

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Государственное образовательное учреждение высшего
профессионального образования
«Поволжский государственный университет
телекоммуникаций и информатики»
(ПГУТИ)

Факультет базового телекоммуникационного образования
(ФБТО)

Кафедра теоретических основ радиотехники и связи
(ТОРС)

**Задания и методические указания
к выполнению курсовой работы по курсу
"ИДЕНТИФИКАЦИЯ И ДИАГНОСТИКА СИСТЕМ"**

зав. каф. ТОРС, проф., д.т.н. Горячкин О.В.
асс., м.н.с. Березовский А.А.

Редактор: зав. каф. ТОРС, проф., д.т.н. Горячкин О.В.

Самара, 2011г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

- 1.1. Основные понятия курса ИДС
- 1.2. Компенсация линейных искажений в скалярном канале формирования изображений
- 1.3. Слепая идентификация линейного векторного канала формирования изображений

2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

- 2.1. Описание предметной области
- 2.2. Количественные параметры ТЗ

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

- 3.1. Некоторые сведения о среде программирования Scilab
- 3.2. Имитация линейных искажений в канале формирования изображений
- 3.3. Коррекция линейных искажений в скалярном канале формирования изображений с известными характеристиками (фильтр Колмогорова-Винера)
- 3.4. Слепая коррекция линейных искажений в векторном канале формирования изображений

ЛИТЕРАТУРА

ВВЕДЕНИЕ

Курсовая работа по курсу «Идентификация и диагностика систем» выполняется в 9-м семестре и состоит из трех частей, каждая из которых предполагает самостоятельную разработку программного обеспечения имитационного математического моделирования многомерных каналов линейных динамических систем. Все программное обеспечение разрабатывается в среде Scilab. Результаты выполнения курсовой работы представляются в виде напечатанного отчета. При оформлении отчета по курсовой работе все результаты должны сопровождаться подробными комментариями, примерами работы программ и результатами.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Некоторые основные понятия курса ИДС

"Формирование моделей на основе результатов наблюдений и исследование их свойств - вот, по существу основное содержание науки" [1].

В рамках курса "Идентификация и прогнозирование систем" рассматривается проблематика построения математических моделей динамических систем по статистическим данным наблюдений за их поведением в процессе организованного эксперимента.

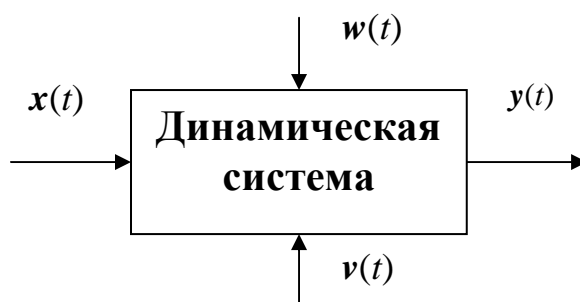


Рисунок 1 - Соотношения между переменными идентифицируемой системы

На рисунке 1 показаны основные соотношения между наблюдаемыми выходными сигналами - $y(t)$, входными сигналами - $x(t)$, регистрируемой помехой - $w(t)$ и нерегистрируемой помехой $v(t)$.

Классическая постановка задачи идентификации системы предполагает отбор математической модели системы из множества

моделей кандидатов, объединенных нашим априорным знанием, физической природой, удобством математического описания и другими разумными соображениями, определение параметров выбранной математической модели по наблюдаемым сигналам, при известных входных сигналах и зарегистрированных помехах и неизвестных сигналах нерегистрируемых помех.

В огромном количестве приложений широкий класс физических моделей для динамических систем может быть описан в рамках линейной модели взаимодействия входных, выходных сигналов и помех.

В наиболее общем случае модель линейной системы в непрерывном времени описывается следующим выражением:

$$\mathbf{y}(t) = \int_{-\infty}^t \mathbf{H}(t, \tau) \mathbf{x}(\tau) d\tau + \mathbf{v}(t), \quad (1)$$

где: $\mathbf{y}(t)$ - наблюдаемый векторный сигнал со значениями в \mathbf{C}^m , $\mathbf{H}(t, \tau)$ - $m \times n$ неизвестная матрица импульсных характеристик (ИХ) с элементами $\{h_{i,j}(\tau)\}$; $\mathbf{v}(t)$ - аддитивная нерегистрируемая помеха (векторный случайный процесс со значениями в \mathbf{C}^m , как правило с независимыми компонентами); $\mathbf{x}(t)$ - информационный сигнал со значениями в \mathbf{C}^n .

Напомним, что импульсной характеристикой линейной системы называют реакцию системы на i -м выходе при подаче на j -й вход δ -функции.

Системы, описываемые выражением (1) называют системами с множественным входом и множественным выходом (в англоязычной литературе Multiple-Input Multiple-Output или MIMO).

В частном случае, когда $\mathbf{H}(t, \tau) = \mathbf{H}(t - \tau)$ мы имеем стационарную линейную систему, при этом (1) имеет вид:

$$\mathbf{y}(t) = \int_{-\infty}^t \mathbf{H}(t - \tau) \mathbf{x}(\tau) d\tau + \mathbf{v}(t). \quad (2)$$

Если в (2) $n = 1$ и $m > 1$, то модель системы может быть описана более простым выражением:

$$\mathbf{y}(t) = \int_{-\infty}^t \mathbf{h}(t - \tau) x(\tau) d\tau + \mathbf{v}(t), \quad (3)$$

где $\mathbf{h}(\tau)$ - неизвестная импульсная характеристика m -мерного канала; $x(\tau)$ - комплексный информационный сигнал со значениями в \mathbb{C} .

Системы, описываемые моделями вида (3) называют системами с одним входом и множественным выходом (Single-Input Multiple-Output или SIMO).

В случае, если $n = 1$ и $m = 1$, то мы имеем модель системы с одним входом и выходом (Single-Input Single-Output или SISO), описываемую известным интегралом Дюамеля:

$$y(t) = \int_{-\infty}^t h(t - \tau)x(\tau)d\tau + v(t). \quad (4)$$

Задачи идентификации системы на основе моделей (1-3) и (4) далее мы будем называть задачами идентификации векторного и скалярного канала соответственно.

В случае дискретизации по времени выходных, наблюдаемых сигналов линейной системы с одним входом и множественным выходом (SIMO) модель системы можно приближенно представить в виде линейной дискретной свертки

$$\mathbf{y}(l) = \mathbf{y}(t)|_{t=lT} = \sum_{n=0}^{L-1} \mathbf{h}(n)x(l-n) + \mathbf{v}(l), \quad (5)$$

где: $x(l)$ - информационная последовательность, $\mathbf{h}(l)$ - неизвестная векторная импульсная характеристика дискретного канала системы, L - максимальная длина дискретной импульсной характеристики векторного канала (системы), $\mathbf{v}(l)$ - последовательность отсчетов шума.

Под идентифицируемостью линейной системы понимается возможность восстановления импульсной характеристики системы с точностью до комплексного множителя по выходным сигналам, при известных входных сигналах.

Рассмотрим условия идентифицируемости дискретной линейной системы, описываемой моделью SIMO (5). Пусть $\mathbf{y}^{(k)} = (y_0^{(k)}, \dots, y_{N-1}^{(k)})$ - вектор отсчетов выходного сигнала k -го канала, тогда

$$\mathbf{y}^{(k)} = \mathbf{X}_H(L)\mathbf{h}^{(k)}, \quad (6)$$

$$\mathbf{X}_H(L) = \begin{pmatrix} x_0 & x_1 & \dots & x_{L-1} \\ x_1 & x_2 & \dots & x_L \\ x_2 & x_3 & \dots & x_{L+1} \\ \vdots & \vdots & & \vdots \\ x_{N-1} & x_N & & x_{N+L-2} \end{pmatrix},$$

где $\mathbf{h}^{(k)} = (h_0^{(k)}, \dots, h_{L-1}^{(k)})$, $\mathbf{X}_H(L)$ - ганкелева матрица, составленная из отсчетов информационной последовательности $x_i = x(-L+1+i)$. Тогда идентифицируемость SIMO системы сразу следует из идентифицируемости каждого скалярного канала. На этот счет известна следующая теорема, являющиеся прямым следствием теоремы Кронекера-Капелли.

Теорема 1. Для идентифицируемости скалярного канала по известной информационной последовательности для любых значений $\mathbf{h} = (h_0, \dots, h_{L-1})$ необходимо и достаточно, чтобы $\text{rank}(\mathbf{X}_H(L)) \geq L$.

Рассмотрим теперь задачу слепой идентификации векторного канала. Условия слепой идентифицируемости канала обычно формулируются в отсутствие шумов. При этом различают задачи статистической и детерминированной идентификации, имея в виду модель информационной последовательности.

С практической точки зрения это означает, что в случае детерминированной идентификации нам доступны одна или крайне ограниченное количество реализаций входного сигнала, для статистической идентификации мы имеем в принципе неограниченную выборку.

Под идентифицируемостью системы вслепую понимается возможность восстановления передаточной функции и/или импульсной характеристики канала с точностью до комплексного множителя только по выходным сигналам.

Пусть идентифицируемый канал описывается следующими выражениями:

$$y_l^{(k)}(z) = \sum_{i=0}^{L-1} h_i^{(k)} x_{i+l}(z). \quad (7)$$

В этом выражении: $y_l^{(k)}(z)$ - полином степени $(t-1)$ над полем комплексных чисел, образованный блоком из t отсчетов на выходе k -го канала, $k = 1, \dots, M$, $l = 0, \dots, N-1$ - номер блока выходных отсчетов; $L = \max\{L_1, \dots, L_M\}$ - максимальная длина векторного канала; $x_l(z)$ - полином степени $(t-1)$ над полем комплексных чисел, образованный блоком из t информационных отсчетов на входе канала.

Наша задача найти условия, которым должны удовлетворять информационная последовательность и отсчеты векторного канала, при выполнении которых набор полиномов $\{y_l^{(1)}(z), \dots, y_l^{(M)}(z)\}$ кольца $C[z]$, $l = 0, \dots, N-1$ однозначно характеризует коэффициенты канала $h_0^{(1)}, \dots, h_{L-1}^{(1)}, \dots, h_0^{(M)}, \dots, h_{L-1}^{(M)}$ и коэффициенты информационной последовательности.

Пример 1. Пусть $M = 2$ и пусть известно, что $L = 2$.

а) Если $\mathbf{y}^{(1)} = (0, -2, 0, 0)$, $\mathbf{y}^{(2)} = (2, 0, -2, 2)$ то задача слепой идентификации имеет единственное решение $\mathbf{h}^{(1)} = (1, 1)$, $\mathbf{h}^{(2)} = (1, -1)$ и $\mathbf{x} = (1, -1 - 1, 1, -1)$.

б) Если $\mathbf{y}^{(1)} = (2, 0, -2, 0)$, $\mathbf{y}^{(2)} = (0, 2, 0, -2)$, то задача слепой идентификации имеет, по крайней мере, два решения 1) $\mathbf{h}^{(1)} = (1, 1)$, $\mathbf{h}^{(2)} = (1, -1)$ и $\mathbf{x} = (1, 1 - 1, -1, 1)$, 2) $\mathbf{h}^{(1)} = (1, -1)$, $\mathbf{h}^{(2)} = (-1, -1)$ и $\mathbf{x} = (1, -1 - 1, 1, 1)$.

Заметим, что если бы мы знали информационную последовательность, т.е. решали задачу классической идентификации, то в обоих случаях имелось бы единственное решение для канала. Действительно, в рассмотренном примере $\text{rank}(\mathbf{X}_H(2)) = 2$ и соответственно имеется единственное нормальное решение для каждого канала при известном входе.

Однако как это видно из примера задача слепой идентификации требует значительно более жестких ограничений на информационную последовательность, чем задача классической идентификации.

Рассмотрим случай детерминированной идентификации векторного канала для $M = 2$ в полиномиальной интерпретации.

Анализируя структуру преобразования (7) легко заметить, что если $N = L$, то справедливо следующее соотношение:

$$\sum_{l=0}^{L-1} y_l^{(1)}(z) h_l^{(2)} - \sum_{l=0}^{L-1} y_l^{(2)}(z) h_l^{(1)} = 0. \quad (8)$$

В этом уравнении $2L$ неизвестных $h_0^{(1)}, \dots, h_{L-1}^{(1)}, h_0^{(2)}, \dots, h_{L-1}^{(2)}$. Выберем $2L$ различных значений формальной переменной z_1, \dots, z_{2L} . Тогда используя (8) мы можем записать $2L$ однородных линейных уравнений относительно $2L$ неизвестных коэффициентов канала.

В матричной форме, получим:

$$\mathbf{Y}_1(z_1, \dots, z_{2L}) \mathbf{h} = \begin{pmatrix} y_0^{(1)}(z_1) & \dots & y_{L-1}^{(1)}(z_1) & -y_0^{(2)}(z_1) & \dots & -y_{L-1}^{(2)}(z_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y_0^{(1)}(z_{2L}) & \dots & y_{L-1}^{(1)}(z_{2L}) & -y_0^{(2)}(z_{2L}) & \dots & -y_{L-1}^{(2)}(z_{2L}) \end{pmatrix} \cdot \begin{pmatrix} h_0^{(2)} \\ \vdots \\ h_{L-1}^{(2)} \\ h_0^{(1)} \\ \vdots \\ h_{L-1}^{(1)} \end{pmatrix} = 0 \quad (9)$$

Для того, чтобы система линейных уравнений (9) имела единственное нетривиальное решение в соответствии с теоремой Кронекера-Капелли необходимо и достаточно чтобы ранг матрицы $\mathbf{Y}_1(z_1, \dots, z_{2L})$ был равен $(2L-1)$. Используя формулы (7) матрицу системы уравнений (9) можно представить в виде:

$$\mathbf{Y}_1(z_1, \dots, z_{2L}) = \begin{pmatrix} x_0(z_1) & \dots & x_{2L-2}(z_1) \\ \vdots & & \vdots \\ x_0(z_{2L}) & \dots & x_{2L-2}(z_{2L}) \end{pmatrix} \cdot \begin{pmatrix} h_0^{(1)} & \dots & 0 & -h_0^{(2)} & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{L-1}^{(1)} & & h_0^{(1)} & -h_{L-1}^{(2)} & & -h_0^{(2)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & h_{L-1}^{(1)} & 0 & \dots & -h_{L-1}^{(2)} \end{pmatrix} = (10)$$

$$= \mathbf{X}(z_1, \dots, z_{2L}) \mathbf{H}_{1,2}.$$

Т.о. для идентифицируемости системы необходимо и достаточно, чтобы выполнялись два условия:

1) ранг $(2L \times 2L - 1)$ матрицы $\mathbf{X}(z_1, \dots, z_{2L})$ равен $(2L-1)$, для любых различных z_1, \dots, z_{2L} ;

2) ранг $(2L-1 \times 2L)$ матрицы $\mathbf{H}_{1,2}$ равен $(2L-1)$.

Для того чтобы матрица $\mathbf{H}_{1,2}$ имела ранг $(2L-1)$ необходимо что бы нашелся не равный нулю минор порядка $(2L-1)$. Используя перестановку строк матрицу $\mathbf{H}_{1,2}$ для случая $L_2 \geq L_1$ можно представить в виде:

$$\mathbf{H}_{1,2} = \mathbf{T}^{-1} \begin{pmatrix} & & & 0 & \dots & 0 & 0 & 0 \\ & & & \vdots & & & & \vdots \\ & & & -h_0^{(2)} & & & & h_0^{(1)} \\ & & & \vdots & \ddots & & \vdots & \vdots \\ & & & -h_{L_2-1}^{(2)} & & -h_{L_2-1}^{(2)} & & h_{L_1-1}^{(1)} \\ & & 0 & \vdots & \ddots & \vdots & & \vdots \\ & & & \vdots & & \vdots & & \vdots \\ & & & 0 & \dots & -h_{L_2-1}^{(2)} & & 0 \end{pmatrix}. \quad (11)$$

Где \mathbf{T} - матрица перестановки, $Syl(h_1(z), h_2(z))$ это матрица Сильвестра, образованная коэффициентами полиномов $h_1(z)$ и $-h_2(z)$, которые соответствуют каналам $h_0^{(1)}, \dots, h_{L_1-1}^{(1)}$ и $h_0^{(2)}, \dots, h_{L_2-1}^{(2)}$. Поскольку $\det(Syl(h_1(z), h_2(z)))$ равен результанту полиномов $h_1(z)$ и $h_2(z)$, то в соответствии с известным фактом теории результантов [1] $\det(Syl(h_1(z), h_2(z))) = 0$ только если полиномы $h_1(z)$ и $h_2(z)$ имеют общий корень. Тогда если длина $L_2 = L$, то главный минор матрицы (8) равен $(-h_{L-1}^{(2)})^{L-L_1+1} \det(Syl(h_1(z), h_2(z)))$, т.е. не равен нулю.

Первое условие содержит в себе еще два ограничения которые становятся более очевидны, если представить матрицу $\mathbf{x}^{(z_1, \dots, z_{2L})}$ в виде:

$$\mathbf{X}^{(z_1, \dots, z_{2L})} = \begin{pmatrix} 1 & z_1 & \dots & z_1^{t-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_{2L} & \dots & z_{2L}^{t-1} \end{pmatrix} \cdot \begin{pmatrix} x_0 & x_1 & \dots & x_{2L-2} \\ x_1 & x_2 & \dots & x_{2L-1} \\ x_2 & x_3 & \dots & x_{2L} \\ \vdots & \vdots & & \vdots \\ x_t & x_{t+1} & & x_{t+2L-2} \end{pmatrix} = \quad (12)$$

$$= \mathbf{V}_t^{2L}(z_1, \dots, z_{2L}) \cdot \mathbf{X}_H(2L-1).$$

Где: $\mathbf{V}_t^{2L}(z_1, \dots, z_{2L})$ - матрица Вандермонда имеет ранг $2L-1$ если $t \geq 2L-1$, $\mathbf{X}_H(2L-1)$ - Ганкелева матрица, составленная из отсчетов информационной последовательности.

Линейная сложность детерминированной последовательности это наименьшее значение D такое, что $\mathbf{X}_H(D)$ имеет полный ранг по столбцам или существуют такие не равные нулю одновременно $\{\lambda_j\}$ для которых:

$$x_i = - \sum_{j=1}^D \lambda_j x_{i-j} \quad i = D, \dots, t + 2L - 2.$$

Линейная сложность характеризует степень предсказуемости детерминированной последовательности ограниченной длины. Для того чтобы матрица $X_H(2L-1)$ имела полный ранг по столбцам, линейная сложность информационной последовательности должна быть больше $(2L-2)$.

Теперь мы можем объяснить пример 1. В случае а) идентификация возможна, т.к. линейная сложность входной последовательности равна 3, в случае б) несмотря на то, что каналы не имеют общих нулей, идентификация неоднозначна т.к. линейная сложность входной последовательности равна 2.

Т.о. мы определили необходимые и достаточные условия идентифицируемости векторного канала для случая $M=2$. Сформулируем этот результат в виде следующей теоремы, обобщив его на случай $M > 2$.

Теорема 2. Для идентифицируемости детерминированного векторного канала необходимо и достаточно выполнения следующих условий:

- 1) полиномы $h_1(z), \dots, h_M(z)$ не должны иметь общих корней;
- 2) линейная сложность информационной последовательности должна быть больше $(2L-2)$;
- 3) длина информационной последовательности должна быть больше $(4L-3)$ или длина вектора данных больше $(3L-2)$.

Условия теоремы по существу гарантируют следующие интуитивные требования:

- 1) все каналы в системе должны отличаться друг от друга, например они не могут быть идентичны;
- 2) входная последовательность должна быть достаточно сложна. Она не может быть нулевой, константой или одиночной синусоидой;
- 3) в наличии должно быть достаточно отсчётов выхода.

1.2. Компенсация линейных искажений в скалярном канале формирования изображений

Задача компенсации искажений в системах формирования изображений (восстановление изображений) является одним из при-

ложений техники идентификации систем. Компенсация линейных искажений изображений различного происхождения (радиометрических, радиоастрономических, оптических, акустических, рентгеновских, инфракрасных) это задача восстановления двумерного, пространственно ограниченного, неотрицательного сигнала, искаженного линейным оператором.

Непрерывная модель такого сигнала также может быть описана в виде интеграла двумерной свертки

$$y(t, \tau) = \iint_D h(t - \theta, \tau - \sigma) x(\theta, \sigma) d\theta d\sigma + v(t, \tau), \quad (13)$$

где $y(t, \tau)$ и $x(\theta, \sigma)$ положительные, пространственно ограниченные функции, $h(\theta, \sigma)$ - импульсная характеристика искажающей системы (функция размытия точки).

Источники линейных искажений это, например дефокусировка объектива оптической системы формирования изображения, скоростной сдвиг (смаз) изображения вследствие движения объекта в процессе экспозиции, различного рода дифракционные ограничения (т.е. ограничение пространственного спектра изображения регистрирующим устройством), влияние среды распространения (например, атмосферная турбулентность).

Если исследователю известно изображение на входе линейной системы, то возникает задача идентификации импульсной характеристики искажающего канала. Если исследователю известна импульсная характеристика искажающего изображение линейного канала, тогда возникает задача коррекции изображения. Обе задачи эквивалентны в силу симметрии оператора свертки и могут быть решены разработкой линейного оптимального фильтра.

Пусть восстанавливаемое изображение является реализацией случайного процесса (поля) с известным априорным распределением, тогда алгоритм восстановления для простой функции потерь совпадает с алгоритмом максимума апостериорной вероятности (МАН), который с учетом особенностей задачи можно записать в виде

$$\hat{x}(t, \tau) = \arg \max_{x(t)} p(x(t, \tau) | y(t, \tau)) = \arg \max_{x(t)} p(y(t, \tau) | x(t, \tau)) p(x(t, \tau)), \quad (14)$$

где $p(x(t, \tau) | y(t, \tau))$ - апостериорное распределение восстанавливаемого сигнала, $p(y(t, \tau) | x(t, \tau))$ - функционал правдоподобия наблю-

даемого изображения, $p(x(t, \tau))$ - априорное распределение восстанавливаемого изображения.

Рассмотрим случай гауссовских случайных процессов. Пусть восстанавливаемый сигнал является реализацией гауссовского случайного процесса с корреляционной функцией $B_x(t_1, t_2, \tau_1, \tau_2)$ и нулевым математическим ожиданием. Шум в уравнении (13) также является гауссовским случайным процессом с корреляционной функцией $B_v(t_1, t_2, \tau_1, \tau_2)$ и нулевым математическим ожиданием.

Тогда мы можем легко записать функционал апостериорного распределения наблюдаемого сигнала в виде

$$\begin{aligned}
 p(x(\theta, \sigma) | y(t, \tau)) = C \times \\
 \times \exp \left\{ -\frac{1}{2} \iiint \iiint \left(y(t_1, \tau_1) - \iint_D h(t_1 - \theta, \tau_1 - \sigma) x(\theta, \sigma) d\theta d\sigma \right) \times \right. \\
 \left. \times B_v^{-1}(t_1, t_2, \tau_1, \tau_2) \left(y(t_2, \tau_2) - \iint_D h(t_2 - \theta, \tau_2 - \sigma) x(\theta, \sigma) d\theta d\sigma \right) dt_1 dt_2 d\tau_1 d\tau_2 \right\} \\
 \times \exp \left\{ -\frac{1}{2} \iiint \iiint x(\theta_1, \sigma_1) B_x^{-1}(\theta_1, \sigma_1, \theta_2, \sigma_2) x(\theta_2, \sigma_2) d\theta_1 d\theta_2 d\sigma_1 d\sigma_2 \right\}.
 \end{aligned} \tag{15}$$

Где $B_x^{-1}(\theta_1, \sigma_1, \theta_2, \sigma_2)$ и $B_v^{-1}(t_1, t_2, \tau_1, \tau_2)$ - обратные корреляционные функции.

Продифференцируем логарифм полученного функционала по искомому сигналу

$$\begin{aligned}
 d \ln(p(x(\theta, \sigma) | y(t, \tau))) = \iiint \iiint \iiint_D h(t_1 - \theta, \tau_1 - \sigma) g(\theta, \sigma) \times \\
 \times B_v^{-1}(t_1, t_2, \tau_1, \tau_2) y(t_2, \tau_2) d\theta d\sigma dt_1 dt_2 d\tau_1 d\tau_2 - \\
 - \iiint \iiint \iiint_D \iiint_D B_v^{-1}(t_1, t_2, \tau_1, \tau_2) h(t_1 - \theta, \tau_1 - \sigma) \times \\
 \times x(\theta_1, \sigma_1) h(t_2 - \theta, \tau_2 - \sigma) g(\theta_2, \sigma_2) d\theta_1 d\sigma_1 d\theta_2 d\sigma_2 dt_1 dt_2 d\tau_1 d\tau_2 \\
 - \iiint \iiint x(\theta_1, \sigma_1) B_x^{-1}(\theta_1, \sigma_1, \theta_2, \sigma_2) g(\theta_2, \sigma_2) d\theta_1 d\theta_2 d\sigma_1 d\sigma_2.
 \end{aligned} \tag{16}$$

Для нахождения решения приравняем полученный дифференциал к нулю, и учтем, что это равенство должно выполняться для любой функции $g(\theta_2, \sigma_2)$, тогда

$$\begin{aligned}
0 &= \iiint \iiint h(t_1 - \theta, \tau_1 - \sigma) B_v^{-1}(t_1, t_2, \tau_1, \tau_2) y(t_2, \tau_2) dt_1 dt_2 d\tau_1 d\tau_2 \\
&- \iiint \iiint \iiint_D B_v^{-1}(t_1, t_2, \tau_1, \tau_2) h(t_1 - \theta, \tau_1 - \sigma) x(\theta_1, \sigma_1) h(t_2 - \theta, \tau_2 - \sigma) d\theta d\sigma dt_1 dt_2 d\tau_1 d\tau_2 \\
&- \iint x(\theta_1, \sigma_1) B_x^{-1}(\theta_1, \sigma_1, \theta, \sigma) d\theta_1 d\sigma_1.
\end{aligned} \tag{17}$$

или

$$\begin{aligned}
&\iint x(\theta_1, \sigma_1) H(\theta_1, \sigma_1, \theta, \sigma) d\theta_1 d\sigma_1 = G(\theta, \sigma) \\
&H(\theta_1, \sigma_1, \theta, \sigma) = \\
&\iiint \iiint B_v^{-1}(t_1, t_2, \tau_1, \tau_2) h(t_1 - \theta_1, \tau_1 - \sigma_1) h(t_2 - \theta, \tau_2 - \sigma) dt_1 dt_2 d\tau_1 d\tau_2 + \\
&+ B_x^{-1}(\theta_1, \sigma_1, \theta, \sigma) \\
&G(\theta, \sigma) = \iiint \iiint h(t_1 - \theta, \tau_1 - \sigma) B_v^{-1}(t_1, t_2, \tau_1, \tau_2) y(t_2, \tau_2) dt_1 dt_2 d\tau_1 d\tau_2.
\end{aligned} \tag{18}$$

Последнее выражение представляет собой интегральное уравнение Фредгольма I-го рода относительно искомого сигнала. Решение уравнения Фредгольма I-го рода легко получить с помощью преобразования Фурье в случае, когда первый интеграл в (18) является сверткой. Тогда $H(\theta_1, \sigma_1, \theta, \sigma) = H(\theta_1 - \theta, \sigma_1 - \sigma)$, применяя теорему о свертки, после несложных преобразований искомое решение записывается в виде обратного двумерного преобразования Фурье

$$x(\theta, \sigma) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{G(j\omega_1, j\omega_2)}{H(j\omega_1, j\omega_2)} \exp(j(\omega_1\theta + \omega_2\sigma)) d\omega_1 d\omega_2. \tag{19}$$

$G(j\omega_1, j\omega_2)$, $H(j\omega_1, j\omega_2)$ - спектральные плотности, полученные вычислением соответствующих прямых преобразований Фурье:

$$G(j\omega_1, j\omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\theta, \sigma) \exp(-j(\omega_1\theta + \omega_2\sigma)) d\theta d\sigma, \tag{20}$$

$$H(j\omega_1, j\omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\theta, \sigma) \exp(-j(\omega_1\theta + \omega_2\sigma)) d\theta d\sigma. \tag{21}$$

Если восстанавливаемое изображение и шум являются стационарными по двум координатам случайными процессами (однород-

ными случайными полями) с энергетическими спектрами $F_x(\omega_1, \omega_2)$ и $F_v(\omega_1, \omega_2)$ соответственно, тогда

$$x(\theta, \sigma) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{h^*(j\omega_1, j\omega_2)y(j\omega_1, j\omega_2)}{|h(j\omega_1, j\omega_2)|^2 + \frac{F_v(\omega_1, \omega_2)}{F_x(\omega_1, \omega_2)}} \exp(j(\omega_1\theta + \omega_2\sigma)) d\omega_1 d\omega_2. \quad (22)$$

Т.о. мы получили выражение двумерного фильтра Винера, который является оптимальным восстанавливающим фильтром. В этом выражении:

$$h(j\omega_1, j\omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\theta, \sigma) \exp(-j(\omega_1\theta + \omega_2\sigma)) d\theta d\sigma, \quad (23)$$

$$y(j\omega_1, j\omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y(\theta, \sigma) \exp(-j(\omega_1\theta + \omega_2\sigma)) d\theta d\sigma. \quad (24)$$

Используя (22) легко перейти к задаче восстановления цифровых изображений, искажение которых задано в виде двумерной дискретной свертки

$$y(k, m) = \sum_l \sum_n h(k-l, m-n)x(l, n) + v(k, m). \quad (25)$$

Перепишем (22) в виде обратного двумерного дискретного преобразования Фурье и учтем, что восстанавливаемое изображение и шумовое изображения имеют некоррелированные отсчеты (пиксели), тогда

$$x(l, n) = \frac{1}{N^2} \sum_k \sum_m \frac{h_\omega^*(k, m)y_\omega(k, m)}{|h_\omega(k, m)|^2 + SNR^{-1}} \exp\left(j\frac{2\pi}{N}(kl + mn)\right). \quad (26)$$

В последнем выражении SNR - отношение сигнал/шум, заданное условием задачи, $x(l, n)$ - отсчеты восстанавливаемого изображения, $h_\omega(k, m)$, $y_\omega(k, m)$ - отсчеты двумерных дискретных преобразований Фурье импульсной характеристики и наблюдаемого изображения, полученные с помощью дискретных преобразований Фурье:

$$h_\omega(l, n) = \sum_k \sum_m h(k, m) \exp\left(-j\frac{2\pi}{N}(kl + mn)\right),$$

$$y_{\omega}(l, n) = \sum_k \sum_m y(k, m) \exp\left(-j \frac{2\pi}{N} (kl + mn)\right). \quad (27)$$

1.3. Слепая идентификация линейного векторного канала формирования изображений

Слепая коррекция изображений (blind image deconvolution) задача, возникающая в случае отсутствия априорной информации об ИХ канала формирования. Особенно актуальна задача слепой коррекции линейных искажений изображений в задачах дистанционного зондирования Земли, астрономии, медицине.

Возможности слепой идентификации скалярных двумерных каналов несколько шире, чем одномерных. Это обстоятельство не раз отмечалось в литературе [2] и исторически привело к более интенсивному внедрению методов слепой обработки в данном случае.

Хорошо известно, например, что ковариационные функции стационарного процесса на выходе линейной системы не содержат информации о фазе её передаточной функции, и слепая идентификация канала по модулю передаточной функции возможна только для узкого класса систем с минимальной фазой.

Интересно, что для дискретных случайных полей это, вообще говоря, не так. Т.е. для двумерных дискретных сигналов возможности восстановления фазы по модулю передаточной функции значительно шире. Этот несколько неожиданный результат был получен методом математического моделирования Фьенапом в 1978г.

Объяснение этому факту заключается в том, что в кольце полиномов от двух и более переменных над полем комплексных чисел существует достаточно мощное множество неприводимых полиномов в отличие от кольца полиномов от одной переменной где, как известно, не существует неприводимых полиномов, степень которых больше 1.

Поэтому если двумерный дискретный сигнал имеет z -преобразование, неразложимое на более простые множители, то очевидно, используя единственность факторизации многочлена на неприводимые множители мы можем восстановить дискретный сигнал по его автокорреляции или что эквивалентно восстановлению фазы передаточной функции по ее амплитудному спектру [2].

Естественно, что данное свойство двумерных сигналов можно использовать и для решения задачи детерминированной слепой идентификации канала формирования изображения.

Рассмотрим модель двумерной дискретной свертки, которой запишем в виде произведения полиномов кольца $C[z_1, z_2]$:

$$y(z_1, z_2) = h(z_1, z_2)x(z_1, z_2) \quad (28)$$

где:

$$y(z_1, z_2) = \sum_l \sum_n y(l, n)z_1^l z_2^n; \quad h(z_1, z_2) = \sum_l \sum_n h(l, n)z_1^l z_2^n;$$

$$x(z_1, z_2) = \sum_l \sum_n x(l, n)z_1^l z_2^n$$

Если полиномы $h(z_1, z_2)$ и $x(z_1, z_2)$ неприводимы в кольце $C[z_1, z_2]$, то факторизуя $y(z_1, z_2)$ мы решаем проблему слепой идентификации.

Конечно, практическое применение подобного подхода существенно ограничено сложностью процедуры факторизации полиномов от многих переменных и наличием шума.

Алгоритм, имеющий некоторое практическое значение и основанный на свойстве неприводимости полиномов (28) известен как алгоритм «нулевого листа». Алгоритм использует свойства поверхностей, точки которых являются корнями полиномов канала и истинного изображения.

Дополнительным некоторым ограничением области применения данного подхода является использование предположения о пространственной ограниченности сигналов.

Помимо свойств z -преобразований от сигналов конечной протяженности для слепой идентификации используются также неотрицательность истинного изображения, различные параметрические модели функции рассеяния точки.

В данном разделе мы рассмотрим алгоритм слепой идентификации линейного векторного канала формирования изображений системы SIMO. В этом случае, если все ИХ подканалов в системе отличаются друг от друга, входное изображение достаточно сложно, в наличии достаточно отсчетов выходных изображений, то возможно восстановление всех ИХ подканалов одновременно "вслепую" (см. теорему 2).

Пусть наблюдаемые искаженные изображения заданы M двумерными дискретными свертками, искаженными независимыми аддитивными помехами.

$$y^i(k, m) = \sum_l \sum_n h^i(k-l, m-n)x(l, n) + v^i(k, m), \quad i = 1, \dots, M. \quad (29)$$

В отсутствии шума очевидна справедливость следующего равенства для неизвестных отсчетов импульсных характеристик

$$\sum_k \sum_m h^j(l-k, n-m)y^i(k, m) - \sum_k \sum_m h^i(l-k, n-m)y^j(k, m) = 0, \\ i = 1, \dots, M, \quad j = 1, \dots, M. \quad (30)$$

Наличие шума нарушает равенство, поэтому будем искать решение применяя универсальный метод наименьших квадратов, т.е.

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h}} \sum_i \sum_j d_{i,j} \\ \sum_l \sum_n \left| \sum_k \sum_m h^j(l-k, n-m)y^i(k, m) - \sum_k \sum_m h^i(l-k, n-m)y^j(k, m) \right|^2, \quad (31)$$

при дополнительном условии, исключающем тривиальное нулевое решение:

$$\sum_i \sum_k \sum_m |h^i(k, m)|^2 = 1, \quad (32)$$

где $d_{i,j}$ - нормирующие множители, необходимые в случае различия дисперсии аддитивного шума в подканалах.

Используя метод множителей Лагранжа, равенство Парсевалья для двумерного дискретного преобразования Фурье, получим следующее выражение

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h}} J(\mathbf{h}), \\ J(\mathbf{h}) = \sum_i \sum_j d_{i,j} \sum_l \sum_n \left| h_{\omega}^j(l, n)y_{\omega}^i(l, n) - h_{\omega}^i(l, n)y_{\omega}^j(k, m) \right|^2 + \lambda \sum_i \sum_k \sum_m \left| h_{\omega}^i(k, m) \right|^2. \quad (33)$$

Одним из основных методов решения задач типа (33) является метод градиентного спуска. В рамках этого метода минимизация функционала $J(\mathbf{h})$ проводится в рамках итерационного процесса

$$\mathbf{h}^k = \mathbf{h}^{k-1} - \beta_k \text{grad}(J(\mathbf{h}^{k-1})) \quad (34)$$

Вычисляя градиент выражения (33) получим следующую формулу для расчета спектральных отсчетов импульсных характеристик на k -м шаге итерации.

$$h_{\omega}^{j,k}(l,n) = (1 - \beta_k) h_{\omega}^{j,k-1}(l,n) + \beta_k \frac{y_{\omega}^j(l,n) \sum_{i,j \neq i} d_{i,j} h_{\omega}^{i,k-1}(l,n) y_{\omega}^{*j}(l,n)}{\sum_{i,j \neq i} d_{i,j} |y_{\omega}^j(l,n)|^2 + \lambda}. \quad (35)$$

2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

В рамках курсовой работы по курсу ИДС предлагается выполнение следующих заданий:

1. Разработка ПО в среде SciLab, реализующее работу линейной SIMO-системы формирования изображений.

Используйте выражение (29). Исходным изображением является полутонное изображение Вашего лица (используйте любой графический редактор для подготовки исходного изображения), в виде приемлемом для опознавания, при защите курсовой работы. Размер изображения произволен и ограничивается только вычислительной мощностью используемого ПК, рекомендуется не менее 320×240 .

Моделируемая SIMO-система (29) формирования изображений преобразует исходное изображение в три наблюдаемых изображения с помощью оператора линейной свертки с тремя различными функциями рассеяния точки (ФРТ), варианты параметров которых заданы в таблице 1.

Таблица 1 - Исходные данные.

Порядковый номер в группе	Первая ФРТ		Вторая ФРТ		Третья ФРТ	
	Длина смаза	Дисперсия	Длина смаза	Дисперсия	Длина смаза	Дисперсия
1	15	1,1	15	4	15	3,6
2	14	1,2	14	3,9	14	3,5

3	13	1,3	13	3,8	13	3,4
4	12	1,4	12	3,7	12	3,3
5	11	1,5	11	3,6	11	3,2
6	10	1,6	10	3,5	10	3,1
7	15	1,7	15	3,4	15	3
8	14	1,8	14	3,3	14	2,9
9	13	1,9	13	3,2	13	2,8
10	12	2	12	3,1	12	2,7
11	11	2,1	11	3	11	2,6
12	10	2,2	10	2,9	10	2,5
13	15	2,3	15	2,8	15	2,4
14	14	2,4	14	2,7	14	2,3
15	13	2,5	13	2,6	13	2,2
16	12	2,6	12	2,5	12	2,1
17	11	2,7	11	2,4	11	2
18	10	2,8	10	2,3	10	1,9
19	15	2,9	15	2,2	15	1,8
20	14	3	14	2,1	14	1,7
21	13	3,1	13	2	13	1,6
22	12	3,2	12	1,9	12	1,5
23	11	3,3	11	1,8	11	1,4
24	10	3,4	10	1,7	10	1,3
25	15	3,5	15	1,6	15	1,2
26	14	3,6	14	1,5	14	1,1
27	13	3,7	13	1,4	13	4
28	12	3,8	12	1,3	12	3,9
29	11	3,9	11	1,2	11	3,8
30	10	4	10	1,1	10	3,7

Следующий программный код позволяет рассчитать три массива ФРТ с параметрами, заданными в таблице 1 и удовлетворяющие условиям теоремы 2.

```

//Генерирует смаз вправо с размытием
clear
a=10 //Длина смаза
c=4 //Дисперсия размытия
motionkern=zeros(2*a,2*a); // Предзаполнение ФРТ нулями
motionkern(a,a:2*a)=[a:-1:0]; //Генерация убывающего смаза
motionkern=imfilter(motionkern,fspecial('gaussian',a/2,c));
//Добавление к смазу размытия
motionkern=motionkern./sum(motionkern) //Нормировка
imwrite(motionkern,'h1_true.bmp');
//Генерирует смаз влево с размытием
clear

```

```

a=4 //Длина смаза
c=1 //Дисперсия размытия
motionkern=zeros(2*a,2*a); // Предзаполнение ФРТ нулями
motionkern(a,1:a)=[1:a]; //Генерация убывающего смаза
motionkern=imfilter(motionkern,fspecial('gaussian',a/2,c));
//Добавление к смазу размытия
motionkern=motionkern./sum(motionkern) //Нормировка
imwrite(motionkern,'h2_true.bmp');
//Генерирует смаз вниз с размытием
clear
a=15 //Длина смаза
c=2 //Дисперсия размытия
motionkern=zeros(2*a,2*a); // Предзаполнение ФРТ нулями
motionkern(a:2*a,a)=[a:-1:0]'; //Генерация убывающего смаза
motionkern=imfilter(motionkern,fspecial('gaussian',a/2,c));
//Добавление к смазу размытия
motionkern=motionkern./sum(motionkern) //Нормировка
imwrite(motionkern,'h3_true.bmp');

```

2. Разработка ПО в среде SciLab, реализующее алгоритм компенсации линейных искажений в скалярном канале формирования изображений.

3. Разработка ПО в среде SciLab, реализующее алгоритм слепой идентификации линейного векторного канала формирования изображений.

4. Результаты курсовой работы оформляются в виде напечатанного документа содержащего:

- 1) Краткое введение в рассматриваемую тематику;
- 2) Результаты выполнения пунктов 1)-3) данного ТЗ;
- 3) Выводы по работе.

Разработанные в рамках КР программы должны быть представлены в отчете в виде словесного описания, блок-схемы алгоритмов работы, требуемых исходных данных и результатов их обработки.

Всяческие рационализаторские предложения в части совершенствования ПО и самостоятельные изыскания в части алгоритмов приветствуются и поощряются.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

3.1. Некоторые сведения о среде программирования Scilab

Scilab - кроссплатформенное ПО с открытым исходным кодом для инженерных и математических расчетов использующее собственный язык программирования высокого уровня ориентированный на численные методы.

Scilab можно использовать для обработки сигналов, изображений, статистического анализа, численной оптимизации, решения задач динамического моделирования и многих других задач. Scilab является одной из многочисленных свободных альтернатив системе программирования MATLAB.

Синтаксис языка Scilab во многом совпадает с синтаксисом MATLAB. Последнюю версию Scilab можно найти здесь: <http://www.scilab.org>

Для выполнения курсовой работы должен быть установлен scilab версии не ниже 5.3.3, последней актуальной версии на момент составления задания.

Для выполнения задания также необходим пакет расширения Scilab Image and Video Processing версии не ниже 0.5.3-3 (SIVP). Установить SIVP можно двумя способами — в меню «Инструменты» командного окна Scilab выбрать пункт «Управление модулями ATOMS», в открывшемся окне слева найти каталог Image Processing, в нем пакет SIVP. При выборе пакета справа появится краткое описание пакета, ссылка на авторов и кнопка «Установить».

При запуске scilab запускается главное окно в котором можно вводить команды построчно, это наиболее простой способ использования Scilab.

Чтобы выбрать директорию для чтения и записи файлов (рабочую папку) нужно в меню «Файл» командного окна выбрать пункт «Сменить текущий каталог».

Для написания scilab-скрипта запустите встроенный текстовый редактор с подсветкой синтаксиса из меню «Инструменты» или кнопкой в верхнем левом углу экрана. Набрав текст программы нажмите F5 для сохранения и запуска. После того как отработает

скрипт в командном окне будет находиться весь его вывод, в том числе сообщения об ошибках и предупреждениях, если они есть.

Переменные, использованные в текущей сессии можно увидеть запустив «Окно обзора переменных» из меню «Инструменты». Очистить переменные позволяет команда `clear`.

Для вызова подробной справки по Scilab нажмите F1 находясь в командном окне. Нажатие F1 в окне редактора вызовет справку по редактору.

Приведем некоторые минимальные сведения по работе с Scilab. Справка по функциям вызывается строкой `help <имя функции>` и позволяет получить подробное и точное описание каждой упоминаемой здесь функции.

3.1.1. Многомерные массивы и действия над ними

Основными переменными в работе являются многомерные массивы, индексация которых соответствует, в основном индексации массивов в MATLAB [4]. В руководствах Scilab многомерный массив называют «гиперматрица».

Примеры работы с массивами иллюстрирует следующий программный код:

```
x=[1 2 3; 4 5 6; 7 8 9]//Задание матрицы
A=rand(5,5,5); //Генерация случайно гиперматрицы размера
5*5*5
B=zeros(8,8,8); //Генерация гиперматрицы нулей размером
8*8*8
A(:); //Вывод в столбик всех элементов матрицы A
A(:, :, 1); //Вывод первой подматрицы гиперматрицы A
A(1:2, 1:3, :); //Вывод для всех пяти подматриц гиперматрицы A
элементов 1-3 столбцов, 1-2 строк
A(1:2, 1:3, :) < A(3:4, 2:4, :); //Сравнение матриц
A=A(1:2, 1:3, :). * double(A(1:2, 1:3, :) < A(3:4, 2:4, :)) //Обнуление тех
элементов A(1:2, 1:3, :) которые больше A(3:4, 2:4, :)
m=size(A); //Вектор размеров по всем измерениям
гиперматрицы A
n=ndims(B); //Число измерений гиперматрицы B
```

3.1.2. Псевдослучайные числа

Для генерации массива псевдослучайных чисел с гауссовским распределением используются функция `rand(A, 'normal')`.

Если \mathbf{A} - некая матрица то будет сформирован двумерный массив случайных величин с тем же размером, что и матрица \mathbf{A} . Если вместо матрицы \mathbf{A} параметр задан в виде a, b , т.е. вызывается `rand(a,b,'normal')`, то будет сформирована матрица размером $a \times b$.

Пример формирования массива псевдослучайных чисел.

```
WGN=rand(truey,"normal");
```

3.1.3. Дискретное преобразование Фурье (ДПФ)

Для вычисления ДПФ в Scilab используются функция `fft(<массив данных>)` для прямого, и `ifft(<массив данных>)` для обратного. Размерность ДПФ определяется типом массива данных. Если входной массив - вектор, то вычисляется одномерное ДПФ, если входной массив - матрица, то вычисляется двумерное преобразование Фурье.

Примеры:

```
X=fft(x(1,:,1))//Одномерное преобразование Фурье от первой строки первой подматрицы гиперматрицы x
```

```
X=fft(x)//Двумерное ФФТ от матрицы x
```

```
x=ifft(X)//Обратное Фурье преобразования для матрицы X
```

3.1.4. Циклы и условия

В курсовой работе часто используются цикл типа `for`. Синтаксис оператора:

```
for [Переменная_цикла=вектор значений переменной]
[тело цикла]
end
```

Пример организации цикла:

```
for i=1:L
h(:,i)=real(ifft((conj(X)./(abs(X).^2+K)).*Y(:,i)))
end
```

Здесь также используются функции `real()` - реальная часть комплексного числа, для матриц вычисляется поэлементно, `abs()` - модуль, `.^` - поэлементное возведение в степень элементов матрицы, `conj()` - поэлементная операция комплексного сопряжения элементов матрицы.

Для проверки условий используется конструкция `if {условие} then тело условия end`.

Пример реализации условного оператора:

```
if i<>j then
    tmp1 = tmp1 + conj(Y(:,i)).*H2(:,i,kk-1)./d(j,i);
end;
```

В случае использования цикла типа `while` тело условия будет выполняться пока условие не станет истинным. Пример цикла `while` из справочной системы Scilab:

```
e=1; a=1; k=1;
while norm(a-(a+e),1) > %eps,
    e=e/2;
    k=k+1;
end
e,k
```

3.1.5. Работа с изображениями

Для чтения и записи изображений используются функции `imread` и `imwrite`, для переноса изображения в класс `double` в диапазоне значений `[0..1]` используется функция `im2double`

Примеры:

```
y(:,1)=im2double(imread('y1.bmp'));
imwrite(h(:,1,k),'h1.bmp');
```

Функция `fspecial` генерирует некоторые стандартные функции размытия точки (ФРТ) с мощностью единица и заданными параметрами, например, размером.

В частности строчка

```
h(:,:,j,k)=fspecial('gaussian',initPSFsize,1)
```

запишет в `h(:,:,j,k)` значения импульсной характеристики двумерного гауссова ФНЧ с размером `initPSFsize` и среднеквадратическим отклонением 1.

3.1.6. Измерение времени вычислений

Часто при проведении сложных расчетов целесообразно контролировать время вычислений. Для этого в начальной точке измерения помещается функция `tic`, в конце измеряемого промежутка времени значение функции `toc()` присваивается некоторой константе.

Пример:

```
tic
```

```
for j=1:L
```

```
h(:,:,j,k)=fspecial('gaussian',initPSFsize,1);
```

```
end
```

```
time=toc();
```

```
disp('Time');
```

```
disp(time);
```

Функция `disp()`, использованная в этом примере позволяет выводить на экран матрицы, строки и переменные.

3.2. Имитация линейных искажений в канале формирования изображений

При реализации модели линейной SIMO-системы формирования изображений Вам предлагается следующий программный код, часть которого скрыта (прямоугольные зоны текста), и должна быть восстановлена Вами в процессе выполнения курсовой работы:

```
//Моделирование искажений и зашумления
```

*//входным файлом является x_true.bmp, "неискаженное" изображение, в результате выполнения скрипта выходными являются
//файлы с ФРТ h1_true-h3_true.bmp и файлы с искаженными изображениями y1-y3.bmp*

// Режим отображения
mode(0);

// Отображать предупреждение для исключений при выполнении операций с плавающей запятой
ieee(1);

//Читаем истинное изображение и преобразуем его в класс double, учитывая что у прочитанного изображения максимальная яркость была 255 а минимальная – 0

truey = mat2gray(imread("x_true.bmp"),[0,255]);

// Во всех примерах используется ФРТ размером 15 на 15 следующего вида

//h1=1/217[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 2 0 0 0 0 0 0 0
0 0 0 0 0 1 7 7 7 7 1 0 0 0 0
0 0 0 0 0 2 7 9 8 7 2 0 0 0 0
0 0 0 0 0 1 7 8 9 9 1 0 0 0 0
0 0 0 0 0 0 0 0 8 9 8 0 0 0 0
0 0 0 0 0 0 0 0 0 8 9 8 0 0 0
0 0 0 0 0 0 0 0 0 0 8 9 8 0 0
0 0 0 0 0 0 0 0 0 0 0 7 9 3 0
0 0 0 0 0 0 0 0 0 0 0 0 3 7 3
0 0 0 0 0 0 0 0 0 0 0 0 0 3 5
];*

//h2=1/136[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0*

```
000000131000000
000000242000000
000000353000000
000000464000000
000000575000000
000000686000000
000002797200000
000002777200000
000001222100000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
```

];

```
//h3=1/69*[000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
001111121000000
156778892000000
001111121000000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
000000000000000
```

];

// для выполнения задания надо считать и преобразовать свою ФРТ, сгенерированную заранее согласно варианту

h1=	Чтение ИХ 1-го - 3-го подканалов SIMO с диска и преобразование их в класс double с диапазоном значений от 0 до 1
h2=	
h3=	

//Поворот функций размытия точки

```
h1=h1($:-1:1,$:-1:1);  
h2=h2($:-1:1,$:-1:1);  
h3=h3($:-1:1,$:-1:1);
```

//Искажение

```
y(:, :, 1) =  
y(:, :, 2) =  
y(:, :, 3) =
```

Вычисление дискретной свертки исходного изображения с ИХ 1-го - 3-го подканалов SIMO системы (1 в 3).

//Зашумление

//Генерация положительного белого гауссова шума

```
for i=1:3
```

```
WGPN=rand(truey, "normal");
```

```
for a1_temp=1:size(truey,1)
```

```
for a2_temp=1:size(truey,2)
```

```
while WGPN(a1_temp,a2_temp)<0
```

```
WGPN(a1_temp,a2_temp)=rand(1);
```

```
end
```

```
if WGPN(a1_temp,a2_temp)>1
```

```
WGPN(a1_temp,a2_temp)=1;
```

```
end
```

```
end
```

```
end
```

//Зашумление

```
y(:, :, i)=y(:, :, i)+1/50*WGPN;
```

```
end
```

//Запись искаженных изображений

```
imwrite(y(:,:,1),"y1.bmp");  
imwrite(y(:,:,2),"y2.bmp");  
imwrite(y(:,:,3),"y3.bmp");
```

Результаты работы программы показаны на рисунке 2.

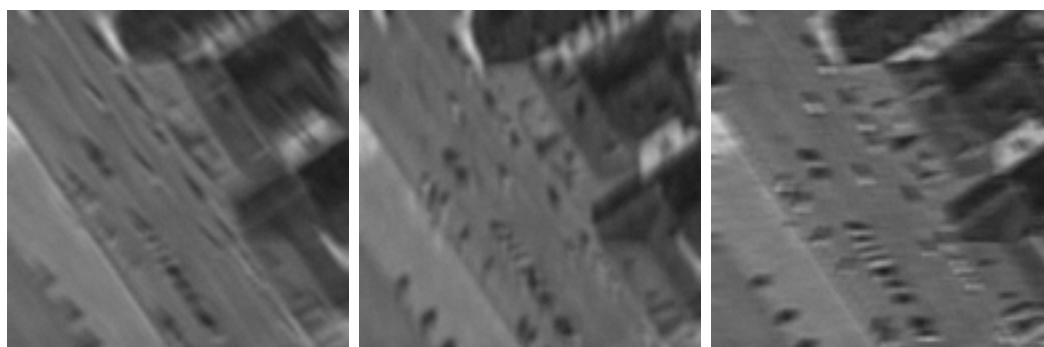


Рисунок 2 - Исходное изображение (сверху), выходы 3-х подканалов искажающей системы последовательно слева - направо.

ФРТ 3-х подканалов SIMO системы формирования изображений, заданные массивами h_1, h_2, h_3 должны быть введены в соответствии с Вашим вариантом. Исходное изображение $true_{uey}$ - полутонное изображение Вашего лица, в виде приемлемом для опознания преподавателем при защите курсовой работы. Размер изображения произволен и ограничивается только вычислительной мощностью используемого ПК, рекомендуется не менее 256×256 .

3.3. Коррекция линейных искажений в скалярном канале формирования изображений с известными характеристиками (фильтр Колмогорова-Винера)

При реализации алгоритма коррекции линейных искажений в скалярном канале формирования изображений с известными характеристиками Вам предлагается следующий программный код, часть которого скрыта, и должна быть восстановлена в процессе выполнения курсовой работы:

```
//Восстановление изображений по Винеру
clear
//Чтение изображений
y(:,:,1)=im2double(imread('y1.bmp'));
y(:,:,2)=im2double(imread('y2.bmp'));
y(:,:,3)=im2double(imread('y3.bmp'));
//Чтение ФРТ
h_true(:,:,1)=im2double(imread('h1_true.bmp'));
h_true(:,:,2)=im2double(imread('h2_true.bmp'));
h_true(:,:,3)=im2double(imread('h3_true.bmp'));
//Инициализация
L=size(h_true,3);
K=0.008
imsize=size(y,1);
Yestwnr=zeros(imsize,imsize,L);
yestwnr=zeros(imsize,imsize,L);
//Прямые Фурье преобразования
for i=1:size(h_true,3);
h_pad=zeros(imsize,imsize);
h_pad(58:72,58:72)=h_true(:, :, i);
H_true(:, :, i)=fftshift(fft2(h_pad));
Y(:, :, i)=fftshift(fft2(y(:, :, i)));
end
for i=1:L
//Фильтрация

Yestwnr(:, :, i)=


Алгоритм фильтрации Винера в частотной области


end
//Обратные БПФ
```

```

for i=1:size(h_true,3);

yestwnr(:,:,i)=fftshift(real(ifft(fftshift(Yestwnr(:,:,i))))));
end
//Запись изображений
imwrite(yestwnr(:,:,1),'yestwnr1.bmp')
imwrite(yestwnr(:,:,2),'yestwnr2.bmp')
imwrite(yestwnr(:,:,3),'yestwnr3.bmp')

```

Результаты работы программы показаны на рисунке 3.

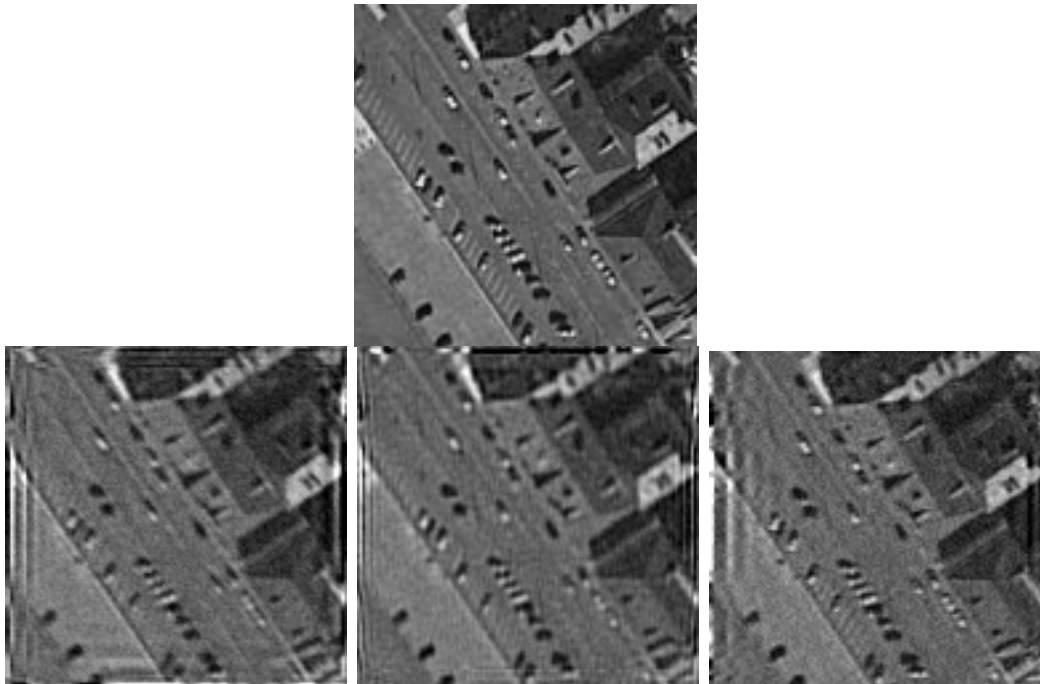


Рисунок 3 - Исходное изображение (вверху), выходы подканалов после коррекции последовательно слева - направо.

3.4. Слепая коррекция линейных искажений в векторном канале формирования изображений

При реализации алгоритма слепой идентификации линейного векторного канала формирования изображений в векторном канале формирования изображений с известными характеристиками Вам предлагается следующий программный код, часть которого скрыта, и должна быть восстановлена Вами в процессе выполнения курсовой работы:

```

//// Multiframe blind deconvolution
clear
tic
stacksize('max')
//// чтение искаженных входных изображений
y(:,:,1)=im2double(imread('y1.bmp'));
y(:,:,2)=im2double(imread('y2.bmp'));
y(:,:,3)=im2double(imread('y3.bmp'));
//// количество входных изображений
L=size(y,3);

//// 0. ПАРАМЕТРЫ АЛГОРИТМА
K=40; //количество итераций,
//200<рекомендуемое значение >40
AdditionalPSFIterations = 7; //количество итераций без проек-
ции, внутри каждой итерации ~7
alfa1=0.012; //Шаг для оценки NI
alfa2=0.012; //Шаг для оценки TI
initPSFsize=15; //размер инициализирующей ФРТ
lambda1=1; //множитель Лагранжа
lambda3=0.00001; //параметр программы- оценщика
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//предполагаем квадратное изображение в //1
imsize=size(y,1);
//предзаполнение массивов нулями
h=zeros(initPSFsize,initPSFsize,L,K);
H=zeros(imsize,imsize,L,K);
x_temp=zeros(imsize,imsize,L,K);
x=zeros(imsize,imsize,K);
h_est=zeros(initPSFsize,initPSFsize,L);
n=zeros(imsize,imsize,K);
H_est_tmp=zeros(imsize,imsize,L);
H2=zeros(imsize,imsize,L,AdditionalPSFIterations)
//// ИНИЦИАЛИЗАЦИЯ АЛГОРИТМА
k=1;
x(:,:,k)=sum(y,3)/L;
for j=1:L
// h(5:10,5:10,j,k)=1

```



```

// h(:,:,j,k)=h(:,:,j,k)./sum(h(:,:,j,k))
h(:,:,j,k)=fspecial('gaussian',initPSFsize,1);
end
    Y(:,:,1)=fft2(y(:,:,1))
    Y(:,:,2)=fft2(y(:,:,2))
    Y(:,:,3)=fft2(y(:,:,3))
    X(:,:,k)=fft2(x(:,:,k));
//Преобразуем ФРТ в ОПФ
for j=1:L
h_pad=zeros(imsz,imsz)
h_pad(58:72,58:72)=h(:,:,j,k) // чтобы ОПФ была размером с
изображение надо дополнить её нулями по краям, для ФРТ 15
на 15 и изображения 128 на 128 в данном примере размер окна
58:72,58:72, для выполнения задания здесь и ниже используйте
свой размер окна согласно заданной во варианте ФРТ и вы-
бранному вами размеру изображения.

    H(:,:,j,k)=fft2(h_pad)
End

//Значения дисперсий, выбраны произвольно, на сходимость ал-
горитма влияния не оказывают
sigma(1)=0.0035;//
sigma(2)=0.0020;//
sigma(3)=0.0033;//
///НАЧАЛО ИТЕРАЦИЙ
for k=2:K
    ///Вычисление множителя D
    yN = size(x(:,:,1),1);
    xN = size(x(:,:,1),2);
    for i=1:L,
        for j=1:L,
            d(i,j) =sum(sum(sigma(i)^2.*abs(H(:,:,i,k-1)).^2+...
                sigma(j)^2.*abs(H(:,:,j,k-1)).^2))/yN/xN;
        end;
        d(i,i) = sigma(i)^2;
    end;
    ///Оценщик по методу взаимных отношений (35)

```

```

for i=1:L
    H2(:,:,i,1)= H(:,:,i,k-1)
end
    for kk=2:AdditionalPSFIterations,
        for j=1:L,
            tmp1 = 0;
            tmp2 = 0;
            for i=1:L,
                if i<>j then
                    tmp1 = tmp1 + Элементы алгоритма слепой иден-
                    tmp2 = tmp2 + тификации канала (35)
                end;
            end;
        end;

        H_est_tmp(:,:,j)=
        (lambda1.*Y(:,:,j).*tmp1)./(lambda3+lambda1.*tmp2);
        end;
        for j=1:L,
            H2(:,:,j,kk) =
            H2(:,:,j,kk-1)*(1-alfa2)+alfa2*H_est_tmp(:,:,j);
        end;
    end;
    for i=1:L
        H(:,:,i,k)=H2(:,:,i,kk)
    end
// конец программы-оценщика
///// Проекция функции рассеяния точки
for j=1:L
    h_pad=real(iff(H(:,:,j,k)));
    h(:,:,j,k)=h_pad(58:72,58:72)
    h(:,:,j,k)=h(:,:,j,k).*double(h(:,:,j,k)>=0);
    h(:,:,j,k)=h(:,:,j,k)./sum(h(:,:,j,k)+%eps);
end
///// конец проекции
//Фурье преобразование для последующих итераций
for j=1:L
    h_pad=zeros(imsz,imsz)

```

```

h_pad(58:72,58:72)=h(:,j,k) // чтобы ОПФ была размером с
изображение
H(:,j,k)=fft2(h_pad)
end
end
/// КОНЕЦ ИТЕРАЦИЙ
/// Запись результатов
imwrite(h(:,1,k),'h1.bmp');
imwrite(h(:,2,k),'h2.bmp');
imwrite(h(:,3,k),'h3.bmp');

time=toc();
disp('Time');
disp(time);
///Вывод результатов на экран
disp('h1 true')
disp(imread('h1_true.bmp'))
disp('h1 est')
disp(imread('h1.bmp'))

```

Вывод результатов работы программы на экран

```

/// КОНЕЦ

```

Сравните заданные ИХ (ФРТ) и оцененные алгоритмом слепой идентификации. Воспользовавшись программой коррекции линейных искажений скалярного канала (раздел 3.3.) самостоятельно восстановите исходное изображение, используя полученные в данном разделе оценки каналов. Сравните с восстановленными изображениями из раздела 3.3. Сделайте выводы.

ЛИТЕРАТУРА

1. Льюнг Л. Идентификация систем. Теория для пользователя: Пер. с англ. / Под ред. Я.З. Цыпкина. - М.: Наука, 1991. - 432с.
2. Горячкин О.В. Методы слепой обработки сигналов и их приложения в системах радиотехники и связи. – М.: Радио и связь, 2003. – 230с.
3. Горячкин О.В. Лекции по статистической теории систем радиотехники и связи. – М.:Радиотехника, 2008. – 192 с.