

Министерство образования Российской Федерации
ФГБОУ ВПО «Сибирская государственная автомобильно-дорожная
академия (СибАДИ)»

Кафедра автоматизации производственных процессов
и электротехники

**ИДЕНТИФИКАЦИЯ
ОБЪЕКТОВ АВТОМАТИЗАЦИИ
ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ
В СРЕДЕ MATLAB**

Методические указания к лабораторным работам

Составители: А-й А. Руппель, А-р А. Руппель

Омск

Издательство СибАДИ

1. Основные понятия и определения

Трудоемкость автоматизации технологических процессов во многом определяется степенью имеющейся информации о технологических объектах управления (ТОУ), их статических и динамических характеристиках. Обычно под объектом управления понимается часть окружающего нас мира, поведение которой нас интересует, и на которую мы можем целенаправленно воздействовать, то есть управлять ею.

Для облегчения работы с разнообразными объектами управления их разбивают на группы:

- статические объекты;
- динамические объекты;
- линейные объекты;
- нелинейные объекты;
- непрерывные объекты;
- дискретные объекты;
- стационарные объекты;
- нестационарные объекты;
- объекты с сосредоточенными параметрами;
- объекты с распределенными параметрами и т.д.

Определение характеристик ТОУ происходит по-разному: здесь рассматриваются методы, связанные с проведением специального экспериментального исследования ТОУ, в результате которого получается массив экспериментальных данных $[u_i, y_i]$, где u_i – входные переменные, y_i – выходные переменные ТОУ, i – номер опыта (всего может быть N опытов). Наиболее полная информация о ТОУ содержится в их математических моделях.

Под моделью обычно понимается выраженная в той или иной форме информация о наиболее существенных характеристиках ТОУ. По способу представления данной информации выделяют следующие типы моделей:

- словесные, или вербальные модели;
- физические модели (уменьшенные копии реальных объектов, иногда другой физической природы, позволяющие имитировать процессы в исследуемом объекте);
- математические модели (информация об исследуемом объекте или системе представляется в виде математических терминов).

В свою очередь математические модели делятся на:

- графические;
- табличные;
- алгоритмические;
- аналитические.

В частности, аналитические модели представляют собой отражение взаимосвязей между переменными объекта в виде математической форму-

лы или группы таких формул.

Моделирование основано на двух основополагающих признаках:

- на принципе практической ограниченности фундаментальных законов природы;
- на принципе подобия, означающем, что явления различной физической природы могут описываться одинаковыми математическими зависимостями.

Процедуру построения модели принято называть идентификацией, при этом данный термин относится к построению аналитических математических моделей динамических объектов.

Динамический объект — это объект, выход которого зависит не только от текущего значения входных сигналов, но и от их значений в предыдущие моменты времени. Идентифицируемый объект принято представлять в виде, показанном на рис. 1.1, где t — время; $u(t)$ — контролируемый (иногда управляемый) входной сигнал; $y'(t)$ — теоретический выход объекта; $y(t)$ — наблюдаемый выход объекта; $e(t)$ — случайная аддитивная помеха, отражающая действие неучитываемых факторов (шум наблюдения).

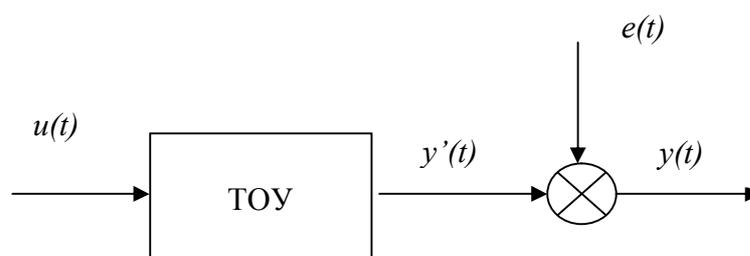


Рис. 1.1 Общее представление идентифицируемого объекта ТОУ

Обычно предполагают, что связь между входным и «теоретическим» выходным сигналами задается в виде некоторого оператора Ψ (оператор — правило преобразования какой-либо функции в другую функцию):

$$y'(t) = \Psi[u(t)], \quad (1.1)$$

при этом наблюдаемый выход объекта может быть описан соотношением:

$$y(t) = \Psi[u(t)] + e(t). \quad (1.2)$$

Принцип суперпозиции позволяет объединить все действующие помехи в одну общую $e(t)$ и приложить ее к выходу линейной модели. При рассмотрении задач идентификации все помехи считают статически независимыми.

Цель идентификации заключается в том, что на основании наблюдений

за входным $u(t)$ и выходным $y(t)$ сигналами на каком-то интервале времени определить вид оператора, связывающего входной и теоретический выходной сигналы.

Перед началом экспериментальных исследований проводят априорный анализ перечня входных переменных с целью отбора и включения в состав модели приоритетных (или лимитирующих), оказывающих наиболее сильное воздействие на выходные переменные $y(t)$. В первую очередь в их состав включают управляющие входные переменные, с помощью которых осуществляется регулирующее воздействие на ТОУ.

Понятие «оценка модели ТОУ» означает, что в процессе математической обработки массива данных при необходимости можно менять структуру модели, например вместо линейной модели можно использовать нелинейные различных типов.

Если структура модели не меняется, производится только оценка параметров модели.

Известны два принципиально различающихся подхода к получению оценки модели (1.1) или (1.2): экспериментально-статистический, когда модель представляют в виде формального уравнения (системы уравнений), связывающего входные и выходные переменные в определенном (обычно относительно небольшом) диапазоне изменения переменных, и аналитический, когда модель выводится из физических представлений о сути процессов в ТОУ.

Рассмотрим простейший сглаживающий фильтр (RC-цепь), показанный на рис. 1.2.

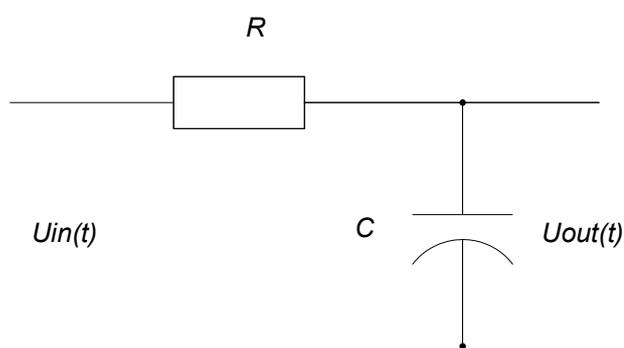


Рис. 1.2. Простейший сглаживающий фильтр

Исходя из известных законов электротехники, для него можно записать:

$$u(t) = RCdy(t)/dt + y(t), \quad (1.3)$$

где $U_{in}(t) = u(t)$, $U_{out}(t) = y(t)$.

Воспользоваться полученным соотношением (дифференциальным уравнением первого порядка) для определения выхода объекта при извест-

ном входном сигнале, однако, нельзя до тех пор, пока не установлены численные значения параметров R и C , входящих в модель. Более того, рассматривая другие примеры, можно прийти к выводу, что теоретический анализ с использованием известных физических законов, процессов и явлений, происходящих в объектах, дает возможность установить только структуру модели с точностью до ряда неизвестных параметров. Если такая структура (с точностью до вектора коэффициентов β) известна, то при известном входном сигнале $u(t)$ описание объекта можно представить в виде:

$$y(t) = F(\beta, t) + e(t), \quad (1.4)$$

где F – функция известного вида, зависящая от β и времени t .

Последнее уравнение позволяет после проведения эксперимента, заключающегося в фиксации входного и выходного сигналов на каком-то интервале времени, провести обработку экспериментальных данных и каким-либо методом (например, методом наименьших квадратов) найти оценку вектора параметров β . Отметим, что при экспериментальном определении параметров модели необходимо обеспечить:

- подбор адекватной структуры модели;
- выбор такого входного сигнала, чтобы по результатам эксперимента можно было найти оценки всех параметров модели.

Наиболее просто задача определения параметров решается для линейных объектов, для которых выполняется принцип суперпозиции. В задачах идентификации под линейными объектами чаще понимаются объекты, линейные по входному воздействию.

С учетом изложенного можно уточнить понятие идентификации.

Под идентификацией динамических объектов понимают процедуру определения структуры и параметров их математических моделей, которые при одинаковом входном сигнале объекта и модели обеспечивают близость выхода модели к выходу объекта при наличии какого-то критерия качества.

Обычно идентификация – многоэтапная процедура. Основные ее этапы следующие:

1. Структурная идентификация заключается в определении структуры математической модели на основании теоретических соображений.
2. Параметрическая идентификация включает в себя проведение идентифицирующего эксперимента и определение оценок параметров модели по экспериментальным данным.
3. Проверка адекватности – проверка качества модели в смысле выбранного критерия близости выходов модели и объекта.

Отметим, что в связи с многообразием объектов и различных подходов к их моделированию существует множество вариантов задачи параметрической идентификации, классификация которых показана на рис. 1.3.

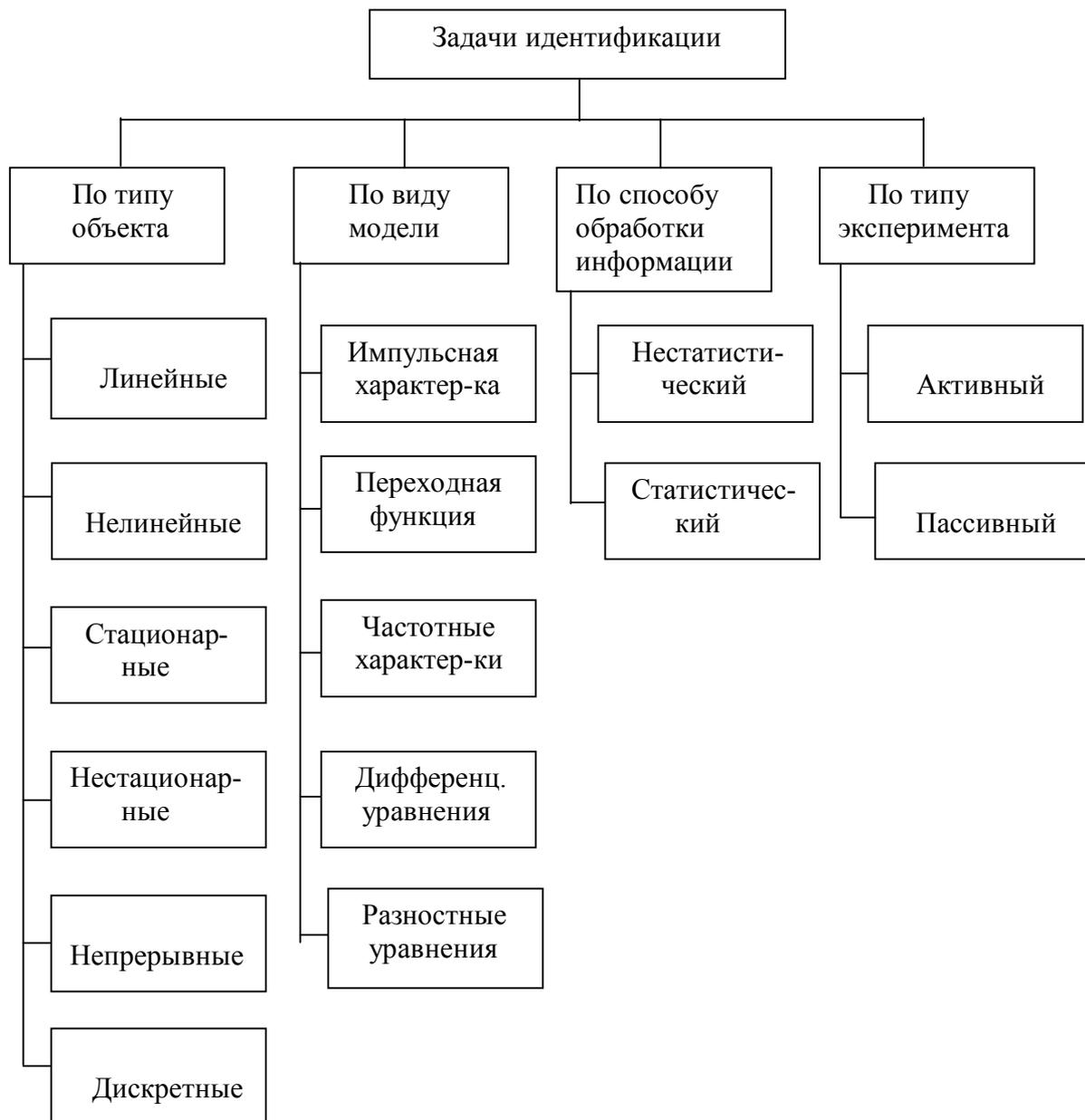


Рис. 1.3. Классификация задач идентификации

2. Виды моделей объектов

Рассмотрим основные виды моделей линейных непрерывных стационарных динамических объектов и их взаимосвязь (действием шума $e(t)$ пока пренебрегаем).

1. *Дифференциальное уравнение.* Наиболее универсальная модель, имеющая форму

$$\sum_{i=0}^{na} a_i y^{(i)}(t) = \sum_{j=0}^{nb} b_j u^{(j)}(t), \quad (2.1)$$

где na – порядок модели ($na > nb$), a_i и b_j – постоянные коэффициенты (параметры модели), $u^{(i)}(t)$ и $y^{(i)}(t)$ – производные, соответственно, входного и выходного сигналов.

2. *Передаточная функция.* Данная характеристика определяется как отношение преобразования Лапласа выходного и входного сигналов, что с учетом свойств данного преобразования и вышеприведенной формулы дает

$$W(p) = \frac{L\{y(t)\}}{L\{u(t)\}} = \frac{Y(p)}{U(p)} = \frac{\sum_{j=0}^{nb} b_j p^j}{\sum_{i=0}^{na} a_i p^i}, \quad (2.2)$$

где $L\{\bullet\}$ – символ преобразования Лапласа, p – комплексная переменная (оператор Лапласа).

3. *Импульсная характеристика (ИХ) $w(t)$.* Под ИХ понимается реакция предварительно возмущенного объекта (то есть объекта с нулевыми начальными условиями) на входной сигнал в виде δ -функции.

4. *Переходная функция $h(t)$.* Это реакция предварительно невозмущенного объекта на входной сигнал в виде единичного скачка. Из теории автоматического управления известны следующие соотношения между этими характеристиками:

$$L\{w(t)\} = W(p), \quad w(t) = h'(t), \quad L\{h(t)\} = W(p)/p \quad (2.3)$$

При нулевых начальных условиях связь между выходными и входными сигналами описывается интегралом свертки:

$$y(t) = \int_{-\infty}^{\infty} \omega(t - \tau) u(\tau) d\tau, \quad (2.4)$$

или в операторной форме:

$$Y(p) = W(p) * U(p). \quad (2.5)$$

5. *Частотные характеристики.* Частотные характеристики объекта определяются его комплексным коэффициентом передачи при подстановки вместо оператора Лапласа p комплексного аргумента $j\omega$, который является Фурье-преобразованием ИХ.

Модуль комплексного коэффициента передачи $|W(j\omega)| = A(\omega)$ представляет собой, как известно, амплитудно-частотную характеристику (АЧХ) объекта с передаточной функцией $W(p)$, а аргумент $\arg(W(j\omega)) = \varphi(\omega)$ – фазочастотную характеристику (ФЧХ).

Графическое представление $W(j\omega)$, на комплексной плоскости при изменении ω от 0 до ∞ , то есть график амплитудно-фазовой характеристики

(АФХ) в полярных координатах в отечественной литературе называется годографом, а в англоязычной – диаграммой Найквиста.

В теории управления часто используется логарифмическая амплитудно-частотная характеристика (ЛАЧХ), равная $20 \lg |W(j\omega)|$.

6. *Модель для переменных состояния.* В 70-е годы Г. Розенброком был создан метод «размытых» частотных характеристик, предназначенный для автоматизированного проектирования систем с несколькими входами и выходами, ориентированный на использовании средств вычислительной техники и названный методом переменных состояния (МПС).

В основе этого метода лежит представление дифференциальных уравнений в нормальной форме Коши, которое дополняется алгебраическими уравнениями, связывающими выходные переменные с переменными состояния:

$$\begin{aligned}x' &= Ax + Bu \\ y &= Cx,\end{aligned}\tag{2.6}$$

где u – вектор входных воздействий; y – вектор выходных воздействий; x – вектор переменных состояния; A , B , C – матрицы коэффициентов размерности $n \times n$, $n \times m$, $r \times n$ соответственно; n – число переменных состояния или максимальная степень производной исходного дифференциального уравнения; m – число входов; r – число выходов.

Математическим аппаратом метода переменных состояния являются матричное исчисление и вычислительные методы линейной алгебры. Метод переменных состояния содействовал значительному развитию теории управления. На языке МПС выполнена большая часть работ по оптимальному управлению, фильтрации, оцениванию.

Для систем с одним входом и одним выходом уравнения переменных состояния можно сформулировать следующим образом. При выборе n координат системы (объекта) в качестве переменных ее состояния (такими координатами, например, могут быть выходной сигнал $y(t)$ и $n-1$ его производных) принимаем x_i , $i = 1, 2, \dots, n$ и данную систему можно описать следующими уравнениями для переменных состояния:

$$\begin{aligned}x'(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t),\end{aligned}\tag{2.7}$$

где $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ – вектор-столбец переменных состояния; A , B , C , и D при скалярных $u(t)$ и $y(t)$ – соответственно матрица размера $n \times n$, векторы размера $n \times 1$ и $1 \times n$ и скаляр (при векторных $u(t)$ и $y(t)$ – матрицы соответствующих размеров).

Применение при нулевых начальных условиях, к последним уравнениям преобразования Лапласа позволяет получить следующее выражение для передаточной функции:

$$W(p) = C(pI - A)^{-1} B + D, \quad (2.8)$$

где I – единичная матрица.

В качестве примера рассмотрим преобразование исходного дифференциального уравнения :

$$\frac{d^2 y}{dt^2} + 2 \frac{dy}{dt} + 5y = 3u, \quad (2.9)$$

или в другой форме записи:

$$y'' + 2y' + 5y = 3u \quad (2.10)$$

в уравнения переменных состояния. Обозначим переменные состояния:

$$x(1) = y, \quad x(2) = y', \quad \text{т.е.} \quad x = \begin{bmatrix} y \\ y' \end{bmatrix}.$$

Перепишем уравнение (2.10) следующим образом:

$$y'' = -2y' - 5y + 3u, \quad (2.11)$$

и дополним его уравнением вида:

$$y' = 0y + 1y' + 0u. \quad (2.12)$$

Тогда уравнение (2.10) можно записать следующим образом

$$\begin{bmatrix} y' \\ y'' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} * \begin{bmatrix} y \\ y' \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} * u,$$

$$y = [1 \quad 0] * \begin{bmatrix} y \\ y' \end{bmatrix} + [0] * u. \quad (2.13)$$

Учитывая принятые обозначения для переменных состояния полученные уравнения приобретают вид:

$$\begin{cases} \frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} * x + \begin{bmatrix} 0 \\ 3 \end{bmatrix} * u \\ y = [1 \quad 0] * x + [0] * u \end{cases} \quad (2.14)$$

Обозначим матрицы системы уравнений (2.14):

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \quad C = [1 \quad 0], \quad D = [0].$$

Тогда, в конечном итоге уравнения переменных состояния примут следующий вид:

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (2.15)$$

Отметим, что все приведенные модели являются эквивалентными, то есть, зная любую из них, можно получить все остальные

.Для объектов, функционирование которых по тем или иным причинам представляется для дискретного времени $t_k = kT$ (в данном случае T – интервал дискретизации), то есть для дискретных объектов , наиболее общим видом описания является разностное уравнение (аналог дифференциально-

го):

$$y_k + a_1 y_{k-1} + \dots + a_{na} y_{k-na} = b_1 u_k + b_2 u_{k-1} + b_3 u_{k-2} + \dots + b_{nb} u_{k-nb+1}, \quad (2.16)$$

где $y_{k-i} = y[(k-i)T]$, $u_{k-j} = u[(k-j)T]$.

Связь между сигналами может быть отражена также

• через дискретную свертку:

$$y_k = \sum_{i=0}^k \omega_i u_{k-1}, \quad (2.17)$$

где ω – ординаты решетчатой весовой функции объекта, или, с использованием аппарата Z – преобразования:

$$Y(z) = \sum_{k=0}^{\infty} y_k z^{-k}, \quad \text{где } z = e^{pT} \quad (2.18)$$

• через дискретную передаточную функцию:

$$W(z) = \frac{Y(z)}{u(z)} = \frac{B(z)}{A(z)}, \quad (2.19)$$

которая определяется на основании разностного уравнения после применения к обеим частям этого уравнения Z – преобразования:

$$\begin{aligned} (1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na}) Y(z) = \\ = (b_1 + b_2 z^{-1} + b_3 z^{-2} + \dots + b^{nb} z^{-nb+1}) U(z) \end{aligned} \quad (2.20)$$

Заметим, что Z – преобразованием решетчатой импульсной переходной характеристики является $W(z)$, то есть $Z\{\omega_i\} = W(z)$.

Отметим далее, что на практике в большинстве случаев измерение непрерывных сигналов производится в дискретные моменты времени, что представляет определенное удобство при последующей обработке данных на ЭВМ. Поэтому представление непрерывных объектов дискретными моделями является актуальной задачей. Хотя такое представление может быть осуществлено с некоторой степенью приближенности.

Возможны следующие способы перехода от непрерывных моделей к дискретным.

1. С применением Z – преобразования со следующей цепочкой переходов:

$$W(p) \rightarrow L^{-1}\{W(p)\} = \omega(t) \rightarrow \omega(kT) = \omega_k \rightarrow W(z) = Z\{\omega_k\}. \quad (2.21)$$

2. С заменой разности производных в дифференциальном уравнении, описывающем непрерывный объект:

$$\frac{dy(t)}{dn} \approx \frac{y_k - y_{k-1}}{T}, \quad \frac{d^2 y(t)}{dt^2} \approx \frac{y_k - 2y_{k-1} + y_{k-2}}{T^2} \quad \text{и т.д.} \quad (2.22)$$

(данный подход дает приемлемую точность только при малых T).

3. С заменой $p = \frac{2}{T} \frac{z-1}{z+1}$ (приближенный способ, предложенный А. Тастиным и называемый билинейным преобразованием).

Для дискретных объектов также может быть использовано описание через переменные состояния:

$$\begin{aligned} X_k &= AX_{k-1} + Bu_{k-1}, \\ Y_k &= CX_k + Du_k, \end{aligned} \quad (2.23)$$

Переходную функцию и частотные характеристики – так же, как и для непрерывных сигналов.

Отметим только, что множитель $z^{-1} = e^{-pT}$ представляет собой оператор задержки.

3. Приложение MATLAB пакет System Identification Toolbox для идентификации объектов автоматизации технологических процессов

3.1 Назначение пакета System Identification Toolbox

Пакет System Identification Toolbox содержит средства для создания математических моделей линейных динамических объектов (систем) на основе наблюдаемых входных/выходных данных. Он имеет удобный графический интерфейс, позволяющий организовывать данные и создавать модели. Методы идентификации, входящие в пакет применимы для решения широкого класса задач – от проектирования систем управления и обработки сигналов до анализа временных рядов. Основные свойства пакета следующие:

- простой и гибкий интерфейс;
- предварительная обработка данных, включая фильтрацию, удаление трендов и смещений;
- выбор диапазона данных для анализа;
- эффективные методы авторегрессии;
- возможности анализа отклика систем во временной и частотной областях;
- отображение нулей и полюсов передаточной функции системы;
- анализ невязок при тестировании модели.

Графический интерфейс пакета упрощает как предварительную обработку данных, так и диалоговый процесс идентификации модели. Операции загрузки и сохранения данных, выбора их диапазона, исключения смещений и трендов выполняются минимальными усилиями и доступны из главного меню.

Представление данных и моделей систем и объектов организовано таким образом, что в процессе интерактивной идентификации пользователь легко может вернуться к предыдущему этапу работы. Для начинающих пользователей существует возможность просматривать последующие этапы. Специалисту графические средства позволяют отыскать любую из ранее полученных моделей и оценить ее качество с другими моделями.

Пакет поддерживает все традиционные виды моделей, включая модели передаточных функций, описания для переменных состояния (как для непрерывного, так и для дискретного времени) и другие, с произвольным числом входов и выходов.

3.2. Виды моделей пакета System Identification Toolbox

Основные виды моделей, используемых при идентификации объектов автоматизации, были рассмотрены ранее. Принимая во внимание это обстоятельство, и обозначая моменты дискретного времени тем же символом t , что и непрерывное время (в данном случае $t = 0, 1, 3, \dots$), приведем несколько распространенных моделей дискретных объектов, используемых в пакете System Identification Toolbox для временной области, учитывающих действие шума наблюдения.

1. Модель авторегрессии AR (AutoRegressive) – считается самым простым описанием:

$$A(z) y(t) = u(t), \text{ где } A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na}. \quad (3.1)$$

2. ARX – модель (Autoregressive with eXternal input) – более сложная модель:

$$A(z) y(t) = B(z) u(t) + e(t), \quad (3.2)$$

Или в развернутом виде:

$$y(t) + a_1 y(t-1) + \dots + a_{na} y(t-n) = b_1 u(t) + b_2 u(t-1) + \dots + b_{nb} u(t-m) + e(t).$$

Здесь и ниже $e(t)$ – дискретный белый шум.

$$B(z) = b_1 + b_2 z^{-1} + \dots + b_{bn} z^{-nb+1}.$$

3. ARMAX-модель (AutoRegressive-Moving Average wiht eXternal input – модель авторегрессии скользящего среднего):

$$A(z) y(t) = B(z) u(t - nk) + C(z) e(t), \quad (3.3)$$

где nk – величина задержки (запаздывания),

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{nc} z^{-nc}.$$

4. Модель «вход-выход» (в иностранной литературе такая модель называется «Output-Error», то есть «выход-ошибка», сокращенно OE):

$$y(t) = \frac{B(z)}{F(z)} u(t - nk) + e(t), \quad (3.5)$$

где $F(z) = 1 + f_1 z^{-1} + f_2 z^{-2} + \dots + f_{nf} z^{-nf}$.

5. Так называемая модель Бокса-Дженкинса (BJ):

$$y(t) = \frac{B(z)}{F(z)} u(t - nk) + \frac{C(z)}{D(z)} e(t), \quad (3.6)$$

полиномы $B(z)$, $F(z)$, $C(z)$ определены ранее, а полином $D(z)$ определяется по формуле:

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{nd} z^{-nd}.$$

Данные модели можно рассматривать, как частные случаи обобщенной параметрической линейной структуры:

$$A(z)y(t) = \frac{B(z)}{F(z)} u(t - nk) + \frac{C(z)}{D(z)} e(t), \quad (3.7)$$

при этом все они допускают расширение для многомерных объектов (имеющих несколько входов и выходов).

6. Модель для переменных состояния (State-space):

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t) + v(t), \end{aligned} \quad (3.8)$$

где A , B , C , D – матрицы соответствующих размеров, $v(t)$ – коррелированный белый шум наблюдений.

Возможна и другая (так называемая обновленная или каноническая) форма представления данной модели:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ke(t), \\ y(t) &= Cx(t) + Du(t) + e(t), \end{aligned} \quad (3.9)$$

где K – некоторая матрица (вектор столбец), $e(t)$ – дискретный белый шум (скаляр).

Следует обратить особое внимание, что в своей работе пакет System Identification Toolbox использует три внутренних вида матричного представления моделей, которые с помощью операторов и функций пакета преобразуются во все выше перечисленные виды моделей объектов:

- так называемый **тета-формат** (для временных моделей);
- **частотный формат** (для частотных моделей);
- **формат нулей и полюсов**.

3.3. Основные операторы и функции пакета System Identification Toolbox

Ниже приводятся основные операторы и функции пакета System Identification Toolbox, которые набираются в командной строке MATLAB или могут быть использованы при написании программ для m-файлов. Наиболее полную информацию о содержании, написании и использовании этих функций можно получить в литературе [] и справочной части help MATLAB.

Подсказка:

idhelp – используется для вызова подсказки о возможностях пакета.

Демонстрационные

iddemo – используется для вызова демонстрационных примеров.

Графический Интерфейс Пользователя:

ident – команда вызова графического интерфейса пользователя.

midprefs – команда задает (изменяет) директорию для файла midprefs.mat, хранящего информацию о начальных параметрах графического интерфейса пользователя при его открытии.

Моделирование и прогноз:

predict – команда осуществляет прогноз выхода объекта по его тета-модели и с учетом информации о его предыдущих фактических значениях выхода (рекомендуется для расчета прогноза значений временной последовательности).

pe – вычисляет ошибку модели при заданном входе и известном выходе объекта.

idsim – возвращает выход модели тета-формата.

Обработка Данных:

iddata – создает файл объекта данных.

detrend – удаляет тренд из набора данных.

idfilt – команда фильтрует данные с помощью фильтра Баттерворта.

idinput – команда генерирует входные сигналы для идентификации.

merge – объединяет несколько экспериментов.

misdata – оценивает и заменяет потерю входных и выходных данных в файле созданном с помощью команды **iddata**.

esample – восстанавливает форму квантованного сигнала данных преобразованием и интерполяцией и изменяет частоту дискретизации.

Функции непараметрического оценивания:

covf – выполняет расчет авто- и взаимных корреляционных функций совокупности экспериментальных данных.

cra – определяет оценку ИХ методом коррелированного анализа для одномерного (один вход – один выход) объекта.

etfe – возвращает оценку дискретной передаточной функции для обобщенной линейной модели одномерного объекта в частотной форме.

impulse – выводит на дисплей импульсную характеристику модели.

spa – возвращает частотные характеристики объекта и оценки спектральных плотностей его сигналов для обобщенной линейной модели объекта (возвращает модель объекта в частотном формате).

step – выводит на дисплей переходную (временную) характеристику модели объекта (реакция на единичное ступенчатое воздействие).

Функции параметрического оценивания:

ar – оценивает параметры модели авторегрессии (AR), то есть коэффициенты полинома $A(z)$, при моделировании скалярных временных последовательностей.

armax – оценивает параметры модели ARMAX.

arx – оценивает параметры моделей ARX и AR.

bj – оценивает параметры модели Бокса-Дженкинса.

lvar – оценивает параметры скалярной AR-модели.

iv4 – оценивает параметры для моделей ARX с использованием четырехступенчатого метода инструментальной переменной.

n4sid – используется для оценивания параметров моделей для переменных состояния в канонической форме при произвольном числе входов и выходов.

ivx – оценивает параметры ARX-моделей методом инструментальной переменной.

oe – оценивает параметры OE-модели.

pem – оценивает параметры обобщенной многомерной линейной модели.

Функции задания структуры модели:

idpoly – создавать модель объекта в виде полинома.

idss – создает модель объекта в виде переменных состояния.

idarx – создает многопараметрическую ARX-модель объекта.

idgrey – создает модель объекта, созданную пользователем.

arx2th – создает матрицу модели тета-формата по полиномам ARX-модели многомерного объекта.

canform – создает каноническую форму модели для переменных состояния многомерного объекта.

mf2th – преобразует структуру модели для переменных состояния в тета-формат.

poly2th – создает модель тета-формата из исходной модели «вход-выход».

Функции извлечения данных о моделях:

arxdata – возвращает матрицы коэффициентов полиномов ARX-моделей, а также их среднеквадратические отклонения.

polydata – возвращает матрица коэффициентов полиномов по формуле 3.7 данной работы.

ssdata – функция возвращает матрицы(и величину интервала дискретизации в дискретном случае) ss-моделей (моделей переменных состояния).

tfdata – данная функция возвращает числитель и знаменатель передаточной функции.

zpkdata – функция возвращает нули, полюсы и обобщенные коэффициенты передачи для каждого канала модели тета-формата или LTI-модели (если используется пакет Control System Toolbox с именем sys).

idfrd – данная функция создает частотную модель объекта в frd-формате.

idmodred – функция уменьшает порядок модели объекта.

c2d, d2c – первая функция преобразует непрерывную модель в дискретную. Вторая – наоборот.

ss, tf, zpk, frd – функции создания моделей стационарных систем в виде модели переменных состояния (ss), передаточной функции по ее заданным нулям и полюсам (zpk), передаточной функции, записанной в операторном виде (tf) и в частотном виде (frd).

Функции отображения модели:

bode, bodeplot, ffplot – функции отображения логарифмических частотных характеристик.

plot – отображение входных-выходных данных для данных объекта.

present – функция отображения вида модели тета-формата с оценкой среднеквадратического отклонения, функции потерь и оценки точности модели..

pzmap – отображает нули и полюсы модели (с областями неопределенности).

nyquist – отображает диаграмму Найквиста (гадограф АФХ) передаточной функции.

view – отображение LTI-модели (при использовании пакета Control System Toolbox).

Фнкции проверки адекватности модели:

compare – функция позволяет сравнить выходы модели и объекта с выводом на дисплей сравнительных графиков и указанием оценки адекватности модели.

resid – функция вычисляет остаточную ошибку для заданной модели и соответствующие корреляционные функции.

Функции выбора структуры модели:

aic, fpe – функции вычисляются информационный критерий АИС и конечную ошибку модели.

arxstruc – функция вычисляет функции потерь для ряда различных конкурирующих ARX-моделей с одним выходом.

selstruc – функция осуществляет выбор наилучшей структуры модели из ряда возможных.

3.4. Идентификация с использованием блоков Simulink

В состав библиотеки System ID Blocks блоков Simulink входят блоки, позволяющие производить оценивание ряда типовых моделей (рис. 3.1):

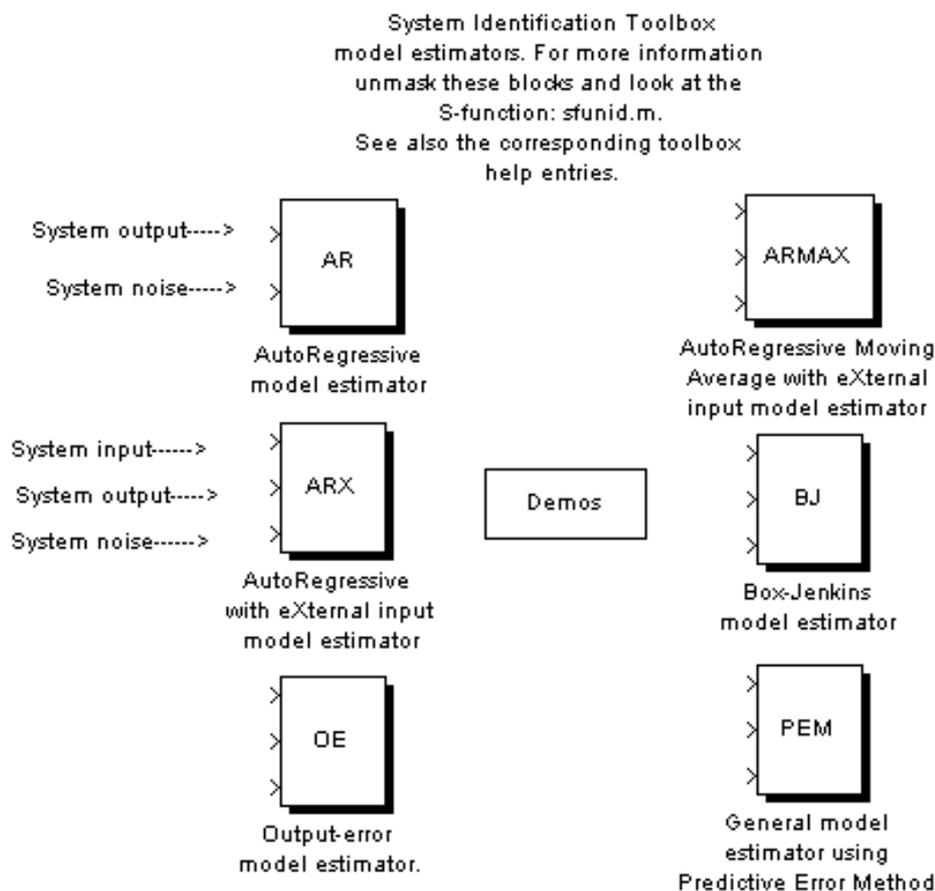


Рис. 3.1. Библиотека блоков идентификации Simulink

- модели авторегрессии AR (AutoRegressive model estimator);
- ARX-модели (AutoRegressive Moving Average with eXternal input model estimator);

- ARX-модели (AutoRegressive Moving Average with eXternal input model estimator);
- модели Бокса-Дженкинса BJ (Box-Jenkins model estimator);
- обобщенной линейной модели (General model estimator using Predictive Error Method);
- модели «вход-выход» OE (Output-error model estimator).

Правила работы с данными блоками аналогичны правилам для других блоков Simulink. Полученная модель отображается в основном окне MATLAB.

4. Применение пакета System Identification Toolbox для идентификации объектов автоматизации технологических процессов

4.1. Постановка задачи идентификации и ее основные этапы

В качестве примера использования пакета System Identification Toolbox для идентификации технологических объектов управления возьмем один из наиболее распространенных технологических процессов, применяемых в строительной и дорожной индустрии – тепловой процесс обработки строительных материалов. Перечень тепловых объектов промышленности строительных и дорожных материалов достаточно велик, так как почти во всех технологических процессах имеется тепловое воздействие. Здесь в качестве примера ТОУ можно привести котельные установки, сушильные барабаны в производствах цемента, керамических изделий, асфальтобетонных смесей, стекла и т.д. Кроме того, автоматизация тепловых процессов применяется при производстве железобетонных изделий, горячего водоснабжения и отопления промышленных и гражданских сооружений.

Характерной особенностью тепловых ТОУ является управляющее воздействие – изменение тепловой мощности, определяющее температурный режим процессов, их производительности, а также качество получаемых изделий, материалов и полуфабрикатов. Кроме основного в тепловых ТОУ есть другие воздействия: изменение давления (например, в варочной части печи при производстве стекла), изменение зазора (в клапанах, задвижках и т.д.), частоты вращения рабочих органов. Поэтому в целом тепловые ТОУ являются достаточно сложными многомерными объектами, и температура в их рабочей зоне определяется не только основным входным воздействием, но и всеми перечисленными воздействиями.

Допустим, что имеется некоторый тепловой технологический процесс, в котором входным воздействием на ТОУ является энергия электрического тока, выраженная в кВт, а выходным регулируемым параметром – темпе-

ратура в градусах Цельсия. Процесс идентификации ТОУ включает следующие этапы:

1. априорный анализ ТОУ с целью выбора структуры модели;
2. проведение предварительного (небольшого по объему) исследования объекта с целью уточнения оценки структуры модели (этот этап желателен, особенно при отсутствии априорной информации о ТОУ);
3. разработка методики основного экспериментального исследования ТОУ, составление плана эксперимента;
4. проведение основного экспериментального исследования для получения массива данных (u_i, y_i) ;
5. математическая обработка массива данных (с использованием пакета System Identification Toolbox) с целью определения параметров модели и ее адекватности, доверительных границ параметров и выходной координаты модели.

Экспериментальные методы исследования моделей ТОУ обычно базируются на трех допущениях:

1. ТОУ является системой с сосредоточенными параметрами;
2. статические и динамические свойства ТОУ неизменны во времени (ТОУ стационарен);
3. уравнения моделей ТОУ линеаризуются в малом, т.е. при небольших отклонениях $\pm \Delta y_i$ от выбранной "рабочей" точки (рабочего режима) ТОУ.

Массив данных $[u_i, y_i]$ образуется в результате трудоемкой операции расшифровки регистрограмм по приборам измерительной системы. Однако широкое развитие микропроцессорной и вычислительной техники и внедрение ее в производственные технологические процессы позволили существенно усовершенствовать техническое обеспечение идентификации ТОУ. Обработка массива данных с помощью пакета System Identification Toolbox предполагает следующие этапы:

1. обработка и преобразование данных с целью создания файла данных;
2. непараметрическое оценивание данных с целью предварительно определения основных характеристик ТОУ;
3. параметрическое оценивание данных с целью создания различных видов моделей (описанных во втором разделе) в тета-формате;
4. задание структуры модели;
5. изменение и уточнение структуры модели (если это необходимо);
6. проверка адекватности и сравнение различных моделей с целью выбора наилучшей;
7. преобразование модели из тета-формата в вид удобный для дальнейшего использования при анализе и синтезе системы управления.

На каждом этапе идентификации имеется возможность графического отображения результатов моделирования и извлечения необходимой информации об объекте или данных.

4.2. Обработка и преобразование данных

В принятом примере ТОУ в результате проведенного эксперимента был получен массив данных, состоящий из 1000 значений входного параметра – электрической мощности нагревателей в кВт и 1000 значений выходного параметра – температуры в объекте в градусах Цельсия. Интервал дискретизации (промежутки времени, через которые производились измерения входных и выходных величин) в ходе эксперимента был принят равным 0.1 с. Этот массив данных при использовании в дальнейшем в пакете System Identification Toolbox необходимо объединить в единый файл, содержащий необходимую информацию о входных и выходных параметрах объекта, их значениях и размерностях измерения. Для этого загрузим в рабочую область MATLAB исходные данные для составления файла данных с помощью команды:

```
>> load datta
```

В результате выполнения команды в рабочей области появились массив входных переменных u и массив выходного параметра y . Интервал дискретизации указывается дополнительно:

```
>> ts=0.1;
```

Для объединения исходных данных в единый файл пользуются командой:

```
>> dan=iddata(y,u,ts)
```

Результат выполнения команды комментируется следующей фразой MATLAB:

```
Time domain data set with 1000 samples.  
Sampling interval: 0.1
```

```
Outputs    Unit (if specified)  
  y1
```

```
Inputs     Unit (if specified)  
  u1
```

Сформированный файл указывает, что он содержит результаты 1000 измерений с интервалом дискретизации 0.1 с. Входными переменными является массив $u1$, а выходным параметром – $y1$.

Для наглядности сформированного файла необходимо в его структуру ввести обозначения входных и выходных данных:

```
>> set(dan,'InputName','Мощность','OutputName','Температура');
```

Данная команда эквивалентна командам:

```
>> dan.outputn = 'Температура';
```

```
>> dan.inputn = 'Мощность';
```

Для указания размерностей параметров файла данных необходимо воспользоваться командами:

```
>> dan.inputUnit = 'кВт';
```

```
>> dan.outputUnit = 'град Ц';
```

В конечном итоге сформированный файл данных `dan.m` имеет следующий вид:

```
Time domain data set with 1000 samples.  
Sampling interval: 0.1
```

```
Outputs      Unit (if specified)  
Температура  град Ц
```

```
Inputs      Unit (if specified)  
Мощность    кВт
```

Полную информацию о файле `dan.m` можно получить воспользовавшись командой:

```
>> get ( dan );
```

```
ans =
```

```
      Domain: 'Time'  
      Name: []  
      OutputData: [1000x1 double]  
                y: 'Same as OutputData'  
      OutputName: {'Температура'}  
      OutputUnit: {'град Ц'}  
      InputData: [1000x1 double]  
                u: 'Same as InputData'  
      InputName: {'Мощность'}  
      InputUnit: {'кВт'}  
      Period: Inf  
      InterSample: 'zoh'  
      Ts: 0.1000  
      Tstart: []  
      SamplingInstants: [1000x0 double]  
      TimeUnit: ''  
  
      ExperimentName: 'Exp1'  
      Notes: []  
      UserData: []
```

Для графического представления данных можно воспользоваться командой `plot (dan)`, либо командой `idplot (datta)`, однако в последнем случае графики не будут содержать информации о названии переменных и их размерностях. Исходные данные с использованием команды `plot (dan)` приведены на рис. 4.1.

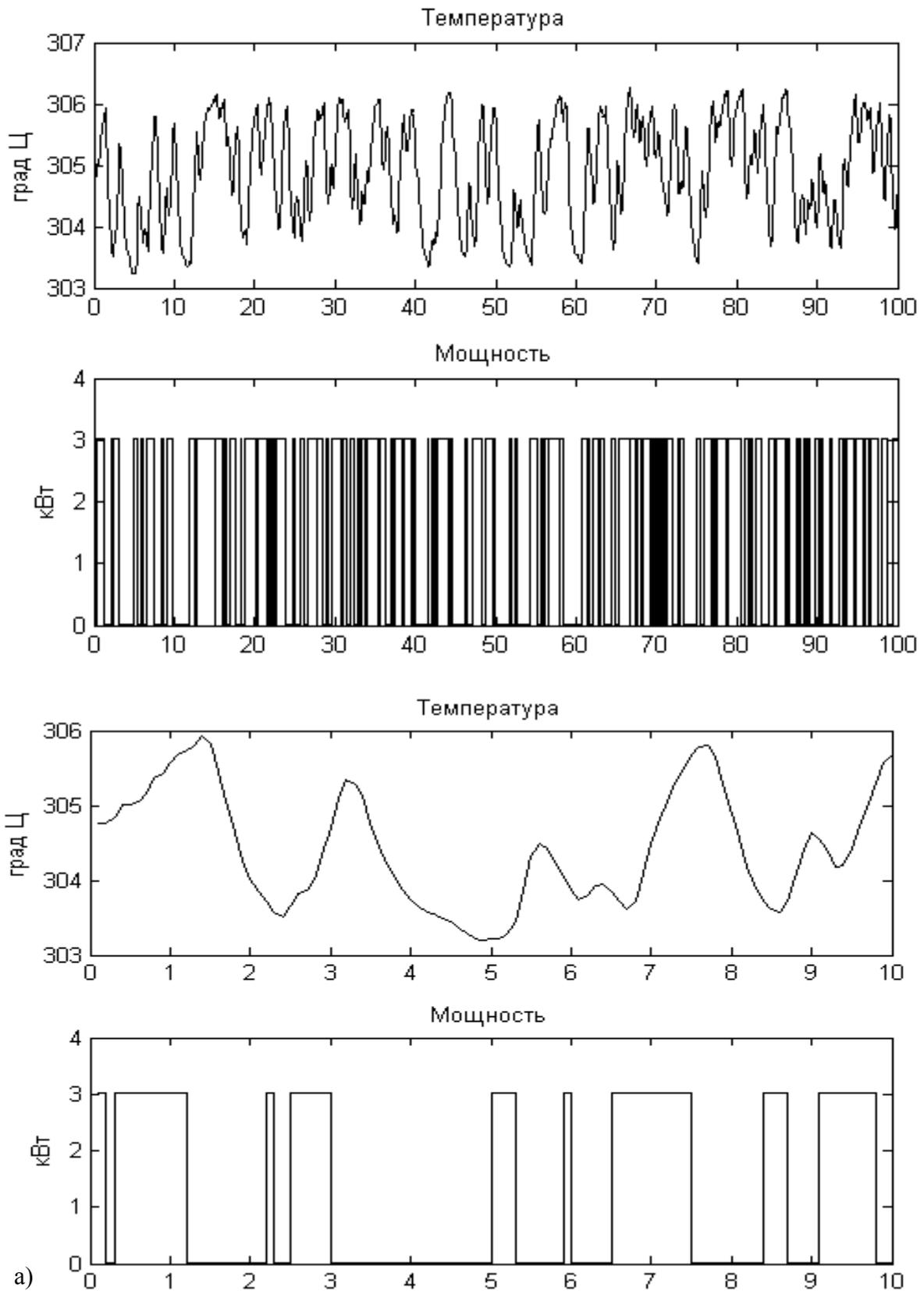


Рис. 4.1. Исходные данные для идентификации ТОУ:
а) во всем временном интервале; б) на интервале времени от 0 до 10 с.

Для дальнейшего использования полученных исходных данных необходимо провести предварительную обработку этих данных с целью удаления тренда из набора данных и если необходимо отфильтровать данные с помощью имеющихся средств в пакете System Identification Toolbox.

Для удаления тренда пользуются функцией:

```
>> zdan = dtrend (dan);
```

В результате этого получен новый файл zdan.m, в котором отсутствует постоянная составляющая сигналов. Этот файл в дальнейшем будет нами использован для построения моделей ТОУ. Кроме указанной команды удаления тренда в пакете System Identification Toolbox имеются другие функции обработки данных эксперимента, которые приведены в описании пакета System Identification Toolbox.

Применение этих функций производится в тех случаях, когда проведен предварительный анализ ТОУ и определены возможные помехи либо некоторые другие динамические характеристики, либо появляется необходимость изменить интервал дискретизации в случае повышенной погрешности представления модели ТОУ в ходе параметрического оценивания его.

Следующим этапом идентификации является непараметрическое оценивание исходных данных.

4.3. Непараметрическое оценивание исходных данных

Функции непараметрического оценивания исходных данных используются в основном для определения статистических характеристик массивов исходных данных.

Следует напомнить, что в ходе планирования и проведения эксперимента сказать заранее, по какой реализации пойдет процесс, невозможно. Поэтому процесс получения массива входных и выходных переменных ТОУ можно назвать стохастическим и для его обработки применить статистические характеристики.

Как известно, статистические свойства случайной величины x определяют по ее *функции распределения* (интегральному закону распределения) $F(x)$ или *плотности вероятности* (дифференциальному закону распределения) $\omega(x)$.

Случайные величины могут иметь различные законы распределения: равномерный, нормальный, экспоненциальный и др. Во многих задачах автоматического управления очень часто приходится иметь дело с нормальным законом распределения (или законом Гаусса), который получается если случайная величина определяется суммарным эффектом от действия большого числа различных независимых факторов.

Кроме этого, случайная величина x при нормальном законе распределения полностью определяется *математическим ожиданием* (средним).

Функции $F(x)$ и $\omega(x)$ являются простейшими статистическими характе-

ристиками случайного процесса, но они не дают достаточного представления о том, какой характер будут иметь отдельные реализации случайного процесса.

Чтобы в какой-то мере охарактеризовать внутреннюю структуру случайного процесса, т.е. учесть связь между значениями случайного процесса в различные моменты времени или, иными словами, учесть степень изменчивости случайного процесса, необходимо ввести понятия *корреляционной (автокорреляционной) функции* и *спектральной плотности* случайного процесса.

Корреляционной функцией случайного процесса $X(t)$ называют неслучайную функцию двух аргументов $R(t_1; t_2)$, которая для каждой пары произвольно выбранных значений моментов времени t_1 и t_2 равна математическому ожиданию произведения двух случайных величин случайного процесса, соответствующих этим моментам времени.

Между дисперсией случайного процесса и корреляционной функцией существует прямая связь – *дисперсия случайного стационарного процесса равна значению корреляционной функции*.

Статистические свойства связи двух случайных процессов $X(t)$ и $G(t)$ можно можно охарактеризовать *взаимной корреляционной функцией* $R_{xg}(t_1, t_2)$. Взаимная корреляционная функция $R_{xg}(\tau)$ характеризует взаимную статистическую связь двух случайных процессов $X(t)$ и $G(t)$ в разные моменты времени, отстоящие друг от друга на промежуток времени τ .

Если случайные процессы $X(t)$ и $G(t)$ статистически не связаны друг с другом и имеют равные нулю средние значения, то их взаимная корреляционная функция для всех τ равна нулю. Однако обратный вывод о том, что если взаимная корреляционная функция равна нулю, то процессы не-взаимосвязаны, можно сделать вывод лишь в отдельных случаях (в частности, для процессов с нормальным законом распределения), общей же силы обратный закон не имеет.

Анализируя свойства корреляционной функции можно сделать вывод: чем слабее взаимосвязь между предыдущим $X(t)$ и последующим $X(t+\tau)$ значениями случайного процесса, тем быстрее убывает корреляционная функция $R_x(\tau)$. Случайный процесс, в котором отсутствует связь между предыдущими и последующими значениями, называют *чистым случайным процессом* или *белым шумом*. В случае белого шума время корреляции $\tau_R=0$ и корреляционная функция представляет собой δ -функцию (функцию веса).

При исследовании автоматических систем управления удобно пользоваться еще одной характеристикой случайного процесса, называемой *спектральной плотностью*.

Спектральная плотность $S_x(\omega)$ случайного процесса $X(t)$ определяется как преобразование Фурье корреляционной функции $R_x(\tau)$. Физический смысл спектральной плотности состоит в том, что она характеризует

распределения мощности сигнала по частотному спектру.

В пакете System Identification Toolbox имеется четыре функции **covf**, **cra**, **efte** и **spa**, характеризующие статистические характеристики массива входных и выходных данных ТОУ.

Функция **covf** выполняет расчет авто- и взаимных корреляционных функций массива экспериментальных данных, записанных в файле *datta.m*.

Написание этой функции следующее:

```
R = covf (dan, M),
```

где *M* – максимальная величина дискретного аргумента, для которой рассчитывается корреляционная функция, минус единица (по умолчанию можно поставить []);

R – матрица, элементы первого столбца которой – значения дискретного аргумента, элементы второго столбца – значения оценки автокорреляционной функции выходного сигнала (возможно отфильтрованного), элементы третьего столбца – значения оценки автокорреляционной функции входного сигнала (возможно "обеленного"), элементы четвертого столбца – значения оценки взаимной корреляционной функции.

Однако более удобно пользоваться функцией **cra**, которая определяет оценку ИХ методом корреляционного анализа для одномерного (один вход – один выход) объекта:

```
cra ( z );
```

```
[ ir , R, cl ] = cra ( z, M, na, plot );
```

```
cra ( R ),
```

где *z* – матрица экспериментальных данных (в нашем случае *datta*);

M – максимальное число дискретного аргумента;

na – порядок модели авторегрессии (степени многочлена $A(z)$), которая используется для расчета параметров "обеляющего" фильтра $\Phi(z)$, по умолчанию *na* = 10. При *na* = 0 в качестве идентифицирующего используется непреобразованный входной сигнал;

plot – *plot* = 0 означает отсутствие графика, *plot* = 1 (по умолчанию) – график полученной оценки ИХ вместе с 99%-м доверительным коридором, *plot* = 2 – выводятся графики всех корреляционных функций.

ir – оценка ИХ (вектор значений);

cl – 99%-й доверительный коридор для оценки ИХ.

Для нашего примера эти величины имеют следующие значения:

```
>> [ir, R, cl] = cra(zdan, [], [], 2)
```

```
ir =
```

```
-0.0021
```

```
-0.0009
```

```
0.0048
```

```
0.0686
```

```
0.1273
```

```
0.1413
```

```
0.1300
```

```
0.1087
```

0.0859
0.0666
0.0503
0.0381
0.0293
0.0208
0.0167
0.0127
0.0097
0.0092
0.0079
0.0071
0.0076

R =

-20.0000	-0.0064	-0.0006	-0.0333
-19.0000	-0.0044	-0.0336	-0.0313
-18.0000	-0.0023	0.0101	-0.0295
-17.0000	0.0001	0.0047	-0.0392
-16.0000	0.0022	0.0334	-0.0424
-15.0000	0.0041	-0.0395	-0.0406
-14.0000	0.0060	0.0556	-0.0295
-13.0000	0.0077	-0.0109	-0.0204
-12.0000	0.0096	-0.0065	-0.0146
-11.0000	0.0117	0.0370	-0.0053
-10.0000	0.0142	-0.0252	-0.0003
-9.0000	0.0174	0.0030	0.0029
-8.0000	0.0220	-0.0026	0.0081
-7.0000	0.0282	0.0012	0.0158
-6.0000	0.0367	-0.0037	0.0098
-5.0000	0.0481	0.0017	0.0102
-4.0000	0.0626	-0.0011	0.0079
-3.0000	0.0802	0.0007	-0.0003
-2.0000	0.0991	0.0068	0.0019
-1.0000	0.1154	-0.0043	-0.0014
0	0.1232	1.3795	-0.0071
1.0000	0.1154	-0.0043	-0.0031
2.0000	0.0991	0.0068	0.0160
3.0000	0.0802	0.0007	0.2294
4.0000	0.0626	-0.0011	0.4261
5.0000	0.0481	0.0017	0.4728
6.0000	0.0367	-0.0037	0.4351
7.0000	0.0282	0.0012	0.3637
8.0000	0.0220	-0.0026	0.2873
9.0000	0.0174	0.0030	0.2229
10.0000	0.0142	-0.0252	0.1682
11.0000	0.0117	0.0370	0.1274
12.0000	0.0096	-0.0065	0.0979
13.0000	0.0077	-0.0109	0.0696
14.0000	0.0060	0.0556	0.0560
15.0000	0.0041	-0.0395	0.0424

16.0000	0.0022	0.0334	0.0324
17.0000	0.0001	0.0047	0.0308
18.0000	-0.0023	0.0101	0.0264
19.0000	-0.0044	-0.0336	0.0238
20.0000	-0.0064	-0.0006	0.0254

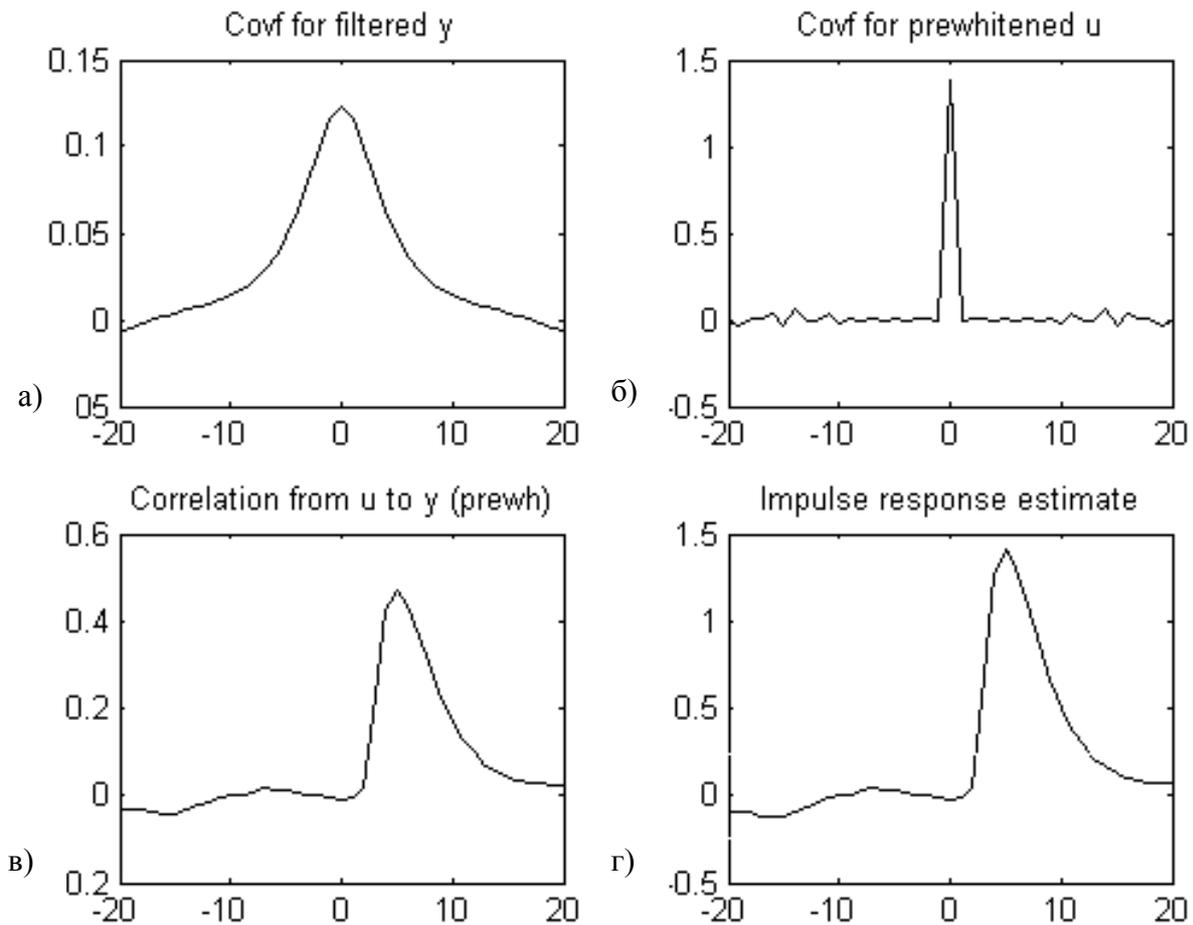


Рис. 4.2. Графики функций: а) автокорреляционная функция выходного сигнала; б) автокорреляционная функция входного сигнала; в) взаимная корреляционная функция; г) импульсная характеристика.

Анализируя полученные зависимости, следует учесть, что по оси абсцисс откладываются промежутки времени $\tau = t_i - t_{i-1}$, а по оси ординат значения корреляционных функций для входного u и выходного y сигналов; значения взаимокорреляционной функции и импульсной характеристики.

Для входной величины с увеличением τ наблюдается резкий спад корреляционной зависимости, что характеризует высоко частотные изменения состояния этой величины. Выходная величина более плавно изменяет свои состояния от одного момента времени к другому и, следовательно, взаимосвязь между предыдущим и последующим значениями выходного сигнала более тесная, чем у входного сигнала.

Для получения частотных характеристик экспериментальных данных воспользуемся функциями оценивания частотных характеристик

```
>> g=spa(dan);
>> bodeplot(g).
```

Результатом выполнения команд является вывод графиков АЧХ и ФЧХ (см. рис. 4.3).

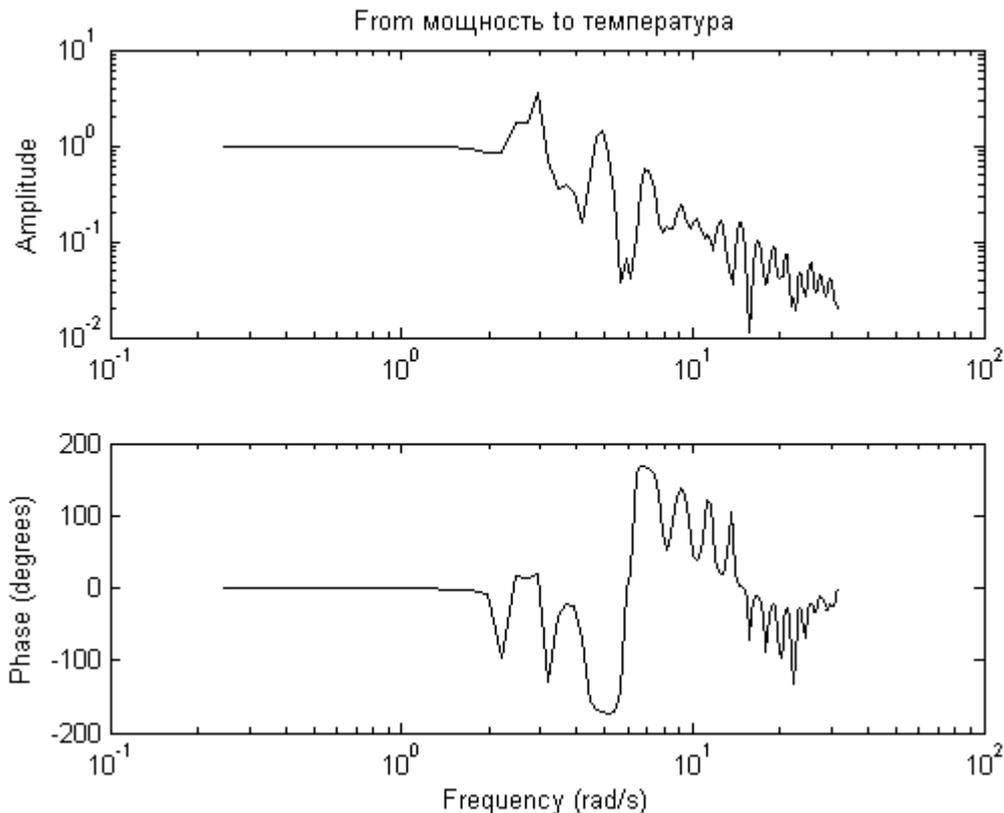


Рис. 4.3. Графики АЧХ и ФЧХ для массива экспериментальных Данных

Полученные зависимости подтверждают высокочастотную составляющую значений входного и выходного сигналов. Границы изменения частот на графиках установлены по умолчанию.

В пакете System Identification Toolbox MATLAB имеется возможность устанавливать границы изменения частот с помощью команды

```
>> w=logspace(w1,w2,N),
```

где w_1 – нижняя граница диапазона частот (10^{w_1}), w_2 – верхняя граница диапазона частот (10^{w_2}) и N – количество точек графика.

Для построения АФХ, ФЧХ и $S(\omega)$ – функции спектральной плотности шума $e(t)$ вычислим g – оценку АФХ и ФЧХ в частотном формате и $priv$ – оценку спектральной плотности шума с помощью команды

```
>> [g,priv]=spa(dan,[],w);
```

Графики АФХ, ФЧХ и $S(\omega)$ построим с доверительным коридором в три среднеквадратических отклонения с помощью команды

```
>> bodeplot([g p], 'sd', 3, 'fill') ,
```

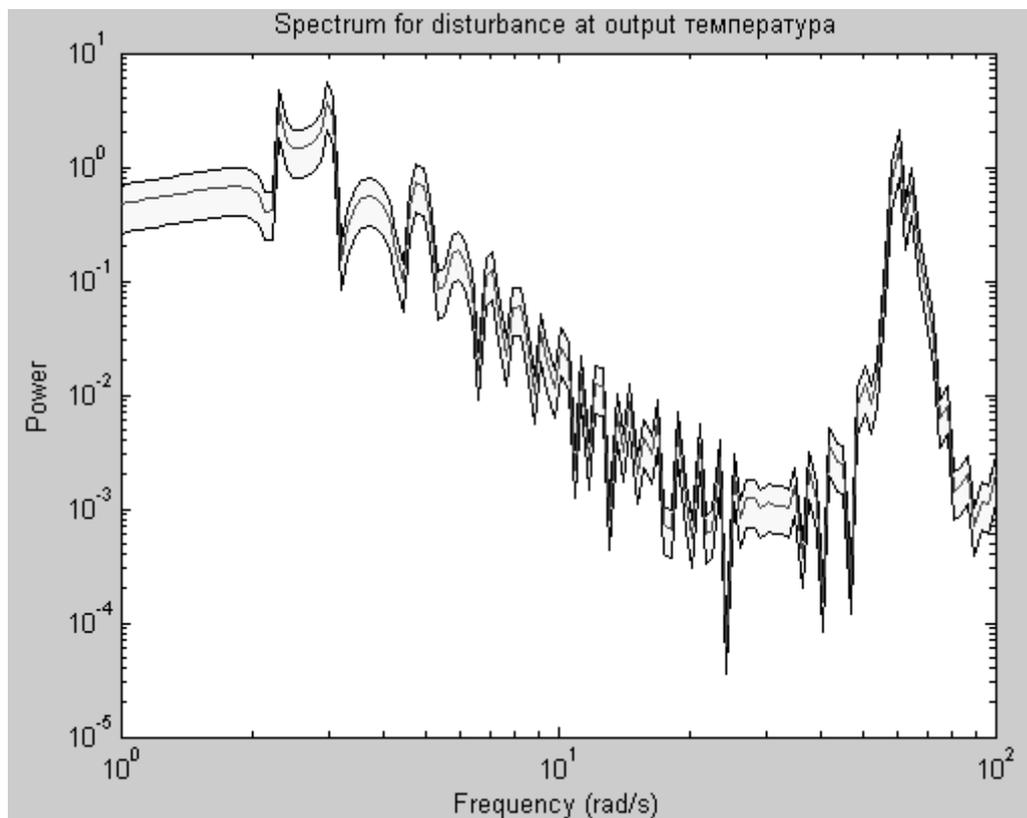
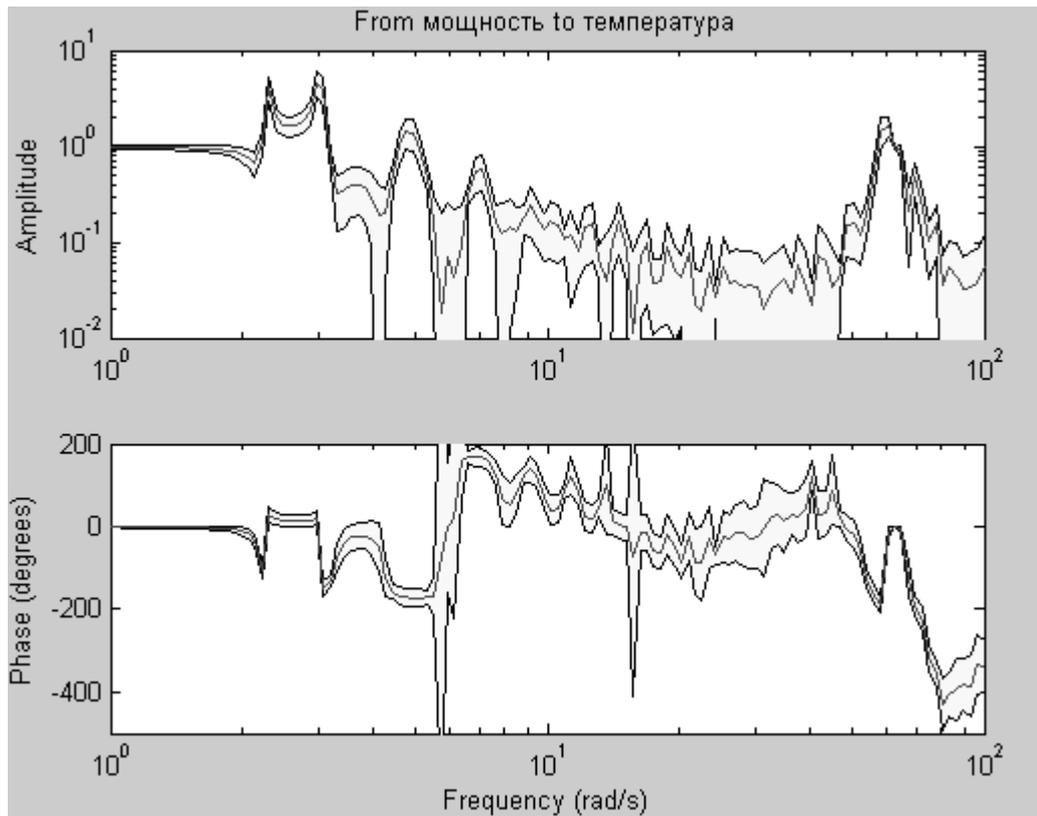


Рис. 4.4 Графики АФХ и ФЧХ и $S(\omega)$ с доверительным коридором

где 'sd' – указывает на сплошную линию доверительного коридора (по умолчанию эта линия штриховая); 3 – величина доверительного коридора в три среднеквадратических отклонения; 'fill' – способ заливки доверительного коридора (в данном случае желтым цветом, см. рис. 4.4).

4.4. Параметрическое оценивание данных

Параметрическое оценивание экспериментальных данных проводится с целью определения параметров модели заданной структуры путем минимизации выбранного критерия качества модели (чаще всего – среднего квадрата рассогласования выходов объекта и его постулируемой модели).

Для проведения параметрического оценивания массив экспериментальных данных необходимо разделить условно на две части (не обязательно равные)

```
>> zdanv=zdan(1:500);  
>> zdane=zdan(501:1000).
```

Первая часть массива данных будет использоваться для параметрического оценивания и построения модели ТОУ. Вторая часть необходима будет для верификации (проверки качества) модели, определения адекватности полученной модели и определения погрешностей идентификации.

Следует напомнить, что *параметрическая идентификация в пакете System Identification Toolbox MATLAB производится в дискретном виде и полученные модели, являются дискретными.*

В третьей главе были рассмотрены различные виды моделей ТОУ, используемых в пакете System Identification Toolbox MATLAB, которые с различной степенью достоверности описывают объект автоматизации. Для выбора наиболее приемлемой структуры и вида моделей при параметрическом оценивании экспериментальных данных в пакете System Identification Toolbox MATLAB имеются специальные функции параметрического оценивания, задания структуры модели, изменения и уточнения структуры модели и выбора структуры модели.

Функция **ar** оценивает параметры модели авторегрессии (AR), т.е. коэффициенты полинома $A(z)$ (см. зависимость 3.1), при моделировании скалярных временных последовательностей. Для ее использования необходимо из массива экспериментальных данных, записанных в файле dan выделить выходную переменную y с помощью команды

```
>> y=dan.y,
```

что равносильно команде

```
>> y=get(dan,'y')
```

```
>> dar=ar(y,4)
```

Discrete-time IDPOLY model: $A(q)y(t) = e(t)$

$A(q) = 1 - 2.195 q^{-1} + 1.656 q^{-2} - 0.4167 q^{-3} - 0.04415 q^{-4}$

Estimated using AR ('fb'/'now') from data set y

Loss function 0.0104048 and FPE 0.0104884

Sampling interval: 1.

Полная информация о модели авторегрессии `dar` может быть получена с помощью команды

```
>> present(dar)
```

Discrete-time IDPOLY model: $A(q)y(t) = e(t)$

$A(q) = 1 - 2.195 (+0.03165) q^{-1} + 1.656 (+0.07533) q^{-2} - 0.4167 (+0.07542) q^{-3} - 0.04415 (+0.03174) q^{-4}$

Estimated using AR ('fb'/'now') from data set `y`

Loss function 0.0104048 and FPE 0.0104884

Sampling interval: 1

Created: 07-Jan-2004 08:47:27

Last modified: 07-Jan-2004 08:47:27.

В информации приведены сведения о том, что модель является дискретной и для оценивания ее параметров используется прямой-обратный метод (разновидность метода наименьших квадратов), на что указывает строковая переменная `'fb'` (используется по умолчанию); для построения модели используются только имеющиеся данные `y`, на что указывает строковая переменная `'now'` (используется по умолчанию); определены: функция потерь Loss function, как остаточная сумма квадратов ошибки и так называемый теоретический информационный критерий Акейке (Akaike's Information Theoretic Criterion – AIC) FPE; интервал дискретизации Sampling interval.

Для изменения параметров моделирования можно воспользоваться другой разновидностью функции `ar`

```
>> [dar,refl]=ar(y,n,approach,win,maxsize,T),
```

где `n` – порядок модели (число оцениваемых коэффициентов); `approach` – аргумент (строковая переменная) определяет метод оценивания:

- `'ls'` – метод наименьших квадратов;
 - `'yw'` – метод Юла-Уокера;
 - `'burg'` – метод Бэрга (комбинация метода наименьших квадратов с минимизацией гармонического среднего);
 - `'gl'` – метод с использованием гармонического среднего;
- аргумент `win` (строковая переменная) используется в случае отсутствия части данных:
- `'now'` – используется по умолчанию;
 - `'prw'` – отсутствующие данные заменяются нулями, так что суммирование начинается с нулевого момента времени;
 - `'row'` – последующие отсутствующие данные заменяются нулями, так что суммирование расширяется до момента времени $N+n$;
 - `'ppw'` – и начальные, и отсутствующие данные заменяются нулями (используется в алгоритме Юла-Уокера);

возвращаемые величины:

- `dar` – полученная модель авторегрессии;

- `self` – информация о коэффициентах и функции потерь;

аргумент `maxsize` определяет максимальную размерность задачи; `T` – интервал дискретизации.

Следующая функция `arx` оценивает параметры модели ARX, представленной зависимостью 3.2:

```
>> darx=(zdanv,[na nb nk]);
```

где `na` – порядок (число коэффициентов) полинома $A(z)$; `nb` – порядок полинома $B(z)$; `nk` – величина задержки.

Известно, что с увеличением порядка полиномов улучшается степень адекватности модели реальному объекту. Однако при этом получаются громоздкие выражения, и увеличивается время моделирования. Поэтому для нахождения оптимального порядка полиномов воспользуемся *функциями выбора структуры модели*. Зададим пределы изменения порядка модели:

```
>> NN=struc(1:10,1:10,1);
```

Вычислим функции потерь:

```
>> v=arxstruc(zdane,zdanv,NN);
```

Выберем наилучшую структуру порядков полиномов:

```
>> [nn,vmod]=selstruc(v,'plot');
```

где `'plot'` – строковая переменная, определяющая вывод графика зависимости функции потерь от числа оцениваемых коэффициентов модели (см. рис. 4.5).

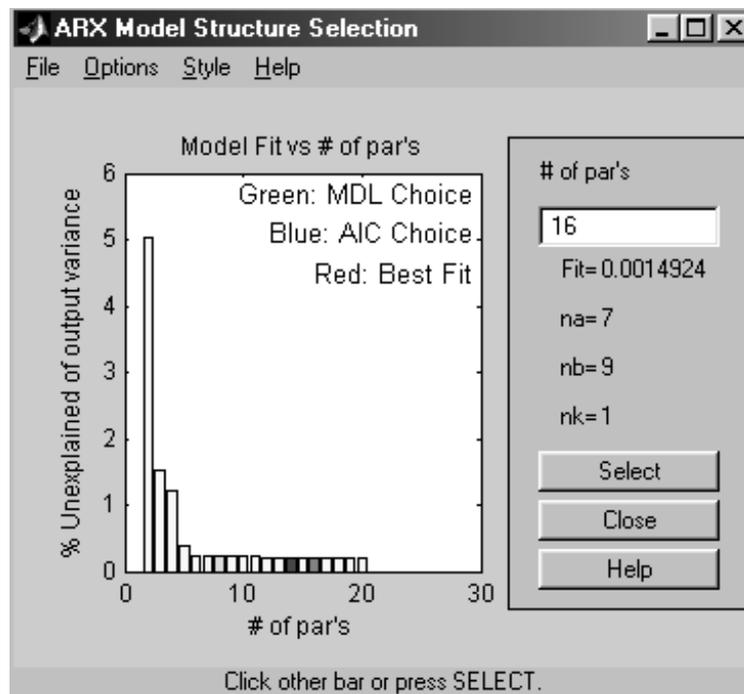


Рис. 4.5. Окно выбора структуры модели

В появившемся окне столбики указывают на величину функции потерь. Подведя курсор к соответствующему столбику, в правом поле окна поя-

вятся значения порядков полиномов na , nb , nk . Воспользуемся рекомендацией, указанной в поле графика и выберем столбик, окрашенный красным цветом для оптимального значения порядков полиномов, которые указаны на рис. 4.5.

Взамен строковой переменной 'plot' возможны варианты

- 'log' – выводится график логарифма функции потерь;
- 'aic' – график не выводится, но возвращается структура, минимизирующая так называемый теоретический информационный критерий Акейке (Akaike's Information Theoretic Criterion – AIC) FPE:

$$v_{mod} = \frac{v}{(1 + 2(d/N))}, \quad (4.1)$$

где v – значение функции потерь, d – число оцениваемых коэффициентов модели, N – объем экспериментальных данных, используемых для оценивания;

- 'mdl' – возвращается структура, обеспечивающая минимум так называемого критерия Риссанена минимальной длины описания (Rissanen's Minimum Description Length – MDL):

$$v_{mod} = \frac{v}{(1 + \log(N)(d/N))}; \quad (4.2)$$

- при строковой переменной, равной некоторому численному значению a , выбирается структура, которая минимизирует:

$$v_{mod} = \frac{v}{(1 + a(d/N))}; \quad (4.3)$$

- v_{mod} – значение соответствующего критерия.

Выбор наилучшей структуры порядков полиномов можно осуществить с помощью более простой команды

```
>> nn=selstruc(v,0)
nn =
    7    9    1.
```

С учетом выбранной структуры модели определим вид модели ARX:

```
>> darx=arx(zdanv,nn)
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)
A(q) = 1 - 0.9601 q^-1 + 0.03384 q^-2 - 0.1035 q^-3 + 0.089 q^-4
        - 0.02827 q^-5 + 0.1383 q^-6 - 0.06615 q^-7
B(q) = 0.0009708 q^-1 + 0.003863 q^-2 + 0.06428 q^-3 + 0.06245 q^-4
        + 0.01997 q^-5 - 0.007896 q^-6 - 0.01869 q^-7 - 0.01931 q^-8
        - 0.007944 q^-9
Estimated using ARX from data set zdanv
Loss function 0.00139284 and FPE 0.00148493
Sampling interval: 0.1.
```

Следующая функция `arma` оценивает параметры ARMAX модели:

```
>> darmax=arma(zdanv,[2 2 2 1])
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + C(q)e(t)
A(q) = 1 - 1.648 q^-1 + 0.7054 q^-2
B(q) = 0.005017 q^-1 + 0.04556 q^-2
C(q) = 1 - 0.278 q^-1 - 0.4333 q^-2
Estimated using ARMAX from data set zdanv
Loss function 0.00723841 and FPE 0.00741677
Sampling interval: 0.1.
```

Функция `oe` оценивает параметры OE модели:

```
>> zoe=oe(zdanv,[2 2 1])
Discrete-time IDPOLY model: y(t) = [B(q)/F(q)]u(t) + e(t)
B(q) = -0.04799 q^-1 + 0.1169 q^-2
F(q) = 1 - 1.513 q^-1 + 0.5894 q^-2
Estimated using OE from data set zdanv
Loss function 0.0180258 and FPE 0.0183195
Sampling interval: 0.1.
```

Функция `bj` оценивает параметры модели Бокса-Дженкинса:

```
>> zbj=bj(zdanv,[2 2 2 2 1])
Discrete-time IDPOLY model: y(t) = [B(q)/F(q)]u(t) + [C(q)/D(q)]e(t)
B(q) = 0.01242 q^-1 + 0.03574 q^-2
C(q) = 1 + 0.5362 q^-1 + 0.1415 q^-2
D(q) = 1 - 0.9029 q^-1 + 0.2424 q^-2
F(q) = 1 - 1.657 q^-1 + 0.7119 q^-2
Estimated using BJ from data set zdanv
Loss function 0.00672765 and FPE 0.00695098
Sampling interval: 0.1.
```

Функция `n4sid` используется для оценивания параметров моделей переменных состояния в канонической форме при произвольном числе входов и выходов:

```
>> zn4s=n4sid(zdanv,[1:10],[1:10]),
```

где в первых квадратных скобках задается интервал порядков модели `order`, и предварительные расчеты выполняются по моделям для всех заданных порядков от 1 до 10 с выводом графика, позволяющего выбрать оптимальный порядок. После этого необходимо в командной строке MATLAB набрать этот порядок и продолжить вычисление коэффициентов модели, нажав клавишу `enter` (см. рис. 4.6).

Во вторых квадратных скобках задается так называемый дополнительный порядок, используемый алгоритмом оценивания. Должен быть больше, чем порядок, задаваемый предыдущим параметром (по умолчанию дополнительный порядок равен $(1.2 * \text{order} + 3)$). При этом выбирается оптимальный порядок без вывода соответствующего графика.

Результатом выполнения команды является вывод результатов оценивания

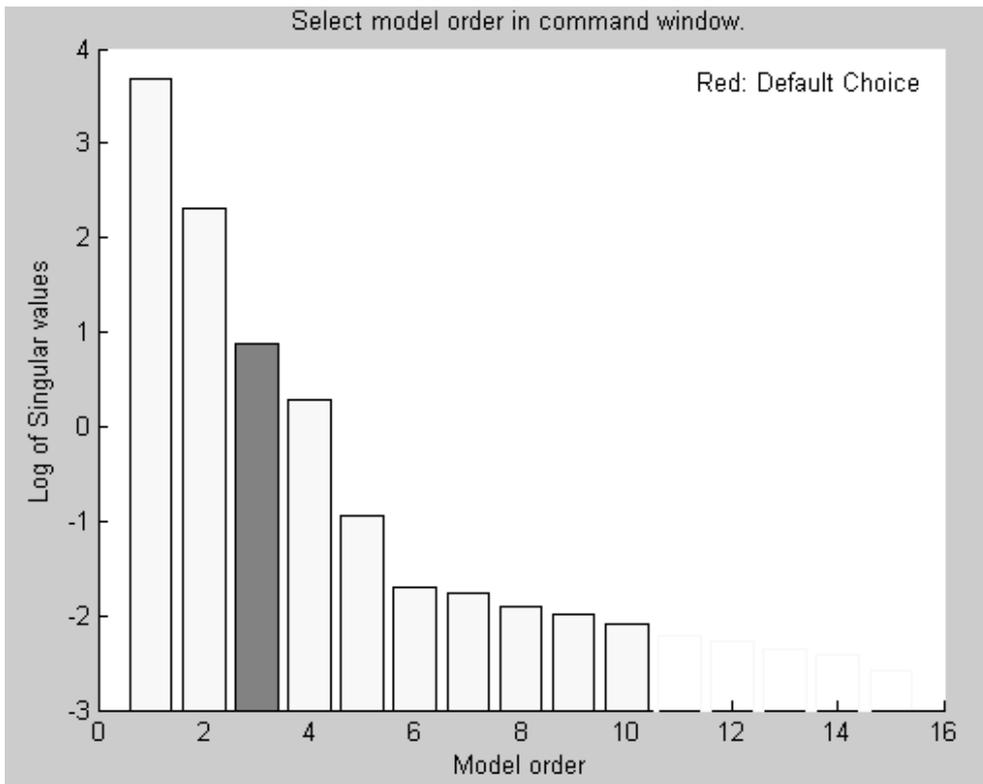


Рис. 4.6. График для выбора оптимального порядка модели

```
>> zn4s=n4sid(zdanv,[1:10],[1:10])
Warning: Input arguments must be scalar.
Select model order:('Return' gives default) 3
State-space model: x(t+Ts) = A x(t) + B u(t) + K e(t)
                   y(t) = C x(t) + D u(t) + e(t)

A =
      x1      x2      x3
x1  0.96014 -0.21598  0.062944
x2  0.24873  0.66536  0.2574
x3 -0.036067 -0.64612  0.14772

B =
      мощность
x1 -0.00029117
x2 -0.012463
x3 -0.032588

C =
      температура      x1      x2      x3
      18.093  0.078884 -0.1539

D =
      мощность
      температура  0

K =
      температура
x1  0.026294
x2 -0.00982
x3 -0.076941
```

```
x(0) =
      x1      0
      x2      0
      x3      0
```

Estimated using N4SID from data set zdanv
 Loss function 0.00213802 and FPE 0.0022164

Sampling interval: 0.1.

Функция `pem` оценивает параметры обобщенной многомерной линейной модели

```
>> zpem=pem(zdanv)
```

```
State-space model: x(t+Ts) = A x(t) + B u(t) + K e(t)
                  y(t) = C x(t) + D u(t) + e(t)
```

A =

	x1	x2	x3
x1	0.9525	-0.20939	0.049221
x2	0.25496	0.65832	0.2538
x3	-0.038767	-0.64818	0.14259

B =

	мощность
x1	-0.0002445
x2	-0.011953
x3	-0.041018

C =

	x1	x2	x3
температура	18.093	0.078583	-0.15352

D =

	мощность
температура	0

K =

	температура
x1	0.05191
x2	-0.006843
x3	-0.067671

```
x(0) =
      x1      0
      x2      0
      x3      0
```

Estimated using PEM from data set zdanv
 Loss function 0.00157651 and FPE 0.0016343

Sampling interval: 0.1.

4.5. Преобразование моделей

Следует напомнить, что полученные модели представлены в так называемом тета-формате (внутреннем виде матричных моделей) и являются дискретными.

Для преобразования моделей из тета-формата в вид удобный для даль-

нейшего использования при анализе и синтезе систем автоматизации технологических процессов в пакете System Identification Toolbox MATLAB имеются специальные функции.

Функция **th2arx** преобразует модель тета-формата в ARX модель

```
>> [A,B]=th2arx(darx)
```

A =

```
Columns 1 through 7
1.0000 -0.9601  0.0338 -0.1035  0.0890 -0.0283  0.1383
Column 8
-0.0661
```

B =

```
Columns 1 through 7
0  0.0010  0.0039  0.0643  0.0625  0.0200 -0.0079
Columns 8 through 10
-0.0187 -0.0193 -0.0079
```

Функция **th2ff** вычисляет частотные характеристики и соответствующие стандартные отклонения по модели в тета-формате. В качестве аргумента функции может выступать любая из рассмотренных ранее моделей, например **darx**

```
>> [g,phiv]=th2ff(darx)
```

IDFRD model g.

Contains Frequency Response Data for 1 output and 1 input
and SpectrumData for disturbances at 1 output
at 128 frequency points, ranging from 0.24544 rad/s to 31.416 rad/s.
Output Channels: температура
Input Channels: мощность
Estimated from data set zdanv using ARX.

IDFRD model phiv.

Contains SpectrumData for 1 signal
at 128 frequency points, ranging from 0.24544 rad/s to 31.416 rad/s.
Output Channels: температура
Estimated from data set zdanv using ARX/

Функция **th2poly** преобразует матрицу модели тета-формата в матрицы обобщенной (многомерной) линейной модели

```
>> [A,B,C,D,K,Ian,T]=th2poly(zpem)
```

A =

```
1.0000 -1.7534  1.0765 -0.2489
```

B =

```
0  0.0009  0.0012  0.0688
```

C =

```
1.0000 -0.8043  0.2732  0.1086
```

D =

```
1
```

K =

```
1
```

Ian =

```
0.0016
```

```
T =  
    0.1000
```

Здесь параметр `lan` определяет интенсивность шума наблюдений.

Функция `th2ss` преобразует тета-модель в модель для переменных состояния. В качестве аргумента функции может выступать любая из рассмотренных ранее моделей, например `darmax`

```
>> [A,B,C,D,K,x0]=th2ss(darmax)
```

```
A =  
    1.6483    1.0000  
   -0.7054         0
```

```
B =  
    0.0050  
    0.0456
```

```
C =  
     1     0
```

```
D =  
     0
```

```
K =  
    1.3703  
   -1.1387
```

```
x0 =  
     0  
     0
```

Функция `th2tf` преобразует модель тета-формата многомерного объекта в вектор передаточных функций, связанных с выбранным входом

```
>> [num,den]=th2tf(zn4s)
```

```
num =  
     0 -0.0012  0.0069  0.0632  
den =  
    1.0000 -1.7732  1.1013 -0.2554.
```

Для представления передаточной функции в виде привычном для нашего взгляда воспользуемся командой `tf` из пакета `Control System Toolbox`

```
>> zzn4s=tf(num,den,0.1)
```

```
Transfer function:  
-0.001236 z^2 + 0.006901 z + 0.06318  
-----  
z^3 - 1.773 z^2 + 1.101 z - 0.2554
```

Преобразуем дискретную модель тета-формата в непрерывную модель и представим ее в виде передаточной функции

```
>> sn4s=thd2thc(zn4s);
```

```
>> [num,den]=th2tf(sn4s);
```

```
>> sysn4s=tf(num,den)
```

```
Transfer function:  
    0.5732 s^2 - 14.16 s + 135.4  
-----  
s^3 + 13.65 s^2 + 86.04 s + 142.9
```

Для обратного преобразования непрерывной модели в дискретную существует функция `thc2thd`.

Функция `th2zp` рассчитывает нули, полюса и статические коэффициенты передачи (коэффициенты усиления) модели тета-формата `zn4s` в общем случае многомерного объекта

```
>> [zero,k]=th2zp(zn4s)
zero =
    1.0000    61.0000    21.0000    81.0000
   10.4662    8.7202    0.7842    0.0085
   -4.8834    3.3276    0.4945 + 0.2848i    0.0095 + 0.0176i
         Inf         Inf    0.4945 - 0.2848i    0.6312
k =
    1.0000
    0.9476
    0.0145
```

Информацию о нулях и полюсах модели `zn4s` можно получить с помощью команды

```
>> [zero,polus]=getzp(zero)
zero =
    10.4662
   -4.8834
polus =
    0.7842
    0.4945 + 0.2848i
    0.4945 - 0.2848i
```

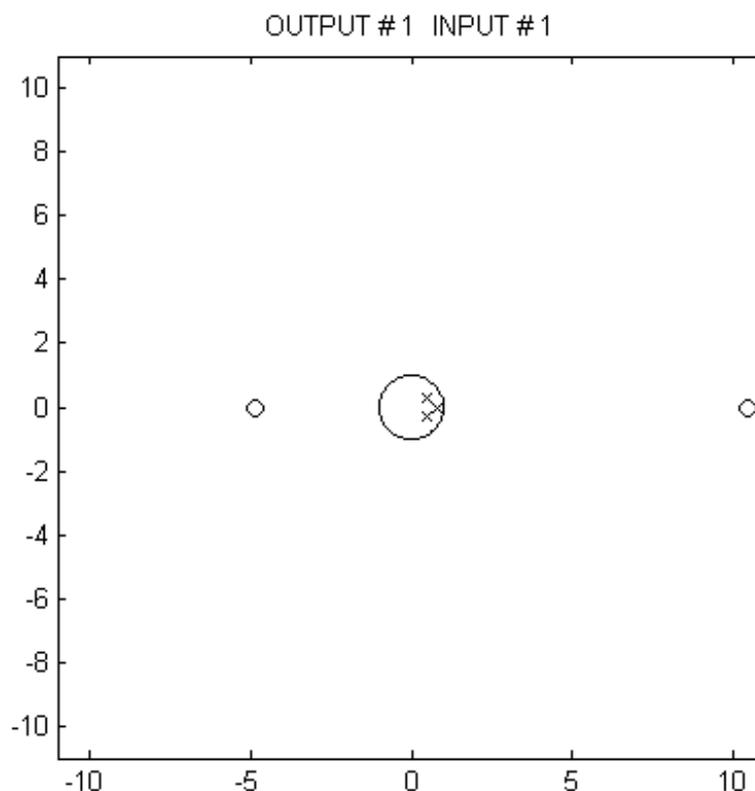


Рис. 4.7. графики нулей и полюсов модели `zn4s`

На рис. 4.7. представлен график нулей (обозначены кружком) и полюсов (обозначены крестиком) модели zn4s, который получен с помощью команды

```
>> zpplot(zpform(zero)).
```

Данные графика показывают, что модель является устойчивой, т. к. полюса модели находятся внутри окружности с радиусом, равным 1 и проходящей через точку с координатами $(-1; j0)$.

Аналогичные команды преобразования моделей имеются в пакете Control System Toolbox, которые описаны в третьей главе.

4.6. Проверка адекватности модели

Одним из важных этапов идентификации объектов автоматизации является проверка качества модели по выбранному критерию близости выхода модели и объекта, т.е проверка ее адекватности.

В пакете System Identification Toolbox MATLAB в качестве такого критерия принята оценка адекватности модели fit , которая рассчитывается по формуле

$$\text{fit} = \frac{\text{norm}(y_h - y)}{\sqrt{N}} \quad (4.4)$$

где norm – норма вектора; y_h и y – выходы модели и объекта соответственно; N – количество элементов массива данных.

Для проверки адекватности полученных ранее моделей воспользуемся функцией

```
>> compare(zdane,zn4s,zpem,zoe,zbj,darx,darmax).
```

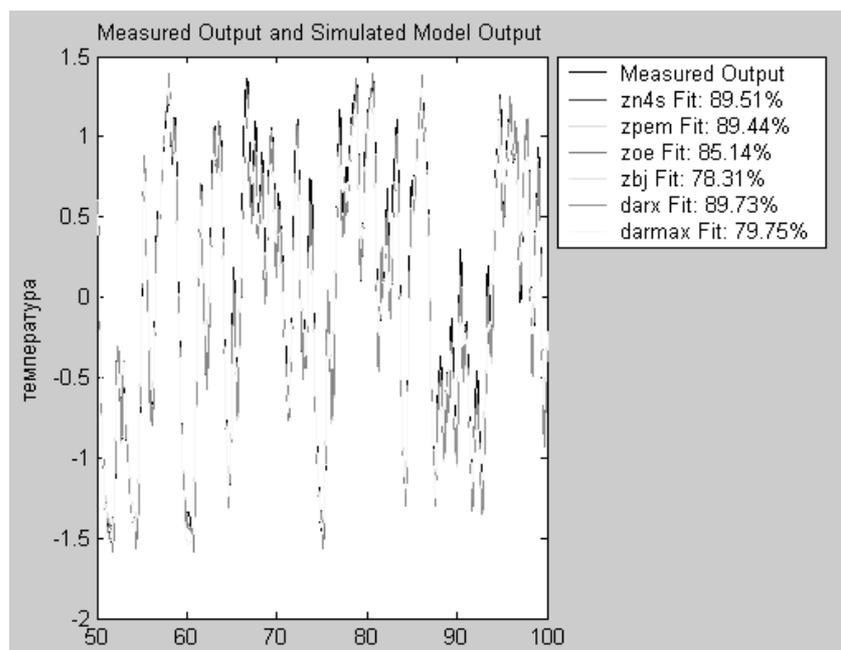


Рис. 4.7. График выходов объекта и модели

Результатом выполнения команды является вывод графика выходов объекта и построенных моделей (см. рис. 4.7). На графике цветными линиями представлены выходы полученных моделей и значения критерия адекватности, выраженного в процентах. Наилучшие показатели имеют модели `darx`, `zn4s` и `zprem`.

В пакете System Identification Toolbox MATLAB имеется возможность прогнозировать ошибку моделирования при заданном входном воздействии $u(t)$ и известной выходной координате объекта $y(t)$. Оценивание производится методом прогноза ошибки Preictive Error Method, сокращенно PEM, который заключается в следующем. Пусть модель исследуемого объекта имеет вид так называемой обобщенной линейной модели

$$y(t) = W(z) u(t) + v(t), \quad (4.5)$$

где $W(z)$ – дискретная передаточная функция любой из ранее рассмотренных моделей. При этом шум $v(t)$ может быть представлен как

$$v(t) = H(z) e(t), \quad (4.6)$$

где $e(z)$ – дискретный белый шум, который собственно и характеризует ошибку модели; $H(z)$ – некоторый полином от z , приводящий дискретный белый шум к реальным помехам при измерении выходных параметров объекта.

Из данных выражений следует, что

$$e(t) = H^{-1}(z) [y(t) - W(z) u(t)]. \quad (4.7)$$

Функция `resid` вычисляет остаточную ошибку e для заданной модели, а также r – матрицу значений автокорреляционной функции процесса $e(t)$ и значения взаимокорреляционной функции между остаточными ошибками $e(t)$ и выходами объекта автоматизации $y(t)$ вместе с соответствующими 99 %-ми доверительными коридорами. Кроме указанных значений выводятся графики данных функций. Для того, чтобы воспользоваться функцией `resid` необходимо сформировать матрицу данных

```
>> u=dan.u;
>> y=dan.y;
>> z=[y u].
```

В качестве примера сравним остаточные ошибки и соответствующие корреляционные функции для полученных моделей `darx` и `zbj`, имеющих максимальную и минимальную оценки адекватности с помощью команд

```
>> [e,r]=resid(z,darx);
>> [e,r]=resid(z,zbj);
```

Приведенные графики (см. рис. 4.8 и 4.9) характеризуют равномерное распределение остаточных ошибок во всем диапазоне изменения интервалов времени τ . Причем значения остаточных ошибок для модели `darx` практически в два раза больше, чем для модели `zbj`.

Таким образом, в ходе оценки адекватности различных моделей объекта автоматизации технологического процесса тепловой обработки материалов определены модели `darx`, `zn4s` и `zprem`, значения критерия адекватности которых максимальны и, следовательно, могут быть использованы

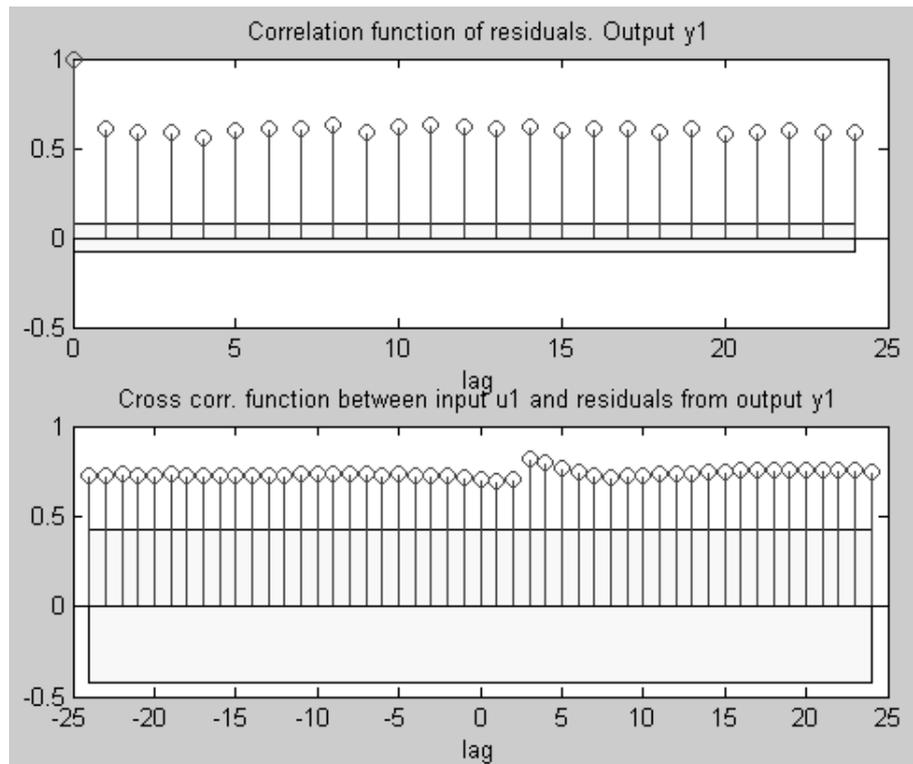


Рис. 4.8. График автокорреляционной и взаимокорреляционной функций для модели zbj

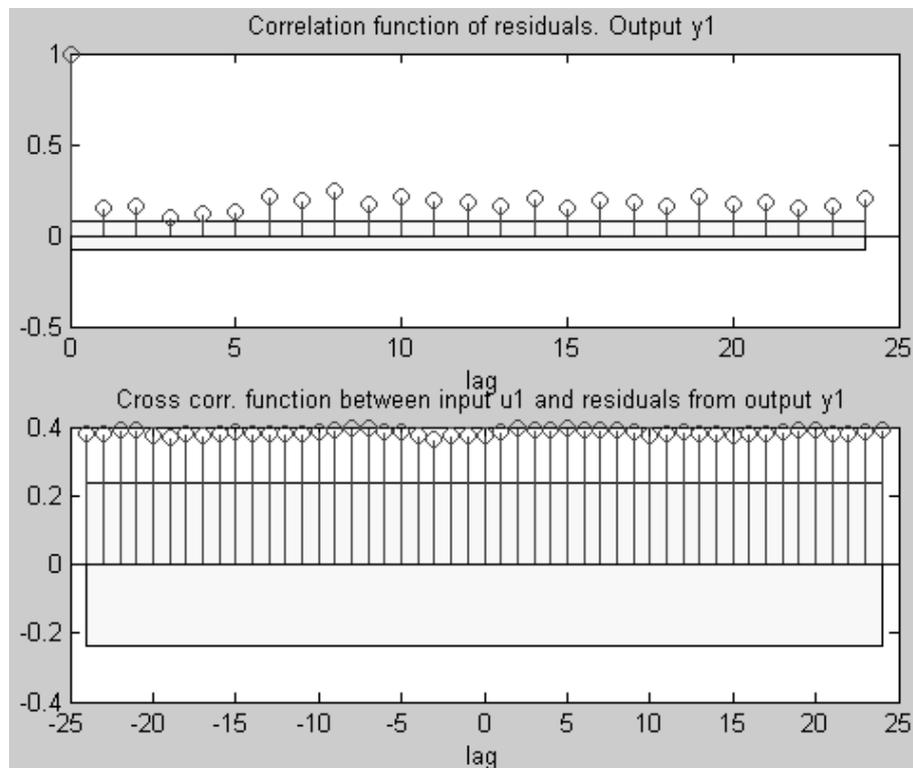


Рис. 4.9. График автокорреляционной и взаимокорреляционной функций для модели darx

в дальнейшем при анализе и синтезе систем автоматизации.

4.7. Графический интерфейс System Identification Toolbox

Приведенный выше процесс идентификации ТОУ позволяет использовать все возможности пакета System Identification Toolbox MATLAB. Однако он имеет удобный графический интерфейс, позволяющий в удобной форме, легко и просто организовывать данные и создавать модели.

Графический интерфейс пакета System Identification Toolbox MATLAB запускается из режима командной строки командой

```
>> ident
```

В результате ее выполнения появляется диалоговое окно, показанное на рис. 4.10.

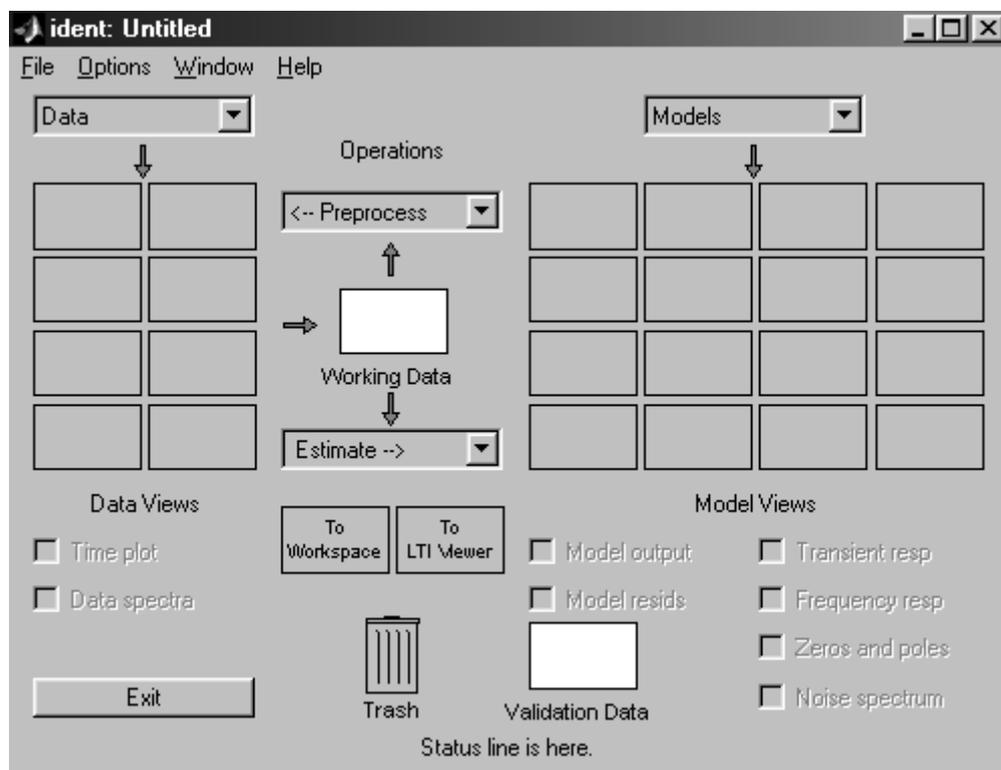


Рис. 4.10. Диалоговое окно графического интерфейса System Identification Toolbox

На начальной стадии идентификации с использованием графического интерфейса после проведения эксперимента необходимо сформировать файл данных, как это было показано в разделе 4.2. В нашем примере таким файлом является `dan`.

В верхнем левом углу окна выберем в раскрывающемся списке Data (Данные) вариант `Import`. Это приведет к открытию диалогового окна, показанного на рис. 4.11.

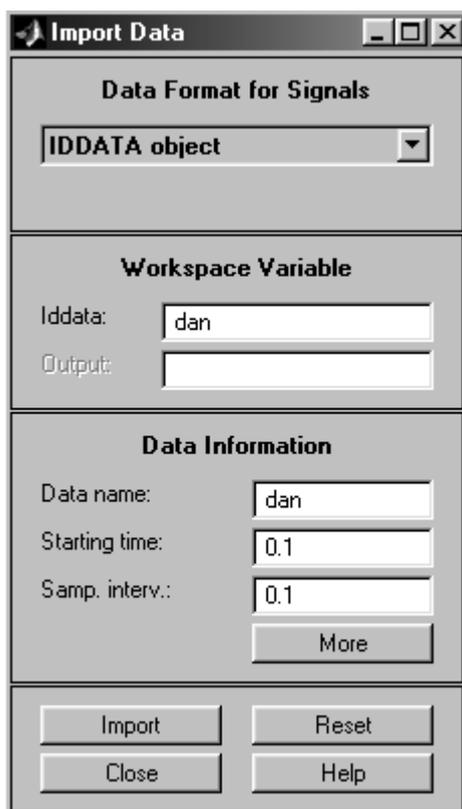


Рис. 4.11. Диалоговое окно импорта данных

В поле окна Data Format fore Signals выберем IDDATA object. В поле окна Workspace Variable в графе Iddata вводим название нашего файла данных dan. В случае выбора в поле окна Data Format fore Signals позиции Vectjr/Matrix Data необходимо в поле окна Workspace Variable в позиции Input ввести имя переменной входных данных (в нашем случае – u), а в позиции Output – имя переменной выходов ТОУ (в нашем случае – y).

В поле окна Data Information можно ввести произвольное название данных, которое в дальнейшем будет использоваться при построении моделей. В результате нажатия на клавишу More появляется дополнительное окно с информацией об исходных данных (см. рис. 4.12), хранящихся в файле dan. В случае, когда в графический интерфейс вводятся вектор входных значений u и вектор выходов объекта y, то в соответствующие поля окна импорта данных можно ввести обозначения входных и выходных переменных и их размерности.

В поле окна Notes можно ввести какой-либо поясняющий текст, что-нибудь вроде

% Первый опыт идентификации .

После всех этих действий нажмем кнопку Import. Далее нажатием кнопки Close (Закреть) закроем окно импорта данных.

Указанные действия приведут к появлению в верхнем левом углу окна

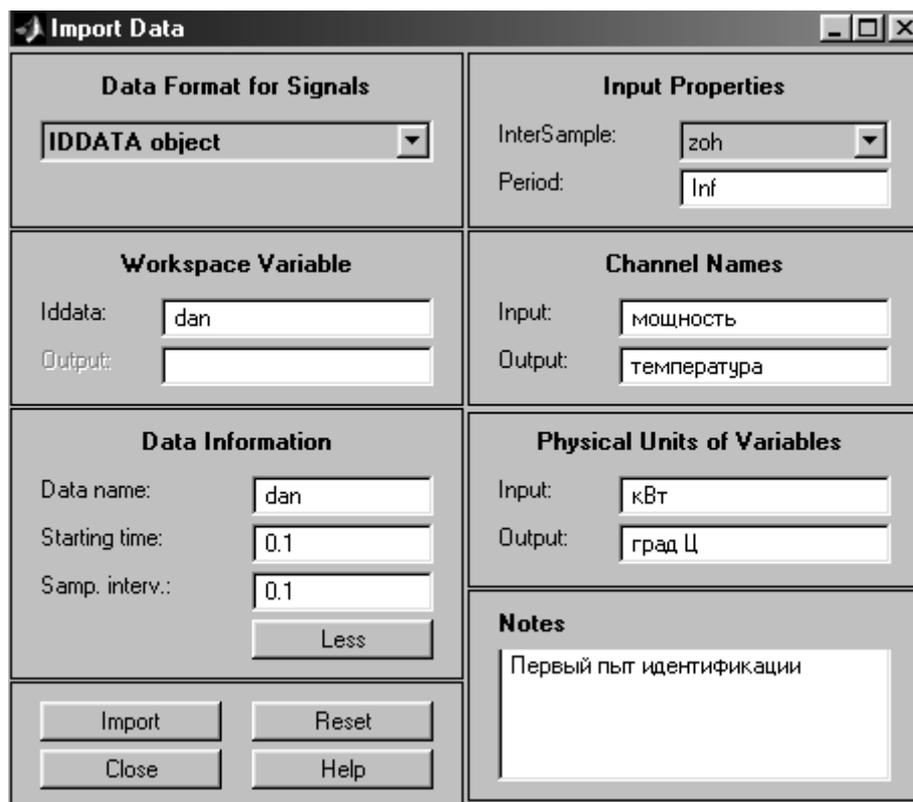


Рис. 4.12. Расширенное диалоговое окно импорта данных

интерфейса значка в виде цветной линии, сопровождаемого надписью «dan».

Это означает, что импортируемые данные введены в среду интерфейса. Сообщения об этих данных также появляются в значках Working Data (Рабочие данные) и Validation Data (Данные для проверки модели).

Проведем исследование исходных данных, для чего установим флажок Time plot (Временной график) в левой нижней части окна интерфейса. почти сразу появится графическое окно, содержащее графики сигналов «температура» и «мощность» от времени (рис. 4.13).

Можно увеличить интересующие нас места графиков, чтобы лучше их рассмотреть, - для этого необходимо мышью выделить некоторую прямоугольную область. Выделенный участок сигнала сразу же увеличится в размерах так, что будет занимать все соответствующее окно. Для возвращения к исходному масштабу необходимо дважды щелкнуть мышью на увеличенном графике. А вообще, изменение масштабов графика производится однократным щелчком мыши в области графика: левой кнопкой – для увеличения изображения, правой – для его уменьшения (такое изменение масштабов возможно, только если в меню Style (Стиль) окна Time plot выбран вариант Zoom).

Заметим, что меню окна имеет ряд команд, позволяющих достаточно подробно изучать вид входных и выходных сигналов.

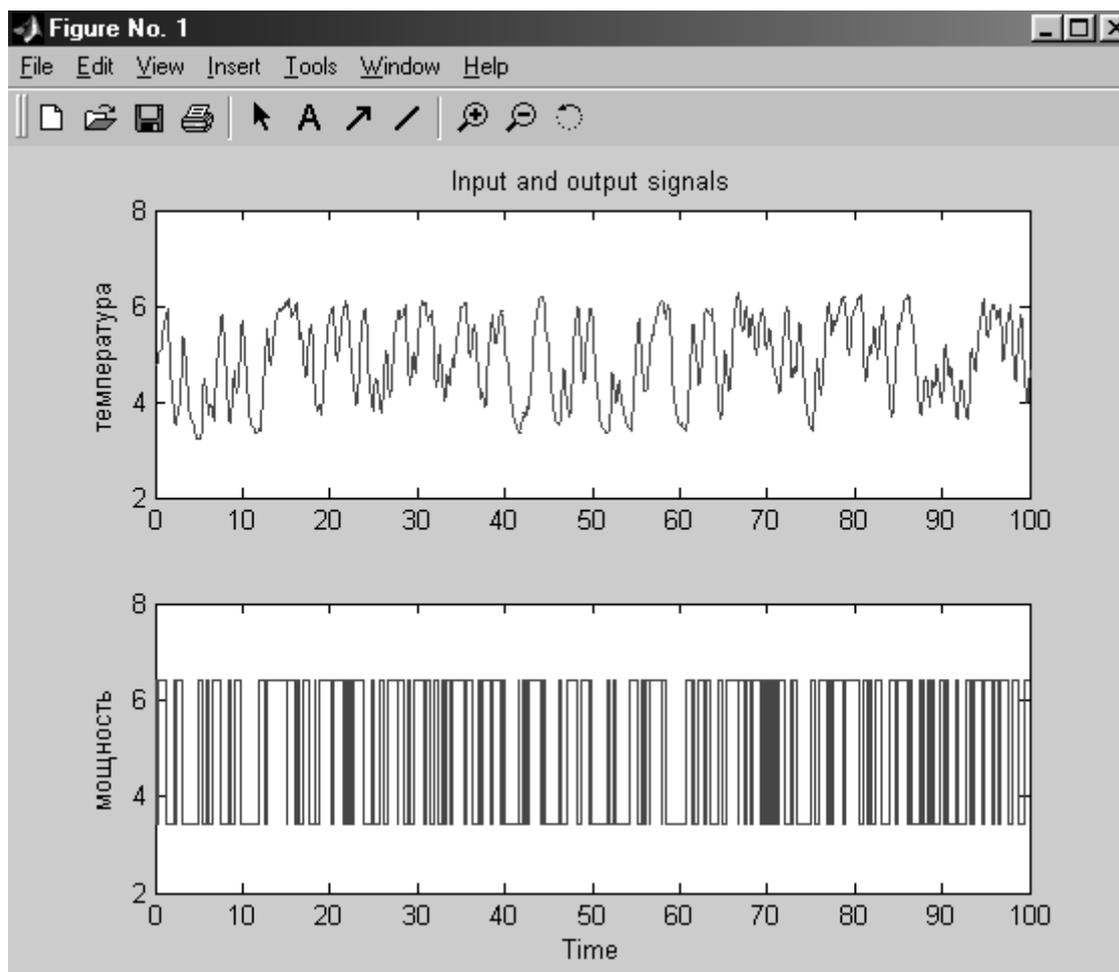


Рис. 4.13. Временные диаграммы сигналов

Закроем окно Time plot и вернемся в основное окно интерфейса.

Проведем предварительную обработку сигналов исследуемого объекта. С этой целью активизируем раскрывающийся список Preprocess (Предварительная обработка). Для предварительной обработки данных в этом списке предлагаются следующие операции:

- **Select channels...** – позволяет выбрать новые данные из подмножества данных (если таковые имеются), находящихся в рабочей области данных. Для этого необходимо выбрать соответствующие данные в предлагаемом списке и нажать кнопку Insert. Для возврата к исходным данным необходимо нажать кнопку Revert (см. рис. 4.14.а).

- **Select experiments...** – позволяет выбрать из предлагаемого списка необходимый эксперимент, если таковых было проведено несколько, и все они внесены в рабочую область данных. После этого необходимо нажать кнопку Insert. Для возврата к исходным данным необходимо нажать кнопку Revert (см. рис. 4.14.в).

- Merge experiments... – используется в случае анализа данных нескольких экспериментов, с целью группировки данных из разных экспериментов (см. рис. 4.14.б).

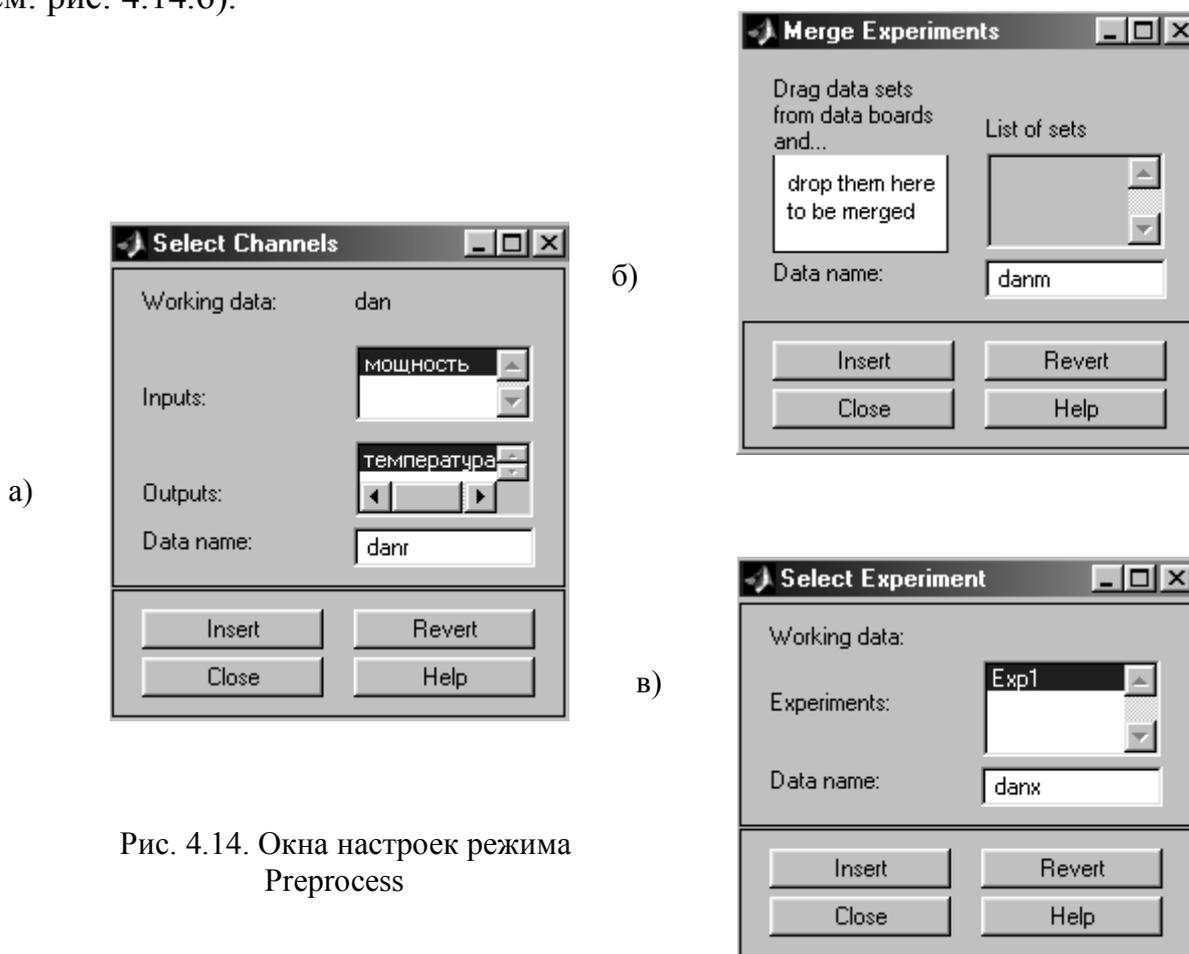


Рис. 4.14. Окна настроек режима Preprocess

В нашем случае файл dan содержит данные одного эксперимента, поэтому указанными настройками пользоваться нет необходимости.

- Remove means... (Убрать среднее) и Remove trend... (Убрать тренд) – используется при обработке данных эксперимента для удаления из массива данных среднего значения и тренда. Для построения модели эксперимента необходимо воспользоваться данными установками. Результатом операции явится появление в одном из значков Data Views (Вид данных) в левой верхней части окна интерфейса Ident информации о новых данных с новым именем dand. Можно опять активизировать окно Time plot и увидеть, что там появилось изображение двух новых сигналов, отличающихся от исходных отсутствием постоянной составляющей. Для их более удобного просмотра целесообразно воспользоваться командой меню окна Options ► Autorange (Автоматическое масштабирование). Чтобы просмотру не мешали исходные графики, можно убрать их, просто щелкнув мышью на значке с их представлением (в левой верхней части окна интерфейса, при этом сразу изменится масштаб изображения).

- Select range...(Выбор диапазона) – при выборе этой настройки появляется окно, показанное на рис. 4.15. Диапазон можно задавать в тексто-

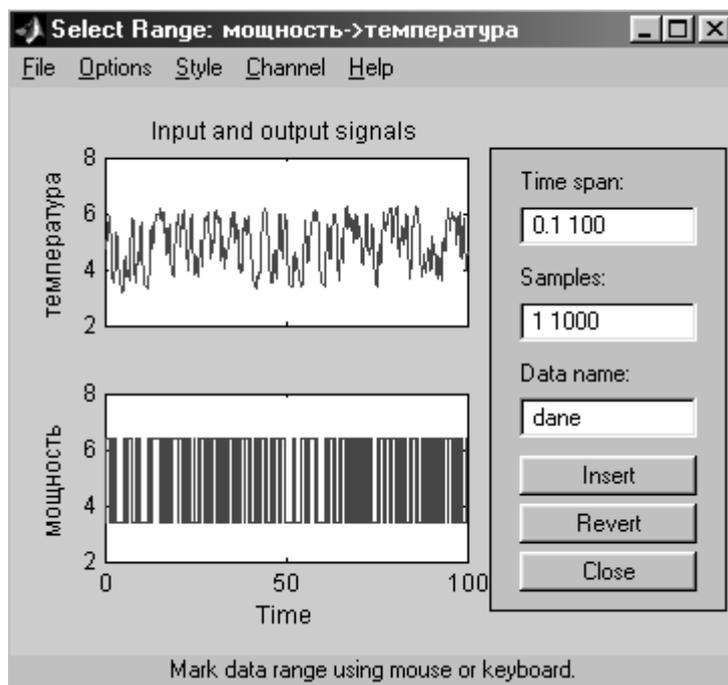


Рис. 4.15. Окно выбора диапазона

вом окошке Time span (Временной диапазон), либо с помощью мыши – выделяя прямоугольную область на графике Time plot. Любым из этих способов укажем диапазон от 1 до 50 с и нажмем кнопку Insert. Результатом проделанной операции явится появление нового значка с именем dane, символизирующего эти (усеченные) данные в окне интерфейса. Повторим операцию задания диапазона с целью формирования данных для проверки модели – например, из оставшейся части начального диапазона, в результате чего появится еще один значок данных с именем danv. Закроем окно выбора диапазона.

- Filter... – используется для фильтрации данных с помощью фильтра Баттерворта. В окне настройки указывается диапазон частот, в котором необходимо провести фильтрацию данных – либо с помощью мыши на графике Frequency Plot, либо задавая этот диапазон в текстовом окошке Range. После этого необходимо нажать кнопку Insert. Результатом проделанной операции явится появление нового значка с именем danf, символизирующего эти (отфильтрованные) данные в окне интерфейса. Закроем окно фильтра.

- Quick Start... – используется для запуска режима быстрого старта, при котором нет необходимости проводить ранее проведенные установки. При выборе этого режима установки производится:

- удаление тренда из массива экспериментальных данных с выводом значка dand в область Data Views (Вид данных) в левой верхней части окна интерфейса Ident;

- формирование усеченных массивов данных с именами dane и danv

для построения моделей.

После проведения предварительной обработки данных можно приступить к нахождению оценки модели. Перетащим с помощью мыши данные `dane` в область Working Data, а данные `danv` – в область Validation Data.

Заметим, что если на каком-то этапе работы с данными допущена ошибка и мы хотим удалить неправильные данные, необходимо перетащить значок с именем этих неправильных данных на значок Trash (Мусор) в центральной нижней части интерфейса, и они исчезнут.

Нахождение оценки модели производится, нажатием на кнопку Estimate (Оценить). В предложенном списке необходимо выбрать один из видов следующих моделей:

- Correllation model... – при выборе этого вида модели в окне Model Views (Вид моделей) появится значок с именем `imp`, символизирующий о том, что оценивание модели произведено импульсной характеристикой.
- Spectral models... – при выборе этого вида модели в окне Model Views (Вид моделей) появится значок с именем `spad`, символизирующий о том, что произведено оценивание частотных характеристик модели.
- Parametric models... – данный выбор приведет к открытию диалогово-

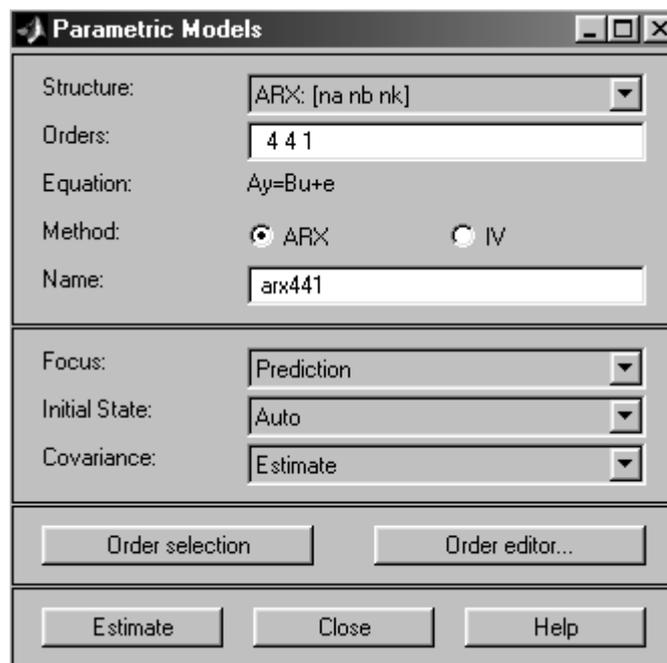


Рис. 4.16. Диалоговое окно задания структуры модели

го окна задания структуры модели (см. рис. 4.16).

По умолчанию пользователю предлагается модель типа ARX (см. выше) с параметрами $na=4$, $nb=4$, $nk=1$. Можно согласиться или не согласиться с данными значениями. В последнем случае можно изменить непосредственно в строке окна или с помощью редактора модели (Order editor), вы-

зывается нажатием соответствующей кнопки. Результатом выполнения этой операции является появление диалогового окна редактора порядка модели (см. рис. 4.17).

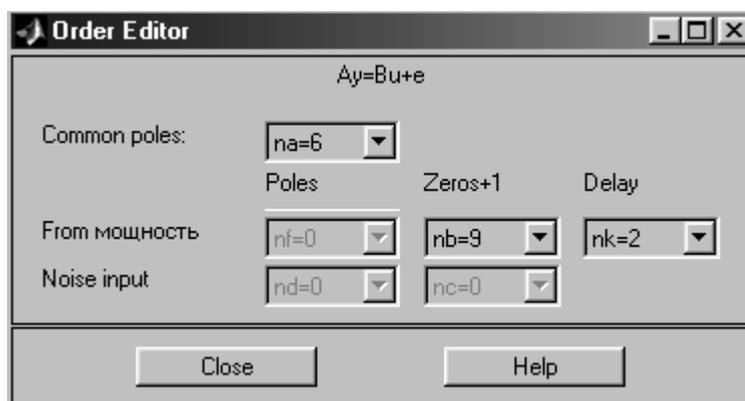


Рис. 4.18. Диалоговое окно редактора порядка модели

В области Common poles необходимо задать диапазоны изменения параметров, например $na=1:10$, $nb=1:10$, $nk=1:10$. Далее появится окно выбора структуры модели, как на рис. 4.5 и после нажатия кнопки Estimate в окне интерфейса Model Views (Вид моделей) появится значок с именем arx692, символизирующий о том, что получена модель ARX с оптимальными значениями порядков полиномов.

Аналогичным образом можно получить другие параметрические модели, воспользовавшись раскрывающимся списком в верхней части окна задания структуры модели (возможный выбор – модели типа ARX, ARMAX, OE, BJ, State Space и модель, задаваемая пользователем). Оценивание каждой модели сопровождается появлением соответствующего значка окне Model Views интерфейса.

Общий вид окна интерфейса с результатами построения моделей представлен на рис. 4.19.

Для анализа моделей графический интерфейс пакета System Identification Toolbox MATLAB представляет весьма значительные возможности. Так для сравнения параметрических моделей достаточно активизировать графическое окно Model output (соответствующий флажок расположен в средней части окна интерфейса). Вид этого окна для выбранных моделей такой же, как на рис. 4.7.

Активизация окна Model resides приведет к построению автокорреляционной и взаимокорреляционной функций с 99% доверительными коридорами, графики которых имеют аналогичный вид с рис. 4.8 и 4.9.

Для получения переходных характеристик моделей и их импульсных характеристик достаточно активизировать окно Transient resp, в котором будут отражены графики переходных процессов тех моделей, чьи значки

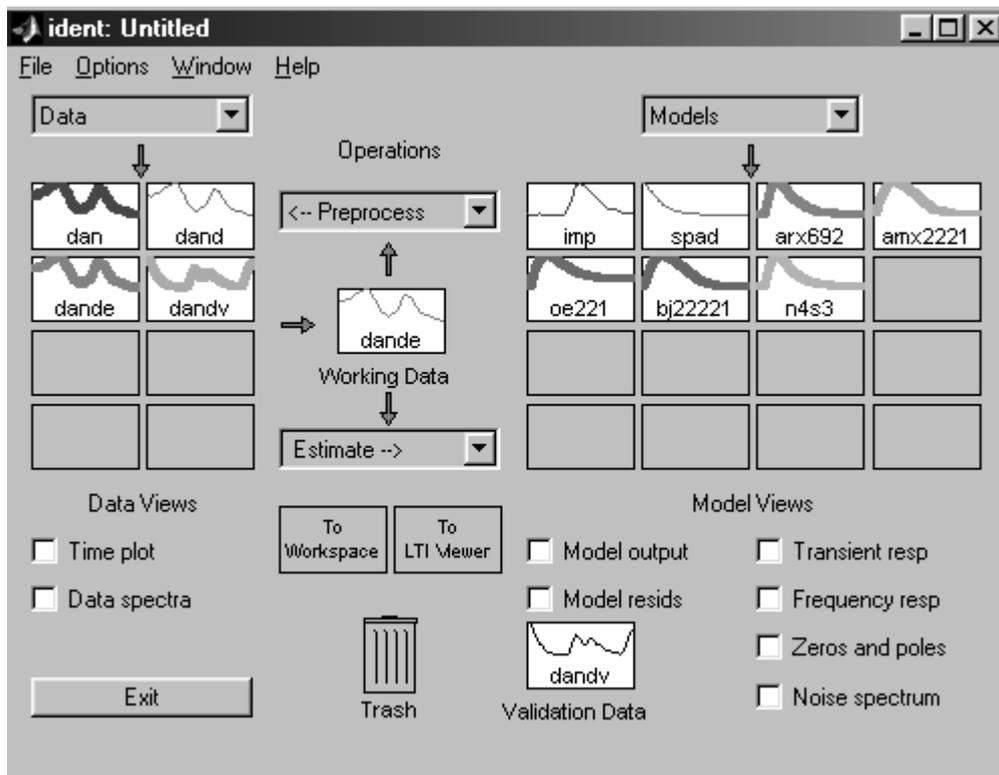


Рис. 4.19. Окно графического интерфейса с результатами построения моделей

активизированы в окне графического интерфейса (активизированной считается та модель, у которой значок имеет широкую цветную линию). Вид

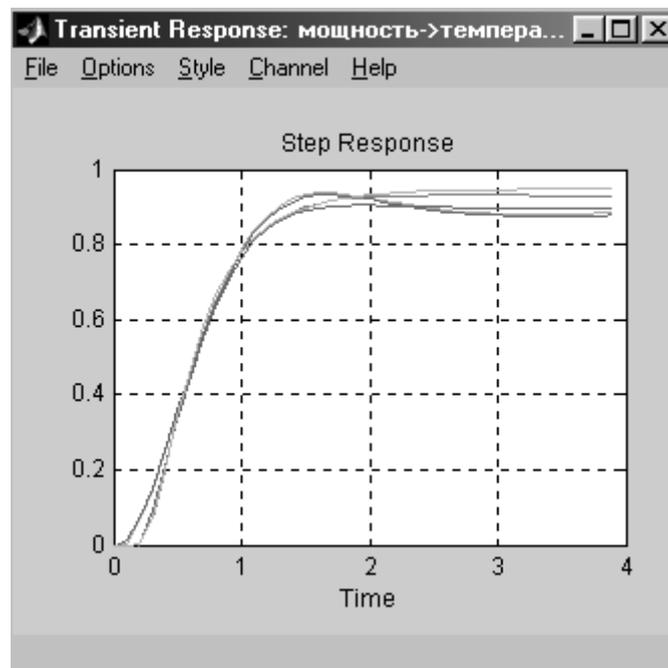


Рис. 4.20. Графики переходных процессов полученных моделей

графиков переходных процессов показан на рис. 4.20. Если активизировать команду меню Options ► Impulse response, то на указанном графике появятся импульсные характеристики моделей.

Частотные характеристики моделей можно получить, если активизировать окно Frequency resp.

Кроме того, графический интерфейс позволяет получить графики нулей и полюсов моделей (окно Zeros and poles), а также графики спектров шумов (окно Noise spectrum) полученных моделей.

Возможности анализа моделей значительно расширяются использованием возможностей программы LTI Viewer (детальное описание которой приведено в []). Для этого достаточно перетащить значок анализируемой модели в прямоугольник To LTI Viewer.

Сохранение построенных моделей производится в два этапа:

- сначала модель вводится в рабочее окружение системы MATLAB (перетаскиванием значка модели в прямоугольник с именем To Workspace в центре рабочего окна интерфейса), при этом модель будет фигурировать в рабочем пространстве MATLAB под тем же именем, что и в среде интерфейса;

- затем модель сохраняется командой сохранения в режиме командной строки save (как любая переменная MATLAB).

Отметим, что можно сохранить все рабочее пространство интерфейса (при его закрытии даже появится соответствующая подсказка) в виде файла с расширением (по умолчанию) sid, при этом в следующем сеансе работы можно загрузить все полученные результаты.

4.8. Анализ модели ТОУ

Для анализа модели ТОУ возьмем модель zn4s, имеющую один из наилучших показателей адекватности. В разделе 4.5 были получены различные виды этой модели:

- zn4s – дискретная модель тета-формата (LTI-формата)

$$\text{State-space model: } \begin{aligned} x(t+Ts) &= A x(t) + B u(t) + K e(t) \\ y(t) &= C x(t) + D u(t) + e(t) \end{aligned}$$

A =

	x1	x2	x3
x1	0.96014	-0.21598	0.062944
x2	0.24873	0.66536	0.2574
x3	-0.036067	-0.64612	0.14772

B =

	МОЩНОСТЬ
x1	-0.00029117
x2	-0.012463
x3	-0.032588

C =

	x1	x2	x3
температура	18.093	0.078884	-0.1539

D =

	мощность
температура	0

K =

температура	
x1	0.026294
x2	-0.00982
x3	-0.076941

x(0) =

x1	0
x2	0
x3	0

Estimated using N4SID from data set zdanv
 Loss function 0.00213802 and FPE 0.0022164
 Sampling interval: 0.1,

где выражение Sampling interval (интервал дискретизации) указывает на то, что модель представлена в дискретном виде;

• sn4s – непрерывная модель тета-формата (LTI-формата)

State-space model: $dx/dt = A x(t) + B u(t) + K e(t)$
 $y(t) = C x(t) + D u(t) + e(t)$

A =

	x1	x2	x3
x1	-0.19842	-1.6567	1.7776
x2	2.6343	-1.0547	5.106
x3	1.474	-13.427	-12.397

B =

	мощность
x1	0.02622
x2	-0.022958
x3	-0.65355

C =

	x1	x2	x3
температура	18.093	0.078884	-0.1539

D =

	мощность
температура	0

K =

температура	
x1	0.34674
x2	0.10592
x3	-1.4143

x(0) =

x1	0
x2	0
x3	0

Estimated using N4SID from data set zdanv
Loss function 0.00213802 and FPE 0.0022164;

Как видно в представленных моделях значения коэффициентов матриц А, В, С, D, К различны. Это объясняется тем, что для непрерывной модели произведено Z-преобразование с целью получения дискретной модели;

- zzn4s – дискретная модель в виде передаточной функции

$$W(z) = \frac{-0.001236z^2 + 0.006901z + 0.06318}{z^3 - 1.7732z^2 + 1.101z - 0.2554}; \quad (4.8)$$

- sysn4s – непрерывная модель в виде передаточной функции

$$W(s) = \frac{0.5732s^2 - 14.16s + 135.4}{s^3 + 13.65s^2 + 86.04s + 142.9}. \quad (4.9)$$

Следует напомнить, что приведенные виды являются одной и той же моделью, записанной в разных формах и форматах. Проанализируем динамические характеристики модели. Построим переходную характеристику ТОО для дискретной и непрерывной моделей и определим основные показатели переходного процесса. Для этого можно воспользоваться командой `step(zn4s,sn4s)`, либо командой `plot(zn4s,sn4s)`. Различие заключается в том, что в последнем случае представляется возможность использовать все достоинства LTI view (см. рис. 4.21).

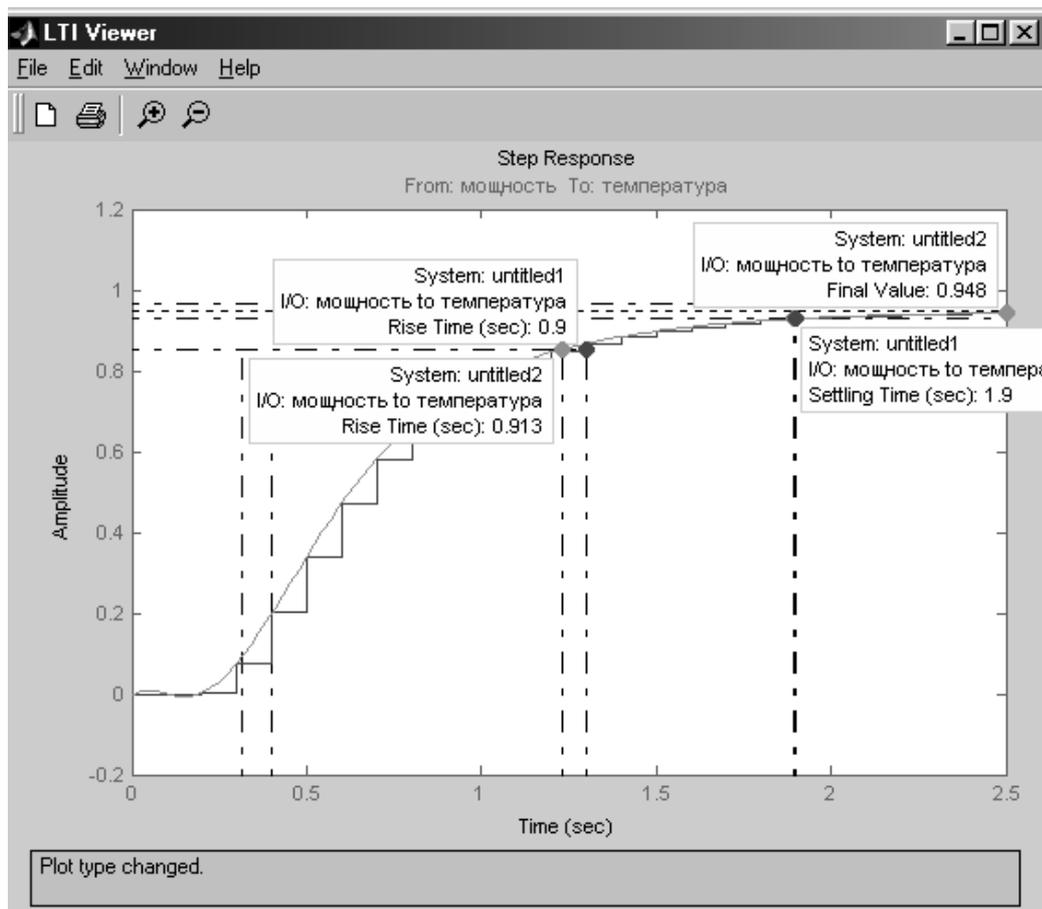


Рис. 4.21. Графики переходных процессов модели zn4s

На графиках переходных процессов ступенчатой линией представлен переходной процесс дискретной модели, а сплошной линией – непрерывной модели. Кроме того, в поле графика указаны основные характеристики переходного процесса:

- время нарастания переходного процесса (Rise time) – 0,913 с для непрерывной модели и 0,9 с для дискретной модели;
- время регулирования (Setting time) – 1,9 с для обеих моделей;
- установившееся значение выходной координаты (Final value) – 0,948 для обеих моделей.

Для построения импульсной характеристики моделей необходимо воспользоваться командой `impulse(zn4s,sn4s)`, либо, щелкнув правой кнопкой мыши в поле графика LTI view, выбрать опцию Plot Types ► Impulse (см. рис. 4.22).

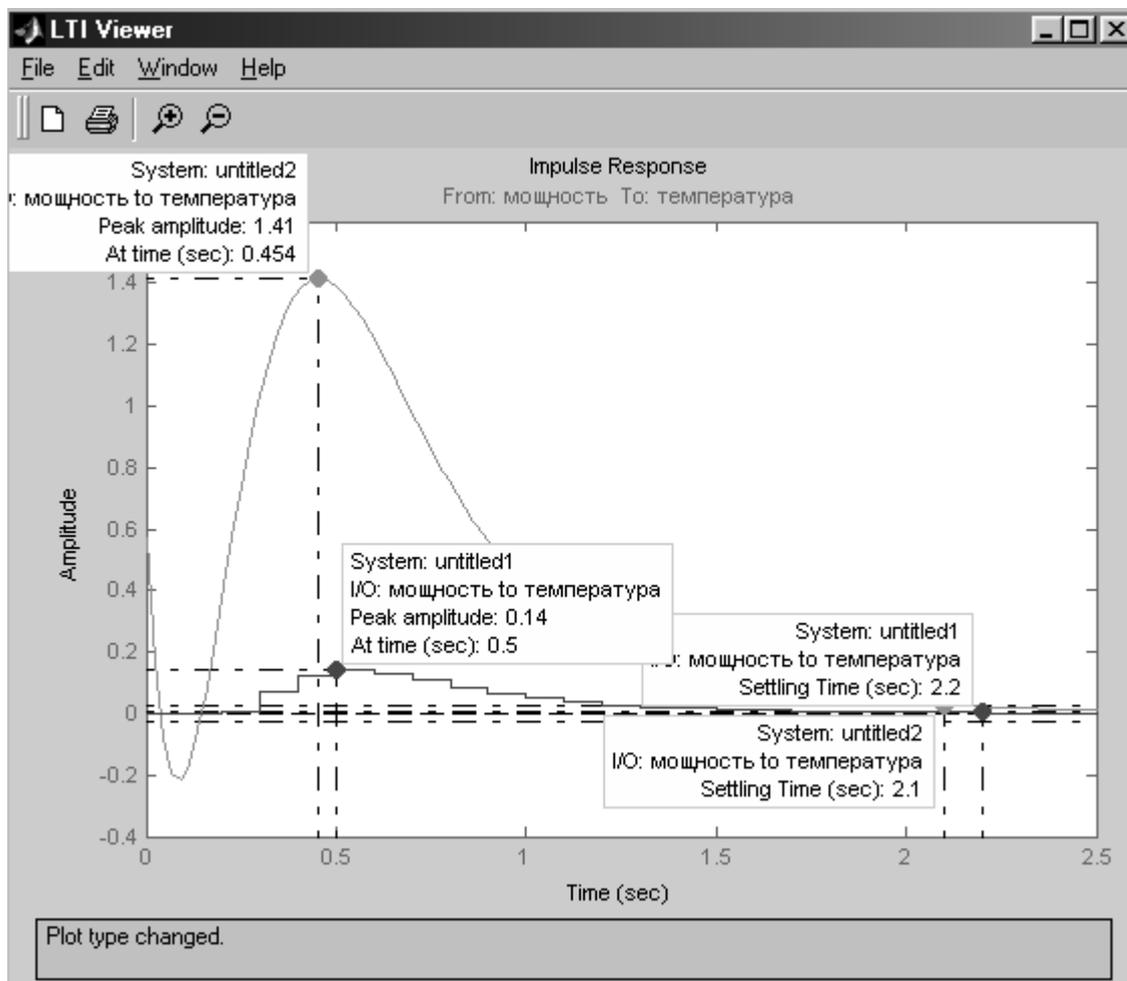


Рис. 4.22. Графики импульсной характеристики

Основными характеристиками модели ТОУ при подаче на вход единичного импульсного воздействия являются:

- пиковая амплитуда (Peak amplitude) составляет для дискретной модели 0,14, а для непрерывной – 1,41.

- время регулирования составляет для дискретной модели 2,2 с, а для непрерывной модели – 2,1 с.

Определим статический коэффициент усиления модели ТОУ с помощью команды

```
>> k=dcgain(sysn4s)
k =
    0.9476.
```

Определим частотные характеристики моделей с помощью команды `bode (zn4s,sn4s)` либо, щелкнув правой кнопкой мыши в поле графика LTI view, выбрать опцию `Plot Types ► Bode` (см. рис. 4.23).

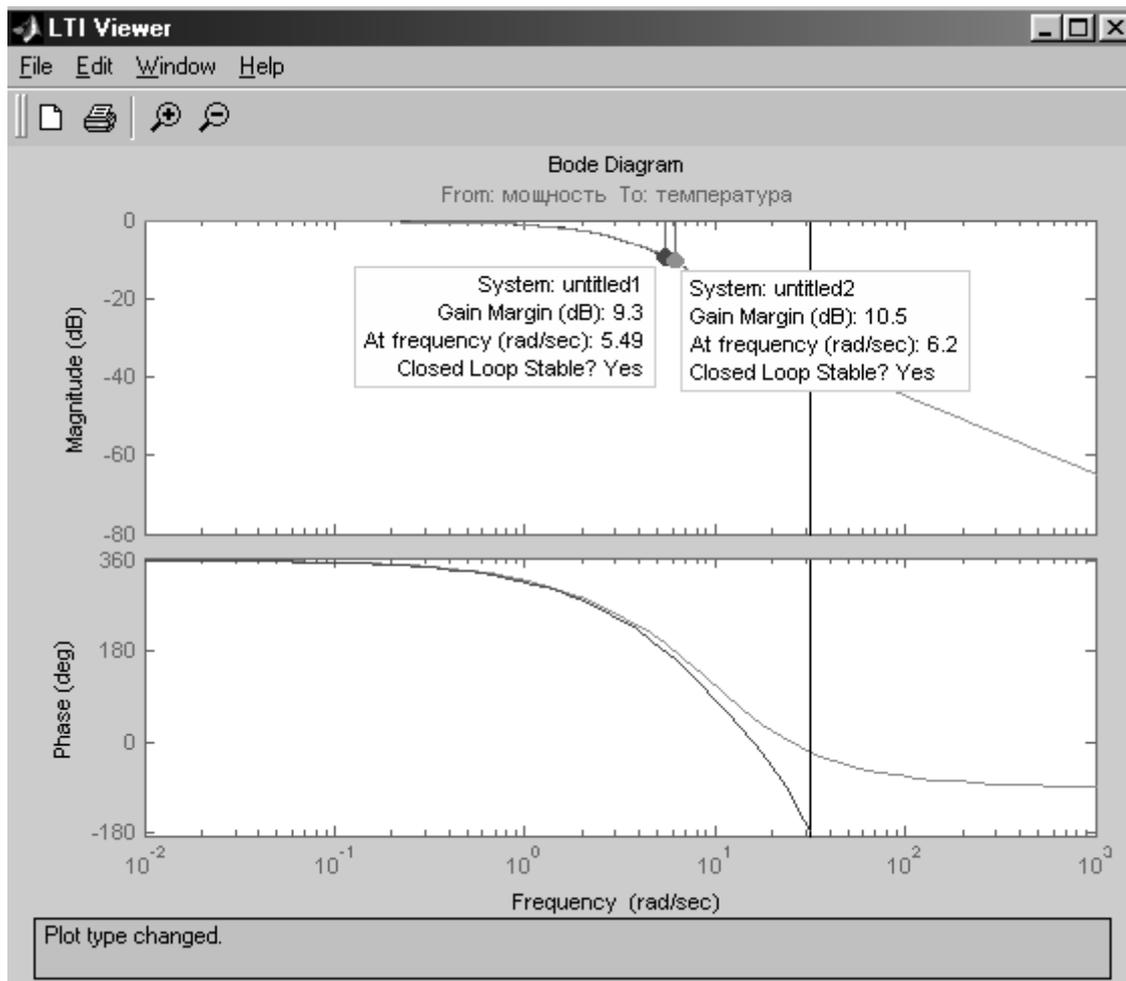


Рис. 4.23. Частотные характеристики моделей

На графиках частотных характеристик указаны значения запасов устойчивости по амплитуде (Gain Margin), которые для дискретной модели составляет 9,3 dB, а для непрерывной модели – 10,5 dB.

Значения запасов устойчивости можно определить также и в режиме командной строки MATLAB с помощью команд:

```
>> [Gm,Pm,Wcg,Wcp]=margin(sysn4s) – для непрерывной модели:
```

```

Gm =
    3.3642
Pm =
    Inf
Wcg =
    6.1977
Wcp =
    NaN.
>> [Gm,Pm,Wcg,Wcp]=margin(zzn4s) – для дискретной модели:
Gm =
    2.9158
Pm =
    Inf
Wcg =
    5.4941
Wcp =
    NaN,

```

где G_m – запас устойчивости по амплитуде в натуральных величинах на частоте W_{cg} , P_m – запас устойчивости по фазе на частоте W_{cp} .

Для определения запасов устойчивости в логарифмическом масштабе необходимо выполнить следующие операции:

```

>> Gmlog=20*log10(Gm) – для дискретной модели:
Gmlog =
    9.2951
>> Gmlog=20*log10(Gm) – для непрерывной модели:
Gmlog =
    10.5377.

```

Как видно, определение запасов устойчивости последним способом позволяет значительно точнее вычислять эти значения, чем на графиках частотных характеристик.

Анализ частотных характеристик показывает, что модели `zzn4s` и `sysn4s` являются устойчивыми с соответствующими запасами устойчивости по амплитуде. Запас устойчивости по фазе равен бесконечности.

Этот вывод подтверждается так же комплексной амплитудно-фазовой характеристикой АФХ, которая в зарубежной литературе называется диаграммой Найквиста (см. рис. 4.24), так как годограф АФХ не пересекает точку комплексной плоскости с координатами $-1, j0$.

Для построения АФХ необходимо воспользоваться командой `nyquist(zzn4s,sysn4s)`, либо, щелкнув правой кнопкой мыши в поле графика LTI view, выбрать опцию Plot Types ► Nyquist.

Определить устойчивость моделей можно с помощью карты нулей и полюсов по расположению нулей моделей относительно окружности с единичным радиусом на комплексной плоскости, как это было показано на рис. 4.7.

Построить карту нулей и полюсов моделей можно так же с помощью команды `pzmap(zzn4s,sysn4s)`, либо – `pzmap(zn4s,sn4s)`.

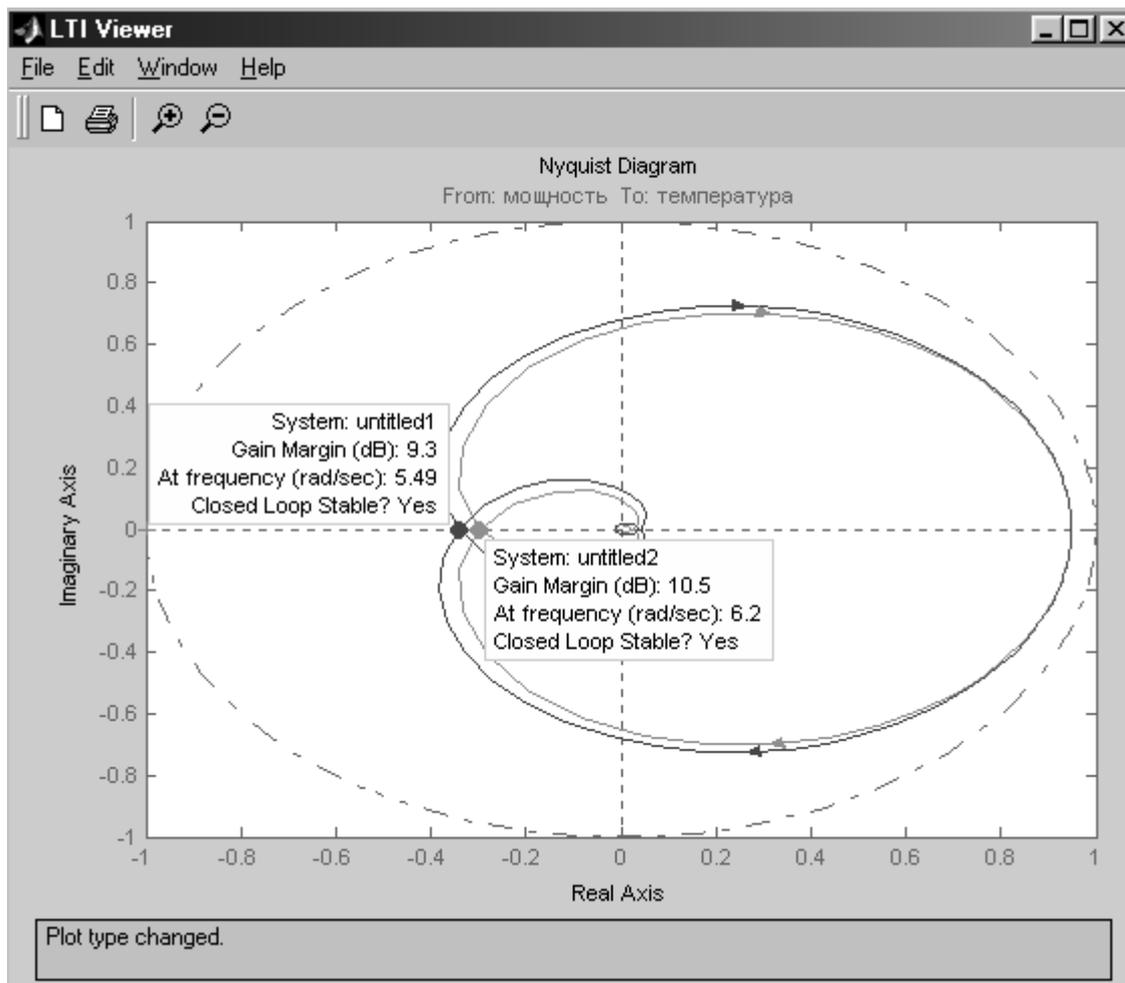


Рис. 4.23. Годограф АФХ с указанием значений запасов устойчивости для непрерывной и дискретной моделей

Следует напомнить, что модель объекта автоматизации рассматривается нами в виде, показанном на рис. 1.1, при этом выходной параметр объекта автоматизации (в нашем случае температура) $y(t)$ складывается из двух составляющих: теоретический выход объекта $y(t)$ и аддитивная помеха $e(t)$, вызванная влиянием внешних факторов на объект автоматизации. Последняя составляющая является чисто случайной величиной и характеризуется статистическими параметрами. В ходе идентификации в разделе 4.6 нами были получены зависимости, определяющие составляющую $e(t)$ и корреляционные зависимости между $e(t)$ и выходом объекта $y'(t)$.

Для наглядности построим график изменения $e(t)$ и определим основные статистические характеристики помехи с помощью команды plot (e) (см. рис.4.24).

Для получения статистических характеристик необходимо в строке меню графика в позиции Tools выбрать опцию Data statistics. Результатом выполнения команды явится окно, в котором будут указаны основные статистические характеристики случайного процесса изменения во времени $e(t)$

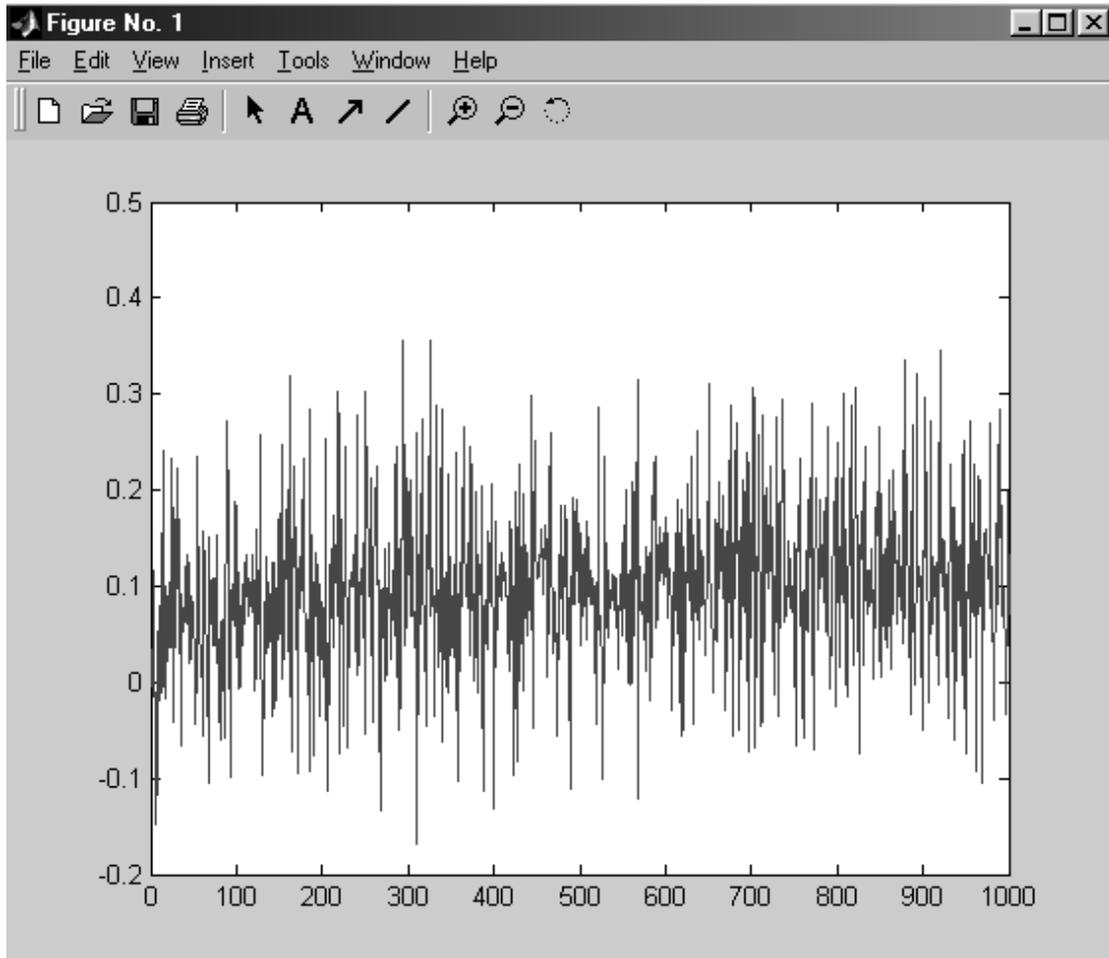


Рис. 4.24. График аддитивной помехи $e(t)$

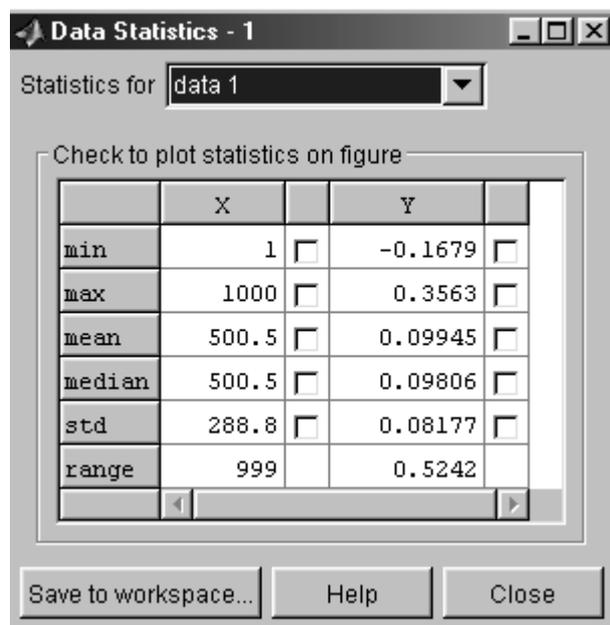


Рис. 4.25. Статистические характеристики $e(t)$

(см. рис. 4.25), к которым относятся:

- min и max – минимальное и максимальное значения помехи. Для нашего случая – 0,1976 и 0,3563 соответственно;
- mean – арифметическое среднее значение (0,09945);
- median – медиана процесса (0,09806);
- std – среднеквадратическое отклонение (0,08177);
- range – диапазон изменения помехи от минимального до максимального значения (0,5242).

Во всех случаях размерность аддитивной помехи такая же, как и выходная величина объекта автоматизации – °С.

Полученные статистические характеристики помехи могут быть полезны в дальнейшем при синтезе системы автоматического регулирования температуры теплового объекта автоматизации.

Для решения задач анализа и синтеза систем управления важно знать ответ на другой не менее важный вопрос, чем полученные временные, частотные и статистические характеристики: обладает ли объект свойством управляемости в смысле возможности его перевода из заданной начальной точки (или области) в заданную конечную точку (или область)? До второй половины девятнадцатого столетия проблема управляемости – проблема установления обладания объектом свойством управляемости решалась чисто интуитивно на основе инженерных знаний и опыта. В настоящее время, с развитием метода переменных состояния стало возможным строгое определение свойства управляемости и установление критерия управляемости.

Решение проблемы управляемости основано на анализе уравнений переменных состояния вида 3.8 или 3.9 и формулируется следующим образом: *объект называется вполне управляемым, если выбором управляющего воздействия $u(t)$ на интервале времени $[t_0, t_k]$ можно перевести его из любого начального состояния $y(t_0)$ в произвольное заранее заданное конечное состояние $y(t_k)$.*

Критерием управляемости линейных стационарных объектов является условие: *для того чтобы объект был вполне управляем, необходимо и достаточно, чтобы ранг матрицы управляемости*

$$M_U = (B \ AB \ A^2B \ \dots \ A^{n-1}B) \quad (4.10)$$

равнялся размерности вектора состояний n

$$\text{rang } M_U = n. \quad (4.11)$$

В пакете Control System Toolbox имеется функция ctrb, формирующая матрицу управляемости в пространстве состояний. Для того, чтобы воспользоваться этой функцией необходимо вычислить матрицы A , B , C , D с помощью команды:

```
>> [A,B,C,D]=ssdata(sn4s)
```

```

A =
    -0.1984  -1.6567  1.7776
     2.6343  -1.0547  5.1060
     1.4740 -13.4267 -12.3967
B =
     0.0262
    -0.0230
    -0.6535
C =
    18.0931  0.0789 -0.1539
D =
     0.

```

Следует обратить внимание, что для расчета матриц используется непрерывная модель, так как дискретная модель имеет другие значения, а в критерии управляемости используются матрицы линейных непрерывных стационарных объектов.

Вычислим матрицу управляемости:

```

>> Mu=ctrb(A,B)
Mu =
     0.0262  -1.1289  20.6164
    -0.0230  -3.2437  43.5862
    -0.6535   8.4487 -62.8481.

```

Определим ранг матрицы управляемости:

```

>> n=rank(Mu)
n =
     3.

```

Таким образом, для исследуемой модели объекта размерность вектора состояний, определяемая размером матриц A и B равна трем и ранг матрицы управляемости M_U также равен трем, что позволяет сделать вывод о том, что объект автоматизации является вполне управляемым, т.е. для него имеется такое управляющее воздействие $u(t)$, которое способно перевести на интервале времени $[t_0, t_k]$ объект из любого начального состояния $y(t_0)$ в произвольное заранее заданное конечное состояние $y(t_k)$.

При синтезе оптимальных систем с обратной связью сами управления получаются как функции от фазовых координат. В общем случае фазовые координаты являются абстрактными величинами и не могут быть исследованы. Поддается измерению (наблюдению) вектор $y = (y_1, \dots, y_k)^T$, который обычно называют выходным вектором или выходной переменной, а его координаты – выходными величинами. Выходная переменная функционально связана с фазовыми координатами, и для реализации управления с обратной связью необходимо определить фазовые координаты по измеренным значениям выходной переменной. В связи с этим возникает проблема наблюдаемости, заключающаяся в установлении возможности состояния определения состояния объекта (фазового вектора) по измеренным значениям выходной переменной на некотором интервале.

Решение проблемы наблюдаемости основано на анализе уравнений переменных состояния вида 3.8 или 3.9 и формулируется следующим образом: объект называется вполне наблюдаемым, если по реакции $y(t_1)$ на выходе объекта, на интервале времени $[t_0, t_1]$ при заданном управляющем воздействии $u(t)$ можно определить начальное состояние вектора переменных состояния $x(t)$, являющихся фазовыми координатами объекта.

Критерием наблюдаемости линейных стационарных объектов является условие: для того, чтобы объект был вполне наблюдаемым, необходимо и достаточно, чтобы ранг матрицы наблюдаемости

$$M_Y = (C^T \ A^T C^T \ (A^T)^2 C^T \ \dots \ (A^T)^{n-1} C) \quad (4.12)$$

равнялся размерности вектора состояния

$$n = \text{rang } M_Y. \quad (4.13)$$

Определим матрицу наблюдаемости и ее ранг с помощью функций пакета Control System Toolbox:

```
>> My=obsv(A,C)
```

```
My =
```

```
    18.0931    0.0789   -0.1539
   -3.6091  -27.9915   34.4730
   -22.2084 -427.3557 -576.6903
```

```
>> n=rank(My)
```

```
n =
```

```
    3.
```

Таким образом, для исследуемой модели объекта размерность вектора состояний, определяемая размером матриц A и C равна трем и ранг матрицы наблюдаемости M_Y также равен трем, что позволяет сделать вывод о том, что объект автоматизации является вполне наблюдаемым, т.е. для него всегда можно определить по значениям выходной величины $y(t)$ вектор переменных состояния, необходимый для синтеза системы управления.