \varphi), \qquad (6)$$

где l_T — расстояние между осью луча и осью пропеллера, F_T — сила тяги пропеллера.

При дисбалансе сил возникает угловое ускорение:

$$\ddot{\theta} = \frac{F_T \cdot l_T - m \cdot g \cdot l_m \cdot sin(\theta - \varphi)}{J}, \qquad (7)$$

где J - момент инерции луча.

Необходимо добавить сопротивления среды. К ним можно отнести силу трения покоя в подшипнике, силу сухого трения подшипника, силу вязкого сопротивления смазки и силу сопротивления воздуха. Силу сухого трения и трения покоя я исключу, так как хорошо смазал подшипники. Включим оставшиеся силы и получим уравнение динамики системы:

$$\ddot{\theta} = \frac{F_T \cdot l_T - m \cdot g \cdot l_m \cdot sin(\theta - \varphi) - k_v\dot{\theta} - k_a\dot{\theta}^2 \cdot sign(\dot{\theta})}{J}, \qquad (8)$$

где k_v - коэффициент вязкого сопротивления смазки, k_a - коэффициент сопротивления воздуха.

Момент инерции и расстояние до центра масс вычислены в САПРе. Для этого я взвесил каждую деталь и прописал массу в физические свойства модели. Несмотря на то, что равномерность распределения массы по объёму является большим допущением для деталей, напечатанных на FDM-принтере, результат оказался впечатляюще удовлетворительным.
k_v и k_a - эмпирические коэффициенты. Я получил их значения экспериментально.

Для этого закрепил мачту горизонтально, отвёл луч на произвольные 39 ° , отпустил и позволил ему свободно колебаться под действием силы тяжести и получил функцию затухающих колебаний в табличном виде $\theta_{exp}(t)$. При этом дифференциальное уравнение обретает вид:

$$\begin{cases} \ddot{\theta} = \dfrac{-m \cdot g \cdot l_m \cdot sin(\theta - \varphi) - k_v\dot{\theta} - k_a\dot{\theta}^2 \cdot sign(\dot{\theta})}{J} \\ \theta(0) = -39° \\ \dot{\theta} = 0 \end{cases} \qquad (9)$$

Затем написал MATLAB-скрипт, численно решающий приведённое уравнение с такими же начальными условиями, как у $\theta_{exp}(t)$. Обозначу вычисленные данные как $\theta_{sim}(t)$. Оба графика представлены на рисунке 3.

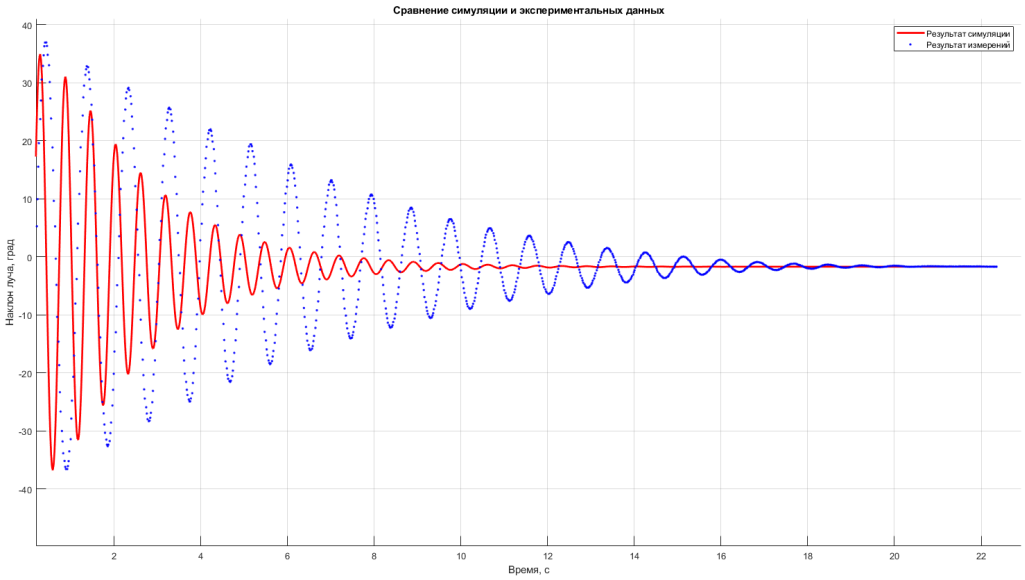

Рисунок 3. Это фиаско

Как можно увидеть, частоты значительно различаются, примерно в два раза. Она зависит от параметров l_m и J . Подобрав ручками J , я понял, что увеличение его вдвое исправляет проблему. После проверки масс всех деталей, я вспомнил, что САПР указывает момент инерции относительно центра масс. Чтобы пересчитать значения J для вращения вокруг O , нужно применить теорему Гюйгенса-Штейнера:

$$J = J_C + m \cdot l_m^2, \qquad (10)$$

где J_C - момент инерции, рассчитанный САПРом.
Отклонение частоты уменьшилось до 3%, что видно на рисунке 4.

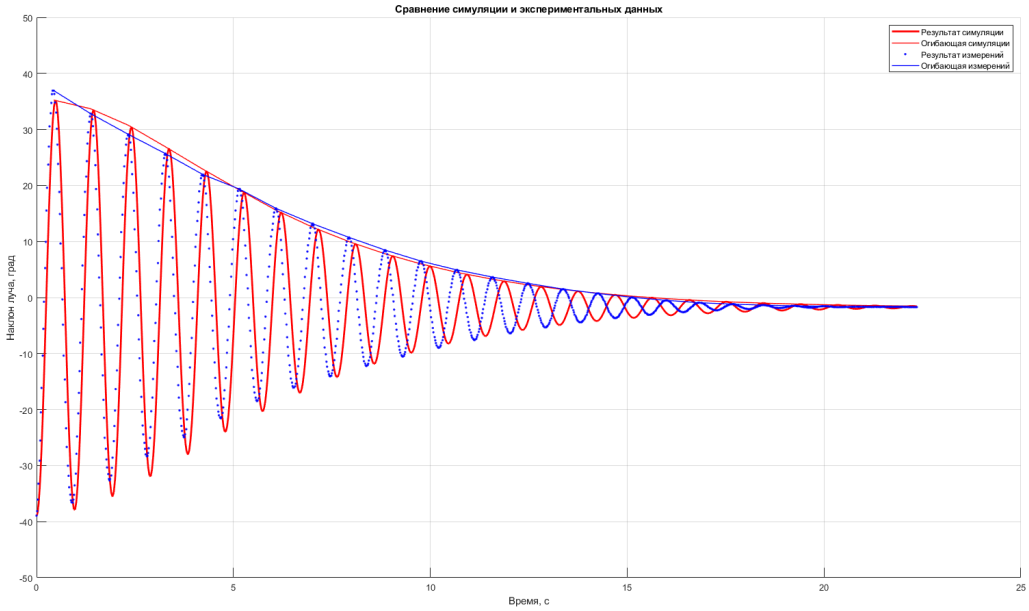

Рисунок 4. Графики отлично накладываются

Теперь можно подобрать k_v и k_a . Так как диффур исчерпывающий, можно надеяться, что при правильном выборе k_v и k_a графики совпадут. Хорошо бы сделать подбор автоматическим и оптимальным по времени. Сперва нужно выбрать критерий совпадения графиков. Сумма квадратов невязок отлично подходит на роль меры.

$$RSS(k_v, k_a) = \sum_{i=0}^{N} (\theta_{sim}(t_i, k_v, k_a) - \theta_{exp}(t_i))^2 \qquad (11)$$

Но вот незадача. Временные метки у $\theta_{exp}(t)$ и $\theta_{sim}(t)$ различаются, выбирать пары значений θ с одинаковым аргументом не получится. К тому же, частоты незначительно различаются, на шестнадцатом периоде колебания сдвинуты на $\pi/2$, поэтому RSS будет меняться хаотично при переборе коэффициентов.
Однако известно, что k_v и k_a характеризуют форму огибающей и практически не влияют на период колебаний. Значит, достаточно сравнивать огибающие этих графиков. Для этого из $\theta_{exp}(t)$ и $\theta_{sim}(t)$ выделяются пики, обозначенные $\gamma_{exp}(t)$ и $\gamma_{sim}(t)$. Затем надо интерполировать $\gamma_{sim}(t)$ сплайном, подставить временные метки из $\theta_{exp}(t)$ и получить соответствующие значения $\gamma_{simInt}(t)$. Теперь можно сравнить огибающие. Таким образом, вычисление k_v и k_a сводится к минимизации функции RSS:

$$RSS(k_v, k_a) = \sum_{i=0}^{N} (\gamma_{simInt}(t_i, k_v, k_a) - \gamma_{exp}(t_i))^2 \to \min_{k_a, k_v \in R^2} \qquad (12)$$

Я воспользовался готовой функцией fmincon из пакета Optimization Toolboxt. С начальным приближением $k_v = 0, k_a = 0$ решение заняло 16 итераций. Результат можно видеть на рисунке 4, а полученные значения в таблице ниже.

J	кг · м ^{2}	5,52e-3
m	кг	0,181
l_m	м	0,139
ψ	град	1,71
k_v	попугаи	1,85e-3
k_a	попугаи	1,06e-3

Симуляция PID-регулятора

Наконец, можно приступить к моделированию PID-регулятора. Ограничу задачу достижением и поддержанием угла луча, но в будущем попробую также регулировать скорость вращения луча.

Итак, формула PID-регулятора:

$$\begin{cases} u(t) = K_P \cdot e_{rr}(t) + K_I \cdot \int_0^t e_{rr}(\tau)\, d\tau - K_D \cdot \dfrac{de_{rr}(t)}{dt} \\ e_{rr}(t) = \theta_{set} - \theta(t) \end{cases}, \qquad (13)$$

где θ_{set} - установочное значение, к которому стремится $\theta(t)$, $e_{rr}(t)$ - ошибка регулирования, $u(t)$ - управляющее воздействие, стремящееся уменьшить ошибку. K_P, K_I, K_D - коэффициенты при пропорциональной, интегральной и дифференциальной составляющих регулятора соответственно.

Значения коэффициентов и определяют работу регулятора, их и предстоит подобрать. Сперва необходимо врезать в диффур PID-регулятор.

$$\begin{cases} \ddot{\theta}(t) = \frac{F_{max} \cdot u(t) \cdot l_T - m \cdot g \cdot l_m \cdot sin(\theta(t) - \varphi) - k_v \dot{\theta}(t) - k_a \dot{\theta}^2(t) \cdot sign(\dot{\theta}(t))}{J} \\ u(t) \in [0,\ 1] \\ \dot{\theta}(0) = 0 \\ \theta(0) = \theta_0 \end{cases}, \qquad (14)$$

где F_{max} - максимальная тяга, которую может выдать мотор. θ_0 - начальный угол.

Регулятор управляет тягой, которая может быть только положительной и имеет ограничение сверху. Поэтому удобно ограничить также управляющий параметр $u(t) = [0; 1]$. Для решения диффура необходимо принять начальные условия $\dot{\theta}(0)$ и $\theta(0)$. Регулятор начинает работу при неподвижном опущенном луче, поэтому они равны нулю.

Это дифференциальное уравнение не решается аналитически. Хорошо, что MATLAB может решить его численно. Скрипт выглядит так:

```
% Параметры
l_m = norm([138.760 -4.145 ])/1000*1; % Расстояние до центра масс, м
m = 0.181;    % Масса, кг
psi = atan2(-4.145, 138.760); % Угол между балкой и вектором до центра масс, рад
J = 2036.662 /10^6+m*l_m^2; % Момент инерции, кг*м^2
g = 9.81; % м/с^2
kv = 0.0001850; % Сопротивление смазки, попугаи
ka = 0.0001058; % Сопротивление воздуха, попугаи
l_T = 0.254; % Плечо пропеллера, м
F_T=10; % Максимальная тяга пропеллера, Н
alpha = deg2rad(90); % Цель

% Начальные условия
O0 = deg2rad(0);    % Начальный угол
omega0 = 0;   % Начальная угловая скорость
integral0 = 0;
Y0 = [O0; omega0; integral0];

% Время моделирования
tspan = [0 10];

% Решение диффура
[t, Y] = ode45(@(t, Y) pendulumODE(t, Y, l_m, m, g, J, psi, kv, ka, F_T, l_T, al
Y0);

% Графики
figure(1); % Создаю график с номером 1 / Обращаюсь к графику 1
clf(figure(1)) % Чищу график 1

plot(t, rad2deg(Y(:,1)), 'r', 'LineWidth', 1.5);
```

```
30 xlabel('Время, с');
31 ylabel('Наклон луча, град)');
32 title('Динамика угла');
33 ylim([0 120])
34 grid on;
35
36 yline(rad2deg(alpha))
37
38 function dYdt = pendulumODE(t, Y, l, m, g, J, psi, kv, ka, F_T, l_T, alpha, Kp,
39     if Y(1) >= 0.99*pi
40         Y(2) = 0;
41         Y(1) = 0.99*pi;
42     end
43     O = Y(1);          % Угол O
44     omega = Y(2);      % Угловая скорость dO/dt
45     e = (alpha-O);
46
47     d_integral=e;
48
49     if Y(3)>10
50         Y(3) = 10;
51     end
52     if Y(3)<-10
53         Y(3) = -10;
54     end
55     integral = Y(3);
56
57     U = Kp*e+Ki*integral-Kd*omega+Ks*sin(alpha);
58
59     if U>1
60         U = 1;
61     end
62     if U < 0.001
63         U = 0.001;
64     end
65     F = F_T*U;
66
67     domegadt = (F*l_T-m*g*l*sin(O-psi)-kv*omega*t-ka*omega^2*sign(omega)*t)/J;
68
69     dYdt = [omega; domegadt; d_integral];  % Возвращаем вектор производных
70 end
```

Функция ode45 (строка 23) решает диффур итерациями, поэтому на каждом шаге можно выполнять математические операции над переменными, сравнивать и менять их значения. Это позволяет встроить в диффур различные проверки. Например, на строках 59-64 реализовано ограничение управляющего параметра.

Пора перейти к подбору коэффициентов. Для этого надо выбрать критерии качества процесса регулирования. Их много, но меня интересуют следующие три:

- Время переходного процесса ($T_{ПП}$). Это время, за которое регулируемое значение входит в полосу шириной 2Δ и больше не выходит из неё. Желаемое значение - 0.

- Перерегулирование ($\frac{\theta_{max} - \theta_{33}}{\theta_{33}}$). Желаемое значение - 0.

- Ошибка регулирования ($\frac{\theta_{\infty} - \theta_{33}}{\theta_{33}}$). Желаемое значение - 0.

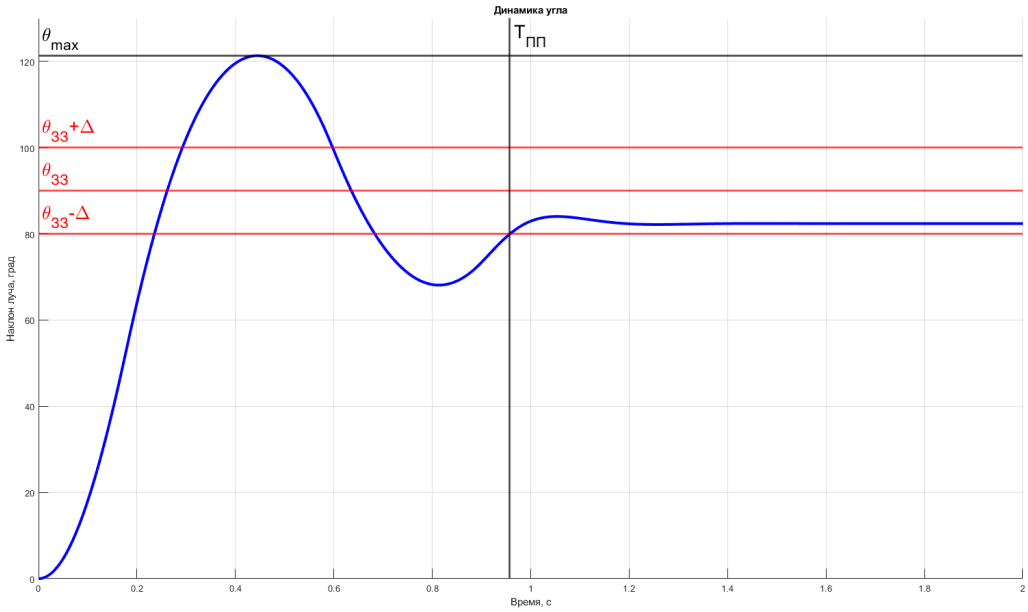

Рисунок 5. Основные критерии качества процесса регулирования

Необходимо подобрать коэффициенты регулятора так, чтобы все три параметра были минимальны. Это непростая задача, ведь добиться идеального переходного процесса нереально. А значит, мне придётся подумать и отдать приоритет одним характеристикам, установив ограничения для других. Но я не знаю, как сравнивать величины, имеющие разные размерности, так почему бы не воспользоваться хорошим методом дважды. Я буду минимизировать отличие моего переходного процесса от идеального.

Идеальный переходный процесс представляет собой ступенчатую функцию:

$$\theta_{ideal}(t) = \begin{cases} 0, & t < 0 \\ \theta_{33}, & t \geq 0 \end{cases}$$

Регулирование начинается при t=0, значит меня интересует только $\theta_{ideal} = \theta_{33}$.Тогда функция меры приобретает следующий вид:

$$err(P, I, D) = \int_0^\infty |\theta_{ideal}(t) - \theta(t, P, I, D)|\, dt,$$

где $\theta(t)$ - результат симуляции, P, I, D - коэффициенты PID-регулятора.
Это называется интегральной оценкой качества переходного процесса.

Таким образом, минимизируя $err(P, I, D)$, можно получить приемлемый результат, который будет протестирован в следующей части.

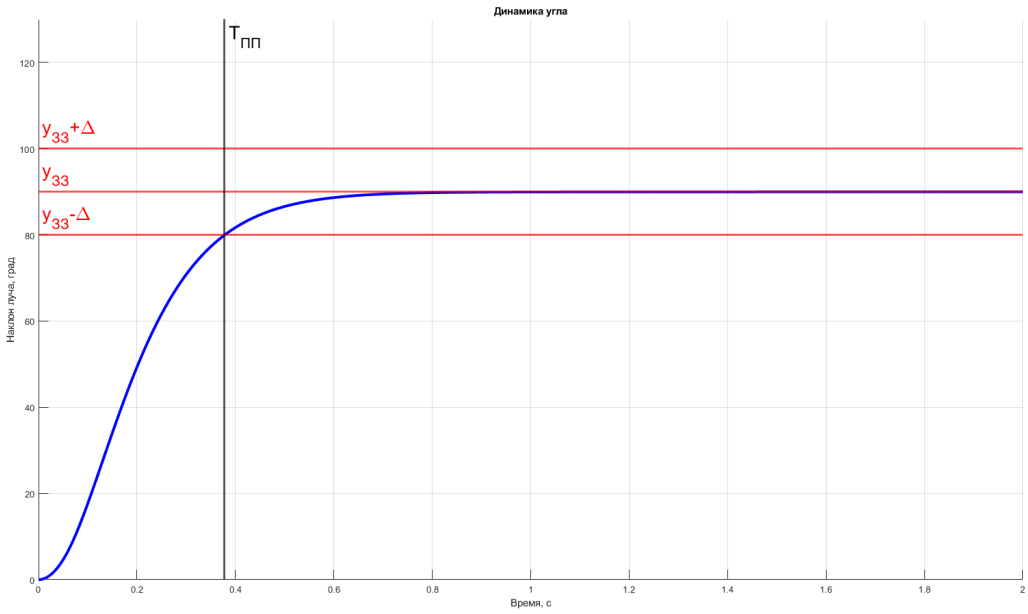

Рисунок 6. Приемлемый процесс регулирования

Нажмите "/" для вызова меню

Далее к настройкам

ХАБР ИЩЕТ АВТОРОВ

Мы ищем авторов в контент-студию — помогать компаниям делать крутые статьи для техноблогов и мегапроекты.

Хотите писать не только на Хабр, но и для Хабра? Давайте познакомимся!

О РЕДАКТОРЕ

Редактор публикаций может работать в режимах WYSIWYG и Markdown.

Переключение осуществляется с помощью иконки в правом верхнем углу.

Подробнее о работе редактора можно узнать в этом разделе.

ПАМЯТКА АВТОРУ

- Соблюдайте правила сайта
- Следуйте советам и заботливо оформляйте публикации
- Используйте хаб «Я пиарюсь» для рекламных публикаций
- Загружайте картинки меньше 8МБ для тела публикации и меньше 1МБ для обложки публикации