1. **Исследование переходных процессов в Python, общие положения.**

Функционирование технических систем происходит в условиях внешней среды.

Любое изменение внешних управляющих или возмущающих воздействий приводит к возникновению переходного процесса.

**Переходный процесс** — в теории систем представляет реакцию динамической системы на приложенное к ней внешнее воздействие с момента приложения этого воздействия до некоторого установившегося значения во временной области.

В переходном процессе могут возникать большие амплитуды отклонений внутренних параметров, сопровождающиеся значительным повышением деформаций и напряжений в конструктивных элементах технических систем.

При изменении внешнего воздействия u(t) выходной сигнал технической системы v(t) может быть представлен состоящим из двух составляющих:

v(t)=vв(t)+vп(t),

**где**

**• vв(t)** – вынужденная установившаяся составляющая, определяемая частным решением неоднородного дифференциального уравнения, описывающего функционирование динамической технической системы.

• **vп(t)** – переходная составляющая, характеризующая свободный переходный процесс и определяемая общим решением однородного дифференциального уравнения без правой части.

При анализе переходных процессов применяются следующие виды воздействий:

* ступенчатое;
* импульсное;
* кусочно-линейное;
* гармоническое

**Переходной характеристикой** называется реакция технической системы на ступенчатое воздействие

Переходную характеристику рекомендуют получать при нулевых начальных условиях и единичном ступенчатом воздействии. Для оценки качества переходного процесса используются следующие показатели.

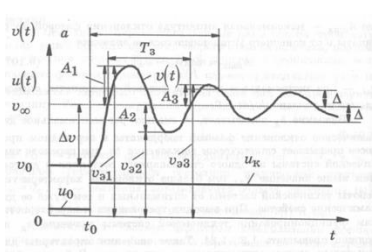
1) **Время переходного процесса** – tп, характеризуется длительностью пребывания технической системы в неустановившемся состоянии.

2) **Коэффициент динамичности -** kд, характеризуется максимальным отклонением выходного сигнала от его значения в установившемся конечном состоянии.

3) **Колебательность** – K, определяет число колебаний за время tп.

4) Декремент колебания

5) Перерегулирование



**Время переходного процесса**

Длительность переходного режима даже при быстром затухании динамического процесса теоретически бесконечна, поэтому на  
практике считают переходный процесс завершенным, если значение выходного сигнала перестает отличаться от установившегося конечного значения не более чем на определенную величину.

**Расчет переходного процесса в Python**

**Получение переходного процесса**

1. Зададим функцию ступенчатого воздействия

2. Решим ОДУ с учетом функции ступенчатого воздействия

3. Получим график переходного процесса

4. Определим числовое значение перемещения в установившемся режиме

**# Описание системы дифференциальных уравнений**

def dempf(y,t):

F=0

if t>0.5:

F=5

return [y[1], -2\*n\*y[1]-p\*\*2\*y[0]+F

1. **Алгоритм вычисления времени переходного процесса.**

**Переходный процесс** — в теории систем представляет реакцию динамической системы на приложенное к ней внешнее воздействие с момента приложения этого воздействия до некоторого установившегося значения во временной области.

Длительность переходного режима даже при быстром затухании динамического процесса теоретически бесконечна, поэтому на практике считают переходный процесс завершенным, если значение выходного сигнала перестает отличаться от установившегося конечного значения не более чем на определенную величину.

Коридор стабилизации Эта величина называется коридором стабилизации установившегося состояния Δ.

Обычно **Δ =5%\* |v(t)-v(∞)| |v(t)-v(∞)|** - установившееся состояние процесса

Время переходного процесса равно интервалу времени, измеряемому от момента начала ступени сигнала воздействия до момента последнего пересечения переходной характеристикой линии коридора стабилизации.

Время переходного процесса характеризует быстродействие технической системы.

1. **Алгоритм вычисления декремента, коэффициента динамичности и колебательности переходного процесса**

**Переходный процесс** – в теории систем представляет динамической системы, до некоторого устоявшегося состояния. Возникает под влиянием возмущающих воздействий, изменяющих её состояние, структуру или параметры, а также вследствие ненулевых начальных условий.

Коэффициент динамичности определяется по формуле: **Кд = 1+Amax/ ΔV .**

где

Amax – max амплитуда отклонения выходного сигнала V от конечного установившегося ее значения.

**Amax =max|UЭi-U∞|**

где Vэi – i-e экстремальное значение фазовой координаты определяемое из условия dv/dt=0

**Декремент колебаний** определяется по формуле D=A1/A2, где А1,А2 – амплитуды 2-х смежных отклонений фазовой оординаты от значения V∞. Чем выше значение D, тем быстрее затухают колебания.

**Колебательность К** определяется числом амплитудных значений VЭi за время tП или числом полупериодов колебаний. При высоких требованиях к переходным процессам K<2, при умеренных K<5.

1. **Обзор пакетов имитационного и схемотехнического моделирования**

**Имитационное моделирование** — это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, и с ней проводятся эксперименты с целью получения информации об этой системе. Экспериментирование с моделью называют **имитацией** (имитация — это постижение сути явления, не прибегая к экспериментам на реальном объекте).

Существует два подхода к имитационному моделированию - **статический и динамический**. **Статические** модели представляют собой системы уравнений, которые решаются один раз. **Динамические** модели включают в себя еще одну переменную - время.

В свою очередь, динамические модели бывают двух типов: **непрерывные** и **дискретные**. В **непрерывных** моделях время изменяется линейно, а процесс - в непосредственной зависимости от времени. Непрерывная модель позволяет моделировать загрязнения окружающей среды или поток жидкости в трубопроводе. В **дискретных** моделях переменными являются события и временные интервалы.

Пакеты имитационного моделирования:

* **Process Charter** 1.0.2 компании Scitor
* **Powersim 2.01** фирмы Modell Data AS
* **Ithink 3.0.61** производства High Performance Systems
* **Extend+BPR 3.1** компании Imagine That!
* **Pilgrim** производства МЭСИ, Россия;
* **Vensim** фирмы Ventana Systems

В пакете **Process Charter** модель строится с помощью блок-схемы. **Powersim** и **Ithink** используют систему обозначений **Systems Dynamics**. И, наконец, **Extend** применяет компоновочные блоки. Все продукты, кроме **Process Charter**, позволяют проводить анализ чувствительности, т. е. многократно исполнять модель с различными входными параметрами, чтобы сравнить результаты нескольких прогонов.

**1**. Пакет **Process Charter** в основном ориентирован на дискретное моделирование. Однако благодаря тому, что модели, созданные с помощью пакета **Process Charter**, изменяются во времени с постоянной скоростью, продукт можно настроить и на непрерывное моделирование.

**Недостатки** пакета **Process Charter:**

* небольшое количество возможностей;
* слабая поддержка моделирования непрерывных компонентов;
* ограниченный набор средств для анализа чувствительности и построения диаграмм.

**2**. Пакет **Powersim** является прекрасным средством создания непрерывных моделей. Однако с точки зрения дискретного моделирования он недостаточно эффективен. **Powersim** подходит пользователям, которым требуется построение непрерывных моделей, и которые хотят изучить довольно сложную систему обозначений **Systems Dynamics**.

**Powersim** содержит многие стандартные средства **Windows- приложений**, такие как меню и инструментальные линейки, и поддерживает технологии **Dynamic Data Exchange (DDE)** и **Object Linking and Embedding (OLE)**.

**Недостатки** пакета **Powersim**:

* ограниченная поддержка дискретного моделирования.

**3**. Пакет **Ithink**. C точки зрения непрерывного моделирования он отстает от **Powersim**, однако лучше поддерживает дискретное моделирование. Кроме того, пакет **Ithink** снабжен превосходными обучающей программой и документацией, а также большим количеством блоков для составления модели. Пакет выпускается в двух версиях – **Basic** и **Authoring**.

Версия **Authoring** позволяет разработчику включать в модель линейки с движками и другие средства управления моделью, а также вводить диаграммы и прочие изображения прямо в модель, чтобы пользователи могли контролировать процесс моделирования и сразу видеть его результаты.

**Недостатки** пакета **Ithink**:

* поддерживает меньшее количество функций, чем **Powersim**

**4**. Пакет **Extend**, обладающий средством построения непрерывных и дискретных моделей, широким диапазоном заранее сформированных блоков, поддержкой сторонних поставщиков и возможностью расширения является самым мощным продуктом из рассматриваемых нами.

**Недостатки** пакета **Extend**:

* используется в полном объеме только на компьютерах типа Macintosh;
* высокая стоимость.

**5**. Пакет **Pilgrim** поддерживает дискретно-непрерывное моделирование

К достоинствам пакета **Pilgrim** следует отнести свойство коллективного управления процессом моделирования, невысокую стоимость, наличие интерфейсов с базами данных и создание пользовательских блоков с помощью языка программирования С++.

**Недостатки** пакета **Pilgrim**:

* сложная система обозначений;
* необходимо знать язык программирования С++.

**6**. Пакет **Vensim** предназначен для построения системно-динамических моделей, также как **Powersim** и **Ithink**. Он поддерживает непрерывное моделирование; разработан фирмой **Ventana Systems**.

Пакет **Vensim** обладает средствами оптимизации и статистики; возможностью создавать подключаемые к другим программам DLL-библиотеки; простой графического интерфейса, нацеленного на профессионалов; расширяемой библиотекой функций.

**Недостатки** пакета **Vensim**:

* нет возможности конвертирования данных;
* небольшое количество встроенных математических функций в версии PLE.

**MicroCAP** - это универсальный пакет программ **схемотехнического** анализа, предназначенный для решения широкого круга задач. Характерной особенностью этого пакета является наличие удобного и дружественного графического интерфейса, что делает его особенно привлекательным для непрофессиональной студенческой аудитории. Несмотря на достаточно скромные требования к программно-аппаратным средствам ПК, его возможности достаточно велики. С его помощью можно анализировать не только аналоговые, но и цифровые устройства. Возможно также и смешанное моделирования аналого-цифровых электронных устройств, реализуемое в полной мере опытным пользователем пакета, способным в нестандартной ситуации создавать собственные макромодели, облегчающие имитационное моделирование без потери существенной информации о поведении системы.

Пакеты программ схемотехнического моделирования и проектирования семейства **Micro-Cap** фирмы **Spectrum Software** относятся к наиболее популярным системам автоматизированного проектирования (САПР) электронных устройств.

1. **Назначение и концепции моделирования в пакете Xcos.**

В Xcos моделируемый объект (система, устройство, процесс) представляется графически своей функциональной параметрической блок-схемой, включающей блоки элементов системы и связи между ними. Функциональные блоки элементов моделируемой системы могут, в свою очередь, представлять вложенные подсистемы со своей организацией, образуя иерархические структуры. Ценность Xcos заключается и в обширной, открытой для изучения и модификации библиотеке компонентов (блоков).

Она включает источники сигналов с практически любыми временными зависимостями, масштабирующие, линейные и нелинейные преобразователи с разнообразными формами передаточных характеристик, квантующее устройство, интегрирующие и дифференцирующие блоки и т. д. Как программное средство Xcos - типичный представитель визуально-ориентированного языка программирования.

**Функции пакета:**

- Создание моделей из имеющихся в библиотеке пакета блоков и их редактирование

- Запуск моделей на обработку

- Графическая интерпретация результатов моделирования

- Сервисные функции по работе с файлами моделей

Xcos обеспечивает доступ ко всем возможностям Scilab и является одновременно самостоятельной его компонентой, при работе с которой не обязательно иметь навыки в использовании других инструментов пакета Scilab

**Моделирование в Xcos базируется на следующих принципах:**

- Схема технического устройства или процесса представляется в виде блочной диаграммы

- Каждый блок диаграммы должен содержаться в библиотеке блоков пакета

- Блоки соединяются между собой линиями потока

- Блоки могут иметь различное число входов и выходов, называемых портами

- Входной порт обозначается треугольником, один из углов которого направлен в блок , а выходной порт – треугольником, один из углов которого направлен из блока

Последовательность создания и обработки модели:

- Создание файла модели

- Настройка параметров блока

- Соединение блоков модели

- Запуск модели на выполнение (предварительно задав параметры расчёта)

1. **Описание основных блоков и создание подсистем (суперблоков) в Xcos.**

**Суперблок** – это фрагмент модели, оформленный в виде отдельного блока.

Использование суперблоков предназначено для уменьшения количества одновременно отображаемых блоков на экране, что облегчает восприятие модели, а также позволяет создавать и отлаживать фрагменты модели по отдельности, что повышает технологичность создания модели.

Создание: Выделение части модели и вызов меню для объединения в суперблок. Блок SUPER\_f

**Источники сигналов и воздействий (по одному выходу, нет входов)**

|  |  |  |
| --- | --- | --- |
| CONST\_m | Источник постоянного сигнала | Задает постоянный по уровню сигнал |
| GENSIN\_f | Источник синусоидального сигнала | Формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением |
| RAMP | Источник линейно изменяющегося воздействия | Формирует линейный сигнал вида y = Slope\* time + Initial value |
| STEP\_FUNCTION | Генератор ступенчатого сигнала | Формирует ступенчатый сигнал |
| CLOCK\_c | Управление модельным временем | Формирует сигнал управления модельным временем |

**Регистрирующие устройства (имеют входы, нет выходов)**

|  |  |  |
| --- | --- | --- |
| CSCOPE | Осциллограф | Строит графики исследуемых сигналов в функции времени, позволяет наблюдать за изменениями сигналов в процессе моделирования |
| CSCOPEХY | Графопостроитель | Строит график одного сигнала в функции другого |

**Системы с непрерывным временем**

|  |  |  |
| --- | --- | --- |
| DERIV | Блок вычисления производной | Выполняет численное дифференцирование входного сигнала |
| LNTEGRAL\_f | Интегрирующий блок | Выполняет интегрирование входного сигнала |
| CLR | Блок передаточной функции | Блок передаточной характеристики задает передаточную функцию в виде отношения полиномов |
| CLSS | Блок модели динамического объекта | Блок создает динамический объект, описываемый уравнениями в пространстве состояний |
| PID | ПИД-регулятор | Блок, реализующий ПИД закон регулирования |

**Математические операции**

|  |  |  |
| --- | --- | --- |
| ABS\_VALUE | Блок вычисления модуля | Выполняет вычисление абсолютного значения величины сигнала |
| BIGSOM\_f | Блок вычисления суммы | Выполняет вычисление суммы текущих значений сигналов |
| PRODUCT | Блок умножения | Выполняет вычисление произведения текущих значений сигналов |
| SIGNUM | Блок определения знака сигнала | Определяет знак входного сигнала |
| GAINBLK\_f | Усилители | Выполняет умножение входного сигнала на постоянный коэффициент |
| INVBLK\_f  LOGBLK\_f  POWERBLK\_f  SQRT | Блоки вычисления математических функций | Выполняет вычисление математической функции |
| TrigFun | Блок вычисления тригонометрических функций | Выполняет вычисление тригонометрической функции |
| MAX\_f  MIN\_f | Блоки определения минимального или максимального значения | Определяют максимальное или минимальное значение из всех сигналов, поступающих на его входы |

**Блоки маршрутизации сигналов**

|  |  |  |
| --- | --- | --- |
| INTRPLBLK\_f | Блок одномерной таблицы | Задает в табличной форме функцию одной переменной. |
| MUX | Блок шинного формирователя | Формирует шину из сигналов различных типов |

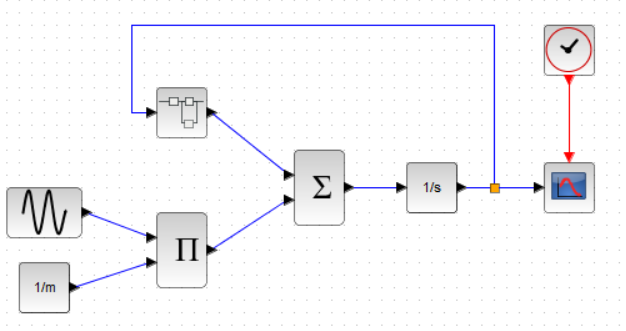
1. **Моделирование технических систем, описанных интегро-дифференциальными уравнениями в Xcos.**

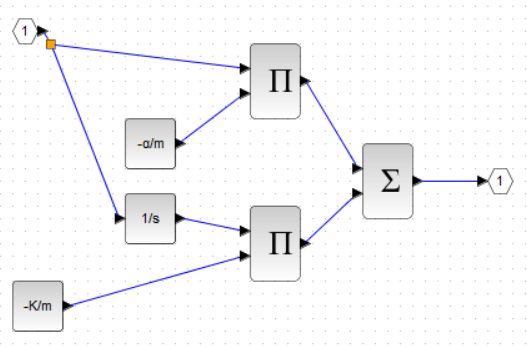
Пусть схема описывается следующим интегро-дифференциальным уравнением.

Для построения визуализированной схемы Simulink преобразуем его к нормализованному виду, чтобы производная была в левой части уравнения:

**Порядок составления схемы следующий:**

1. Правая часть итегро-дифференциального уравнения, описывающего схему, включает две составляющие, которые моделируются отдельно: одна – содержит источник нагружающей силы , другая моделирует остальные элементы механической системы .
2. Смоделируем первую составляющую в виде источника синусоидального сигнала с параметрами: амплитуда – 50, частота – 5. Умножим ее на 1/m
3. Смоделируем вторую составляющую в виде суперблока с одним входом и одним выходом. Для этого включим в модель суперблок раскроем его и смоделируем два слагаемых, причем для моделирования интеграла используется блок интегрирования.
4. Для того, чтобы найти значение v(t), нужно сложить две составляющие и проинтегрировать полученный сигнал. Следует заметить, что результат интегрирования v(t) является входным сигналом для подсистемы.
5. Задать в командном режиме для модели необходимые параметры(m, a, K):
6. Задать время моделирования.
7. Запустить модель на обработку, получить график функции скорости v(t).
8. Добавить в модель блок интегрирования для получения функции перемещения массы. Построить график функции перемещения.





1. **Моделирование технических систем, описанных дифференциальными уравнениями в Xcos**.

Для графического отображения сигнала как функции времени в **Xcos** используются блоки **CSCOPE** и **CMSCOPE** из палитры Регистрирующие устройства. **Блок** **CSCOPE** имеет один вход и отображает один или множество сигналов в единой системе координат. Скаляр (число) на входе осциллографа **CSCOPE** отображается как единственный сигнал вектор — **как множество сигналов.**

Блок **CMSCOPE** имеет два и более входов и отображает сигналы в отдельных системах координат в едином графическом окне.

Блок **CMSCOPE**, аналогично **CSCOPE**, отображает векторный вход в виде множества сигналов в одной системе координат. Однако, в отличие от CSCOPE, для него требуется явно указать размерности каждого из входов. Размерности входов задаются в параметрах блока переменной **Input port sizes** (размерности входных портов) — вектор, первый элемент которого относится к первому графику, второй — ко второму.

**Решение дифференциальных уравнений**

Реализация модели гидравлического демпфера в пакете Xcos системы Scilab

Для решения дифференциального уравнения его нужно привести к дифференциальному уравнению вида:

Смоделировать расчет переходного процесса в последовательном колебательном контуре, описанном системой дифференциальных уравнений вида:

**При решении данной задачи эти параметры имеют следующие значения**:  
R=1  
L=0.001  
C=0.0001

Используя специальные возможности этого раздела, пользователь может не только имитировать работу электрических схем во временной области, но и выполнять различные виды анализа таких схем.

1. **Сравнительный анализ формирования моделей в Xcos и в модулях «Электрические блоки» и «Mechanics»**

Для моделирования электроэнергетических систем в Scilab существует раздел «Электрические блоки», который содержит модели типовых устройств электроэнергетики.

* Принцип составления схем и использование этих блоков существенно отличается от составления схем в Xcos
* В Xcos соединяемые блоки представляют собой программы математического преобразования входных величин блока в выходные независимо от их физического содержания, а линии соединения переносят некоторый информационный сигнал.
* Соединение блоков из раздела «Электрические блоки» рассматривается как имитация электрических соединений, линии соединений рассматриваются как идеальная проводная связь, обеспечивающая передачу электрического сигнала от выхода одного блока к входу другого.
* Сами электрические блоки рассматриваются как модели электрических процессов, протекающих в устройстве, поведение которого моделируются.
* Соединительные линии между блоками являются, по сути, электрическими проводами, по которым ток может протекать также в двух направлениях.
* В Xcos-моделях же информационный сигнал распространяется только в одном направлении – от выхода одного блока к входу другого.
* После того, как электрическая модель построена, она может быть запущена на расчет так же как любая модель Xcos.
* Перед началом каждого расчета происходит инициализация модели.
* Xcos-блоки и электрические блоки не могут быть непосредственно соединены друг с другом.
* Сигнал от Xcos-блока можно передать к электрическому блоку через управляемые источники тока или напряжения, а наоборот – с помощью измерителей тока или напряжения.
* При инициализации вычисляется модель пространства состояний электрической схемы и строится эквивалентная модель, которая может рассчитываться в Xcos.

**Пакет Mechanics**

* Для моделирования механических систем в Scilab существует пакет Mechanics, который содержит модели специальных типовых элементов механизмов и машин, имитирующих их движение.
* Линии соединения входов и выходов блоков имитируют жесткое соединение выходной части одного механизма с входной частью другого.
* Это соединение моделирует передачу силового воздействия между частями разных механизмов.
* Согласно третьему закону Ньютона передачу силы нельзя рассматривать как однонаправленное воздействие, поэтому в линиях связь моделей Mechanics нет стрелок.
* В Xcos-моделях же информационный сигнал распространяется только в одном направлении – от выхода одгного блока к входу другого.
* Xcos – блоки и Mechanics-блоки не могут быть непосредственно соединены друг с другом.
* Сигнал от Xcos – блока модно передаь к М – блоку и обратно через специальные блоки, имеющие один Xcos – вход или выход и один М – вход или выход.

1. **Общие понятия теории автоматического управления (ТАУ**)

В общем случае **управление** – это динамический процесс перевода системы из текущего состояния в заданное.

Управление процессами вручную не всегда представляется возможным, поскольку в системах управления человек-оператор становится «слабым звеном» по быстродействию, силовым возможностям, информационной избыточности или недостаточности и вообще потому что он - человек.

**Автоматическое управление** - это управление, осуществляемое без участия человека. Под разработкой автоматических систем регулирования понимается, во-первых, выбор соответствующих регуляторов, датчиков и исполнительных устройств, во-вторых, расчет параметров настроек этого оборудования. **Теория автоматического управления** (ТАУ) представляет собой математический инструмент для решения задач синтеза таких систем.

Физические величины, определяющие ход технологического процесса, называются **параметрами технологического процесса**. Например, температура, давление, расход, напряжение, состав и т.д.

Параметр технологического процесса, который необходимо поддерживать постоянным или изменять по определенному закону, называется **регулируемой величиной** или **регулируемым параметром**.

Значение регулируемой величины в данный момент времени называется **мгновенным значением**.

Значение регулируемой величины, полученное в данный момент времени на основании измерений, называется ее **измеренным значением**.

**Объект управления** (объект регулирования) – система, требуемый режим выходных параметров которой должен изменяться или поддерживаться управляющими воздействиями.

**Управление** – формирование управляющих воздействий (чаще всего –энергетических потоков), обеспечивающих требуемый режим работы объекта управления (ОУ).

**Регулирование** – частный вид управления, когда задачей является обеспечение постоянства какой-либо выходной величины ОУ.

**Входное воздействие (Х)** – воздействие, подаваемое на вход системы или устройства.

**Выходное воздействие (Y)** – воздействие, выдаваемое на выходе системы или устройства, характеризует состояние объекта управления.

**Внешнее воздействие**– воздействие внешней среды на систему.

**Любая система управления (ручного, автоматического или автоматизированного) в обязательном порядке содержит четыре элемента (или четыре множества элементов), объединенных в замкнутый контур передачи воздействий:**

**- объект управления,**

**- управляющая часть,**

**- датчик (датчики),**

**- исполнительное устройство (устройства).**

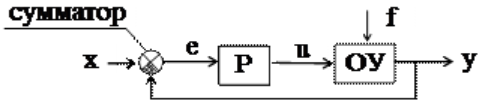


**Датчик** (Д) – устройство или комплекс устройств, преобразующих измеряемый параметр технологического процесса в вид, удобный для дальнейшей передачи и использования.

Датчик измеряет технологический параметр, преобразует его в другой вид энергии и передает управляющей части.

**Управляющая часть** реализует алгоритмы управления. В автоматических системах управления этой частью является регулятор, для систем ручного управления – человек-оператор. В управляющей части генерируются управляющие воздействия на объект управления (например, решения на включение/выключение, изменения напряжения и т.д.). Для реализации управляющих воздействий служат **исполнительные устройства** (ИУ).

Работа датчиков и исполнительных устройств в отличие от управляющей части заключается лишь в преобразовании энергии, изменения информации в них практически не происходит (если не считать погрешности). Поэтому при анализе и синтезе систем управления чаще эти части СУ опускают, считая их коэффициенты усиления равными «1». Наиболее часто в ТАУ при расчетах пользуются общей схемой одноконтурной АСР.



**x** - задающее воздействие (задание), **e**=**х**-**у** - ошибка регулирования, **u** - управляющее воздействие, **f** - возмущающее воздействие (возмущение).

Действие сумматора заключается в суммировании поступающих к нему сигналов. Если какой-либо сектор сумматора зачернен, то сигнал, поступающий в данный сектор, берется со знаком «минус». Поэтому в данной схеме ошибка **е** определяется как разность между **х** и **у**.

**Управляющее воздействие** (**u**) - воздействие управляющего устройства на объект управления.

**Управляющее устройство** (**Р**) - устройство, осуществляющее расчёт управляющего воздействия на объект управления с целью обеспечения требуемого режима работы.

**Возмущающее воздействие** (**f**) – ненаблюдаемое и неизмеряемое случайное воздействие, которое изменяет состояние объекта управления и приводит к изменению выходных параметров.

**Отклонение, рассогласование** (**е**=**х**-**у**) - разность между заданным (**х**) и действительным (**у**) значениями регулируемой величины.

**Автоматическая система регулирования** (АСР) (система автоматического регулирования САР) - автоматическая система с замкнутой цепью воздействия, в котором управление (u) вырабатывается в результате сравнения истинного значения у с заданным значением х.

Дополнительная связь в структурной схеме АСР, направленная от выхода к входу рассматриваемого участка цепи воздействий, называется обратной связью (ОС).

**Обратная связь** может быть отрицательной или положительной.

**Принцип функционирования** одноконтурной АСР: регулятор производит постоянное сравнение текущего значения регулируемой величины **у** с заданным значением **х**, определяя ошибку **е**=**х**–**у**. Если текущее значение равно заданному, то регулятор не изменяет управляющее воздействие (АСР работает в установившемся режиме), в противном случае управляющее воздействие на объект **u** изменяется в соответствии с величиной ошибки. Чем больше ошибка регулирования (и дольше она наблюдается), тем больше изменение управляющего воздействия.

1. **Принципы управления и основные виды САУ.**

САУ можно представить в виде функциональной схемы, элементы которой называются функциональными звеньями.

Эти звенья изображаются прямоугольниками, в которых записывается функция преобразования входной величины в выходную.

Эти величины могут иметь одинаковую или различную природу, например, входное и выходное электрическое напряжение, или электрическое напряжение на входе и скорость механического перемещения на выходе и т.п.

**Принципы управления**

**Принято различать три фундаментальных принципа управления:**

* Принцип разомкнутого управления.
* Принцип компенсации.
* Принцип обратной связи.

**Сущность принципа разомкнутого управления состоит в следующим:**

* Программа управления жестко задана ЗУ.
* Управление не учитывает влияние возмущений на параметры процесса.
* Примерами систем, работающих по принципу разомкнутого управления, являются часы, магнитофон, компьютер и т.п.

**УУ**

**ОУ**

задание

*r(t)*

*vd(t)*

*v(t)*

управление

возмущение

выход

*u(t)*

*y(t)*

**Недостатки:**

1. Необходима полная информация о модели ОУ.
2. Невозможно устранить влияние возмущений.

**Достоинства и недостатки принципа компенсации:**

**Достоинства:**

* Быстрота реакции на возмущения.
* Он более точен, чем принцип разомкнутого управления.

**Недостатки:**

* Невозможность учета подобного образом всех возможных возмущений.

Сущность принципа обратной связи (Получила наибольшее распространение в технике) – способ сохранения состояния равновесия системы.

Обратная связь является основным механизмом (Способом) поддержания состояния равновесия (гомеостазис) в: **технических системах, в живой природе, в социальных системах.**

Обратная связь (Управление по отклонению) – способ сохранения состояния равновесия системы.

**УУ**

**ОУ**

задание

*r(t)*

*vd(t)*

*v(t)*

управление

Возмущение, *M(t)*

выход

*u(t)*

*y(t)*

Обратная связь

**Достоинства:**

1. Управление при неполной информации о модели ОУ.
2. НЕ требуется измерение возмущающих воздействий.
3. **Определение передаточной функции, получение передаточной функции по дифференциальному уравнению.**

**Передаточная функция** – одна из способов математического описания динамической системы.

Используется в основном в теории управления, связи и цифровой обработки сигналов. Представляет собой дифференциальный оператор, выражающий связь между входом и выходом линейной стационарной системы.

**Преобразование дифференциальных уравнений по Лапласу** дает возможность ввести удобное понятие придаточной функции, характеризующей динамические свойства системы.

Передаточной функцией называется отношение изображения выходного воздействия Y(p) к изображению входного X(p) при нулевых начальных условиях.

Передаточная функция является дробно-рациональной функцией комплексной переменной:

где:

Передаточная функция имеет порядок, который определяется порядком полинома знаменателя (n).

Изображение выходного сигнала можно найти как:

При расчете настроек регуляторов широко используются достаточно простые динамические модели промышленных объектов управления.

В частности, использование моделей инерционных звеньев первого или второго порядка с запаздыванием для расчета настроек регуляторов обеспечивает в большинстве случаев качественную работу реальной системы управления. В зависимости от вида переходной характеристики (кривой разгона) задаются чаще всего одним из трех видов передаточной функции объекта управления:

1. В виде передаточной функции инерционного звена первого порядка:

Где: К-коэффициент усиления, Т- постоянная времени, τ- запаздывание, которые должны быть определены в окрестности номинального режима работы объекта.

2. Для объекта управления без самовыравнивания передаточная функция имеет вид:

3.Более точнее динамику объекта описывает модель второго порядка с запаздыванием:

 Дифференциальное уравнение первого порядка

1) Преобразуем уравнение по Лапласу – заменяем x(t) и y(t) на x(p) и y(p), а вместо знака производной запишем оператор p Tpy(p) + y(p) = x(p)

2) Вынесем Описание: img-DWOnBSза скобки и найдем отношение y(p)/x(p), которое является передаточной функцией

2. Астатическое звено второго порядка.

+ Обозначив , получим

1. **Типовые звенья САУ, их временные характеристики**

Звенья систем регулирования могут иметь разную физическую основу (электрические, пневматические, механические и др. звенья), но относится к одной группе. Соотношение входных и выходных сигналов в звеньях одной группы описываются одинаковыми [передаточными функциями](https://automation-system.ru/main/item/21-peredatochnaya-funkcziya.html).

Простейшие [типовые звенья](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html):

* усилительное,
* интегрирующее идеальное, интегрирующее реальное,
* дифференцирующее (идеальное, реальное),
* апериодическое инерционное,
* колебательное,
* запаздывающее.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип звена и его  передаточная функция | | Временные характеристики позиционных звеньев | | |
| Переходная функция *h*(*t*) | Функция веса *w*(*t*) | |
| 1. Безынерционное  *W*(*p*) =*K* | |  | Типовые звенья САУ и их характеристики | |
| 2. Апериодическое 1-го порядка | | Типовые звенья САУ и их характеристики | Типовые звенья САУ и их характеристики | |
| 3. Колебательное | | Типовые звенья САУ и их характеристики | Типовые звенья САУ и их характеристики | |
| 4. Идеальное  *W*(*p*) = *K*/*p*, *K* = 1/*T* | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики | |
| 5. С замедлением | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики |
| 6. Изодромное | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики |
| Временные характеристики дифференцирующих звеньев | | | |
| 7. С замедлением | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики |
| 8. Форсирующее  *W*(*p*) = *K*(1+*Tp*) | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики |
| Временные характеристики звена запаздывания на постоянное время τ | | | |
| 9. Запаздывающее  *W*(*p*) = *e*– τ*p* | Типовые звенья САУ и их характеристики | | Типовые звенья САУ и их характеристики |

1. **Передаточные функции типовых звеньев САУ**

Простейшие типовые звенья:

* усилительное,
* интегрирующее идеальное, интегрирующее реальное,
* дифференцирующее (идеальное, реальное),
* апериодическое инерционное,
* колебательное,
* запаздывающее.

 Переходные характеристики и передаточные функции типовых звеньев:

|  |  |
| --- | --- |
| **а) Усилительное звено** | **б) Идеальное интегрирующее** |
| **в) Реальное интегрирующие** | **г)Реальное дифференцирующее** |
| **д)Апериодическое инерционное** | **е) Колебательное звено** |

**Пояснения к рисунку:**

а) [Усилительное звено](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html), [пропорциональное звено](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html) усиливает входной сигнал в К раз. Уравнение звена у = К\*х, передаточная функция W(р) = К (где - К коэффициент усиления).

б) Идеальное интергрирующее звено имеет выходную величину пропорциональную интегралу входной величины. При подаче сигнала на вход звена выходной сигнал постоянно возрастает.

в) Реальное [интегрирующее звено](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html) имеет передаточную функцию. Реальное интегрирующее звено является звеном с запаздыванием. [Переходная характеристика](https://automation-system.ru/main/item/70-opredelenie-parametrov-perexodnyx-xarakteristik.html) в отличие от идеального звена является кривой.

г) Идеальные дифференцирующие звенья физически не реализуемы. Реальные  
дифференцирующие звенья представляют собой дифференцирующие звенья большинства обьектов.

д) Апериодическое звено 1-го порядка, где Т– постаянная времени. Большинство тепловых обьектов являются [апериодическими звеньями](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html).

е) [Колебательное звено](https://automation-system.ru/main/item/67-tipovye-zvenya-sistem-regulirovaniya.html). При подаче на вход ступенчатого воздействия амплитудой х0 переходная кривая будет иметь один из двух видов: апериодический или колебательный.

ж) Запаздывающее звено. Передаточная функция звена:  или

Выходная величина Y повторяет входную величину X с некоторым запаздыванием .

1. **Соединение структурных звеньев САУ**

При рассмотрении структурных схем в САУ предполагается, что звенья являются направленными, то есть преобразуют сигнал в направлении от входа к выходу и что выполняется условие независимости передаточных функций от наличия соседних звеньев. Различают три типа соединения звеньев:

* последовательное;
* параллельное;
* встречно-параллельное (соединение по принципу обратной связи).

**1. Последовательное соединение звеньев**

Последовательное соединение трех звеньев.

x=x1 y1=x2 y2=x3 y3=y

W2(p)

W3(p)

W1(p)

**2. Параллельное соединение звеньев**

На рисунке ниже представлено параллельное соединение трех звеньев.

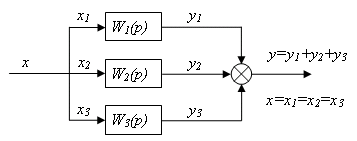


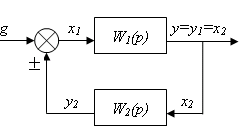
Рис. Параллельное соединение звеньев

Передаточная функция всей цепи определяется следующими соотношениями:

На основании полученного выражения для Описание: http://mc-plc.ru/tau/24c-273-25c-2ed/image020.png звеньев можно записать

**3. Встречно-параллельное соединение звеньев**

Далее представлено встречно-параллельное соединение трех звеньев.



Цепь, содержащая звено W1(p)(W2(p)), называется прямой (обратной) связью.

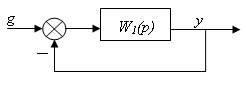
Зная передаточные функции W1 и W2, найти передаточную функцию всей цепи.

Откуда

**Классификация обратных связей**

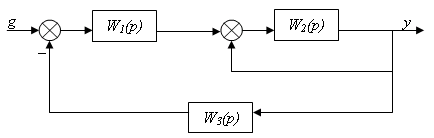
Предположим, что W2 (p)>0, тогда, если к сумматору обратная связь (ОС) подходит со знаком + (-), то ОС называется положительной (отрицательной). При положительной ОС в знаменателе должен стоять (-), при отрицательной – (+).

Если W2 (p)≡1, то ОС называется **единичной**. В этом случае структурная схема имеет вид



Если звено W2 (p) является позиционным, то ОС называется **жёсткой**. Если звено W2 (p) дифференцирующего типа, то ОС **называется гибкой или скоростной**.

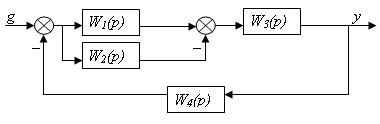
САУ могут содержать несколько обратных связей, например, так, как показано далее.



Единичная ОС является **местной ОС**, а связь, содержащая звено W3 (p), является **главной**. Главная ОС в подавляющем большинстве случаев является отрицательной.

Полученные соотношения между передаточными функциями для трех типов соединения звеньев позволяют получить передаточные функции и для сложных соединений звеньев.

Пусть дана структурная схема системы.



Передаточная функция всей цепи запишется в следующем виде:

1. **Устойчивость линейных динамических систем**

Под **устойчивостью** системы понимается способность ее возвращаться к состоянию установившегося равновесия после снятия возмущения, нарушившего это равновесие.

**Неустойчивая система** непрерывно удаляется от равновесного состояния или совершает вокруг него колебания с возрастающей амплитудой.

Устойчивость линейной системы определяется не характером возмущения, а структурой самой системы.

**Устойчивость линейных динамических систем(ЛДС)**

**Аксиома 1**: Устойчивость определяется внутренним состоянием ДС.

**Аксиома 2**: Устойчивость не является абсолютным свойством ДС.

**Устойчивость «вход-выход»**

Обычно для инженеров практиков в первую очередь важно, чтобы система не «пошла вразнос», то есть, чтобы управляемая величина не росла неограниченно при всех допустимых входных сигналах.

Если это так, говорят, что система обладает устойчивостью «вход-выход» (при ограниченном входе выход также ограничен).

Понятие «**техническая устойчивость**» относится к автономной системе, у которой все входные сигналы равны нулю.

Положением **равновесия** называют состояние системы, которая находится в покое, то есть, сигнал выхода y(t) – постоянная величина, и все его производные равны нулю.

Систему выводят из положения равновесия и убирают все возмущения. Если с течением времени (при t →∞) система возвращается в положение равновесия, она называется устойчивой.

Если выходная координата остается ограниченной (не уходит в бесконечность), система называется **нейтрально устойчивой**, а если выход становится бесконечным – **неустойчивой**.

**Метод оценки устойчивости ЛДС по корням характеристического уравнения**

Дифференциальное уравнение → характеристическое уравнение → корни

Корни в общем виде:

**Анализ корней**

1. Корни вещественны, отрицательны

**ЛДС – устойчивая без колебаний.**

2. Комплексные **(- a ± jw)** с отрицательной вещественной частью - **ЛДС устойчива с колебаниями**.

3. Один из корней равен **0 - ЛДС нейтральная**.

4. Корни мнимые (незатухающие колебания) **ЛДС на границе устойчивости**.

5. Корень вещественный положительный - **ЛДС неустойчива**.

6. Корни комплексные с **(+ a).** **ЛДС неустойчива (колебания).**

**Критерий устойчивости по корням**:

ЛДС устойчива, если корни характеристического уравнения лежат слева от мнимой оси и неустойчива, если хотя бы один из корней лежит справа от мнимой оси. Мнимая ось – граница устойчивости.

Исходя из расположения на комплексной плоскости, корни с отрицательными вещественными частями называются **левыми**, с положительными - **правыми**.

Поэтому **условие устойчивости линейной САУ** можно сформулировать следующим образом: для того, чтобы система была устойчива, **необходимо и достаточно, чтобы все корни ее характеристического уравнения были левыми**.

Если хотя бы один корень правый, то система **неустойчива**.

Если один из корней равен нулю, а остальные левые, то система находится **на** **границе апериодической устойчивости**.

Если равны нулю вещественные части одной или нескольких пар комплексно -сопряженных корней, то система находится **на границе колебательной устойчивости**.

**Критерий устойчивости**

Для устойчивости системы необходимо и достаточно, чтобы корни характеристического уравнения имели отрицательные действительные части, т.е. располагались в левой полуплоскости комплексной плоскости.

Если система устойчива, то все коэффициенты характеристического уравнения имеют одинаковые знаки.

1. **Общая характеристика стандартных функций работы с объектами САУ в Python**

Функции для работы с объектами САУ находятся в модуле matlab библиотеки control.

**Подключается через строку:**

**from control. matlab import \***

Родительский объект (класс) – LTI (Linear Time-Invariant System – ЛСС) – содержит информацию, не зависящую от конкретного представления и типа ЛСС.

Дочерние объекты – определяются конкретной формой представления ЛСС:

**tf-объект** (Transfer Function – передаточная функция) – характеризуется векторами коэффициентов полиномов числителя и знаменателя рациональной передаточной функции;

**zpk-объект** (Zero-Pole-Gain – нули – полюса – коэффициент передачи);

**ss-объект** (State Space – пространство состояний)

**а)** **Создание модели в виде передаточной функции:**

Общий вид: **W = tf(num, den)**

num и den – массивы коэффициентов полиномов числителя и знаменателя передаточной функции, расположенных в прядке убывания степеней.

**б) [num,den]=tfdata(sys)**

num и den – результаты работы функции: числитель и знаменатель передаточной функции.

**в) Переходная характеристика динамического объекта:**

Общий вид: **y,t = step(lti-объект)**

y – переходная характеристика, **t** – время, y и t одномерные массивы. Размерность массивов выбирается автоматически.

г) Функция **bode** предназначена для получения диаграммы Боде – графиков логарифмических АЧХ и ФЧХ.

Общий вид: **A,PH,W = bode(lti-объект)** – получение вектора амплитуды, фазы и частоты

д) Передаточная функция замкнутой системы определяется выражением (контур с отрицательной обратной связью):

H=feedback(H1,H2)

е) **Функиця nyquist** предназначена для получения диаграммы Найквиста (годографа).

Общий вид: **nyquist(lti-объект)** – построение диаграммы Найквиста заданного объекта.

**[Re,Im,W] = nyquist(lti-объект)** – получение векторов вещественной и мнимой частей комплексного коэффициента передачи для заданного набора частот и вектора частот.

ж) Получение полюсов модели: **pole(lti-объект).**

**p = pole(H1)** – возвращает массив корней характеристического уравнения.

з) Функция pzmap позволяет получить карту нулей и полюсов модели.

**[p,z] = pzmap(lti-объект)** – формируются векторы полюсов и нулей модели.

1. **Примеры создания и исследования ЛСС-моделей в Python.**

#Создание трех звеньев в виде придаточных функций и объединение их в единую систему (W1 и W2 последовательно, параллельно с W3)

W1 = tf([2,1],[1,1])

W2 = tf([3],[1,0])

W3 = tf([4,0],[4,1])

W = W1 \* W2 + W3

#Получение переходной характеристики y и времени t заданной модели W и построение графика

y,t = step(W)  
plt.plot(t,y)

plt.grid()

#Получение вектора амплитуды фазы и частоты и

#Построение диаграммы Боде – графиков логарифмических АЧХ и ФЧХ

mag, phase, omega = bode(W)

plt.figure(2)

plt.plot(omega,mag)  
plt.grid()

#Получение корней характеристического уравнения и вывод их в консоль

p = pole(W)

print(p)

#Нахождение максимального значения амплитуды вывод точки на график и координат в консоль

maxi = argmax(mag)

plt.scatter(omega[maxi],mag[maxi])

print(omega[maxi],mag[maxi])

plt.show()

1. **Получение временных характеристик типовых звеньев САУ в Python. Примеры**

(Временная) Переходная характеристика динамического объекта:

Общий вид: **y,t = step(lti-объект)**, где y – переходная характеристика, t – время, y и t одномерные массивы. Размерность массивов выбирается автоматически.

(Частотная) Функция **bode** предназначена для получения диаграммы Боде – графиков логарифмических АЧХ и ФЧХ.

Общий вид: **A,PH,W = bode(lti-объект)** – получение вектора амплитуды, фазы и частоты

Типовые звенья:

**1)Усилительное звено**

Передаточная функция: **w = tf([K],[1])**

График – прямая параллельная оси времени

**2)Интегрирующее звено**

Передаточная функция: **w = tf([K],[1,1])**

График – прямая под углом к оси времени

**3)Идеальное Дифференцирующее звено**

Передаточная функция: **w = tf([K,0],[1])**

**4)Реальное Дифференцирующее звено**

Передаточная функция: **w = tf([K\*Т,0],[Т,1])**

**5)Апериодическое звено первого порядка**

Передаточная функция: **w = tf([K],[T,1])**

**6)Апериодическое звено второго порядка**

Передаточная функция: **w = tf([K],[T1\*Т1,T2,1])**

**7)Колебательное звено**

При **p=T2/(2\*T1) < 1 и Т = Т1**

Передаточная функция: **w = tf([K],[T\*Т,T\*р,1])**

#Создание интегрирующего звена

W = tf([3],[1,1])

#Получение переходной характеристики y и времени t заданного звена W и построение графика

y,t = step(W)

plt.plot(t,y)

plt.grid()

plt.show()

W = tf([1, 0], [1, 1])

y, t = step(W)

plt.plot(t, y)

plt.show()

1. **Объединение типовых звеньев САУ, получение общей передаточной функции в Python**

**Передаточная функция звеньев:**

Последовательное соединение звеньев: H = H2\*H1

Параллельное соединение звеньев: H = H2+H1

Передаточная функция замкнутой системы определяется выражением (контур с отрицательной обратной связью):

H=feedback(H1,H2)

#Создание трех звеньев в виде придаточных функций и объединение их в единую систему (W1 и W2 последовательно, параллельно с W3)

W1 = tf([2,1],[1,1])

W2 = tf([3],[1,0])

W3 = tf([4,0],[4,1])

W = W1 \* W2 + W3

1. **Определение устойчивости ЛСС-систем в Python**

**Устойчивость системы** – способность ее возвращаться к состоянию установившегося равновесия после снятия возмущения, нарушившего это равновесие.

Оценка устойчивости ЛДС по корням характеристического уравнения:

Корни вещественны, отрицательны: ЛДС устойчивая без колебаний.

Комплексные с отрицательной вещественной частью: ЛДС устойчивая с колебаниями.

Один из корней равен 0: ЛДС нейтральная.

Корни мнимые (незатухающие колебания): ЛДС на границе устойчивости.

Корни вещественные положительные: ЛДС не устойчивая без колебаний.

Корни комплексные с положительной вещественной частью: ЛДС неустойчивая с колебаниями.

W = tf([2],[2,3,4])  
pf, ps = pole(W)  
print(pf,ps)  
if pf.real == 0 and ps.real == 0:  
 if pf.imag == 0 or ps.imag == 0:  
 print('Система нейтральная')  
 else:  
 print('Система на границе устойчивости')  
elif pf.real != 0 and ps.real != 0:  
 if pf.real > 0 and ps.real > 0:  
 print('Система не устойчивая')  
 elif pf.real < 0 and ps.real < 0:  
 print('Система устойчивая')  
 if pf.imag == 0 and ps.imag == 0:  
 print('Система без колебаний')  
 elif pf.imag != 0 and ps.imag != 0:  
 print('Система с колебаниями')  
else:  
 print('Не удалось определить устойчивость системы')

1. **Моделирование ЛСС-моделей в Xcos, пример**

**Теория линейных стационарных систем** – раздел теории динамических систем, изучающий поведение и динамические свойства .ЛСС. Используется для изучения процессов управления техническими системами, для цифровой обработки сигналов и в других областях науки и техники.

Для описания ЛСС могут применятся несколько способов:

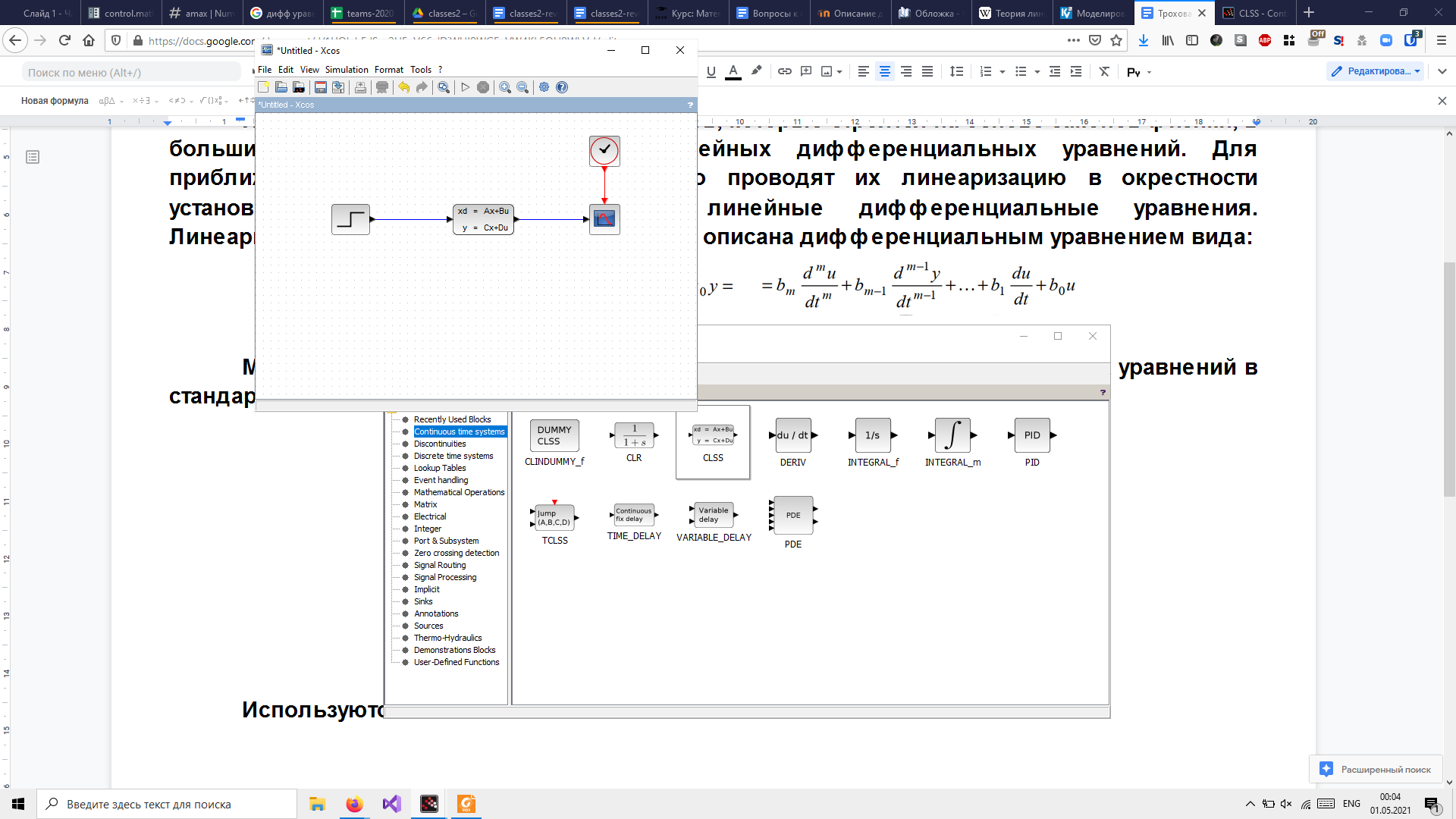
* ДУ
* Модель в пространстве состояний

Исходные уравнения динамики объектов, которые строятся на основе законов физики, в большинстве случаев имеют вид нелинейных ДУ. Для приближённого анализа и синтеза обычно проводят их линеаризацию в окрестности установившегося режима и получают линейные ДУ. Линеаризованная модель объекта может быть описана ДУ вида:

Модель в пространстве состояний связана с записью ДУ в стандартной форме Коши (в виде системы уравнений первого порядка):

**x** – вектор переменных состояния размера ; **u** – вектор входных сигналов (вектор управления) размера и **y** – вектор выходных сигналов размера . Кроме того, **A, B, C** и **D** – постоянные матрицы. Согласно правилам матричных вычислений, матрица **A** должна быть квадратной размера , матрица **C** - , матрица **B** имеет размер , матрица **C** - и матрица **D** - . Для систем с одним входом и одним выходом матрица **D** – скалярная величина.

**Используются блок CLSS.**

****

1. **Общая постановка задач идентификации. Параметрическая идентификация, алгоритм реализации**

**Задача**: по заданным входному и выходному воздействиям определить оператор системы.

Два вида задач идентификации – чёрный ящик: ДифУр не известен, только известен I/O; серый ящик: известны некоторые параметры.

**Параметрическая идентификация** – идент. параметров при известной структуре дин модели. В этом случае задача поиска параметров => параметрическая оптимизация (min ошибки модели).

Метод наименьших квадратов. Заметим, что для линейных по идентифицируемым параметрам моделей функция является положительно определенной квадратической функцией своих аргументов и для при , , , следовательно, существует единственная точка минимума, которая может быть определена из решения системы уравнений.

**Временной области идентификация:**

Пусть u – input, x = output; Задача - восстаносить диф.-, или интегральный оператор между ними.

Оценка временных и частотных характеристик динамических систем без параметризации модели M, что численно соответствует определению вектора значений искомой характеристики.

Временные характеристики:

•метод импульсного воздействия;

•метод ступенчатого воздействия;

•корреляционный метод;

**Метод импульсного воздействия**

Заключается в оценке импульсной характеристики системы

при подаче импульсного воздействия

Выход системы:

Вычислить оценку можно следующим образом:

В итоге получим оценку:

Ошибка оценки:

Недостатки метода:

• для уменьшения ошибки нужно увеличивать магнитуду α импульсного

воздействия, что может привести к проявлению нелинейных эффектов в системе;

• для физических непрерывных систем такое воздействие вообще может быть не допустимо условиями эксплуатации.

**Метод ступенчатого воздействия**

Заключается в оценке импульсной характеристики той же системы, но при подаче на вход ступенчатого воздействия

Выход системы:

Вычислить оценку можно следующим образом:

Ошибка оценки:

**Недостатки метода:**

• для уменьшения ошибки нужно увеличивать магнитуду α ступенчатого

воздействия;

• могут быть большие вычислительные ошибки (при малом α и сильном

уровне шума), как конфликт больших и малых чисел.

**Достоинства метода:**

• можно определять основные характеристики качества динамической

системы: запаздывание, усиление системы, постоянные времени;

• ступенчатое воздействие физически хорошо воспринимается системами (особенно непрерывными).

**24. Общая постановка обратной задачи динамики, примеры**

По полученному выходу после оператора получить вход.

Для механики: определение сил и моментов, действующих на меха-ю систему.

Пример задач: определение плотности, теплопроводности, упругости и т. д.

Пример (полный): движение точки описывается ДУ. – координата, – сила, действующая на точку.

Пример (полный) 2: Существует ненаблюдаемое напрямую тело, видна только его тень. Обратная задача состоит в определении формы ненаблюдаемого тела.

Проблемы:

уравнение для решения обратной задачи– могут быть нелинейными.

решение обычно не одно. Для единственности необходимы спец. ограничения.

“Постановка”

, где у– искомая входная функция.

1. **Исследование переходных процессов в Python, общие положения.**
2. **Алгоритм вычисления времени переходного процесса.**
3. **Алгоритм вычисления декремента, коэффициента динамичности и колебательности переходного процесса**
4. **Обзор пакетов имитационного и схемотехнического моделирования**
5. **Назначение и концепции моделирования в пакете Xcos.**
6. **Описание основных блоков и создание подсистем (суперблоков) в Xcos.**
7. **Моделирование технических систем, описанных интегро-дифференциальными уравнениями в Xcos.**
8. **Моделирование технических систем, описанных дифференциальными уравнениями в Xcos.**
9. **Сравнительный анализ формирования моделей в Xcos и в модулях «Электрические блоки» и «Mechanics»**
10. **Общие понятия теории автоматического управления (ТАУ)**
11. **Принципы управления и основные виды САУ.**
12. **Определение передаточной функции, получение передаточной функции по дифференциальному уравнению**
13. **Типовые звенья САУ, их временные характеристики**
14. **Передаточные функции типовых звеньев САУ**
15. **Соединение структурных звеньев САУ**
16. **Устойчивость линейных динамических систем**
17. **Общая характеристика стандартных функций работы с объектами САУ в Python**
18. **Примеры создания и исследования ЛСС-моделей в Python.**
19. **Получение временных характеристик типовых звеньев САУ в Python. Примеры**
20. **Объединение типовых звеньев САУ, получение общей передаточной функции в Python**
21. **Определение устойчивости ЛСС-систем в Python**
22. **Моделирование ЛСС-моделей в Xcos, пример**
23. **Общая постановка задач идентификации. Параметрическая идентификация, алгоритм реализации**
24. **Общая постановка обратной задачи динамики, примеры**