

Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Кафедра «Информационно-измерительная техника»

681.5(07)
Н48

С.Г. Некрасов, Р.А. Хажиев, Н.В. Николайзин

ИДЕНТИФИКАЦИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ
С ИНСТРУМЕНТАМИ SYSTEM IDENTIFICATION TOOLBOX
В СИСТЕМЕ MATLAB

Учебное пособие к лабораторным работам

Челябинск
Издательский центр ЮУрГУ
2013

УДК [681.51.015.026](076.5)
Н48

*Одобрено
учебно-методической комиссией
приборостроительного факультета.*

Рецензенты:

В.С. Жабреев, В.И. Поклонский

Н48 Некрасов, С.Г.
Идентификация динамических объектов с инструментами System Identification Toolbox в системе Matlab: учебное пособие к лабораторным работам/ С.Г. Некрасов, Р.А. Хажиев, Н.В. Николайзин. – Челябинск: Издательский центр ЮУрГУ, 2013. –108с.

В учебном пособии содержатся необходимые материалы для выполнения лабораторных работ и контрольные вопросы по курсу «Диагностика и идентификация систем». Учебное пособие предназначено для подготовки магистра техники и технологии направления 200100 «Приборостроение» магистерской программы «Измерительные информационные технологии».

УДК 681.51.015.026(076.5)

©Издательский центр ЮУрГУ, 2013

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1. НЕОБХОДИМЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	
1.1. Задача идентификации	7
1.2. Математическое описание моделей объекта управления	8
1.3. Параметрические модели в System Identification Toolbox	10
2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕГРЕССИОННЫХ И АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДАННЫХ	
2.1. Цель работы	
2.2. Постановка задачи	
2.3. Справочные данные по Matlab	14
2.4. Пример выполнения лабораторной работы	
2.4.1. Построение полиномиальной регрессионной модели	15
2.4.2. Построение многомерной регрессии	21
2.4.3. Построение авторегрессионной модели	22
2.5. Задания на лабораторную работу	28
3. НЕПАРАМЕТРИЧЕСКАЯ И ПАРАМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ЛИНЕЙНОЙ СИСТЕМЫ ХАРАКТЕРИЗАЦИИ ДАТЧИКОВ ТЕМПЕРАТУРЫ	
3.1. Цель работы	
3.2. Постановка задачи	30
3.3. Справочные сведения пакета Matlab	
3.3.1. Описание используемых функций	31
3.3.2. Принцип работы функции <code>sga()</code> расчета ИХ	33
3.4. Пример выполнения лабораторной работы	
3.4.1. Подготовка и обработка исходных данных	35
3.4.2. Непараметрические методы идентификации	38
3.4.3. Параметрические методы идентификации	40
3.4.4. Верификация моделей по экспериментальным данным	45
3.5. Задание на лабораторную работу	50
4. ПОСТРОЕНИЕ НЕЛИНЕЙНЫХ ARX-МОДЕЛЕЙ	
4.1. Цель работы	
4.2. Постановка задачи	51
4.3. Справочные сведения по системе MATLAB	52
4.4. Пример построения модели SISO	54
4.4.1. Создание регрессоров	55
4.4.2. Использование пользовательских регрессоров для нелинейных оценок	58
4.5. Пример построения модели ММО	60
4.6. Задание на лабораторную работу	63

5. ИДЕНТИФИКАЦИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ ДАННЫХ ЧАСТОТНОЙ ОБЛАСТИ	
5.1. Цель работы	
5.2. Постановка задачи	65
5.3. Справочные сведения	
5.3.1. Описание используемых функций	66
5.3.2. Выбор структуры модели при использовании функции procest	68
5.4. Пример выполнения лабораторной работы	
5.4.1. Подготовка данных в частотной области	69
5.4.2. Анализ и обработка данных в частотной области.....	70
5.4.3. Сжатие данных.....	74
5.4.4. Модели на основе частотных данных.....	76
5.4.5. Частотно-временное преобразование данных	79
5.5. Задание на лабораторную работу	82
ПРИЛОЖЕНИЯ	84
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	109

ВВЕДЕНИЕ

Задачу идентификации характеристик системы можно рассматривать как задачу сопряженную задаче управления системой, т.к. нельзя управлять системой, если она не идентифицирована либо заранее, либо в процессе управления. Например, мы не можем управлять автомобилем, пока не познакомимся с его реакцией на поворот руля, нажатие педали акселерометра или тормоза, т.е. пока не ознакомимся со свойствами автомобиля. Такой процесс освоения автомобиля и называется процессом идентификации. Результатом идентификации всегда является возможность предсказания или прогнозирования поведения объекта на возможные управляющие воздействия, при этом свойства изучаемого объекта управления (ОУ) обычно описывают математической моделью, например, системой дифференциальных уравнений.

Обычно под ОУ понимается часть окружающего нас мира, на которую мы можем целенаправленно воздействовать, то есть управлять ею. Под моделью обычно понимается выраженная в той или иной форме информация о наиболее существенных характеристиках ОУ.

Для предсказания поведения ОУ лучше всего иметь его математическую модель. Иногда, для составления математической модели системы можно состыковать наперед известные математические модели его подсистем. В этом благоприятном случае математическая модель составляется аналитически. Однако очень часто исследователь не имеет полной информации и часть параметров ОУ остается неизвестной. Такие параметры приходится определять, проводя специальные эксперименты, по результатам которых проводятся расчеты, в которых и находят неизвестные параметры объекта. В качестве возможных экспериментов могут быть такие, как снятие нелинейных и линейных статических и динамических характеристик ОУ, включая его переходные характеристики, интегральные оценки и частотные характеристики.

При идентификации систем часто применяют метод декомпозиции и идентифицируют систему по частям, бывает также полезна различная информация о свойствах системы. Целесообразно знать порядок математической модели, которой можно описать систему. Например, в случае механической системы порядок равен удвоенному числу ее степеней свободы, а если электрическая система содержит лишь сопротивления, транзисторы и диоды, а также n емкостей, то она описывается дифференциальным уравнением n -ого порядка.

Часто стоит вопрос о соответствии заданного устройства тем требованиям и нормам, которые к нему предъявляются. Такая задача называется задачей диагностики. Хорошо известны проблемы диагностики в медицине. Часто они решаются с помощью математических методов и применением ЭВМ (компьютерная томография). Но, не менее широко

используется и техническая диагностика. Она применяется при выпуске сложных устройств, когда все они, или некоторая выборка тщательно проверяются. Например, существует предполетный контроль самолетов и ракет, когда проверяются все системы и по результату проверки выносятся решения о возможности полета. В задачах диагностики широко употребляются методы идентификации тех параметров системы, которые затруднительно измерять непосредственно.

Не менее важным применением идентификации является автоматизация технологических процессов, которая во многом определяется количеством имеющейся информации о технологических объектах управления, их статических и динамических характеристиках.

Определение свойств и характеристик ОУ происходит по-разному, но в этой работе в основном рассматриваются методы, связанные с проведением специального экспериментального исследования, в результате которого получается массив экспериментальных данных $[u_i, y_i]$, где u_i – входные переменные, y_i – выходные переменные ОУ, i – номер опыта.

В первом разделе и приложениях к пособию приводятся справочные сведения, необходимые для проведения лабораторного практикума, которые касаются теории управления, статистики, линейной алгебры. Они адаптированы к требованиям образовательного стандарта высшего профессионального образования по направлению подготовки 200100 Приборостроение (квалификация «магистр»). Это позволяет студенту использовать теоретические сведения, соответствующие предшествующему списку изучаемых дисциплин, отсутствие которых может привести к неприятию изучаемого материала. Представлены лабораторные работы, раскрывающие основные методы параметрической и непараметрической идентификации линейных и нелинейных динамических систем.

При разработке пособия была создана творческая группа во главе с профессором кафедры «Информационно-измерительная техника» ЮУрГУ С.Г. Некрасовым и магистрантами кафедры Н.В. Николайзиным и Р.А. Хажиевым. Разработка лабораторных работ выполнена в SystemIdentificationToolbox¹ системы MATLAB с использованием справочных и демонстрационных материалов системы.

¹Кроме пакета SystemIdentificationToolbox можно использовать также программные пакеты для моделирования блок-схем, включая MATLAB Simulink, Vissim, 20sim и т.д.

1. НЕОБХОДИМЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Задача идентификации

Идентифицируемый объект управления (ОУ) принято [1, 2] представлять в виде на рис. 1.1, где t – время; $U(t)$ – контролируемый входной сигнал; $\tilde{Y}(t)$ – теоретический выход объекта; $Y(t)$ – наблюдаемый выход объекта; $e(t)$ – случайная аддитивная помеха, отражающая действие не учитываемых факторов (шум наблюдения).

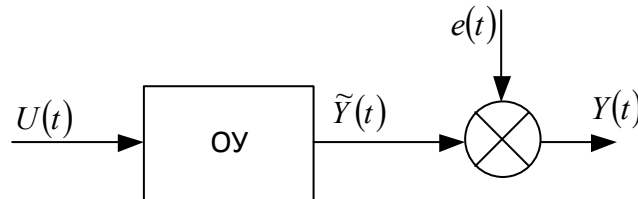


Рис. 1.1. Идентифицируемый объект

Обычно предполагают, что связь между входным и «теоретическим» выходным сигналами задается в виде некоторого оператора Ψ (оператор – правило преобразования):

$$\tilde{Y}(t) = \psi[U(t)], \quad (1.1)$$

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 статистически независимыми.

Цель идентификации заключается в том, чтобы на основании наблюдений за входным $U(t)$ и выходным $Y(t)$ сигналами на каком-то интервале времени определить вид оператора, связывающего входной и теоретический $\tilde{Y}(t)$ выходной сигналы [1, 2, 4, 5].

С математической точки зрения подобная задача является задачей аппроксимации экспериментальных данных некоторым уравнением. Чаще по виду кривой можно оценить уравнение описывающее систему. Затем необходимо вычислить коэффициенты в найденном уравнении. Как правило, в реальных условиях не существует точных формул, позволяющих вычислить эти коэффициенты. Поэтому на практике используются различные методы поиска коэффициентов, такие как метод координатного спуска и метод сопряженных градиентов, симплекс метод, метод наименьших квадратов и другие.

Перед началом экспериментальных исследований проводится априорный анализ перечня входных переменных с целью отбора и включения в состав модели приоритетных, 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 ОУ. Наиболее просто задача определения параметров решается для линейных объектов (линейных по входному воздействию), для которых выполняется принцип суперпозиции.

С учетом выше 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 (см. ГОСТ 20913-75 «Автоматизированные системы управления технологическими процессами»).

1.2. Математическое описание моделей объекта управления

Приведем в качестве справки данные по основным видам моделей линейных непрерывных стационарных динамических объектов [1, 2, 4, 5, 9, 10].

Дифференциальное уравнение. Описание ОУ при помощи дифференциальных уравнений является традиционным, универсальным и рассматривает связь между входным сигналом $u(t)$ и выходным сигналом $y(t)$ в виде уравнения:

$$\sum_{i=0}^{na} a_i y^{(i)}(t) = \sum_{j=0}^{nb} b_j u^{(j)}(t), \quad (1.3)$$

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

Передаточная функция $W(p)$. Данная характеристика определяется как отношение преобразования Лапласа выходного и входного сигналов при нулевых начальных условиях, что, с учетом свойств данного преобразования и выражения (1.3), позволяет получить ее в следующем виде [9, 10]:

$$W(p) = \frac{L\{y(t)\}}{L\{u(t)\}} = \frac{Y(p)}{U(p)} = \frac{\sum_{j=0}^{nb} b_j p^j}{\sum_{i=0}^{na} a_i p^i}, \quad (1.4)$$

где L – символ преобразования Лапласа, $p = \alpha + j\omega$ – оператор Лапласа.

Модель для переменных состояния. В основе этого метода лежит представление дифференциальных уравнений ОУ в нормальной форме Коши, которое дополняется алгебраическими уравнениями, связывающими выходные переменные с переменными состояния [9, 10]:

$$\begin{cases} X'(t) = AX(t) + BU(t) + GZ(t), \\ Y(t) = CX(t) + DU(t), \end{cases} \quad (1.5)$$

где $U(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T$ – вектор-столбец входных воздействий размерностью m (T – символ транспонирования матрицы);
 $Y(t) = [y_1(t), y_2(t), \dots, y_r(t)]^T$ – вектор выходных воздействий размерностью r ;
 $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ – вектор переменных состояния размерностью n ;
 $Z(t) = [z_1(t), z_2(t), \dots, z_k(t)]^T$ – вектор возмущающих воздействий размерностью k .

Здесь A – матрица динамической системы размерности $n \times n$, B – матрица входа (или управления) размерностью $n \times m$, G – матрица возмущений размерности $n \times k$, C – матрица выхода или измерений размерности $r \times n$, D – матрица компенсаций размерностью $r \times m$. Кроме того, здесь n – число переменных состояния или максимальная степень производной исходного дифференциального уравнения, m – число входов по управляющим воздействиям, r – число выходов, k – число входов по возмущениям системы. Последнее уравнение в системе (1.10) называют уравнением выхода и оно служит для перехода от вспомогательных переменных состояния x_i к искомым y_i .

Решению подвергается первое матричное уравнение системы (1.10). В общем случае такое решение можно получить на основе известного метода вариации произвольных постоянных [9, 10] в следующем виде:

$$X(t) = e^{A(t-t_0)} X(t_0) + \int_{t_0}^t e^{A(t-\tau)} (Bu(\tau) + Gz(\tau)) d\tau, \quad (1.6)$$

где $X(t_0)$ – вектор начальных состояний системы, $e^{A(t-\tau)}$ – экспоненциал матрицы A .

В качестве математической модели объекта могут также быть использованы его *переходная и импульсная характеристика (ИХ)*, амплитудно-фазо-частотные характеристики и некоторые другие [9, 10].

Чаще всего на практике измерение непрерывных сигналов производится в дискретные моменты времени, что представляет определенное удобство при последующей обработке данных на ЭВМ. Поэтому существует задача представления непрерывных объектов дискретными моделями.

Существуют несколько способов перехода от непрерывных моделей к дискретным [10, 11], например, с применением преобразования Лапласа и z -преобразования. Часто также используют свойства оператора задержки $z^{-1} = e^{-pT}$, который работает по следующим правилам: $z^{-1}u_k = u_{k-1}$, $z^{-2}u_k = u_{k-2}$,
 \dots

Однако, наиболее часто используется дискретизация дифференциальных уравнений с использованием метода конечных разностей, т.е. заменой на «левые» разности производных в дифференциальном уравнении, описывающем непрерывный объект:

$$\frac{dy(t)}{dt} \approx \frac{y_k - y_{k-1}}{T}, \quad \frac{d^2y(t)}{dt^2} \approx \frac{y_k - 2y_{k-1} + y_{k-2}}{T^2}, \quad (1.7)$$

Например, дискретный аналог модели системы для переменных состояния (1.10) можно получить с использованием метода конечных разностей в следующем виде:

$$\begin{aligned} X_{k+1} &= AX_k + BU_k + GZ_k \\ Y_k &= CX_k + DU_k. \end{aligned} \quad (1.8)$$

1.3. Параметрические модели в SystemIdentificationToolbox

Рассматриваемые в Toolbox методы идентификации ориентированы на численные методы и методы цифровой обработки сигналов. Поэтому необходимо, чтобы наблюдаемые сигналы $u(t)$ и $y(t)$ были представлены последовательностями дискретных отсчетов, иначе говоря, таблицами экспериментальных данных.

Это обстоятельство вынуждает рассматривать искомую модель системы, как дискретную. При необходимости на основе полученного описания дискретной модели может быть создана непрерывная модель реальной системы. Вопросы соответствия передаточных функций дискретных систем и их непрерывных прототипов связаны с обеспечением сходимости дискретной задачи. Показано, что из устойчивости и аппроксимации дискретной задачи следует ее сходимость к непрерывной.

В SystemIdentificationToolbox поддерживаются как линейные, так и нелинейные модели, но все они должны иметь вид моделей дискретных объектов. Рассмотрим сначала линейные дискретные модели.

Все модели можно рассматривать, как частные случаи обобщенной параметрической линейной модели [1, 6]:

$$A(z)y(t) = \frac{B(z)}{F(z)}u(t - nk) + \frac{C(z)}{D(z)}e(t), \quad (1.9)$$

где $A(z) = 1 + a^1z^{-1} + a^2z^{-2} + \dots + a^nz^{-n}$,

$B(z) = b_0 + b_1z^{-1} + \dots + b_mz^{-m}$,

$C(z) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc}$,

$F(z) = 1 + f_1z^{-1} + f_2z^{-2} + \dots + f_{nf}z^{-nf}$, $D(z) = 1 + d_1z^{-1} + d_2z^{-2} + \dots + d_{nd}z^{-nd}$.

Отсюда, в частности, можно получить следующие модели:

1. Модель авторегрессии AR (AutoRegressive):

$$A(z)y(t) = e(t), \quad (1.10)$$

где $e(t)$, – входной сигнал, аналогичный по спектральным свойствам белому шуму.

2. ARX модель (Autoregressive with External Input):

$$A(z)y(t) = B(z)u(t) + e(t), \quad (1.11)$$

3. $ARMAX$ модель (AutoRegressive-MovingAverage with External Input – модель авторегрессии скользящего среднего с внешним входом):

$$A(z)y(t) = B(z)u(t - nk) + C(z)e(t), \quad (1.12)$$

где nk – величина задержки (запаздывания).

4. OE модель («Output-Error» или «выход-ошибка»):

$$y(t) = \frac{B(z)}{F(z)}u(t - nk) + e(t), \quad (1.13)$$

5. Модель Бокса-Дженкинса (BJ):

$$y(t) = \frac{B(z)}{F(z)}u(t - nk) + \frac{C(z)}{D(z)}e(t), \quad (1.14)$$

Отношение полиномов $C(z)/D(z)$ определяет дискретную передаточную функцию сглаживающего (обесцвечивающего) фильтра.

Все эти модели допускают расширение для многомерных объектов (MIMO-модели, имеющие несколько входов и выходов).

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

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \quad (1.15)$$

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

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

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ge(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \quad (1.16)$$

где G – матрица, $e(t)$ – дискретный белый шум.

На практике, все объекты являются нелинейными и их выход является нелинейной функцией входного сигнала. Линейная модель всегда является результатом некоторой идеализацией исходной задачи. Тем не менее, линейной модели часто бывает достаточно, чтобы точно описать динамику системы. В большинстве случаев, сначала нужно попробовать, достаточно ли для описания поведения системы линейной модели.

Возможности нелинейных моделей могут потребоваться в следующих случаях [4]:

1. Линейная модель недостаточно адекватна. Это бывает, когда линейная модель обеспечивает плохое соответствие измеренным

выходным сигналам и не может быть улучшена за счет изменения структуры модели или её порядка. Нелинейные модели имеют гораздо большую гибкость в описании сложных явлений, чем линейные модели такого же порядка.

2. Физическая система имеет слабую нелинейность. Из физической сути или анализа данных иногда можно понять, что система является слабо нелинейной. В этом случае можно построить линейную модель, а затем использовать эту модель в качестве модели начального приближения для нелинейного оценивания. Нелинейная оценка может улучшить соответствие модели экспериментальным данным благодаря корректному описанию динамики объекта, не описываемой линейной моделью.

3. Физическая система нелинейна по своей сути. К таким явлениям относятся сухое трение в механических системах, насыщение силового привода и т.д. В этом случае часто используют модель структуры Гаммерштейна-Винера [13], которая состоит из последовательно соединенных статических нелинейных объектов и динамических линейных элементов. Такие нелинейные модели могут быть полезны для описания систем, работающих в окрестности рабочих точек. В некоторых случаях системе может соответствовать несколько линейных моделей, причем каждая модель является точной только при конкретных условиях эксплуатации.

4. Физические системы, в которых линейная и нелинейная динамика рассматривается отдельно. Это бывает, когда система дает различные результаты при различных уровнях входного сигнала. Можно иметь несколько наборов данных, которые возбуждают линейные и нелинейные динамические процессы отдельно. В этом случае имеет смысл сначала оценить линейную модель системы, используя первый набор данных, а затем использовать полученную модель в качестве исходной модели для оценки нелинейных моделей с использованием второго набора данных.

Различают два класса нелинейных регрессий [4, 5]:

– регрессии, нелинейные относительно включенных в анализ входных и выходных (объясняющих) переменных (регрессоров), но линейные по оцениваемым параметрам (коэффициентам уравнений);

– регрессии, нелинейные по оцениваемым параметрам. Например, рассмотренные выше линейные структуры ARX и ARMAX моделей могут быть расширены до нелинейных следующим образом:

– использованием нелинейных ARX регрессоров, т.е. нелинейных выражений задержанных по времени входных и выходных переменных, например $y(t-1)^2$, $u(t-1) \cdot y(t-2)$, $tg(u(t-1))$, $u(t-1) \cdot y(t-3)$.

– заменой взвешенной суммы линейных регрессоров нелинейной ARX моделью, которая имеет более гибкую нелинейную функцию

отображения $F(y(t-1), y(t-2), y(t-3), \dots, u(t), u(t-1), u(t-2), \dots)$, аргументами для F являются модели регрессоров u и y .

Для наглядности нелинейную модель ARX структуры можно отобразить структурной схемой на рис. 1.2.

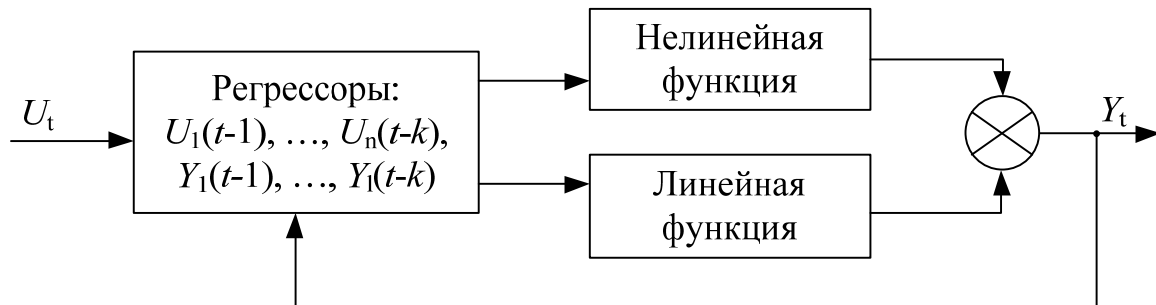


Рис.1.2. Структурная схема нелинейной ARX модели

На схеме видно, что в нелинейной ARX модели (в SystemIdentificationToolbox это объект IDNLARX), каждый выход является функцией регрессоров, которая зависит от предыдущих по времени значений входов и выходов. Типичные регрессоры просто задерживают входные и выходные переменные и называются обычными (standard) регрессорами. Кроме того, можно использовать более совершенные регрессоры в форме произвольно определяемых пользователем функций задержки входных и выходных переменных. Такие регрессоры называют пользовательскими (custom) регрессорами. По умолчанию, все регрессоры (обычные и пользовательские) модели попадают на входы обоих функциональных блоков модели (линейного и нелинейного), при этом можно отключить один из блоков или выбрать подмножество регрессоров для обработки в нелинейном функциональном блоке.

Как в случае линейных, так и нелинейных моделей параметрическая идентификация – всегда многоэтапная процедура. Основные ее этапы следующие:

1. Структурная идентификация, которая заключается в определении структуры математической модели на основании теоретических и эвристических соображений, априорной информации об исследуемом процессе.

2. Выбор критериев близости объекта и модели, основанных на специфике задачи.

3. Параметрическая идентификация, которая включает в себя проведение идентифицирующего эксперимента и определение оценок параметров модели по экспериментальным данным.

4. Проверка адекватности – проверка качества модели в смысле выбранного критерия близости выходов модели и объекта.

Непараметрическая идентификация обычно проводится с целью получения импульсной характеристики оцениваемой модели объекта с использованием временных экспериментальных данных на основе статистических моделей сигналов. Эта функция путем соответствующих преобразований может быть трансформирована в частотный коэффициент передачи, передаточную функцию или системную функцию линейного объекта. Решение задачи параметрической или непараметрической идентификации на основе метода наименьших квадратов подробно рассмотрено в приложении 1.

2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕГРЕССИОННЫХ И АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДАННЫХ

2.1. Цель работы

Целью работы является сравнение регрессионных и авторегрессионных моделей для оценки соответствия результатов модельного прогноза по экспериментальным данным.

2.2. Постановка задачи

Задачи идентификации возникают при управлении работой сложных объектов управления или любых других объектов, параметры которых недоступны прямым измерениям. Решение такого рода задач часто основано на возможностях регрессионного анализа, то есть на построении регрессионных и авторегрессионных моделей (см. приложение 3).

Задачей данной лабораторной работы и является определение соответствия типа и вида регрессионной модели поставленной задаче идентификации. В лабораторной работе использованы демонстрационные и справочные данные MatLab и System Identification Toolbox.

2.3. Справочные данные по Matlab

Таблица 2.1

Функция	Синтаксис	Описание
plot()	plot(x, y)	Производит построение графика функции $y(x)$
corrcoef()	corrcoef(x, y)	Возвращает матрицу коэффициентов корреляции, рассчитанные по матрице X, строки которой являются наблюдениями а столбцы переменными
ones()	ones(x, y)	Возвращает единичную матрицу размерностью x на y
size()	size(A)	Возвращает размерность массива A
stem()	stem(x, y)	Производит построение графика с отображением данных в виде линии отходящей от базовой линии (ось x) – столбчатая диаграмма, гистограмма

Функция	Синтаксис	Описание
polyfit()	polyfit(x, y, n)	Функция аппроксимирует зависимость $y(x)$ полиномом n -го порядка. Возвращает массив коэффициентов полинома
poly2sym()	poly2sym(p)	Функция составляет символьный полином из массива, который содержит коэффициенты полинома
subs()	subs(p(x), y)	Функция замещает в символьном выражении вида $p(x)$ символьную переменную x на значение y
backslash	C\D	Осуществляет левое деление (оператор «backslash») и обеспечивает тем самым решение системы линейных уравнений вида $C*y = D$ в виде $x = C\D$ или $x = mldivide(C,D)$, где C и D матрицы
abs()	abs(a)	Вычисляет модуль
max()	max(A)	Находит максимальный элемент массива
tf()	s = tf('s')	Указывает, что s является переменной Лапласа во всех дальнейших выражениях

2.4. Пример выполнения лабораторной работы

2.4.1. Построение полиномиальной регрессионной модели

Обычно временные последовательности данных описывают поведение какого-либо объекта во времени. Для построения регрессионной модели объекта нужны исходные данные, поэтому воспользуемся возможностями Matlab. Файл `census.mat` содержит данные о населении США с 1790 по 1990 год. Командой `loadcensus` загрузите его. Теперь workspace содержит две новые переменные: `cdate` и `pop`. Первая из них – год, вторая – соответствующая им численность населения. Найдём модель роста численности населения, погрешность которой была бы не более 4 млн. человек.

Построим график зависимости численности населения США `pop` от времени `cdate`:

```
% Загрузка данных для расчета
loadcensus
```

```
% Построение графика загруженных данных
plot(cdate, pop)
```

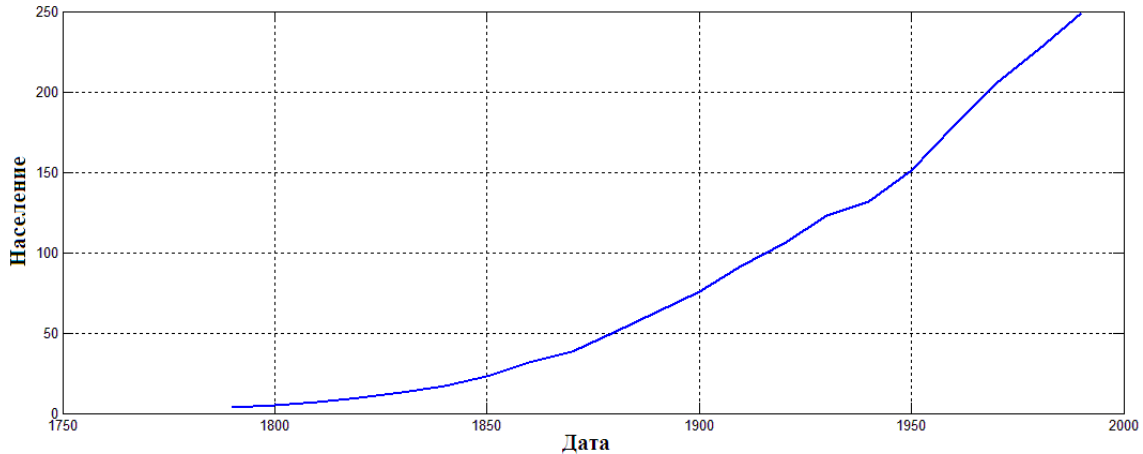


Рис. 2.1. График зависимости народонаселения США (*pop*) от времени (*cdate*)

Прежде чем заниматься построением модели, необходимо убедиться в том, что между входом *cdate* и выходом *pop* имеется взаимосвязь, то есть выбранная нами в качестве входной величина действительно оказывает влияние на интересующую нас выходную величину. Взаимосвязь некоторых зависимостей, представленных данными (векторами или матрицами), называется корреляцией. Общепринятой мерой этой связи является коэффициент корреляции. Его близость к 1 указывает на высокую степень такой зависимости.

Выясним, имеется ли какая-то зависимость между переменными *cdate* и *pop*.

```
%Вычисление коэффициента корреляции
Corr = corrcoef(pop, cdate)

Corr =
    1.000    0.959
    0.959    1.000
```

Диагональные элементы представляют дисперсии для столбцов, а недиагональные элементы $corr(i, j)$ представляют ковариации столбцов i и j . В данном случае зависимость является очень сильной и наше предположение о влиянии *cdate* на *pop* подтвердилось.

Найдем теперь функцию, описывающую отношения между переменными. Исходя из вида графика рис. 2.1, можно предположить, что данные могут быть представлены параболой, т.е. полиномиальной функцией вида $y = a_0 + a_1t + a_2t^2$. Нам неизвестны коэффициенты модели a_0 , a_1 и a_2 . Для их определения имеется 21 уравнение (число 21 здесь равно числу пар данных в переменных *cdate* и *pop*, хотя достаточным является только три пары). Построение таких моделей часто приводит к переопределенным системам уравнений. Неизвестные величины можно

вычислить методом наименьших квадратов, минимизируя сумму квадратов отклонений экспериментальных данных от вычисленных по модели.

Остановимся на этом подробнее. Сформируем функцию отклонений (невязок) экспериментальных и модельных значений:

$$\delta_i = y_i - a_0 - a_1 t_i - a_2 t_i^2$$

и определим сумму квадратов отклонений:

$$S = \sum_{i=1}^N \delta_i^2$$

или в развернутом виде:

$$S = \sum_{i=1}^N \delta_i^2 = \sum_{i=1}^N (y_i - a_0 - a_1 t_i - a_2 t_i^2)^2$$

Далее, для нахождения минимальных значений суммы квадратов отклонений приравняем к нулю производные от S по искомым коэффициентам регрессии:

$$\frac{\partial S}{\partial a_0} = -2 \sum_{i=1}^N (y_i - a_0 - a_1 t_i - a_2 t_i^2) = 0,$$

$$\frac{\partial S}{\partial a_1} = -2 \sum_{i=1}^N (y_i - a_0 - a_1 t_i - a_2 t_i^2) \cdot t_i = 0,$$

$$\frac{\partial S}{\partial a_2} = -2 \sum_{i=1}^N (y_i - a_0 - a_1 t_i - a_2 t_i^2) \cdot t_i^2 = 0.$$

Преобразуем уравнения к следующему виду:

$$\sum_{i=1}^N y_i = a_0 N + a_1 \sum_{i=1}^N t_i + a_2 \sum_{i=1}^N t_i^2,$$

$$\sum_{i=1}^N y_i t_i = a_0 \sum_{i=1}^N t_i + a_1 \sum_{i=1}^N t_i^2 + a_2 \sum_{i=1}^N t_i^3,$$

$$\sum_{i=1}^N y_i t_i^2 = a_0 \sum_{i=1}^N t_i^2 + a_1 \sum_{i=1}^N t_i^3 + a_2 \sum_{i=1}^N t_i^4.$$

Таким образом, мы приходим к системе линейных уравнений с тремя неизвестными, решение которых обычно не вызывает затруднений.

В Matlab существует оператор `\` (backslash или наклонная черта влево), который обеспечивает решение системы линейных уравнений вида $Cy = D$ в виде $y = C \backslash D$ или $y = \text{mldivide}(C, D)$, где C и D матрицы, имеющие согласованное количество строк и столбцов. Сформируем матрицу коэффициентов системы:

```
Y = [ ones(size(cdate)) cdate.^2 ]
Y =
```

```

     1     1790     3204100
     1     1800     3240000
     1     1810     3276100
     .     .     .     .     .     .     .     .
```

1	1970	3880900
1	1980	3920400
1	1990	3960100

С учетом изложенного выше, можно записать:

```
% Поиск коэффициентов полинома
a = Y \ pop;
a =
1.0e+004 *
    2.112959211923060
   -0.002350974599542
    0.000000654113049
```

В результате искомая функция имеет вид, приведенный на графике рис. 2.2, что позволяет визуально оценить отклонение расчетных и экспериментальных данных.

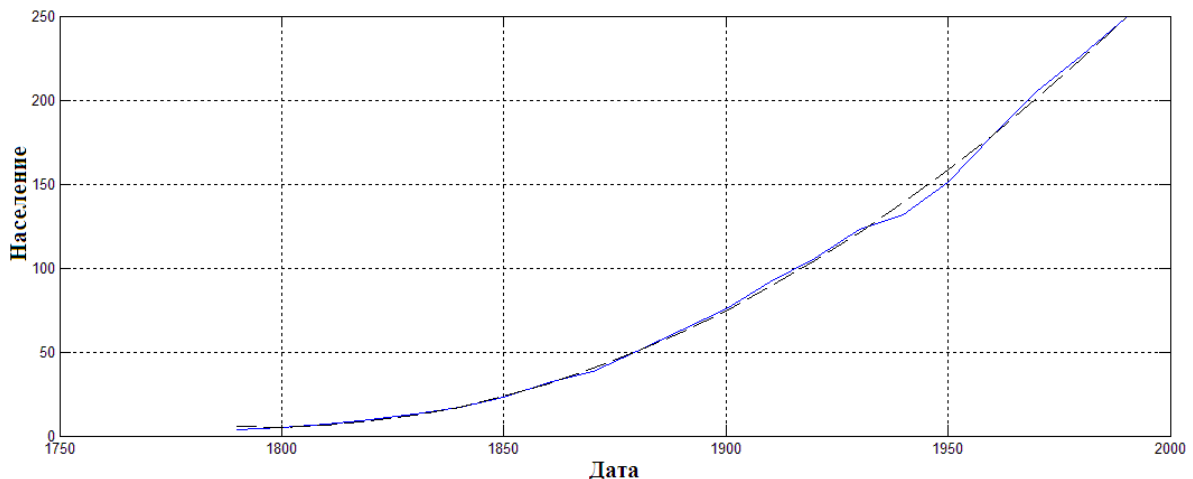


Рис. 2.2. График зависимости народонаселения США (*pop*) от времени (*cdate*)(эксперимент — , полиномиальная модель — — —)

Обозначим в программе найденный полином как *pop2* и определим погрешность моделирования *Delta*. Построим график погрешности (см. рис. 2.3):

```
%Построение полинома модели
pop2 = a(1)+a(2)*t+a(3)*(t.^2);
%Вычисление погрешности
Delta = abs(pop - pop2);
%Определение максимальной погрешности
maxDelta = max(Delta)
%Построение диаграммы погрешностей
stem(cdate, Delta);

maxDelta =
    7.536132425940025
```

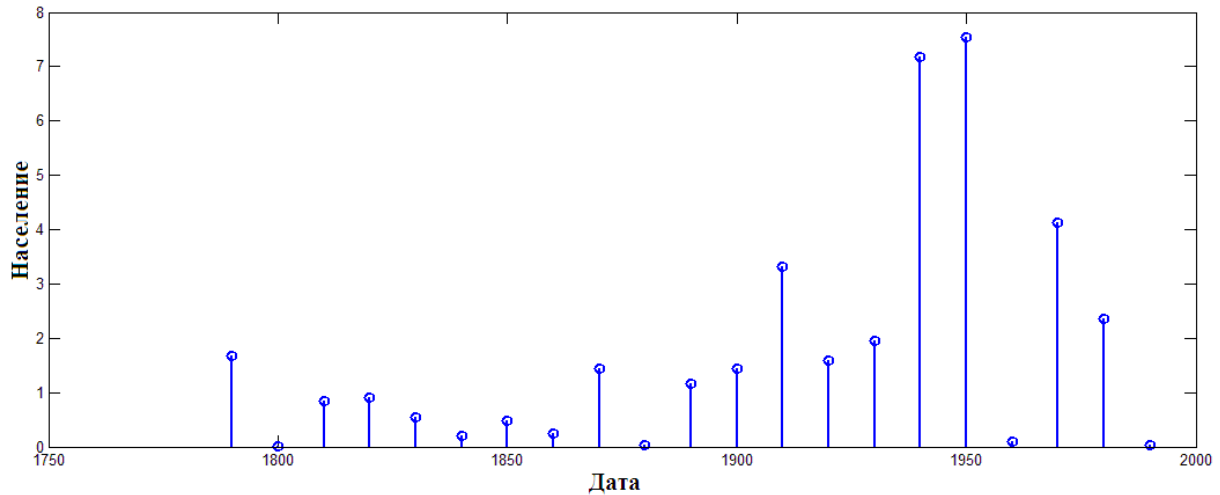


Рис. 2.3. Гистограмма погрешности на всем диапазоне времени

Из этих графиков видно, что полином второго порядка аппроксимирует данные со значительной погрешностью. Мы можем увеличить порядок полинома или исследовать некоторую другую функцию, чтобы получить лучшую аппроксимацию.

Экспериментально выявлено, что наилучшее приближение достигается при аппроксимации функции полиномом седьмого порядка (при помощи функции *polyfit()*):

```
%Аппроксимация зависимости pop от cdate полиномом 7 порядка
[ p ] = polyfit(cdate, pop, 7);

%Создание символьного полинома
popPoly = poly2sym(p);

%Расчет народонаселения в каждой дате
rezPoly = subs(popPoly, cdate);
plot(cdate, pop, cdate, rezPoly)
grid on

%Вычисление погрешности
Delta = abs(pop - rezPoly);

%Определение максимальной погрешности
maxDelta = max(Delta)

maxDelta =
    3.679394531249997
```

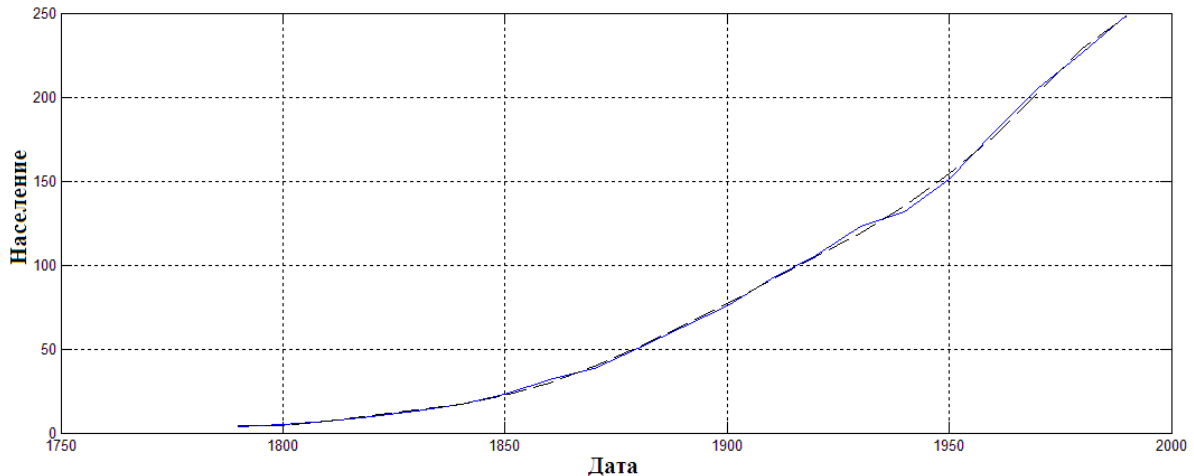


Рис. 2.4. Результат аппроксимации полиномом седьмого порядка (эксперимент —, экспоненциальная модель - - -)

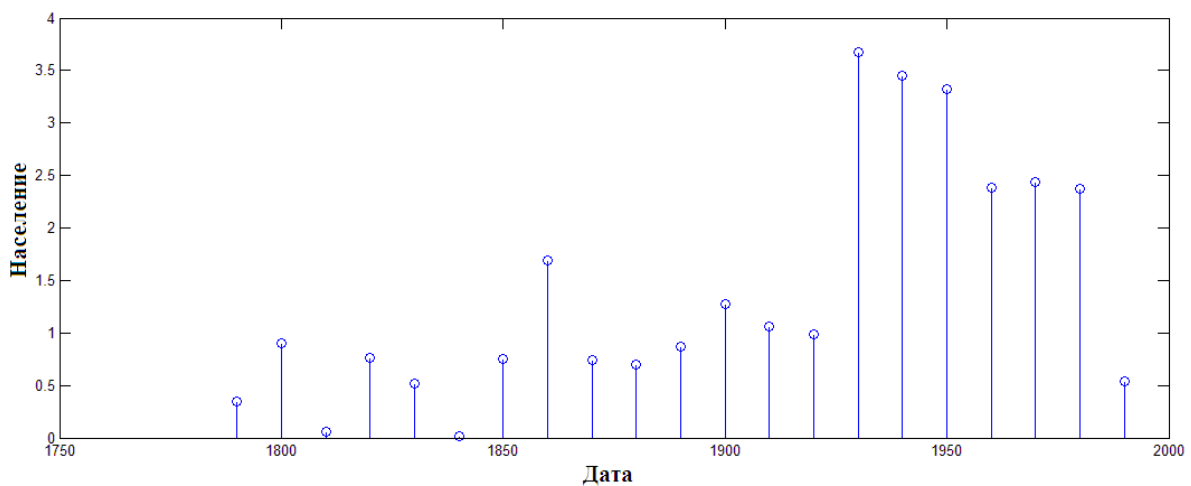


Рис.2.5. Гистограмма погрешности на всем диапазоне времени

Исходя из графика, представленного на рис. 2.5 можно сделать вывод, что условие задачи выполняется, так как максимальная погрешность модели роста численности населения, не превышает 4 млн. человек.

В 2000 году численность населения составила 281 422 тыс. человек. Найдем ошибку прогноза по найденной модели. Сделать это можно применив функцию подстановки *subs()* к найденному нами полиному *popPoly*, а именно:

```
%Расчет численности в 2000 г
pop2000 = subs(popPoly, 2000 )
%Расчет ошибки прогноза
deltaPop = abs(pop2000 - 281.422)
```

```
pop2000 =
    2.462791748046875e+002
deltaPop =
    35.142825195312525
```

Исходя из полученных результатов, можно сказать, что ошибка прогноза модели очень большая и составляет более 35 миллионов человек.

2.4.2. Построение многомерной регрессии

Если y – функция более чем одной независимой переменной, матричные уравнения, которые выражают отношение между переменными, должны быть расширены, чтобы учесть влияние всех независимых переменных.

Предположим, что измерялось количество некоторого произведенного объектом вещества y для нескольких значений параметров x_1 и x_2 . Наблюдения выглядят следующим образом:

$$\begin{aligned}x_1 &= [0,2 \ 0,5 \ 0,6 \ 0,8 \ 1,0 \ 1,1]; \\x_2 &= [0,1 \ 0,3 \ 0,4 \ 0,9 \ 1,1 \ 1,4]; \\y &= [0,17 \ 0,26 \ 0,28 \ 0,23 \ 0,27 \ 0,24].\end{aligned}$$

Многомерная регрессионная модель этой системы в простейшем случае полинома первой степени будет иметь вид:

$$Y = a_0 + a_1x_1 + a_2x_2.$$

Неизвестные коэффициенты a_0 , a_1 и a_2 по-прежнему определяются методом наименьших квадратов. Создадим систему уравнений, формируя регрессионную матрицу X , и решим ее относительно коэффициентов a_0 , a_1 и a_2 , используя оператор `\` (backslash).

```
x1 = [.2 .5 .6 .8 1.0 1.1]';
x2 = [.1 .3 .4 .9 1.1 1.4]';
y = [.17 .26 .28 .23 .27 .24]';
%Матрица коэффициентов системы
X = [ones(size(x1)) x1 x2];
%Вычисление коэффициентов
a = X\y
a=
    0.1018
    0.4844
   -0.2847
```

Это значит, что модель имеет вид $Y = 0.1018 + 0.4844x_1 - 0.2847x_2$.

Оценим качество модели, вычислив максимум модуля отклонения вычисленных данных от эксперимента

```
%Вычисление выхода системы по модели
Y = X*a;
%Вычисление ошибки модели
MaxErr = max(abs(Y-y))
MaxErr =
    0.0038
```

Такую максимальную ошибку можно считать удовлетворительной.

2.4.3. Построение авторегрессионной модели

Рассмотрим набор экспериментальных данных, представленный на рис. 2.10. Это типичный график изменения во времени сигнала на выходе динамической системы. Не будем конкретизировать пока задачу и проведем лишь ее качественный анализ.

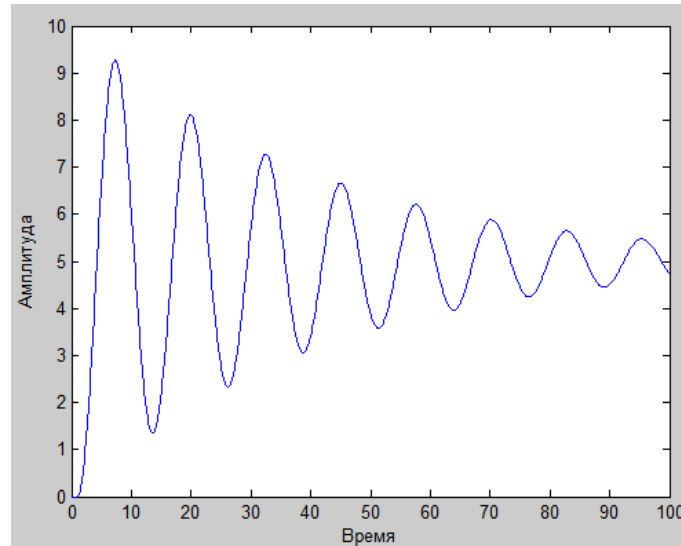


Рис. 2.10. График изменения во времени выходной величины y

Для описания такого набора данных можно воспользоваться построением полиномиальной регрессионной модели. Для повышения скорости решения задачи можно использовать GUI (GraphicalUserInterface – графический интерфейс пользователя) BasicFitting пакета MATLAB, работа в котором здесь не рассматривается и предполагается самостоятельное изучение его возможностей.

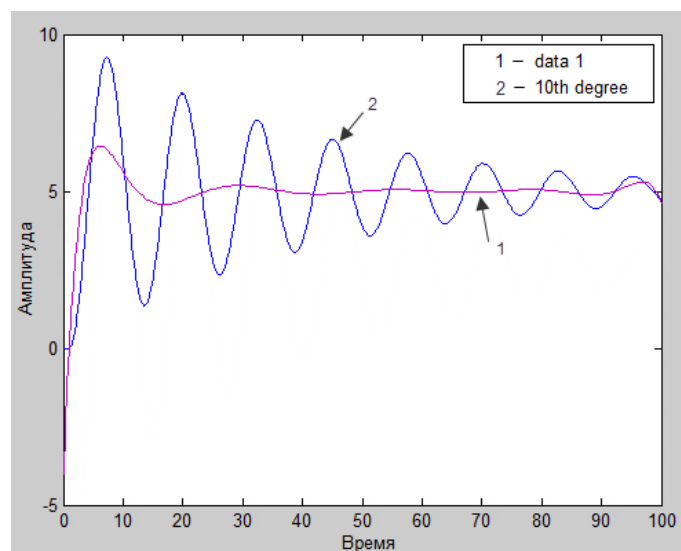


Рис. 2.11. Построение полиномиальной модели при помощи GUI BasicFitting пакета MATLAB

Видим на рис. 2.11, что даже десятый порядок полинома дает весьма значительную погрешность. Что же можно сделать в данном случае?

Заметим, что график этого набора экспериментальных данных весьма сильно напоминает переходную характеристику колебательного звена или, иначе, решение дифференциального уравнения второго порядка (это лишь предварительное заключение):

$$a_2 \cdot \frac{d^2 y}{dt^2} + a_1 \cdot \frac{dy}{dt} + a_0 \cdot y = b \cdot u \text{ для } u = 1(t).$$

Здесь входное воздействие u является единичной функцией или функцией Хевисайда, которая обычно используется для изучения свойств линейных динамических систем. Заменяем производные в уравнении на конечные разности по формулам:

$$\frac{dy}{dt} \approx \frac{y_k - y_{k-1}}{T}, \quad \frac{d^2 y}{dt^2} \approx \frac{y_k - 2 \cdot y_{k-1} + y_{k-2}}{T^2}.$$

Дифференциальное уравнение второго порядка преобразуется в разностное уравнение вида:

$$y_k + a_1^* \cdot y_{k-1} + a_2^* \cdot y_{k-2} = b^* \cdot u,$$

или иначе

$$(1 + a_1^* \cdot z^{-1} + a_2^* \cdot z^{-2}) \cdot Y(z) = b^* \cdot U(z)$$

сподлежащими определению параметрами модели a_1, a_2, b .

Отсюда можно найти

$$y_k = -a_1^* \cdot y_{k-1} - a_2^* \cdot y_{k-2} + b^* \cdot u.$$

Теперь хорошо видна связь текущего значения функции не только с текущим значением аргумента, но и с предыдущими значениями функции. Таково фундаментальное свойство динамических систем – «память», т.е. зависимость текущего состояния системы от предыстории.

Для нахождения неизвестных коэффициентов a_1, a_2, b используем SystemIdentificationToolbox. Графический интерфейс (GUI) пакета SystemIdentificationToolbox запускается из командной строки командой `ident`. Запустив его, можно импортировать данные с помощью оконного интерфейса (рис. 2.12). Для оценки параметров модели надо задать ее структуру. Для этого в раскрывающемся списке Estimate (Выбор) остановимся на варианте Parametricmodels (Параметрические модели). Данный выбор приведет к открытию диалогового окна задания структуры модели. По умолчанию пользователю предлагается ARX-модель (ARX–AutoRegressivewiththeXternalinput– авторегрессионная с внешним входом).

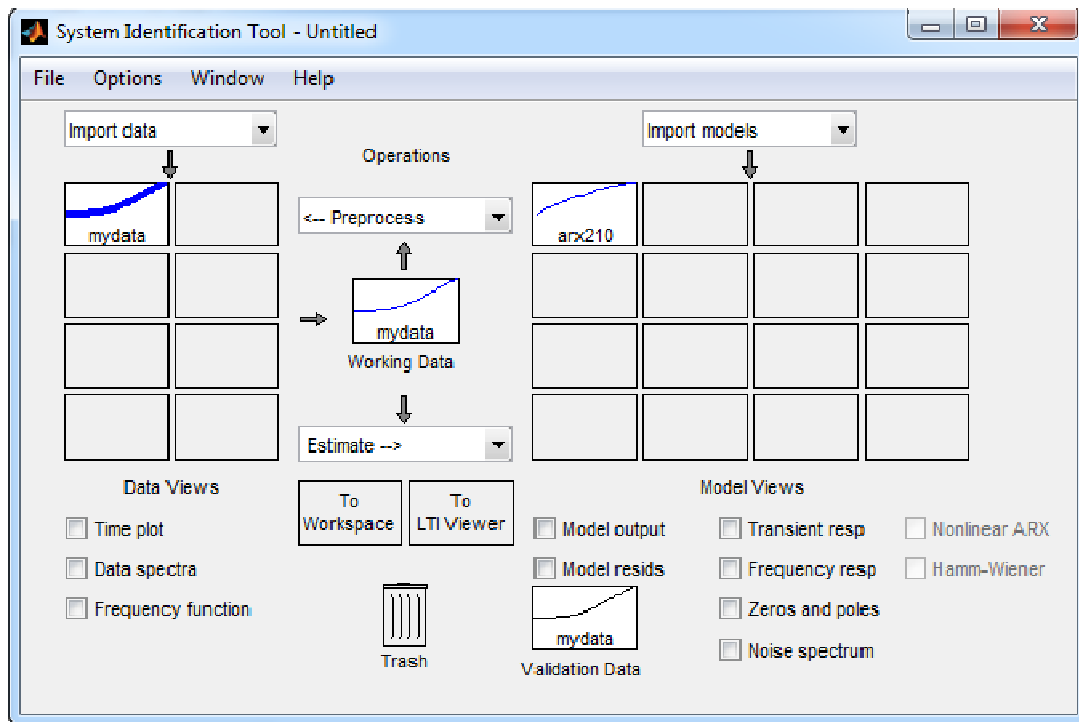


Рис. 2.12. Построение авторегрессионной модели при помощи GUI SystemIdentificationToolbox пакета MATLAB

Создадим временные последовательности входных и выходных данных динамического объекта, для чего воспользуемся возможностями системы Matlab. Определим передаточную функцию следующего вида:

$$W = \frac{5e^{-p}}{(K^2 p^2 + 0,1Kp + 1)}$$

Для этого, сначала, необходимо определить переменную p , как параметр передаточной функции с помощью команды $p = tf('p')$, а затем сформировать передаточную функцию определенного выше вида. Пусть $K = 2$, тогда:

```
p = tf('p');
K = 2;
W = (5*exp(-p))/(K^2*p^2 + 0.1*K*p + 1)
```

Для получения массива значений реакции такой системы на ступенчатое воздействие, воспользуемся возможностями пакета Simulink. Из пакета ControlSystemToolbox необходимо взять модуль LTISystem и загрузить в него созданную передаточную функцию W из рабочего пространства (WorkSpace). Поставьте на вход системы ступенчатое воздействие (Step) и определите начало воздействия в момент времени $t = 0$. Затем с помощью блока ToWorkspace выгрузите массив значений отклика системы в рабочую область. Параметры симуляции подберите таким образом, чтобы при фиксированном шаге симулирования 0,1

секунда конечное значение амплитуды колебаний не превышало 10 % от максимального значения.

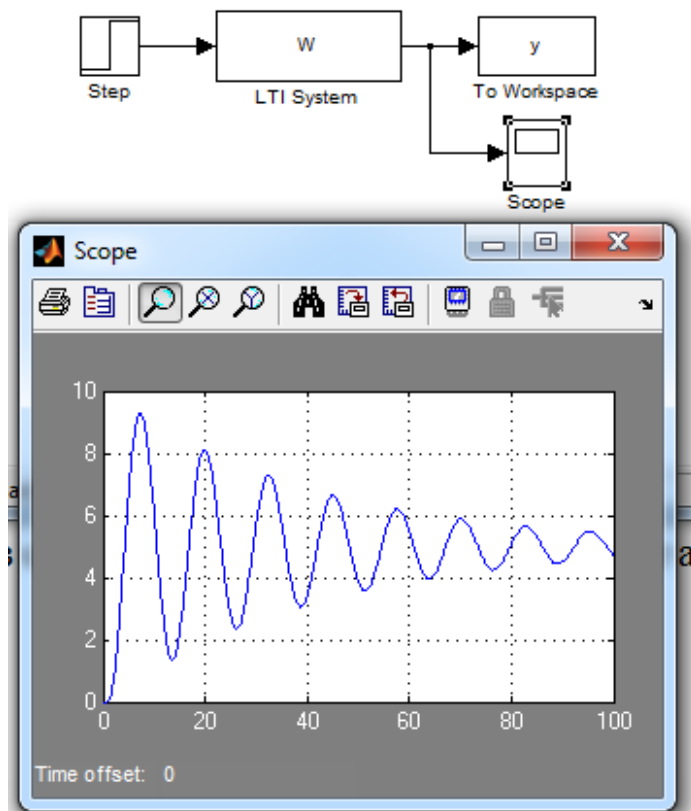


Рис. 2.13. Модель системы в Simulink и её отклик на ступенчатое воздействие

Теперь можно запустить графический интерфейс пакета SystemIdentificationToolbox и загрузить в него параметры системы. Для большего удобства лучше записать значения отклика системы на ступенчатое воздействие, полученные после работы Simulink в виде структуры y , в отдельную переменную, например Y . В графический интерфейс загрузите данные как «определенные во временной области», при этом в качестве переменной входа укажите 1 (ступенчатое воздействие), а в качестве выхода значения, полученные из Simulink. Поскольку на вход системы все время действует ступенька, входная переменная должна быть массивом, состоящим из единиц. Размерности массивов входа и выхода должны совпадать.

Выберем простейший вариант (рис. 2.14) ARX-модели (в поле Estimate выбрать LinearParametricModels) с параметрами $na = 2$, $nb = 1$, $nk = 0$ (nk – количество звеньев с запаздыванием). Результатом будет arx210-модель.

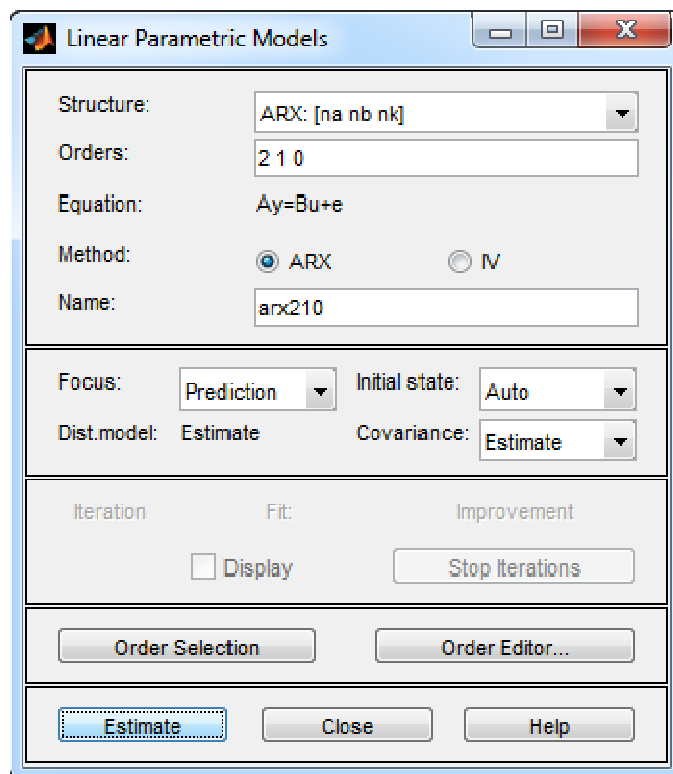


Рис. 2.14. Задание параметров ARX–модели системы

Получим следующий результат (рис. 2.15):

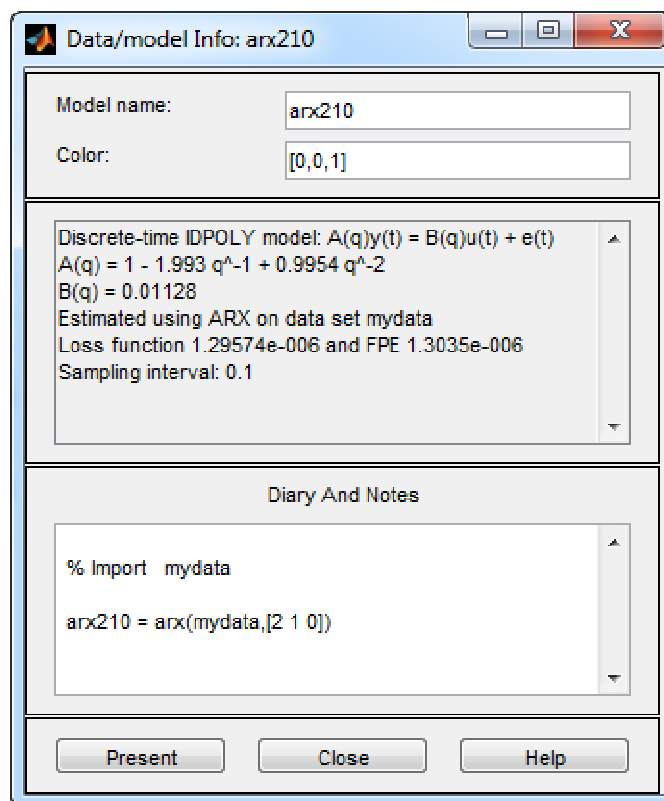


Рис. 2.15. ARX-модель системы

Исходя из значений полученных коэффициентов полинома A и B с учетом вида ARX-модели, оценка модели выглядит так:

$$y_k = 1,993y_{k-1} - 0,9954y_{k-2} + 0,01128u$$

Рассчитаем выходной сигнал на основе полученной ARX-модели. Для этого используем первые два дискретных значения выхода системы рассчитанные в Simulink, а остальные получим по ARX-модели

```

Simout = Y;
% Вычисление значения выходного сигнала по параметрам
% модели и по предыдущим 2-м значениям выхода
simout(3) = 0.01128+1.993*simout(2)- 9954*simout(1);

%Автоматизация процесса вычисления выхода модели
for i=4:1001
    simout(i) = 0.01128+1.993*simout(i-1)-0.9954*simout(i-2);
end

%Построение графика выходной величины
plot(tout, Y, tout, simout)
legend('Y', 'simout')

```

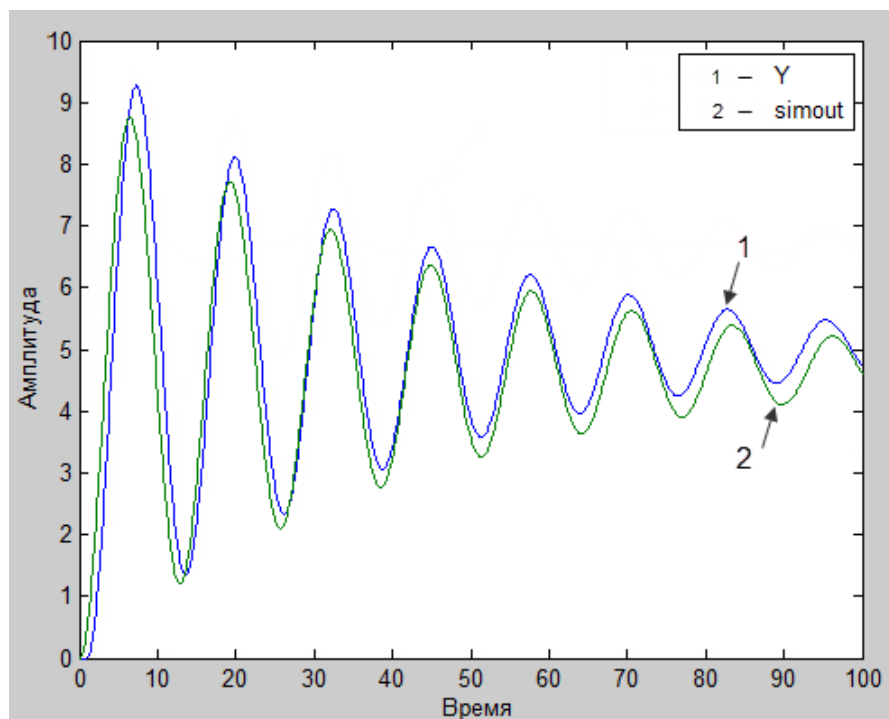


Рис. 2.16. Графики исследуемой системы и полученной модели

Ошибка модели, как следует из графика на рис. 2.16, во много раз меньше, чем у полиномиальной модели десятого порядка при существенно меньшем объеме вычислений. Эту ошибку можно еще более уменьшить путем изменения порядка ARX-модели.

2.5. Задания на лабораторную работу

1. Изучить теоритические сведения.
2. Уяснить задание на лабораторную работу.
3. Выполнить пример лабораторной работы и сохранить введенную в рабочей области последовательность операторов в виде скрипт-файла.
4. Ввести в рабочую область временной ряд данных на основе передаточной функции апериодического типа в соответствии с вариантом задания и построить полиномиальную и экспоненциальную модель на основе полученного при изучении скрипт-файла.
5. Построить модель вида многомерной регрессии в соответствии с вариантом задания.
6. Ввести в рабочую область временной ряд данных на основе передаточной функции колебательного типа в соответствии с вариантом задания и построить ARX-модель с помощью GUI пакета SystemIdentificationToolbox.
7. Сделать выводы.
8. Оформить отчет.

Варианты заданий

Для построения одномерных регрессионных моделей объекта создайте:

1. Временной ряд значений на основе передаточной функции апериодического типа, представленной в таблице 2.2 (одномерная регрессия).

2. Для построения многомерной регрессии создайте временной ряд значений на основе статического звена в виде полинома следующего вида:

$$Y = A_0 + A_1x_1 + A_2x_1x_2 + A_3x_2^2 + A_4x_1^2.$$

На вход x_1 и вход x_2 подайте сигналы апериодического типа по п.1 в соответствии с вашим вариантом i и вариантом $i+1$. Для создания статического звена используйте блоки умножения, сложения, возведения в квадрат в стандартном математическом наборе блоков Simulink.

3. Для построения одномерных авторегрессионных моделей объекта создайте временной ряд значений на основе передаточной функции колебательного типа, выбрав ее в таблице 2.2.

Для получения массивов временных значений получите реакцию таких звеньев на ступенчатое воздействие, воспользовавшись возможностями пакета Simulink, аналогично разделу 2.4.4. Значение временного шага и время наблюдения определите экспериментально, воспользовавшись рекомендациями раздела 2.4.4 (ARX-модель).

Таблица 2.2

Вариант	Одномерная регрессия	Многомерная регрессия	ARX-модель
1	$W = \frac{K+4}{(0.05(K+1)p+K)}$	$a_0 + a_1x_1 + a_2x_1x_2 + a_3x_2^2$	$W = \frac{5+p^2}{(K^2p^2+0.1Kp+1)}$
2	$W = \frac{-K}{(Kp+K)}$	$a_1x_2 + a_2x_1x_2 + a_1x_1^2 + a_3x_2^2$	$W = \frac{K+2*p}{(K^2p^2+0.1Kp+1)}$
3	$W = \frac{3K+1}{(Kp+0.1K)}$	$a_1x_2 + a_2x_1^3x_2 + a_1x_1^2 + a_3x_2^3$	$W = \frac{5Ke^{-2p}}{(K^2p^2+0.5Kp+3)}$
4	$W = \frac{-2K}{(0.01Kp+0.1K)}$	$a_0 + a_1x_1^3 + a_2x_2 + a_3x_2^2$	$W = \frac{2+Kp^2}{(0.3K^2p^2+0.5)}$
5	$W = \frac{(K-3)e^{-p}}{(0.25(K-3)p+(K-2))}$	$a_0 + a_1\sqrt{x_1} + a_2x_2 + a_3x_1x_2^2$	$W = \frac{Ke^{-p}}{(K^2p^2-0.1Kp+3)}$

Таблица 2.3

Вариант	K	$a0$	$a1$	$a2$	$a3$
1	2	0,1	0,5	-0,2	0.05
2	4	0,2	0,15	-0,3	-0.03
3	6	0,5	0,25	-0,4	0.04
4	3	0,8	0,75	-0,7	-0.02
5	0,7	0,35	0,95	-0,6	-0.15

Содержание отчета

Отчет оформляется в соответствии с требованиями к оформлению лабораторных работ в соответствии с СТП ЮУрГУ и должен содержать:

- 1) титульный лист;
- 2) формулировку цели работы;
- 3) исходные данные варианта задания, краткие теоретические сведения;
- 4) результаты работы;
- 5) общие выводы;
- 6) приложения (листинг программы с комментариями).

Контрольные вопросы

1. Каким методом находятся неизвестные коэффициенты модели?
2. Какие виды аппроксимаций можно произвести с помощью пакета basicfilling?
3. Как производится скалярное и векторное перемножение матриц в matlab?
4. Какой тип уравнений описывает поведение динамической системы?

3. НЕПАРАМЕТРИЧЕСКАЯ И ПАРАМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ЛИНЕЙНОЙ СИСТЕМЫ ХАРАКТЕРИЗАЦИИ ДАТЧИКОВ ТЕМПЕРАТУРЫ

3.1. Цель работы

Целью работы является идентификация структуры и параметров системы характеристики датчиков по каналу тепловая мощность–температура с использованием методов параметрической и непараметрической идентификации.

3.2. Постановка задачи

Для идентификации технологических объектов управления (ТОУ) возьмем один из наиболее распространенных технологических процессов, применяемых в измерительной технике, – тепловой процесс характеристики датчиков температуры. Фактически, это процесс калибровки (градуировки) датчиков температуры, что характеризует их метрологические свойства (отсюда и название – характеристика). Аналогично можно характеризовать датчики давления, датчики расхода жидкости и газа, естественные показания которых обычно существенно зависят от температуры.

Характеризация происходит в термокамере, в которую загружаются соответствующие датчики (иногда несколько десятков). Управляющим воздействием является изменение тепловой мощности, определяющее температурный режим процессов. Кроме основного в тепловых ТОУ есть другие воздействия: изменение давления, влажности, изменение зазора (в клапанах, задвижках датчиков и т.д.) и др.

Конечное время распространения тепловых процессов (тепловая инерция) при наличии множества датчиков в термокамере (это приводит к увеличению теплоемкости, а значит и тепловой инерции), а также распределение температуры в пространстве камеры приводят к появлению динамической и пространственной погрешности процесса характеристики. Поэтому, в целом, тепловые ТОУ являются достаточно сложными многомерными объектами, температура которых в рабочей зоне определяется множеством влияющих воздействий.

Исследуем основной тепловой процесс в термокамере, в котором входным воздействием на ОУ является электрическая мощность нагревательного элемента, выраженная вВт, а выходным регулируемым параметром – температура в центре камеры, выраженная в градусах Цельсия.

Обработка массива данных мощность – температура с помощью пакета System Identification Toolbox предполагает следующие этапы:

- обработка и преобразование данных с целью создания файла данных;
- непараметрическое оценивание данных с целью предварительного определения основных характеристик ОУ;
- параметрическое оценивание данных с целью создания различных видов моделей;
- проверка адекватности и сравнение различных моделей с целью выбора наилучшей.

3.3. Справочные сведения пакета Matlab

3.3.1. Описание используемых функций

Описание основных функций, используемых в данной лабораторной работе, приводится в таблице 3.1.

Таблица 3.1

Функция	Синтаксис	Описание
iddata()	data = iddata(y, u, Ts)	Функция-конструктор, создает объект типа IDDATA, для которого доступны все функции и команды анализа систем. Он включает в себя: u, y – входные и выходные экспериментальные данные; Ts – интервал дискретизации
detrend()	detrend(z)	Функция удаляет тренд из набора данных «z»
cra()	cra(z) [ir, R, cl] = cra(z, M, na, plot)	Функция определяет оценку импульсной характеристики методом корреляционного анализа (непараметрический метод). z – iddata-объект с экспериментальными данными; M – максимальное число дискретного аргумента, определяющего задержку входа; na – порядок модели авторегрессии (степень многочлена $A(z)$), используемая для расчета «обеляющего» фильтра $\Phi(z)$, по умолчанию na = 10, а при na = 0 в качестве идентифицирующего используется не преобразованный 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. Возвращаемые величины: ir – оценка ИХ (вектор);

Продолжение табл. 3.1

Функция	Синтаксис	Описание
		R – матрица, элементы 1–го столбца которой – значения дискретного аргумента, 2–го столбца – значения оценки АКФ выходного сигнала (возможно фильтрованного), 3–го столбца – значения оценки АКФ входного сигнала (возможно «обеленного»), 4–го столбца – значения оценки ВКФ. cl – 99% доверительный коридор для оценки ИХ
bode()	bode(sys)	Функция возвращает диаграммы Боде – графики ЛАЧХ и ФЧХ. Аргументы функции: sys – имя исследуемой модели
spa()	[g, phiv] = spa(z)	Функция возвращает частотные характеристики (АЧХ и ФЧХ) одномерного объекта и оценки спектральных плотностей его сигналов для обобщенной линейной модели объекта (модель объекта в частотном формате). z – матрица экспериментальных данных; g – оценка $W(e^{j\omega T})$ в частотном формате; phiv – оценка спектральной плотности шума $v(t)$
pem()	sys = pem(z) sys = pem(z,nn)	Функция оценивает параметры обобщенной многомерной линейной модели вида: $A(z)y(t) = \frac{B_1(z)}{F_1(z)}u_1(t - nk_1) + \frac{B_2(z)}{F_2(z)}u_2(t - nk_2) + \dots + \frac{B_{nu}(z)}{F_{nu}(z)}u_{nu}(t - nk_{nu}) + \frac{C(z)}{D(z)}e(t)$ sys – имя (идентификатор) построенной модели; z – матрица экспериментальных данных; nn – задается в виде nn = [na, nb, nc, nd, nf, nk], где na, nb, nc, nd, nf, nk – в многомерном случае являются вектор-столбцами
impulse()	impulse (sys) impulse (sys1, sys2, ..., sysN) [y, t, x] = impulse (sys)	Функция возвращает импульсную характеристику модели или нескольких моделей. Имеет следующие аргументы: sys – имя (идентификатор) построенной модели; y, x, t – векторы, содержащие значения переходного процесса, переменных состояния и моментов времени; при возвращении данных величин график переходного процесса не отображается
compare()	compare(data, sys)	Функция проверки адекватности модели. Производит сравнение выхода модели sys с экспериментальными данными data с выводом на дисплей сравнительных графиков с указанием критерия адекватности

Функция	Синтаксис	Описание
nyquist()	nyquist (sys); [re, im, w] = nyquist(sys)	Функция возвращает диаграмму Найквиста (годограф Найквиста). Аргументы функции: sys – имя (идентификатор) построенной модели; re, im – соответственно векторы значений вещественной и мнимой частей комплексного коэффициента передачи для заданного набора частот; w – вектор частот
step()	step (sys) step (sys1, sys2, ..., sysN) [y, t, x] = step (sys)	Функция рассчитывает и строит реакцию модели на единичную ступенчатую функцию, то есть возвращает переходную функцию системы. sys – имя (идентификатор) модели для которой строится переходная функция; y, x, t – векторы, содержащие значения переходного процесса, переменных состояния и моментов времени; при возвращении данных величин график переходного процесса не отображается
arx()	sys = arx (z, nn)	Функция оценивает параметры моделей AR и ARX. Аргументы функции: z = [y, u] – матрица экспериментальных данных; в случае многомерного входа u представляет собой матрицу с числом столбцов, равным числу входов; nn = [na, nb, nc, nk] – задаваемые параметры ARMAX-модели (степени полиномов и величина задержки); Функция может использоваться для оценивания параметров многовходовой модели вида: $A(z)y(t) = B_1(z)u_1(t - nk_1) + B_2(z)u_2(t - nk_2) + \dots + B_{nu}(z)u_{nu}(t - nk_{nu}) + e(t)$
pzmap()	pzmap (sys) [p, z] = pzmap (sys)	Функция возвращает карту нулей z и полюсов p. В последней форме записи график не выводится, а возвращаются векторы, элементы которых являются нулями и полюсами модели sys

3.3.2. Принцип работы функции cra() расчета ИХ

Данный раздел пособия можно пропустить при чтении, т.к. информация является адаптированным переводом документации на функцию *cra()* [14] и можно воспользоваться первоисточником.

Функция *cra()* предварительно «обеляет» входную последовательность *u* экспериментальных данных (см. приложение 3), пропуская их через

обеляющий фильтр с той целью, чтобы шумы подчинялись нормальному закону распределения. Выходная последовательность данных y аналогично подвергается фильтрации и затем вычисляются функции ковариации отфильтрованных данных ииус выводом результатов в виде графиков. Так же на графике выводится функция взаимной корреляции входного и выходного сигналов («обеленных»). Корреляция при положительных значениях задержек M говорит о том, что входная переменная x имеет зависимость с более поздними значениями выходной переменной y . С другой стороны, существенная корреляция для отрицательных значений задержек M говорит о наличии обратной связи между входом и выходом исследуемой системы.

Функция $cra()$ производит оценку импульсной характеристики (ИХ) системы (см. более подробно приложение 1), которая выводится на экран в виде графика с 99 %-ным доверительным коридором. Как утверждается в описании функции, на графике результирующая ИХ масштабируется так, что высота импульса соответствует величине $1/T$, а его длительность T , где T – интервал дискретизации исходной выборки данных (ts в примере ниже).

Выходная матрица R содержит информацию о функциях ковариации (см. приложение 2). Первый столбец матрицы представляет собой индексы задержки, второй – функцию ковариации выхода, третий столбец – функцию ковариации входа, а четвертый – взаимокорреляционную функцию входного и выходного сигналов.

Входной дискретный параметр M определяет величину задержек, для которых вычисляются функции корреляции. Данная величина определяется от $-M$ до M , так, что размерность матрицы R равна $(2M+1) \times 4$. Импульсная характеристика вычисляется на интервале задержек от 0 до M , причем по умолчанию $M = 20$.

С целью предварительного «обеления» исходных данных вход системы представляется в виде AR-модели порядка na , указываемого в качестве входного параметра функции (см. табл. 3.1). Это значит, что полученная параметрическая авторегрессионная модель наблюдений будет формировать обеленный входной сигнал. Полученные коэффициенты AR-модели позволят синтезировать оптимальный для этого сигнала обеляющий фильтр, причем вычисление AR-коэффициентов производится в расчете на минимизацию ошибки линейного предсказания по её дисперсии (см. приложение 3). По умолчанию, порядок модели na принимается равным 10. При $na = 0$ исходные данные не «обеляются».

3.4. Пример выполнения лабораторной работы

3.4.1. Подготовка и обработка исходных данных

Подготовим входные и выходные данные для дальнейшей работы. Пусть имеем некоторые измерительные данные работы термокамеры. Создадим вектор-столбцы выходных (y) и входных (u) данных в системе Matlab. Данные можно сохранить в единый mat-файл (например «testData») для удобства импорта в SystemIdentificationToolbox. В данном случае вектор y содержит 1000 измерений температуры в центре камеры, а вектор u – 1000 измерений потребляемой мощности нагревательного элемента.

Сохранить полученные данные в единый mat-файл можно следующим образом.

1. Создаем новую переменную y : *File*→*New*→*Variable*. В окне Workspace появится новая переменная «unnamed». Переименуем её как « y ». Откроем полученную переменную в редакторе переменных VariableEditor двойным щелчком мыши и в появившемся окне заполним массив измерительными данными. Для переменной u выполним аналогичные процедуры.

2. Полученные переменные y и u сохраняем в mat-файл. В окне Workspace нажимаем кнопку *Save* и задаем имя mat-файла «data».

Для импорта подготовленных данных в рабочую область, воспользуемся командой:

```
load testData;
```

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

```
ts = 0.08;
```

Для дальнейшей работы с данными необходимо сформировать объект типа «iddata» (объект идентификационных данных, с которыми может работать SystemIdentificationToolbox), наполнив его нашими экспериментальными данными. Необходимость преобразования исходных векторов y и u в единый объект класса *iddata* обусловлена тем, что для него доступны все функции и команды анализа систем. Конструктор класса *iddata* выглядит следующим образом:

```
sys_obj = iddata(y, u, ts),
```

где параметр ts – заданный выше интервал дискретизации.

В результате мы получили новую переменную *sys_obj*, которую можно использовать в качестве данных во всех процедурах оценки и анализа систем. Для получения информации о свойствах, созданного нами объекта «*sys_obj*», можно воспользоваться функцией *get()*:

```
info_obj = get(sys_obj).
```

Возвращаются данные в виде структуры *info_obj*, описывающей наш объект идентификации:

```
info_obj=  
Domain: 'Time'  
Name: ''  
OutputData: {[1000x1 double]}  
OutputName: {'y1'}  
OutputUnit: {''}  
InputData: {[1000x1 double]}  
InputName: {'u1'}  
InputUnit: {''}  
Period: {[Inf]}  
InterSample: {'zoh'}  
Ts: {[0.0800]}  
Tstart: {[ ]}  
SamplingInstants: {[1000x0 double]}  
TimeUnit: ''  
ExperimentName: {'Exp1'}  
UserData: [ ]
```

Для более удобного представления данных необходимо ввести в структуру обозначения входных и выходных данных. Назовем наши переменные «*y*» и «*u*» как «*Temperature*» и «*Power*» соответственно и определим их единицы измерения путем непосредственного обращения к полям структуры «*info_obj*» объекта «*sys_obj*»:

```
sys_obj.InputName = 'Power';  
sys_obj.OutputName = 'Temperature';  
sys_obj.TimeUnit = 'Seconds';  
sys_obj.InputUnit = 'Watt';  
sys_obj.OutputUnit = '^o C'.
```

Также имеется возможность работать не со всеми экспериментальными данными, а лишь с их частью. Выделим из нашей исходной (генеральной) совокупности выборку из 300 значений температуры и мощности для оценки модели:

```
ze = sys_obj(1 : 300).
```

Выведем полученный объект на экран (рис. 3.1), для чего воспользуемся командой *plot()*, либо командой *idplot()*, однако в последнем случае графики не будут содержать информации о названии переменных и их размерностях:

```
plot(ze(200 : 300)).
```

В данном случае будут выведены 100 выборок сигнала *u* и *y* в интервале от 200 по 300.

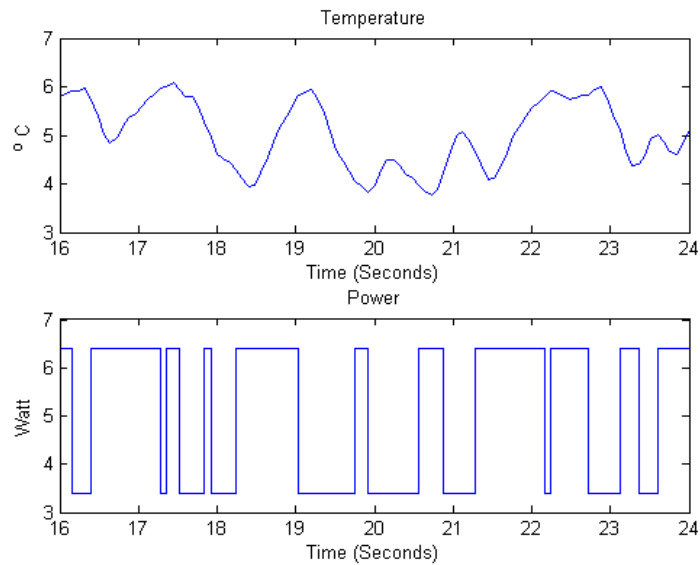


Рис. 3.1. Данные для оценки модели

Из полученных графиков видно, что данные имеют ненулевое среднее. Для дальнейшего использования исходных данных необходимо провести предварительную обработку этих данных с целью удаления тренда и, если необходимо, отфильтровать данные с помощью имеющихся средств пакета `SystemIdentification`. Удалим постоянную составляющую, при помощи функции `detrend()`:

```
ze = detrend(ze) .
```

Выведем на экран полученные результаты с удаленной постоянной составляющей (рис. 3.2):

```
plot(ze(200 : 300)) .
```

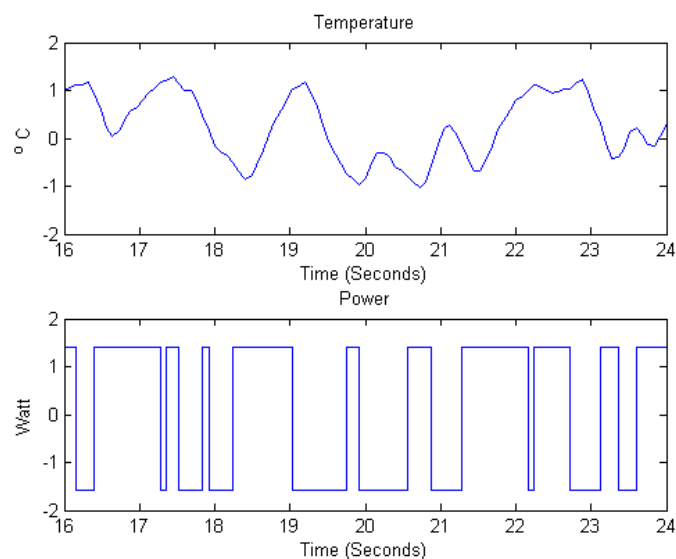


Рис. 3.2. Данные модели с удаленной постоянной составляющей

В результате этого получен новый файл данных, в котором отсутствует постоянная составляющая сигнала. Этот файл в дальнейшем будет нами использован для построения моделей ТОУ.

Кроме приведенной команды удаления тренда в пакете System Identification Toolbox имеются другие функции обработки данных эксперимента, с которыми можно ознакомиться в описании пакета [12]. Применение этих функций требует осторожности и 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.

3.4.2. Непараметрические методы идентификации

Определим оценку импульсной характеристики, воспользовавшись функцией *cra()*, установив параметры «*M*», «*na*» выбранными по умолчанию (см. табл. 3.1) и для повышения точности статистической модели используем весь массив экспериментальных данных. В этом случае вызов функции будет выглядеть следующим образом:

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

Для более наглядного представления, поместим результаты работы функции в таблицу 3.2.

Таблица 3.2

ir	R				cl
-0.0023	-20.0000	0.0171	-0.0599	0.0135	0.0435
-0.0014	-19.0000	0.0161	0.1115	0.0416	
0.0003	-18.0000	0.0147	0.0011	0.0655	
0.0657	-17.0000	0.0125	0.0455	0.0575	
0.1233	-16.0000	0.0100	0.0070	0.0531	
0.1361	-15.0000	0.0075	0.0149	0.0591	
0.1235	-14.0000	0.0060	-0.0444	0.0663	
0.1037	-13.0000	0.0050	-0.0571	0.0710	
0.0800	-12.0000	0.0056	-0.0075	0.0576	
0.0641	-11.0000	0.0073	0.0260	0.0439	
0.0504	-10.0000	0.0095	0.0028	0.0153	
0.0368	-9.0000	0.0127	-0.0022	-0.0025	
0.0270	-8.0000	0.0171	0.0011	-0.0131	
0.0180	-7.0000	0.0227	0.0052	-0.0062	
0.0186	-6.0000	0.0303	-0.0008	-0.0169	
0.0127	-5.0000	0.0403	-0.0022	-0.0065	
0.0088	-4.0000	0.0525	0.0043	0.0102	
0.0028	-3.0000	0.0673	-0.0093	-0.0038	
-0.0010	-2.0000	0.0835	0.0120	-0.0078	
0.0006	-1.0000	0.0979	-0.0132	0.0085	
0.0019	0	0.1053	1.2445	-0.0079	
.	

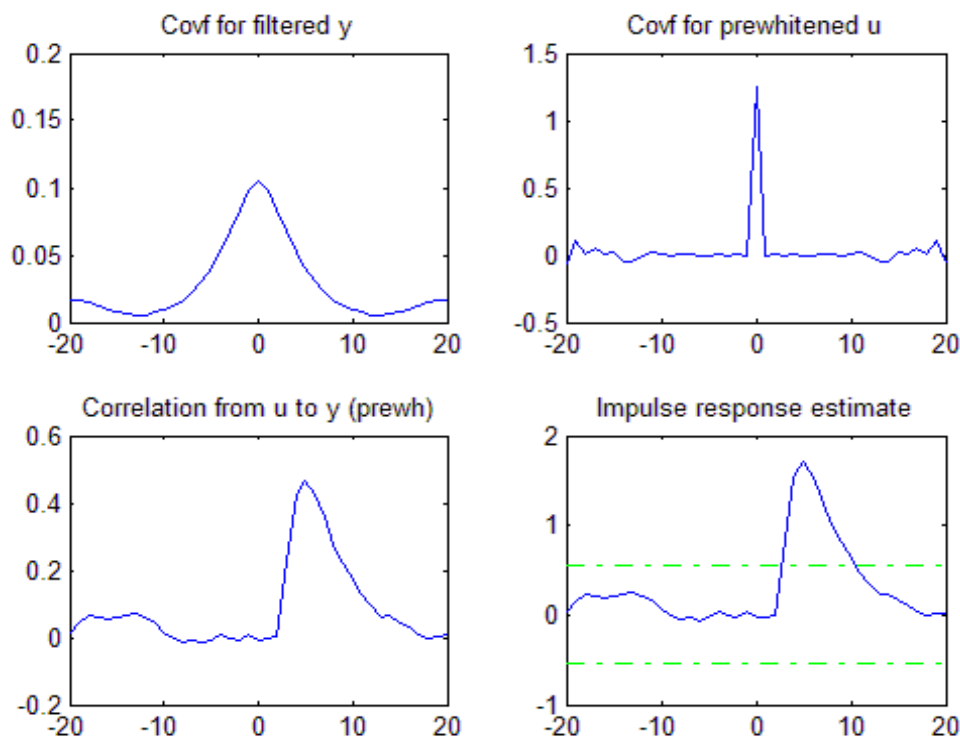


Рис. 3.3. Графики функций: а) АКФ или ковариация (covf) выходного сигнала; б) АКФ (covf) входного сигнала; в) ВКФ (prewh) входного и выходного сигнала; г) импульсная характеристика

Анализируя полученные зависимости на рис. 3.3, следует учесть, что по оси абсцисс откладываются промежутки времени $\tau = t_i - t_{i-1} = ts$ (точнее, количество M задержек t), а по оси ординат значения ИХ и корреляционных функций (АКФ, ВКФ) для входного u и выходного y сигналов.

Корреляционные функции на графиках взяты из матрицы R , в которой первый столбец определяет значение отсчетов во времени, второй и третий столбец – АКФ выходного y и входного u сигнала (четные функции), а последний столбец содержит ВКФ сигналов y и u . ВКФ является, как видим, масштабной копией оценки импульсной характеристики системы на рис. 3.3 г. Видим также, что корреляция осуществляется при положительных значениях t_i , следовательно, входная переменная u имеет зависимость с более поздними значениями выходной переменной y .

Для получения частотных характеристик по экспериментальным данным воспользуемся функциями оценивания частотных характеристик $spa()$ и $bodeplot()$, которые реализуют переход из временной в частотную область на основе преобразования Фурье:

```
gs = spa(ze);
bodeplot(gs).
```

Результатом выполнения является вывод графиков АЧХ и ФЧХ (рис. 3.4).

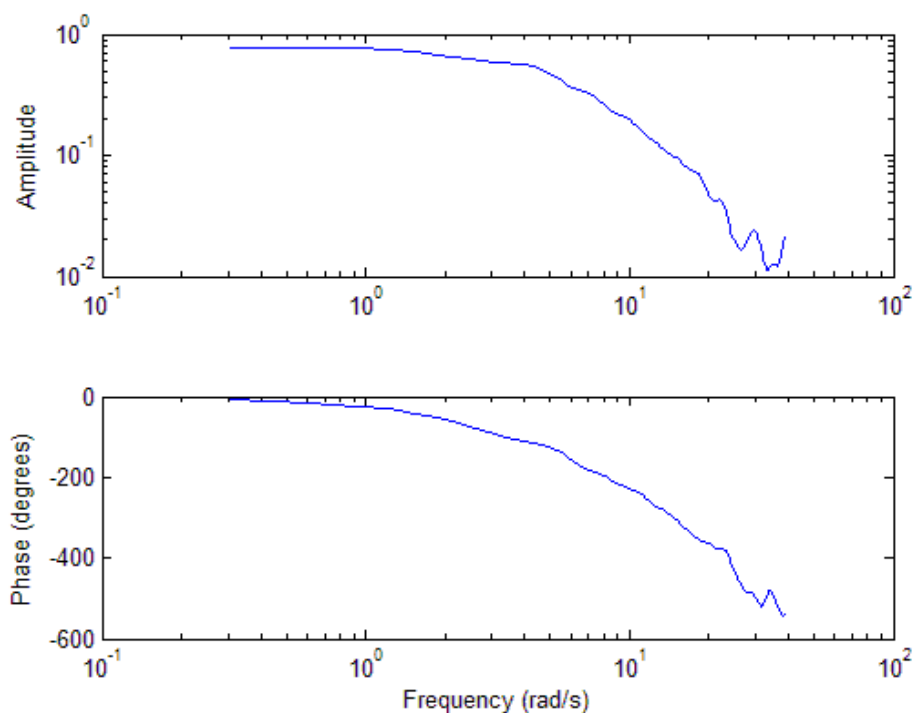


Рис. 3.4. АЧХ и ФЧХ модели, полученные непараметрическим методом *spra()*

3.4.3. Параметрические методы идентификации

Параметрическое оценивание экспериментальных данных проводится с целью определения параметров модели заданной структуры путем минимизации выбранного критерия качества модели. Для проведения параметрического оценивания массив экспериментальных данных в разделе 3.4.1 был разделен условно на две части. Первая из которых («*ze*») будет использоваться непосредственно для параметрической оценки и построения модели ТОУ, а оставшаяся часть – для определения адекватности полученной модели (верификации) и определения погрешностей идентификации.

ARX-модель. Построим ARX-модель, описание которой приводится в разделе 1.3. Параметры модели [na nb nk] (порядок полинома A, порядок полинома B, задержка входа) выберем следующими: $m2 = \text{arg}(ze, [2\ 2\ 3])$.

Известно, что с увеличением порядка полиномов улучшается адекватность модели реальному объекту. Однако при этом увеличивается громоздкость модели и время моделирования, поэтому необходимо определять оптимальный порядок в зависимости от заданной точности моделирования.

Чтобы увидеть построенную модель *m2*, введем название модели в окне командной строки и нажмем клавишу Enter, получим:

```
Discrete-time IDPOLY model:  A(q)y(t) = B(q)u(t) + e(t)
A(q) = 1 - 1.274 q^-1 + 0.3935 q^-2
```



```

B(q) = 0.06662 q^-3 + 0.04448 q^-4
Estimated using ARX on data set ze
Loss function 0.00166284 and FPE 0.00170718
Sampling interval: 0.08 Seconds

```

Переходная характеристика модели $m2$ несущественно отличается от переходной характеристики модели ze , вычисленной по экспериментальным данным:

```

impulse(ze, 'sd', 3);
impulse(m2, 'sd', 3).

```

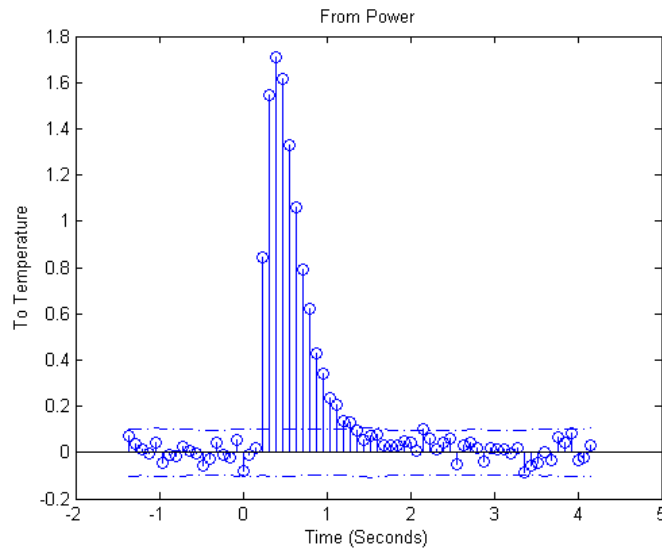


Рис. 3.5. Импульсный отклик системы

Модель в пространстве состояний. Простой способ оценки модели заключается в использовании аппарата моделей пространства состояний. Проведем оценку модели с использованием функции *rem()*:

```
m1 = rem(ze) .
```

Порядок модели определится автоматически (см. табл. 3.1). Для того чтобы посмотреть параметры расчетной модели введем название модели в окне командной строки. В результате получим:

```

m1
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      0.95246  -0.21026  0.053065
x2      0.25434   0.6483   0.23707
x3     -0.051355 -0.66102  0.13742
B =
      Power

```

```

x1    0.00031518
x2     0.01563
x3     0.057909
C =
x1          x2          x3
  Temperature    -14.058    0.094752    0.042548
D =
          Power
Temperature    0
K =
          Temperature
x1    -0.066092
x2     0.0094344
x3     0.092941
x(0) =
x1          0
x2          0
x3          0

```

Чтобы получить отдельные параметры и свойства полученной модели, например, матрицу состояний «*A*» объекта, полученного выше, мы можем использовать оператор «точка» для обращения к полю объекта *m1*:

```
A = m1.a
```

Для того, чтобы узнать, какие поля объекта *m1* могут быть возвращены для дальнейшей работы с ними пользователя, можно воспользоваться командой:

```

get(m1);

ans =
  A: [3x3 double]
  B: [3x1 double]
  C: [-14.0584 0.0948 0.0425]
  D: 0
  K: [3x1 double]
  X0: [3x1 double]
  dA: []
  dB: []
  dC: []
  dD: []
  dK: []
  dX0: []
SSParameterization: 'Free'
As: [3x3 double]
  Bs: [3x1 double]
  Cs: [NaNNaNaN]
  Ds: 0
Ks: [3x1 double]

```

```

X0s: [3x1 double]
StateName: {3x1 cell}
InitialState: 'Auto'
nk: 1
DisturbanceModel: 'Estimate'
CanonicalIndices: 'Auto'
Name: ''
Ts: 0.0800
InputName: {'Power'}
InputUnit: {'Watt'}
OutputName: {'Temperature'}
OutputUnit: {'^o C'}
TimeUnit: 'Seconds'

```

Анализ модели в пространстве состояний. Получим диаграммы Боде при помощи функции *bode()*, как показано ниже, по которым обычно определяют частоту среза, критическую частоту и другие динамические характеристики модели:

```
bode(m1)
```

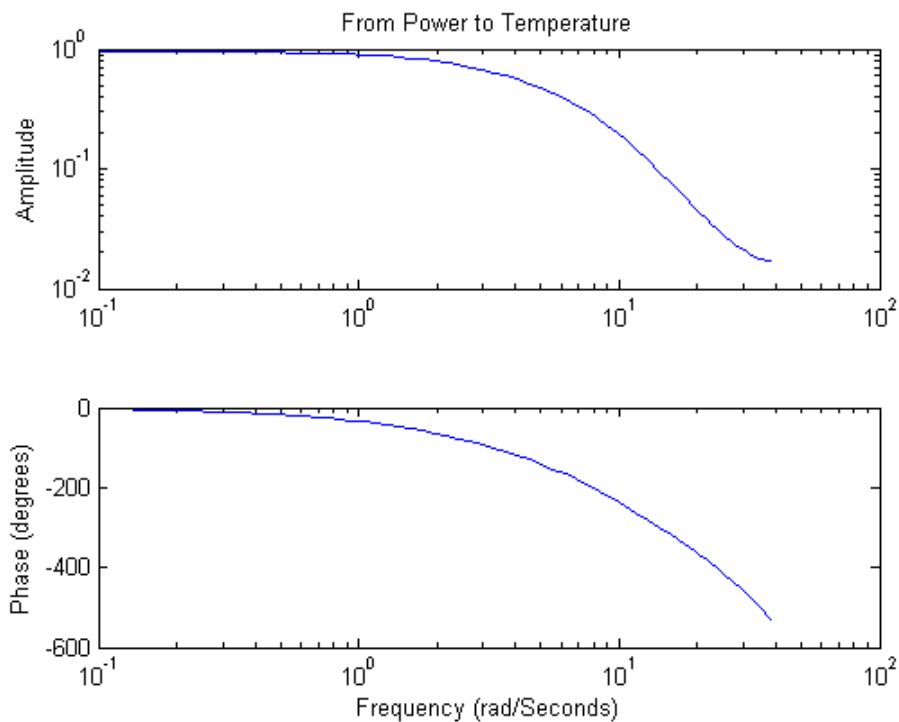


Рис. 3.6. АЧХ и ФЧХ расчетной модели с автоматически выбранным порядком

Так же можно построить годограф Найквиста, воспользовавшись функцией *nyquist()*:

```
nyquist(m1, 'sd', 3)
```

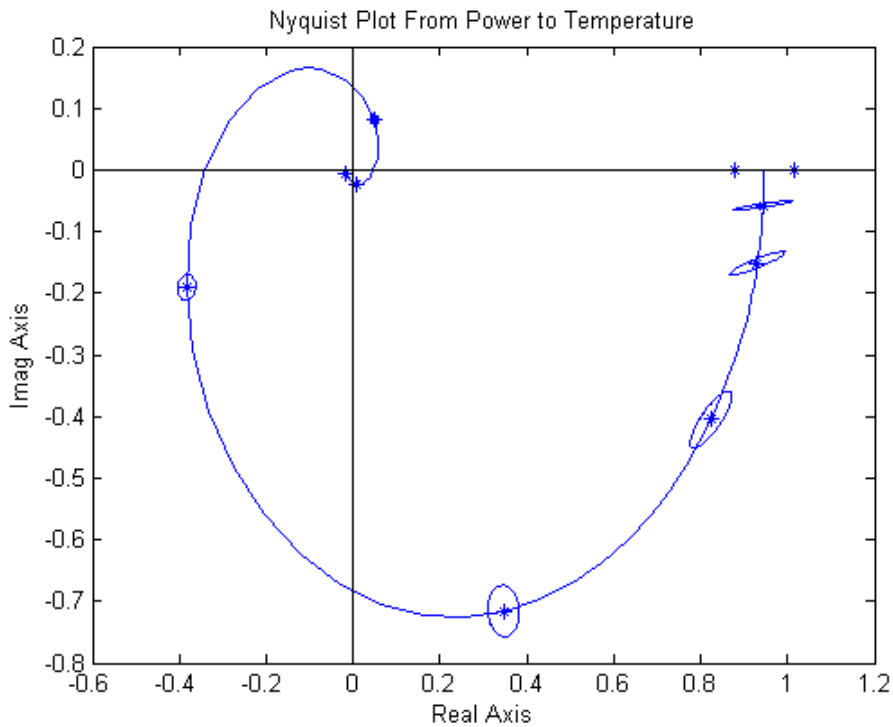


Рис. 3.7. Годограф Найквиста

Видим, что в данном случае система является устойчивой, так как годограф при изменении частоты ω от $-\infty$ до $+\infty$ не охватывает особой точки $(-1, j0)$.

Параметр $sd = 3$ позволяет вывести на графике доверительный интервал, где значение 3 отвечает за его ширину в три среднеквадратических отклонения, называемый интервалом максимально возможных отклонений, в который попадает 99 % случайных отклонений данных. Следовательно, если задан параметр 'sd', то на диаграмме будут точки с выделением эллипсами, в которых истинное значение может быть определено с доверительной вероятностью, соответствующей стандартному среднеквадратическому отклонению или кратное ему (распределение Гаусса). По умолчанию данные точки строятся на каждом десятом значении частоты. Если в качестве параметра «*standarddeviation*» задать, например, 'sd+5', то точки с соответствующей доверительной вероятностью будут строиться на каждой пятой частоте.

Переходная характеристика модели *m1* несколько отличается от переходной характеристики модели *ze*, вычисленной непосредственно из экспериментальных данных:

```
step(m1, 'b', ze, 'r')
```

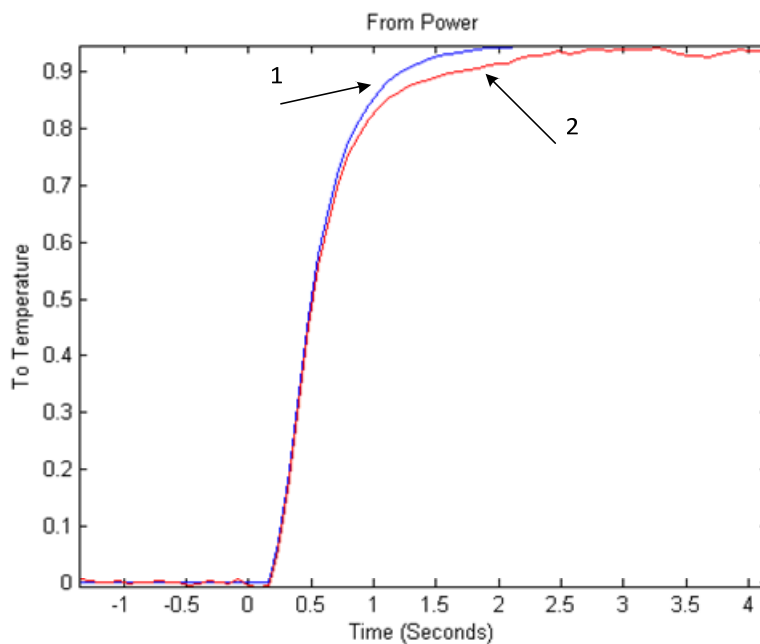


Рис. 3.8. Сравнительные переходные характеристики на основе расчетной модели (голубой цвет 1) и экспериментальных данных (красный цвет 2)

3.4.4. Верификация моделей по экспериментальным данным

Важнейшим этапом идентификации моделей объектов является проверка качества модели по выбранному критерию близости выхода модели и объекта, то есть проверка ее адекватности. В пакете System Identification Toolbox MATLAB в качестве такого критерия принята оценка адекватности модели *fit*, которая рассчитывается по следующей формуле:

$$fit = 100 * (1 - norm(yh - y) / norm(y - mean(y))),$$

где y – измеряемые выходные данные; yh – данные, полученные в результате моделирования; $norm()$ – функция расчета нормы вектора; $mean()$ – функция расчета среднего значения.

Проведем сравнение результатов моделирования с реальными измерительными данными. В качестве измерительных данных воспользуемся частью исходных данных *dyu*, которые не использовались в построении модели (например, точки 800...900). Используем функцию сравнения *compare()*, как показано ниже, чтобы увидеть качество прогноза:

```
%Выбор данных для сравнения
zv = data(800 : 900);
% Удаление постоянной составляющей
zv = detrend(zv);
% Сравнение с экспериментальными данными
compare(zv, m1);
```

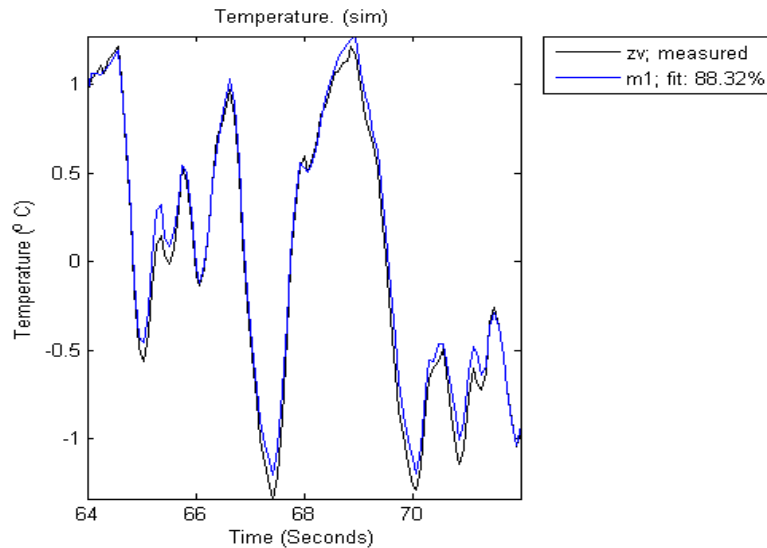


Рис. 3.9. Верификация модели

Функция `compare()` выводит на дисплей сравнительные графики с указанием критерия адекватности модели, выраженного в процентах (параметр `fit, %`). Из графиков можно сказать, что полученная модель довольно точно описывает поведение реального объекта.

Сравним расчетные модели. Для сравнения производительности различных параметрических моделей, например, `m1` и `m2` с моделью экспериментальных данных `zv`, можно снова воспользоваться известной командой сравнения `compare()`:

```
compare(zv, m1, 'b', m2, 'r').
```

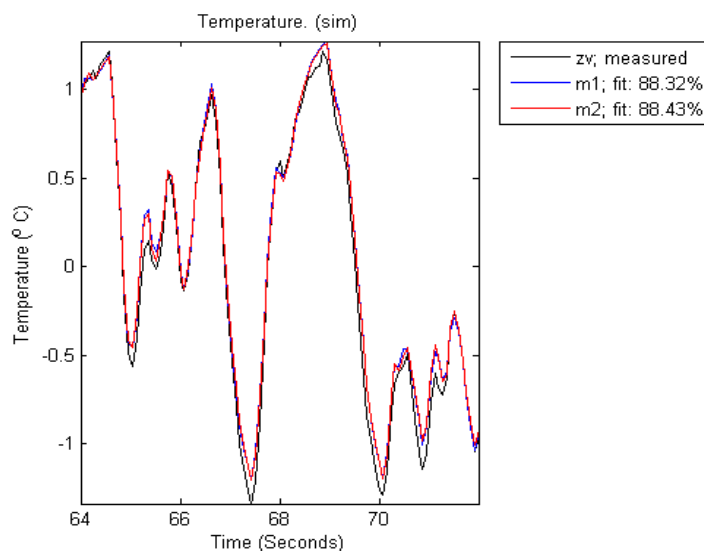


Рис. 3.10. Сравнение моделей `m1` и `m2` с моделью `zv`

Видим, что обе модели `m1` и `m2` дают примерно одинаковые результаты.

В соответствии со свойствами авторегрессионных моделей (см. приложение 3) полюсы стационарной модели должны лежать в окружности единичного радиуса на плоскости параметров, что автоматически гарантирует ее устойчивость, а нахождение хотя бы одного полюса на границе единичной окружности определяет границу устойчивости модели. Карту нулей и полюсов для двух моделей можно получить с использованием функции `pzmap()`, выделив модель *m1* голубым (*b*), а модель *m2* красным (*r*) цветом:

```
pzmap(m1, 'b', m2, 'r')
```

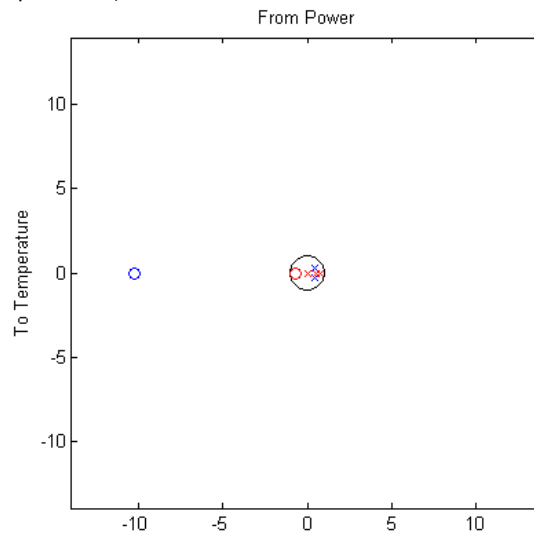


Рис. 3.11. Полюса и нули двух моделей *m1* и *m2* (нули обозначены окружностями)

Для увеличенного представления графика используем следующий оператор:

```
pzmap(m1, 'b', m2, 'r', 'sd', 3, 'axis', 1)
```

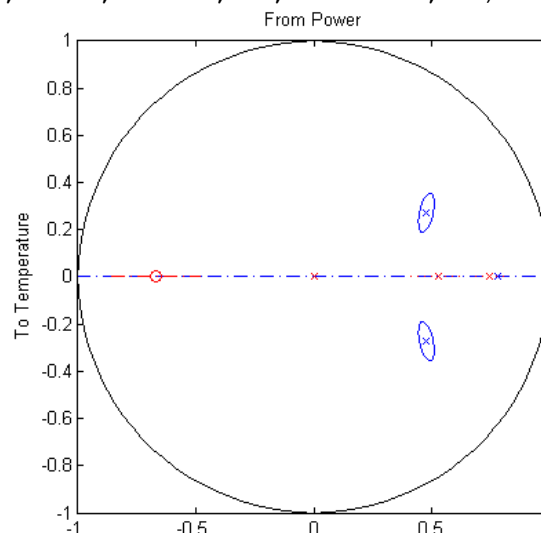


Рис. 3.12. Увеличенное представление нулей и полюсов моделей

Данные графика показывают, что модель действительно является устойчивой. Амплитудно-частотные характеристики (АЧХ) и фазо-частотные характеристики (ФЧХ), соответствующие двум оцениваемым моделям могут отображаться при помощи функции *bode()* в следующем виде:

```
bode(m1, m2) .
```

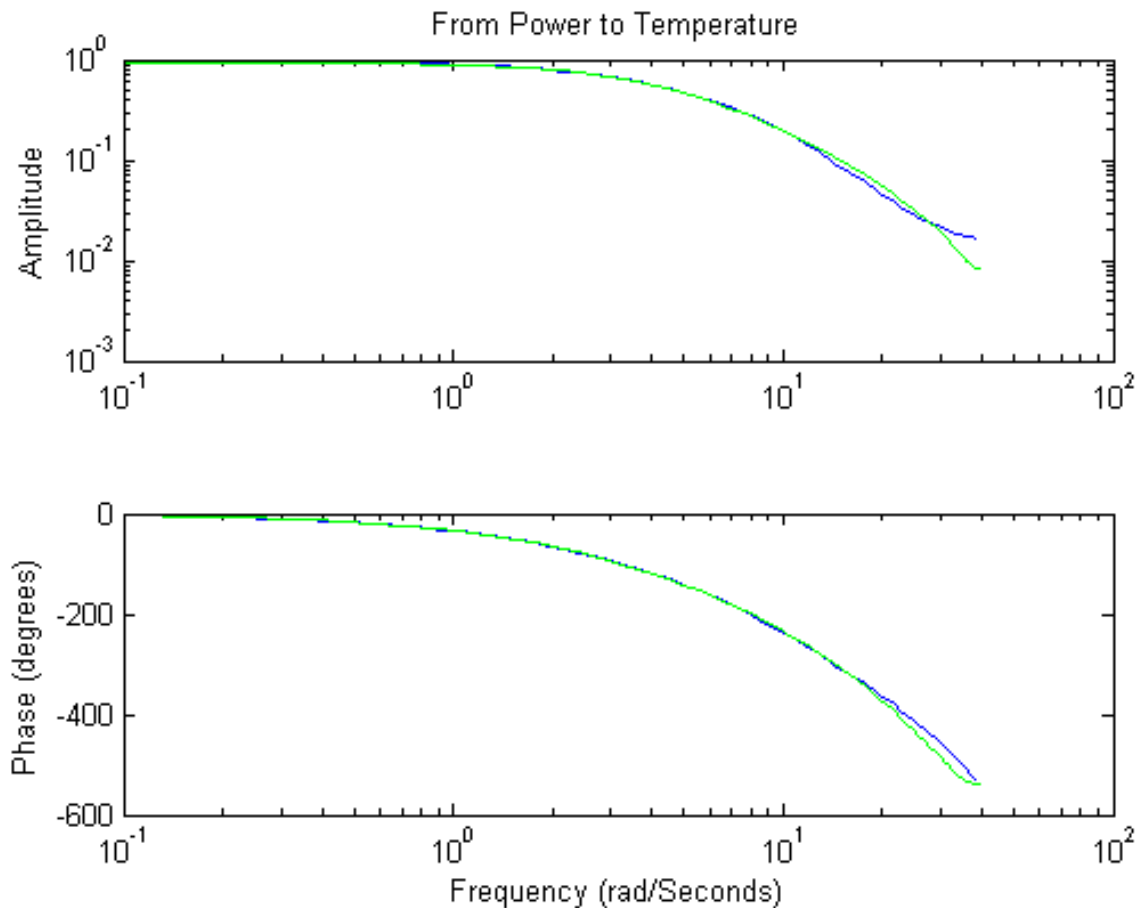


Рис. 3.13. Сравнительные АЧХ и ФЧХ оцениваемых моделей

АЧХ и ФЧХ оцениваемых моделей можно сравнить с моделью «qs», полученной на основе непараметрического спектрального метода (*sps()*), описанного в разделе 3.3. Результаты сравнения представлены на рис. 3.14.

```
bode(m1, m2, gs);
```

```
% «Легенда» графиков - графики именованы графики по порядку их  
% вывода на экран.
```

```
legend('m1', 'm2', 'gs')
```

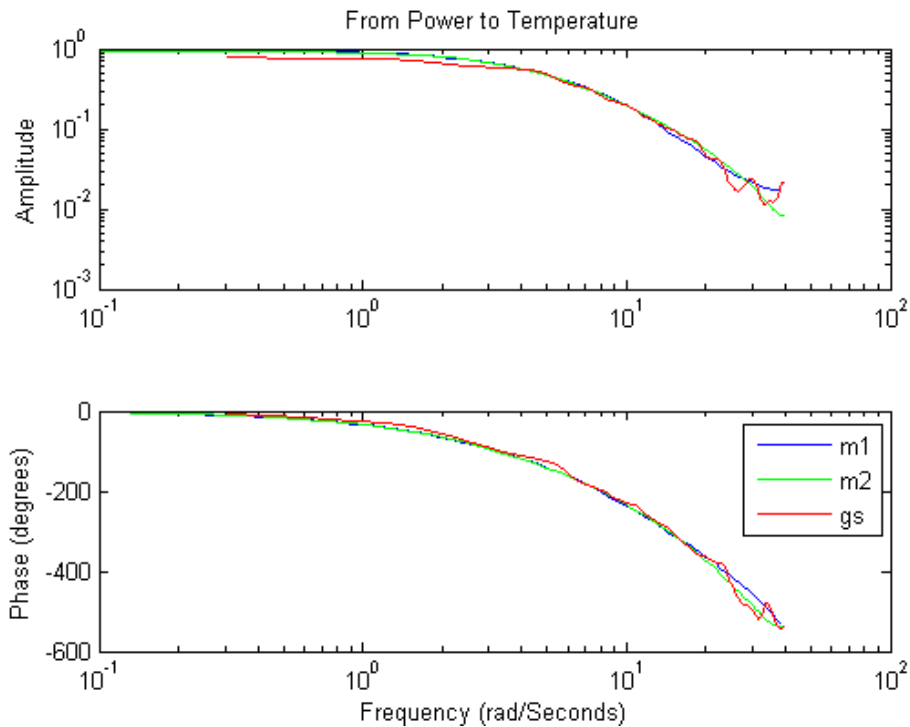



Рис. 3.14. Диаграммы Бode параметрических моделей $m1$ и $m2$ и непараметрической модели спектрального оценивания gs

Видно, что результирующие АЧХ и ФЧХ трех моделей очень близки. Кроме того, с целью оценки точности моделей диаграмма Найквиста также может быть проанализирована в точках, решение в которых можно определить с заданной доверительной вероятностью (выделено эллипсами):

```
nyquist(m1, m2, gs, 'sd', 3)
```

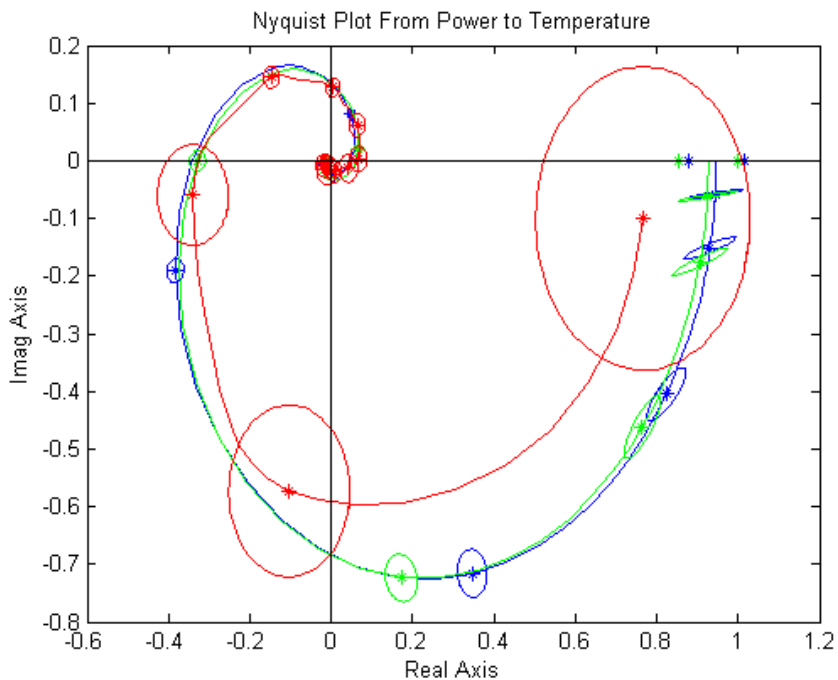


Рисунок 3.15. Годограф Найквиста моделей $m1$, $m2$, gs

Видим, что все модели определяют объект устойчивым, однако непараметрическая модель qs имеет более низкую точность, что видно из увеличенного значения величины доверительного интервала. Параметрические модели $m1$, $m2$ дают примерно одинаковые результаты.

3.5. Задание на лабораторную работу

1. Изучить теоретические сведения.
2. Загрузить и подготовить набор экспериментальных данных «dryer5» при помощи команды `load dryer5`. Данные разбить на две части (для построения модели и для ее верификации) по варианту.
3. При помощи методов непараметрического оценивания получить АКФ входного и выходного сигналов, ВКФ, импульсную характеристику, частотные характеристики. Построить графики, сделать выводы.
4. Построить модель в пространстве состояний PEM (объем выборки определяется по варианту в табл. 2.2). Оценить адекватность полученной модели, используя оставшуюся часть выборки данных.
5. Построить авторегрессионную модель ARX . Параметры na , nb , nk выбрать исходя из варианта. Выявить адекватность полученной модели.
6. Проанализировать полученные модели системы (PEM , ARX), сравнив импульсные переходные характеристики, диаграммы Боде и Найквиста. Привести в описании работы соответствующие диаграммам и характеристикам критерии устойчивости и, следуя им, сделать выводы об устойчивости анализируемой системы.
7. Сравнить полученные модели между собой.
8. Построить карту нулей и полюсов системы по расчетным моделям. Оценить устойчивость системы по конфигурации полюсов.
9. Оформить отчет.

Таблица 2.2

Исходные данные для выполнения лабораторной работы

Вид оценивания	Параметр		Вариант				
	Параметрическое оценивание	Объем выборки	для построения моделей	100–400	300–600	700–1000	200–500
для верификации			700–800	601–700	100–200	600–1000	200–499
Параметры ARX -модели ($na-nb-nk$)		5-2-8	2-4-9	3-7-4	4-1-6	2-3-4	
Непараметрическое оценивание	Величина задержки M для оценки $ИХ$ системы.		10	15	20	25	30
	Порядок AR -модели обеляющего фильтра для оценки $ИХ$ системы.		0	10	5	15	20

Содержание отчета

Отчет оформляется в соответствии с требованиями к оформлению лабораторных работ в вузе, и должен содержать:

- 1) титульный лист;
- 2) формулировку цели работы;
- 3) исходные данные варианта задания, краткие теоретические сведения;
- 4) результаты работы;
- 5) общие выводы;
- 6) приложения (листинг программы с комментариями).

Контрольные вопросы

1. Что понимают под термином «идентификация систем»?
2. Перечислите основные этапы идентификации.
3. Какие существуют методы перехода от непрерывной модели к дискретной и обратно?
4. Объясните разницу между параметрическими и непараметрическими методами идентификации. В чем их преимущества и недостатки?
5. Что такое автокорреляционная и взаимокорреляционная функции?
6. Объясните необходимость «обеления» исходных данных при непараметрической оценке.
7. Изложите суть авторегрессионного метода идентификации.
8. Опишите эксперимент для идентификации во временной области.

4. ПОСТРОЕНИЕ НЕЛИНЕЙНЫХ ARX-МОДЕЛЕЙ

4.1. Цель работы

Получение навыков применения нелинейных ARX-моделей для идентификации динамических объектов. Ознакомление с методикой построения нелинейных ARX-моделей в пакете программ MATLAB.

4.2. Постановка задачи

В реальной жизни линейных объектов и процессов не существует. Все они характеризуются той или иной степенью нелинейности. Существует ряд ситуаций, когда линейной модели не достаточно для адекватного описания поведения объекта. В этом случае необходимо учитывать его нелинейность и, следовательно, строить нелинейную модель. В лабораторной работе использованы демонстрационные и справочные данные MatLabi SystemIdentificationToolbox.

4.3. Справочные сведения по системе MATLAB

System Identification Toolbox предоставляет несколько нелинейных оценок $g(x)$ для нелинейных моделей ARX. Нелинейность формируется в виде вейвлета (wavenet), сигмоидальной сети (sigmoidnet), бинарного дерева (treepartition), многоуровневой нейронной сети (neuralnet), линейной оценки (linear) [12]. По умолчанию используется нелинейная оценка в виде вейвлета.

Известно, что вейвлет-анализ является частью частотно-временного (разновидность спектрального) анализа [10], в котором роль простых колебаний играют функции особого рода, называемые вейвлетами. Базисная функция вейвлет – это некоторое «короткое» колебание, но не только. Понятие частоты спектрального анализа здесь заменено масштабом, и, чтобы перекрыть «короткими волнами» всю временную ось, введен сдвиг функций во времени. Базис вейвлетов – это функции типа $f((t-b)/a)$, где b – сдвиг, a – масштаб. Функция $f(t)$ должна иметь нулевую площадь и осциллирует возле некоторого значения времени $t=b$. Фурье-преобразование таких функций имеет вид полосового фильтра и при различных значениях масштабного параметра 'a' это будет набор полосовых фильтров. Семейства вейвлетов во временной или частотной области используются для представления сигналов и функций в виде суперпозиций вейвлетов на разных масштабных уровнях декомпозиции (разложения) сигналов.

Вид такого вейвлета и имеют нелинейные оценки $g(x)$ моделей ARX

$$g(x) = \sum_{n=0}^N \alpha_n f(\beta_n(x(t - \gamma_n))) \quad (4.1)$$

где $f(z)$ – базисный вейвлет, $z = \beta_n(x(t - \gamma_n))$ – регрессор; α_n – координаты сигнала в выбранном базисе, x – сигнал, участвующий в формировании регрессора, γ_n – временная задержка сигнала x .

Понятие базиса известно студентам направления 200100 «Приборостроение», которые в предшествующем курсе «Преобразование измерительных сигналов» изучают обобщенные ряды Фурье и знают, что в гильбертовом пространстве существует ортогональный нормированный базис сигналов, в котором возможно разложение сигнала в ряд по этому базису, а коэффициенты ряда играют роль координат сигнала в выбранном базисе. Аналогом такого базиса в трехмерной декартовой системе координат являются орты i, j, k , а координатами любого вектора в нем являются проекции вектора на оси x, y, z с вышеуказанными ортами.

Аналогичный нелинейный вид имеет одноуровневая сигмоидальная сеть, в которой:

$$k(z) = (e^z + 1)^{-1}. \quad (4.2)$$

Оценка *linear* обладает линейной зависимостью относительно регрессоров, но в ней возможно задание нелинейных пользовательских регрессоров.

Перечень использованных в работе операторов для формирования нелинейных ARX-моделей приведены в таблице 4.1.

Таблица 4.1

Функция	Синтаксис	Описание
<code>nlarx()</code>	<i>nlarx</i> (данные, порядок модели, вид нелинейности, пользовательские регрессоры, регрессоры в нелинейном блоке)	Функция построения нелинейной ARX модели. В результате создается объект типа «структура» содержащий поля с информацией о параметрах модели. Аргумент <i>порядок модели</i> представляет из себя массив [<i>na nb nk</i>], где параметры модели <i>na, nb, nk</i> для SISO-объекта являются скалярными числами, а для ММО-объекта – матрицами, размерности которых зависят от числа входов и выходов модели
<code>getreg()</code>	<i>getreg</i> (модель)	Функция выдает строку выражений или значений регрессоров в модели <code>idnlarx</code>
<code>compare()</code>	<i>compare</i> (набор контрольных данных, модель 1, ..., модель n)	Функция проверки адекватности полученных моделей
<code>polyreg()</code>	<i>polyreg</i> (модель, степень полинома, взаимодействующие регрессоры)	Функция создания списка пользовательских регрессоров, путем возведения в указанную <i>степень</i> обычных регрессоров используемых в модели с указанием их <i>перекрестного взаимодействия</i>
<code>addreg()</code>	<i>addreg</i> (модель, пользовательские регрессоры)	Функция объединения списка регрессоров имеющейся модели и списка пользовательских регрессоров
<code>customreg()</code>	<i>customreg</i> (@функция, переменные, задержки переменных, векторизация)	Конструктор пользовательских регрессоров. Позволяет в качестве регрессора использовать любые функции с любым взаимодействием переменных. Структуру функции можно задавать перед использованием оператора, при этом в качестве

Функция	Синтаксис	Описание
		<p>формальных входных аргументов, в функцию будут пересылаться переменные указанные оператором <i>customreg</i>, в том порядке, в котором они перечисляются. Например:</p> $F = @(x1,x2,x3)\sin(x1.*x2+x3)$ <p>где (x1, x2,x3) – список <u>формальных</u> входных аргументов функции $\sin(x1.*x2+x3)$, место которых в дальнейшем занимают регрессоры задачи</p>

4.4.Пример построения модели SISO

Файл *icEngine.mat* содержит один набор данных с 1500 значениями входа и выхода двигателя внутреннего сгорания с интервалом дискретизации 0,04 секунды. Вход $u(t)$ это напряжение управления ожидания воздушного клапана (т.е. дроссельная заслонка, управляемая обычно педалью газа в автомобиле, но в данном случае с помощью электромагнитного воздушного клапана), а выход $y(t)$ – частота вращения двигателя [(об/мин)/100]. Данные загружаются и разделяются на набор данных для оценки моделей (*ze*) и набор данных для проверки моделей (*zv*).

```
load icEngine
z = iddata(y, u, 0.04);
ze = z(1 :1000);
zv = z(1001 : 1500);
```

Параметры модели [*na nb nk*] (порядок полинома *A*, порядок полинома *B*, задержка входа), которые также используются в линейной модели ARX, позволяют определить стандартные регрессоры. Выберем параметры модели. Для этого примера будем использовать [*na nb nk*] [4 2 10], что соответствует стандартным регрессорам $y(t-1)$, $y(t-2)$, $y(t-3)$, $y(t-4)$, $u(t-10)$, $u(t-11)$.

Обратите внимание, что используется линейная оценка и, таким образом, полученная модель похожа на линейную модель ARX.

```
m0 = nlarx(ze, [4 2 10], linear);
% Значение «linear» определяет использование
% только линейного блока
```

Следовательно, *m0* – нелинейная модель объекта в линейном приближении. Название входа, выхода и название списка регрессоров этой

модели показано ниже. Обратите внимание, что по умолчанию используются имена 'u1', 'y1'.

```
m0.InputName
ans =
    'u1'
m0.OutputName
ans =
    'y1'
getreg(m0)
Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
```

4.4.1. Создание регрессоров

Простые регрессоры, заданные строкой. Простые пользовательские регрессоры можно задать в виде строки с помощью служебного слова 'customreg':

```
m1 = nlarx(ze, [4 2 10], linear, 'customreg', ...
{'u1(t-10)^2', 'y1(t-1)^2'})
```

Несмотря на использование только линейного расчетного блока «linear», полученная модель является нелинейной, поскольку в качестве параметров используются нелинейные регрессоры.

Общий перечень регрессоров как стандартных, так и пользовательских, возвращается функцией *getreg()*.

```
getreg(m1)
Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
u1(t-10)^2
y1(t-1)^2
```

Давайте сравним две модели на контрольном наборе данных *zv*.

```
compare(zv, m0, m1)
```

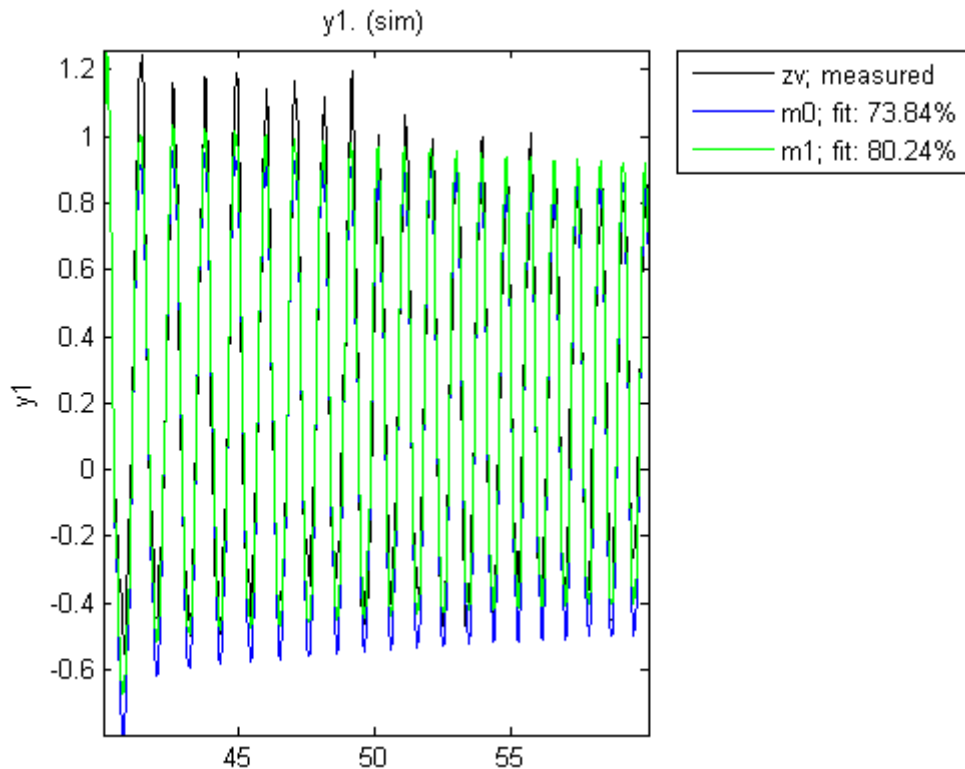


Рис.4.1. График проверки адекватности полученной модели

Как видно из графика, модель *m1*, использующая пользовательские регрессоры, имеет лучшее совпадение с поведением системы, чем модель, построенная с использованием только стандартных регрессоров.

Пользовательские регрессоры могут определяться любым набором предшествующих входных и выходных переменных, и не обязательно соответствующих списку стандартных регрессоров. В следующем примере стандартные регрессоры отсутствуют

```
m2 = nlarx(ze, [0 0 0], 'linear', 'customreg', ...
{'u1(t-10)^3', 'u1(t-11)^3', 'y1(t-1)', 'sin(y1(t-4))'});
getreg(m2)
Regressors:
u1(t-10)^3
u1(t-11)^3
y1(t-1)
sin(y1(t-4))
```

Полиномиальные регрессоры. Полиномиальные регрессоры могут быть созданы с помощью команды *polyreg()*:

```
m3 = nlarx(ze, [4 2 10], 'linear '); %Создание модели
pr = polyreg(m3, 'maxpower', 2, 'crossterm', 'off');
% Создание пользовательского регрессора в виде полинома 2-го
% порядка (maxpower=2) от обычных регрессоров без
% их взаимодействия(crossterm=off')
```



```

%Dобавление пользовательских регрессоров в модель
m3 = addreg(m3, pr);
%Создание новой модели с учетом пользовательских регрессоров
m3 = nlarx(ze, m3);

getreg(m3)
Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
y1(t-1).^2
y1(t-2).^2
y1(t-3).^2
y1(t-4).^2
u1(t-10).^2
u1(t-11).^2

```

Данная операция помогает автоматизировать создание регрессоров, особенно если порядок полинома A и B заранее неизвестен.

Конструктор пользовательского регрессора исследуемого объекта. Использованная ранее форма строки, удобна для создания лишь простых пользовательских регрессоров. При необходимости задания регрессора в виде сложной функции, удобнее воспользоваться конструктором пользовательского регрессора, являющегося более мощным средством для создания произвольных регрессоров.

В следующем примере, регрессор создается как функция *cos* переменной с именем 'u1' и задержкой 10 отсчетов, другими словами: $r1 = \cos(u1(t-10))$. Логическое значение последнего входного аргумента показывает - является пользовательский регрессор вектором (1 или *true*) или скаляром (0 или *false*). Векторизованный (векторный, представленный в виде вектор-столбца) регрессор быстрее в расчетах, поскольку определенные в MATLAB векторные операции, происходят существенно быстрее скалярных, но требует осторожного использования функций первых входных аргументов:

```

r1 = customreg(@cos, 'u1', 10, true)

Custom Regressor:
String expression: cos(u1(t-10))
      Function: @cos
      Arguments: {'u1'}
      Delays: 10
      Vectorized: 1
TimeVariable: 't'

```

В следующем примере участвуют 3 переменные. Обратите внимание, что вместо оператора «*» для векторизованного регрессора используется оператор «.*» (поэлементное умножение).

```
F = @(x1,x2,x3) sin(x1.*x2+x3); % где (x1,x2,x3)
% Список формальных входных аргументов функции sin(x1.*x2+x3),
r2 = customreg(F, {'y1', 'u1', 'u1'}, [1 10 11], true)
% где {'y1', 'u1', 'u1'} - список фактических аргументов
% используемых функцией F с задержками
% [1 10 11] соответственно
```

```
CustomRegressor:
String expression: sin(y1(t-1).*u1(t-10)+u1(t-11))
    Function: @(x1,x2,x3) sin(x1.*x2+x3)
    Arguments: {'y1' 'u1' 'u1'}
    Delays: [1 10 11]
    Vectorized: 1
TimeVariable: 't'
```

Оба созданных пользовательских регрессора можно использовать в модели IDNLARX.

```
m4 = nlarx(ze, [4 2 10], 'linear', ...'customreg', [r1;r2]);
% где [r1; r2] - созданные ранее пользовательские регрессоры
getreg(m4)
```

```
Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
cos(u1(t-10))
sin(y1(t-1).*u1(t-10)+u1(t-11))
```

4.4.2. Использование пользовательских регрессоров для нелинейных оценок

Как правило, пользовательские регрессоры используются для нахождения линейных оценок, как в предыдущих примерах, но их также можно применять и для нелинейных оценок.

Для примера рассмотрим построение нелинейной оценки модели с использованием вейвлета.

```
m5 = nlarx(ze, [4 2 10], 'wavenet', ...'customreg', [r1; r2]);
getreg(m5)
```

```

Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
cos(u1(t-10))
sin(y1(t-1).*u1(t-10)+u1(t-11))

```

Поскольку пользовательские регрессоры, как правило, нелинейные, то образуется избыточность нелинейности, когда нелинейная оценка использует пользовательские нелинейные регрессоры. Чтобы избежать такой избыточности, пользовательские регрессоры можно исключить из нелинейной функции модели IDNLARX, указав, что она должна использовать только стандартные регрессоры. Это значит, что нелинейная функция IDNLARX модели использует только стандартные регрессоры, а линейная функция – как стандартные, так и пользовательские.

```

m6 = nlarx(ze, [4 2 10], 'wavenet', 'customreg',...
[r1;r2], 'nlreg', 'standard');
getreg(m6)

```

```

Regressors:
y1(t-1)
y1(t-2)
y1(t-3)
y1(t-4)
u1(t-10)
u1(t-11)
cos(u1(t-10))
sin(y1(t-1).*u1(t-10)+u1(t-11))

```

Чтобы узнать, какие регрессоры используются в нелинейной функции модели обратимся к полю *nlreg*IDNLARX-объекта, содержащему порядковые номера используемых регрессоров.

```

m6.nlreg

ans =
     1     2     3     4     5     6

```

Различные модели могут быть сопоставлены друг с другом (см. рис. 4.2) следующим образом:

```

compare(zv, m1, m2, m3, m4, m5, m6)

```

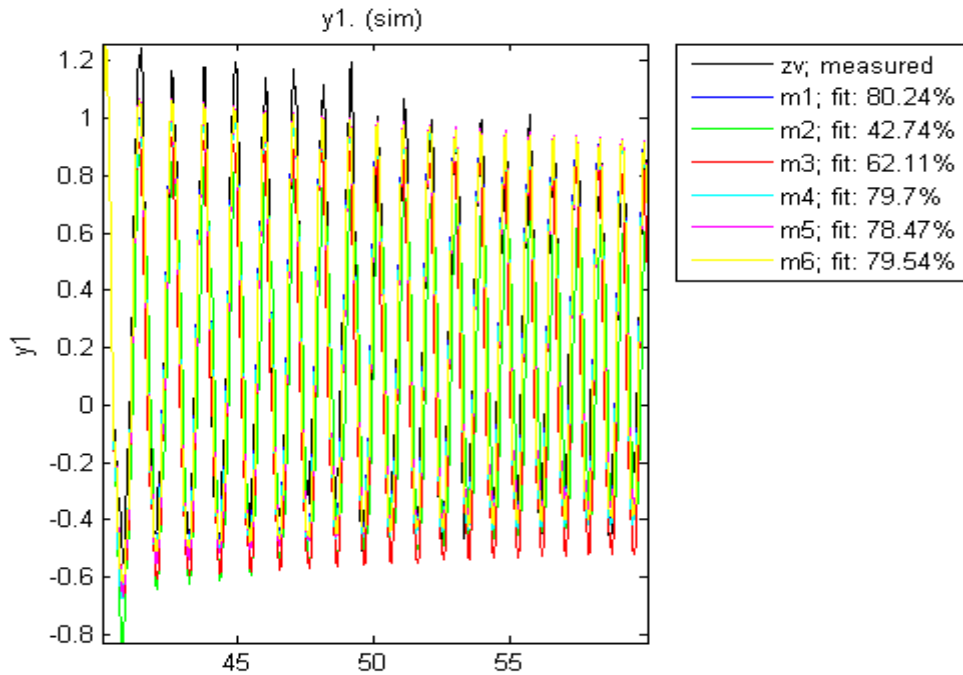


Рис. 4.2. График проверки адекватности полученных моделей

Наилучшее совпадение экспериментальными данными имеет модель *m1*. Хуже всего себя показала модель *m2*, использующая только пользовательские регрессоры. При использовании нелинейных функций оценивания моделей (*m5*, *m6*) исключение пользовательских регрессоров из обработки в нелинейном блоке приносит лучшие результаты (*m6*).

4.5. Пример построения модели ММО

Файл `motorizedcamera.mat` содержит набор данных с 188 выборками данных, собранных при работе моторизованной видеокамеры с частотой дискретизации 0,02 секунды. Входной вектор $u(t)$ состоит из 6 переменных: 3 компоненты скорости в ортогональной системе координат X - Y - Z [м/с], и 3 скорости вращения вокруг осей X , Y , Z [рад/с]. Выходной вектор $y(t)$ содержит 2 переменные: позиция (в пикселях) точки (координаты X и Y), которая является изображением (плоским), снятым на камеру неподвижной точки в 3D-пространстве. Мы создаем объект `IDDATA z` для перевода загруженных данных:

```
loadmotorizedcamera
z = iddata(y, u, 0.02, 'Name', ...
'MotorizedCamera', 'TimeUnit', 's')
```

Использование пользовательских регрессоров в случае ММО не очень отличается от случая SISO. По сути это означает, что каждый выход имеет свой собственный регрессор. Обычно регрессоры всех выходов

собираются в массив ячеек перед вставкой в модель MIMO IDNLARX. Массив имеет размерность $N \times 1$, где N – число выходов, а 1 – число, определенное в примере ниже (2 + 6 + 6).

```
nanbnk = [ones(2,2), 2*ones(2,6), ones(2,6)];
% Создается массив параметров модели, состоящий из 3 матриц,
% причем первая квадратная матрица размер (na x na).
% Количество столбцов для последних двух матриц равно nb и nk
% и определяется числом входов модели (6), а количество строк
% для всех матриц равно na – числом выходов (2) модели, при
% этом, число строк (2) для всех матриц определяется
% количеством выходов модели. Элементы матрицы определяют
% порядок модели или задержку по текущему входу или выходу.
m11 = nlarx(z, nanbnk, 'linear', 'customreg', ...
{{'u1(t-1)^2', 'y2(t-1)^3'};{'u5(t-1)*u6(t-1)'}});
% Поскольку создается модель MIMO-объекта, пользовательские
% регрессоры для каждого выхода системы отделяются
% фигурными скобками {}
getreg(m11)
```

Regressors:

```
nanbnk =
  1  1  2  2  2  2  2  2  1  1  1  1  1  1
  1  1  2  2  2  2  2  2  1  1  1  1  1  1
getreg(m11)
```

Regressors:

Таблица 4.2

For output 1	For output 2
y1(t-1)	y1(t-1)
y2(t-1)	y2(t-1)
u1(t-1)	u1(t-1)
u1(t-2)	u1(t-2)
u2(t-1)	u2(t-1)
u2(t-2)	u2(t-2)
u3(t-1)	u3(t-1)
u3(t-2)	u3(t-2)
u4(t-1)	u4(t-1)
u4(t-2)	u4(t-2)
u5(t-1)	u5(t-1)
u5(t-2)	u5(t-2)
u6(t-1)	u6(t-1)
u6(t-2)	u6(t-2)
u1(t-1)^2	u5(t-1)*u6(t-1)
y2(t-1)^3	

Можно также использовать конструктор пользовательского регрессора. Следующие регрессоры эквивалентны предыдущим, созданным в виде строки, за исключением того, что здесь регрессоры могут быть векторизованы с помощью конструктора.

```
F1 = @(x)x.^2;
r1 = customreg(F1, 'u1', 1, true);
F2 = @(x)x.^3;
r2 = customreg(F2, 'y2', 1, true);
F3 = @(x1,x2)x1.*x2;
r3 = customreg(F3, {'u5','u6'}, [1 1], true);
m12 = nlarx(z, nanbnk, 'linear', 'customreg',...
{[r1; r2], r3});
```

Численная эквивалентность двух моделей можно проверить путем сопоставления их с выходными данными:

```
compare(z, m11, m12)
```

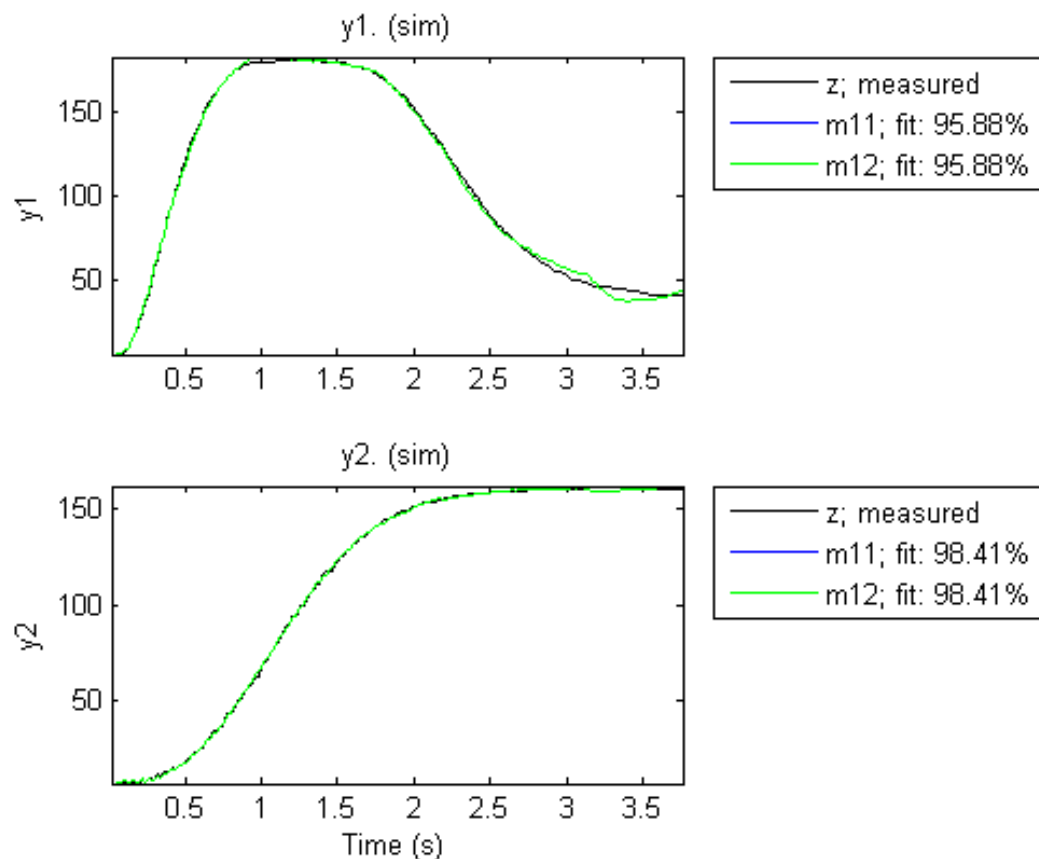


Рис. 4.3. Проверка адекватности полученных моделей

Выбранная модель очень хорошо описывает поведение реальной системы. Так как обе полученные модели *m11* и *m12* по сути одинаковые и имеют только различные способы формирования, они показывают одинаковые результаты.

4.6. Задание на лабораторную работу

1. Изучить теоретические сведения.
2. Загрузить и подготовить набор экспериментальных данных SISO-объекта «icEngine» при помощи команды loadicEngine. Данные разбить на две части (для построения модели и верификации) по варианту (номеру в журнале).
3. Построить линейную и нелинейную ARX-модели с использованием обычных регрессоров. Параметры na , nb , nki вид нелинейности выбрать исходя из варианта. Проанализировать адекватность полученной модели;
4. Построить линейную ARX-модель с нелинейными пользовательскими регрессорами, заданными в виде строки, согласно варианту задания.
5. Построить линейную ARX-модель с полиномиальными пользовательскими регрессорами.
6. Построить линейную ARX-модель с нелинейными пользовательскими регрессорами, сформированными с помощью конструктора.
7. Построить нелинейную ARX-модель с пользовательскими регрессорами.
8. Сравнить полученные модели между собой, сделать выводы.
9. Прodelать операции в п.п. 2...8 для ММО-объекта (набор экспериментальных данных хранится в файле motorizedcamera.mat).
10. Оформить отчет.

Содержание отчета

Отчет оформляется в соответствии с требованиями к оформлению лабораторных работ и должен содержать:

- 1) титульный лист;
- 2) формулировку цели работы;
- 3) исходные данные варианта задания, краткие теоретические сведения;
- 4) результаты работы;
- 5) общие выводы;
- 6) приложения (листинг программы с комментариями).

Контрольные вопросы

1. Что означают параметры na , nb , nk в ARX-модели?
2. В каких случаях может понадобиться построение нелинейной модели?
3. Какие существуют классы нелинейных регрессий?

Исходные данные для выполнения лабораторной работы представлены в таблице 4.3.

Таблица 4.3

Вариант		1	2	3
Объем выборки	для построения модели	200–800	1–700	100–1000
	для контроля	1100–1300	800–1050	1001–1400
Параметры ARX-модели		2–3–4	5–2–8	4–1–6
Пользовательские регрессоры	При использовании совместно с обычными	$u^2(t-5);$ $y^2(t-1).$	$u(t-6)*y^2(t-2);$ $y^3(t-1).$	$u^3(t-8);$ $y^2(t-4).$
	При использовании без обычных	$u^2(t-2);$ $u(t-6);$ $u^4(t-6);$ $y^3(t-1).$	$u^3(t-1);$ $u(t-2);$ $\cos^2(y(t-1));$ $y(t-2).$	$u^2(t-2);$ $u^3(t-3);$ $y^3(t-1);$ $\sin(y(t-4)).$
Функции для конструктора пользовательских регрессоров		$u(t-2)*u^2(t-5);$ $\sin^2(y(t-1));$ $u^3(t-3)*e^{u(t-1)*y(t-2)}$	$u(t-1)*y^2(t-2);$ $\cos(u^3(t-3));$ $y^2(t-1)*tg(u(t-2)).$	$y^3(t-2)*y^2(t-3);$ $2*tg(5*y(t-2));$ $\sin^3(u(t-1)*u^3(t-3)).$
Вид нелинейности		wavenet	sigmoidnet	wavenet

Продолжение табл. 4.3

Вариант		4	5
Объем выборки	для построения модели	900–1500	500–1200
	для контроля	100-400	1–300
Параметры ARX-модели		2–4–9	3–7–4
Пользовательские регрессоры	При использовании совместно с обычными рег.	$y(t-1)*y(t-2);$ $y^2(t-1).$	$u(t-4)*u^2(t-7);$ $u^3(t-9).$
	При использовании без обычных рег.	$u^2(t-3);$ $y(t-1);$ $y^2(t-2);$ $e^{2*y(t-3)}.$	$\sin(u^3(t-1));$ $u^2(t-4);$ $y^3(t-2);$ $y^2(t-3)$
Функции для конструктора пользовательских регрессоров		$u^3(t-1)*y(t-1);$ $2*\sin(u^2(t-2));$ $2*\cos(y(t-1)*u^3(t-1)).$	$u^4(t-1)*u^2(t-8);$ $tg^3(u^2(t-3));$ $y^2(t-2)*e^{u^2(t-2)*y(t-3)}$
Вид нелинейности		sigmoidnet	wavenet

5. ИДЕНТИФИКАЦИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ ДАННЫХ ЧАСТОТНОЙ ОБЛАСТИ

5.1. Цель работы

Ознакомиться с функциями и методами обработки частотных данных в `SystemIdentificationToolbox`.

5.2. Постановка задачи

Здесь представлены специализированные средства для идентификации линейных динамических систем по их частотному отклику. Частотные методы направлены на идентификацию непрерывных систем и являются дополнением к более традиционным дискретным. Используя частотный подход, можно добиться построения лучшей модели, избежать ошибок дискретизации, выделять постоянную составляющую сигнала, существенно улучшить отношение сигнал/шум. `Toolbox` обеспечивает идентификацию непрерывных и дискретных линейных стационарных систем, автоматическую генерацию входных сигналов, а также графическое изображение нулей и полюсов передаточной функции результирующей системы.

Экспериментальные данные, полученные в частотной области (`Frequency Domain Data`), широко распространены. Часто подобные данные получают в виде частотных характеристик различных процессов с использованием частотного (спектрального) анализатора. Однако, иногда, бывают случаи, когда часть данных удобно получить во временной области, и применить к ним Фурье преобразование (FT), чтобы весь массив данных был однородно представлен в частотном виде. В этом случае в `SystemIdentificationToolbox` данные частотной области представляются аналогично данным во временной области, то есть используются *iddata* объекты, при создании которых параметр *'domain'* указывается равным *'Frequency'*. Данные в частотной области не могут быть использованы для оценки нелинейных моделей.

В лабораторной работе требуется провести идентификацию нескольких динамических объектов с использованием данных частотной области. В работе использованы демонстрационные и справочные данные `MatLabi SystemIdentificationToolbox`.

5.3.Справочные сведения

5.3.1. Описание используемых функций

Описание основных функций, используемых в данной лабораторной работе, приводится в таблице 5.1.

Таблица 5.1

Функция	Синтаксис	Описание
loglog()	<i>loglog</i> (x, y)	Функция равносильна функции <i>plot</i> , за исключением того, что она использует по обеим осям логарифмический масштаб
semilogx()	<i>semilogx</i> (x, y)	Команда <i>semilogx</i> использует логарифмический масштаб по оси x и линейный масштаб по оси y
idfrd()	<i>sys = idfrd</i> (response, frequency) <i>sys = idfrd</i> (response, frequency, Ts)	Конструктор IDFRD объекта. Аргументы функции: <i>frequency</i> – вектор-строка частот; <i>response</i> – вектор-строка значений комплексного коэффициента передачи; <i>Ts</i> – интервал дискретизации исходных данных по времени
bode()	<i>bode</i> (sys) <i>bode</i> (sys, w)	Функция возвращает диаграммы Бode – графики ЛАЧХ и ФЧХ. Аргументы функции: <i>w</i> – диапазон частот ($w = [wmin, wmax]$); <i>sys</i> – имя исследуемой модели
oe()	<i>sys = oe</i> (z, nn)	Функция оценивает параметры ОЕ-модели. <i>sys</i> – имя (идентификатор) построенной модели; <i>z</i> – матрица экспериментальных данных; <i>nn</i> – содержит информацию о степенях полиномов модели $B(z)$, $F(z)$ и о величине задержки <i>nk</i> . Задается в виде $nn = [nb, nf, nk]$
pem()	<i>sys = pem</i> (z,nn)	Функция оценивает параметры обобщенной многомерной линейной модели вида: $A(z)y(t) = \frac{B_1(z)}{F_1(z)}u_1(t - nk_1) + \frac{B_2(z)}{F_2(z)}u_2(t - nk_2) + \dots + \frac{B_{nu}(z)}{F_{nu}(z)}u_{nu}(t - nk_{nu}) + \frac{C(z)}{D(z)}e(t)$ <i>sys</i> – имя (идентификатор) построенной модели; <i>z</i> – матрица экспериментальных данных; <i>nn</i> –задается в виде $nn=[na, nb, nc, nd, nf, nk]$, где <i>na, nb,nc,nd, nf, nk</i> – в многомерном случае являются вектор-столбцами

Продолжение табл. 5.1

Функция	Синтаксис	Описание
procest()	$sys = procest (data,type)$	Функция оценивает модель процесса с использованием временных или частотных данных. Имеет следующие аргументы: sys – имя (идентификатор) построенной модели; $data$ – вектор экспериментальных данных; $type$ – параметр, определяющий структуру модели
compare()	$compare (data, sys)$	Функция проверки адекватности модели. Производит сравнение выхода модели sys с экспериментальными данными $data$ с выводом на дисплей сравнительных графиков с указанием критерия адекватности
spafdr()	$g = spafdr (data)$ $g = spafdr (data, resol, w)$	Функция оценки частотной характеристики с переменным разрешением по частоте. Аргументы функции: $data$ – экспериментальные данные; $resol$ – разрешение по частоте (по умолчанию 100); w – вектор частот
spa()	$G = spa (data)$ $G = spa(data, winSize, freq)$ $G = spa (data, winSize, freq, MaxSize)$	Функция оценки частотной характеристики с фиксированным разрешением по частоте в неопределенном или в определенном диапазоне частотвектора $freq$, при этом $winSize$ – скалярная величина, устанавливающая размер окна Ханна для оконного преобразования Фурье, а $MaxSize$ – положительное число, которое подразделяет все данные на группы в количестве не превышающем $MaxSize$, что повышает скорость вычислений
iddata()	$data = iddata (y, u, Ts)$ $data = iddata(y, u, Ts, 'Frequency' , W)$	Функция-конструктор, создает объект типа IDDATA, для которого доступны все функции и команды анализа систем. Он включает в себя: u, y – входные и выходные экспериментальные данные; Ts – интервал дискретизации; ' $Frequency$ ' – идентификатор использования данных в частотной области ($domain='Frequency'$); W – вектор частоты
fft()	$fft (data)$	Функция быстрого преобразования Фурье исходных данных $data$
ifft()	$ifft (data)$	Функция обратного быстрого преобразования Фурье данных $data$

5.3.2. Выбор структуры модели при использовании функции *procest*

В данном случае, модель строится с использованием следующего конструктора (см. табл. 5.1):

```
sys = procest(data, type),
```

где параметр *type* – это спецификатор, который определяет структуру модели [15] и строится по следующей схеме:

P_k – структура модели имеет k -полюсов, причем k может быть задано в количестве 0, 1, 2, или 3;

Z – структура модели имеет нули ($T_z \neq 0$). Как следует из условия физической реализуемости динамической системы [9, 10] порядок знаменателя передаточной функции должен быть больше порядка числителя и, следовательно, модель процесса, не имеющая полюсов, не может иметь нулей (при $k = 0$);

D – модель процесса имеет задержку (запаздывание) ($T_d \neq 0$);

I – модель процесса включает в себя интегратор;

U – модель процесса имеет затухание (underdamped) и, следовательно, включает в себя пару комплексных полюсов.

Приведем пример создания структуры модели с использованием таких идентификаторов [16]:

– спецификатор вида «P1D» определяет модель с одним полюсом и задержкой:

$$sys = \frac{K_p}{1 + T_{p1}s} e^{-T_d s},$$

где K_p , T_{p1} , и T_d – идентифицируемые параметры модели;

– «P2U» создает модель процесса с парой комплексных полюсов (затухающую модель):

$$sys = \frac{K_p}{(1 + 2\zeta T_w s + (T_w s)^2)},$$

где K_p , T_w , и $Zeta$ (т.е. ζ) – идентифицируемые параметры модели;

– «P3ZDI» создает модели с тремя полюсами. Все полюсами являются действительными, потому что строка не содержит символа U . Эта модель также включает в себя нуль, задержку и интегратор:

$$sys = K_p \frac{1 + T_z s}{s(1 + T_{p1}s)(1 + T_{p2}s)(1 + T_{p3}s)} e^{-T_d s}.$$

5.4. Пример выполнения лабораторной работы

5.4.1. Подготовка данных в частотной области

Загрузим mat-файл с частотными данными – файл *demofr*:

```
loaddemofr
```

Данный демонстрационный комплексный mat-файл был разработан компанией MathWorks целью ознакомления пользователей с приемами и методами обработки данных в частотной области SystemIdentificationToolbox, и является универсальным, так как включает в себя максимально широкий спектр экспериментальных данных, таких как:

1) Данные амплитудно-фазо-частотной (АФЧХ) характеристики (переменные *AMP*, *PHA*, *W*):

- «*AMP*»– вектор-строка амплитуды сигнала (1000 значений);
- «*PHA*»– вектор-строка фазовых характеристик (1000 значений);
- «*W*»– вектор-строка частоты(1000 значений).

2) Набор данных во временной области, собранный в объекте IDDATA для удобства его использования, под названием «*ztime*».

3) Набор данных во временной области, включающий в себя 501 значение некоторой измеряемой выходной величины *Y* и входной величины *U*, к которым применено преобразование Фурье.

Данные частотных характеристик можно представить в виде комплексных векторов частотного коэффициента передачи или просто как АФЧХ. Для этого случая в *System Identification Toolbox* существуют IDFRD объекты, которые являются инструментом для накопления АФЧХ, где пользователь указывает данные и вектор частоты. Полученные IDDATA или IDFRD объекты могут быть использованы для оценки моделей любой сложности (например, ОЕ, РЕМ, idproc и т.д.). Оценка и проверка качества модели с использованием данных частотной области проводится аналогично данным, полученным во временной области, что дает определенную гибкость в оценке и анализе моделей, при этом можно одновременно оценить обе модели, сравнить или объединить их.

Для начала выведем эти данные на экран. Воспользуемся функциями *loglog()* и *semilog()*.

```
subplot(211), loglog(W, AMP), title('Amplitude Response');  
subplot(212), semilogx(W, PHA), title('Phase Response').
```

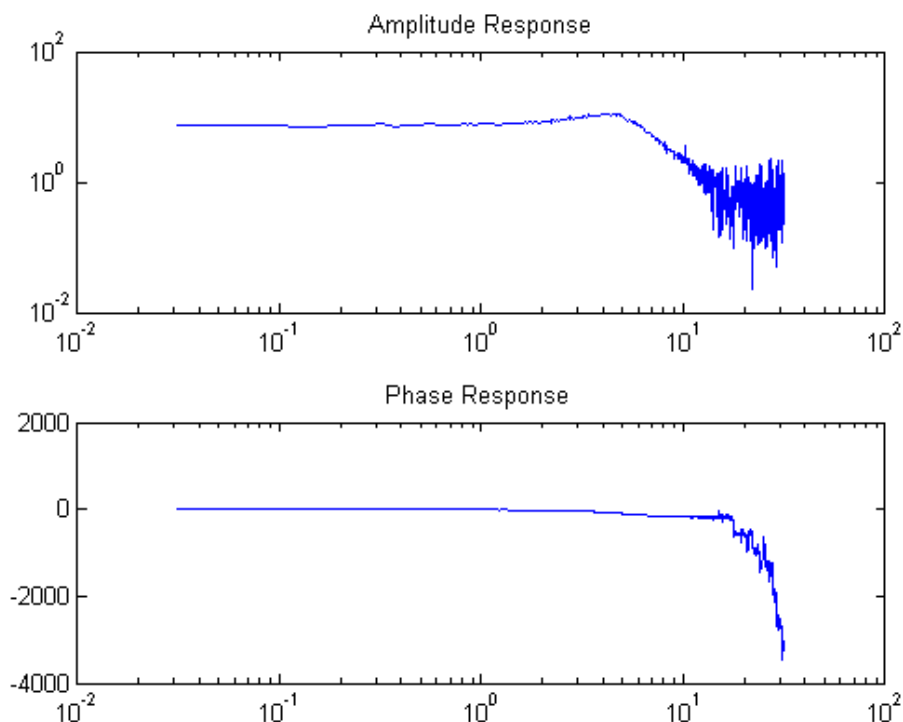


Рис. 5.1. Графики амплитуды АМР и фазы РНА сигнала от частоты

Теперь сохраним эти экспериментальные данные в виде IDFRD объекта с использованием функции *idfrd()*. Модель в форме *idfrd* характеризует систему в частотной области и хранит в себе частотную характеристику сигнала. Сформируем частотный коэффициент передачи *zfr* в показательной форме:

```
%zfr - фактический параметр, который передается в функцию-
% конструктор idfrd и определяет формальный параметр response.
zfr = AMP.*exp(1i*PNA*pi/180);
% Временная дискрета с размерностью секунда (s).
Ts = 0.1;
%Получен идентификационный объект частотной области idfrd.
gfr = idfrd(zfr, W, Ts);
```

Отсюда видим, что *frd*—это объект частотной области, *aidfrd*—идентификационный объект частотной области, построенный на значениях комплексного коэффициента передачи. Объект IDFRD включает в себя дополнительные опции для получения различной дополнительной информации об объекте, например, такой как спектры, которые не доступны при использовании FRD объектов.

5.4.2. Анализ и обработка данных в частотной области

Теперь IDFRD-объект *gfr* содержит данные, которые могут быть обработаны и проанализированы разными способами. Для просмотра данных, можно воспользоваться функцией *bode()*:

```
clf
bode(gfr), legend('gfr')
```

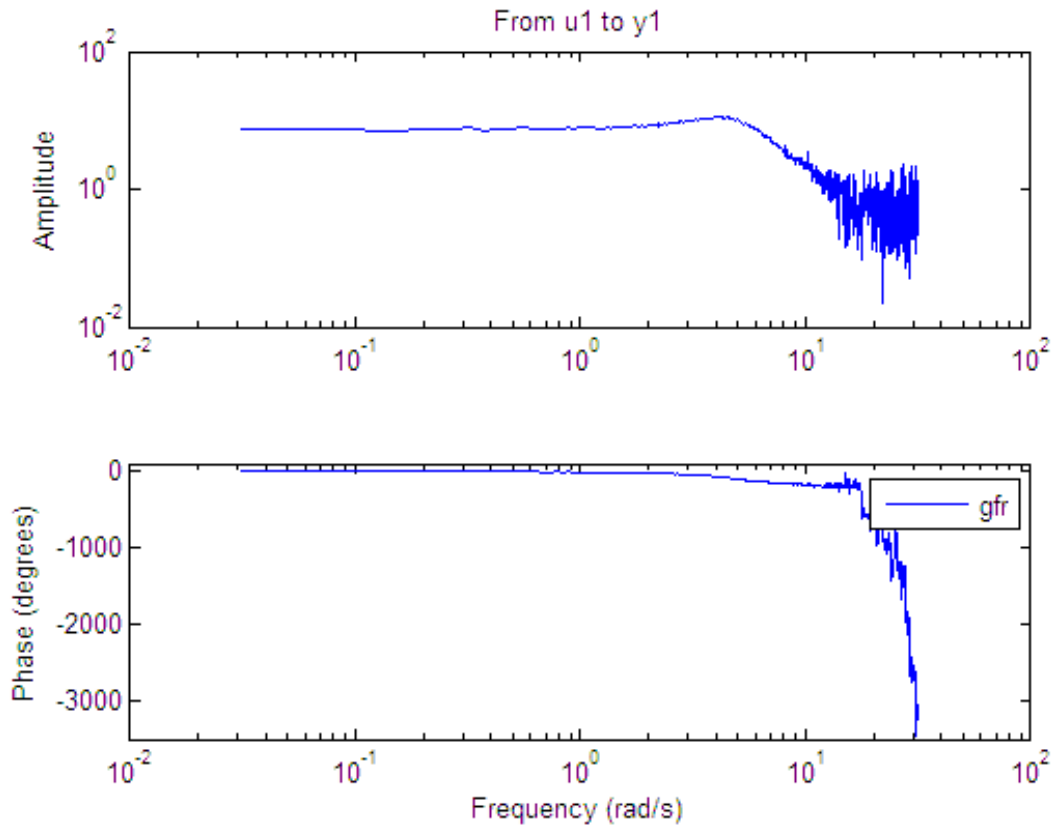


Рис. 5.2. Частотные характеристики *gfr*-объекта

Теперь данный объект можно представить в виде полиномиальной модели вида ОЕ, модели в пространстве состояний РЕМ, или непрерывно-временной модели PROCEST. Единственным ограничением является невозможность построения зашумленных моделей.

```
% Дискретная модель «вход-выход»или модель «выход-ошибка»,
m1 = oe(gfr,[2 2 1])
% при этом [2 2 1] - аргументы[nb, nf, nk] - соответствующие
% степени полиномов B(z),F(z)и задержкенk модели ОЕ.
% Непрерывная модель в пространстве состояний
ms = rem(gfr)
% Непрерывная модель 2-го порядка с «подавленными» полюсами
mproc = procest(gfr, 'P2UDZ')
compare(gfr, m1, ms, mproc)
```

Модель *mproc* (притуре = «P2UDZ»)реализует передаточную функцию, содержащую «два комплексных полюса + задержка + один нуль»:

$$sys = K_p \frac{1 + T_z s}{1 + 2\zeta T_\omega s + (T_\omega s)^2} e^{-T_d s}$$

В результате получены три модели, описывающие объект *gfr-m1*, *ms*, *mproc*, которые можно сравнить с помощью функции *compare()*:

```

m1 =
Discrete-time OE model:  y(t) = [B(z)/F(z)]u(t) + e(t)
B(z) = 0.9982 z^-1 + 0.4974 z^-2
F(z) = 1 - 1.499 z^-1 + 0.6998 z^-2
Sample time: 0.1 seconds
Parameterization:
Polynomial orders:  nb=2  nf=2  nk=1
  Number of free coefficients: 4
  Use «polydata», «getpvec», «getcov» for parameters and
  their uncertainties.
Status:
Estimated using OE on frequency response data «gfr».
Fit to estimation data: 88.04% (prediction focus)
FPE: 0.2501, MSE: 0.2494
ms =
Continuous-time 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
x1  -0.9005   6.636
x2  -3.308   -2.669
B =
      u1
x1  -32.9
x2  -28.33
C =
      x1      x2
y1  -0.5073   0.499
D =
      u1
y1   0
K =
      y1
x1   0
x2   0
Parameterization:
  FREE form (all coefficients in A, B, C free).
Feedthrough: none
  Disturbance component: estimate
  Number of free coefficients: 10
  Use «idssdata», «getpvec», «getcov» for parameters and
  their uncertainties.
Status:
Estimated using SSEST on frequency response data «gfr».
Fit to estimation data: 73.16% (prediction focus)
FPE: 0.2511, MSE: 1.255

```



```

mproc =
Process model with transfer function:
          1+Tz*s
G(s) = Kp * ----- * exp(-Td*s)
          1+2*Zeta*Tw*s+(Tw*s)^2
Kp = 7.4612
    Tw = 0.20246
    Zeta = 0.36238
    Td = 0
Tz = 0.013617
Parameterization: 'P2DUZ'
  Number of free coefficients: 5
  Use «getpvec», «getcov» for parameters and their
  uncertainties.
Status:
Estimated using PROCEST on frequency response data «gfr».
Fit to estimation data: 73.2% (prediction focus)
FPE: 0.2504, MSE: 1.251

```

Обратите внимание, что функция `mproc = procest(gfr, 'P2UDZ')` создает непрерывную модель в виде передаточной функции, в которой параметр `U` отвечает за наличие или отсутствие (если параметра нет) комплексных полюсов. После расчета параметров моделей функция `compare()` выводит на дисплей сравнительные графики с указанием оценки адекватности модели (параметр `fit, %`), что и позволяет сравнить выходы моделей и объекта.

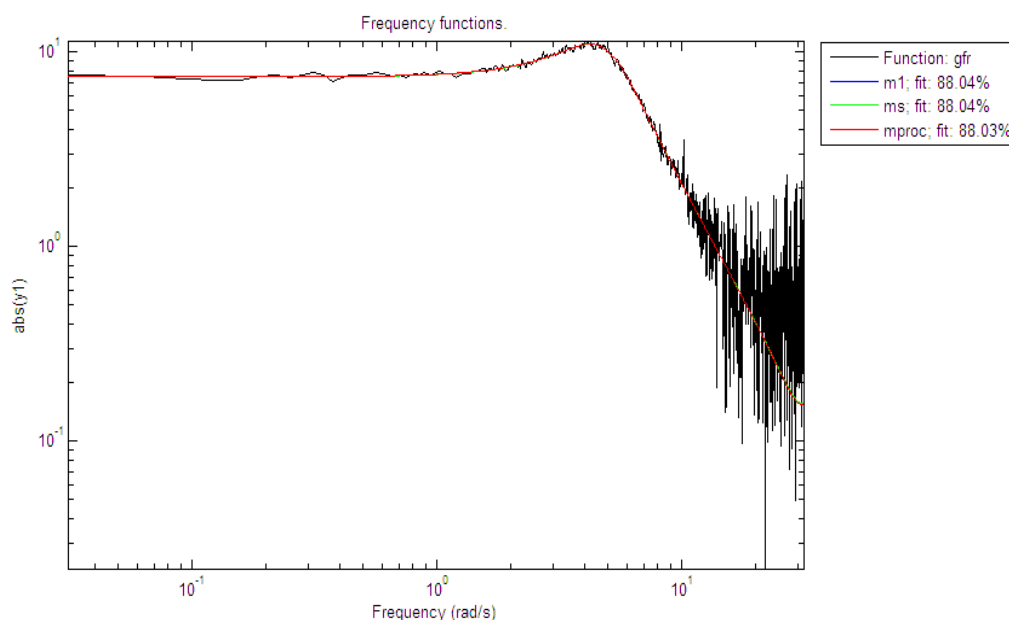


Рис. 5.3. Результаты сравнения полученных моделей

Как уже упоминалось, различные типы линейных моделей могут оцениваться в непрерывных или дискретных областях на основе временных и частотных данных. Выше были использованы данные в

частотной области, поэтому сейчас, для верификации полученных результатов используем данные, полученные во временной области. Набор данных во временной области, соответствующий исследуемому объекту, обозначен как *ztime* и содержится в загрузенном *mat*-файле *demofr*. Для верификации воспользуемся командой *compare()* следующим образом:

```
% Верификация во временной области  
compare(ztime, m1, ms, mproc)
```

Результаты сравнения моделей представлены на рис. 5.4.

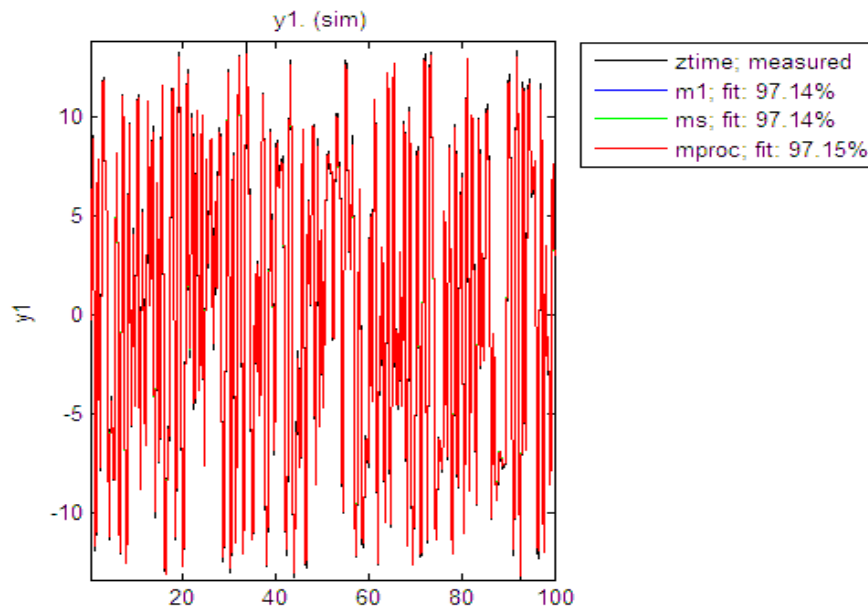


Рис. 5.4. Верификация моделей на основе временных данных

Исходя из результатов, представленных на рисунке 5.4, видим, что все полученные нами модели довольно точно описывают поведение исследуемого объекта (параметр *fit* не ниже 97 %).

5.4.3. Сжатие данных

Важной особенностью при работе с данными в частотной области является то, что их можно сжимать (редукция данных) с незначительной информационной потерей. Команда *spafdr()* позволяет производить сжатие-сглаживание данных, например, с логарифмическим шагом. Приведем пример сжатия данных, содержащихся в объекте *gfrs* разрешением, обеспечивающим 100 значений частоты с логарифмическим изменением шага (принято по умолчанию). Здесь разрешение – это число элементарных дискретных значений в исходных данных на один отсчет преобразованных данных, т.е. при равномерном шаге сжатия разрешение равнялось бы 1/10 исходных значений, что естественно хуже.

Аналогично, можно сжимать данные и во временной области *ztime*.

```
% Спектральная оценка на основе сжатия частотно-зависимым
% логарифмическим шагом
sgfr = spafdr(gfr)
% Спектральная оценка на основе сжатия временных данных
sz = spafdr( ztime );
% Очистка диаграмм
clf
bode(gfr, sgfr, sz )
% Построение диаграммы Боде. Задание границ по осям X и Y.
axis([pi/100 10*pi, -272 105])
% Т.е. ось частоты будет иметь границы от- pi/100 до 10*pi
% рад/с, а ось фазы -272 до 105 градусов.
legend('gfr(rawdata)', 'sgfr', 'sz', 'location', 'southwest')
% «Легенда» графиков - графики именуются графики по порядку их
% вывода на экран.
```

Сжатые данные *sgfr* и *sz* представим для наглядности на диаграмме Боде на совмещенных осях с исходными данными *gfr*(рис. 5.5):

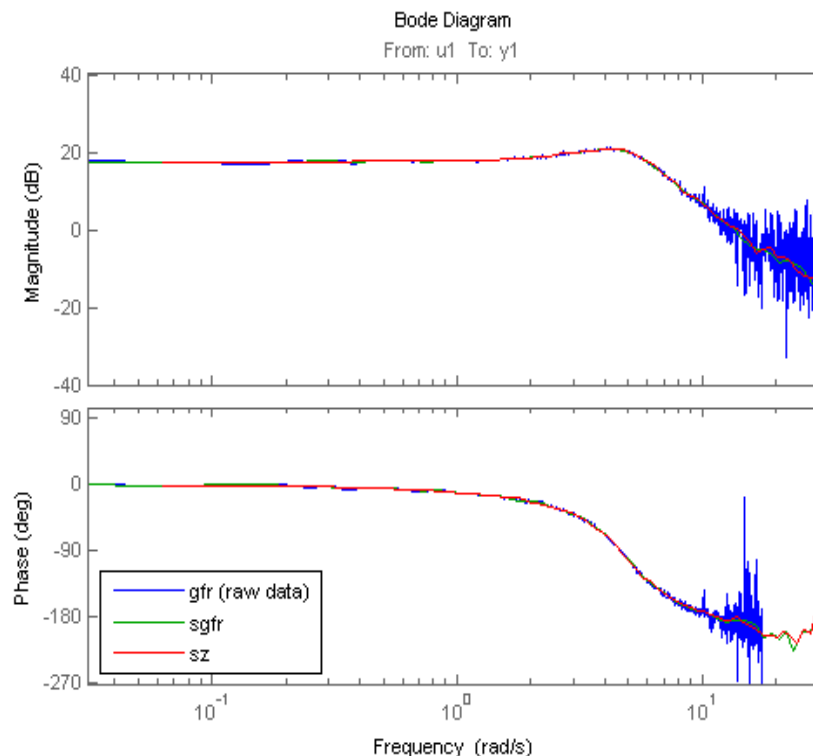


Рис. 5.5. Диаграмма Боде на основе исходных и сжатых данных

На диаграмме видно, что данные в объекте *gfr* были также сглажены. Теперь их можно использовать для оценки, т.е. для построения моделей объекта, например:

```
% Модель «вход-выход» [2 2 1] - аргументы [nb, nf, nk]
msm = oe(sgfr, [2 2 1]);
```

```
% Сравнение моделей
compare(ztime, msm, m1)
```

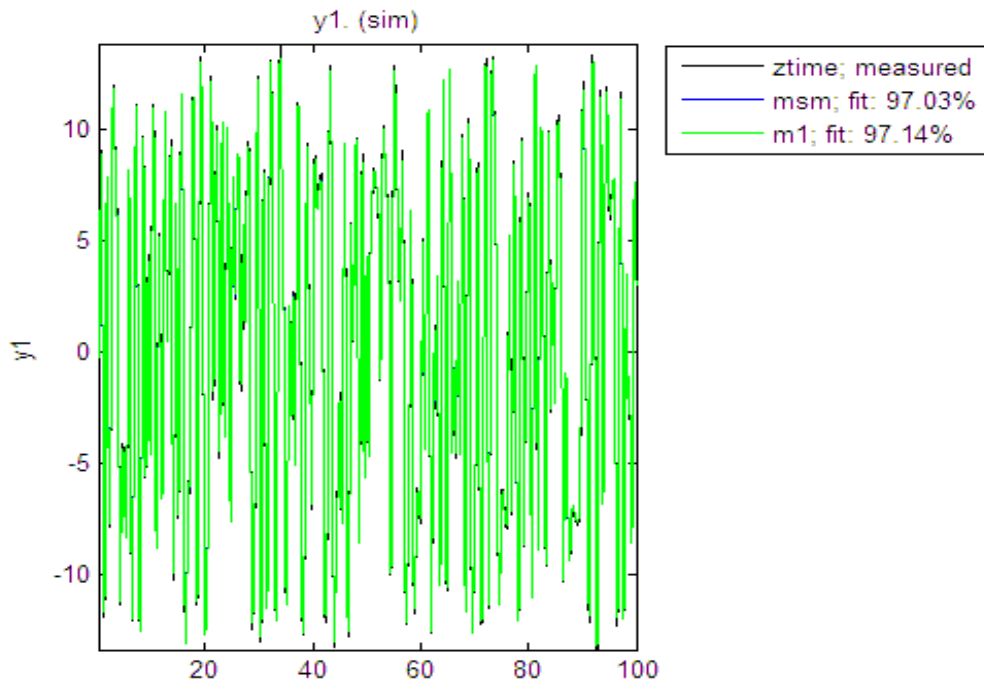


Рис. 5.6. Сравнение модели на основе исходных и сжатых данных

Исходя из результатов, представленных на рис. 5.6, видим, что модель *msm*, построенная на основе сжатых данных *sgfr*, ничем не уступает по точности (см. параметр *fit*) модели *m1*, полученной на основе исходных необработанных данных *gfr*, но объем сжатых данных при этом уменьшен в 10 раз. Данный факт позволяет значительно сэкономить вычислительные ресурсы компьютера практически без потери точности.

5.4.4. Модели на основе частотных данных

Известно, что при наличии сигналов во временной области на входе и выходе системы можно, применив к ним преобразование Фурье, получить представление сигнала в частотной области. Подобные данные *Y* и *U* во временной области, к которым было применено преобразование Фурье, приведены в загруженном ранее *mat*-файле *demofr*. При помощи функции *loglog()* они выглядят (рис. 5.7) следующим образом:

```
% Формируем массив частот, при этом апостроф - символ
% транспонирования - используется для создания вектор-столбца
Wfd = (0:500) '*10*pi/500;

subplot(211),
loglog(Wfd, abs(Y)), title('The amplitude of the output')
```

```
subplot(212),
loglog(Wfd, abs(U)), title('The amplitude of the input')
```

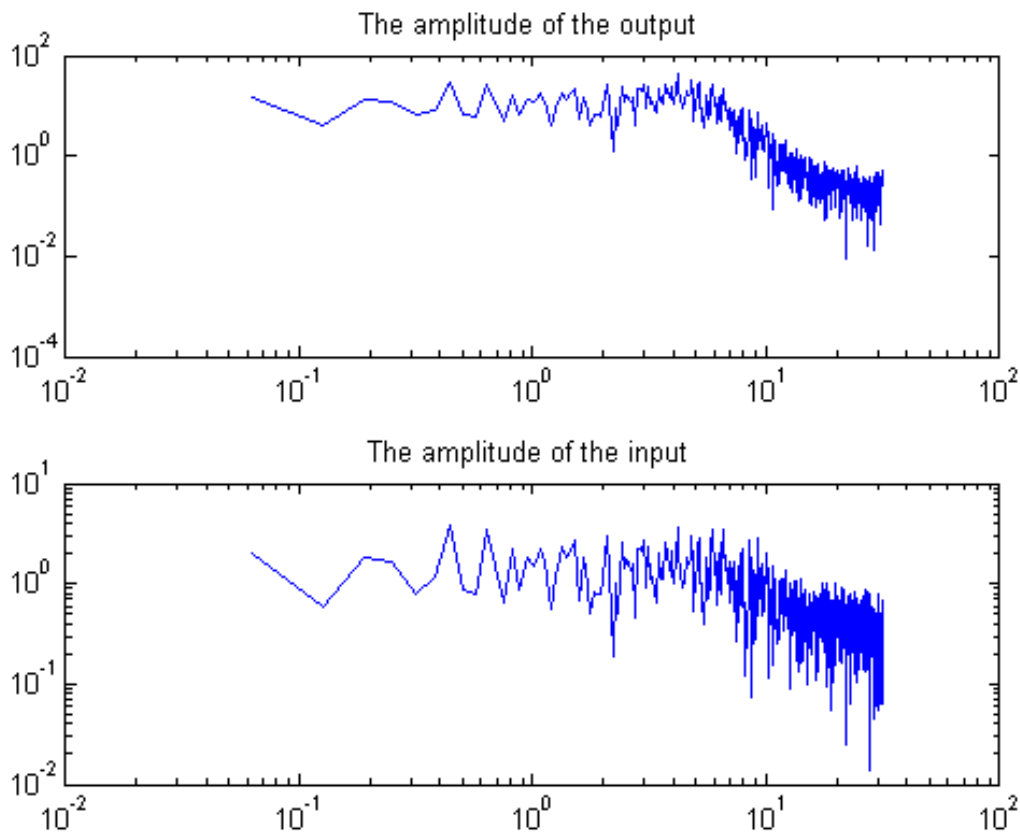


Рис. 5.7. Фурье-преобразование исходных данных из временной области

Для дальнейшей работы с данными в частотной области необходимо сформировать объект типа `IDDATA`, который позволит нам выполнять оценку моделей по исходным данным. Конструктор объекта `IDDATA` в случае его использования с частотными данными выглядит следующим образом:

```
ZFD = iddata(Y,U,'ts',0.1,'Domain','Frequency','Freq',Wfd)
```

```
Frequency domain data set with responses at 501 frequencies,
ranging from 0 to 31.416 rad/seconds
```

```
Sample time: 0.1 seconds
```

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

```
  y1
```

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

```
  u1
```

Полученный набор данных в частотной области `ZFD` позволяет построить модель в пространстве состояний при помощи функции `ret()`, указывая в качестве аргумента, созданный нами объект

IDDATA. Применим далее процедуру оценки адекватности модели *compare()*:

```
mf = pem(ZFD);
compare(ztime, mf, m1);

mf =
Continuous-time 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
x1 -1.501    6.791
x2 -3.115   -2.059
B =
      u1
x1 -28.11
x2 -33.39
C =
      x1      x2
y1 -0.5844   0.4129
D =
      u1
y1  0
K =
      y1
x1  0
x2  0
Parameterization:
  FREE form (all coefficients in A, B, C free).
Feedthrough: none
  Disturbance component: estimate
  Number of free coefficients: 10
  Use «idssdata», «getpvec», «getcov» for parameters and
  their uncertainties.
Status:
Estimated using SSEST on frequency domain data «ZFD».
Fit to estimation data: 97.21% (prediction focus)
FPE: 0.04263, MSE: 0.04195
```

Результаты работы функции сравнения моделей исследуемого объекта на временных данных представлены на рис.5.8.

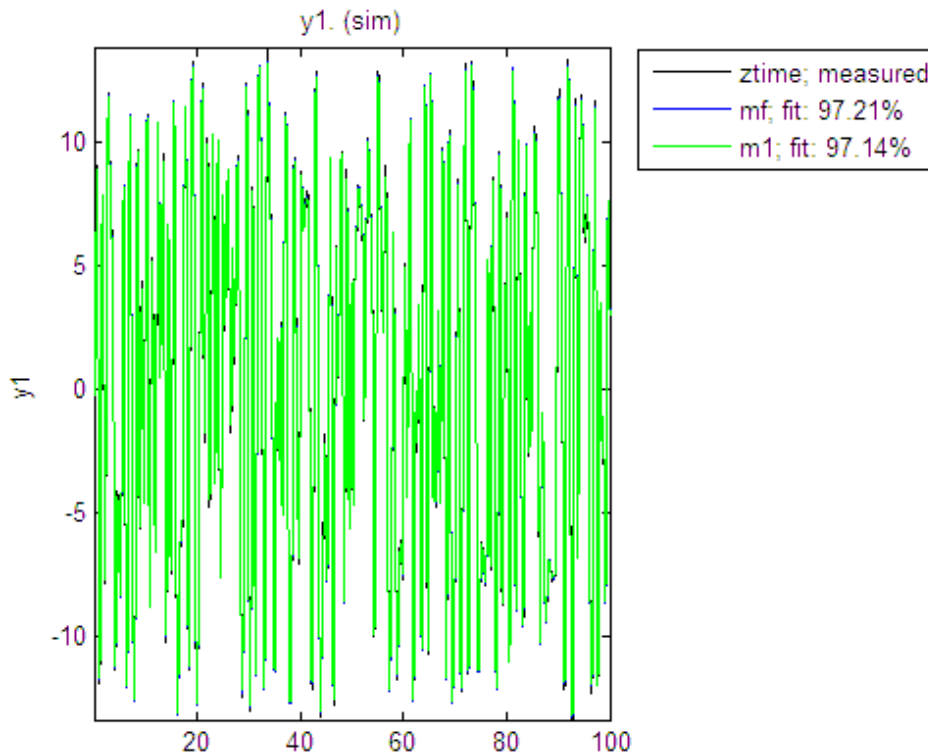


Рис. 5.8. Сравнение модели РЕМ с исходными данными

Исходя из результатов, представленных на рисунке 5.8, можно сказать, что полученная модель в пространстве состояний довольно точно описывает поведение исследуемого объекта (параметр *fit* не ниже 97 %).

5.4.5. Частотно-временное преобразование данных

Как известно, частотные характеристики динамического объекта можно получить, имея временные данные в виде реакции объекта на функцию Дирака (импульсную дельта-функцию) или функцию Хевисайда (ступенчатую функцию), если применить к такой временной реакции прямое преобразование Фурье (FFT). Это позволяет получить спектральные функции входного и выходного сигнала, из которых следуют частотные характеристики объекта. Однако иногда проще получить частотные характеристики объекта экспериментально, подавая на его вход гармоническое воздействие в виде временной синусоиды или косинусоиды, частота которой должна меняться с некоторой достаточно малой дискретной величиной. Набор дискретных амплитуд на входе и выходе объекта, а также набор дискретных фаз входного и выходного сигнала дает намискомые частотные данные. Далее, найдя частное от деления амплитуд на выходе и входе объекта, можно получить амплитудно-частотную характеристику.

Два этих способа эквивалентны, но если в первом способе можно попробовать использовать непрерывную запись сигнала, например на

магнитофон, то в последнем случае, при получении частотных данных надо правильно подобрать дискрету по частоте и выбрать ее достаточно малой, чтобы не пропустить искомым собственным частотам объекта (особенно если объект высокооборотный). Затем уже можно провести сжатие данных с помощью процедуры *spafdr()*.

Итак, временные и частотные данные могут быть преобразованы при использовании преобразования Фурье (FFT и IFFT). Эти функции так же адаптированы к IDDATA объектам. Например, имея временные данные *ztime* и частотные данные *dataf*, можно получить из первых частотные, применив быстрое преобразование Фурье (FFT), а из вторых – временные, применив обратное преобразование Фурье (IFFT):

```
dataf = fft(ztime)
datat = ifft(dataf)
```

```
dataf =
Frequency domain data set with responses at 501 frequencies,
ranging from 0 to 31.416 rad/seconds
Sample time: 0.1 seconds
Outputs      Unit (if specified)
  y1
Inputs      Unit (if specified)
  u1
datat =
Time domain data set with 1000 samples.
Sample time: 0.1 seconds
Outputs      Unit (if specified)
  y1
Inputs      Unit (if specified)
```

Обратите внимание, как соотносятся дискреты по времени и частоте. В исходных данных дискрета по времени *st* (samptime) равна 0,1 с, а максимальная частота выбрана, как обратная этой дискрете величина $1/st = 10$ Гц, что соответствует значению круговой частоты (π) 31,416 рад/с. Следовательно, величина дискреты по частоте равна $31,416/500 = 0,06283$ рад/с, при этом набор как временных, так и частотных данных включает 501 выборку.

Данные во временной и частотной областях могут быть сжаты при помощи функции *SPAFDR*:

```
g1 = spafdr(ztime)
g2 = spafdr(ZFD);
bode(g1, g2)
```

```
g1 =
IDFRD model.
```


Contains Frequency Response Data for 1 output(s) and 1 input(s), and the spectra for disturbances at the outputs. Response data and disturbance spectra are available at 100 frequency points, ranging from 0.06283 rad/s to 31.42 rad/s. Sample time: 0.1 seconds
 Output channels: 'y1'
 Input channels: 'u1'
 Status:
 Estimated using SPAFDR on time domain data «ztime».

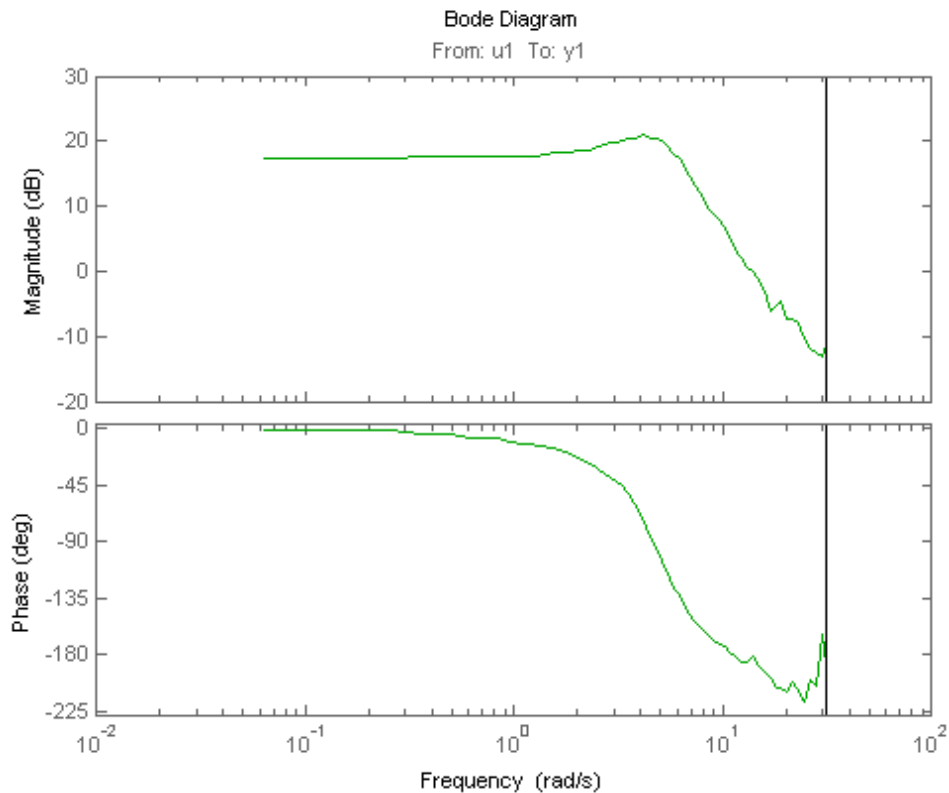


Рис. 5.9. Частотные характеристики объекта исследования

Данные амплитудно-фазо-частотной характеристики *gfr* в виде идентификационного *idgfr* объекта также могут быть сглажены (уменьшено разрешение и количество данных) при помощи той же функции SPAFDR:

```
g3 = spafdr(gfr);
```

Эти сглаженные и сжатые данные частотной характеристики опять могут быть преобразованы в идентификационные данные в частотной области с помощью, например, команд *idfrd()* или *iddata()*, например:

```
gfd = iddata(g3);
```

Frequency domain data set with responses at 100 frequencies, ranging from 0.031416 to 31.416 rad/seconds
 Sample time: 0.1 seconds

Outputs Unit (if specified)
 y1
Inputs Unit (if specified)
u1

Выводы

1) Данные во временных и частотных областях могут использоваться для оценки динамических объектов непрерывными и дискретными моделями вида ОЕ, РЕМ или непрерывно-временной модели PROCEST.

2) Модели могут быть представлены в областях, отличных от тех, в которых они были изначально построены, используя методы преобразования Фурье или сжатия данных: *fft* (БПФ), *ifft* (ОБПФ), *spafdr* и *sra*.

5.5. Задание на лабораторную работу

1) Изучить теоретические сведения.

2) Загрузить и подготовить набор экспериментальных данных «demofr» при помощи команды `load demofr`.

3) Построить модели *m1*, *m2*, *m3*, структуры которых необходимо определить по варианту (см. параметр *m* – для ОЕ модели и *sys* – для procest модели в табл. 4.2).

4) Оценить адекватность построенных моделей, получить сравнительные графики, сделать выводы.

5) Сравнить результаты моделирования с имеющимися в файле «demofr» данными во временной области, сделать выводы.

6) Произвести сжатие исходных данных до *N*-значений частоты. Количество точек *N* выбирается по варианту (см. параметр «*N*» в табл. 4.2). На основе полученных данных построить новую модель структуры ОЕ, получить оценку её адекватности и провести сравнение с предыдущими результатами моделирования на основе несжатых данных. Сделать соответствующие выводы.

7) Оформить отчет.

Исходные данные для выполнения лабораторной работы представлены в таблице 4.2.

Таблица 4.2

Параметр	Вариант				
	1	2	3	4	5
nn	nb = 3, nf = 2, nk = 1	nb = 3, nf = 3, nk = 2	nb = 2, nf = 2, nk = 2	nb = 3, nf = 2, nk = 2	nb = 3, nf = 3, nk = 1
sys	$\frac{K_p e^{-T_d s}}{1 + 2\xi T_\omega s + (T_\omega s)^2}$	$\frac{K_p e^{-T_d s}}{(1 + T_{p1}s)(1 + T_{p2}s)}$	$\frac{K_p(1 + T_2 s)}{1 + 2\xi T_\omega s + (T_\omega s)^2}$	$\frac{K_p e^{-T_d s}}{s(1 + T_{p1}s)(1 + T_{p2}s)}$	$\frac{K_p(1 + T_2 s)e^{-T_d s}}{s(1 + T_{p1}s)(1 + T_{p2}s)}$
N	150	200	250	300	350

Содержание отчета

Отчет оформляется в соответствии с требованиями к оформлению лабораторных работ и должен содержать:

- 1) титульный лист;
- 2) формулировку цели работы;
- 3) исходные данные варианта задания, краткие теоретические сведения;
- 4) результаты работы;
- 5) общие выводы;
- 6) приложения (листинг программы с комментариями).

Контрольные вопросы

1. Назовите основные этапы обработки и анализа частотных данных;
2. Дайте определение Фурье преобразования и обратного Фурье преобразования;
3. В каких случаях более предпочтительно использование временных данных с применением преобразования Фурье, чем данных, полученных в частотной области?

ПРИЛОЖЕНИЯ

Приложение 1

Параметрическая и непараметрическая идентификация на основе метода наименьших квадратов

П1.1. Решение задачи параметрической идентификации на основе метода наименьших квадратов

Решение задачи идентификации в пакете *SystemIdentificationToolbox* на основе рассмотренных в разделе 1.3 моделей рассмотрено в лабораторных работах. Однако такой подход не дает нам полного представления о внутренних процессах и способах решения задачи, поэтому рассмотрим задачу идентификации с использованием одного из наиболее эффективных методов – метода наименьших квадратов.

П1.1.1. Решение во временной области

Пусть необходимо получить дискретную передаточную функцию (или системную функцию), которую можно привести к следующему стандартному виду, соответствующему структуре ARX-модели:

$$G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} = \frac{B(z)}{A(z)}. \quad (\text{П1.1})$$

Отсюда можно получить z-уравнение системы:

$$U(z)(b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}) = Y(z)(1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}). \quad (\text{П1.2})$$

С учетом свойств оператора задержки (см. раздел 1), можно получить также конечно-разностное уравнение:

$$y_k + a_1 y_{k-1} + \dots + a_n y_{k-n} = b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m},$$

где $k = t_k$ – это текущий дискретный момент времени ($k = 1, \dots, N$), из которого можно найти выходную величину системы:

$$y_k = [b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m} - a_1 y_{k-1} - \dots - a_n y_{k-n}]. \quad (\text{П1.3})$$

Таким образом, модель линейной дискретной системы полностью определяется коэффициентами $a_1 \dots a_n$, $b_0 \dots b_m$.

Задача идентификации сводится к нахождению оценок коэффициентов так, чтобы получить результат наилучшим образом согласующийся с экспериментальными данными, т.е. чтобы в каждый момент дискретного времени k получить достаточно малое рассогласование между выходным сигналом модели \hat{y}_k и системы y_k :

$$\xi_k = y_k - \hat{y}_k \rightarrow 0. \quad (\text{П1.4})$$

Предположим, что в ходе экспериментального исследования системы были получены последовательности дискретных отсчетов входного и

выходного сигналов системы (временные данные). Обозначим эти последовательности соответственной $u_1 \dots u_N$ и $y_1 \dots y_N$. Для каждого дискретного отсчета составим уравнение:

$$y_k - [b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m} - a_1 y_{k-1} - \dots - a_n y_{k-n}] = 0. \quad (\text{П1.5})$$

В результате получим систему из N линейных уравнений:

$$\begin{cases} y_1 - [b_0 u_1 + b_1 u_0 + \dots + b_m u_{1-m} - a_1 y_0 - \dots - a_n y_{1-n}] = 0; \\ y_2 - [b_0 u_2 + b_1 u_1 + \dots + b_m u_{2-m} - a_1 y_1 - \dots - a_n y_{2-n}] = 0; \\ \dots \\ y_N - [b_0 u_N + b_1 u_{N-1} + \dots + b_m u_{N-m} - a_1 y_{N-1} - \dots - a_n y_{N-n}] = 0. \end{cases}$$

Решение системы относительно $a_1 \dots a_n$ и $b_0 \dots b_m$ дает искомые коэффициенты. Для описания решения этой задачи в матричном виде объединим теперь все неизвестные коэффициенты в один вектор-столбец θ .

$$\theta = (b_0 \quad b_1 \quad \dots \quad b_m \quad a_1 \quad a_2 \quad \dots \quad a_n)^T.$$

Сформируем также вектор-столбец, содержащий выходные координаты системы Y . Составим матрицу экспериментальных данных Φ по следующему правилу: каждая строка матрицы соответствует моменту дискретного времени и представляет собой вектор

$$\Phi_k = (u_k \quad u_{k-1} \quad \dots \quad u_{k-m} \quad -y_{k-1} \quad -y_{k-2} \quad \dots \quad y_{k-n}), \quad k=1 \dots N.$$

Тогда система уравнений может быть записана в матричном виде

$$\Phi \theta = Y, \quad (\text{П1.6})$$

где

$$Y = (y_1 \quad y_2 \quad \dots \quad y_N)^T,$$

$$\Phi = \begin{pmatrix} u_1 & u_0 & \dots & u_{1-m} & -y_0 & \dots & -y_{1-n} \\ u_2 & u_1 & \dots & u_{2-m} & -y_1 & \dots & -y_{2-n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ u_{N-1} & u_{N-2} & \dots & u_{N-1-m} & -y_{N-2} & \dots & -y_{N-1-n} \\ u_N & u_{N-1} & \dots & u_{N-m} & -y_{N-1} & \dots & -y_{N-n} \end{pmatrix}.$$

В задачах подобного типа Φ называют *матрицей регрессии*, а компоненты этой матрицы называют *регрессорами*. Как матрица Φ , так и столбец свободных членов Y известны из эксперимента и задача формально сводится к решению системы линейных уравнений относительно вектора θ .

$$\theta = \Phi^{-1} Y, \quad (\text{П1.7})$$

где Φ^{-1} – обратная матрица Φ .

Существуют математические проблемы решения этой задачи. Система уравнений (6) является переопределенной (число уравнений больше числа

неизвестных), но, вместе с тем, может быть и несовместной, то есть не иметь точных решений. Кроме того, обращение матрицы коэффициентов системы может быть невозможным, так как она может не быть квадратной или может оказаться вырожденной (определитель равен нулю). Поэтому, в методе наименьших квадратов [12], ищут псевдорешение, под которым понимают такой вектор θ , который делает вектор невязки $\Phi\theta - Y$ минимальным по евклидовой норме. Реализация подобного решения задачи идентификации рассмотрена в разделе 2.

П1.1.2. Решение в частотной области

Существует также возможность использования авторегрессионных моделей в частотной области. Пусть экспериментальные данные представляют собой дискретные отсчеты комплексных частотных спектров входного $U(j\omega_k)$ и выходного $Y(j\omega_k)$ ($k = 1, \dots, N$) сигналов динамического объекта. Сделав замену $z = e^{j\omega T}$, можем получить частотный коэффициент передачи объекта, равный отношению спектров выходного и входного сигналов, который можно представить в следующем виде:

$$G(z)|_{z=e^{j\omega T}} = \frac{b_0 + b_1 e^{-j\omega T} + b_2 e^{-2j\omega T} + \dots + b_m e^{-jm\omega T}}{1 + a_1 e^{-j\omega T} + a_2 e^{-2j\omega T} + \dots + a_n e^{-jn\omega T}} = \frac{Y(j\omega)}{U(j\omega)}. \quad (\text{П1.8})$$

Здесь значения параметров m и n задают структуру модели и должны определяться, исходя из эвристических соображений, предшествующего опыта и должны корректироваться при тестировании. Перепишем (8) в виде уравнения:

$$Y(j\omega) = U(j\omega)(b_0 + b_1 e^{-j\omega T} + b_2 e^{-2j\omega T} + \dots + b_m e^{-jm\omega T}) - \\ - Y(j\omega)(a_1 e^{-j\omega T} + a_2 e^{-2j\omega T} + \dots + a_n e^{-jn\omega T}). \quad (\text{П1.9})$$

Имея набор частот ω_k и соответствующих значений входного U_k и выходного сигналов Y_k , (частотные данные) получим систему из N уравнений, для которой по аналогии с (П1.6) определим матрицу регрессоров и вектор свободных членов. Для краткости введем обозначения:

$$U_k = U(j\omega_k), \quad Y_k = Y(j\omega_k); \quad q_k^{-n} = e^{-jn\omega_k T},$$

тогда получаем:

$$R\theta = Y, \quad (\text{П1.10})$$

где

$$R = \begin{pmatrix} U_1 & U_1 q_1^{-1} & \dots & U_1 q_1^{-m} & -Y_1 q_1^{-1} & \dots & -Y_1 q_1^{-n} \\ U_2 & U_2 q_2^{-1} & \dots & U_2 q_2^{-m} & -Y_2 q_2^{-1} & \dots & -Y_2 q_2^{-n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ U_{N-1} & U_{N-1} q_{N-1}^{-1} & \dots & U_{N-1} q_{N-1}^{-m} & -Y_{N-1} q_{N-1}^{-1} & \dots & -Y_{N-1} q_{N-1}^{-n} \\ U_N & U_N q_N^{-1} & \dots & U_N q_N^{-m} & -Y_N q_N^{-1} & \dots & -Y_N q_N^{-n} \end{pmatrix},$$

$$Y = (Y_1 \ Y_2 \ \dots \ Y_N)^T.$$

Решение полученной системы уравнений полностью аналогично решению системы (П1.6).

П1.2. Непараметрические методы идентификации

Непараметрические методы описания динамики системы используют также дискретные отсчеты экспериментальных данных. При этом не выдвигаются предположения в отношении структуры модели. В ходе планирования и проведения эксперимента сказать заранее, по какой реализации пойдет процесс, невозможно. Поэтому процесс получения массива входных и выходных переменных объекта можно назвать стохастическим и для его обработки применить статистические методы.

Как известно, статистические свойства случайной величины x определяют по ее *функции распределения* (интегральному закону распределения) $F(x)$ или *плотности вероятности* (дифференциальному закону распределения) $W(x)$. Функции $F(x)$ и $W(x)$, математическое ожидание, дисперсия являются основными статистическими характеристиками случайного процесса, но они не дают достаточного представления о том, какой характер будут иметь отдельные реализации случайного процесса. Более подробно ознакомиться с характеристиками случайного процесса можно в приложении 2.

П1.2.1. Решение во временной области

Непосредственно из экспериментальных данных проще всего можно получить переходную характеристику $h(t)$ и импульсную характеристику $w(t)$ [9, 10, 11]. Учитывая, что для линейных динамических объектов свертка входного сигнала $u(t)$ с импульсной характеристикой дает реакцию системы теоретически на любой входной сигнал, то, имея отсчеты

импульсной характеристики, можно определить реакцию объекта на входной сигнал в любой момент времени t_k :

Продолжение приложения 1

$$y_k = \sum_{i=0}^n w(t_k) \cdot u_{k-i}. \quad (\text{П1.11})$$

Оценку импульсной характеристики можно получить, решив уравнение (П1.11) относительно отсчетов $w_k = w(t_k)$, однако, такой способ не даст удовлетворительного решения в случае, если измерения подвержены воздействиям помех. При наличии помех, целесообразно применить корреляционный метод [1,2,5, 8].

Чтобы учесть связь между значениями случайного процесса в различные моменты времени или, иными словами, учесть степень изменчивости случайного процесса, обычно используют *корреляционные функции* и *спектральные плотности* случайного процесса (см. приложение 2).

Автокорреляционной функцией (АКФ) эргодического стационарного случайного процесса $u(t)$ называют неслучайную функцию $R_u(\tau)$, которая для каждой пары произвольно выбранных значений моментов времени t_1 и $t_2 = t_1 - \tau_u$ равна:

$$R_u(\tau) = \lim_{\tau_u \rightarrow \infty} \frac{1}{\tau_u} \int_{-\tau_u/2}^{+\tau_u/2} u(t) \cdot u(t - \tau) dt. \quad (\text{П1.12})$$

Статистические свойства связи двух случайных процессов $u(t)$ и $y(t)$ можно охарактеризовать *взаимной корреляционной функцией* (ВКФ) $R_{yu}(\tau)$:

$$R_{yu}(\tau) = \lim_{\tau_u \rightarrow \infty} \frac{1}{\tau_u} \int_{-\tau_u/2}^{+\tau_u/2} u(t) \cdot y(t - \tau) dt. \quad (\text{П1.13})$$

Если случайные процессы $y(t)$ и $u(t)$ статистически не связаны друг с другом и имеют равные нулю средние значения, то их взаимная корреляционная функция для всех τ равна нулю. Показано, что чем слабее взаимосвязь между предыдущим $u(t)$ и последующим $u(t+\tau)$ значениями случайного процесса, тем быстрее убывает корреляционная функция $R_u(\tau)$ с ростом времени корреляции τ . Случайный процесс, в котором отсутствует связь между предыдущими и последующими значениями, называют *чистым случайным процессом* или *белым шумом*. В случае белого шума время корреляции $\tau = 0$ и корреляционная функция представляет собой δ -функцию.

В случае если $u(t)$ – входной сигнал системы, $y(t)$ – выходной сигнал, а $w(t)$ – её импульсная характеристика, справедливо соотношение [1]:

$$R_{yu}(\tau) = \int_{-\infty}^{+\infty} R_u(t)w(\tau - t)dt. \quad (\text{П1.14})$$

Продолжение приложения 1

В дискретном представлении формула (П1.14) выглядит следующим образом:

$$\hat{R}_{yu}(k) = \sum_{i=0}^n \hat{R}_u(i)w(k-i) \quad \text{или} \quad \hat{R}_{yu}(k) = \sum_{i=0}^n \hat{w}(i)\hat{R}_u(i-k). \quad (\text{П1.15})$$

Решив данную систему уравнений относительно $\hat{w}(i)$, можно получить n -отсчетов оценки импульсной характеристики системы.

Систему уравнений в данном случае удобно представить в матричной форме. Введем матрицу R_u , состоящую из дискретных отсчетов автокорреляционной функции:

$$R_u = \begin{pmatrix} R_u(0) & R_u(1) & \dots & R_u(n-1) & R_u(n) \\ R_u(-1) & R_u(0) & \dots & R_u(n-2) & R_u(n-1) \\ \dots & \dots & \dots & \dots & \dots \\ R_u(1-n) & R_u(2-n) & \dots & R_u(0) & R_u(1) \\ R_u(-n) & R_u(1-n) & \dots & R_u(-1) & R_u(0) \end{pmatrix}. \quad (\text{П1.16})$$

С учетом того, что исследуемые процессы вещественные, корреляционная функция является четной и матрицу (П1.16) можно переписать в виде:

$$R_u = \begin{pmatrix} R_u(0) & R_u(1) & \dots & R_u(n-1) & R_u(n) \\ R_u(1) & R_u(0) & \dots & R_u(n-2) & R_u(n-1) \\ \dots & \dots & \dots & \dots & \dots \\ R_u(n-1) & R_u(n-2) & \dots & R_u(0) & R_u(1) \\ R_u(n) & R_u(n-1) & \dots & R_u(-1) & R_u(0) \end{pmatrix}. \quad (\text{П1.17})$$

Если импульсную характеристику w и взаимную корреляционную функцию представить как вектор-столбцы:

$$w = \begin{pmatrix} w(0) \\ w(1) \\ \dots \\ w(n-1) \\ w(n) \end{pmatrix} \quad \text{и} \quad R_{yu} = \begin{pmatrix} R_{yu}(0) \\ R_{yu}(1) \\ \dots \\ R_{yu}(n-1) \\ R_{yu}(n) \end{pmatrix}, \quad (\text{П1.18})$$

то система уравнений в матричной форме будет записана так:

$$R_u w = R_{yu}. \quad (\text{П1.19})$$

С учетом (П1.19) решение системы можно определить по формуле:

$$w = R_u^{-1} R_{yu}. \quad (\text{П1.20})$$

Продолжение приложения 1

Оценки корреляционных функций, которые являются коэффициентами в системе уравнений, можно получить по формулам:

$$\hat{R}_u(k) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) u(n-k) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) \cdot u(n+k), \quad (\text{П1.21})$$

$$\hat{R}_{yu}(k) = \frac{1}{N} \sum_{n=0}^{N-1} y(n) u(n-k) = \frac{1}{N} \sum_{n=0}^{N-1} y(n+k) \cdot u(n).$$

Это и дает в итоге возможность получить искомую оценку импульсной характеристики.

П1.2.2. Решение в частотной области

Для оценки частотных характеристик исследуемого линейного объекта на основе непараметрических методов наиболее просто использовать тот факт, что комплексный коэффициент передачи системы является отношением изображений Фурье выходного и входного сигналов системы, полученных при нулевых начальных условиях. Далее, считая, что контролируемые сигналы представлены последовательностями своих дискретных отсчетов, можно получить выражение для оценки комплексного коэффициента передачи системы:

$$\hat{K}(j\omega) = \frac{\hat{Y}(j\omega)}{\hat{U}(j\omega)} \quad (\text{П1.22})$$

Здесь $\hat{U}(j\omega)$ и $\hat{Y}(j\omega)$ – оценки изображений Фурье входных и выходных сигналов, полученные на основе известных формул дискретного преобразования Фурье [10, 11]:

$$\hat{Y}(j\omega_k) = \hat{Y}(k) = \sum_{n=0}^{N-1} y(n) e^{-j2\pi n k / N}, \quad \hat{U}(j\omega_k) = \hat{U}(k) = \sum_{n=0}^{N-1} u(n) e^{-j2\pi n k / N} \quad (\text{П1.23})$$

где k – номера отсчетов дискретного сигнала; n – номера отсчетов спектральной функции, следующие через одинаковые промежутки $f_s/2N$ (f_s – частота дискретизации), N – число отсчетов за рассматриваемый период.

Частотный спектр, вычисленный на основе (П1.22), может получиться неравномерным, поэтому для сглаживания спектра оценку (П1.22) можно заменить следующим выражением [1, 11]:

$$\widehat{K}(j\omega) = \frac{\int_{-\pi}^{\pi} W(\Omega - \omega) |U|^2 \widehat{K}(j\Omega) d\Omega}{\int_{-\pi}^{\pi} W(\Omega - \omega) |U|^2 d\Omega}. \quad (\text{П1.23})$$

Здесь $W(\omega)$ является весовой функцией, причем в качестве весовых функций обычно применяются частотные оконные функции [1, 11] типа косинусных окон Хэмминга, Блэкмана-Хэрриса и др.

Окончание приложения 1

Аналогичные рассуждения можно провести с использованием спектральных плотностей на основе корреляционной теории.

Приложение 2

Характеристики и свойства случайного процесса

Приводимые ниже теоретические сведения общеизвестны, являются справочными и служат дополнением предшествующему курсу «Теория вероятностей и математическая статистика» направления 200100 «Приборостроение» магистерской программы «Измерительные информационные технологии».

Случайным процессом называется функция двух переменных $\xi = \xi(\omega, t)$, где ω – элементарный исход эксперимента, t – время. Важнейшими характеристиками случайного процесса (СП) являются его моменты, а также стационарность, эргодичность и спектр мощности.

П2.1. Моменты случайных величин

Одномерная плотность распределения вероятности $W(x, t)$ определяет вероятность

$$W(x, t) dx = P(x \leq \xi(t) \leq x + dx) \quad (\text{П2.1})$$

того, что случайная величина $x(t)$ лежит в интервале $x \leq \xi(t) \leq x + dx$ и с помощью функции $W(x, t)$ можно провести усреднение как случайной величины $x(t)$, так и любой функции от нее.

Средним значением СП $\bar{\xi}(t)$ или его первым моментом (математическое ожидание или матожидание) называется интеграл [8]:

$$M(t) = \bar{\xi}(t) = \int_{-\infty}^{+\infty} x W(x, t) dx \quad (\text{П2.2})$$

Второй начальный момент $\bar{\xi}^2(t)$ СП описывается интегралом:

$$D(t) = \overline{\xi^2}(t) = \int_{-\infty}^{+\infty} x^2 W(x, t) dx \quad (\text{П2.3})$$

и определяет среднюю мощность случайного процесса.

При анализе случайных процессов часто интерес представляет флуктуационная составляющая. Введем понятие центрированного СП $\xi^0 = \xi - \overline{\xi}$. Второй *центральный момент* определяет мощность флуктуационной составляющей случайного процесса и называется *дисперсией*:

Продолжение приложения 2

$$\sigma^2(t) = \int_{-\infty}^{+\infty} (x - \overline{\xi}(t))^2 W(x, t) dx \quad (\text{П2.4})$$

Аналогично можно определить также моменты случайного процесса более высокого порядка. Статистическими характеристиками СП, обычно достаточными для их описания, являются математическое ожидание, дисперсия, однако для решения задач идентификации используются корреляционные функции и спектральная плотность сигналов.

Используя одномерную плотность распределения вероятности можно получить параметры СП, которые являются усреднением по множеству (ансамблю) реализаций данного случайного процесса в каком либо его «сечении», то есть в фиксированный момент времени t . Но задание одномерной плотности распределения вероятности не дает возможность определить характер изменения случайного процесса во времени и не характеризует взаимосвязь случайного процесса в различные моменты времени. Для этого вводят понятие двумерной плотности распределения вероятности $W(x_1, x_2, t_1, t_2)$, описывающей связь двух значений $\xi(t_1)$ и $\xi(t_2)$ в произвольные моменты времени t_1 и t_2 :

$$W(x_1, x_2, t_1, t_2) dx_1 dx_2 = P(x_1 \leq \zeta(t_1) \leq x_1 + dx_1 \cap x_2 \leq \zeta(t_2) \leq x_2 + dx_2) \quad (\text{П2.5})$$

С помощью двумерной плотности распределения вероятности можно определить *автокорреляционную функцию* [8]:

$$\begin{aligned} B_x(t_1, t_2) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(t_1) x(t_2) W(x_1, x_2, t_1, t_2) dx_1 dx_2 = \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 W(x_1, x_2, t_1, t_2) dx_1 dx_2 = M[x_1 x_2], \end{aligned} \quad (\text{П2.6})$$

а также *автокорреляционную функцию* центрированного случайного процесса:

$$\begin{aligned} R_x(t_1, t_2) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x_1 - \overline{\xi}(t_1))(x_2 - \overline{\xi}(t_2)) W(x_1, x_2, t_1, t_2) dx_1 dx_2 = \\ &= M[(x_1 - M[x_1])(x_2 - M[x_2])] \end{aligned} \quad (\text{П2.7})$$

или:

$$R(t_1, t_2) = B(t_1, t_2) - \bar{\xi}(t_1)\bar{\xi}(t_2). \quad (\text{П2.8})$$

Если рассматриваются два СП x и y , то по аналогии с выражением (6) в рассмотрение вводят взаимную корреляционную функцию [8]:

$$B_{xy}(t_1, t_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(t_1)y(t_2)W(x_1, y_2, t_1, t_2)dx_1dy_2 = M[xy] \quad (\text{П2.9})$$

и взаимную корреляционную функцию (корреляционный момент, ковариация) центрированного случайного процесса:

Продолжение приложения 2

$$\begin{aligned} R_{xy}(t_1, t_2) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x(t_1) - \bar{\xi}(t_1))(y(t_2) - \bar{\eta}(t_2))W(x_1, y_2, t_1, t_2)dx_1dy_2 = \\ &= M[(x_1 - M[x_1])(y_2 - M[y_2])]. \end{aligned}$$

Корреляционный момент двух независимых случайных величин x и y равен нулю. Из определения корреляционного момента следует, что он имеет размерность, равную произведению размерностей величин x и y , то есть, величина корреляционного момента зависит от единиц измерения случайных величин.

Такая особенность корреляционного момента является недостатком этой числовой характеристики, поскольку сравнение корреляционных моментов различных систем случайных величин становится затруднительным. Для того чтобы устранить этот недостаток, вводят новую числовую характеристику — коэффициент корреляции.

Коэффициентом корреляции r_{xy} случайных величин x и y называют отношение корреляционного момента к произведению средних квадратических флуктуаций (отклонений) этих величин:

$$r_{xy} = R_{xy} / (\sigma_x \sigma_y) \quad (\text{П2.10})$$

Очевидно, коэффициент корреляции независимых случайных величин равен нулю.

П2.2. Стационарность случайного процесса

Случайный процесс называется стационарным [8], если одномерная плотность распределения вероятности и, следовательно, среднее значение и дисперсия СП не зависят от времени, а двумерная плотность распределения вероятности и автокорреляционная функция зависят только от разности временных аргументов:

$$B(t_1, t_2) = B(\tau) \quad (\text{П2.11})$$

где $\tau = t_1 - t_2$.

Если многомерный закон распределения вероятностей распределения мгновенных значений, взятых в разные моменты времени, не зависят от

принятого начала отсчёта, а зависят только от интервалов между выбранными моментами:

$$W_k(x_1, t_1, \dots, x_k, t_k) = W_k(x_1, t_1 + \Delta, x_2, t_2 + \Delta, \dots, x_k, t_k + \Delta), \quad (\text{П2.12})$$

то такой процесс стационарен в узком смысле. Если независимость от начала отсчёта выполняется только до второго корреляционного момента, включая первый, то такой процесс называют стационарным в широком смысле:

$$R_k(t_1, t_2) = R_k(t_2 - t_1) = R_k(\tau). \quad (\text{П2.13})$$

Продолжение приложения 2

П2.2.Эргодичность случайного процесса

Случайный процесс называется эргодическим первого порядка, если его первый момент, полученный усреднением по множеству реализаций, с вероятностью сколь угодно близкой к единице совпадает со средним значением, полученным усреднением по времени одной достаточно длинной реализации [8].

Случайный процесс называется эргодическим второго порядка, если его корреляционная функция (П2.8), с вероятностью сколь угодно близкой к единице совпадает со средним значением произведения СП при сдвинутых аргументах, полученным усреднением по времени одной достаточно длинной реализации. Параметры эргодического СП могут быть определены так:

$$R_k(t_1, t_2) = R_k(t_2 - t_1) = R_k(\tau). \quad (\text{П2.14})$$

Усреднением по времени могут быть найдены также автокорреляционные функции эргодического СП:

$$B(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} \xi(t) \xi(t + \tau) dt, \quad R(t) = B(\tau) - \bar{\xi}^2. \quad (\text{П2.15})$$

В данной работе предполагается, что аддитивно добавленный к полезному сигналу «шум» является эргодическим СП. Кроме того, поскольку плотности распределения вероятностей априорно неизвестны, на практике принимается гипотеза, что исследуемые сигналы обладают свойством эргодичности по отношению к корреляционным функциям.

П2.3.Спектральная плотность случайных процессов

Будем полагать, что случайные процессы x и y стационарны и для них существует взаимно-корреляционная функция $R_{xy}(\tau)$. Для этой корреляционной функции введем в соответствие следующее преобразование[8]:

$$S_{xy}(j\omega) = \int_{-\infty}^{+\infty} R_{xy}(\tau) e^{-j\omega\tau} d\tau, \quad (\text{П2.16})$$

которое и определяет взаимную спектральную плотность S_{xy} случайных сигналов x и y . Очевидно, что S_{xy} представляет собой обратное преобразование Фурье от корреляционной функции. Справедливо и обратное. Зная спектральную плотность сигналов можно найти их корреляционную функцию, используя прямое преобразование Фурье:

$$R_{xy}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{xy}(j\omega) e^{j\omega\tau} d\omega. \quad (\text{П2.17})$$

Окончание приложения 2

Введем, аналогично, спектральную плотность автокорреляционной функции случайного процесса $S(\omega)$, связанную с автокорреляционной функцией $R(\tau)$ преобразованием Фурье:

$$S(\omega) = \int_{-\infty}^{+\infty} R(\tau) e^{-j\omega\tau} d\tau, \quad R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega) e^{j\omega\tau} d\omega. \quad (\text{П2.18})$$

Спектральная плотность автокорреляционной функции случайного процесса $S(\omega)$ описывает мощность случайного процесса, приходящуюся на заданный интервал частот, поэтому эту функцию называют спектральной плотностью мощности случайного процесса.

Таким образом, в отличие от спектрального описания детерминированных функций, когда с помощью преобразования Фурье по заданному частотному спектру можно определить временную форму сигнала, спектральное описание случайного процесса проводится для его статистической характеристики – автокорреляционной функции. С помощью выражения (П2.18) можно выразить среднюю мощность случайного процесса через $S(\omega)$:

$$\overline{\xi^2} = R(0) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega) d\omega. \quad (\text{П2.19})$$

Аналогично можно определить мощность флуктуационной составляющей – дисперсию случайного процесса.

При идентификации часто пользуются понятием «белого» шума. Белым шумом называется случайный процесс, у которого спектральная плотность автокорреляционной функции не зависит от частоты $S(\omega) = S_0$. Это соответствует тому, что мощность белого шума равномерно распределена по всему частотному диапазону. Понятие «белый шум» определяет только спектральную характеристику случайного процесса и оставляет открытым вопрос о законах распределения, т.е. белыми шумами могут быть любые случайные процессы, имеющие равномерный энергетический спектр и различные законы распределения.

Приложение 3

Временной ряд и некоторые характеристики его составляющих

Большая часть технических и естественных процессов часто описывается набором последовательных значений показателя $x_1, x_2, \dots, x_i, \dots, x_N$, зафиксированных в моменты времени $t = t_1, t_2, \dots, t_N$, причем длина интервала $\Delta t = t_{i+1} - t_i$ не обязательно величина постоянная. Такой набор значений $x_i, i = 1, 2, \dots$ обычно называется временным рядом

Продолжение приложения 3

(временной серией). Такой ряд представляет собой дискретный временной процесс и включает в себя два обязательных элемента – отметку времени и значение показателя ряда, полученное тем или иным способом и соответствующее указанной отметке времени. Каждый временной ряд рассматривается как выборочная реализация из бесконечной популяции, генерируемой стохастическим процессом, на который оказывают влияние множество факторов [8].

Временные ряды различаются способом определения значения, временным шагом, памятью и стационарностью(см. [17]).

Интервальный временной ряд представляет собой последовательность, в которой уровень явления (значение временного ряда) относят к результату, накопленному или вновь произведенному за определенный интервал времени. Интервальным, например, является временной ряд показателя выпуска электроэнергии, произведенной за час, день, месяц и другие.

Если же значение временного ряда характеризует изучаемое явление в конкретный момент времени, то совокупность таких значений образует *моментный* временной ряд. Примерами моментных рядов являются физические показатели, такие как температура окружающего воздуха, влажность, давление, измеренные в конкретные моменты времени.

В зависимости от частоты определения значений временного ряда, они делятся на *равноотстоящие* временные ряды и *неравноотстоящие* временные ряды.

Равноотстоящие временные ряды формируются при исследовании и фиксации значений процесса в следующие друг за другом *равные* интервалы времени. *Неравноотстоящими* временными рядами называются те ряды, для которых принцип равенства интервалов фиксации значений не выполняется. К таким рядам относятся, например, все биржевые индексы в связи с тем, что их значения определяются лишь в рабочие дни недели.

В зависимости от характера описываемого процесса временные ряды разделяются на временные ряды *длинной памяти*, временные ряды *короткой памяти*. Под временными рядами с длинной памятью

подразумеваются временные ряды, для которых автокорреляционная функция [8] убывает медленно. К временным рядам с короткой памятью относят временные ряды, автокорреляционная функция которых убывает быстро. Многие физические процессы, такие как потребление электроэнергии, температура воздуха, относятся к временным рядам с длинной памятью, а к временным рядам с короткой памятью относятся, например, временные ряды биржевых индексов.

Продолжение приложения 3

Временные ряды принято разделять на стационарные и нестационарные временные ряды. Стационарным временным рядом называется такой ряд, который остается в равновесии относительно постоянного среднего уровня. Остальные временные ряды являются нестационарными. В работе [8] указано, что и в промышленности, и в торговле, и в экономике, где прогнозирование имеет важное значение, многие временные ряды являются нестационарными, то есть не имеющими естественного среднего значения. Нестационарные временные ряды для решения задачи прогнозирования часто приводятся к стационарным при помощи численного дифференцирования, используя конечно-разностное преобразование.

Методы и модели временных рядов строятся на основе анализа характера изменения значений x_i , которые обычно происходят под воздействием различных причин, в силу их многочисленности и сложности измерения, обосновать и построить «подходящую» для описания процесса x_i , $i = 1, 2, \dots$ многофакторную модель не всегда представляется возможным. В результате в отношении ряда x_i часто выдвигается предположение, что совокупное влияние этих факторов формирует внутренние закономерности в развитии процесса x_i , что дает возможность применить для его описания модель из специфического класса моделей временных рядов.

Модели временных рядов строят не просто для описания поведения динамического объекта, но и, в основном, для *прогноза* его поведения. По истечении времени всегда можно сравнить прогнозное и реальное поведение объекта, что позволяет определить точность решения задачи идентификации. Различают метод и модель прогнозирования временного ряда.

Модель прогнозирования есть функциональное представление, адекватно описывающее временной ряд и являющееся основой для получения будущих значений процесса. Итеративный подход к построению модели прогнозирования состоит из следующих шагов [3,8].

1. На основании предыдущего собственного или стороннего опыта выбирается общий класс моделей для прогнозирования временного ряда на какой-то момент времени (горизонт).

2. Так как найденный класс моделей обычно обширен, то для непосредственной подгонки к исходному временному ряду, развиваются грубые методы идентификации подклассов моделей. Такие методы идентификации используют качественные оценки временного ряда.

3. После определения подкласса модели, необходимо оценить ее параметры и структуру, если модель относится к категории структурных

Продолжение приложения 3

моделей. На данном этапе обычно используются итеративные способы, когда производится оценка участка (или всего) временного ряда при различных значениях изменяемых величин. Как правило, данный шаг является наиболее трудоемким в связи с тем, что часто в расчет принимаются все доступные значения временного ряда.

4. Далее производится диагностическая проверка полученной модели прогнозирования. Чаще всего выбирается участок или несколько участков временного ряда, достаточных по длине для проверочного прогнозирования и последующей оценки точности прогноза. Выбранные для диагностики модели прогнозирования участки временного ряда называются контрольными участками (периодами).

5. В случае если точность диагностического прогнозирования оказалась приемлемой для задач, в которых используются прогнозные значения, то модель готова к использованию, а если недостаточной – то возможно итеративное повторение всех описанных выше шагов, начиная с первого.

Метод прогнозирования представляет собой последовательность действий, которые нужно совершить для получения модели временного ряда. Методы прогнозирования делятся на две группы: интуитивные и формализованные.

Интуитивное прогнозирование применяется тогда, когда объект прогнозирования либо слишком прост, либо, напротив, настолько сложен, что аналитически учесть влияние внешних факторов невозможно. Интуитивные методы прогнозирования не предполагают разработку моделей прогнозирования и отражают индивидуальные суждения специалистов (экспертов) относительно перспектив развития процесса. Такие методы используются для анализа процессов, развитие которых не поддается математической формализации, то есть для которых трудно разработать адекватную модель.

Формализованные методы уже рассматривают модели прогнозирования, которые подразделяются на статистические и структурные модели. В статистических моделях функциональная зависимость между будущими и фактическими

значениями временного ряда, а также внешними факторами задана аналитически. К статистическим моделям относятся регрессионные и авторегрессионные модели, модели экспоненциального сглаживания. В структурных моделях функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана структурно. К структурным моделям относят нейросетевые модели, модели на базе цепей Маркова, модели на базе классификационно-регрессионных деревьев и некоторые другие.

Продолжение приложения 3

П3.1. Статистические модели

Рассмотрим более подробно статистические модели временных рядов. Здесь предполагается, что в развитии процесса Y можно выделить следующие составляющие части [3, 8]:

$$Y(t) = f(t) + s(t) + u(t) + e(t) \quad (\text{П3.1})$$

где $f(t)$ – функция тренда (тенденция развития), $s(t)$ – сезонная компонента, $u(t)$ – циклическая компонента, $e(t)$ – остаточная компонента. Опишем составляющие временного ряда подробнее.

Трендом называют долговременную тенденцию развития некоторого явления. При этом она выражается некоторой монотонной функцией. Например, изменение напряжения на выходе генератора в процессе работы от роста температуры в корпусе прибора.

Сезонная компонента характеризует воздействие факторов, возникающих с определённой периодичностью. Особенностью является то, что их действие заканчивается в течение этого периода. Например, загруженность автотрассы в течение суток, девиация амплитуды напряжения на выходе генератора сигналов в результате действия периодических низкочастотных помех, увеличение и уменьшение шумов на выходе принимающего устройства из-за суточного цикла температуры.

Циклическая компонента – это функция, описывающая явление, действующее с длительным периодом. Особенностью является то, что для выявления циклической компоненты обычно недостаточно только наблюдаемых данных, а требуется анализ общей ситуации, включая, может быть, экономическую, социальную и даже историческую. Например, демографические ямы, частота отключения электричества на подстанциях, количество аварий на транспорте и др.

Остаточная компонента выражает воздействие случайных факторов и при изучении этой компоненты требуется использование статистического и вероятностного анализа

Задача анализа временных рядов состоит в том, чтобы с помощью детерминированной компоненты предсказывать прогнозное значение

временного ряда, а с помощью случайной компоненты предсказывать величину возможного отклонения и вероятность такого отклонения.

Для того чтобы анализ временного ряда обладал в нужной степени достоверностью, в первую очередь необходимо обеспечить качество исходной информации[8]:

- данные должны быть сопоставимы;
- данные должны быть однородными;
- данные должны быть устойчивыми;
- необходим достаточно большой объём данных.

Продолжение приложения 3

Прогнозирование процессов, которые могут быть представлены одномерными временными рядами, сводится к выполнению следующих этапов:

- предварительный анализ данных;
- построение модели, т.е. выбор кривых, описывающих явление, и численное оценивание параметров модели;
- проверка адекватности моделей и оценка их точности;
- выбор лучшей модели;
- расчёт точечного и интервального прогноза.

Опишем некоторые из этих этапов. При предварительном анализе данных происходит выявление аномальных отклонений, проверка наличия тренда, сглаживание временных рядов и расчёт показателей развития динамики процессов.

Выявление аномальных отклонений производится обычно с помощью критерия Ирвина[8]. Пусть x_1, x_2, \dots, x_n – временной ряд. Значение критерия вычисляется по формуле:

$$\lambda_i = \frac{|x_i - x_{i-1}|}{S}, \text{ где } S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (\text{П3.2})$$

где S – несмещённое среднеквадратическое отклонение данного ряда, \bar{x} – выборочное среднее ряда или средний уровень ряда. Далее проверяется гипотеза H_0 – «аномальные данные отсутствуют». По таблицам для критерия Ирвина находится $\lambda_{\text{критическое}}$, если $\lambda_i < \lambda_{\text{критическое}}$, то x_i считается нормальным; если $\lambda_i > \lambda_{\text{критическое}}$, то x_i считается аномальным. В этом случае аномальное значение исключается из ряда данных и вместо него подставляют обычно среднеарифметическое из двух соседних значений.

Выявление тренда чаще происходит визуальным образом и зависит от опытности исследователей. Основная цель – угадать функцию, по которой развивается процесс. Здесь требуется информация о самом явлении, в

зависимости от которой можно использовать или регрессионные модели, или авторегрессионные, или модели экспоненциального сглаживания.

При построении любой из этих моделей используется понятие кривых роста. Обычно используют кривые роста, стараясь выбрать кривую максимально простого вида, например:

$$Y(t) = a_0 + a_1 t \text{ – линейный тренд;}$$

$$Y(t) = a_0 + a_1 t + a_2 t^2 \text{ – квадратичный тренд;}$$

$$Y(t) = a_0 e^{-a_1 t} \text{ – экспоненциальный тренд.}$$

Продолжение приложения 3

Подборка коэффициентов и выбор моделей является типичной задачей регрессионного анализа.

Математическая постановка задачи регрессионного анализа заключается в следующем [3, 8]. Зависимость величины определенного свойства случайного процесса или физического явления Y от другого переменного свойства или параметра X , которое в общем случае также может относиться к случайной величине, зарегистрирована на множестве точек x_k множеством значений y_k , при этом в каждой точке зарегистрированные значения y_k и x_k отображают действительные значения $Y(x_k)$ со случайной погрешностью e_k , распределенной, как правило, по нормальному закону. По совокупности значений y_k требуется подобрать такую функцию $f(x_k, a_0, a_1, \dots, a_n)$, которой зависимость $Y(x)$ отображалась бы с минимальной погрешностью.

Отсюда следует условие приближения:

$$y_k = f(x_k, a_0, a_1, \dots, a_n) + e_k. \quad (\text{П3.3})$$

Функцию $f(x_k, a_0, a_1, \dots, a_n)$ называют регрессией величины y на величину x . Регрессионный анализ предусматривает задание вида функции $f(x_k, a_0, a_1, \dots, a_n)$ и определение численных значений ее параметров a_0, a_1, \dots, a_n обеспечивающих наименьшую погрешность приближения к множеству значений y_k . Как правило, при регрессионном анализе погрешность приближения вычисляется методом наименьших квадратов. Для этого выполняется минимизация функции квадратов остаточных ошибок:

$$\Phi(a_0, a_1, \dots, a_n) = [f(x_k, a_0, a_1, \dots, a_n) - y_k]^2 \quad (\text{П3.4})$$

Для определения параметров a_0, a_1, \dots, a_n функция остаточных ошибок дифференцируется по всем параметрам, полученные уравнения частных производных приравниваются нулю и решаются в совокупности относительно всех значений параметров.

Например, для линейной регрессии, когда $f = a_0 + a_1 x$, необходимо минимизировать следующее выражение:

$$\Phi(a_0, a_1) = \sum_i (a_0 + a_1 x_i - y_i)^2. \quad (\text{ПЗ.5})$$

Если приравнять $\frac{\partial \Phi}{\partial a_0}$ и $\frac{\partial \Phi}{\partial a_1}$ к нулю, то для линейной регрессии можно найти ее параметры a_0 и a_1 в явном виде:

$$a_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}, \quad a_0 = \frac{1}{n} \left(\sum_{i=1}^n y_i - a_1 \sum_{i=1}^n x_i \right) \quad (\text{ПЗ.6})$$

Продолжение приложения 3

При оценке качества построенной модели должны быть вычислены следующие величины: \hat{Y}_i – значение, вычисленное по модели; y_i – наблюдение; $e_i = y_i - \hat{Y}_i$ – остаток в определении y_i .

Модель считается хорошей, если ряд остатков e_i является случайным и подчиняется нормальному закону распределений [8]. Для доказательства используются:

- проверка равенства нулю математического ожидания;
- проверка условия случайности возникновения отдельных отклонений от тренда;
- проверка наличия или отсутствия автокорреляции в отклонении модели;
- соответствие ряда остатков нормальному закону распределения.

Модель является адекватной при условии положительного выполнения всех вышеперечисленных проверок.

ПЗ.2. Модели авторегрессии в корреляционной теории

В случае, когда регрессионная модель не адекватна наблюдаемой последовательности значений на выходе объекта, можно использовать модель авторегрессии. Такая необходимость может возникнуть, когда выходные данные представляют собой, например, затухающие или нарастающие колебания, что соответствует обычно выходу объекта, описываемого дифференциальными уравнениями. Учитывая, что данные поступают в дискретном (табличном) виде, дифференциальные уравнения должны быть также дискретизированы с помощью замены производных в уравнениях левыми разностями, т.е. разностями на предыдущих отсчетах, определяющих «память» исследуемого динамического объекта, а значит и их взаимосвязь.

Такую взаимосвязь в статистике называют *корреляцией* между данными, а поскольку коррелируют друг с другом значения одного и того же ряда, то такое явление называют *автокорреляцией* [8].

Для того чтобы определить насколько процесс является автокоррелированным осуществляют расчет коэффициентов парной корреляции r_τ между значениями этого ряда $x[t]$ и ими же, сдвинутыми на некоторый шаг назад τ . Такие коэффициенты называются *автокорреляционными* (см. формулу 10 приложения 2).

Продолжение приложения 3

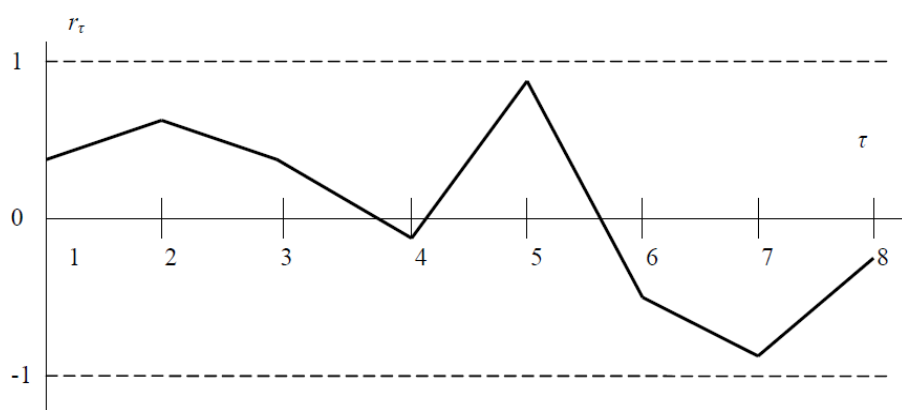


Рис. ПЗ.1. Типовой график коэффициента автокорреляции (коррелограмма)

На график коррелограммы наносят не только значения коэффициентов автокорреляции при соответствующих сдвигах τ , но ещё и соединяют близлежащие точки отрезками прямых линий. В результате, получается некоторая ломаная линия, максимумы и минимумы которой и являются предметом особого изучения, ведь они характеризуют приближение зависимости между значениями ряда $x[t]$ и $x[t-\tau]$ к линейной. Очевидно, что чем ближе величина коэффициента автокорреляции по модулю окажется не менее чем 0,8, то говорят о наличии этой зависимости, а сдвиг во времени τ , соответствующий этому высокому значению коэффициента, называют *лагом*. Если автокорреляционная функция имеет несколько лагов, то говорят о том, что у этого ряда имеются *распределённые лаги*. Поскольку лаг означает наличие зависимости значений самого ряда от его же значений, но сдвинутых на величину лага, то эту зависимость можно описать математически в виде *авторегрессии*:

$$x[t] = \Phi[0] + \Phi[1]x[t-1] + \Phi[2]x[t-2] + \Phi[3]x[t-3] + \dots, \quad (\text{ПЗ.7})$$

где $\Phi[j]$ – коэффициенты авторегрессии (АР).

Применительно к графику автокорреляционной функции рис. ПЗ.1, на котором выделяются два лага, равные 5 и 7 соответственно, можно говорить о том, что модель авторегрессии будет содержать две переменные: $x[t-5]$ и $x[t-7]$. Поскольку при лаге, равном пяти, коэффициент автокорреляции имеет положительный знак, то коэффициент при переменной $x[t-5]$ будет положительным, а так как коэффициент автокорреляции при лаге, равном семи, имеет отрицательный знак, что свидетельствует об обратной линейной зависимости, то и коэффициент при переменной $x[t-7]$ будет отрицательным:

$$x[t] = \Phi[0] + \Phi[5]x[t-5] - \Phi[7]x[t-7]. \quad (\text{ПЗ.8})$$

Для нахождения коэффициентов модели авторегрессии используются

Продолжение приложения 3

соответствующие разделы математической статистики, в большинстве случаев для этого используется метод наименьших квадратов. Остановимся на этом подробнее.

Как здесь уже установлено, в уравнении АР текущий отсчет принято представлять взвешенной суммой предыдущих с коэффициентами веса:

$$x[t] = \sum_{j=1}^p \Phi[j]x[t-j] + e[t], \quad (\text{ПЗ.9})$$

где $e[t]$ – некоррелированные случайные отсчеты, p – порядок модели АР, t – дискретная величина. Величина

$$\tilde{x}[t] = \sum_{j=1}^p \Phi[j]x[t-j] \quad (\text{ПЗ.10})$$

называется предсказанием случайной величины. Разность между текущим значением отсчета и его предсказанием называется ошибкой предсказания:

$$e[t] = x[t] - \tilde{x}[t] \quad (\text{ПЗ.11})$$

Величина $e[t]$ характеризует максимальную точность предсказания текущего отсчета, а ее статистические свойства определяют выбор порядка модели АР.

Характеристическим уравнением для стохастического разностного уравнения (ПЗ.10) называют [2] алгебраическое уравнение

$$x^i = \sum_{j=1}^i \Phi[j]x^{i-j}. \quad (\text{ПЗ.12})$$

Оно имеет i корней, которые обозначим $z_1 \dots z_i$. Корни характеристического уравнения (ПЗ.12) определяют важные свойства процесса $x[t]$, удовлетворяющего уравнению (ПЗ.10). Эти свойства можно сформулировать следующим образом.

1. Если все корни характеристического уравнения (ПЗ.12) по абсолютной величине меньше 1, то $x[t]$ можно представить бесконечной линейной комбинацией случайных величин $x[t]$, $x[t-1]$, ... и,

следовательно, $x[t]$ не коррелирует с $x[t+1]$, $x[t+2]$, При этом последовательность $x[t]$ будет стационарной в широком смысле.

2. Если стационарный СП $x[t]$ удовлетворяет уравнению (ПЗ.10), то характеристическое уравнение (ПЗ.12), корни которого имеет хотя бы один корень, равный единице, то с вероятностью 1 все значения этой последовательности совпадают.

Видим, что сформулированные свойства можно использовать для определения устойчивости дискретной модели авторегрессии. Доказательства этих свойств можно найти в работах [3, 8]. В дальнейшем ограничиваемся рассмотрением случая, когда все корни характеристического уравнения (ПЗ.12) по абсолютной величине меньше единицы.

Продолжение приложения 3

Из (ПЗ.10) видно, что построение АР модели случайного процесса сводится к нахождению коэффициентов АР и определению порядка p . Умножив правую и левую части (ПЗ.10) на $x[t-j]$, а затем усреднив, можно получить систему $p+1$ уравнений:

$$\begin{aligned} R[i] - \sum_{j=1}^p \Phi[j]R[j-i] &= 0 \\ R[0] - \sum_{j=1}^p \Phi[j]R[j] &= D_a \end{aligned} \quad (ПЗ.13)$$

где $R[i] = E\{x[t]x[t-i]\}$ – значения функции корреляции (операция нахождения среднего путем интегрирования по времени следует заменена на операцию суммирования E , т.к. в рассмотрении дискретные величины) случайного процесса, $i=1 \dots p$, D_a – дисперсия ошибок предсказания модели АР, $R[0]$ – дисперсия случайного процесса. Набор уравнений (ПЗ.13) называется полной системой уравнений Юла-Уолкера. Решением этой системы являются коэффициенты АР и дисперсия ошибок предсказания. При выводе уравнений было учтено, что

$$\begin{aligned} E\{e[t]e[t-i]\} &= 0, \quad i \neq 0, \quad i > 0, \\ E\{e[t]x[t-i]\} &= 0, \quad i \neq 0, \quad i > 0. \end{aligned} \quad (ПЗ.14)$$

Соотношения (ПЗ.14) следуют из некоррелированности ошибок предсказания. Решение системы уравнений (ПЗ.14) можно представить в матричном виде:

$$\bar{\Phi} = R_p^{-1} \bar{R}, \quad (ПЗ.15)$$

где

$$\bar{\Phi} = \begin{pmatrix} \Phi[1] \\ \Phi[2] \\ \dots \\ \Phi[p] \end{pmatrix}, \quad R_p = \begin{pmatrix} R[0] & R[1] & \dots & R[p-1] \\ R[1] & R[0] & \dots & R[p-2] \\ \dots & \dots & \dots & \dots \\ R[p-1] & R[p-2] & \dots & R[0] \end{pmatrix}, \quad \bar{R} = \begin{pmatrix} R[1] \\ R[2] \\ \dots \\ R[p] \end{pmatrix}.$$

Как видно из (ПЗ.13), уравнение не изменится, если вместо $R[i]$ использовать нормированные значения функции корреляции $r[i] = R[i]/R[0]$, т.е. коэффициенты корреляции. Очевидно, что при этом параметры модели АР останутся прежними. Как следует из (ПЗ.13), для модели АР первого порядка

$$\Phi[1] = r[1] \quad (\text{ПЗ.16})$$

Для модели АР второго порядка коэффициенты АР равны:

$$\begin{aligned} \Phi[1] &= r[1](1 - r[2]) / (1 - r^2[1]), \\ \Phi[2] &= (r[2] - r^2[1]) / (1 - r^2[1]). \end{aligned} \quad (\text{ПЗ.17})$$

Коэффициенты АР, рассчитанные с помощью уравнений Юла-Уолкера (ПЗ.13) минимизируют дисперсию ошибки предсказания:

Продолжение приложения 3

$$D_a = E \left\{ x[t] - \sum_{j=1}^p \Phi[j] x[t-j] \right\}. \quad (\text{ПЗ.18})$$

Возникает вопрос, что будет, если пропустить входной сигнал через фильтр, построенный на основе модели авторегрессии? Можно получить обеляющий фильтр. Он строится на основе уравнения (ПЗ.11) в виде поперечной структуры трансверсального фильтра, показанной на рис. ПЗ.2.

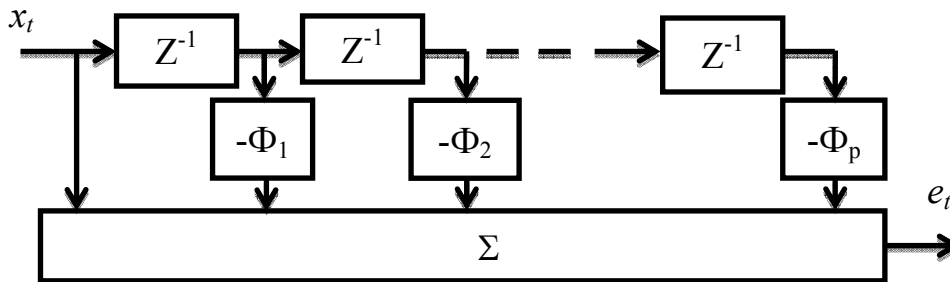


Рис. ПЗ.2. Структура обеляющего фильтра

Такие фильтры можно использовать для обработки входных данных так, чтобы случайные погрешности измерений подчинить гауссовскому закону распределения, что существенно облегчает решение задач идентификации и прогноза.

Идея фильтрации сигналов заложена в моделях авторегрессии, которые принято называть моделями Бокса-Дженкинса.

Дж. Бокс и Г. Дженкинс [3] для описания АР модели широко пользуются различного рода операторами. Первый из них – это оператор сдвига назад $B: Bx[t] = x[t-1]$, а для сдвига на m шагов назад: $B^m x[t] = x[t-m]$. Обратная операция выполняется оператором сдвига вперед: $F = B^{-1}: Fx[t] = x[t+1]$, а для сдвига на m шагов вперед: $F^m x[t] = x[t+m]$. Кроме этих операторов предлагается использовать

разностный оператор со сдвигом назад: $\Delta x[t] = x[t] - x[t-1] = (1-B)x[t]$. В свою очередь оператор, обратный разностному оператору, является оператором суммирования и выражается так:

$$\begin{aligned} Sx[t] = \Delta^{-1}x[t] &= \sum_{\tau=0}^{\infty} x[t-\tau] = x[t] + x[t-1] + x[t-2] + \dots = \\ &= (1 + B + B^2 + \dots)x[t] = (1-B)^{-1}x[t]. \end{aligned}$$

С учетом введенных операторов АР модель вида:

$$x[t] = \Phi[1]x[t-1] + \Phi[2]x[t-2] + \dots + \Phi[p]x[t-p] + e[t]$$

представляет собой авторегрессию p -го порядка, которую, с учётом введённых операторов, можно записать в такой эквивалентной форме:

$$(1 - \Phi[1]B - \Phi[2]B^2 - \dots - \Phi[p]B^p)x[t] = e[t] \quad \text{или} \quad f(B)x[t] = e[t].$$

Отсюда можно выразить текущий отчет временного ряда в виде:

Окончание приложения 3

$$x[t] = f^{-1}(B)e[t]. \quad (\text{П3.19})$$

В этой форме записи появляется новая трактовка авторегрессионной взаимосвязи – белый шум $e[t]$ как бы фильтруется через соответствующий оператори превращается в переменную $x[t]$. Поэтому иногда говорят, что рассматривается модель линейной фильтрации, причем линейной она является потому, что используется линейный фильтр $f^{-1}(B)$. Полученная запись более компактная и на основе этого принципа Дж. Бокс и Г. Дженкинс предложили многообразные модификации и расширения моделей авторегрессии для различных задач моделирования и прогнозирования стационарных процессов, которые уже рассмотрены в разделе 1 пособия. Модели фильтрации представляют собой достаточно большой раздел прогнозирования временных рядов [3, 8].

Отдельно рассматривается задача построения нелинейных АР моделей, у которых существенны вопросы обеспечения устойчивости. Можно заметить, что например нелинейная модель вида:

$$x[t] = \Phi[0] + \Phi[1]x[t-1]^2 \quad (\text{П3.20})$$

может быть неустойчивой, поскольку содержит нелинейный регрессор $x[t-1]^2$ значения ряда на каждом последующем шаге возрастают по сравнению с предыдущим во второй степени. Поэтому, если значения ряда превышают единицу – ряд возрастает, а если они меньше единицы, то составляющие ряда стремятся к нулю с ростом числа шагов. Зная эти особенности моделей, на практике используют «пороговые модели», когда, при достижении некоторого порогового значения значений моделируемого ряда, происходит изменение модели авторегрессии, или используют билинейные модели, когда предыдущие значения ряда раскладываются на две составляющие и некоторые другие (см. также раздел 1 пособия).

В заключение отметим, что порядок модели можно находить из условия не убывания дисперсии ошибки предсказания при дальнейшем повышении порядка. Эффективным методом определения порядка модели АР является метод, основанный на проверке близости корреляционной функции случайного процесса на выходе обесцараженного АР фильтра к корреляционной функции белого шума и другие.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Льюнг, Л. Идентификация систем. Теория для пользователя / Л. Льюнг; перевод с англ. А.С. Манделя; под ред. Я.З. Цыпкина. – М.: Наука, 1991. – 432 с.
2. Эйкхофф, П. Основы идентификации систем управления / П. Эйкхофф; перевод с англ. А.С. Манделя. – М.: Мир, 1975. – 676 с.
3. Бокс, Дж. Анализ временных рядов. Прогноз и управление / Дж. Бокс, Г. Дженникс. – М.: Мир, 1974. – Вып.1. – 406 с.
4. Дейч, А.М. Методы идентификации динамических объектов / А.М. Дейч. – М.: Энергия, 1979. – 240 с.
5. Гроп, Д. Методы идентификации динамических систем / Д. Гроп; перевод с англ. В.А. Васильева; под ред. Е.И. Кринецкого. – М.: Мир, 1979. – 294 с.
6. Дьяконов, В. MATLAB. Анализ, идентификация и моделирование систем / В. Дьяконов, В. Круглов. – СПб.: Питер, 2003. – 444 с.
7. Андерсон, Т. Статистический анализ временных рядов / Т. Андерсон. – М.: Мир, 1976. – 757 с.
8. Бесекерский, В.А. Теория систем автоматического управления / В.А. Бесекерский, Е.П. Попов. – 4-е изд., перераб. и доп. – СПб.: Профессия, 2003. – 752 с.
9. Баскаков, С.И. Радиотехнические цепи и сигналы / С.И. Баскаков. – 3-е изд., перераб. и доп. – М.: Высшая школа, 2000. – 462 с.
10. Сергиенко, А.Б. Цифровая обработка сигналов: учебник для вузов / А.Б. Сергиенко. – СПб.: Питер, 2006. – 751 с.
11. Беклемишев, Д.В. Дополнительные главы линейной алгебры / Д.В. Беклемишев. – М.: Наука, 1983. – 336 с.
12. Документация пакета System Identification Toolbox. – <http://www.mathworks.com/help/ident/ref/>
13. <http://www.inftech.webservis.ru/it/conference/scm/2000/session3/gubarev.htm>.
14. <http://www-rohan.sdsu.edu/doc/matlab/ /toolbox/ident/cra.html>.
15. <http://www.mathworks.com/help/ident/ref/procest.html>.
16. <http://www.mathworks.com/help/ident/ref/idproc.html>.
17. http://www.mbureau.ru/articles/dissertaciya-model-prognozirovaniya-vremennyh-ryadov-glava-1#p_1.