

Финальный проект ряда лабораторных работ по курсу «Теория автоматического управления. Часть 1»

1. Мотивация

Целью, которую мы себе поставили в рамках разработки методических материалов к лабораторным работам, была передача Вам знаний, необходимых для выполнения на достойном уровне (к сожалению, по нашему опыту, люди предоставляют неработоспособные либо некорректные модели и выдают их за что-то серьезное) таких контрольных мероприятий, как курсовая работа по ТАУ (7й семестр СМ7) и выпускная квалификационная работа бакалавра, причем дать именно те знания, которые могут пригодиться в профессиональной деятельности и действительно используются, а не являются атавизмом, доставшимся от старых практик.

Согласно цели, в рамках нашего курса мы ознакомились с основами синтаксиса языка MATLAB – его процедурной части (MATLAB поддерживает объектно-ориентированную модель программирования, но мы не считали важным Вам об этом говорить, потому как ООП в нём реализовано недостаточно строго для новичков, да и это не совсем согласуется с целью, так как понадобилось бы предварительное введение в модель объектов и классов). После этого были даны основы по моделированию систем в Simulink, в том числе с указанием про необходимость иметь ввиду параметры численного моделирования, которые являются определяющими для получения устойчивого моделирования и корректных результатов. Отчасти Вы ознакомились с методами автоматизированного анализа и синтеза линейных и не только систем (в ЛР4 вы синтезировали нелинейную систему, возможно даже и не зная об этом). Наконец, Вам были предоставлены инструменты по декомпозиции сложных моделей в легко интерпретируемые, переносимые и тестируемые конструкции – компоненты: Subsystem, Referenced Subsystem, Referenced Model.

Последняя (почти) задача в нашем курсе будет являться нашим доказательством, что всё показанное действительно используется и применимо

в ваших университетских работах. К моменту защиты настоящего проекта, у Вас на руках будет половина курсовой работы по ТАУ (Вторая половина состоит в учете нелинейностей системы и введении дискретности в соответствующих частях модели).

Ввиду того что курсовая работа по ТАУ выполняется один семестр, задача, которую Вам предстоит решить, является достаточно комплексной, а потому для ускорения её выполнения, а также для предоставления опыта командной работы по разработке программного обеспечения на примере моделей Simulink, мы предлагаем выполнять её командами. Далее, перейдем к описанию организации работы.

2. Организация работы и формы отчетности

Как было упомянуто ранее, Вам необходимо разделиться на команды. Количество человек в команде фиксировано и составляет *4 человека* (команды, состоящие из менее, чем 4 человек, формируются при нехватке людей). Убедительная просьба делиться так, чтобы средний уровень знаний в команде был средним уровнем знаний в группе, иначе Вы можете не справиться с заданием. Как только команда сформирована, ваш тимлид (главный в команде) должен связаться с нами и подтвердить этот факт с предоставлением списка участников.

Для удобства формирования отчетов, в конце настоящего документа находится рабочий лист команды, который необходимо заполнить и предоставить в отчете каждого участника. Рабочий лист презентует разбиение подзадач по членам команды. Каждой подзадаче в нем соответствует своя сложность, указанная в виде числа процентов от объема всей задачи. Для успешной сдачи отчета каждому члену команды необходимо набрать не менее 20%. Каждый из участников формирует отчет только по тем подзадачам, которые ему были назначены. Отчеты могут быть приняты в любой момент до ЛР8.

Защита проекта производится всей командой на ЛР8. Вы предоставляете файлы проекта заранее онлайн, мы его запускаем (обратите внимание на требования к ПО репозитория в README) и удостоверяемся, что всё работает. Далее, мы проверяем, что каждый член команды действительно сделал то, что он сделал согласно стандартной процедуре (последнее может быть произведено в момент сдачи отчета). Если все проверки пройдены и у Вас более не останется долгов по контрольным мероприятиям, курс лабораторных работ считается успешно выполненным и Вам ставится условный зачет (обратите внимание, что зачет необходим для допуска к экзамену по курсу ТАУ). Если же долги у Вас всё-таки имеются, то мы ставим условный незачет и далее снимаем с себя ответственность за Вас и передаем её Вашим преподавателям по курсу ТАУ. Если же у Вас имеются контрольные мероприятия, сданные не в срок, то с ними вопрос решается одним из нескольких способов:

а) В рамках финального проекта вы делаете ту часть системы, или же пользуетесь инструментами, которые соответствуют сданному не в срок мероприятию;

б) Вы получаете индивидуальное задание по соответствующему мероприятию прямо на ЛР8 и делаете его;

После окончания ЛР8 все сданные не в срок мероприятия, с которыми вопрос не решен, становятся незащищенными ЛР и Вам ставится незачет (исправить который, однако, куда легче, чем несданные отчеты).

3. Постановка задачи

В рамках финального проекта Вам предлагается произвести синтез системы стабилизации обратного маятника. Для начала, опишем систему, которой будем управлять, и которую Вы будете моделировать.

Обратный маятник представляет собой небольшое устройство в виде прямоугольного параллелепипеда (корпуса), на котором размещена ось вращения прямоугольного стержня (рис. 1). Параллелепипед надет на жестко прикрепленную к земле рейку, и способен по ней поступательно перемещаться.

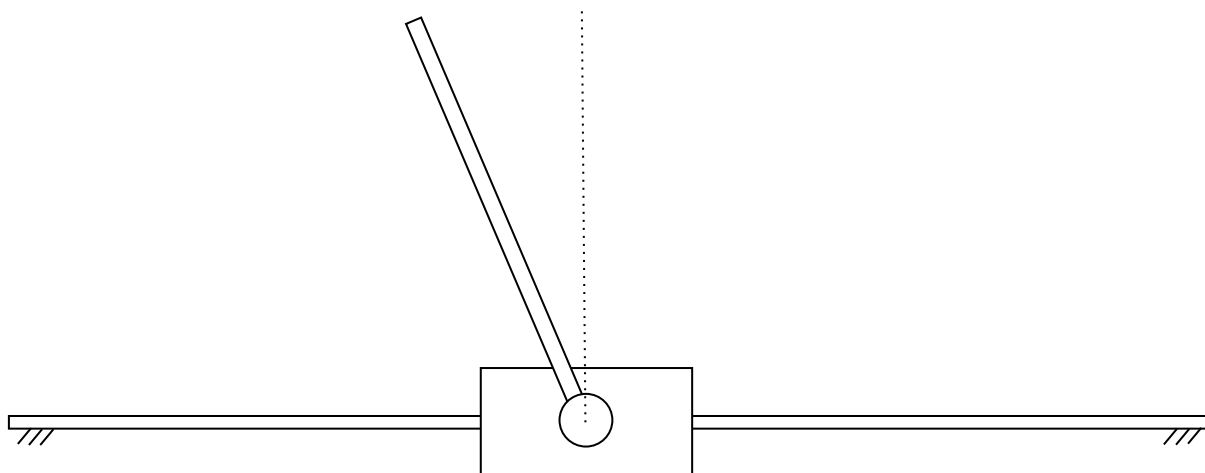


Рис. 1 – Общий вид объекта управления

Изнутри же прямоугольный параллелепипед устроен так, как показано на рис. 2.

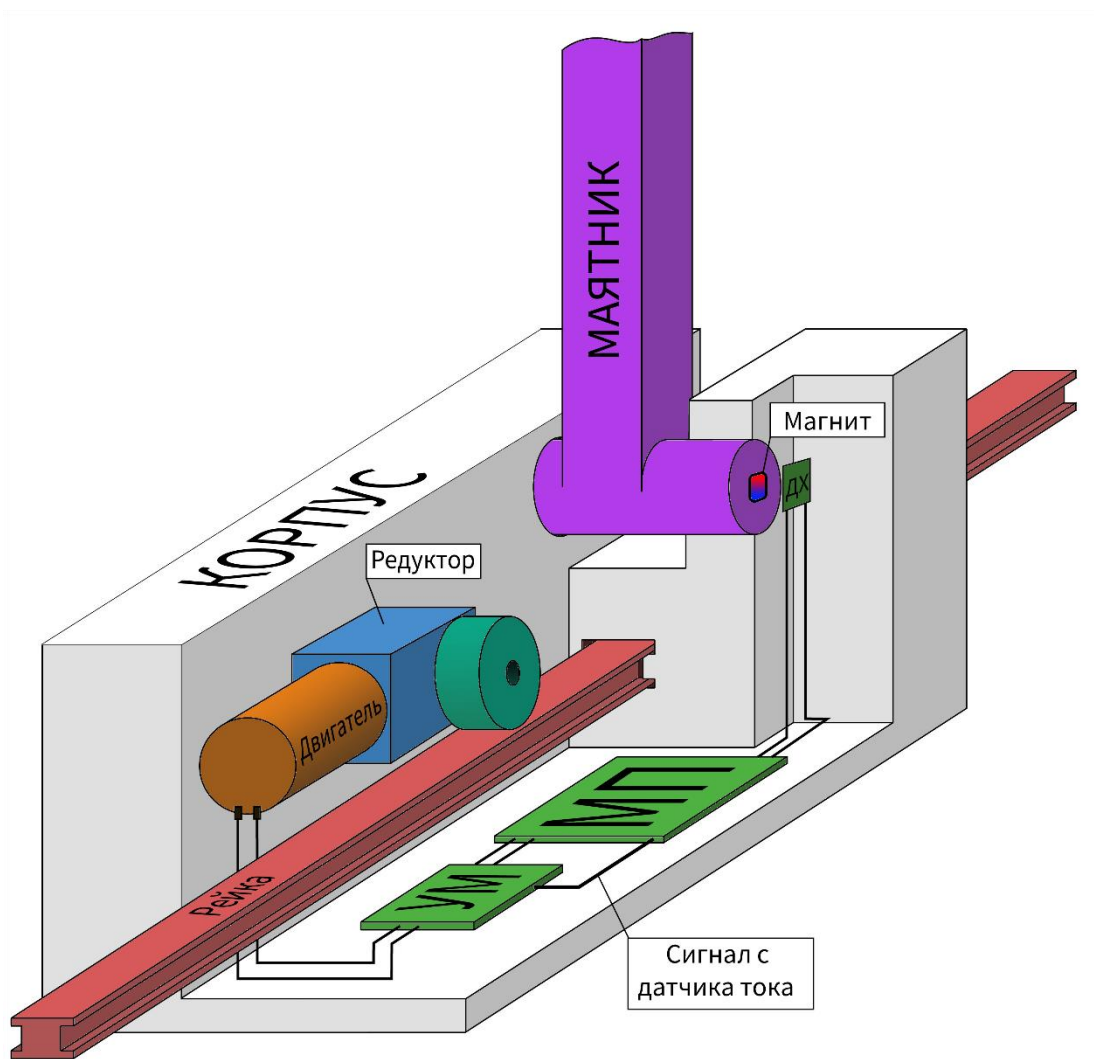


Рис. 2 - Функциональная схема обратного маятника

Перед Вами функциональная схема объекта управления. Главным её управляющим элементом является микроконтроллер (МП), на который с помощью датчика холла (ДХ), прикрепленного к корпусу и магнита, установленного на вращающемся стержне, поступает информация о положении последнего. Плата, на которой располагается ДХ достаточно умна, а потому на МП приходят уже расшифрованные данные об угловом положении и угловой скорости маятника. С другой стороны, МП соединен с усилителем мощности (УМ), который в свою очередь управляет системой, состоящей из двигателя, редуктора и зубчатого колеса, соединенного с рейкой. УМ управляется при помощи сигналов входящего напряжения на двигатель и сигнала выбора направления вращения, поэтому внутри МП должна быть реализована система преобразования знакового сигнала управления в тот, который понимает УМ. Также УМ снабжен специальным шунтовым резистором, включенным последовательно в цепь двигателя, сопротивление которого существенно меньше сопротивления обмоток двигателя и потому практически не влияет на механические характеристики последнего. УМ снимает падение напряжения на этом шунтовом резисторе, таким образом измеряя ток в цепи двигателя (который, как известно, пропорционален моменту на нём), и отправляет его в МП в качестве обратной связи.

При помощи указанного объекта управления при правильном синтезе регулятора возможно с помощью вышеописанного привода балансировать стержнем маятника в вертикальном положении. Это мы и будем реализовывать.

Для системы, приведенной выше, исходя из анализа функциональной схемы и выбора архитектуры коррекции, была составлена структурная схема системы автоматического управления стабилизации углового положения маятника. На рис. 3 приведен её иерархический вид в виде компонентов Simulink (тип компонентов определяется самостоятельно), а на рис. 4 – сама структурная схема.

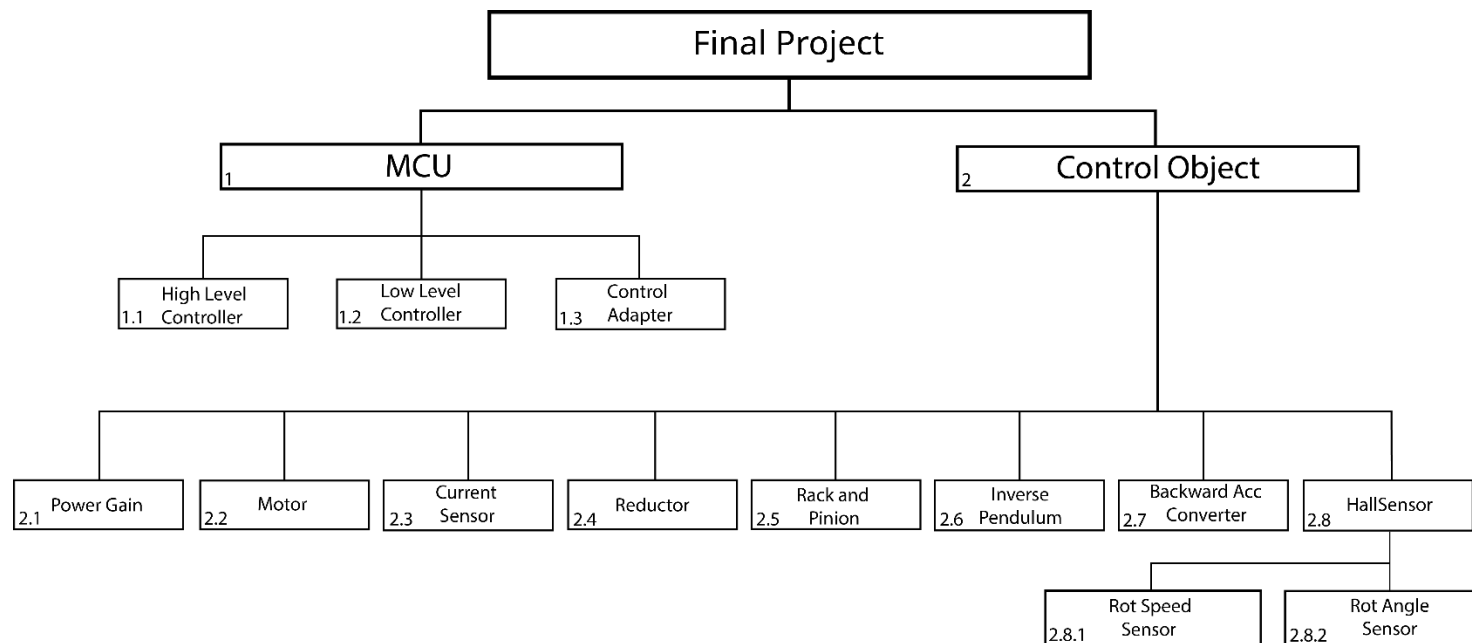


Рис. 3 - Иерархическая структура модели проекта

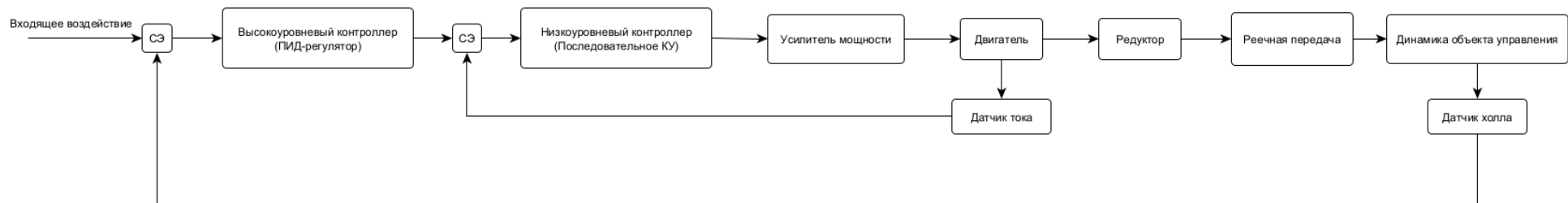


Рис. 4 - Структурная схема САУ

Начнем со структурной схемы САУ. Она состоит из двух контуров обратной связи. Внутренний (низкоуровневый) контур предназначен для стабилизации момента на двигателе, а внешний (высокоуровневый) – для управления угловым положением маятника. Предлагается реализовывать низкоуровневый контроллер в виде последовательного корректирующего устройства (возможно в виде коэффициента усиления), а высокоуровневый - в виде ПИД-регулятора. Отметим, что приведенной структуры хватит не только для стабилизации, но и для выдачи любой наперед заданной траектории движения маятника (другое дело, что ввиду отсутствия замыкания системы по положению на рейке это будет затруднительно)

Переходя к иерархической структуре, просто поясним, что каждый из компонентов означает:

1. MCU – модель микроконтроллера, внутри которой находятся все контроллеры;
 - 1.1. High Level Controller – модель высокоуровневого контроллера;
 - 1.2. Low Level Controller – модель низкоуровневого контроллера;
 - 1.3. Control Adapter – преобразователь выхода низкоуровневого контроллера в управляющие УМ сигналы.
2. Control Object – модель объекта управления;
 - 2.1. Power Gain – модель усилителя мощности;
 - 2.2. Motor – модель двигателя;
 - 2.3. Current Sensor – модель шунтового датчика тока;
 - 2.4. Reductor – модель редуктора;
 - 2.5. Rack and Pinion – модель реечной передачи;
 - 2.6. Inverse Pendulum – модель динамики обратного маятника;
 - 2.7. Backward Acc Converter – вспомогательная модель, необходимая для перевода скоростей и ускорения объекта управления в скорости и ускорения двигателя;
 - 2.8. HallSensor – модель датчика Холла;

2.8.1. Rot Speed Sensor – часть модели датчика холла, измеряющая угловую скорость;

2.8.2. Rot Angle Sensor – часть модели датчика холла, измеряющая угловое положение.

Как и говорилось ранее, Вашим заданием является разработка данной системы. Для этого надо решить подзадачи, на которые она разбита. В следующем разделе будет приведен список подзадач с показаниями к их решению.

4. Порядок решения подзадач, решаемых в рамках задачи

Создание модели усилителя мощности

В рамках данной подзадачи необходимо создать подсистему Power Gain, которая будет моделировать усилитель мощности. Требования к интерфейсу подсистемы следующие:

- Вход ControlVoltage – приходящее с микроконтроллера напряжение, управляющее амплитудой выходного сигнала;
- Вход RotDirection – вход логического (Boolean) типа, управляющий направлением вращения двигателя. Если он равен 1, то вращение должно осуществляться против часовой стрелки, если же 0, - то в обратную сторону;
- Выход PowerVoltage – напряжение, прикладываемое к двигателю.

Напомним, что усилитель мощности – это устройство, преобразующее слаботочный сигнал с небольшой амплитудой напряжения (напряжение с управляющей цепи) в выходной высокоточный сигнал, отвязанный от управляющей цепи. Вам же предстоит промоделировать только его усиление по напряжению. Известно, что диапазон входных напряжений усилителя равен от 0 до +3.3 В, а выходных – от 0 до 24 В.

В качестве отчета представить модель в Simulink с маской.

Создание модели двигателя

Эта задача нами уже ранее решалась, поэтому вопросов возникнуть не должно. Советуем обратить внимание на модель из ЛР4. Отметим, что в двигателе необходимо учесть вязкое трение, но не следует учитывать сухое. Итак, необходимо создать подсистему Motor. Приведем требования к интерфейсу:

- Вход InVoltage – входное напряжение, приходящее из усилителя мощности;
- Вход FbVelocity – обратная связь по скорости, приходящее из динамики обратного маятника через служебный блок преобразования скорости объекта в скорость двигателя;
- Вход FbAcceleration – аналогичен предыдущему, но являющийся обратной связью по ускорению;
- Выход OutTorque – момент на валу;
- Выход OutVelocity – скорость на валу;
- Выход OutAcceleration – ускорение на валу;
- Выход OutCurrent – ток в цепи двигателя.

Важно иметь ввиду, что выходные скорости и ускорения передаются через цепь обратной связи двигателя, но никак не порождаются моментом на нём.

Параметры двигателя следующие (Вы берете ДБУ20-3,7-9-24 ГИ):

Параметры БДПТ	ДБУ20-1,2-6-12 ПИ	ДБУ20-3,7-9-24 ПИ	ДБУ20-3,7-9-24 ГИ
Напряжение питания, В	12	24	24
Активное сопротивление якоря в нормальных условиях, Ом	8,5	14,9	≈10
Скорость холостого хода, об/мин	6760	10900	14400
Ток холостого хода, мА	80	62	60
Начальный пусковой момент, мН*м	≈23	≈26	≈37

Собственный момент сопротивления вращению, мН*м	$\approx(1,05...1,4) \cdot (n_{\text{ВРАЩ}}/10000 \text{ об/мин})^{0,4}$ при (2500...25000) об/мин; $\approx(0,2...0,4)$ при 0 об/мин; $\approx(0,9...1,2)$ при 6000 об/мин		
Скоростной коэффициент K_n , об/мин/В	≈ 565	≈ 455	≈ 600
Наклон механической характеристики $\Delta n/\Delta M$, об/мин/мН*м	≈ 330	≈ 410	≈ 400
Коэффициент момента, мН*м/А	≈ 17	≈ 21	≈ 16
Индуктивность якоря, мкГн	≈ 300	≈ 520	≈ 350
Электромеханическая постоянная времени, мс	≈ 17	≈ 20	≈ 20
Момент инерции ротора, г*мм ²	≈ 500		
Угловое ускорение при пуске, град/с ²	≈ 46	≈ 52	≈ 70
Габариты, мм*мм	Ø20* L46		
Рабочий диапазон температур, °С	Минус 50...+50 (обмотка до +125) До минус 60...+100 (по согласованию)		

В отчете представить модель Simulink с маской и расчет необходимых констант (если вы это делали).

Создание модели редуктора

Также является задачей, которую мы уже решали (см. ЛР4). В модели необходимо учесть вязкое трение, люфты и сухое трение учитывать не нужно. Таким образом, необходимо собрать подсистему Reductor. Требования к интерфейсу:

- Вход InTorque – момент на входном валу, приходящий из двигателя;
- Вход InVelocity – скорость на входном валу, приходящая из двигателя;
- Вход InAcceleration – ускорение на входном валу, приходящее из двигателя;

- Выход OutTorque – момент на выходном валу;
- Выход OutVelocity – скорость на выходном валу;
- Выход OutAcceleration – ускорение на выходном валу.

В качестве параметров принять передаточное отношение редуктора 100, а его момент инерции, приведенный к выходному валу 2.4 кг мм^2 .

В отчете представить модель Simulink с маской.

Создание модели реечной передачи

Данная задача подразумевает создание подсистемы RackAndPinion, очень похожей на подсистему редуктора и моделирующее реечную передачу. Приведем требования к интерфейсу:

- Вход InTorque – момент на шестерне, приходящий из редуктора;
- Вход InVelocity – скорость на шестерне, приходящая из редуктора;
- Вход InAcceleration – ускорение на шестерне, приходящее из редуктора;
- Выход OutForce – сила, действующая на ось шестерни рейкой;
- Выход OutSpeed – скорость центра шестерни;
- Выход OutAcceleration – ускорение центра шестерни.

В качестве численных параметров принять:

- Передаточное отношение передачи 20 рад/мм;
- Момент инерции шестерни 1 кг мм^2 .

В отчете представить модель Simulink с маской.

Создание модели датчика тока

В данной задаче необходимо создать подсистему CurrentSensor, моделирующей датчик тока. Требования к интерфейсу следующие:

- Вход RealCurrent – действительный ток на двигателе;
- Выход MeasuredCurrent – измеренный ток на двигателе.

Как было указано раньше, датчик тока в этой системе является резистором, последовательно соединенным с обмотками двигателя. Измеренное этим датчиком значение тока есть падение напряжения на этом резисторе. Ввиду того что резистор включен с двигателем последовательно, его номинальное сопротивление должно быть очень маленьким, чтобы не влиять на механические характеристики последнего. Поэтому примите его равным 0.001 Ом.

В отчете представить модель Simulink.

Создание служебной модели преобразования скоростей и ускорений

Вам необходимо создать подсистему BackwardAccConverter, которая должна преобразовывать скорость (ускорение) динамики объекта управления в скорость (ускорение) вала двигателя. Это нужно для обеспечения обратной кинематической связи между двигателем и остальными элементами кинематической цепи.

Требования к интерфейсу:

- Вход InMotion – скорость (ускорение) динамики объекта;
- Выход OutMotion – скорость (ускорение) динамики объекта, приведенная к валу двигателя.

В качестве численных параметров Вам понадобятся численные параметры двигателя, редуктора и реечной передачи, что уже были приведены выше.

В отчете представить модель Simulink.

Создание модели датчика Холла

В данной задаче требуется создать подсистему HallSensor, состоящую из двух подсистем меньшего размера – RotSpeedSensor и RotAngleSensor, моделирующих датчик скорости вращения и датчик углового положения обратного маятника соответственно. Требования к интерфейсу HallSensor:

- Вход RealSpeed – действительное значение угловой скорости маятника;

- Вход RealAngle – действительное значение углового положения маятника;
- Выход MeasuredSpeed – измеренное значение скорости;
- Выход MeasuredAngle – измеренное значение углового положения.

Несмотря на то, что настоящий датчик холла несколько сложнее, мы в нашей модели ограничимся тем допущением, что связь между выходным и входным значением датчика линейная и несмещенная. Таким образом, её можно создать с использованием коэффициентов усиления. Считайте, что скорость плата датчика усиливает в 0.2 раза, а угловое положение – в 2 раза.

В отчете представить модель главной подсистемы HallSensor.

Создание модели динамики обратного маятника

Данная задача является одной из самых сложных и включает в себя:

- Запись динамических уравнений обратного маятника, связывающих входное воздействие – силу, действующую на ось шестерни с выходными – перемещениями последнего по рейке и вращением стержня относительно параллелепипеда. При этом считайте, что угол поворота стержня отсчитывается от вертикали и положителен против часовой стрелки;
- Линеаризации этих уравнений относительно вертикального положения маятника (то есть считаем угол отклонения от вертикальной оси малым);
- Создании подсистемы InversePendulum, моделирующей вышеуказанные уравнения.

В получении уравнений динамики Вам поможет знание аналитической механики и, в частности, уравнений Лагранжа второго рода. В процессе записей уравнений не учитывайте трение. Для линеаризации необходимо будет вспомнить о том, как связаны тригонометрические функции со своими аргументами при условии малости последних. Для создания модели Вам,

возможно, будет удобно представить линеаризованные уравнения в модели переменных состояния. Приведем интерфейс блока:

- Вход HorForce – сила, действующая на ось шестерни со стороны рейки;
- Выход x – горизонтальная координата маятника на рейке;
- Выход Vx – горизонтальная скорость маятника на рейке;
- Выход ax – горизонтальное ускорение маятника на рейке;
- Выход Φ – угловое положение маятника;
- Выход w – угловая скорость маятника;
- Выход ϵ – угловое ускорение маятника.

В качестве численных параметров принять:

- Масса основания маятника (параллелепипеда) – 200 г;
- Масса стержня (он однородный) – 100 г;
- Длина стержня – 1 м.

Компоновка модели объекта управления

По результатам предыдущей задачи были собраны все модели нижней иерархии системы `ControlObject`. Поэтому в данной задаче их необходимо правильно соединить, создав подсистему (или подмодель) `ControlObject`. Для правильного соединения Вам необходимо прочитать все предыдущие задачи, обратить внимание на интерфейсы и посмотреть схемы. Не все выходы всех моделей Вам потребуются.

Требования к интерфейсу:

- Вход InCtrlVoltage – входное напряжение усилителя мощности, управляющее амплитудой;
- Вход InRotDirection – направление вращения двигателя (тип `Boolean`, выставить в настройках)
- Выход MeasAngle – измеренное угловое положение маятника;

- Выход MeasVelocity – измеренная угловая скорость объекта;
- Выход MeasCurrent – измеренный ток двигателя.

В отчете представить модель Simulink.

Создание модели адаптера управления

Как было ранее указано, усилитель мощности принимает на вход управление немного в другой форме, в отличие от того, что выдает контроллер нижнего уровня. Поэтому в микроконтроллере должен быть блок, который будет преобразовывать знаковое число – управление с контроллера в пару чисел: модуль управляющего напряжения и направление вращения (положительное против часовой стрелки). Приведем требования к интерфейсу:

- Вход SignedVolControl – входное управляющее напряжение;
- Выход VolControl – модуль управляющего напряжения;
- Выход DirectionControl – выбранное направление вращения.

Примечание. Для работоспособности системы необходимо, чтобы выход DirectionControl выдавал число типа Boolean (логическое). Это можно обеспечить, зайдя в настройки этого выхода и принудительно выставив в нем тип выходного значения.

В отчете представить модель Simulink.

Создание модели низкоуровневого контроллера

Согласно предложенной структурной схеме, низкоуровневый контроллер должен быть реализован в виде последовательного корректирующего устройства. Однако в разрабатываемой в рамках данной задачи подсистеме LowLevelController необходимо также вырабатывать сигнал ошибки привода.

Приведем требования к интерфейсу:

- Вход ReqTorque – желаемый момент на выходе двигателя;
- Вход FbCurrent – ток двигателя, измеренный датчиком;

- Выход SignedVolControl – выходной управляющий сигнал, подаваемый на двигатель.

Указание к выполнению. Необходимо привести ток, измеренный датчиком (а это падение напряжения на шунтовом резисторе) к моменту на двигателе с помощью известных формул, а затем выделить сигнал ошибки и присоединить к передаточной функции последовательного корректирующего устройства.

Важно! В этой задаче не нужно проводить синтез.

В отчете представить модель Simulink.

Создание модели высокоуровневого контроллера

В данной задаче необходимо собрать подсистему высокоуровневого контроллера HighLevelController согласно нашей структурной схеме. Он реализуется в виде ПИД-регулятора. Приведем требования к интерфейсу:

- Вход ReqAngle – задающее желаемое угловое положение маятника;
- Вход ReqVel – производная желаемого углового положения;
- Вход FbAngle – измеренное датчиком холла угловое положение;
- Вход FbVel – измеренная датчиком холла угловая скорость;
- Выход ReqTorque – желаемый момент на двигателе как выход управления.

Таким образом, вам необходимо выработать сигнал ошибки и подать его на ПИД-регулятор. Важно! Не используйте производные и передаточные функции, имитирующие производные в задаче, они у вас уже имеются на входе. Также в этой задаче не нужно проводить синтез.

В отчете представить модель Simulink.

Компоновка модели микроконтроллера

В этой задаче нужно собрать модель микроконтроллера из ранее полученных блоков, таким образом, реализовав подсистему (или подмодель) MCU. В этом Вам помогут ранее приведенные схемы иерархии и описания предыдущих блоков. Приведем требования к интерфейсу.

- Вход ReqAngle – задающее желаемое угловое положение маятника;
- Вход ReqVel – производная желаемого углового положения;
- Вход FbAngle – измеренное датчиком холла угловое положение;
- Вход FbVel – измеренная датчиком холла угловая скорость;
- Выход VolControl – модуль управляющего напряжения;
- Выход DirectionControl – выбранное направление вращения.

В отчете представить модель Simulink.

Компоновка модели САУ стабилизации маятника

В этой задаче необходимо собрать модель высшего уровня иерархии нашей схемы – FinalProject. Требования к интерфейсу не предъявляется, но она должна содержать модель микроконтроллера и модель объекта управления.

В отчете представить модель Simulink.

Настройка внутреннего контура коррекции

Для созданной командой модели необходимо произвести синтез внутреннего корректирующего устройства, которое по сигналу ошибки по моменту на двигателе выделяет желаемое управляющее напряжение на нём. Требования к синтезу необходимо определить самостоятельно исходя из назначения системы. Синтез можно проводить любым удобным способом.

В отчете представить ход синтеза от начала до конца и результирующее корректирующее устройство, а также модель, в которой проводился синтез в случае её наличия.

Настройка внешнего контура коррекции

Для созданной командой модели необходимо произвести синтез внешнего корректирующего устройства, которое по сигналу ошибки по положению маятника выделяет желаемый момент на двигателе. Требования к синтезу необходимо определить самостоятельно исходя из назначения системы. Синтез можно проводить любым удобным способом. Однако синтез этой системы проводить нельзя до синтеза внутреннего контура коррекции.

В отчете представить ход синтеза от начала до конца и результирующее корректирующее устройство.

Настройка параметров моделирования и обеспечение устойчивости симуляции

В данной задаче необходимо обеспечить стабильность и устойчивость симуляции, то есть добиться того, чтобы Ваша модель запускалась и успешно выполнялась. Этого можно добиться с помощью

- Настройки солверов;
- Разрыва алгебраических петель с помощью блоков задержки (Memory, Unit Delay);
- Добавления подавителей помех производных в обратных связях по скорости и ускорению.

Отметим, что подавитель помех реализуется в виде апериодического звена (фильтра низких частот) с достаточно большой, но не сильно (0.01 хватит) постоянной времени.

В отчете представить ход действий по обеспечению адекватности симулирования.

Снятие экспериментальных данных с модели и подтверждение работоспособности системы

Данная часть является завершающей. Необходимо с помощью одного из методов снятия сигналов с модели построить графики всех перемещений объекта

управления, интересующих нас ошибок, токов на двигателе и иных графиков, подтверждающих, что ваша система действительно выполняет свое назначение.

Данные действия требуется провести для нулевого входного сигнала, а также для ненулевого постоянного и синусового, чтобы проверить, что произойдет.

В отчете представить все графики и выводы о проделанной работе.

Исследование возможности задания произвольного горизонтального движения маятника с помощью управления наклоном

Понятно, что система стабилизации, которая была нами создана, в реальном устройстве бы не была единственной системой автоматического управления. В данной задаче предлагается исследовать возможность с помощью полученной системы управлять горизонтальным перемещением маятника при постоянном его наклоне.

Во-первых, нужно проверить, может ли система в неизменном состоянии с помощью хитрой подачи управляющих воздействий (реализации блока генерации управления для системы стабилизации) выполнять то, что требуется. Во-вторых, если не может, то нужно дополнить нашу систему дополнительными физическими устройствами, требуемыми для выполнения задачи, и добиться желаемого эффекта.

В отчете представить порядок действий по реализации системы горизонтального перемещения маятника и доказательства её работоспособности.

Исследование влияния помех в каналах объекта управления и неточностей в системе на её работоспособность

Естественно, что в реальных системах присутствуют случайные воздействия, действующие на систему. Таковыми могут являться ветровые нагрузки, электромагнитные шумы в каналах управления и другие. С другой

стороны, параметры реальных устройств могут отличаться от номинальных на некоторые стороны, то есть быть неточными.

В этой задаче Вам предлагается исследовать, как эти факторы влияют на производительность нашей системы стабилизации. Для этого следует выполнить ряд действий:

- Выделить в системе потенциальные случайные воздействия и встроить их в неё с возможностью отключения и включения случайных воздействий. При этом случайные воздействия должны быть действительно случайными, то есть генерироваться с помощью белого шума либо стандартных распределений. За подробностями реализации случайных величин в MATLAB обращайтесь к документации;
- В ряд наиболее критических блоков системы для тех констант, которые могут быть неточными, ввести случайное отклонение от номинальных значений, которое задается с помощью распределения случайных величин в скрипте инициализации констант, или (что лучше) при запуске модели
- Исследовать, что будет происходить с системой и будет ли она выполнять требования к ней при подаче таких факторов по отдельности и в совокупности. Сделать вывод.

В отчете представить всё, что Вы сделали в её рамках и графики поведения системы.

Исследование влияния наличия нелинейных элементов на работоспособность системы

Также в реальных системах присутствуют нелинейности, то есть связи между элементами системы, которые имеют нелинейный вид. К ним относят:

- Статические нелинейности (Нелинейные связи в алгебраических уравнениях)

- Динамические нелинейности (Нелинейные связи в дифференциальных уравнениях, либо изменяющиеся дифференциальные уравнения)

В рамках этой задачи необходимо каждый блок в модели проанализировать на наличие нелинейностей в реальной системе (ограничение напряжения на двигателе, ограничение тока усилителем мощности, мертвая зона в регулировочной характеристике двигателя, наличие насыщения в обмотках, люфты в редукторе и тд), затем добавить эти нелинейности в модель и проанализировать, что получится. Задавать нелинейности можно с помощью стандартных блоков Simulink, находящихся в разделе нелинейных блоков (Dead Zone, Saturation, Relay, ...)

В отчете необходимо представить всё, что вы сделали и графики поведения системы.

**Командный лист финального проекта ЛР по курсу «Теория
автоматического управления»**

Состав команды (Фамилия Имя Отчество, группа):

- 1.
- 2.
- 3.
- 4.

Таблица распределения по задачам

Подзадача	Стоимость, %	Кто выполняет
Создание модели усилителя мощности	5	
Создание модели двигателя	2	
Создание модели редуктора	2	
Создание модели реечной передачи	3	
Создание модели датчика тока	2	
Создание служебной модели преобразования скоростей	3	
Создание модели датчика Холла	5	
Создание модели динамики обратного маятника	25	
Компоновка модели объекта управления	10	
Создание модели адаптера управления	3	
Создание модели низкоуровневого контроллера	5	
Создание модели высокоуровневого контроллера	5	
Компоновка модели микроконтроллера	2	
Компоновка модели САУ	2	
Настройка внутреннего контура коррекции	7	
Настройка внешнего контура коррекции	8	

Настройка параметров моделирования и обеспечение устойчивости симуляции	8	
Снятие экспериментальных данных с модели и подтверждение работоспособности системы	5	
Исследование возможности задания произвольного горизонтального движения маятника с помощью управления наклоном	35	
Исследование влияния помех в каналах объекта управления и неточностей в системе на её работоспособность	25	
Исследование влияния наличия нелинейных элементов на работоспособность системы	25	

По результатам распределения, участникам соответствует*:

1. %
2. %
3. %
4. %

* Наличие у члена команды более 45 % гарантирует ему автомат при условии, что у остальных участников будет не менее 20% и все согласны с распределением.