«Компьютерное обеспечение инженерных задач»
к.т.н., доцент Красов А.В.

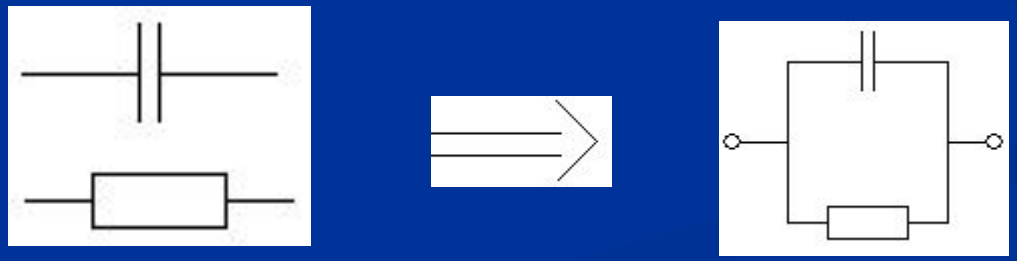
Лекция 2. Математические модели и моделирование. Формы представления моделей.

Основные понятия и определения Математические модели и моделирование

Система. Потребность в использовании понятия «система» возникала для объектов различной физической природы с древних времен: еще Аристотель обратил внимание на то, что целое (т. е. система - авт.) несводимо к сумме частей, его образующих. Единого определения понятия "система" пока не существует.

В различных источниках указываются различные определения системы.

В общем случае, под системой, понимается объект, свойства которого не сводятся без остатка к свойствам составляющих его дискретных элементов (неаддитивность свойств). Интегративное свойство системы обеспечивает ее целостность, качественно новое образование по сравнению с составляющими ее частями.



Элемент. Под элементом принято понимать простейшую неделимую часть системы. элемент - это предел деления системы с точек зрения решения конкретной задачи и поставленной цели. Систему можно расчленить на элементы различными способами в зависимости от формулировки цели и ее уточнения в процессе исследования.

Структура. Структура отражает наиболее существенные взаимоотношения между элементами и подсистемами, которые обеспечивают существование системы и ее основных свойств. Структура – это совокупность элементов и связей между ними. Структура может быть представлена графически, в виде теоретико-множественных описаний, матриц или графов.

Любой элемент системы можно рассматривать как самостоятельную систему (математическую модель, описывающую какой – либо функциональный блок, или аспект изучаемой проблемы), как правило, более низкого порядка. Каждый элемент системы описывается своей функцией. Под функцией понимается присущее живой и костной материи вещественно-энергетические и информационные отношения между входными и выходными процессами. Если такой элемент обладает внутренней структурой, то его называют подсистемой. Такое описание может быть использовано при реализации методов анализа и синтеза систем. Такое иерархическое представление нашло отражение в одном из принципов системного анализа - законе системности, говорящим о том, что любой элемент может быть либо подсистемой в некоторой системе либо, подсистемой среди множества объектов аналогичной категории. Элемент всегда является частью системы и вне ее не представляет смысла.

Термин "**подсистема**" подчеркивает, что выделанная часть системы обладает свойствами системы (в частности, свойством целостности), в отличии от простой группы элементов, для которой не сформулирована подцель и не выполняются свойства целостности.

В зависимости от характера расположения подсистем различают составной и иерархический характер построения модели.

Сложные системы представляют собой множество взаимосвязанных и взаимодействующих между собой подсистем, выполняющих самостоятельные общесистемные функции и цели управления [27], что предусматривает иерархический характер построения модели. Принципы построения иерархических моделей приведено на рис. 1.1. При этом имеет место не только составной, но и иерархический, рекуррентный, принцип построения модели. Свойства *L*-го уровня иерархии раскрываются через свойства подсистем L-1-го уровня и свойства их связей. Кроме принципа рекуррентного объяснения, при таком подходе достигается выполнение принципов неизбыточности и последовательного раскрытия неопределенностей [28]. За счет топологической (структурной) и операторной редукции на каждом уровне сохраняется и используется только необходимая информация.

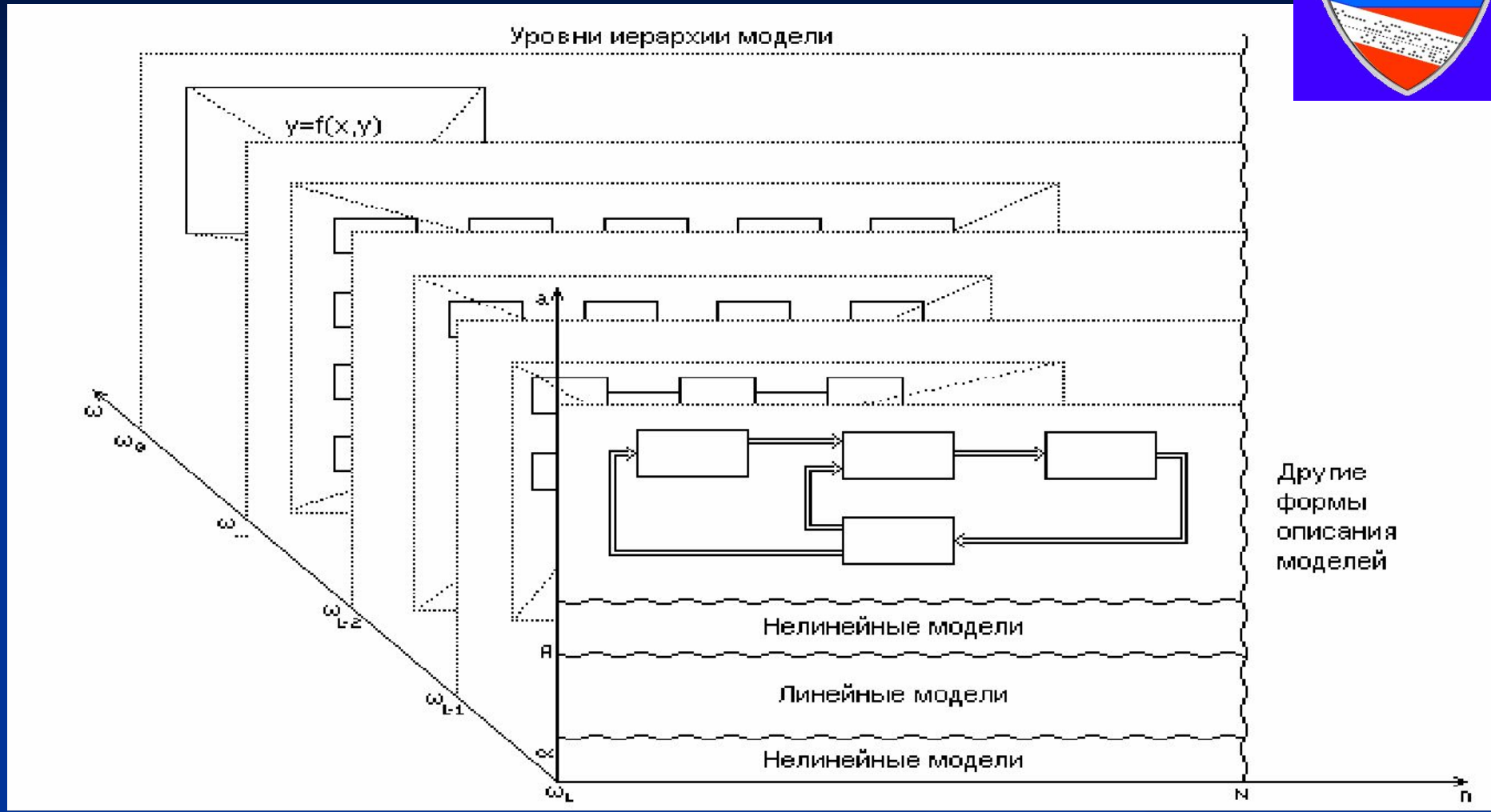
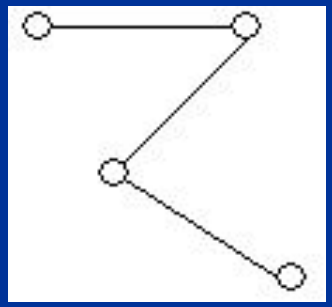


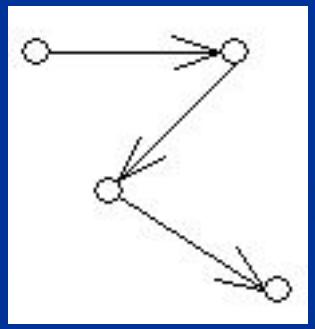
Рис 1.1. Принципы построения иерархических моделей

Связь. Главная причина неаддитивности систем – это наличие причинно–следственных связей элементов. Группа элементов – это не система; наличие связей между ними приводит к качественно новому образованию в системе.

Рассматривают направленные и не направленные связи, сильные и слабые, внешние и внутренние. Под понятием ненаправленных связей предполагают, что информационные потоки, ресурсы или сигналы передаются в любом направлении. Направленные связи указывают направление передачи сигналов. В зависимости от влияния связи на процессы в системе связи делят на сильные и слабые связи. В зависимости от расположения связей разделяют на внешние и внутренние связи. Внутренние связи обеспечивают обмен информации между элементами, расположенными внутри системе. Внешние связи обеспечивают обмен информацией или другими ресурсам между системой и внешними элементами (внешней средой на входе и выходе системы).

Обратная связь является основным элементом реализации регулирования, обеспечивающего устойчивость системы.





Состояние системы. Мгновенный срез значений всех переменных системы называется состоянием. Например, если система описывается векторами u_t, x_t и y_t, где x – входные сигналы, u - внутренние переменные, то. $z_t = \{u_t, x_t, y_t\}$ описывает состояние системы в момент времени .

Поведение. Переход системы из одного состояния в другое $z_{t+1} \to z_{t+2} \to \dots \to z_{t+n}$, для моментов времени $t+1, t+2, \dots, t+n$ ют поведением системы. Поведение системы можно представить как функцию времени $z(t) = f(t, u(t), z(t-1), z(t-2), \dots)$

Внешняя среда. Первым этапом начала исследования любого объекта является отделение объекта от внешней среды. Таким образом, под внешней средой понимается множество элементов, не включенных при описании в модель, но обменивающиеся с моделью потоками информации или других ресурсов. Иными словами, внешняя среда – это та часть исследуемой проблемы, которую мы вывели за рамки описания конкретной модели, но оказывающее на модель влияние. Это влияние или обмен информацией мы рассматриваем как внешние условия функционирования исследуемого объекта, не зависящие от наших действий. Обычно внешнюю среду разделяют на внешнюю среду на входе и выходе модели.

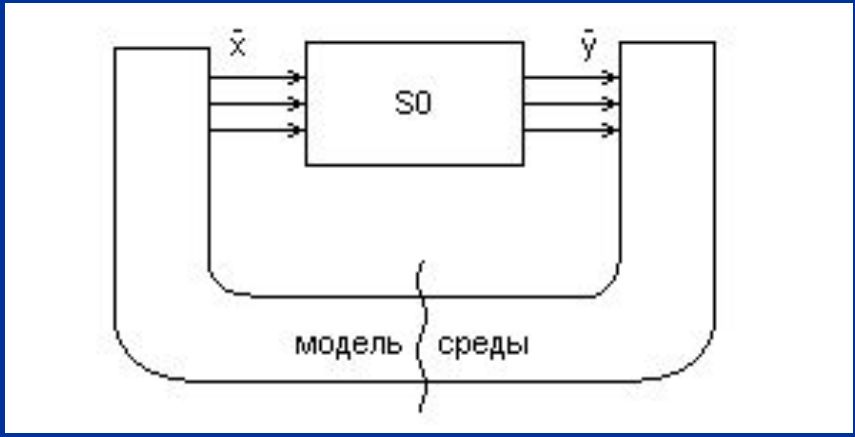


Рис. 1.2. Модель внешней среды на входе и выходе системы

Фактически модель внешней среды можно рассматривать как модель L+1-го уровня иерархии описания, раскрывающая ее взаимодействие с более вышестоящими уровнями. Взаимодействие модели с моделью внешней среды на входе и выходе показано на рис. 1.2.

Модель. Понятие модели трактуется неоднозначно. В основе его лежит сходство процессов, протекающих в реальной действительности и в заменяемой реальный объект, моделью. В философии, под моделью, понимается широкая категория кибернетики, заменяющая изучаемый объект его упрощенным представлением, с целью более глубокого познания оригинала. Под математической моделью (в дальнейшем просто моделью) понимается идеальное математическое отражение исследуемого объекта.

Модели могут использоваться для:

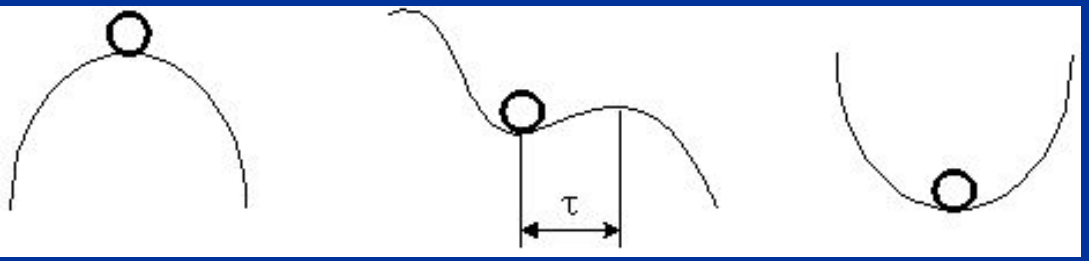
- объяснения поведения;
- предсказания;
- оптимизации.

Важными понятиями, отражающими поведение системы, являются: равновесие, устойчивость, развитие и цель системы.

Равновесие – способность системы, в отсутствии внешних воздействий сохранять свое состояние сколь угодно долго.

Устойчивость – способность системы возвращаться в состояние равновесия, после того, как она была выведена из него внешним воздействием.

Различают неустойчивое равновесное состояние, абсолютно устойчивое равновесное состояние, устойчивое в малом равновесное состояние. Иллюстрация этих понятий, на примере механической системы с шариком приведено на рис. 1.3.



а) б) в)

Рис. 1.3. Иллюстрация равновесных состояний: а – неустойчивого, б – устойчивого в малом, в – абсолютно устойчивое состояние

Рис. 1.3.а иллюстрирует случай неустойчивого равновесия. Малейшего отклонения достаточно для того, чтобы шарик покинул точку равновесия. Случай, представленный на рис. 1.3.б, показывает, когда при малых отклонениях шарик возвращается в исходную точку. Это так называемая устойчивость в малом. На рис. 1.3.в показан случай абсолютной устойчивости, когда при любых отклонениях шарик вернется в точку равновесия.

Цель. Понятие цель, обычно, определяли только для социальных, экономических или биологических систем, но усложнение кибернетических и технических систем потребовало введение в модель понятия целевой функции.

Основные уровни описания модели

На практике используют различные формы абстрактного описания систем, условно разделенные на высшие и низшие уровни описания систем:

Высшие уровни:

- символический, или, иначе, лингвистический;
- теоретико-множественный;
- абстрактно-алгебраический;
- топологический.

Низшие уровни:

- логико-математический;
- теоретико-информационный;
- динамический;
- эвристический.

С теоретико-множественного представления любая абстрактная система - совокупность упорядоченных троек $S=(X,H,Z)$, где $X=X1,..., Xn$ - множество элементов преобразования сигналов и подсистем, $H=H1,..., Hm$ - множество отношений (элементов связи), $Z=Z1,...., Zk$ - множество законов и операций [65, 67, 110]. Хотя данное представление является сильно упрощенным, но с учетом рассматриваемого класса моделей, достаточным для их представления. Более полное абстрактное описание систем в виде восьмерок можно найти в [97], а также [23, 59, 62], являющееся обобщением абстрактного описания конечного автомата [7, 76].

Динамическая управляемая система S определяется как:

$$S=(T, X, U, \Omega, Y, \Gamma, \phi, \eta),$$

где T - множество моментов времени; X - пространство состояний системы; U - множество значений входных (управляющих) воздействий; $\Omega=\{\omega:T\rightarrow U)$ - множество допустимых значений входных воздействий; Y - множество мгновенных значений выходных величин; $\Gamma=\{\gamma:T\rightarrow Y)$ - множество допустимых значений выходных величин; $\phi:T\times X\times U\rightarrow X$ - функция, определяющая состояние системы в момент времени t *по значениям описывающим систему в начальный момент времени* $(x0\in X)$, *и входных воздействий* $\omega\in\Omega$; $\eta:T\times X\rightarrow Y$ - функция, определяющая выходные отображения $y(t)=\eta(t,x(t))$ [55, 76].

1.6. Моделирование систем

Модель является средством для описания, понимания и предсказания известных и новых явлений и процессов. Отсюда следуют основные функции модели — *объяснительная* и *прогностическая*. Модель строится для отражения лишь части свойств исследуемого объекта и поэтому, как правило, *проще оригинала*. Говорят, что модель сходна с познаваемым объектом только по определённой совокупности признаков.

Классификация видов моделирования систем приведена на рис. 1.9.

Рис. 1.9. Классификация видов моделирования

Численное моделирование и вычислительные модели

Обычно различают следующие модели [76].

Фундаментальные (*детальные*) модели, количественно описывающих поведение или свойства системы, начиная с такого числа основных физических допущений (первичных принципов), какое только является возможным. Такие модели предельно подробны и точны для явлений, которые они описывают.

Феноменологические модели используются для качественного описания физических процессов, когда точные соотношения неизвестны, либо слишком сложны для применения. Такие приближенные или осредненные модели обычно обоснованы физически и содержат входные данные, полученные из эксперимента или более фундаментальных теорий. Феноменологическая модель основывается на качественном понимании физической ситуации. При получении феноменологических моделей используются общие принципы и условия сохранения.

При разработке алгоритмов численного интегрирования, лежащих в основе моделирования поведения СУ, учитываются следующие свойства.

1. *Консервативность,* которое представляет собой способность алгоритма удовлетворять физическим законам сохранения. Законы сохранения могут использоваться для построения конечно-разностных алгоритмов, в которых локально и глобально сохраняются некоторые суммы моделируемых физических величин. Если не требовать выполнения законов сохранения, то ошибки аппроксимации и округления могут неограниченно расти, приводя к непредсказуемому поведению даже весьма простых систем. Когда закон сохранения физической системы не “встроен” в алгоритм, то степень фактического выполнения в алгоритме условия сохранения определенной величины служит мерой его точности.

2. *Причинность* есть свойство алгоритма правильно отражать причинно-следственные отношения компонентов исследуемой физической системы.

3. *Положительность* — возможность воспроизведения алгоритмом строго неотрицательных процессов.

4. *Обратимость* алгоритма означает возможность реализации в консервативных системах свойства инвариантности процессов относительно преобразования вида $t \to -t$.

5. *Точность* обуславливается погрешностью вычислений на ЭВМ, численной сходимостью к решению и устойчивостью алгоритма. Считается, что алгоритм сходится, если при последовательном уменьшении шага Δt получается все более точный ответ. Алгоритм считается устойчивым, если небольшая ошибка на любой стадии расчета приводит к небольшой ошибке в решении.

Классы моделей

Модель объекта или системы управления принадлежит тому же классу что и описывающий их оператор преобразования. Разумеется, что можно говорить о классе только математической модели, а не реальной системы.

Таким образом, выделяют следующие признаки классов систем с непрерывным и дискретным временем:

- линейные (Л) или нелинейные (Д);
- стационарные (С) или нестационарные (С);
- детерминированные (Д) или стохастичные (Д);
- сосредоточенные (конечномерные) (К) или распределенные (бесконечномерные) (К).

Оператор линейный если он обладает свойствами однородности и аддитивности, т.е.:

$F(a*g(t)) = a*F(g(t))$ *и*

$F(g(t) + h(t)) = F(g(t))+F(h(t))$.

Свойство линейности называют также принципом суперпозиции. Принцип суперпозиции позволяет выразить реакцию линейной системы на любое воздействие через ее реакцию на элементарные воздействия.

Если принцип суперпозиции не выполняется оператор является нелинейным. Класс нелинейных операторов позволяет описывать системы значительно более широкого класса чем линейные операторы, но математические методы для работы с этим классом операторов значительно сложнее.

Простейший класс — ЛСДК — линейные стационарные детерминированные конечномерные системы. Они имеют форму обыкновенных линейных (дифференциальных (разностных) уравнений с постоянными ванными коэффициентами. Математика разработала весьма развитый аппарат анализа этого класса систем.

Более сложные классы операторов получаются при введении одного из |альтернативных признаков: ЛСДК; ЛСДК; ЛСДК; ЛСДК

Для таких систем существует незначительное число общих методов аналитического исследования; в основном, они разработаны только для частных случаев. Операторы второго уровня сложности получаются введением двух отрицаний:

ЛСДК; ЛСДК; ЛСДК; ЛСДК; ЛСДК; ЛСДК.

При трех отрицаниях получаем операторы третьего уровня сложности:

ЛСДК; ЛСДК; ЛСДК; ЛСДК.

Наконец, операторы четвертого уровня сложности:

ЛСДК

—нелинейные нестационарные стохастичные бесконечномерные. Им соответствуют нелинейные дифференциальные уравнения в частных производных с переменными случайными параметрами.

Для систем, описываемых операторами второго и выше уровней сложности, имеется, как правило, только единственная возможность их анализа и синтеза путем вычислительных экспериментов. Если модель системы образована элементами различных классов, то класс системы определяется классом элемента с максимальным числом отрицаний.

Кроме того выделяются такие понятия как: автономные системы, модели среды на входе и выходе системы и иерархические модели.

Автономные системы - системы которые не имеют входов и выходов (на них не влияет внешняя среда).

Модель среды - описание среды на входе и выходе. Учитывая иерархический характер построения модели, модель среды можно рассматривать как модель вышестоящей системы, в которую входит рассматриваемая нами в данный момент подсистема. Иллюстрация этого приведена на рис. 2.10.

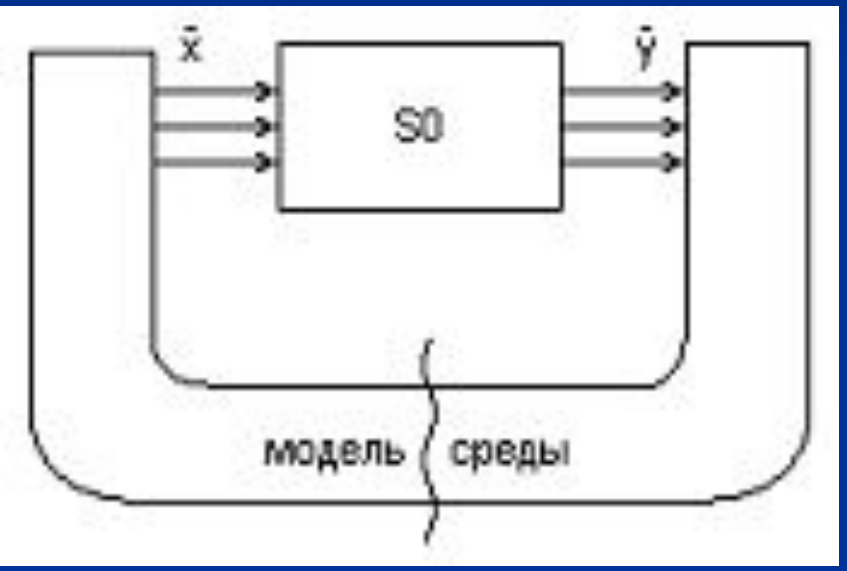


Рис. 2.10. Система L+1-го уровня причинно-следственного уровня иерархии - свернутая модель исследуемой системы с моделями среды на входах и выводах

Иерархические системы - представляют собой множество взаимосвязанных и взаимодействующих между собой подсистем управления, выполняющих самостоятельные общесистемные функции и цели управления [27]. При этом имеет место не только составной, но и иерархический, рекуррентный, принцип построения модели. Свойства *L*-го уровня иерархии раскрываются через свойства подсистем *L-1*-го уровня и свойства их связей. Кроме принципа рекуррентного объяснений, при таком подходе достигается выполнение принципов неизбыточности и последовательного раскрытия неопределенностей [28]. За счет топологической (структурной) и операторной редукции на каждом уровне сохраняется и используется только необходимая информация .

Ранги неопределенностей

Широко используемым понятием при описании моделей систем является понятие рангов неопределенности введенное А.А.Вавиловым.

К моделям нулевого ранга ($Ms(0)=<X>$) относится множество переменных, существенных для описания системы. Геометрически модель нулевого ранга это совокупность вершин, без информации о причинно следственных отношений между ними. Математически модели первого ранга неопределенности можно поставить в соответствие полный неориентированный граф, так как можно предположить наличие связей между всеми переменными.

Модель первого ранга неопределенности ($Ms(1)=<X,G>$) задает топологию системы. Бинарное множество G задает связи между переменными. Математически модели первого ранга неопределенности соответствует ориентированному графу. Для представления в ЭВМ моделей первого ранга неопределенности используются списочная форма или матрицы смежности, инциндентностей и др.

Модель второго ранга неопределенности ($Ms(2)$) или структурная модель содержит кроме топологии информацию о классе операторов.

Модель третьего ранга неопределенности ($Ms(3)$) содержит полную информацию о всех операторах - полная параметрическая модель.

Формы представления модели

Традиционными формами представления моделей являются системы уравнений в нормальной форме Коши и нелинейные дифференциальные уравнения, графы, структурные схемы. Они позволяют описывать не иерархические модели.

Нормальная форма Коши

Единообразное по форме и удобное для использования матричного аппарата математическое описание динамических (обычно “гладких”) систем достигается в пространстве состояний с использованием переменных состояния, т. е. уравнений в форме Коши

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t),$$
$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t), \qquad (1.1)$$

где $t \in \mathbf{R}^1, \mathbf{x} \in \mathbf{R}^n, \mathbf{u} \in \mathbf{R}^m, \mathbf{y} \in \mathbf{R}^r$ — векторы переменных состояния, управления и выходов; $\mathbf{R}^{(\cdot)}$ — (.)-мерное евклидово пространство; $\mathbf{f}: \mathbf{R}^n \to \mathbf{R}^n, \mathbf{h}: \mathbf{R}^n \to \mathbf{R}^r$ — гладкие отображения. Предполагается выполнение условия существования решений, а для большинства практических задач — их единственности. Условия существования и единственности решений выполняются, если u(t) принадлежит одному из следующих наиболее часто используемых классов функций: постоянные, кусочно-постоянные, кусочно-непрерывные, кусочно-гладкие, измеримые (локально-ограниченные), а функция f(t) — удовлетворяет условиям Коши-Липшица

К недостаткам данной формы представления необходимо отнести то, что в ней не сохраняется информации о топологии модели.

Системы нелинейных дифференциальных уравнений(СНДУ) различных порядков

СНДУ являются широко используемой формой представления нелинейных систем управления для численного исследования. В общем виде модель в форме СНДУ записывается следующим образом:

$$\varphi_i(x_1, \dot{x}_1, \ldots, x_1^{(q_i)}, \ldots, x_n, \dot{x}_n, \ldots, x_n^{(q_i)}, f_1, \dot{f}_1, \ldots, f_1^{(p_i)}, \ldots, f_m, \dot{f}_m, \ldots, f_m^{(p_i)}) = 0,$$

$$i = \overline{1 \; m}.$$

начальные условия:

$$x_1(0) = x_{1_0}, \ldots, x_{1_{(q_1)}}^{(q_1)} = x_{1_0}^{(q_1)},$$

$$\ldots\ldots$$

$$x_n(0) = x_{n_0}, \ldots, x_{n_{(q_1)}}^{(q_1)} = x_{n_0}^{(q_1)}$$

где: $f_j, f_j^{(p_j)}, \; j \in \overline{1,m}$ - внешние воздействия и их производные,

$x_i, \dot{x}_i, \ldots, x_i^{(q)}, \; i \in \overline{1,n}$ - внутренние переменные, включая выходные и их производные.

СНДУ более характерна пакетам программ, предполагающим значительные преобразования модели. В форме СНДУ можно представлять более широкий класс моделей чем в НФК.

Недостатком СНДУ представления является, так же как и в случае НФК, отсутствие полной информации о структуре модели, что затрудняет решение многих задач топологического характера. Решение этой проблемы возможно при упорядочивании порядка следования уравнений, так что в *i*-ом уравнении переменная *xi* являлась следствием. Такой подход встречается в ряде работ, например первые версии пакета NOCSYD [А2, А3].

Графы

Использование теории графов для описания моделей систем управления со сложной структурой, стало распространенным в последнее время. Теоретико-графовая форма описания модели позволяет эффективно использовать новые возможности языков

Представление в форме ориентированного (сигнального) графа, в частности структурной схемы, расширяет информацию о модели, по сравнению с НФК и СНДУ, позволяя вводить причинно-следственные отношения. Знание о направленности связей имеет большое значение для задач анализа и синтеза.

В качестве иллюстрации на рис 2.1. приведена диаграмма графа модели странного аттрактора Лоренца [93]. Эта форма представления позволяет эффективнее решать задачи выделения путей и ко нтуров, связности, структурной управляемости и многие другие, чем в форме НФК и отчасти СНДУ.

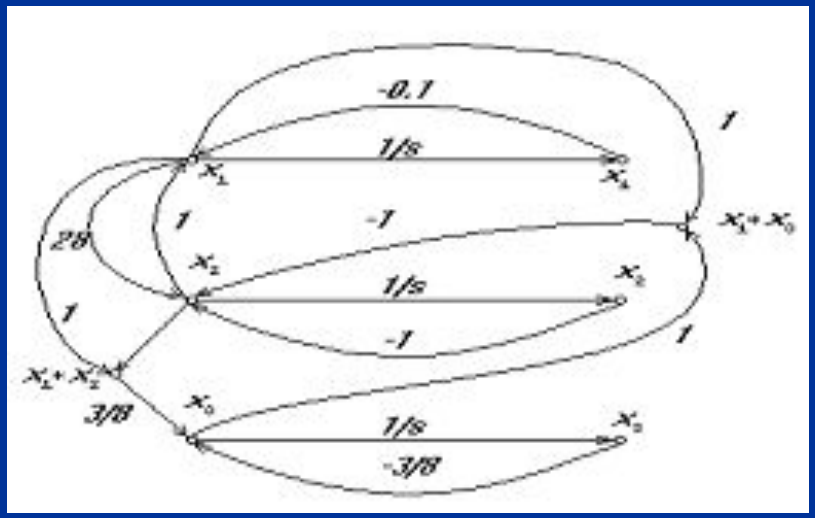


Рис. 2.1. Модель странного аттрактора в форме ориентированного графа

Модель системы представляется ориентированным графом $H=<G,H>$ с множеством переменных $X=x1, \ldots, xn$, N - общее множество вершин, и множеством дуг G - упорядоченных пар номеров смежных вершин (i,j), $G=(i,j)1, \ldots (i,j)n$. Общее количество таких пар обозначено в примерах как Q.

Несмотря на всю компактность и удобство такой записи, на практике чаще используют матрицу смежности R = rij, показывающую наличие дуги между i-ой и j-ой вершинами.

Другим способом представления топологии является матрица изоморфности **D**, в строках которой представлены номера входящих (с плюсом) и выходящих (с минусом) дуг.

Для приведенного на рис. 2.2 примера матрицы смежности и изоморфности имеют вид:

$$\mathbf{R} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}, \quad \mathbf{D} = \begin{vmatrix} +6 & -1 \\ +1,+7 & -2 \\ +2 & -3,-7 \\ +3 & -4 \\ +4 & -5 \\ +5 & -6 \end{vmatrix}.$$

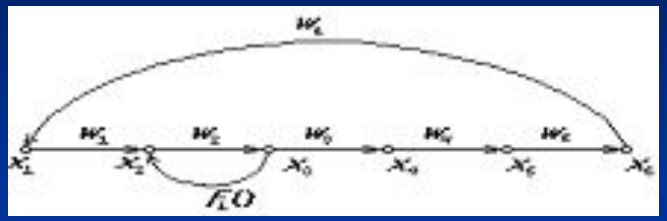


Рис. 2.2. Модель системы в форме графа

Избыточность хранимой информации в матрице смежности (нулевые значения) компенсируются простотой вычислительных алгоритмов и скоростью получения требуемой информации из матрицы. Кроме того, наличие только двух значений 0 или 1, дает возможность использовать для ее представления битовые поля, что дает значительную экономию памяти, и при размерах системы порядка 100 элементов не уступает по затратам ресурсов на хранение матрицы изоморфности, при значительно более простых алгоритмов обработки информации. Использование матриц смежности, инцидентностей, достижимостей и др. имеет большое применение для алгоритмов топологического анализа СС НСУ [107].

Ориентированные графы (структурные схемы) обычно широко используются при описании линейных систем и систем с одновходовыми нелинейностями. Однако возникают некоторые затруднения при описании нелинейных систем, где нелинейные функции могут зависеть от нескольких переменных, например при описании операций умножения и деления.

Гиперграфы

Гиперграф являются теоретико-множественной формой представления дифференциальных уравнений, заданных в общем случае непричинно—следственным способом [53, 54, 56, 73]. Гиперграф определяется как пара

$H = < X, E >$ образующая конечное множество $X=x1,...,xn$ вершин и некоторое семейством $E=e1,...,eq$ ребер - непустых частей X, удовлетворяющих условию $UE=X$ [67]. Одним из способов задания топологии гиперграфа [53], является $\mathbf{R}_{ij} = \| r_{ij} \|$ где

$$r_{ij} = \begin{cases} 1, & \text{если } x_i \in e_j, \\ 0, & \text{если } x_i \notin e_j. \end{cases}$$

Гиперграф является вариантом симплециального комплекса или симплециальной схемы. В ряде работ [75], вводится понятие ориентированного гиперграфа. При этом множество E - определяется как множество ориентированных ребер.

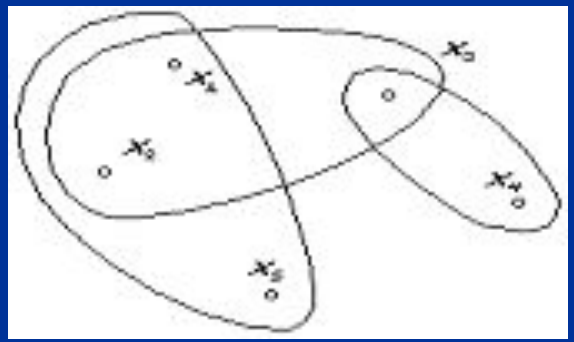


Рис. 2.3. Модель системы в форме гиперграфа

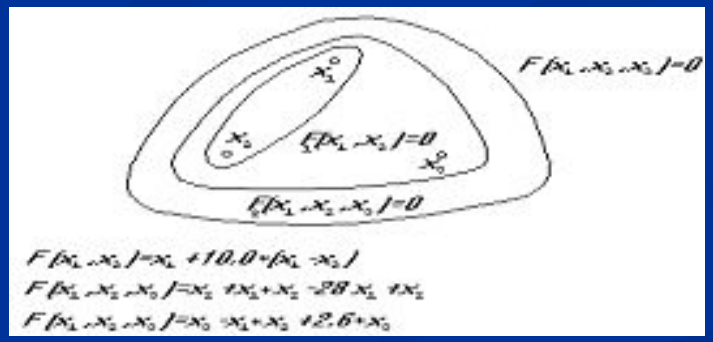


Рис. 2.4. Модель странного аттрактора в форме гиперграфа

Примеры гиперграфов приведены на рис. 2.3 и рис. 2.4. Гиперграф является способом группирования зависимых переменных, без указания причинно-следственных отношений между ними.

Нелинейные гибридные графы

Нелинейные гибридные графы, являются расширением множества обычных направленных графов и гиперграфов. Термин гибридный граф введен из-за объединения в вершинах операций суммирования и интегродифференциальных операций [118]. При переходе к нелинейным гибридным графам вершины по-прежнему являются базисными переменными с суммирующими и ининтегродифференцирующими свойствами. Функциональное преобразование записывается на ребрах графа, однако вид ребер зависит от их функции. В ребрах допускается функции нескольких аргументов, поступающих с базовых. Различные способы суммирования, нелинейные операции произведения в дугах, делают нелинейные гибридные графы удобным средством описания СС НСУ, полностью учитывающую их специфику, при этом графы Мэзо [53, 54] можно рассматривать подмножеством гибридных графов. Математически гибридный граф представляется как пара $H=<X,E>$, образованная конечным не пустым множеством вершин $X = Xs\ U\ Xd\ U\ Xi$, где Xs,Xd,Xi - множества, некоторые из которых могут быть пустыми, суммирующих, дифференцирующих и интегрирующих вершин. Кроме того вводится некоторое множество ребер $E = El\ U\ Enl$, где $El = (i1,j1),...(in,jn)$ - множество причинно-следственных пар задающих линейные отношения, а $Enl = (i1,Einp1),\ ....\ ,(in,Einpn)$ - множества задающие причинно-следственные функциональные отношения общего вида (нелинейные, нескольких переменных, операции умножения, деления и т.д., рис. 1.9), где $i..$ характеризует следствие, а не пустое множество $Einp...$ является номерами входных переменных.

Множество *X* можно рассматривать как модель нулевого ранга неопределенности [28], обозначаемый *Ms(0)*. Геометрически модель нулевого ранга неопределенности представляется набором несвязанных вершин и может рассматриваться как нуль-граф.

Обычно структурная схема модели составляется из типовых элементов. Веденная система содержит не только множество вершин *X* и связей *E*, но и значительные сведения о характере операторов их виде и свойствах. Отсутствуют только конкретные значения символических параметров и начальных значений. Описанную таким образом модель относят ко второму рангу неопределенности, обозначаемую *Ms(2)*.

Полностью определенную модель системы относят к третьему рангу неопределенности [28]. Она образуется добавлением к *Ms(2)* конкретных значений параметров задающих причинно- следственные отношения между входными и выходными переменными. В рассматриваемой в гл. 6 реализации - это символические записи функций, таблицы значений характеристик, коэффициенты уравнений и начальные значения интеграторов.

С учетом взаимосвязи внутренней формы представления, расчетных алгоритмов и рекурсивного характера построения модели и алгоритмов ранги неопределенности в значительной степени при реализации объединяются в одни структуры и скорее являются не этапами построения модели, а представляют собой иерархию их внутреннего представления и особенности организации программных средств для их ввода. Примером может служить организация пакета D2M [165]

Типовые конструкции нелинейных гибридных графов представлены на рис. 2.12.

Случай рис. 2.12.а показывает нелинейную функцию вида $y=f(x)$.

Вариант на рис. 2.12.б изображает операцию возведения в квадрат x^2, это является частным случаем часто встречающейся нелинейной операции произведения двух переменных $z=x*y$, показанным на рис. 2.12.д..

Линейные или линеаризованные операции изображены на рис. 2.12.в и рис. 2.12.е.

И, на конец, на рис. 2.12.г, представлена функция нескольких переменных, изображение которой на обычном графе было бы затруднительно.

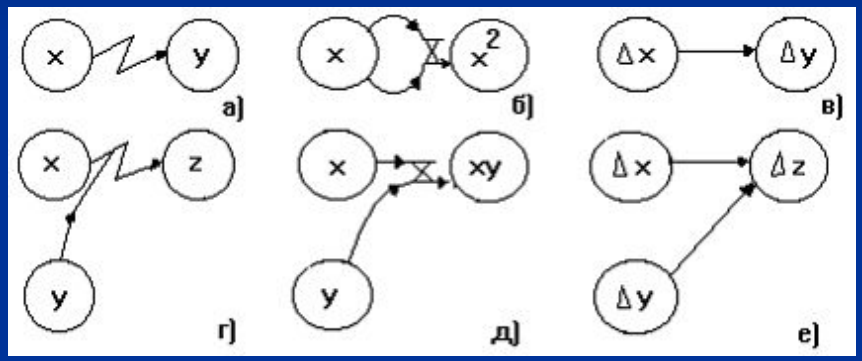


Рис. 2.12. Примеры нелинейных гибридных графов
а- с одним статическим нелинейным преобразованием общего вида; б- квадратичное преобразование; в- линеаризация функции одного элемента; г- функция двух переменных; д- произведение двух переменных; е- линеаризация функции двух переменных

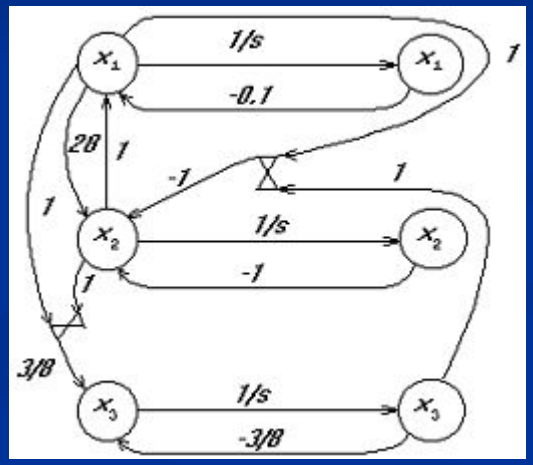


Рис. 2.13. Модель странного аттрактора в форме нелинейного гибридного графа

Рисунок 2.13 представляет нелинейный гибридный граф модели странного атрактора. Нелинейный гибридный граф это наиболее наглядная и простая, из числа рассмотренных, и понятная форма представления детерминированных моделей, сохраняющих полные сведения о топологии и о классе операторов.

Особенности представления чисел в ЭВМ и машинной арифметики

Особенности представления целых чисел

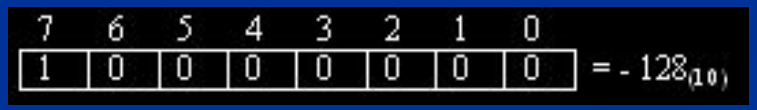
Для представления чисел выделяется фиксированное число двоичных разрядов, в зависимости от типа переменной.

Рассмотрим для примера тип данных char или byte. Для хранения числа выделяется один байт - восемь бит. Компьютер не позволяет работать с каждым битом в отдельности. Даже при использовании в ряде языков, так называемых логических переменных, имеющих значения только “Да” и “Нет”, для их хранения в памяти выделяется не 1 бит а 1 байт.

Итак для хранения коротких целых чисел выделяется 1 байт - 8 двоичных разрядов. Причем самый старший из них кодирует знак.

Например:

А число



Если при использовании коротких целых чисел к 127 прибавить 1, то получится -128. Таким образом ось целых чисел выглядит следующим образом:

При использовании типов int и long количество двоичных разрядов удваивается, соответственно увеличивается диапазон представления чисел.

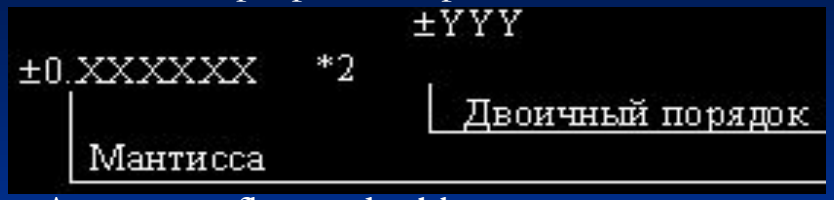
Тип	Кол-во разрядов	Диапазон чисел
int	16	-32638 .. 32637
long	32	-2147483648 .. 7483648

Но обязательно для всех целых типов данных в ЭВМ справедливо следующие:

- целые числа представляются в ЭВМ точно, но на ограниченном диапазоне;
- если к самому большому положительному числу добавить единицу, то получится самое большое по модулю отрицательное число.

Особенности представления вещественных чисел

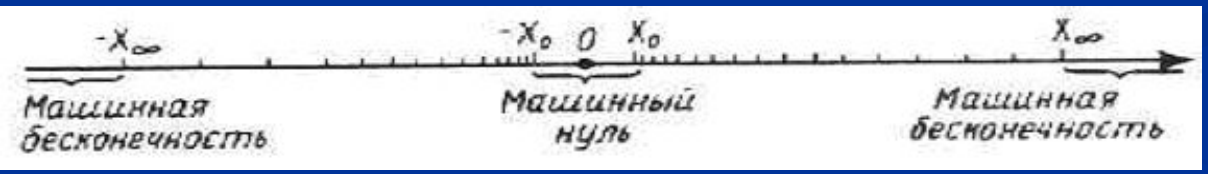
В отличии от целых чисел, вещественные числа помнятся приближенно, с точностью до последнего разряда. Формат записи вещественного числа:



Для типов float и double эти значения равны:

	Размер (байт)	Макс. знач.	Мин. отл. от 0	точность (дес. разрядов)
float	4	1e+73	1e-73	8
double	8	1e+310	1e-310	12

Первая часть вещественного числа - мантисса, определяет точность представления. Вторая часть - двоичный порядок, определяет максимальное значение числа.



На машинной числовой оси числа расположены не равномерно. Плотность их возрастает по мере приближения к нулю и падает с удалением от нуля.

Расстояние от одного числа на оси до другого ближайшего другого числа равно значению последнего разряда мантиссы. Но значение последнего разряда мантиссы определяется двоичным порядком числа, различным на протяжении числовой оси. Близко к машинному нулю, значение последнего разряда мантиссы будет порядка 1e-73, для вещественных чисел, а на краях числовой оси порядка 1e+73.

При выходе числа за пределы диапазона вещественных чисел происходит переполнение. При возникновении такой ситуации процессор генерирует соответствующие прерывание, обрабатывающие данную ситуацию.

Особенности машинной арифметики

При выполнении операций необходимо учитывать следующие специфику, в компьютере существует такое понятие как целочисленная арифметика. Если действия выполняются с целыми числами (или переменными), то и результат этих действий является целым числом.

По этому:

3 / 2 = 1, а ***3.0 / 2 = 1.5.***

Это является справедливым, в том числе и для результатов промежуточных операций. Например:

4.0 * 5 / 2 = ?

если сперва выполнится действие ***4.0 * 5***, то результат будет ***4.0 * 5 / 2 = 10***, а если в начале выполнится действие ***5 / 2***, то результат будет совсем иной ***4.0 * 5 / 2 = 8.0.*** Казалось бы естественным что действия выполняются с лева на право, но строчка записана на языке программирования высокого уровня и в каком порядке эти действия окажутся при трансляции в машинные команды не известно. Причем, в зависимости от оптимизатора, в один раз, они могут оказаться в одном порядке, а при другом проходе, в другом порядке.

В связи с выше изложенным, очень важную, дополнительную роль приобретают скобки. Кроме просто задания математических выражений скобки в программировании используются для задания приоритетов операций. Запись в виде ***(4.0 * 5) / 2*** снимает все вопросы о результате.

Кроме проблем с потерей информации при делении, возможна ситуация с превышением, в результате каких либо математических операций с целыми числами максимального значения допустимого для целого числа. Например:

20 * 10000 / 8 = ?

Если в начале будут выполнена операция ***10000 / 8***, то результат будет ***20 * 10000 / 8 = 25000***. Если в начале будет выполнена операция ***20 * 10000***, то в результате ее выполнения возникнет переполнение и общий результат операции неизвестен, даже если деление происходит на ***8.0*** (***20 * 10000 / 8.0***).

Кроме того если в программе результат этого выражения будет присваиваться вещественной переменной ***A***, например “ ***A=20 * 10000 / 8 ;*** ”, то это так же не будет влиять на результат, так как проблемы возникнут на этапе вычисления значения выражения, а не при выполнении операции присвоения значения.

Погрешности вычислений

Особенность представления вещественных чисел, рассмотренными в предыдущем параграфе, сильно влияют на процессы накопления ошибок при вычислениях.

Например, если Ваш компьютер работает с вещественными числами, мантисса которых имеет всего ***2*** десятичных разряда и складываете ряд чисел со значением ***0.01***.

В начале имеем

```
  0 . 0 0
+
  0 . 0 1
-------
  0 . 0 1
```

(не выделенным нулем отмече н разряд, который на самом деле не хранится. Число ***0.11*** помнится в компьютере в виде ***.11e0*** (с двумя десятичным разрядами в мантиссе).

В процессе суммирования получаем:

```
  1 . 0
+ 0 . 0 1
---------
  1 . 0
```

Дальнейшее добавление чисел меньших значения последнего разряда мантиссы не влияет на результат. Если в начале суммирования последний разряд мантиссы имел значение ***0.01***, то после того как число стало равно ***1.0***, последний разряд мантиссы увеличился на порядок и стал равен ***0.1***.

2.1. Математическая модель системы управления электроприводом

В процессе управления электроприводом [1] регулирующая координата должна наилучшим образом воспроизводить изменения предписанного значения. Однако при этом часто оказывается необходимым ограничить пределы изменения ряда переменных, например токов моторов и тп.

Система управления имеет два контура обратной связи. Второй контур управления начинает работать, в случае если контролируемая координата стремится превысить предельно допустимое значение. При значениях $u_{вых1} > U_{огр}$ замыкается вторая обратная связь, что накладывает дополнительные ограничения на выбор регулятора W_p таким образом что бы он успешно работал при действии как одной обратной связи, так и обоих связей одновременно.

Пример выполнения лабораторной работы

Причинно-следственная математическая модель системы управления электроприводом задается дифференциальными и алгебраическими уравнениями вида:

Уравнение выходного напряжения $$T_{o2}\frac{du_{вых}}{dt}=k_{o2}u_{вых1}$$

Уравнение первого выходного напряжения $$T_{o1}\frac{du_{вых1}}{dt}+u_{вых1}=k_{o1}u_2$$

Уравнение входного напряжения усилителя $$T_{\mu 1}\frac{du_2}{dt}+u_2=k_{п}e$$

Уравнение элемента сравнения. Где $u_{зад}$ – задающее напряжение.

$$e=u_{зад}-u_{oc1}-u_{oc2}$$

Уравнение диодного ограничителя расположенного во второй цепи обратной связи и начинающего работать при $u_{вых1}>U_{огр}$.

$$u_{oc1}=f(u_{вых1})$$

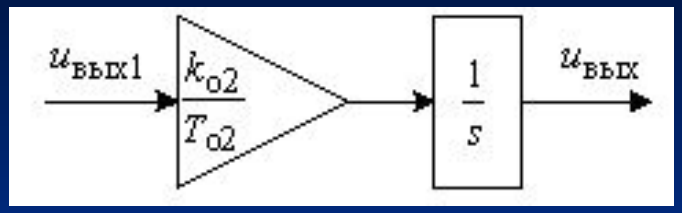
Основная обратная связь системы управления, обеспечивающая стабилизацию по заданному режиму.

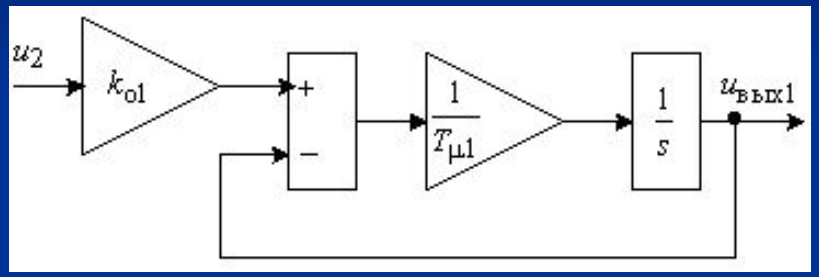
$$u_{oc2}=k_{oc}u_{вых}$$

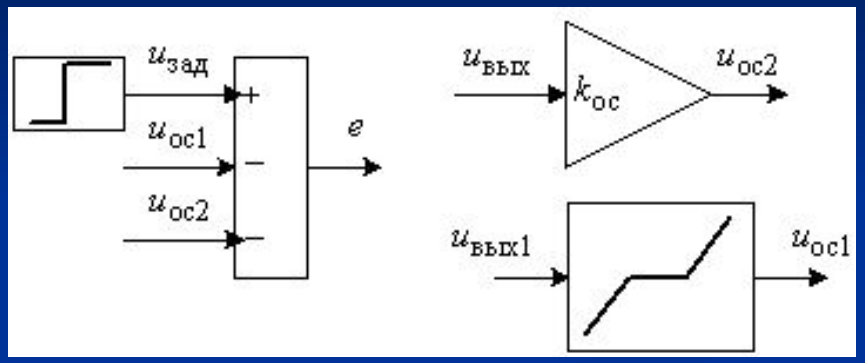
2.2. Составление структурной схемы модели

На основании уравнений, составляем структурные схемы узлов системы управления электроприводом. Для составления моделей линейных элементов рекомендуется воспользоваться эквивалентной схемой представления линейного элемента. Необходимость использования такого представления вызвано тем, что в используемом в курсовом проектировании пакете simulink имеется возможность задать начальные значения только для интегрирующего звена. Данная возможность понадобится для выполнения VI этапа курсового проектирования – расчета переходного процесса при переходе с номинального режима на заданный режим.

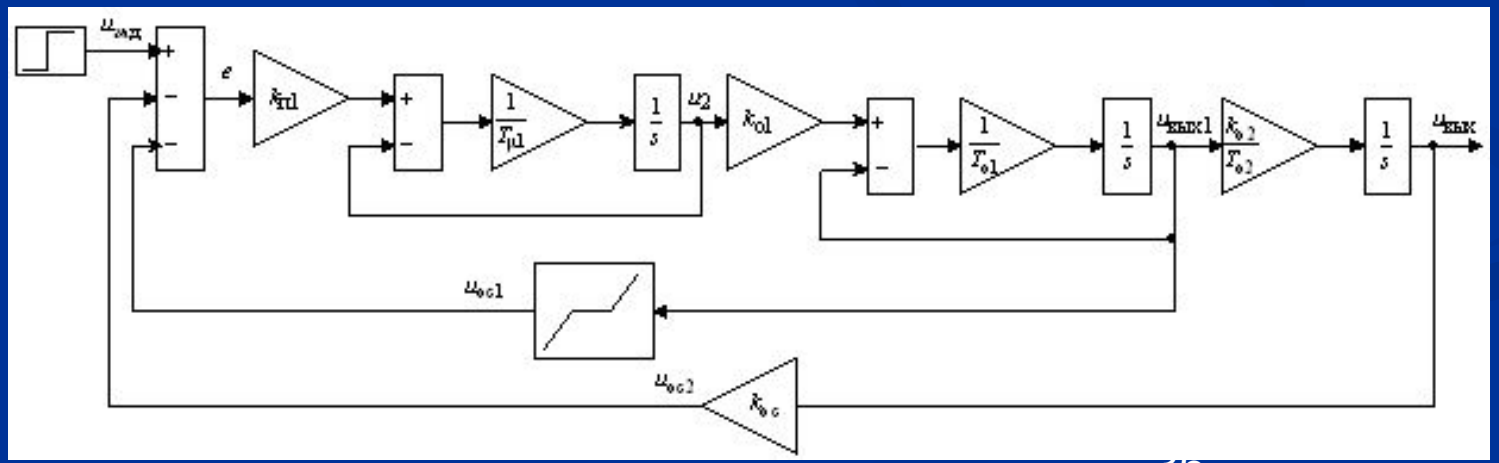
Структурные схемы отдельных фрагментов модели:



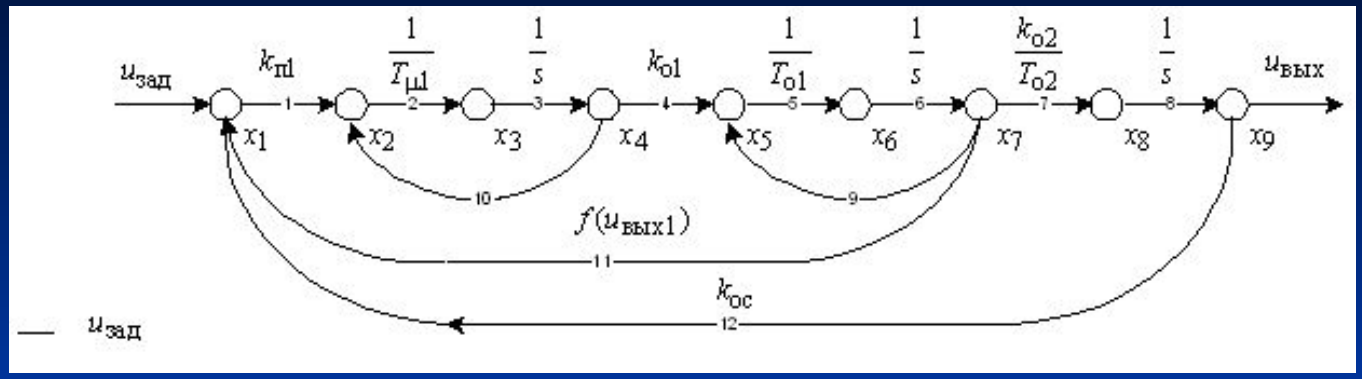




После объединения всех фрагментов получаем структурную схему модели системы управления электроприводом представленную, для выбранного примера



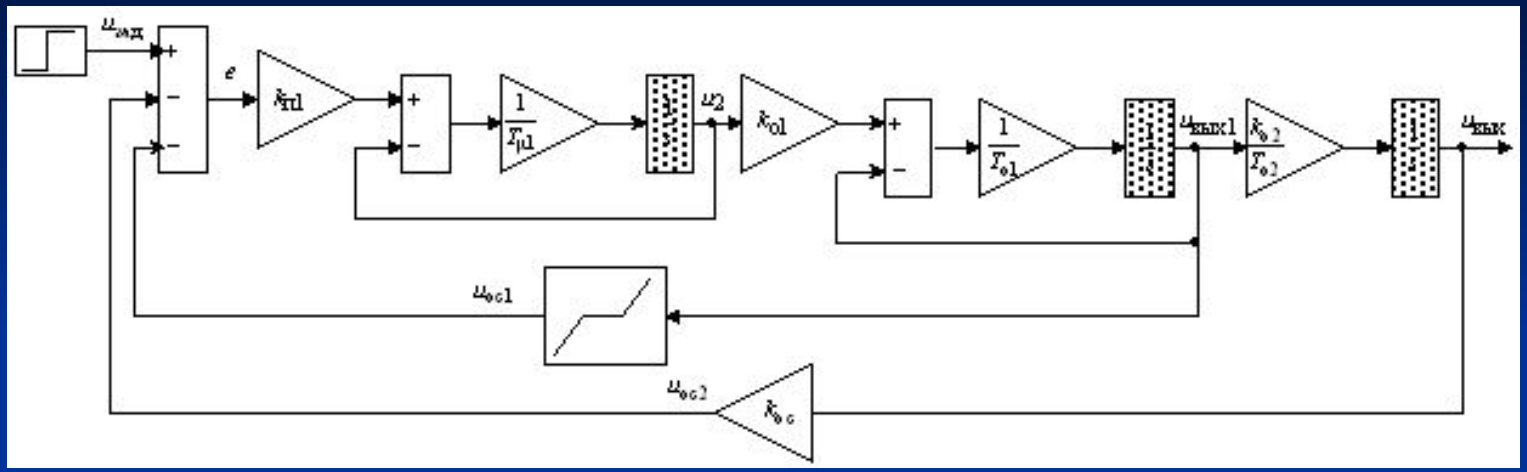
Запишем данную модель в форме графа



Матрицы смежности и изоморфности

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} -1 & +11 & +12 \\ +1 & +10 & -2 \\ +2 & -3 & \\ +3 & -4 & -10 \\ +4 & +9 & -5 \\ +5 & -6 & \\ +6 & -9 & -11 \quad -7 \\ +7 & -8 & \\ +8 & -12 & \end{bmatrix}$$

Представление модели в нормальной форме Коши



$$\frac{du_{\text{вых}}}{\text{dt}} = \frac{1}{T_{o2}} k_{o2} u_{\text{вых1}}$$

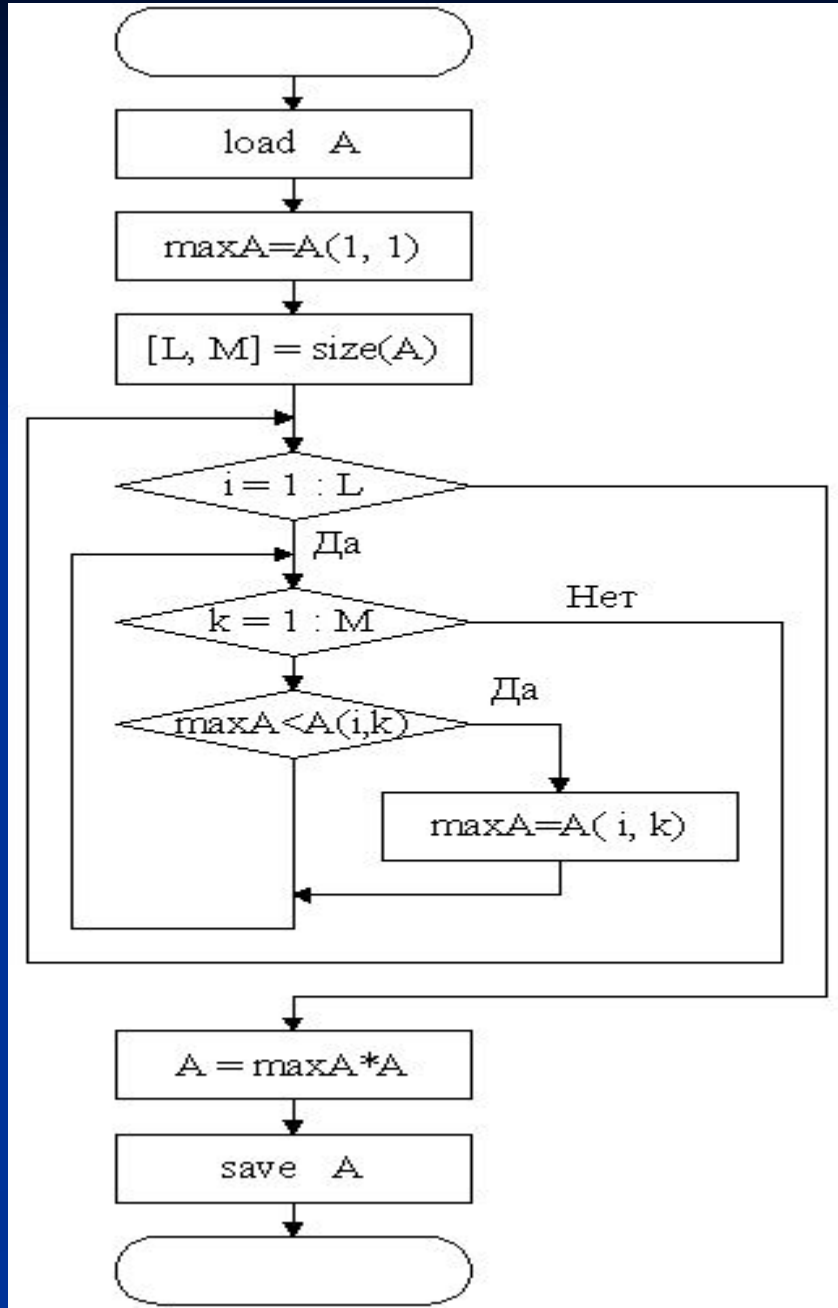
$$\frac{du_{\text{вых1}}}{\text{dt}} = \frac{1}{T_{o1}} \left(k_{o1} u_2 - u_{\text{вых1}}\right)$$

$$\frac{du_2}{dt} = \frac{1}{T_{\mu 1}} \left(k_n \left(u_{\text{зад}} - f\left(u_{\text{вых1}}\right) - k_{o2} u_{\text{вых}}\right) - u_2\right)$$

Программирование в среде MatLab

Задача:

Ввести матрицу из файла и умножить ее на максимальный элемент. Результат сохранить в виде файла.



Пример выполнения

```
load  A.dat % конкретный способ загрузки матрицы взять из задания
maxA = A(1, 1) ;
[L, M] = size(A) ; % определение размеров матрицы
for i = 1 : L
for k = 1 : M
      if  A(i, k) > maxA
            maxA = A(i, k) ;
      end
end
 end
 A = maxA * A ;
 save d:\stud\res.mat A % конкретный способ сохранения матрицы взять из
      % задания
```

Лабораторная работа