Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

ФИЛИПОВСКИЙ ВЛАДИМИР МИХАЙЛОВИЧ

ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Методические указания

Санкт-Петербург
2022

УДК 519.677 (62-503.56)

Филиповский В.М. **Параметрическая оптимизация системы автоматического управления**: Методические указания / В.М. Филиповский. – СПб, 2022. – 38 с.

Пособие соответствует ФГОС ВО по направлению подготовки 27.03.04 «Управление в технических системах» (уровень бакалавриата).

Рассматриваются принципы получения математических моделей систем автоматического управления и способы их моделирования с вычислением целевой функции на траекториях движения системы. Особенное внимание уделяется применению методов минимизации заданного функционала к задаче поиска минимума целевой функции с целью параметрической оптимизации исследуемой системы.

Содержание Указаний опирается на современные методы получения программных моделей динамических систем, решения систем дифференциальных уравнений и различные методы поиска минимума функций по одной и нескольким переменным.

Настоящие Указания предназначены студентам Высшей Школы Киберфизических систем и Управления (программы «Управление в технических системах») в качестве пособия при выполнении курсовых и лабораторных работ по курсам "Вычислительная математика" и "Математические модели технических систем".

Ключевые слова: Динамический объект, система автоматического управления, математическая модель, система дифференциальных уравнений, программная модель, целевая функция, функционал, экстремальная задача, метод минимизации, параметрическая оптимизация.

Содержание

Введение

Оптимизация – это процесс приведения объекта (системы) в оптимальное (наилучшее) состояние. Для проведения оптимизации необходимы: математическая модель объекта, целевая функция и оптимизационный алгоритм. Целевая функция формализует требования, предъявляемые к объекту (минимум ошибки, наибольшее быстродействие, увеличение надежности, снижение стоимости, максимизация прибыли и т.д.). Оптимизационный алгоритм ищет экстремум целевой функции.

Оптимизация осуществляется при помощи поисковых алгоритмов математического программирования и бывает структурной, параметрической и структурно-параметрической. В процессе структурной оптимизации оптимизируется структура объекта, в процессе же параметрической – подбираются параметры (номиналы) элементов таким образом, чтобы минимизировать (максимизировать) целевую функцию.

По наличии (или отсутствии) ограничений на целевую функцию и рабочие параметры различают оптимизацию при наличии ограничений или без ограничений. Так, если при синтезе системы необходимо, чтобы время перехода последней из одного состояния в другое было не больше какой-то заданной величины, то говорят о наложении ограничения на соответствующий критерий. Если же при этом требуется использовать номиналы элементов, значения которых должны попасть в какой-то заданный интервал (например, сопротивления должны быть не меньше 100 Ом и не больше 100 КОм), то тогда имеем дело с ограничениями на рабочие параметры.

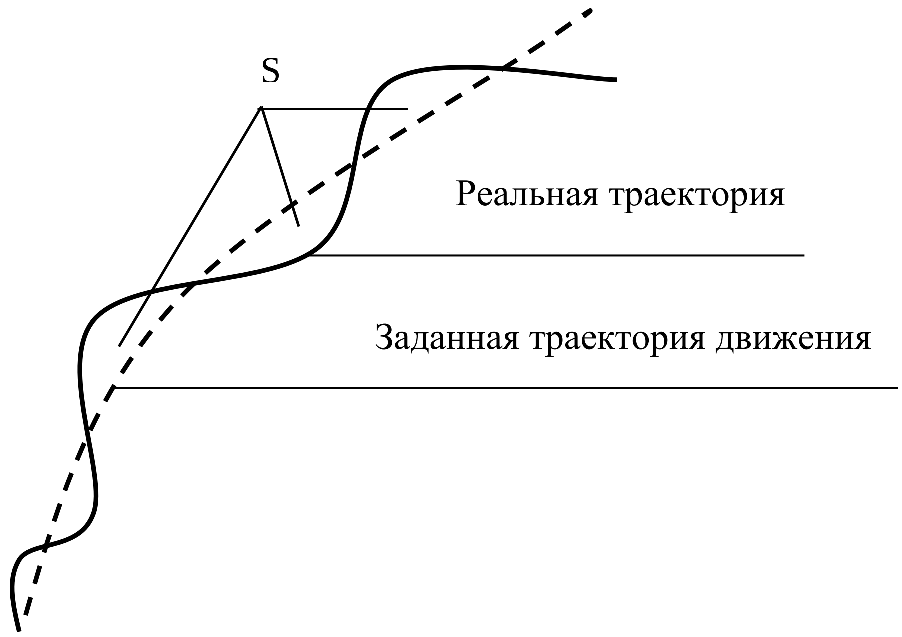
Следует заметить, что существующие оптимизационные алгоритмы обычно позволяют отыскать локальный оптимум целевой функции и не гарантируют нахождение глобального, но это не является критическим. Для увеличения вероятности нахождения глобального оптимума можно увеличить число итераций, использовать несколько алгоритмов, многократно запускать соответствующие алгоритмы и т.д. Современные системы автоматизированного проектирования (САПР) имеют в своем составе различные модули параметрического синтеза и оптимизации.

Автор выражает благодарность студенту кафедры СТУ (ныне аспиранту Высшей школы Киберфизических Систем и Управления) Мешковскому Е.О. за неоценимую помощь в подготовке данных Указаний.

1. Постановка и этапы решения задачи

Параметрическая оптимизация – задача по подбору значений параметров, наилучших для объекта определенной структуры (при условии минимума некоторого функционала - целевой функции), при соблюдении заданных ограничений.

В качестве примера целевая функция системы может быть задана функционалом, характеризующим качество системы в виде интеграла от ошибки – отклонения движения системы от заданного:



Как следует из рисунка, чем меньше площадь между кривыми, изображающими реальную и заданную траектории движения, тем меньше ошибка отработки заданного движения, то есть качественнее система.

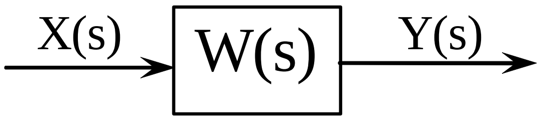
Весь процесс параметрической оптимизации состоит из следующих этапов:

1. Получение математической модели объекта (обычно в виде системы дифференциальных уравнений);
2. Решение системы дифференциальных уравнений и вычисление на траекториях системы (объекта) значения функционала – интегральной характеристики качества системы;
3. Преобразование программы вычисления функционала в функцию, зависящую от варьируемых параметров, подлежащих определению (уточнению) в процессе поиска минимума этого функционала.
4. Программная реализация численных методов поиска минимума функций от одной и нескольких переменных.
5. Отладка программ минимизации с необходимой адаптацией методов на простейших эталонных функциях.
6. Использование программ поиска минимума для решения задачи параметрической минимизации объекта (минимизация функции, определенной на этапе 3).

2. Преобразование формы математической модели системы автоматического управления

Математическая модель оптимизируемого объекта (системы автоматического управления) может быть представлена в различных формах: графы, структурная схема, система дифференциальных или разностных уравнений… При синтезе чаще используют модель системы в виде структурной схемы. Для решения задачи параметрической оптимизации удобнее работать с моделью в виде системы дифференциальных уравнений первого порядка. Следовательно, необходимо выполнить преобразование структурной схемы к системе дифференциальных уравнений.

Структурная схема системы представляет собой совокупность элементарных звеньев объекта и связей между ними, причем звенья на этой схеме изображаются прямоугольниками с соответствующими передаточными функциями:



Здесь $W(s) = \dfrac{Y(s)}{X(s)}$; $W(s) = \dfrac{R(s)}{Q(s)}$ - передаточная функция звена, $X(s)$ и $Y(s)$ – изображения входного и выходного сигналов, $R(s)$ и $Q(s)$ – алгебраические полиномы от s, s – переменная Лапласа. С изображением Лапласа студенты впервые сталкиваются при изучении курса "Теоретических основ электротехники". Передаточные функции они изучат в курсе "Теория управления" на более старших курсах.

Для перехода от структурной схемы к дифференциальным уравнениям необходимо по передаточной функции каждого звена получить соответствующее ему дифференциальное (и/или алгебраическое) уравнение. Из курсов "Теория Автоматического Управления" и "Математические Модели" известно, что математические модели звеньев в пространстве Лапласа и временной области тождественны (изоморфны), если они записаны в операторной форме: типа «вход – выход». С этим фактом студенты более подробно ознакомятся позднее.

Из определения передаточной функции звена следует операторное уравнение «вход – выход»:

$$W(s) = \frac{Y(s)}{X(s)}; \quad W(s) = \frac{R(s)}{Q(s)}; \quad \Rightarrow \quad \frac{Y(s)}{X(s)} = \frac{R(s)}{Q(s)}; \quad \Rightarrow$$

$$\Rightarrow \; Q(s) \cdot Y(s) = R(s) \cdot X(s).$$

Полученное уравнение с точностью до коэффициентов идентично дифференциальному уравнению того же звена (при нулевых начальных условиях):

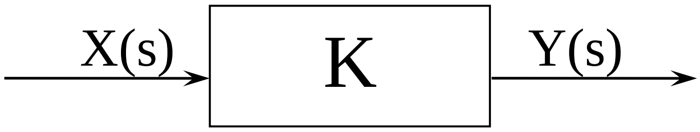
$$Q(p) \cdot y(t) = R(p) \cdot x(t).$$

Здесь $p = \dfrac{d}{dt}$ - оператор дифференцирования, t – время.

Данное дифференциальное уравнение в дальнейшем приводится к системе дифференциальных уравнений первого порядка (в общем случае), записанных в форме Коши.

В структурных схемах различают следующие элементарные звенья:

1. ***Пропорциональное звено (Идеальный усилитель):***



K – некоторый коэффициент, число.

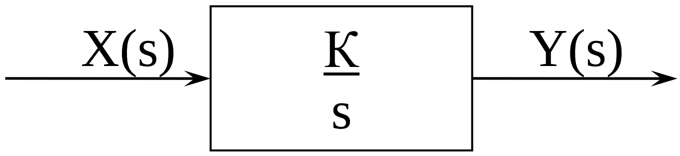
Операторное уравнение для данного звена имеет вид:

$$Y(s) = K * X(s).$$

Ему соответствует вырожденное алгебраическое уравнение:

$$y(t) = K * x(t).$$

2. ***Интегрирующее звено (Интегратор):***



Передаточная функция этого звена: $W(s) = \dfrac{K}{s}$.

В соответствии с определением передаточной функции имеет место соотношение

$$Y(s) = \frac{K}{s} \cdot X(s),$$

из которого следует операторное уравнение «вход – выход»:

$$s \cdot Y(s) = K \cdot X(s),$$

а после замены переменной Лапласа s на оператор дифференцирования $p = \dfrac{d}{dt}$ можно получить дифференциальное уравнение:

$$\frac{d}{dt} y(t) = K \cdot x(t),$$

которое обычно записывается в виде:

$$\dot{y}(t) = K \cdot x(t).$$

3. ***Реальный усилитель (Апериодическое звено):***

$$X(s) \longrightarrow \boxed{\frac{K}{1+Ts}} \longrightarrow Y(s)$$

Выполняя аналогичные действия, последовательно получают:

$$Y(s) = \frac{K}{1+Ts} \cdot X(s),$$

откуда уравнение «вход – выход»:

$$(1+Ts) \cdot Y(s) = K \cdot X(s),$$

затем, после замены переменной s на оператор p, и дифференциальное уравнение:

$$y(t) + T\frac{dy(t)}{dt} = Kx(t),$$

которое в форме Коши будет иметь вид:

$$\dot{y}(t) = -\frac{1}{T}y(t) + \frac{K}{T}x(t).$$

4. ***Звено второго порядка (Колебательное звено):***

$$X(s) \longrightarrow \boxed{\frac{K}{1+eTs+T^2s^2}} \longrightarrow Y(s)$$

Как и ранее, получают:

$$Y(s) = \frac{K}{1+eTs+T^2s^2} \cdot X(s),$$

затем операторное уравнение «вход – выход»:

$$\left(1+eTs+T^2s^2\right) \cdot Y(s) = K \cdot X(s),$$

откуда - дифференциальное уравнение второго порядка:

$$y(t) + eT\frac{dy(t)}{dt} + T^2\frac{d^2y(t)}{dt^2} = K \cdot x(t)$$

или

$$y(t) + eT\dot{y}(t) + T^2\ddot{y}(t) = K \cdot x(t).$$

Последнее уравнение необходимо привести к системе дифференциальных уравнений первого порядка. Данное преобразование можно выполнить множеством способов. Наиболее употребительным является способ приведения к системе дифференциальных уравнений в нормальной форме с использованием вектора фазовых переменных.

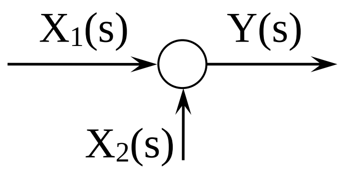
Таким образом, с помощью замены переменных:

$$z_1(t) = y(t); \quad z_2(t) = \dot{y}(t),$$

вместо исходного уравнения второго порядка получают систему двух дифференциальных уравнений первого порядка в форме Коши:

$$\begin{cases} \dot{z}_1(t) = z_2(t); \\ \dot{z}_2(t) = -\dfrac{1}{T^2} z_1(t) - \dfrac{e}{T} z_2(t) + \dfrac{K}{T^2} x(t). \end{cases}$$

5. Сумматор:



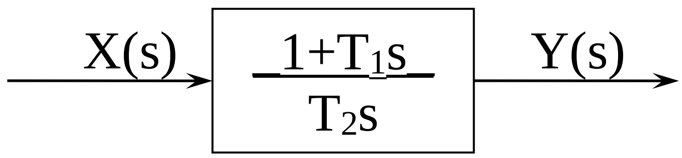
Данному элементу структурной схемы соответствует два алгебраических уравнения: в операторах Лапласа:

$$Y(s) = X_1(s) + X_2(s)$$

и во временной области:

$$y(t) = x_1(t) + x_2(t).$$

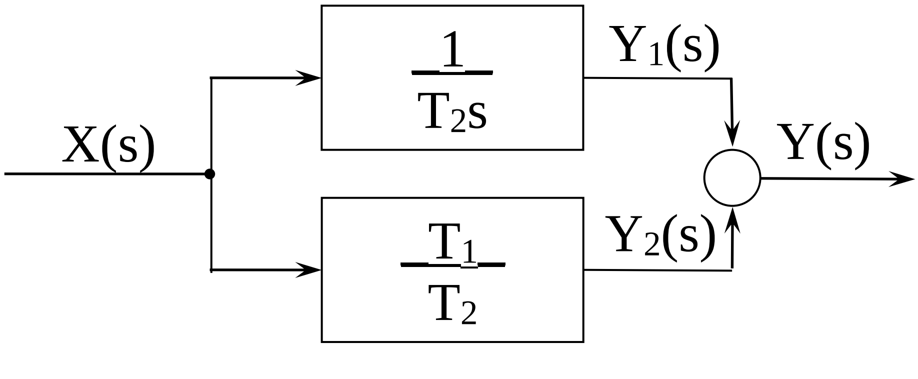
6. Форсирующее звено (Регулятор* или *Корректирующее устройство):



Для преобразования математической модели форсирующего звена передаточная функция записывается в виде:

$$W(s) = \frac{1 + T_1 s}{T_2 s} = \frac{1}{T_2 s} + \frac{T_1}{T_2}.$$

Измененной передаточной функции соответствует структурная схема:



для которой по описанной ранее методике получают дифференциальное уравнение:

$$\dot{y}_1(t) = \frac{1}{T_2} x(t)$$

и два алгебраических:

$$y_2(t) = \frac{T_1}{T_2} x(t) \quad \text{и} \quad y(t) = y_1(t) + y_2(t) \,.$$

Так как математическая модель объекта призвана отражать в математической форме свойства этого объекта (модель адекватна объекту), то имеет смысл учесть и такую особенность некоторых звеньев, как ограничение их выходных сигналов.

Чаще всего ограничению подвержены сигналы в корректирующих устройствах (регуляторах). Для рассматриваемого звена это переменные $y(t)$ и $y_1(t)$, то есть выходной сигнал регулятора и его интегральная составляющая соответственно.

Реализация ограничения значений $y_1(t)$ выполняется с учетом правила:

если **переменная** $y_1(t)$ ***достигла*** **своего крайнего значения и собирается его** ***перейти*** **– выйти** ***за границу области*** **допустимых значений** $\Big($**об этом можно судить по знаку производной:** $\dot{y}_1(t)$ $\Big)$**,** ***то*** **значение** ***производной*** **требуется «*****обнулить*****»:** $\dot{y}_1(t) = 0$**.**

Ограничение сигнала $y(t)$ выполняется с помощью стандартной «срезки»:

если **значение переменной** $y(t)$ ***вышло за границу*** **области допустимых значений,** ***то*** **выполняется корректировка путем** ***записи*** **в переменную** $y(t)$ ***граничного*** **значения.**

Описанные процедуры учета ограничений применяются в программных реализациях с использованием соответствующих операторов:

- для интегральной составляющей выходного сигнала регулятора:

$$\dot{y}_1(t) := \begin{cases} 0, & \textit{если} \quad \mathrm{y}_1(t) \ge \mathrm{y}_{1\max} \;\; \textit{и} \;\; \dot{y}_1(t) > 0; \\ 0, & \textit{если} \quad \mathrm{y}_1(t) \le \mathrm{y}_{1\min} \;\; \textit{и} \;\; \dot{y}_1(t) < 0; \\ \dot{y}_1(t) - \textit{в остальных случаях...} & \end{cases}$$

как следует из последней записи, производная $\dot{y}_1(t)$ не меняет своего значения ***в остальных случаях***!

- для выходной переменной регулятора:

$$y(t) := \begin{cases} y_{max}, & \textit{если} \quad y(t) > y_{max}; \\ y(t), & \textit{если} \quad y_{min} \le y(t) \le y_{max}; \\ y_{min}, & \textit{если} \quad y(t) < y_{min}. \end{cases}$$

или, с использованием кусочно-линейного оператора:

$$y(t) := 0.5 \cdot \left((y_{1min} + y_{1max}) + |y(t) - y_{1min}| - |y(t) - y_{1max}| \right).$$

После того, как для всех элементов (звеньев) структурной схемы динамического объекта получены соответствующие им дифференциальные и алгебраические уравнения, выполняется сведение полученного набора выражений к системе дифференциальных уравнений в форме Коши. Для этого вводится вектор состояния системы, размерность которого равна количеству имеющихся дифференциальных уравнений, и выполняется соответствующая замена переменных.

На следующем этапе преобразования математической модели строится программная модель, где организовано решение полученной системы дифференциальных уравнений одним из известных методов. Наиболее часто для решения задачи Коши используется программа RKF45 из вычислительного пакета Дж. Форсайта.

Рекомендуется следующая последовательность действий для получения программной модели динамического объекта:

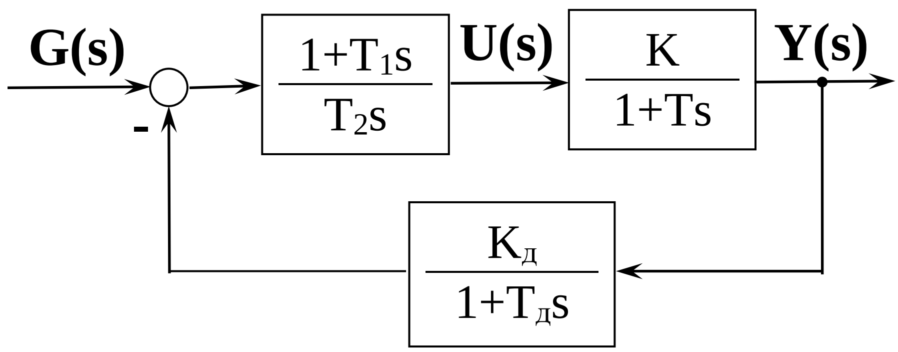
1. В данной структурной схеме выполняется преобразование корректирующих устройств (регуляторов), разбивая их на два параллельно соединенные звена.
2. В преобразованной структурной схеме обозначаются (именуются) выходные переменные всех звеньев системы.
3. Для каждого звена получаются соответствующие дифференциальные и алгебраические уравнения.
4. Переход к векторной форме математической модели системы заключается в том, что переменные, временные зависимости которых могут быть определены только путем решения дифференциальных уравнений, заменяются компонентами вектора состояния. Данные переменные представлены в полученных уравнениях соответствующими производными. Размерность вводимого вектора состояния равна числу полученных дифференциальных уравнений.
5. С учетом произведенной замены выполняется замещение старых переменных новыми именами (соответствующими компонентами вектора

состояния) во всех уравнениях (алгебраических и дифференциальных). Полученную систему уравнений показать преподавателю.

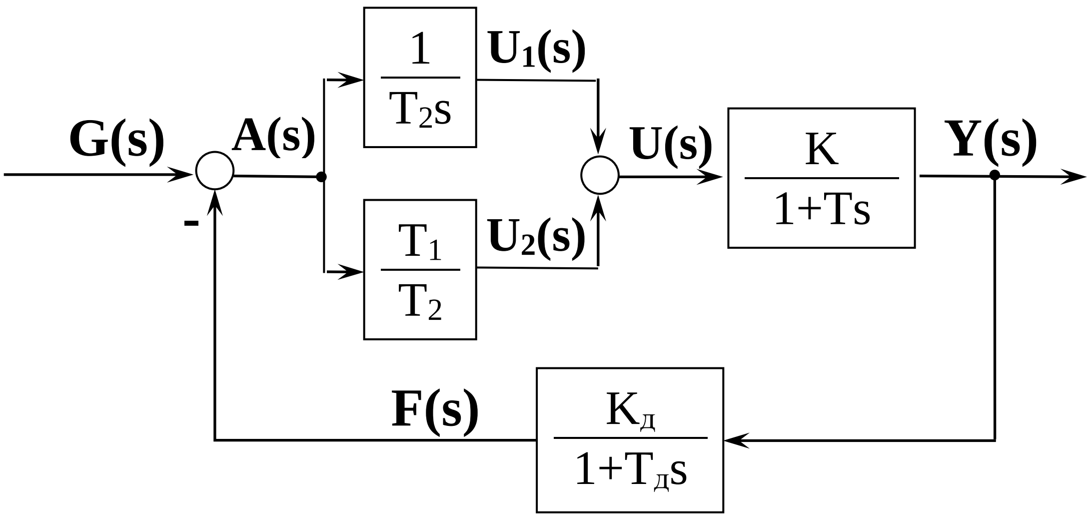
6. Все преобразованные уравнения записываются в процедуру FUN, в которой организуется вычисление производных вектора состояния. Порядок операторов этой процедуре должен соответствовать реальному прохождению сигналов в рассматриваемой системе.
7. Значения параметров всех звеньев системы задаются в главной программе с последующей передачей через ОБЩУЮ ОБЛАСТЬ в процедуру FUN.
8. С помощью программы RKF45 выполняется решение (интегрирование) полученной системы дифференциальных уравнений. Результаты решения в виде графиков изменения переменных вектора состояния показываются преподавателю. Также рекомендуется построить все графики сигналов в регуляторах.
9. Вводятся в FUN операторы, ограничивающие выходные переменные отдельных звеньев системы. При повторном моделировании выполняется коррекция (под руководством преподавателя) входных сигналов системы управления.
10. Правильность введения операторов, ограничивающих выходные переменные звеньев, проверяется при интегрировании системы дифференциальных уравнений на двойном временном промежутке: $t \in [0, 4\pi]$. Проверка заключается в наблюдении их симметричного действия как при положительных граничных значениях этих переменных, так и отрицательных.

3. Получение математической модели заданного динамического объекта

Далее в качестве примера показано получение математической модели системы автоматического управления по ее структурной схеме:



На начальном этапе выполняют преобразования в структурной схеме с учетом представления передаточной функции форсирующего звена суммой двух элементарных функций (как показано в п.6 предыдущего раздела). Здесь же соответствующими именами (буквами) обозначают все выходные (и входные) сигналы звеньев:



Затем для всех элементов структурной схемы получают соответствующие дифференциальные и алгебраические уравнения:

$$a(t) = g(t) - f(t);$$

$$\dot{u}_1(t) = \frac{1}{T_2} a(t);$$

$$u_2(t) = \frac{T_1}{T_2} a(t);$$

$$u(t) = u_1(t) + u_2(t);$$
$$\dot{y}(t) = -\frac{1}{T} y(t) + \frac{K}{T} u(t);$$
$$\dot{f}(t) = -\frac{1}{T_д} f(t) + \frac{K_д}{T_д} y(t).$$

В результате получили три дифференциальных и три алгебраических уравнения.

Для перехода к системе дифференциальных уравнений вводим вектор переменных состояния $X \in R^3$, причем

$$X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} = \begin{pmatrix} u_1(t) \\ y(t) \\ f(t) \end{pmatrix}.$$

С учетом введенных обозначений переписывается система уравнений:

$$a(t) = g(t) - x_3(t);$$
$$\dot{x}_1(t) = \frac{1}{T_2} a(t);$$
$$u_2(t) = \frac{T_1}{T_2} a(t);$$
$$u(t) = x_1(t) + u_2(t);$$
$$\dot{x}_2(t) = -\frac{1}{T} x_2(t) + \frac{K}{T} u(t);$$
$$\dot{x}_3(t) = -\frac{1}{T_д} x_3(t) + \frac{K_д}{T_д} x_2(t).$$

В получившейся системе уравнений подстановку значений соответствующих переменных в дифференциальные уравнения выполнять не обязательно. Лучше оставить все уравнения в первозданном виде, выполнив лишь необходимую замену переменных компонентами вектора состояния (как сделано в примере). Это правило обосновывается тем, что решение системы (интегрирование дифференциальных уравнений) обычно выполняется с помощью соответствующих программ, большинству из которых (как, например, RKF45) для получения очередного значения вектора состояния $X(t+h)$ необходимо иметь только программную реализацию вычисления вектора $\dot{X}(t)$.

То есть требуется написать процедуру, в которой было бы организовано вычисление значений вектора производных состояния системы. В этой процедуре должны быть предусмотрены операторы, вычисляющие значения входных сигналов системы (например $g(t)$, как в рассматриваемом примере), а также учитывающие нелинейности отдельных звеньев.

4. Вычисление функционала качества на траекториях движения объекта

Как отмечено в разделе 1, для задачи параметрической оптимизации необходимо сформировать целевую функцию (функционал), количественно характеризующую поведение системы (объекта), зависящую от варьируемых (изменяемых) параметров этого объекта. В качестве примера целевая функция может определена как площадь, образуемая отклонением реальной траектории движения относительно заданной.

Математически данный функционал может задан интегралом:

$$I = \int_{t_{нач}}^{t_{кон}} \left| у_{зад}(t) - у_{реал}(t) \right| dt.$$

Понятно, что это не единственный способ задания целевой функции. Под знаком интеграла может быть не обязательно абсолютная величина разности, а ее квадрат, не только отклонение реальной траектории движения от заданной, но и производная этого отклонения:

$$I = \int_{t_{нач}}^{t_{кон}} \left(\left(у_{зад}(t) - у_{реал}(t) \right)^2 + k \cdot \left(\dot{у}_{зад}(t) - \dot{у}_{реал}(t) \right)^2 \right) dt.$$

Следует отметить, что целевая функция, как и значение весового коэффициента ***k***, задается экспертом, специалистом в данной предметной области. Программист-математик всего лишь реализует вычисление этого функционала. С учетом того, что объект представлен математической моделью в виде системы дифференциальных уравнеий (как показано в предыдущем разделе), поведение этого объекта может быть получено численными методами. Решение системы дифференциальных уравнений на ЭВМ всегда выполняется пошагово, путем получения последовательных значений как вектора состояния системы $X(t_k)$, так и выходной координаты $y_{реал}(t_k)$, $\kappa = 0, 1, 2, 3, \ldots, n.$ В этом случае вычисление функционала $\boldsymbol{I}$ может быть сведено к суммированию:

$$I \cong \sum_{k=1}^{n} \left(\left(у_{зад}(t_k) - у_{реал}(t_k) \right)^2 + k \cdot \left(\dot{у}_{зад}(t_k) - \dot{у}_{реал}(t_k) \right)^2 \right) h =$$

$$= \sum_{k=1}^{n} \left(\left(\varepsilon(t_k) \right)^2 + k \cdot \left(\dot{\varepsilon}(t_k) \right)^2 \right) h \cong$$

$$\cong \sum_{k=1}^{n} \left(\left(\varepsilon(t_k) \right)^2 \cdot h + k \cdot \frac{\left(\varepsilon(t_k) - \varepsilon(t_{k-1}) \right)^2}{h} \right).$$

Здесь $\varepsilon(t_k) = y_{зад}(t_k) - y_{реал}(t_k)$, $h = t_k - t_{k-1}$ - шаг интегрирования, $n = \left(t_{кон} - t_{нач}\right)/h$.

Практически всегда $y_{зад}(t)$ пропорционально входному сигналу $g(t)$. При выполнении данной работы для всех вариантов (см. Приложение П2) можно положить $y_{зад}(t) = g(t) / k_{дс}$.

Таким образом, задача вычисления функционала есть последовательное суммирование, выполняемое на каждом шаге решения системы дифференциальных уравнений, в соответствии с алгоритмом, в вычислительном блоке 4:

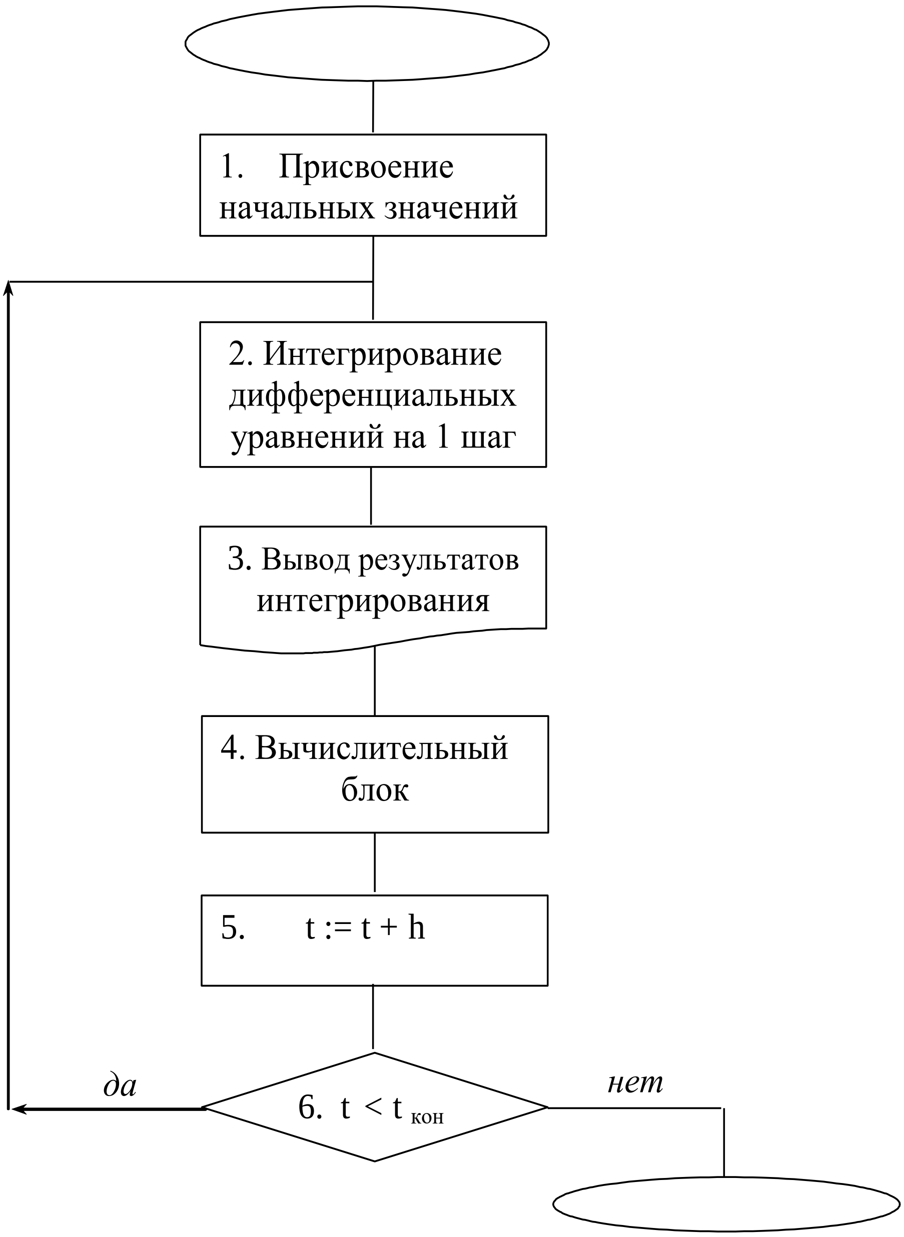


Рис. Схема алгоритма программы интегрирования системы дифференциальных уравнений и вычисления функционала.

Рекомендуется при программной реализации вычисления функционала:

1. Первоначально положить при вычислении функционала значение ***k*** равным 0 (нулю).
2. Результат вычисления функционала ***I*** сравнить с численным значением, полученным по кривой $\varepsilon^2(t)$, где $\varepsilon(t) = x(t) - k_{\partial c} y(t)$ (по площади фигуры, образованной этой кривой и осью абсцисс). Для получения зависимости $\varepsilon^2(t)$ требуется вставить в программу соответствующие операторы.
3. Подобрать значение ***k***, чтоб значение функционала ***I*** выросло в 2-3 раза по сравнению с первоначальным (полученным при ***k*** =0).

Полученную программу следует преобразовать в подпрограмму-функцию, возвращающую значение функционала, входными параметрами которой должны быть изменяемые (варьируемые) параметры объекта. Блок 3 алгоритма (вывод результатов интегрирования) можно исключить.

Рекомендуется:

1. В полученной функции с целью корректной передачи значений параметров функции в подпрограмму FUN ввести операторы:

```
Real function ff(T31, T41)
common /PAR/ T3, T4, K0,...
T3=T31
T4=T41
…
```

2. Проверить функцию ***ff***, вычисляющую функционал, на "реагирование" на изменение ее параметров. Эта проверка может быть выполнена с помощью операторов:

```
T3n=T3/2
T3k=T3*2
Th=(T3k-T3n)/50
do  tt=T3n,T3k,Th
yy=ff(tt,T4)
write (12,*) tt, yy
end do
Если (T3>0) stop 123
…
```

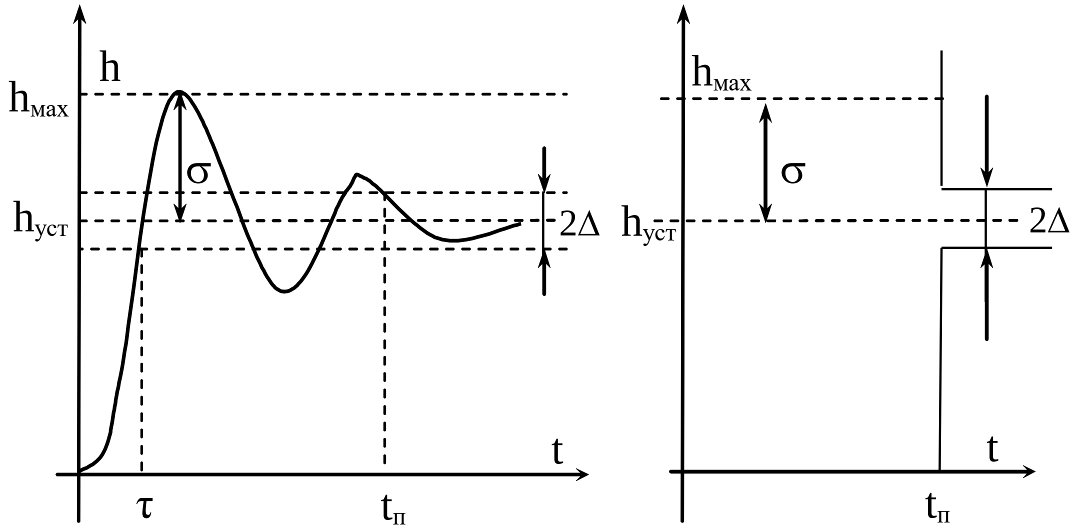
   По другому параметру проверка выполняется аналогичным образом. В результате должны получиться унимодальные гладкие кривые (данные для построения графиков записываются в файл for012.dat).

3. Если при каком-то значении параметра (T3 или T4) наблюдается при вычислении функционала переполнение разрядной сетки, необходимо в момент времени ***t***, когда значения переменных вектора состояния становятся "большими", выполнить оператор (например):

```
I=I+10/t,
```

прекратить дальнейшее интегрирование и перейти в конец функции ***ff***.

Следует отметить, что целевая функция (функционал), используемая в задаче параметрической оптимизации, определяется не обязательно интегральной оценкой качества, как указано в начале данного раздела. В некоторых случаях целевая функция может быть сформулирована по переходной характеристике системы относительно задающего воздействия.



Количественная оценка качества системы по переходной характеристике определяется показателями: перерегулированием $\boldsymbol{\sigma}$, временем первого согласования $\boldsymbol{\tau}$, временем достижения первого амплитудного значения, временем регулирования (временем переходного процесса) $\boldsymbol{t_n}$.

Перерегулирование (определяется величиной первого выброса) - отношение разности максимального значения переходной характеристики и ее установившегося значения к величине установившегося значения. Измеряется обычно в процентах.

$$\sigma = \frac{h_{\max} - h_y}{h_y} \cdot 100\% .$$

Время регулирования - длительность переходного процесса. Правда, в идеальной системе переходный процесс бесконечен. Поэтому временем регулирования считают тот интервал времени, по истечении которого отклонения переходной характеристики от установившегося значения не превышают Δ. Значения Δ обычно принимают 5%, 2%, а иногда и 1%. Но такой выбор всегда оговаривается.

В этом случае целевая функция может быть задана как:

$$I = \mathrm{k}_1\sigma + \mathrm{k}_2\tau + \mathrm{k}_3 t_n ,$$

где $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ - весовые коэффициенты, задаваемые экспертами.

Имеется ввиду, что чем лучше переходная характеристика (чем меньше перечисленные выше показатели), тем лучше система будет отрабатывать произвольное задающее воздействие.

5. Минимизация целевой функции

Автоматизированные методы *параметрической оптимизации* (минимизации целевой функции) основаны на численных методах поиска минимума функций одной и нескольких переменных. И, как показывает практика, большинство методов минимизации функции нескольких переменных используют методы минимизации функции одной переменных.

5.1. О минимизации функций одной переменной

Задачи одномерной минимизации представляют собой простейшую математическую модель оптимизации, в которой целевая функция зависит от одной переменной, а допустимым множеством является отрезок вещественной оси:

$$f(x) \to \min, \quad x \in [a, b].$$

Максимизация целевой функции $f(x) \to \max, \quad x \in [a, b]$ эквивалентна минимизации противоположной функции: $-f(x) \to \min, \quad x \in [a, b]$, поэтому, не умаляя общности, можно рассматривать только задачи поиска минимума.

Имеют место следующие определения:

1. Число $x^* \in U = [a, b]$ называется ***точкой глобального*** *(абсолютного)* ***минимума*** или просто *точкой минимума* функции $f(x)$ на множестве U, если $f(x^*) \le f(x)$ для всех $x \in U$.

Значение $f^* = f(x^*) = \min\limits_{u} f(x)$ называют ***глобальным*** *(абсолютным)* ***минимумом*** или просто *минимумом* функции $f(x)$ на множестве U.

2. Число $\tilde{x} \in U = [a, b]$ называется *точкой* ***локального минимума*** функции $f(x)$, если $f(\tilde{x}) \le f(x)$ для всех $x \in U$, достаточно близких к $\tilde{x}$, то есть, если существует ε > 0 такое, что это неравенство выполняется для любого $x \in \{x \,|\, x \in U, |x - \tilde{x}| < \varepsilon\}$.

3. Пусть функция $f(x)$ ограничена снизу на множестве $U = [a, b]$, то есть $f(x) \ge A > -\infty$ для всех $x \in U$. Число f_* называется ***точной нижней гранью*** функции $f(x)$ на множестве U ($f_* = \inf\limits_{U} f(x)$), если $f(x) \ge f_*$ при всех $x \in U$ и для любого ε > 0 найдется точка $x_\varepsilon \in U$ такая, что $f(x_\varepsilon) < f_* + \varepsilon$ (то есть среди значений $f(x)$ на множестве U найдутся как угодно близкие к f_*).

Для неограниченных снизу функций $f(x)$ полагают $f_* = -\infty$.

4. Из математического анализа известно (согласно *теореме Вейерштрасса*), что непрерывная на замкнутом ограниченном множестве функция достигает на этом множестве своих минимального и максимального значений.

5. Функция $f(x)$ называется ***унимодальной*** на отрезке $[a, b]$, если она непрерывна на $[a, b]$ и существуют числа α и β, $a \le \alpha \le \beta \le b$, такие, что:

1) если $a < \alpha$, то на отрезке $[a, \alpha]$ функция $f(x)$ монотонно убывает;

2) если $\beta < b$, то на отрезке $[\beta, b]$ функция $f(x)$ монотонно возрастает;

3) при $x \in [\alpha, \beta]$ $f(x) = f^* = \min\limits_{[a,\, b]} f(x)$.

Унимодальные функции обладают свойством: их локальный минимум является одновременно и глобальным.

Применение некоторых методов одномерной минимизации возможно только в случае, если скорость изменения целевой функции $f(x)$ на любом участке отрезка $[a, b]$ ограничена некоторым числом, одним и тем же для всех участков. То есть $f(x)$ удовлетворяет на $[a; b]$ ***условию Липшица***.

6. Функция $f(x)$ удовлетворяет на отрезке $[a; b]$ условию Липшица, если существует такое число $\boldsymbol{L > 0}$ (константа Липшица), что для всех x' и x'', принадлежащих $[a, b]$, выполняется неравенство $\left|f(x') - f(x'')\right| \le L\left|x' - x''\right|$.

Для практического определения константы Липшица используется формула конечных приращений: для произвольных точек $x', x'' \in [a, b]$ имеем: $f(x') - f(x'') = f(\xi)(x' - x'')$, где ξ– некоторая точка, лежащая между x' и x''. Отсюда с учетом условия $f'(\xi) \le \max\limits_{[a,b]} f'(x) = L$ определяется искомая величина.

Для решения задачи минимизации функции $f(x)$ на отрезке $[a, b]$ на практике, как правило, применяют приближенные методы. Они позволяют найти решение этой задачи с необходимой точностью в результате определения конечного числа значений функции $f(x)$ и ее производных в некоторых точках отрезка $[a, b]$. Методы, использующие только значения функции и не требующие вычисления ее производных, называются *прямыми методами* минимизации.

Большим достоинством прямых методов является то, что от целевой функции не требуется дифференцируемости и, более того, она может быть не задана в аналитическом виде. Единственное, на чем основаны алгоритмы прямых методов минимизации, это возможность определения значений $f(x)$ в заданных точках.

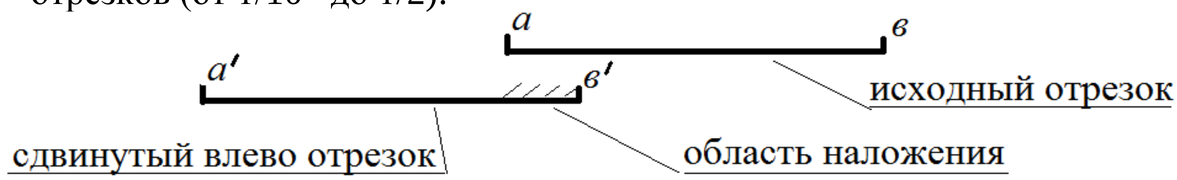
Самым слабым требованием на функцию $f(x)$, позволяющим использовать прямые методы, является ее унимодальность. Поэтому для упрощения задачи будем считать функцию $f(x)$ унимодальной на отрезке $[a, b]$ (что справедливо в окрестности локального минимума).

К прямым методам поиска минимума функции $f(x)$ на отрезке $[a, b]$ относятся метод перебора, методы исключения отрезка (деления отрезка пополам и золотого сечения), методы парабол, касательных, ломаных и так далее. В методах парабол, касательных, ломаных рекомендуется на каждом их шаге использовать метод исключения отрезков. Данный прием позволит организовать определение минимума функции $f(x)$ с заданной точностью.

Наименее эффективным из перечисленных является метод перебора. Для обеспечения заданной точности $\boldsymbol{\varepsilon}$ нахождения минимума необходимо выполнить деление отрезка на $\boldsymbol{n}$ частей: $n \geq \frac{b-a}{\varepsilon}$. В остальных методах точность будет достигнута, когда длина последнего вложенного отрезка будет сопоставима с величиной заданного $\boldsymbol{\varepsilon}$.

Рекомендации:

1. После самостоятельного изучения метода поиска минимума функции показать преподавателю алгоритм его реализации с целью проверки правильности понимания метода с возможной последующей его коррекцией и необходимой адаптацией.
2. Для большинства практических задач является достаточным, когда величина заданной погрешности определяется в пределах 1% (одного процента) от длины исходного отрезка $[a, b]$, то есть принимают $\varepsilon = 0.01 \cdot (b-a)$.
3. В случае, когда минимум функции $f(x)$ отыщется у границы множества $U = [a, b]$, то есть рядом с точками $\boldsymbol{a}$ или $\boldsymbol{b}$, можно попытаться отыскать локальный экстремум функции, сдвигая в соответствующую сторону отрезок $[a, b]$, при условии, что параметр x не выйдет за пределы допустимых значений области определения $f(x)$.
4. При сдвиге отрезка длина нового отрезка должна совпадать с длиной исходного отрезка (не изменяется). Сдвинутый отрезок накладывается на часть исходного отрезка (сдвиг с наложением). Наложение отрезков может составлять любую (по усмотрению студента) часть длины этих отрезков (от 1/10 до 1/2):



5. Если минимум функции оказался внутри исследуемого отрезка, процедуру поиска минимума (равно как и сдвиги отрезка) необходимо закончить.

После написания программы поиска минимума функции одной переменной следует преобразовать ее в функцию вида

$$\text{arg\_min1}(\text{fun}, \text{a}, \text{b}),$$

возвращающую значение аргумента $\tilde{x}$, при котором внешняя (посылаемая в качестве параметра) функция $fun(x)$ достигает наименьшего значения на отрезке $[a, b]$ (или в его ближайшей окрестности):

$$fun(x) \xrightarrow[{[a,\ b]}]{} fun(\tilde{x}) = \min_{[a, b]}.$$

В программе должны быть записаны следующие операторы:

```
external fun      // объявление внешней функции, посылаемой
                  //         параметром  в другую функцию, например:
x=arg_min1(fun, a, b)
...
```

Рекомендуется функцию `arg_min1(fun, a, b)` протестировать на некоторых эталонных функциях, например:

$$fun1(x) = x^2, \quad fun2(x) = \left(x-2\right)^2, \quad fun3(x) = x^6,$$

$$fun4(x) = \left(x-2\right)^6, \quad fun5(x) = \sin(x), \quad fun6(x) = \cos(x),$$

$$fun6(x) = \cos(x), \quad fun7(x) = \sin(5x), \quad fun8(x) = \cos(5x),$$

а также гиперболах: $fun9(x) = (x)^{-1}, \quad fun10(x) = -(x)^{-1},$

определяя отрезки $[a, \quad b]$ как содержащие минимум функции, так и слева и справа от него.

5.2. Методы минимизации функций нескольких переменных

Определения, сформулированные выше для функции одной переменной, можно обобщить для случая функций многих переменных. Пусть функция *n* переменных J (X) определена на множестве $U_n \subset E_n$.

В качестве примера J (X) может быть задано аналитически с помощью выражения вида:

$$J(X) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n).$$

1. Точка $x^* \in \boldsymbol{U_n}$, называется точкой *глобального минимума* функции J (X), если для всех $x^* \in \boldsymbol{U_n}$ выполняется неравенство $J(x^*) \le J(X)$. Значение $J(x^*) = \min\limits_{E_n} J(X) = J^*$ называется *минимумом* функции. Множество всех точек глобального минимума функции J (X) обозначают через $\boldsymbol{U}^*$.

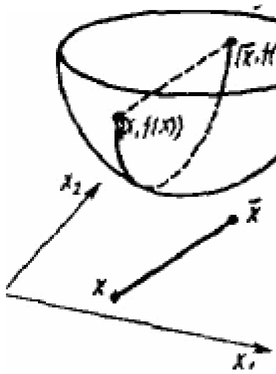
Замечание. Если $\boldsymbol{U}^* \neq 0$, то вместо минимума функции J (X) иногда рассматривают ее точную нижнюю грань $J^* = \inf\limits_{E_n} J(X)$, определение которой в *n*–мерном случае практически не отличается от определения для одномерной функции.

2. Точка $\tilde{X}$ называется точкой *локального минимума* функции J (X), если существует ε–окрестность точки $\tilde{x}$: $U_n(\tilde{x}) = \{X \mid \rho(X, \tilde{X}) < \varepsilon\}$ такая, что для всех $x^* \in U_n(\tilde{X})$ выполняется неравенство $J(\tilde{X}) \le J(X)$.

3. Если допустимое множество $\boldsymbol{U_n}$ в задаче минимизации (максимизации) функции *n* переменных совпадает со всем пространством $\boldsymbol{E}_n$, то говорят о задаче *безусловной оптимизации*

$$J(X) \to \min(\max), \ X \in \boldsymbol{E}_n.$$

В задачах минимизации особую роль играют выпуклые функции, так как локальный минимум выпуклой функции является глобальным минимумом, а строгая выпуклость обеспечивает еще единственность глобального минимума.



В общем случае рассматривается задача $J(X) \to \inf$,

предполагая, что функция $J(X)$ непрерывно дифференцируема на E_n. Согласно определению дифференцируемости функции

$$J(X+h) - J(X) = (J'(X), h) + o(h; X),$$

где $\lim\limits_{|h|\to 0} o(h;X)|h|^{-1} = 0$. Если $J^{'}(X)\neq 0$, то при достаточно малых $|h|$ приращение $J(X+h)-J(X)$ будет определяться дифференциалом функции $dJ(X)=(J'(X),h)$.

Здесь $J'(X)$- градиент вектор-функции $J(X)$, показывающий в n-мерном пространстве направление возрастания этой функции, определяемый как:

$$J'(X)=\begin{pmatrix} \dfrac{dJ(X)}{dx_1} \\ \dfrac{dJ(X)}{dx_2} \\ \ldots\ldots \\ \dfrac{dJ(X)}{dx_n} \end{pmatrix}.$$

Справедливо неравенство Коши-Буняковского

$$-|J'(X)|\cdot|h| \le (J'(X),h) \le |J'(X)|\cdot|h|,$$

причем если $J'(X)\neq 0$, то правое неравенство превращается в равенство только при $h=\alpha J'(X)$, а левое неравенство – при $h=-\alpha J'(X)$, где $\alpha = const \ge 0$. Отсюда ясно, что при $J'(X)\neq 0$ направление наискорейшего возрастания функции $J(X)$ в точке X совпадает с направлением градиента $J'(X)$, а направление наискорейшего убывания – с направлением антиградиента $(-J'(X))$.

Это замечательное свойство градиента лежит в основе ряда итерационных методов минимизации функций нескольких переменных.

Большинство методов, применяемых для решения задачи минимизации функций нескольких переменных, являются итерационными **методами спуска**, для которых каждая итерация (шаг) приводит к уменьшению значения целевой функции:

$$J(X^{k+1})-J(X^k)=J(X^k+h)-J(X^k)<0.$$

В численных методах индекс итерации размещается (чаще всего) сверху справа от обозначения итеративной точки, если она – векторная величина. Это позволяет сохранить обозначения компонент вектора с помощью нижнего индекса.

Все методы спуска работают по единой схеме:

1. Выбирается **начальное приближение** – некоторая точка X^0 и каким-то образом выясняется, что X^0 не есть точка минимума $J(X)$. Целесообразно использовать всю имеющуюся информацию о поведении целевой функции $J(X)$, чтобы выбрать X^0 поближе к точке минимума X^*.

2. Пусть приближение X^k к точке минимума $J(X)$ найдено. Выбирается вектор $h = h^k$, такой, что для всех достаточно малых $\alpha > 0$ справедливо неравенство $J(X^k + \alpha h^k) < J(X^k)$.

Вектор $h^k / \|h^k\|$, а иногда и сам вектор h^k называется **направлением спуска.**

3. Определяется величина шага спуска по направлению h^k, то есть положительное число $\alpha_k > 0$, для которого выполняется неравенство

$$J(X^k + \alpha_k h^k) < J(X^k).$$

4. За очередное приближение к точке минимума принимается

$$X^{k+1} = X^k + \alpha_k h^k.$$

5. Проверяется выполнение критерия окончания итераций (об этом смотри ниже). Если критерий выполняется, то итерации прекращают и полагают $X^* = X^{k+1}$. В противном случае возвращаются к п. 2.

Последовательность точек $X^0, X^1, X^2, \ldots, X^*$, генерируемая методом спуска, называется **траекторией спуска**.

Критерием прекращения процесса вычислений на k-м шаге часто выбирается близость градиента к нулю: $\|J'(X^k)\| < \varepsilon$.

Рекомендация. Для программной реализации вычисления значений координат $J'(X)$, - градиента вектор-функции $J(X)$, может быть использована следующая последовательность операторов:

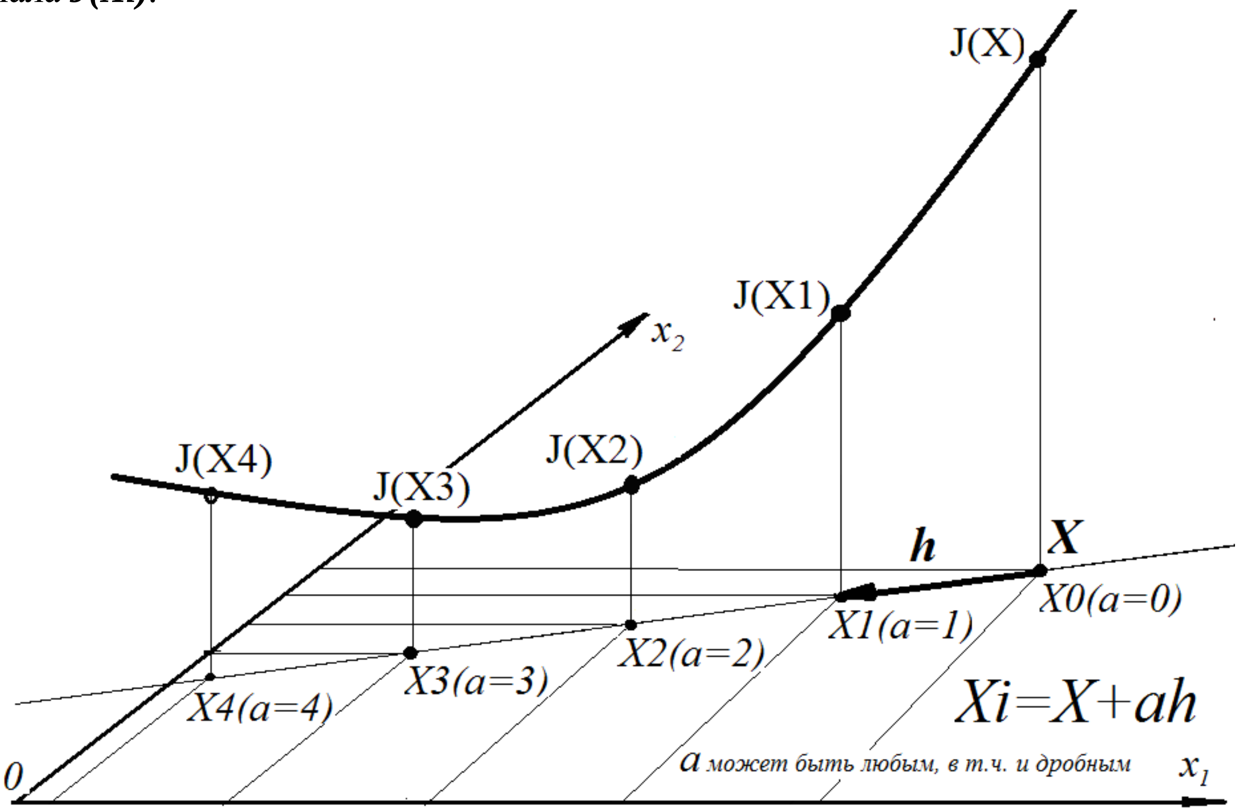
```
do i = 1, n
dx = 0.01*x(i)      // если x(i) ≠ 0 !!!!
d2x=dx*2
x(i) = x(i) + dx
f1 = J(X)
x(i) = x(i) - d2x
f2 = J(X)
pJ(i) = (f1 - f2) / d2x
x(i) = x(i) + dx
end du
```

При необходимости проводится дополнительное исследование поведений функции в окрестности точки X^k для выяснения того, достигается ли в точке X^k минимум функции $J(X)$ или не достигается. В частности, если $J(X)$ – выпуклая функция, то в стационарной точке всегда достигается минимум.

Помимо сравнения градиента с нулем используются также критерии: $\left\| X^{k+1} - X^k \right\| < \varepsilon_1$ и $\left\| J\left(X^{k+1}\right) - J\left(X^k\right) \right\| < \varepsilon_2$, где $\varepsilon, \varepsilon_1, \varepsilon_2$ - заданные положительные числа. Нередко используют различные сочетания этих трех критериев.

Рекомендация. Возможно участие исследователя в определении факта достижения минимума функционала *J(X).* Для этого в программе должна быть организована соответствующая печать с выводом всей необходимой информации.

Все методы спуска решения задачи безусловной минимизации различаются либо выбором направления спуска, либо способом движения вдоль направления спуска. На рис. ниже показано движение на плоскости параметров (частный случай) вдоль линии направления спуска ***h*** при разных значениях шага ***a*** и связь полученной точки ***Xi=X+ah*** со значением функционала ***J(Xi)***.



На рисунке показано движение вдоль линии (траектории) спуска при целых значениях ***a*** только для иллюстрации этого движения. Реально ***a*** может быть каким угодно: целым, дробным, положительным и отрицательным.

В любом случае на каждом шаге метода после определения направления спуска $h = h^k$ задача минимизации сводится к вычислению шага α_k в этом направлении, при котором

$$J(X^{k+1}) = J(X^k + \alpha_k h^k) \xrightarrow[\alpha_k]{} \min .$$

Иными словами, на каждом шаге минимизации функции нескольких переменных необходимо решать задачу минимизации функции одной переменной:

$$f1(\alpha) = J(X^k + \alpha h^k) \xrightarrow[\alpha]{} \min .$$

Проблемы решения данной задачи были рассмотрены в предыдущем пункте. Следовательно, для нахождения α_k необходимо написать функцию $f1(\alpha)$, где с учетом X^k и h^k вычислялись бы последовательно $X^{k+1} = X^k + \alpha h^k$ и $f1 = J\left(X^{k+1}\right)$. Значения X^k и h^k удобно передавать в функцию ***f1(a)*** через общую область (если используется язык ФОРТРАН), или сделать их глобальными переменными (в случае применения иных языков программирования).

Пример функции, подлежащей минимизации по параметру ***a,*** приведен ниже:

```
real function f1(a)
real J
common /x/ x(1)
common /h/ h(1)
common /x1/ x1(1)
common /n/ n
do i = 1, n
x1(i) = x(i) + a * h(i)
end do
f1 = J(x1)
end
```

Далее, воспользовавшись уже готовой функцией поиска минимума функции от одной переменной, определяется искомое

$$\alpha_k = \mathrm{arg\_min1}(f1, a, b) .$$

На этом этапе вычислительного процесса необходимо предварительно задаться отрезком $[a, b]$. В случае задачи безусловного поиска глобального минимума унимодальной функции величина его может быть произвольной. Но если область определения J (X) ограничена, либо решается задача отыскания локального минимума J (X), следует ввести ограничения и на величину шага α_k в направлении h^k. Рекомендуется выбирать значения границ отрезка $[a, b]$, которому должен принадлежать искомый α_k, с учетом выполнения следующего условия:

Любая компонента вектора $X^{k+1} = X^k + \alpha_k h^k$ должна отличаться от соответствующей компоненты вектора X^k не более чем в 2-3 раза.

Таким образом, перед обращением к функции $\mathrm{arg\_min1(f1, a, b)}$ необходимо для определения **a** и **b** решить соответствующую систему неравенств. Упомянутая система неравенств составляется и решается по каждой координате в отдельности. Из полученного множества отрезков следует выбрать самый меньший, вложенный отрезок, удовлетворяющий всем координатным неравенствам одновременно.

Имеет смысл организовать проверку начальных шагов процесса поиска минимума по нескольким переменным. Для этой цели следует выполнить сечение *J* (X) из точки X^k в направлении h^k, задавая различные значения α и вычисляя $f1(\alpha) = J(X^k + \alpha h^k)$. Фрагмент программы, позволяющей выполнить эту проверку, приведен ниже:

```
an = a
ak = b
da = (ak - an) / 50
do aa = an, ak, da
y1 = f1(aa)
печать в файл значений aa и y1
end do
```

По полученному множеству значений построить соответствующую кривую и сравнить ее минимум с результатом функции $\mathrm{arg\_min1(f1, a, b)}$.

В том случае, когда область определения $\boldsymbol{U_n}$ функции *J* (X) ограничена $(U_n \neq E_n)$, в функцию $f1(\alpha)$ необходимо ввести соответствующие корректирующие операторы (см. рис. ниже).

Откорректированное в $f1$ значение α возвратится (через параметр) в вызывающую подпрограмму (функцию).

В предложенном алгоритме коэффициент 0,99 может быть изменен с учетом требуемой точности отыскания минимума функции (чем точнее нужен результат, тем больше девяток будет в нем после запятой).

Полученную программу (как и для скалярного случая) следует протестировать на различных эталонных функциях от нескольких переменных. Имеет смысл проверить в тестовых примерах и ограниченность области определения $J(X)$, например, положительность (отрицательность) некоторых переменных.

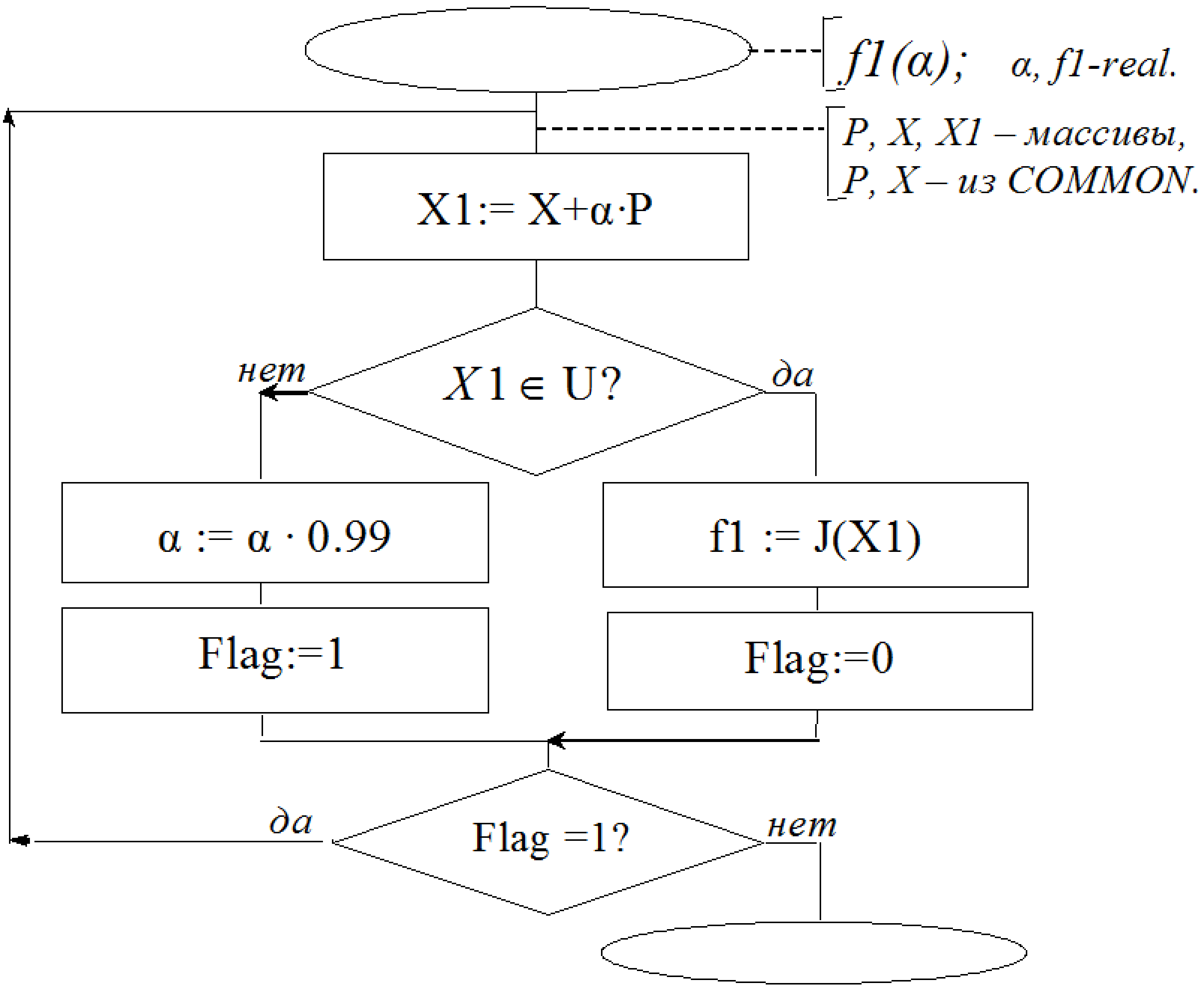


Рис. Программный способ учета ограничений области определения *J(X)*

5.3. Минимизация целевой функции задачи параметрической оптимизации

После успешного тестирования программы минимизации функции нескольких переменных можно приступить непосредственно к задаче параметрической оптимизации исходной системы. Для этого необходимо по реализованным в пунктах 5.1 и 5.2 алгоритмам выполнить минимизацию целевой функции, полученной в разделе 4.

Данная задача практически ничем не отличается от минимизации эталонных функций нескольких переменных. Следует только внимательно отнестись к использованию глобальных переменных (описанию общих областей COMMON), описанию (декларации) всех переменных и их инициализации.

В буферной функции $f1(\alpha)$ необходимо обеспечить неотрицательность варьируемых переменных системы (ограничить их положительной величиной, близкой к 0, например 0.000001) и без ошибок обратиться к минимизируемой целевой функции J (X) (проконтролировать список параметров в описании J (X) и вызове ее).

Как было отмечено в разделе 5.2, имеет смысл проверить начальные шаги процесса поиска минимума целевой функции, выполнив сечение J (X) из точки x^k в направлении h^k, задавая различные значения α и вычисляя $f1(\alpha) = J(x^k + \alpha h^k)$ (до обращения к функции $\mathrm{arg\_min1}(f1, a, b)$). Затем сравнить минимум полученной кривой с результатом выполнения $\mathrm{arg\_min1}(f1, a, b)$.

Рекомендуется на каждом шаге задачи параметрической оптимизации выводить на экран и записывать в файл значения: x^k, направление h^k, полученное α_k и значение J (X) в очередной точке $x^{k+1} = x^k + \alpha_k h^k$. Данная информация позволит проанализировать ход вычислительного процесса и проверить соответствие его результатов реализованным в программе методам поиска минимума. Кроме того, по полученным значениям можно построить траекторию движения к минимуму функционала в плоскостях варьируемых параметров с целью визуализации работы программы.

По окончании работы программы необходимо получить кривые переходного процесса исходной системы (см. разделы 3 и 4) с начальными и оптимальными значениями ее параметров. Анализ этих кривых позволит дать визуальное представление об улучшении качества системы после оптимизации.

Заключение

Параметрическая оптимизация динамического объекта – процесс, состоящий из множества этапов, на каждом из которых приходится решать определённые задачи.

С помощью системы дифференциальных уравнений описывается поведение системы во времени.

В результате решения системы дифференциальных уравнений можно получить количественную характеристику (функционал) поведения (качества) динамической системы.

Используя методы минимизации функций нескольких переменных, в совокупности с методами минимизаций функций одной переменной, можно решить задачу по уменьшению значения функционала системы.

Список используемой литературы

1. Васильев Ф.П. Численные методы решения экстремальных задач: Учебное пособие для вузов. – 2-е издание, переработанное и дополненное – М. : Наука. Главная редакция физико-математической литературы, 1988. – 552 с.
2. Васильев Ф. П. Методы оптимизации. – М.: Факториал Пресс, 2002. – 824 с.
3. Черноруцкий И.Г. Методы оптимизации в теории управления: Учебное пособие: доп. Мин. обр. РФ/ И. Г. Черноруцкий . – СПб.: Питер, 2004. – 256 с.
4. Корнеенко В. П. Методы оптимизации: Учебник / В. П. Корнеенко. – М.: Высш. шк., 2007. – 664 с.
5. Пантелеев А.В. Методы оптимизации в примерах и задачах: Учебное пособие: рек. УМО/ А. В. Пантелеев, Т. А. Летова. – 3-е изд., стер.. – М.: Высш. шк., 2008. – 544 с.
6. Измаилов А.Ф. Численные методы оптимизации : Учебное пособие: рек. УМС/ А. Ф. Измаилов, М. В. Солодов. – 2-е издание, переработанное и дополненное - М.: Физматлит, 2008. – 320 с.
7. Тарасик В.П. Математическое моделирование технических систем. Учебник для вузов. – Мн.: Дизайн-ПРО, 2004. – 640 с.
8. Аверченков В. И. Основы математического моделирования технических систем: Учебное пособие / В. И. Аверченков, В. П. Федоров, М. Л. Хейфец. – 2-е издание, стереотипное – М.: Флинта, 2011. – 271 с.
9. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. - 8-е изд. - М: БИНОМ. Лаборатория знаний, 2015. - 639 с.

Приложения

П1. Задание на выполнение курсовой работы по параметрической оптимизации системы автоматического управления

1. По заданной структурной схеме (см. Приложение П2) системы автоматического управления получить ее математическую модель в виде системы дифференциальных и алгебраических уравнений.
2. Построить программную модель системы на основе полученной математической модели с использованием стандартной программы интегрирования дифференциальных уравнений (RKF45) либо собственноручно написанной.
3. На траекториях движения системы реализовать вычисление интегральной оценки (функционала) качества этой системы.
4. Запрограммировать методы поиска минимума функций одной переменной и нескольких переменных.
5. Выполнить параметрическую оптимизацию заданного динамического объекта, используя запрограммированные методы минимизации функций одной и нескольких переменных (в совокупности): организовать автоматический поиск параметров системы исходя из достижения минимума вычисляемого функционала качества.
6. В плоскости варьируемых параметров построить траекторию движения к минимуму функционала.
7. Получить кривые переходного процесса системы при исходных и оптимальных параметрах.

Методы минимизации функции одной переменной

1. Метод деления отрезка пополам (метод дихотомии). Глава 1 §3.
2. Метод золотого сечения. Глава 1 §4.
3. Метод касательных. Глава 1 §9.
4. Метод парабол. Глава 1 §11
5. Метод равномерного поиска
6. Метод поразрядного приближения
7. Метод Фибоначчи
8. Метод троичного (тернарного) поиска
9. Метод квадрарного (четверичного поиска)

Методы минимизации функции нескольких переменных

1. Метод покоординатного спуска. Глава 5 §11.
2. Метод наискорейшего спуска. Глава 5 §1.
3. Метод сопряженных градиентов. Глава 5 §8.
4. Овражный метод. Глава 5 §1.
5. Метод Хука-Дживса.
6. Метод Дэвидона-Флетчера-Пауэлла.

Примечание. Даны ссылки на литературу**:** Васильев Ф.П. Численные методы решения экстремальных задач.

Информацию по другим методам найти самостоятельно.

Варианты заданий

Номер варианта	Функционал	Метод оптимизации 1	Метод оптимизации N	Система автоматического управления
1		1	6	1
2		2	5	2
3		3	4	3
4		4	3	4
5		5	2	5
6		6	1	6
7		7	6	7
8		8	5	8
9		9	4	9
10		1	3	10
11		2	2	1
12		3	1	2
13		4	6	3
14		5	5	4
15		6	4	5
16		7	3	6
17		8	2	7
18		9	1	8
19		1	3	9
20		2	6	10
21		3	5	1
22		4	4	2
23		5	3	3
24		6	2	4
25		7	1	5

Варьируемые параметры: `T1, T2, T3, T4, K1, K2, K3, K4, K5.`

Параметры системы автоматического управления (для схем П2)

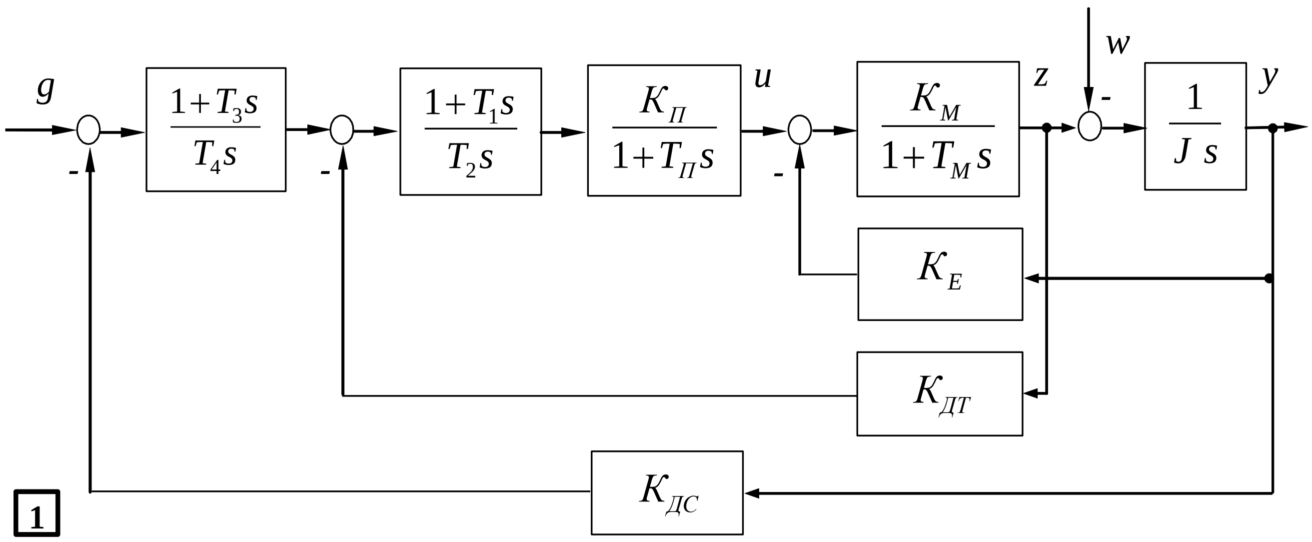
```
J = 2      Ko = 10     Кдт = 0.1   T'1 = 0.006     T'2 = 0.000018

Кп = 25    Ke = 1.2    Кдс = 0.1   Tдс = 0.001     Тм = 0.02

Км = 1.5   Ko1 = 1.5   Ko2 = 0.7   To1 = 0.025     To2 = 0.35    Tп = 0.003

  T1 = 0.025  T2 = 0.0225  T3 = 0.35  T4 = 0.0085

  K1 = 150    K2 = 40      K3 = 40     K4 = 1      K5 = 1
```

(значения последних коэффициентов можно уточнить в процессе моделирования)

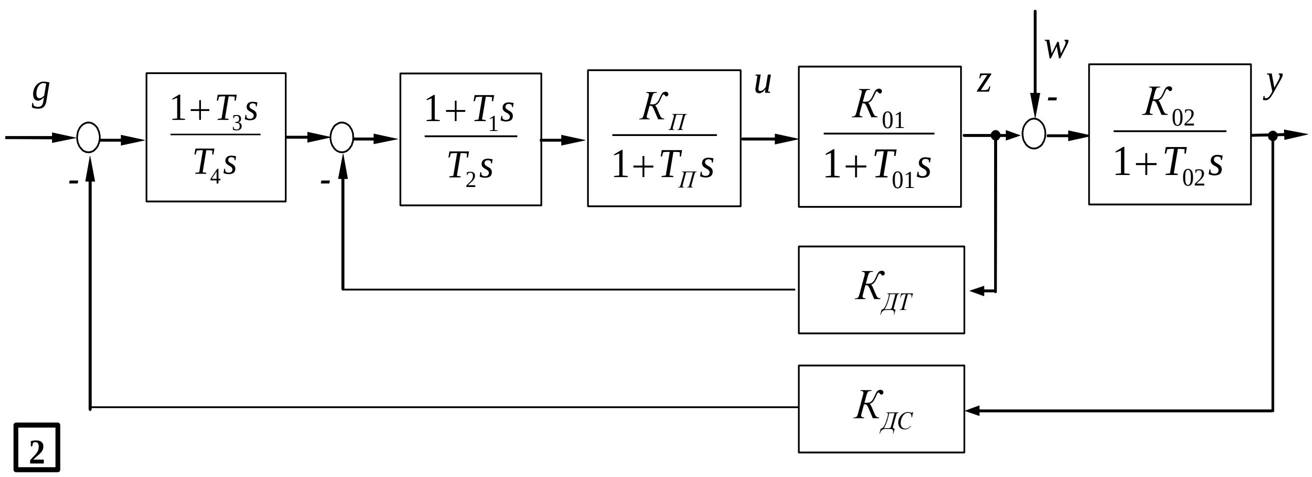
```
g(t) = 10*Sin(0.5*t)     W(t) = 2*Sin(50*t)
```

Начальные условия - нулевые,   период интегрирования T = 2*3.14

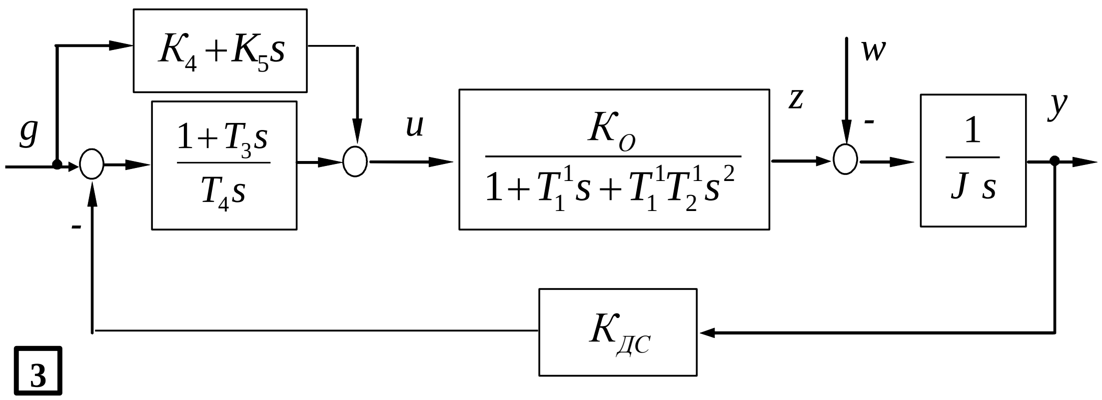
П2. Структурные схемы систем автоматического управления



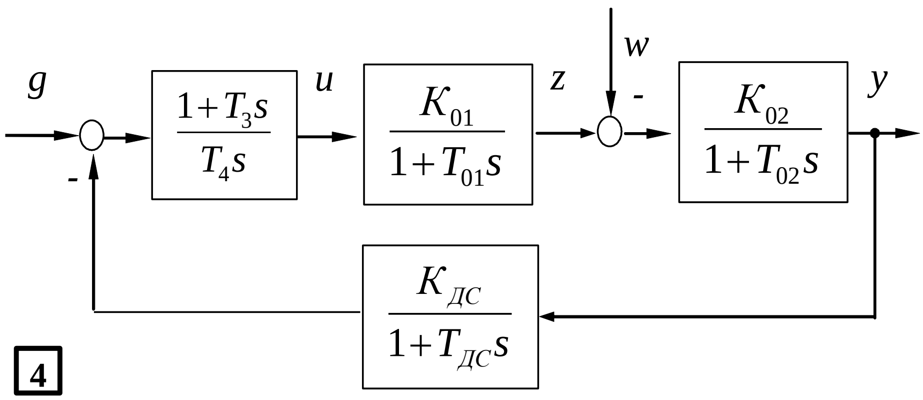
1



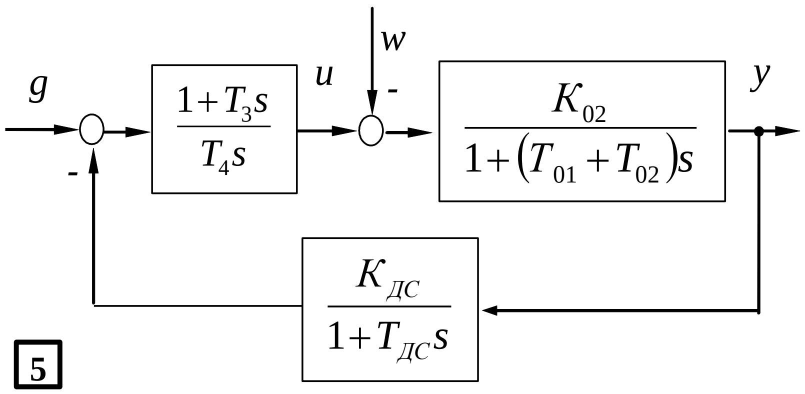
2



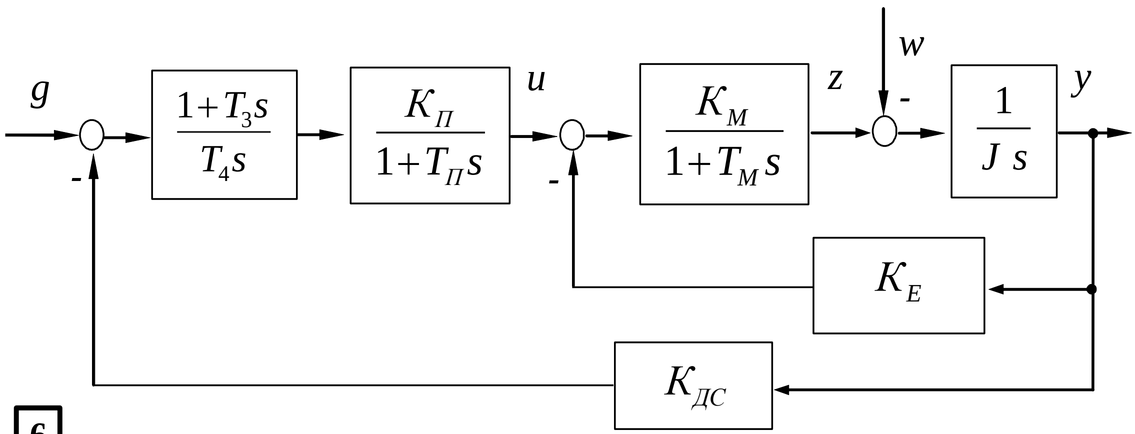
3


g
u
z
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{K_{01}}{1+T_{01}s}$
$\frac{K_{02}}{1+T_{02}s}$
$\frac{K_{ДС}}{1+T_{ДС}s}$

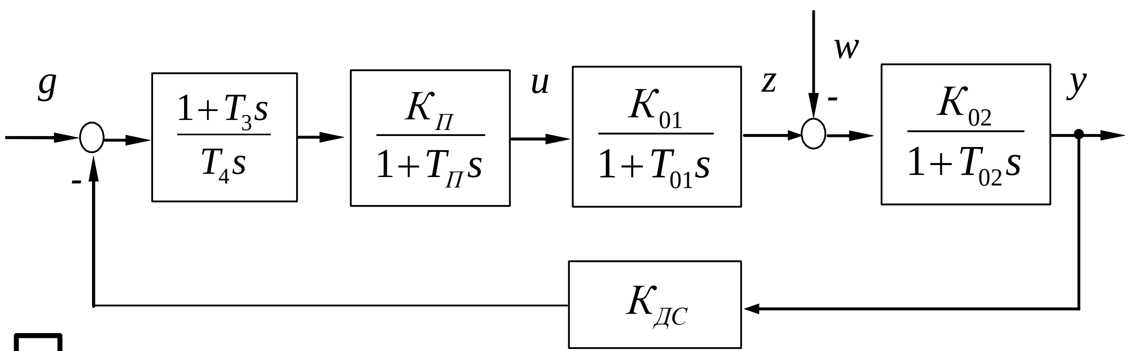
4


g
u
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{K_{02}}{1+(T_{01}+T_{02})s}$
$\frac{K_{ДС}}{1+T_{ДС}s}$

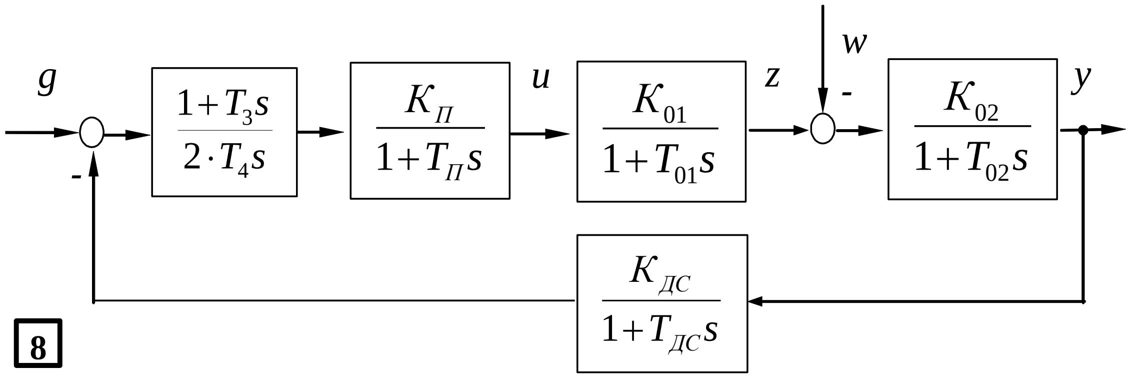
5


g
u
z
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{K_П}{1+T_П s}$
$\frac{K_М}{1+T_М s}$
$\frac{1}{J\ s}$
K_E
$K_{ДС}$

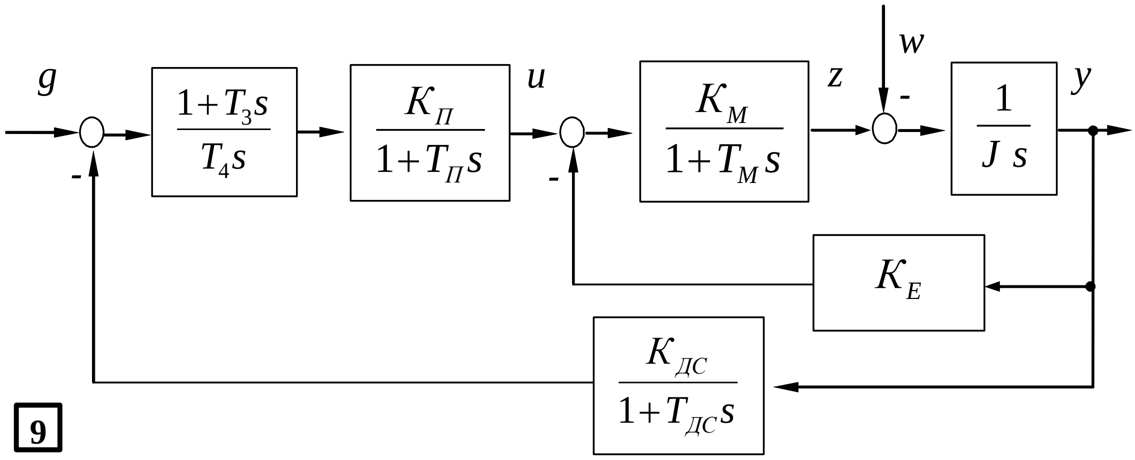
6


g
u
z
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{K_П}{1+T_П s}$
$\frac{K_{01}}{1+T_{01}s}$
$\frac{K_{02}}{1+T_{02}s}$
$K_{ДС}$

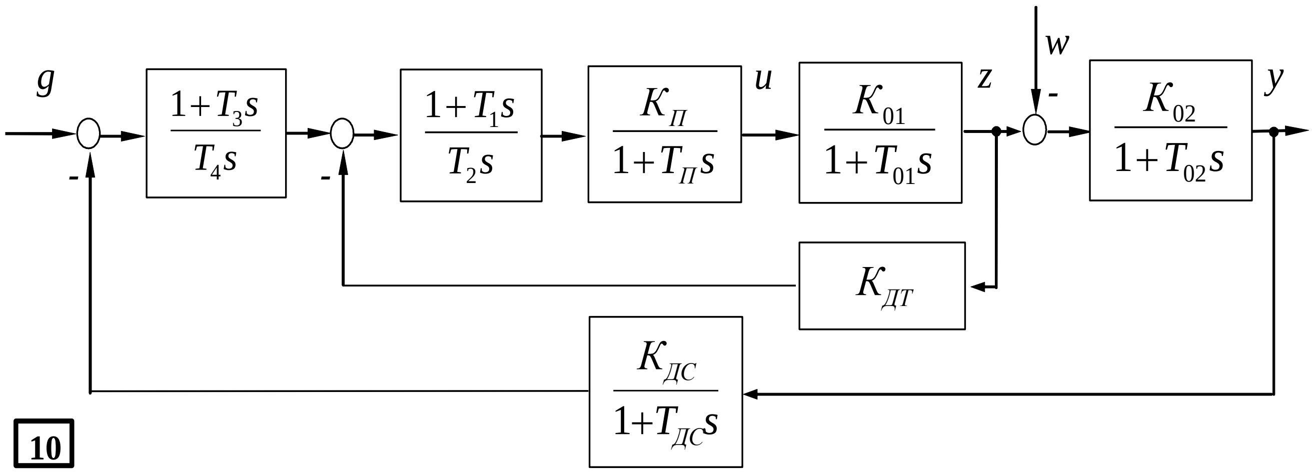
7


g
u
z
w
y
-
$\frac{1+T_3 s}{2 \cdot T_4 s}$
$\frac{K_П}{1+T_П s}$
$\frac{K_{01}}{1+T_{01} s}$
$\frac{K_{02}}{1+T_{02} s}$
$\frac{K_{ДС}}{1+T_{ДС} s}$

8


g
u
z
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{K_П}{1+T_П s}$
$\frac{K_М}{1+T_М s}$
$\frac{1}{J\ s}$
$K_Е$
$\frac{K_{ДС}}{1+T_{ДС} s}$

9


g
u
z
w
y
-
$\frac{1+T_3 s}{T_4 s}$
$\frac{1+T_1 s}{T_2 s}$
$\frac{K_П}{1+T_П s}$
$\frac{K_{01}}{1+T_{01} s}$
$\frac{K_{02}}{1+T_{02} s}$
$K_{ДТ}$
$\frac{K_{ДС}}{1+T_{ДС} s}$

10