

Динамика и управление обратным маятником с маховиком

ТЕОРИЯ И ПРАКТИКА

ДАНИЛОВ ВЛАДИМИР

Оглавление

Введение	2
1. Устройство маятника с маховиком	4
2. Теоретическая часть	5
2.1. Динамика маятника с маховиком	5
2.2. Стабилизация маятника в верхнем неустойчивом положении равновесия	10
2.3. Перевод маятника из нижнего положения равновесия в верхнее	13
3. Практическая часть	13
3.1. Математическое моделирование	14
3.2. Натурные испытания	17
Контрольные вопросы	19
Список литературы	20

Введение

Существуют объекты, у которых при отсутствии управления желаемый режим работы неустойчив. К подобным объектам можно отнести летательные аппараты, у которых центр давления расположен впереди центра масс. Такие летательные аппараты обладают высокой маневренностью, но при этом являются статически неустойчивыми. Для обеспечения устойчивости используется система управления.

Значительные трудности обычно вызывает задача построения управления объектами, в которых число управляющих воздействий меньше числа степеней свободы. В этом случае говорят, что это *объект с дефицитом управляющих воздействий*. В англоязычной литературе такой объект управления называют **under-actuated object**. К таким объектам можно отнести летательные аппараты, большинство автомобилей и обратные маятники [1].

Обратный маятник – это маятник, центр масс которого расположен выше его точки опоры, как правило на конце жёсткого стержня. В то время как обычный маятник устойчиво висит вниз, обратный маятник по своей природе неустойчив и должен постоянно балансироваться чтобы оставаться в вертикальном положении, с помощью системы управления. Существует несколько разновидностей обратных маятников. Рассмотрим некоторые из них.

Классическим примером является маятник на подвижной тележке (рис. 1). Задача состоит в том, чтобы, перемещая тележку вдоль продольной оси, обеспечить установленному на неё стержню устойчивое равновесие. Всем известно, что, перемещая ладонь руки в горизонтальной плоскости, можно удерживать от падения стоящую на ней вертикальную палку. Особенно легко это делать, если палка длинная и её масса сосредоточена на верхнем конце. Имеются также вариации с двумя, тремя и более звеньями (рис. 2).

Другая модель – маятник Фуруты. Здесь устойчивое равновесие обеспечивается путем вращательного движения подвижного основания (рис. 3).

Существуют маятники, у которых основание неподвижно, а управление осуществляется благодаря установленному на конце стержня маховику, приводимому в движение с помощью электродвигателя (рис. 4). Движение такого маятника осуществляется в одной плоскости. Но подобный принцип можно использовать для удержания баланса в нескольких плоскостях. Например, для удержания равновесия подвижного стержня или куба (рис. 5 и 6). Причем такой куб может не только удерживать равновесие, но и перемещаться в пространстве.

Для наглядности, ниже приведен список ссылок на видеозаписи с демонстрацией работы описанных маятников:

- [линейный обратный маятник](#)
- [линейный обратный маятник с тремя звеньями](#)
- [маятник Фуруты](#)
- [обратный маятник с маховиком](#)
- [балансирующий стержень](#)
- [балансирующий куб](#)

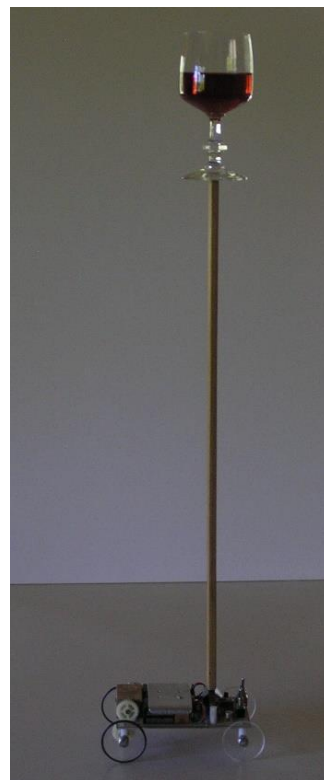


Рисунок 1. Маятник на подвижной тележке.

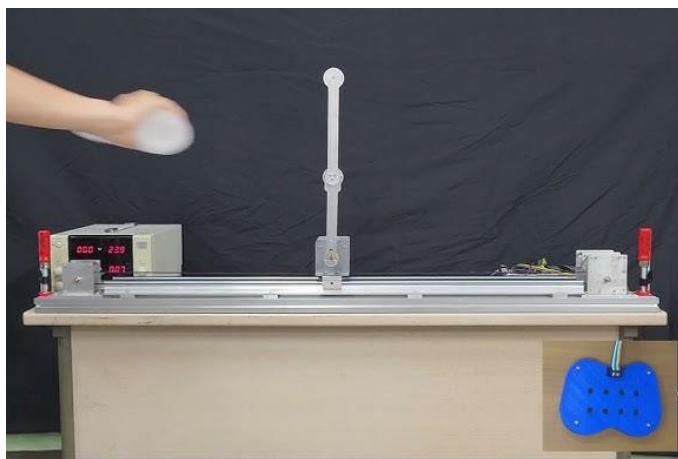


Рисунок 2. Двухзвенный обратный маятник на линейной подвижной платформе



Рисунок 3. Маятник Фуруты.



Рисунок 4. Обратный маятник с маховиком.



Рисунок 5. Балансирующий стержень.

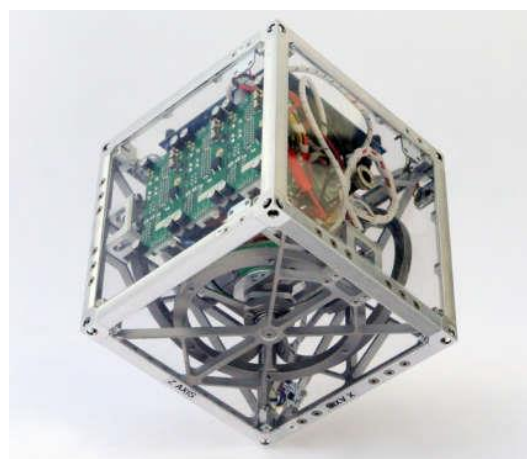


Рисунок 6. Балансирующий куб.

Хотя обратный маятник широко используется в качестве тестового объекта при изучении теории управления, его принципы лежат в основе многих практических устройств. Например, ориентация искусственных спутников земли производится с помощью маховиков, установленных внутри корпуса. Обратные маятники находят применение в таких транспортных средствах, как сегвей, моноколесо и гироскутер. Даже процесс ходьбы можно выразить через модель обратного маятника [2]. При двуногой ходьбе представьте, что стопа опорной ноги является основанием маятника, а корпус – положением центра масс стержня (рис. 7).

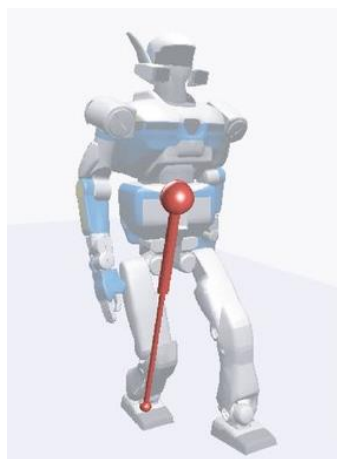


Рисунок 7. Модель трехмерного линейного обратного маятника, используемая в системе управления ходьбы двуногого шагающего робота.

Целью этой практической работы является построение алгоритмов управления маятником с маховиком. Реализуемые желаемые режимы движения – это раскачивание маятника с переводом его в верхнее неустойчивое положение равновесия и его стабилизация в этом положении. Законы изменения управляющего воздействия, при которых реализуется желаемое движение, синтезируются в виде обратной связи.

При выполнении этой работы, предполагается, что вы уже установили и настроили всё необходимое ПО по инструкции из файла *Tutorials/Ru_Инструкция_по_установке.docx*. Также предполагается, что вы уже обладаете базовыми знаниями в теоретической механике и теории автоматического управления. Если у вас возникнут сложности с пониманием материала, обратитесь к учебникам [3] и [5]. Также необходимы навыки работы с Arduino IDE при выполнении практической части работы.

1. Устройство маятника с маховиком

На рис. 8 показан однозвенный маятник 4 с маховиком 1. Маятник способен совершать круговые движения в вертикальной плоскости. Его ось вращения 5 расположена на неподвижном основании. В этой оси отсутствует какой-либо привод. Сам маятник 4 также часто называют стержнем.

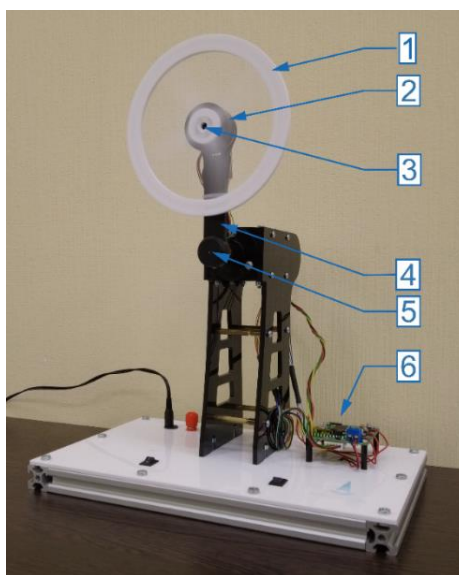


Рисунок 8. Основные компоненты маятника с маховиком. 1 – маховик, 2 – электродвигатель, 3 – ось маховика, 4 – стержень, 5 – ось стержня, 6 – плата управления.

Ось вращения маховика 3 закреплена на стержне и параллельна его оси вращения. Маховик приводится во вращение бесколлекторным электродвигателем 2. Электропривод, как и маховик, смонтирован на стержне. Статор двигателя жестко закреплен на стержне, ось ротора жестко соединена с осью маховика. Плата управления 6 содержит микроконтроллер, драйвер двигателя и трехфазный усилитель мощности. При управлении используется информация об углах поворота стержня относительно неподвижного основания и маховика относительно стержня. Эти углы измеряются магнитными датчиками углового положения, которые подключены к плате.

2. Теоретическая часть

2.1. Динамика маятника с маховиком

На рис. 9 приведена схема маятника. Стержень OB в точке O шарнирно соединён с неподвижной опорой. Ось шарнира перпендикулярна плоскости качания маятника. Симметричный относительно своей оси вращения маховик смонтирован на маятнике так, что его центр расположен на конце стержня в точке B . Маховик может поворачиваться в ту или иную сторону вокруг проходящей через точку B горизонтальной оси, перпендикулярной плоскости качания стержня. Эта ось параллельна оси шарнира O . Ось вращения маховика является продолжением оси ротора электродвигателя. Все необходимые обозначения и параметры маятника приведены в таблице 1. Система имеет две степени свободы. Единственное, с помощью чего можно управлять системой – это момент, вырабатываемый электродвигателем.

Вращение двигателем осуществляется с помощью алгоритма векторного управления. Мы будем подавать на вход управления двигателем желаемое напряжение (т.н. квадратурное напряжение в терминах векторного управления), тем самым имитируя поведение коллекторного двигателя. Соответственно, управляющим параметром для системы управления маятником тоже будет напряжение. Чтобы преобразовать желаемый момент в напряжение, в модель маятника нужно будет дополнительно включить модель коллекторного двигателя.

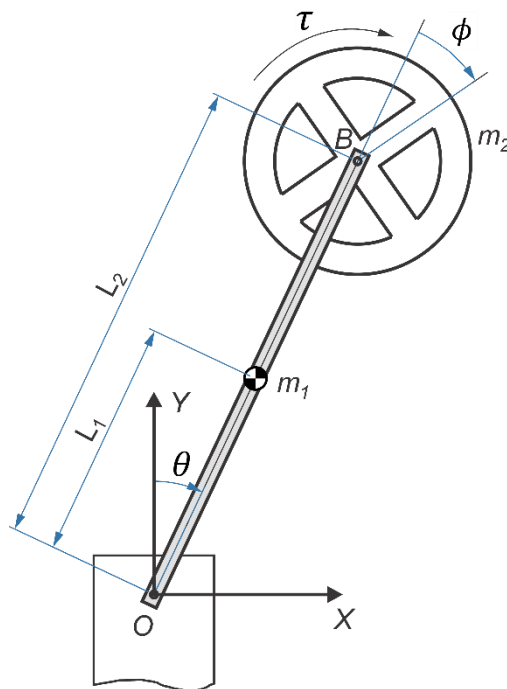


Рисунок 9. Схематичное изображение обратного маятника с маховиком.

Таблица 1. Параметры маятника.

Параметр	Ед. изм.	Описание
L_1	м	Расстояние от точки O до центра масс стержня
L_2	м	Расстояние от точки O до маховика
m_1	кг	Масса стержня
m_2	кг	Масса маховика
θ	рад	Угол поворота стержня
ϕ	рад	Угол поворота маховика
I_1	кг/м ²	Момент инерции стержня
I_2	кг/м ²	Момент инерции маховика
g	м/с ²	Ускорение свободного падения
τ	Н м	Момент, вырабатываемый электродвигателем
μ_p	-	Коэффициент вязкого трения в шарнире O
μ_m	-	Коэффициент вязкого трения между статором и ротором двигателя

Динамическую модель будем строить при помощи метода Лагранжа. Он является одним из классических подходов к получению уравнений движения механических систем с помощью уравнений Лагранжа второго рода. В отличие от второго закона Ньютона, метод Лагранжа оперирует энергетическими величинами: кинетической и потенциальной энергией.

Суть метода заключается в следующем: для системы с n обобщёнными координатами $\eta_1, \eta_2, \dots, \eta_n$ строится так называемый **лагранжиан** $L = T - U$, где T — кинетическая энергия, U — потенциальная энергия системы. Далее, для каждой координаты η_i записывается уравнение Лагранжа второго рода:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\eta}_i} \right) - \frac{\partial L}{\partial \eta_i} = Q_i$$

где Q_i — обобщённая сила.

Подробно изучить принципы построения динамических уравнений можно в книге [3]. Напомним общий алгоритм построения модели методом Лагранжа:

1. Выбрать обобщённые координаты — минимальный набор независимых переменных, полностью описывающих положение системы.
2. Найти обобщённые силы
3. Вычислить кинетическую энергию T , как сумму энергий всех подвижных масс, включая вращательное движение.
4. Вычислить потенциальную энергию U , которая обычно зависит от положения в поле тяжести.
5. Построить лагранжиан $L = T - U$.
6. Применить уравнения Лагранжа второго рода для получения системы дифференциальных уравнений с учетом обобщённых сил Q_i при наличии внешнего управления или трения.

Метод Лагранжа является мощным инструментом для моделирования динамики, особенно в задачах, где важны энергетические соотношения и наличие связей между телами.

После построения системы нелинейных дифференциальных уравнений нам нужно линеаризовать её, затем представить в пространстве состояний, и наконец перевести

дискретную форму. Это вызвано тем, что анализ и синтез управления проще производить, когда динамика системы описана именно в такой форме.

Приступим к построению динамической модели маятника в соответствие с предложенным выше алгоритмом. Сначала в качестве обобщенных координат возьмем θ и ϕ . С их помощью мы можем полностью описать положение системы.

На систему действуют три силы: сила вязкого трения $\mu_p \dot{\theta}$ в неподвижной точке крепления стержня O , сила вязкого трения $\mu_m \dot{\phi}$ между статором и ротором двигателя и крутящий момент τ , создаваемый электродвигателем. Таким образом вектор обобщенных сил будет выглядеть как:

$$\mathbf{Q} = [-\mu_p \dot{\theta} \quad \tau - \mu_m \dot{\phi}]^T.$$

Теперь вычислим общую кинетическую энергию маятника путем сложения кинетических энергий каждого звена по формуле:

$$T = \sum_{i=1}^2 T_i,$$

где $T_i = \frac{1}{2} m_i \dot{x}_i^2 + \frac{1}{2} m_i \dot{y}_i^2 + \frac{1}{2} I_i \dot{\alpha}_i^2$ – формула кинетической энергии в общем виде, m_i – масса i -го звена, I_i – момент инерции i -го звена, x_i и y_i – положение центра масс i -го звена вдоль осей X и Y соответственно, α_i – угол поворота i -го звена. В нашем случае x_i, y_i, α_i :

$$x_1 = L_1 \cos \theta; \quad y_1 = L_1 \sin \theta; \quad \alpha_1 = \theta; \quad (1)$$

$$x_2 = L_2 \cos \theta; \quad y_2 = L_2 \sin \theta; \quad \alpha_2 = \phi. \quad (2)$$

Их производные по времени примут следующий вид:

$$\dot{x}_1 = -L_1 \dot{\theta} \sin \theta; \quad \dot{y}_1 = L_1 \dot{\theta} \cos \theta; \quad \dot{\alpha}_1 = \dot{\theta}; \quad (3)$$

$$\dot{x}_2 = -L_2 \dot{\theta} \sin \theta; \quad \dot{y}_2 = L_2 \dot{\theta} \cos \theta; \quad \dot{\alpha}_2 = \dot{\theta} + \dot{\phi}. \quad (4)$$

Выражение $\dot{\alpha}_2 = \dot{\theta} + \dot{\phi}$ получено согласно *теореме о сложении скоростей*, которая звучит так:

Теорема о сложении скоростей

Абсолютная скорость точки равна векторной сумме её переносной и относительной скоростей:

$$\mathbf{v}_a = \mathbf{v}_e + \mathbf{v}_r,$$

где \mathbf{v}_a – вектор абсолютной скорости, \mathbf{v}_e – вектор переносной скорости, \mathbf{v}_r – вектор относительной скорости.

В нашем случае маховик вращается относительно стержня, следовательно угловая скорость равна $\dot{\theta} + \dot{\phi}$. Более подробно о сложном движении точки в пространстве можно почитать в [3].

Построим уравнение кинетической энергии маятника, подставив в него полученные переменные (3-4):

$$T = \frac{1}{2} \left(m_1 L_1^2 \dot{\theta}^2 \sin^2 \theta + m_1 L_1^2 \dot{\theta}^2 \cos^2 \theta + I_1 \dot{\theta}^2 + m_2 L_2^2 \dot{\theta}^2 \sin^2 \theta + m_2 L_2^2 \dot{\theta}^2 \cos^2 \theta + I_2 (\dot{\theta} + \dot{\phi})^2 \right).$$

Упростив выражение, получим:

$$T = \frac{1}{2}(m_1 L_1^2 + m_2 L_2^2 + I_1 + I_2)\dot{\theta}^2 + I_2 \dot{\theta} \dot{\phi} + \frac{1}{2} I_2 \dot{\phi}^2. \quad (5)$$

Аналогично, потенциальная энергия маятника равна сумме потенциальных энергий стержня и маховика:

$$U = m_1 g y_1 + m_2 g y_2.$$

Подставим сюда выражения (1-2):

$$U = (m_1 L_1 + m_2 L_2) g \cos \theta. \quad (6)$$

Для простоты дальнейших вычислений произведем замену переменных. Пусть $a = m_1 L_1^2 + m_2 L_2^2 + I_1$ и $b = (m_1 L_1 + m_2 L_2) g$. Тогда уравнения (5) и (6) примут вид:

$$T = \frac{1}{2}(a + I_2)\dot{\theta}^2 + I_2 \dot{\theta} \dot{\phi} + \frac{1}{2} I_2 \dot{\phi}^2$$

$$U = b \cos \theta.$$

Теперь построим Лагранжиан:

$$L = T - U = \frac{1}{2}(a + I_2)\dot{\theta}^2 + I_2 \dot{\theta} \dot{\phi} + \frac{1}{2} I_2 \dot{\phi}^2 - b \cos \theta. \quad (7)$$

Запишем уравнения Лагранжа второго рода с учетом обобщенных координат θ , ϕ и вектора обобщенных сил \mathbf{Q} :

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = -\mu_p \dot{\theta} \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = \tau - \mu_m \dot{\phi} \end{cases} \quad (8)$$

Найдем все необходимые производные лагранжиана (7):

$$\frac{\partial L}{\partial \theta} = b \sin \theta; \quad \frac{\partial L}{\partial \dot{\theta}} = (a + I_2)\dot{\theta} + I_2 \dot{\phi}; \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = (a + I_2)\ddot{\theta} + I_2 \ddot{\phi};$$

$$\frac{\partial L}{\partial \phi} = 0; \quad \frac{\partial L}{\partial \dot{\phi}} = I_2 \dot{\theta} + I_2 \dot{\phi}; \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) = I_2 \ddot{\theta} + I_2 \ddot{\phi}.$$

Подставим эти выражения в систему (8):

$$\begin{cases} (a + I_2)\ddot{\theta} + I_2 \ddot{\phi} - b \sin \theta = -\mu_p \dot{\theta} \\ I_2 \ddot{\theta} + I_2 \ddot{\phi} = \tau - \mu_m \dot{\phi}. \end{cases}$$

Преобразуем уравнения так, чтобы $\ddot{\theta}$ и $\ddot{\phi}$ лежали в левой части:

$$\begin{cases} \ddot{\theta} = -\frac{\mu_p}{a} \dot{\theta} + \frac{\mu_m}{a} \dot{\phi} + \frac{b}{a} \sin \theta - \frac{1}{a} \tau \\ \ddot{\phi} = -\frac{\mu_m}{I_2} \dot{\phi} + \frac{\mu_p}{a} \dot{\theta} - \frac{\mu_m}{a} \dot{\phi} - \frac{b}{a} \sin \theta + \left(\frac{1}{I_2} + \frac{1}{a} \right) \tau. \end{cases} \quad (9)$$

Приведенная система дифференциальных уравнений описывает динамику механической части обратного маятника. Как было сказано в начале раздела, сконвертируем прилагаемый двигателем момент в напряжение. Таким образом система управления будет рассматривать текущий двигатель как коллекторный и использовать напряжение в качестве управляющего воздействия.

Соотношение между напряжением и моментом выходного вала двигателя без редуктора и без учета трения описывается с помощью уравнений:

$$V = L_m \frac{di}{dt} + R_m i + K_e \dot{\phi},$$

$$\tau = K_t i.$$

Все параметры и переменные приведены в таблице 2. Подробнее, о том, как получить эти уравнения можно почитать в учебнике [4].

Таблица 2. Параметры электродвигателя.

Параметр или переменная	Ед. измерения	Описание
V	В	Подаваемое на двигатель напряжение
i	А	Ток, проходящий через обмотки двигателя
L_m	Гн	Индуктивность обмотки двигателя
R_m	Ом	Сопротивление обмотки двигателя
K_e	рад/с/В	Скоростная постоянная
K_t	Н м/А	Моментная постоянная

Параметры L_m, R_m, K_e, K_t обычно указываются в документации на двигатель, либо находятся экспериментально. Индуктивность обмоток, как правило, намного ниже сопротивления, соответственно мы можем пренебречь переменной L_m . Тогда соотношение между моментом на валу и током можно выразить как:

$$\tau = \frac{K_t(V - K_e \dot{\phi})}{R_m}.$$

Подставим это выражение в систему (9):

$$\begin{cases} \ddot{\theta} = \frac{b}{a} \sin \theta - \frac{\mu_p}{a} \dot{\theta} + \frac{R_m \mu_m + K_t K_e}{a R_m} \dot{\phi} - \frac{K_t}{a R_m} V \\ \ddot{\phi} = -\frac{b}{a} \sin \theta + \frac{\mu_p}{a} \dot{\theta} - \frac{a + I_2}{a I_2} \left(\mu_m + \frac{K_t K_e}{R_m} \right) \dot{\phi} - \frac{K_t(a + I_2)}{a I_2 R_m} V. \end{cases} \quad (10)$$

Мы получили математическую модель, полностью описывающую поведение обратного маятника. Теперь для синтеза управления и анализа на устойчивость нам необходимо её линеаризовать и привести к форме в пространствах состояния.

Линеаризуем систему (10) в окрестности точки $\theta = 0$. Тогда $\sin \theta \approx \theta$, и система примет вид:

$$\begin{cases} \ddot{\theta} = \frac{b}{a} \theta - \frac{\mu_p}{a} \dot{\theta} + \frac{R_m \mu_m + K_t K_e}{a R_m} \dot{\phi} - \frac{K_t}{a R_m} V \\ \ddot{\phi} = -\frac{b}{a} \theta + \frac{\mu_p}{a} \dot{\theta} - \frac{a + I_2}{a I_2} \left(\mu_m + \frac{K_t K_e}{R_m} \right) \dot{\phi} - \frac{K_t(a + I_2)}{a I_2 R_m} V. \end{cases}$$

Представим теперь эту систему в пространстве состояний матричным уравнением:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}. \quad (11)$$

В качестве вектора состояния выберем $\mathbf{x} = [\theta \quad \dot{\theta} \quad \phi]^T$, а в качестве управляющего воздействия возьмем $\mathbf{u} = [V]$. Само уравнение примет вид:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{b}{a} & -\frac{\mu_p}{a} & \frac{R_m \mu_m + K_t K_e}{a R_m} \\ -\frac{b}{a} & \frac{\mu_p}{a} & -\frac{a + I_2}{a I_2} \left(\mu_m + \frac{K_t K_e}{R_m} \right) \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_t}{a R_m} \\ -\frac{K_t(a + I_2)}{a I_2 R_m} \end{bmatrix} V \quad (12)$$

Теперь переведем уравнение (12) в дискретную форму. Существует множество методов преобразования непрерывных систем в дискретные. Мы рассмотрим метод Эйлера. От других он отличается простотой, но при этом обладает меньшей точностью при высоких значениях шага дискретизации. Чтобы метод был достаточно точным, шаг должен быть порядка 1 мс.

Представим $\dot{\mathbf{x}}$ в виде:

$$\dot{\mathbf{x}} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t},$$

где Δt - шаг дискретизации, \mathbf{x}_k – состояние системы на шаге k .

Подставим это выражение в уравнение (11):

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k.$$

Умножим уравнение на Δt и перенесем \mathbf{x}_k в правую часть:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{A}\mathbf{x}_k\Delta t + \mathbf{B}\mathbf{u}_k\Delta t.$$

Это уравнение можно представить в виде:

$$\mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k \quad (13)$$

где матрицы \mathbf{A}_d и \mathbf{B}_d описывают динамику системы и определяются как:

$$\mathbf{A}_d = \mathbf{I}_n + \mathbf{A}\Delta t, \quad (14)$$

$$\mathbf{B}_d = \mathbf{B}\Delta t, \quad (15)$$

где \mathbf{I}_n – единичная матрица размером $n \times n$. В итоге уравнение (12) будет выглядеть следующим образом:

$$\begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \\ \dot{\phi}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ \frac{b}{a}\Delta t & 1 - \frac{\mu_p}{a}\Delta t & \frac{R_m\mu_m + K_tK_e}{aR_m}\Delta t \\ -\frac{b}{a}\Delta t & \frac{\mu_p}{a}\Delta t & 1 - \frac{a + I_2}{aI_2}\left(\mu_m + \frac{K_tK_e}{R_m}\right)\Delta t \end{bmatrix} \begin{bmatrix} \theta_k \\ \dot{\theta}_k \\ \dot{\phi}_k \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_t}{aR_m}\Delta t \\ -\frac{K_t(a + I_2)}{aI_2R_m}\Delta t \end{bmatrix} V_k. \quad (16)$$

2.2. Стабилизация маятника в верхнем неустойчивом положении равновесия

Рассмотрим задачу стабилизации маятника в верхнем неустойчивом положении равновесия $\theta = 0$, $\dot{\theta} = 0$ в предположении, что в начале процесса стабилизации он уже находится в некоторой окрестности этого желаемого положения. Эта задача может быть решена с помощью регулятора полного состояния. Структурная схема такой системы управления показана на рисунке 10, где \mathbf{K} – матрица весовых коэффициентов. Отметим, что в системе в качестве обратной связи используется весь вектор состояния и управляющее воздействие выглядит как:

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k. \quad (17)$$

При этом задача сводится к выбору таких коэффициентов обратной связи матрицы \mathbf{K} , чтобы система обеспечивала устойчивое вертикальное положение маятника.

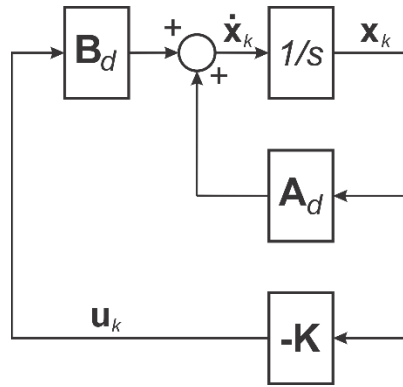


Рисунок 10. Структурная схема системы управления с регулятором полного состояния.

Существует множество способов выбора коэффициентов обратной связи, или, по-другому, *синтеза* управления. В текущей работе мы рассмотрим вариант решения задачи линейно-квадратичного управления. Далее по тексту, регулятор, синтезированный с помощью решения этой задачи, будем называть LQR от английского Linear Quadratic Regulator.

LQR – это один из видов оптимальных регуляторов, использующий квадратичный функционал качества, который необходимо минимизировать:

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) \Delta t,$$

где \mathbf{Q} и $\mathbf{R} \in \mathbb{R}^{n \times n}$ – заданные положительно определенные матрицы весовых коэффициентов. Чем больше элементы матрицы \mathbf{Q} , тем быстрее система будет стараться уменьшить ошибку управления, а чем больше элементы \mathbf{R} – тем плавнее будет поведение системы. И наоборот, чем элементы \mathbf{Q} меньше, тем больше статическая ошибка, и чем меньше элементы \mathbf{R} , тем агрессивнее будет управление. Таким образом, функционал качества выражает компромисс между отклонением от нуля и затратами на управление. Подбор весовых коэффициентов осуществляется по следующим принципам:

- начинать нужно с диагональных единичных матриц:

$$\mathbf{Q} = \text{diag}(q_1, q_2, \dots, q_n), \quad \mathbf{R} = \text{diag}(r_1, r_2, \dots, r_n).$$

- значения q_i увеличивают наказание за отклонение соответствующего состояния x_i от нуля.
- значения r_i увеличивают штраф за управление u_i , приводя к более плавному управлению.

В нашем случае используется дискретная система (16). Функционал качества для дискретной системы имеет вид:

$$J = \sum_{k=0}^{\infty} (\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k) \Delta t.$$

Закон управления по отрицательной обратной связи (17) должен минимизировать этот функционал. Согласно утверждению из [5] матрица \mathbf{K} вычисляется как:

$$\mathbf{K} = (\mathbf{R} + \mathbf{B}_d^T \mathbf{P} \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P} \mathbf{A}_d, \quad (18)$$

где \mathbf{P} – положительно определенная матрица, являющаяся решением дискретного алгебраического уравнения Риккати:

$$\mathbf{P} = \mathbf{A}_d^T \mathbf{P} \mathbf{A}_d - \mathbf{A}_d^T \mathbf{P} \mathbf{B}_d (\mathbf{R} + \mathbf{B}_d^T \mathbf{P} \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P} \mathbf{A}_d + \mathbf{Q}.$$

Это уравнение имеет единственное решение \mathbf{P} , если система управляема, а матрицы \mathbf{Q} и \mathbf{R} – положительно определенные.

Система называется *управляемой*, если существует такое допустимое управление \mathbf{u} , которое бы переводило начальное состояние \mathbf{x}_0 в конечное состояние \mathbf{x}_f за конечный интервал времени $[t_0, t_f]$.

Критерий управляемости линейных стационарных систем

Линейный стационарный объект управляем тогда и только тогда, когда матрица управляемости

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}]$$

имеет максимальный ранг, т.е. когда её ранг равен n .

Напомним, что ранг матрицы равен числу линейно независимых строк, числу линейно независимых столбцов или порядку отличного от нуля минора максимальной размерности.

После синтеза закона управления необходимо убедиться, что замкнутая система будет устойчива. Для этого воспользуемся критерием устойчивости для дискретных систем.

Критерий устойчивости дискретных систем

Для устойчивости дискретной системы необходимо и достаточно, чтобы все корни характеристического уравнения системы находились внутри круга единичного радиуса с центром в начале координат.

Разберем подробнее смысл этого критерия. Мы имеем замкнутую систему вида:

$$\mathbf{x}_{k+1} = \mathbf{A}_d \mathbf{x}_k + \mathbf{B}_d \mathbf{u}_k \quad (19)$$

$$\mathbf{u}_k = -\mathbf{K} \mathbf{x}_k. \quad (20)$$

Подставим \mathbf{u}_k из уравнения (20) в (19), вынесем \mathbf{x}_k за скобку и получим:

$$\mathbf{x}_{k+1} = (\mathbf{A}_d - \mathbf{B}_d \mathbf{K}) \mathbf{x}_k.$$

Характеристическое уравнение системы будет иметь вид:

$$(\mathbf{A}_d - \mathbf{B}_d \mathbf{K}) = \lambda.$$

Таким образом, согласно критерию устойчивости, все собственные значения λ_i матрицы $(\mathbf{A}_d - \mathbf{B}_d \mathbf{K})$ должны удовлетворять условию $|\lambda_i| < 1$. Если хотя бы одно собственное значение $|\lambda_i| > 1$, то система будет неустойчива. Какое-либо значение $|\lambda_j| = 1$ при всех остальных $|\lambda_i| < 1$ определяет границу устойчивости дискретной системы.

Для решения алгебраического уравнения Риккати применяют итерационные методы (например метод Ньютона) и на практике чаще всего его решают при помощи методов, имеющихся в Matlab или библиотеке Control Systems в языке Python. Используя эти же инструменты находят матрицу \mathbf{K} и производят анализ управляемости и устойчивости системы. После этого полученные значения матрицы \mathbf{K} применяются в управлении аппаратной платформой. Применительно к нашему обратному маятнику, \mathbf{K} будет представлять собой вектор, так как имеется всего одно управление. Описанный в этой главе алгоритм стабилизации будет выглядеть так:

1. Получить с датчиков значения θ и ϕ .

2. Вычислить угловые скорости $\dot{\theta}$ и $\dot{\phi}$ путем вычитания текущих показаний датчиков из предыдущих, которые затем делятся на время между измерениями. На примере угловой скорости стержня это выглядит так:

$$\dot{\theta}_k = \frac{\theta_k - \theta_{k-1}}{\Delta t}.$$

3. Вычислить желаемое значение напряжения на двигателе:

$$V = -\mathbf{K}\mathbf{x}_k = -(k_1\theta + k_2\dot{\theta} + k_3\dot{\phi}),$$

где k_i – i -й элемент вектора \mathbf{K} .

4. Значение V отправить в систему управления двигателем

2.3. Перевод маятника из нижнего положения равновесия в верхнее

Изначально маятник покоится в нижнем положении равновесия. Для перехода в верхнее положение равновесия его нужно сначала раскачать, а затем, поймав в верхнем положении, стабилизировать [6].

При раскачивании маятника ему нужно сообщить энергию, достаточную для перевода в верхнее положение. Полная энергия маятника E , без учета вращения относительно него маховика, описывается выражением:

$$E = \frac{1}{2}(m_1L_1^2 + m_2L_2^2 + I_1 + I_2)\dot{\theta}^2 + (m_1L_1 + m_2L_2)g \cos \theta.$$

Желаемая энергия маятника E^* , покоящегося в верхнем положении равновесия, равна $(m_1L_1 + m_2L_2)g$. Закон управления, который будет обеспечивать раскачивание маятника до тех пор, пока его энергия не достигнет величины E^* выглядит как:

$$u = k(E^* - E) \operatorname{sign}(\dot{\theta} \cos \theta), \quad (21)$$

где $k > 0$ – коэффициент обратной связи. Множитель $\operatorname{sign}(\dot{\theta} \cos \theta)$ обеспечивает подкачку энергии в сторону равновесия. Такой метод называется *управлением с формированием энергии*. В иностранной литературе его обычно называют Energy Shaping Control. Закон управления (21) обеспечивает рост полной энергии системы до заданного значения, знак угловой скорости стержня определяет направление вращения.

Слежение за заданным значением E^* энергии в соответствии с законом управления (21) прекращается, когда система попадает в некоторую область притяжения. После этого включается закон управления (17), который уже доводит маятник в желаемое верхнее положение равновесия и стабилизирует его в нём.

Дополнительно можно отметить, что если необходимо перевести маятник в нижнее положение равновесия с последующим торможением, то можно использовать этот же подход. В этом случае заданное значение энергии нужно взять с обратным знаком:

$$E^* = -(m_1L_1 + m_2L_2)g.$$

3. Практическая часть

При работе с реальными объектами управления, как правило, сначала производится описание объекта управления, построение регулятора, синтез и последующий анализ в среде математического моделирования. Затем полученный закон управления и коэффициенты обратной связи, удовлетворяющие желаемым требованиям качества системы, переносят на аппаратную платформу. Часто в роли объекта управления выступают устройства, которые при неправильно указанных коэффициентах, могут выйти из строя или причинить вред человеку. К таким устройствам можно отнести летательные

аппараты, автомобили или шагающие роботы. Именно поэтому сначала производится моделирование и уже потом – тестирование на реальном объекте. Ход нашей практической работы будет строиться по этому же принципу.

Напомним, что перед выполнением дальнейшей части работы, вам следует установить и настроить ПО по инструкции в файле *Tutorials/Ru_Инструкция_по_установке.docx*.

3.1. Математическое моделирование

- Откройте файл *Python/model.ipynb*. В нём в качестве примера реализована модель коллекторного двигателя. Его динамика описывается системой уравнений:

$$\begin{cases} \ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{K}{J}i \\ \frac{di}{dt} = -\frac{K}{L}\dot{\theta} - \frac{R}{L}i - \frac{1}{L}V, \end{cases} \quad (22)$$

где θ – угол поворота выходного вала двигателя, i – ток в обмотках статора, V – напряжение, b – коэффициент трения, J – момент инерции ротора, K – постоянная противо-ЭДС, L – индуктивность обмоток, R – сопротивление обмоток. В матричной форме эти уравнения выглядят как:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K}{J} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \frac{1}{L} V. \quad (23)$$

- В блоке *Pendulum Physical Parameters* приведены все необходимые физические параметры исследуемого объекта управления. Обычно эти параметры берутся из документации к составным устройствам или получаются экспериментальным путем. В данном случае в блоке приведены параметры коллекторного двигателя. Впишите сюда все необходимые параметры маятника, используя табл. 3. Названия переменным давайте такие же, как в графе «Обозначение».

Таблица 3. Физические параметры маятника.

Параметр	Обозначение	Ед. изм.	Значение	Описание
L_1	L1	м	0.06	Расстояние от точки O до центра масс стержня
L_2	L2	м	0.088	Расстояние от точки O до маховика
R	R	м	0.0655	Радиус маховика
m_1	m1	кг	0.1975	Масса стержня
m_2	m2	кг	0.2545	Масса маховика
I_1	I1	кг/м ²	$m1 * L1^{**2} / 12$	Момент инерции стержня
I_2	I2	кг/м ²	$m2 * R^{**2} / 2$	Момент инерции маховика
g	g	м/с ²	9.81	Ускорение свободного падения
μ_p	b_p	-	0.007	Коэффициент вязкого трения в шарнире O
μ_m	b_m	-	0.0	Коэффициент вязкого трения между статором и ротором двигателя
L_m	Lm	Гн	0.01	Индуктивность обмотки двигателя
R_m	Rm	Ом	22	Сопротивление обмотки двигателя
K_e	Ke	рад/с/В	0.29	Скоростная постоянная
K_t	Kt	Н м/А	0.27	Моментная постоянная
V_{max}	Vmax	В	12	Максимальное напряжение двигателя

- Блок *Nonlinear model* описывает систему уравнений (22). Преобразуйте этот блок так, чтобы он описывал нелинейную модель обратного маятника (10). Используйте следующий шаблон кода:

```

# constants
a = ...
b = ...

# function that describes nonlinear dynamics
def nonlinear_dynamics(theta, d_theta, d_phi, voltage):
    # Input:
    #   theta - pendulum angle [rad]
    #   d_theta - pendulum angular velocity [rad/s]
    #   d_phi - wheel angular velocity [rad/s]
    #   voltage - motor input voltage [volt]

    _dd_theta = ...
    _dd_phi = ...

    _theta = theta + d_theta * dt
    _d_theta = d_theta + _dd_theta * dt
    _d_phi = d_phi + _dd_phi * dt

    return np.array([_theta, _d_theta, _d_phi])

```

Чтобы удостовериться в корректности работы вашего кода, используйте блок *Nonlinear Model Simulation*, который производит симуляцию и вывод графика переходных процессов. При правильно составленной модели, вы должны увидеть такой график как на рис. 11.

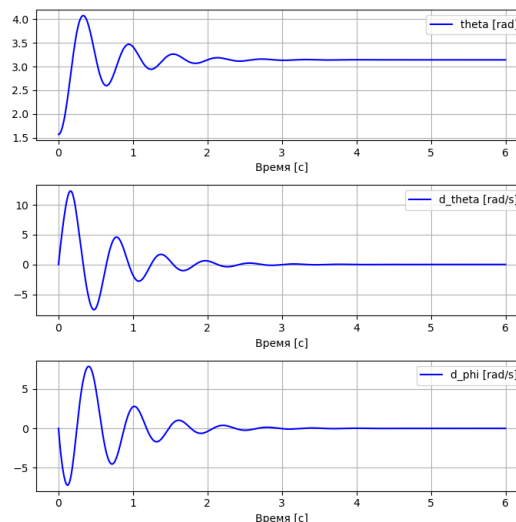


Рисунок 11. Результаты моделирования нелинейной модели.

Объясните, почему объект ведет себя именно так. Попробуйте изменить параметры x_0 и u , чтобы лучше понять его поведение.

- В блоке *State Space Model* описывается линеаризованная модель двигателя в пространстве состояний (23). Преобразуйте этот блок так, чтобы он описывал модель маятника (12). Используйте приведенный ниже шаблон:


```

# constants
a = ...
b = ...

# elements of matrix A and vector B
a21 = ...
a22 = ...
a23 = ...
a31 = ...
a32 = ...
a33 = ...

b2 = ...
b3 = ...

# State transition matrix A and control vector B
A = np.array([ [0, 1, 0],
               [a21, a22, a23],
               [a31, a32, a33]])

B = np.array([ [0],
               [b2],
               [b3]])

# Output matrices and vectors
C = np.array([ [1, 0, 0], # pend angle
               [0, 1, 0], # pend angular vel
               [0, 0, 1]]) # wheel speed

D = np.array([[0], [0], [0]])

```

Чтобы удостовериться в корректности работы вашего кода, используйте блок *State Space Model Simulation*, который производит симуляцию и вывод графика переходных процессов. При правильно составленной модели, вы получите график как на рис. 12.

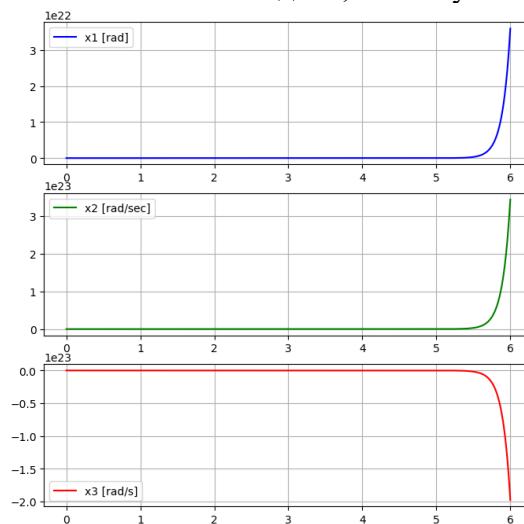


Рисунок 12. Результаты моделирования линейной модели.

Объясните, почему состояние линейной модели уходит в бесконечность в отличие от нелинейной.

- Блок *Convert to Discrete Time Domain* производит преобразование непрерывной модели в дискретную, используя средства библиотеки *Control Systems Library*. Пользуясь уравнениями (14) и (15) напишите собственную реализацию такого преобразования.
- В блоке *Controllability Analysis* произведите анализ управляемости системы. Переходите к следующим пунктам только, если система управляема. Если окажется,

что система не управляема, то скорее всего вы допустили ошибку во время выполнения предыдущих пунктов.

- Блок *Stabilization in upright position via LQR Control* производит решение линейно-квадратичной задачи управления. Получение матрицы коэффициентов обратной связи **K** производится с помощью функции `dlqr()` библиотеки Control Systems Library. В библиотеке `scipy` существует функция `solve_discrete_are(Ad, Bd, Q, R)`, решающая дискретное уравнение Риккати. На вход она принимает значения матриц **A_d**, **B_d**, **Q** и **R** в указанно порядке, а на выходе даёт решение **P**. Она уже импортирована в проект в блоке 1. Напишите собственную реализацию получения матрицы **K**, пользуясь предложенной функцией и уравнением (18).
- В блоке *Stability Analysis* проанализируйте устойчивость замкнутой системы, пользуясь критерием устойчивости дискретных систем со стр. 12. Чтобы получить собственные числа матрицы используйте функцию библиотеки `numpy`:

```
eigvals = np.linalg.eigvals(M).
```

На вход, вместо **M**, подайте матрицу **(A_d – B_dK)**, на выходе вы получите все собственные числа.

- В блоках *Closed-Loop System Simulation* производится симуляция замкнутой системы, используя линейные и нелинейные уравнения. Сравните полученные результаты. Поэкспериментируйте с весовыми коэффициентами матриц **Q** и **R**, посмотрите, как меняются переходные процессы. Выберите наилучшие по вашему мнению коэффициенты.
- В блоке *Swing-Up Control* производится управление раскачиванием маятника с последующим переходом в режим стабилизации. Коэффициент обратной связи **k** и порог переключения режимов управления ϵ заданы случайным образом. Вам необходимо подобрать такие значения этих переменных, чтобы маятник, раскачавшись, смог перейти в режим стабилизации. Значение **k** подбирайте, опираясь на формируемый в блоке график *Energy*. Старайтесь подобрать его таким, чтобы ошибка между **E** и **E*** была минимальной. Значение ϵ выбирайте таким, чтобы избежать преждевременного переключения в режим стабилизации, когда маятник ещё не успел достичь области притяжения. В нашем случае максимально возможная ошибка между **E** и **E*** составляет приблизительно 0.7 Дж. Соответственно, если вы поставите, например, $\epsilon = 0.4$, то контроллер перейдет в режим стабилизации, не успев раскачаться.

3.2. Натурные испытания

В процессе выполнения вышеописанных пунктов вы должны были получить коэффициенты матрицы **K** для режима стабилизации, коэффициент обратной связи **k** для управления раскачиванием и порог переключения режимов управления ϵ . Эти значения нам понадобятся для управления аппаратной платформой маятника. Чтобы применить их, сделайте следующие действия:

- откройте файл *Firmware/wheel_pendulum_practicum/wheel_pendulum_practicum.ino* с помощью Arduino IDE. В этом файле реализовано взаимодействие с датчиками, управление двигателем и реализованы режимы раскачивания, торможения и стабилизации. Объявление коэффициентов обратной связи **K** и **k** вместе с порогом переключения ϵ реализовано в строках 17-21. Функция `control()` вызывается каждые 10 мс. В ней реализованы контуры стабилизации, раскачивания и торможения. Изучите эту функцию и разберитесь, при каких условиях включается тот или иной режим управления.

- Вставьте полученные вами в ходе работы значения коэффициентов вместо указанных в строках 17-21.
- Подключите программатор к плате управления маятником, а затем к компьютеру через порт USB.
- Скомпилируйте и загрузите прошивку (кнопка *Upload* в Arduino IDE).

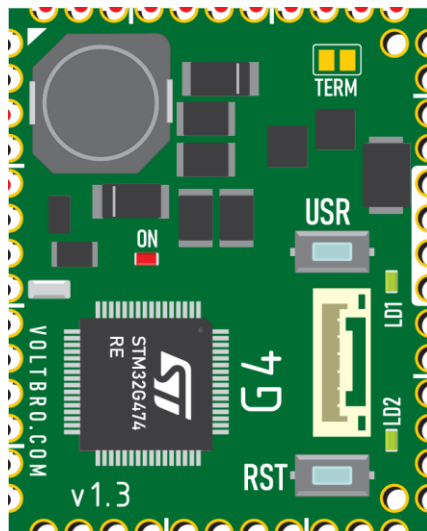



Рисунок 13. Микроконтроллерный модуль VB32G4

- После выполнения загрузки прошивки, нажмите кнопку RST на плате управления (рис. 13). Двигатель начнет вращаться. В этот момент система инициализирует все необходимые параметры управления двигателем. Дождитесь окончания вращения двигателя.
- Нажмите кнопку USB. Должно включиться управление маятником.

 Если через секунду после нажатия на кнопку USB двигатель не пришел в движение, сделайте следующее:

- качните маятник рукой
- отключите программатор, выключите и снова включите плату управления

Если вы включили управление маятником, он раскачался и встал в режим стабилизации, значит вы правильно настроили коэффициенты обратной связи. Часто, при разработке алгоритмов управления, получается так, что коэффициенты, отлично подходящие для модели, не всегда подходят к реальному объекту. Это связано с расхождением между моделью и реальностью. Чаще всего это происходит из-за погрешности вычислений физических параметров объекта управления и неучтенных параметров в уравнениях движения (например, силы трения, упругости, ограничения по скорости двигателя и т.д.). В таком случае, сначала стоит попробовать подобрать другие коэффициенты. Если это не поможет, необходимо произвести уточнение модели. Соответственно, в случае проблем со стабилизацией, попробуйте подобрать другие коэффициенты, меняя весовые коэффициенты матриц \mathbf{Q} и \mathbf{R} . Также, в случае проблем с раскачиванием или порогом переключения, попробуйте изменить коэффициенты k и ϵ . Если ничего не помогло, еще раз убедитесь, что вы правильно составили математическую модель в файле *model.ipynb*.

Дополнительно проверьте, как ведет себя маятник в режиме стабилизации, когда вы используете разные весовые коэффициенты матриц \mathbf{Q} и \mathbf{R} .

Контрольные вопросы

1. Почему обратный маятник с маховиком является системой с дефицитом управляющих воздействий?
2. Как вычисляется матрица коэффициентов обратной связи в линейно-квадратичной задаче управления?
3. В чём заключается физический смысл матриц **Q** и **R** в линейно-квадратичном функционале качества?
4. Какой критерий используется для проверки управляемости системы?
5. Какой критерий используется для проверки устойчивости дискретной системы?
6. Зачем линеаризуют нелинейную модель маятника?
7. Какую роль играет маховик в управлении системой?
8. Что произойдет, если понизить коэффициенты матрицы **R** в LQR?
9. Какие физические принципы лежат в основе метода управления с формированием энергии?

Список литературы

1. Формальский А.М. Управление движением неустойчивых объектов. – Физматлит, 2012
2. Буданов В. М., Данилов В.А., Капытов Д.В., Климов К.В. Малогабаритный четырехногий шагающий робот на базе бесколлекторных моторов. – ТиСУ РАН, №3, 2025.
3. Голубев Ю. Ф. Основы теоретической механики: Учебник. 3-е издание, переработанное и дополненное. – Издательство Московского университета, 2019.
4. Дементьев Ю. Н., Чернышев А.Ю., Чернышев И.А. Электрический привод. – Томский политехнический университет, 2010.
5. Ким Д.П. Теория автоматического управления. Том 2. Многомерные, нелинейные, оптимальные и адаптивные системы. – Физматлит, 2007.
6. Александров В. В., Болотин Ю. В. Спецпрактикум по теоретической и прикладной механике. – Издательство Московского университета, 2019.