

**ОПТИМИЗАЦИЯ КОЭФФИЦИЕНТА РАЗМЫТОСТИ ЯДРА  
В НЕПАРАМЕТРИЧЕСКОМ МОДЕЛИРОВАНИИ**

Е. Д. Михов

Сибирский федеральный университет  
Российская Федерация, 660041, г. Красноярск, просп. Свободный, 79  
E-mail: edmihov@mail.ru

*Исследуется проблема моделирования дискретно-непрерывных процессов в пространстве входных-выходных переменных. Моделирование данных процессов может осуществляться при помощи различных параметрических и непараметрических методов. Рассмотрено моделирование при помощи непараметрических методов. Такое решение было принято ввиду того, что непараметрическая теория, в отличие от параметрической теории, предполагает, что известны только качественные характеристики процесса. Зачастую, моделируемые объекты обладают неизвестной, сложной структурой. Учитывая эти факты, использование и развитие непараметрической теории продолжает быть актуальной задачей современности. Результаты статьи могут быть использованы для моделирования и управления оборудованием космических аппаратов.*

*При построении модели объекта при помощи ядерных оценок, важным параметром является коэффициент размытости ядра. Рассмотрены алгоритмы оптимизации коэффициента размытости ядра, а именно: метод перебора, метод деформируемого многогранника и генетический алгоритм. В качестве критерия оптимизации была выбрана среднеквадратичная ошибка модели исследуемого процесса, вычисленная при помощи скользящего экзамена. Стоит еще также сказать, что будут представлены результаты при оптимизации вектора параметров размытости ядра (для каждого входного воздействия) и при оптимизации общего коэффициента на все входные взаимодействия. Как выясняется, точность модели с одним оптимизированным параметром размытости ядра несколько уступает точности модели с оптимизированным вектором параметров размытости ядра, при этом вычисление коэффициента размытости ядра выполняется в разы быстрее и, как следствие, быстрее строится модель. Данные результаты могут быть крайне полезны при моделировании и управлении в условиях быстрого поступления информации и меняющейся обстановки.*

*Ключевые слова: непараметрическая модель, непараметрические алгоритмы, коэффициент размытости, оптимизация.*

Vestnik SibGAU  
Vol. 16, No. 2, P. 338–342**OPTIMIZATION OF COEFFICIENT OF BLURRING  
OF THE KERNEL IN NONPARAMETRIC MODELLING**

E. D. Mihov

Siberian Federal University  
79, Svobodny Av., Krasnoyarsk, 660041, Russian Federation  
E-mail: edmihov@mail.ru

*A modeling of discrete-continuous processes in space "input-output" variables. Modeling of these processes can be carried out using various parametric and nonparametric. This article deals with modeling using nonparametric methods. This decision was taken in view of the fact that non-parametric theory, in contrast to the parametric theory assumes that the only known qualitative characteristics of the process. The modeling objects often have an unknown and complex structure. Given these facts, the use and development of nonparametric theory continues to be an urgent task of our time. Our results can be used to equipment spacecraft modeling and them managing.*

*When building a model of the object by means of nuclear grade, an important parameter – the coefficient blur kernel. The algorithms optimize the ratio blur kernel, namely the method of enumeration, the flexible polyhedron method and genetic algorithm. As an optimization criterion was selected standard error of the test process models, calculated using the sliding test. It is worth also say that the results will be presented in the optimization parameter vector blur kernel (for each input action), and in the optimization of the overall coefficient on the interaction of all the input. As it turns out, the accuracy of the model to optimize the parameters of a blur kernel is slightly inferior to the accuracy of the model with optimized parameter vector blur kernel, and the calculation of the coefficient of blur kernel runs much faster*

and, as a consequence, the model will be built. These results can be extremely useful in modeling and managing the rapid flow of information and the changing environment.

*Keywords: nonparametric model, nonparametric algorithms, diffuseness coefficient, optimization.*

**Введение.** Идентификация многих стохастических объектов часто сводится к идентификации статических систем. Наиболее общая схема исследуемого дискретно-непрерывного процесса может быть представлена на нижеследующем рисунке[1–3]:

На рис. 1 приняты следующие обозначения:  $A$  – исследуемый объект (процесс);  $\overline{x}(t)$  – выходной вектор процесса;  $\overline{u}(t)$  – вектор управляющих воздействий;  $\overline{\mu}(t)$  – вектор входных неуправляемых, но измеряемых переменных процесса;  $\overline{\lambda}(t)$  – вектор входных неуправляемых и неизмеряемых переменных процесса;  $\xi(t)$  – случайное воздействие,  $\omega^i(t), i = 1, 2, \dots, k$  – переменные процесса, контролируемые по длине объекта;  $(t)$  – непрерывное время;  $H^\mu, H^u, H^x, H^z, H^q, H^\omega$  – каналы связи, соответствующие различным переменным, включающие в себя средства контроля, приборы для измерения наблюдаемых переменных;  $\mu_t, u_t, x_t, \omega_t$  – означают измерение  $\mu(t), u(t), x(t), \omega(t)$  в дискретное время;  $\overline{h^\mu}(t), \overline{h^u}(t), \overline{h^x}(t), \overline{h^\omega}(t)$  со значком вверху – случайные помехи измерений соответствующих переменных процесса.

**Идентификация в узком и широком смысле.** При моделировании разнообразных дискретно-непрерывных процессов в настоящее время доминирует теория идентификации в узком смысле [4; 5]. Ее содержание состоит в том, что на первом этапе, на основании имеющейся априорной информации, определяется параметрический класс оператора объекта  $A$ , например:

$$\tilde{x}_\alpha(t) = A^\alpha(u(t), \alpha), \tag{1}$$

где  $A^\alpha$  – параметрическая структура модели;  $\alpha$  – вектор параметров.

На втором этапе осуществляется оценка параметров  $\alpha$  на основе имеющейся выборки  $\{x_i, u_i, i = \overline{1, s}\}$ ,  $s$  – объем выборки. Успех решения задачи идентификации в этом случае существенно зависит от того, насколько «удачно» определен оператор (1).

Идентификация в широком смысле предполагает отсутствие этапа выбора параметрического класса оператора. Часто оказывается значительно проще определить класс операторов на основе сведений качественного характера, например, линейности процесса или типа нелинейности, однозначности либо неоднозначности и др. В этом случае задача идентификации состоит в оценивании этого оператора на основе выборки  $\{x_i, u_i, i = \overline{1, s}\}$  [6; 7]:

$$\tilde{x}_s(t) = A_s(u(t), \overline{x}_s, \overline{u}_s), \tag{2}$$

где  $\overline{x}_s = (x_1, x_2, \dots, x_s), \overline{u}_s = (u_1, u_2, \dots, u_s)$  – временные векторы. Оценка оператора  $A_s$  может быть осуществлена средствами непараметрической статистики. При-

мечательным здесь является то, что при этом исключается этап выбора параметрической структуры. Тем самым можно утверждать, что идентификация в этом случае, а это вариант идентификации в широком смысле, является более адекватной реальным задачам практики.

**Непараметрическая идентификация.** Непараметрическая идентификация представляется в виде моделирования при помощи ядерных оценок (3) [8; 9]:

$$x_{s+1}^* = \frac{\sum_{i=1}^s u_{(i)1} \prod_{j=1}^k \Phi\left(\frac{u_{n,s+1}^* - u_{n,i}}{c_s^x}\right)}{\sum_{i=1}^s \prod_{j=1}^k \Phi\left(\frac{u_{n,s+1}^* - u_{n,i}}{c_s^x}\right)}, \tag{3}$$

где  $\Phi^*$  – это ядерная «сглаживающая» функция (4), а  $c_s^x$  – коэффициент размытости ядра:

$$\Phi(o) = \begin{cases} 1 - |o|, & \text{если } |o| \leq 1, \\ 0, & \text{если } 1 < |o|. \end{cases} \tag{4}$$

Стоит сказать, что от выбранного коэффициента размытости напрямую зависит качество построенной модели. Данный коэффициент определяет степень участия элементов выборки в вычислении  $\hat{x}_{s+1}$  в точке  $u_m$  (рис. 2).

**Ход исследования.** Смысл исследования заключается в выборе наиболее точного и быстрого способа оптимизации, а также в выяснении вопроса о необходимости оптимизации коэффициента размытости для каждого входного воздействия. Все исследования проведены на машине с 4-ядерным процессором, с частотой ядер 2,8 ГГц. Программы написаны в среде Visual Studio 2010, на языке программирования C#.

Для начала оптимизируем вектор коэффициента размытости ядра при помощи метода деформируемых многогранников. Слабостью данного метода можно считать то, что при нахождении минимума он может «застрять» в локальном экстремуме. Для того чтобы определить, применим ли данный метод для оптимизации  $c_s$ , построим график зависимости среднеквадратичной ошибки ( $\sigma$ ) от  $c_s$  (рис. 3).

Как видно на рис. 3, данная зависимость плавная, и в ней нет большого количества локальных минимумов. В связи с этим можно проводить оптимизацию  $c_s$  при помощи метода деформируемого многогранника [10; 11].

Алгоритм данного метода следующий:

Параметрами метода являются:

- коэффициент отражения  $\alpha > 0$ , обычно выбирается равным 1;
- коэффициент сжатия  $\beta > 0$ , обычно выбирается равным 0,5;
- коэффициент растяжения  $\gamma > 0$ , обычно выбирается равным 2.

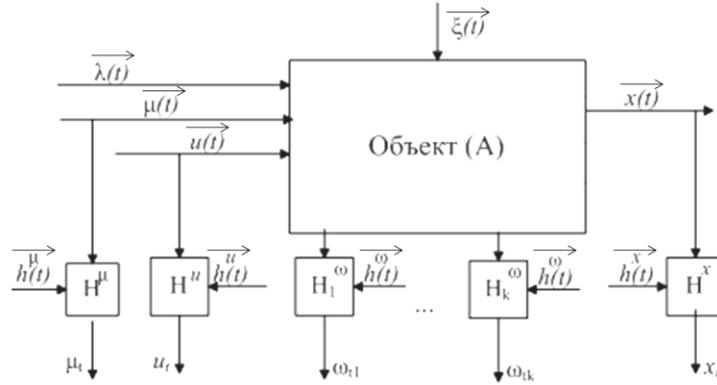


Рис. 1. Общая схема исследуемого процесса

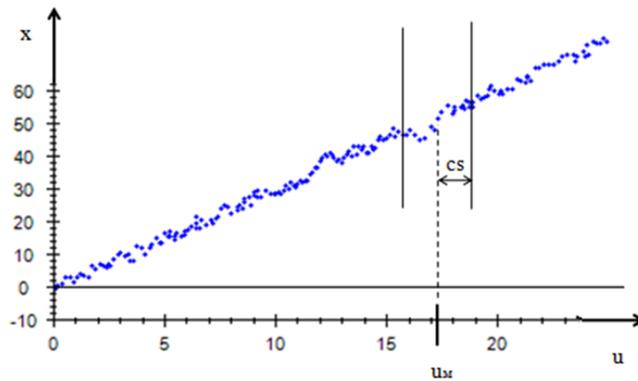


Рис. 2. Определение коэффициента размытости ядра

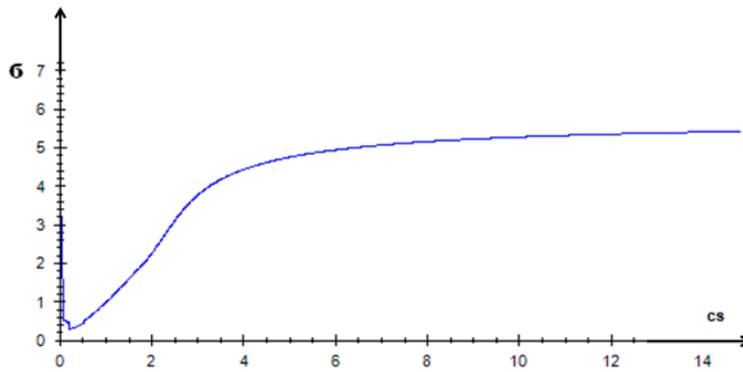


Рис. 3. Зависимость среднеквадратичной ошибки от коэффициента размытости ядра

1. «Подготовка». Вначале выбираются  $n + 1$  точки  $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})$ ,  $i = 1 \dots n + 1$ , образующие симплекс  $n$ -мерного пространства; в этих точках вычисляются значения функции:

$$f_1 = f(x_1), f_2 = f(x_2), \dots, f_{n+1} = f(x_{n+1}). \quad (5)$$

2. «Сортировка». Из вершин симплекса выбираем три точки:  $x_h$  с наибольшим (из выбранных) значением функции  $f_h$ ,  $x_g$  со следующим по величине значением  $f_g$  и  $x_l$  с наименьшим значением функции  $f_l$ . Целью дальнейших манипуляций будет уменьшение по крайней мере  $f_h$ .

3. Найдём центр тяжести всех точек, за исключением  $x_h$ :

$$x_c = \frac{1}{n} \sum_{i=1, i \neq h}^n x_i. \quad (6)$$

4. «Отражение». Отразим точку  $x_h$  относительно  $x_c$  с коэффициентом  $\alpha$ , получим точку  $x_r$  и вычислим в ней функцию  $f_r = f(x_r)$ . Координаты новой точки вычисляются по формуле

$$x_r = (1 + \alpha)x_c - \alpha x_h. \quad (7)$$

5. Далее смотрим, насколько нам удалось уменьшить функцию, ищем место  $f_r$  в ряду  $f_h, f_g, f_l$ .

Если  $f_r < f_l$ , то направление выбрано удачное и можно попробовать увеличить шаг. Производим

«растяжение». Новая точка  $x_e = (1 - \gamma)x_c + \gamma x_r$  и значение функции  $f_e = f(x_e)$ .

Если  $f_e < f_r$ , то можно расширить симплекс до этой точки: присваиваем точке  $x_h$  значение  $x_e$  и заканчиваем итерацию (шаг 9).

Если  $f_r < f_e$ , то переместились слишком далеко: присваиваем точке  $x_h$  значение  $x_r$  и заканчиваем итерацию (шаг 9).

Если  $f_l < f_r < f_g$ , то выбор точки неплохой (новая лучше двух прежних). Присваиваем точке  $x_h$  значение  $x_r$  и переходим на шаг 9.

Если  $f_g < f_r < f_h$ , то меняем местами значения  $x_r$  и  $x_h$ . Также нужно поменять местами значения  $f_r$  и  $f_h$ . После этого идём на шаг 6.

Если  $f_h < f_r$ , то просто идём на следующий шаг 6.

В результате (возможно, после переобозначения)  $f_l < f_g < f_h < f_r$ .

6. «Сжатие». Строим точку  $x_s = \beta x_h + (1 - \beta)x_c$  и вычисляем в ней значение  $f_s = f(x_s)$ .

7. Если  $f_s < f_h$ , то присваиваем точке  $x_h$  значение  $x_s$  и идём на шаг 9.

8. Если  $f_s > f_h$ , то первоначальные точки оказались самыми удачными. Делаем «глобальное сжатие» симплекса – гомотетию к точке с наименьшим значением  $x_l$ :

$$x_i \leftarrow x_l + \frac{(x_i - x_l)}{2}, \quad i \neq l. \quad (8)$$

9. Последний шаг – проверка сходимости. Может выполняться по-разному, например, оценкой дисперсии набора точек. Суть проверки заключается в том, чтобы проверить взаимную близость полученных вершин симплекса, что предполагает и близость их к искомому минимуму. Если требуемая точность ещё не достигнута, можно продолжить итерации с шага 2.

После этого вектор коэффициентов размытости ядра будет оптимизироваться при помощи генетического алгоритма [12–15]. Генетический алгоритм выглядит следующим образом:

1. Перед первым шагом нужно случайным образом создать начальную популяцию. Даже если она окажется совершенно неконкурентоспособной, вероятно, что генетический алгоритм всё равно достаточно быстро переведёт её в жизнеспособную популяцию. Таким образом, на первом шаге можно особенно не стараться сделать слишком уж приспособленных особей, достаточно, чтобы они соответствовали формату особей популяции и на них можно было подсчитать функцию приспособленности (Fitness). Итогом первого шага является популяция  $H$ , состоящая из  $N$  особей.

2. Размножение в генетических алгоритмах обычно половое – чтобы произвести потомка, нужны несколько родителей, обычно два.

Размножение в разных алгоритмах определяется по-разному – оно, конечно, зависит от представления данных. Главное требование к размножению, чтобы потомок или потомки имели возможность унаследовать черты обоих родителей, «смешав» их каким-либо способом.

Особь для размножения обычно выбираются из всей популяции  $H$ , а не из выживших на первом шаге элементов  $H_0$  из-за того, что проблема многих генетических алгоритмов – недостаток разнообразия (diversity) в особях. Достаточно быстро выделяется один-единственный генотип, который представляет собой локальный максимум, а затем все элементы популяции проигрывают ему отбор, и вся популяция «забывается» копиями этой особи. Есть разные способы борьбы с таким нежелательным эффектом. Один из них – выбор для размножения не самых приспособленных, а вообще всех особей.

3. К мутациям относится все, что и к размножению: есть некоторая доля мутантов  $m$ , являющаяся параметром генетического алгоритма, и на шаге мутаций нужно выбрать  $mN$  особей, а затем изменить их в соответствии с заранее определёнными операциями мутации.

На этапе отбора нужно из всей популяции выбрать определённую её долю, которая останется «в живых» на этом этапе эволюции. Есть разные способы проводить отбор. Вероятность выживания особи  $h$  должна зависеть от значения функции приспособленности  $Fitness(h)$ . Сама доля выживших  $s$  обычно является параметром генетического алгоритма и её просто задают заранее. По итогам отбора из  $N$  особей популяции  $H$  должны остаться  $sN$  особей, которые войдут в итоговую популяцию  $H'$ . Остальные особи погибают.

**Вычислительный эксперимент.** Моделируемый процесс имеет два входных воздействия и один выходной параметр. Обучающая выборка была взята в количестве 300. Помеха, воздействующая на объект, была равна 7 %. Критерием оптимизации была выбрана среднеквадратичная ошибка  $\sigma$ :

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (9)$$

Выведем результаты в виде таблицы.

Результаты оптимизации cs

Метод оптимизации	Оптимизируемый параметр	Время нахождения оптимального cs, миллисекунд	Среднеквадратичная ошибка $\sigma$
Метод деформируемого многогранника	Вектор cs	1118	0,750869
Метод деформируемого многогранника	Скаляр cs	100	0,755853
Перебор возможных значений	Вектор cs	26808	0,780678
Перебор возможных значений	Скаляр cs	934	0,78102
Генетический алгоритм	Вектор cs	39067	0,758035
Генетический алгоритм	Скаляр cs	37028	0,761118

Как мы можем видеть из таблицы, оптимизация вектора коэффициента размытости занимает во много раз больше времени, чем оптимизация скалярного значения, при этом модель практически не становится лучше.

Также стоит отметить, что оптимизация при помощи метода деформируемых многогранников действует гораздо быстрее, чем оптимизация при помощи стандартного перебора или генетического алгоритма.

**Заключение.** В статье приведены такие методы оптимизации, как метод деформируемых многогранников, генетический алгоритм и простой перебор возможных значений, и показаны результаты работы данных методов.

Также была доказана возможность использования методов локальной оптимизации для нахождения наилучшего коэффициента размытости ядра.

Было проведено сравнение между оптимизацией при помощи метода деформируемых многогранников, генетического алгоритма и оптимизацией при помощи стандартного перебора, где было доказано превосходство первого в задаче оптимизации коэффициента размытости ядра.

Доказано, что оптимизация вектора коэффициентов размытости является нецелесообразной вследствие больших затрат времени и малого изменения в точности модели исследуемого процесса.

#### Библиографические ссылки

1. Tweedle V., Smith R. A mathematical model of Bieber Fever // *Transworld Research Network*. 2012. Vol. 37/661, № 2. P. 157–177.
2. Lingefard T. Faces of mathematical modeling // *ZDM*. 2006. Vol. 38, № 2. P. 96–112.
3. Советов Б. Я., Яковлев С. А. Моделирование систем : учебник для вузов. М. : Высш. шк., 2001. 343 с.
4. Арнольд В. Теория катастроф. М. : Наука, 1990. 128 с.
5. Медведев А. В. Некоторые замечания к *H*-моделям безынерционных процессов с запаздыванием // *Вестник СибГАУ*. 2014. № 2(54). С. 24–34.
6. Медведев А. В. Анализ данных в задаче идентификации // *Компьютерный анализ данных моделирования*. Т. 2. Минск : БГУ, 1995. С. 201–206.
7. Медведев А. В. *H*-модели для безынерционных систем с запаздыванием // *Вестник СибГАУ*. 2012. № 5(45). С. 84–89.
8. Цыпкин Я. З. Адаптация и обучение в автоматических системах. М. : Наука, 1968. 400 с.
9. Фельдбаум А. А. Основы теории оптимальных автоматических систем. М. : Физматгиз, 1963. 552 с.
10. Рубан А. И. Методы анализа данных : учеб. пособие. 2-е изд., испр. и доп. Красноярск : ИПЦ КГТУ, 2004. 319 с.
11. Marco A., Rodolphe Le Riche. Globalized Nelder-Mead methods for engineering optimization // *ELSIVIER Science direct*. 2004. Vol. 1. P. 2–10.
12. Prayoth Kumsawat A Genetic Algorithm Optimization Technique for Multiwavelet – Based Digital Audio Watermarking // *EURASIP Journal on Advances in Signal Processing*. 2010. Vol. 1. P. 15–25.
13. Colin R. Reeves Genetic Algorithms for the Operations Researcher // *INFORMS Journal on Computing*. 1997. Vol. 9, no. 3. P. 231–250.
14. Jeffrey J. The application of genetic algorithm in GIS network analysis // *International Archives of Photogrammetry and Remote Sensing*. 2000. Vol. 33, part B 4. P. 1184–1191.
15. Raymond C., Ooi Koon B. A Comparison between Genetic Algorithms and Evolutionary Programming based on Cutting Stock Problem // *Engineering Letters*. 2007. P. 115.

#### References

1. Tweedle V., Smith R. A mathematical model of Bieber Fever. *Transworld Research Network*, 2012, Vol. 37/661, No. 2, P. 157–177.
2. Lingefard T. Faces of mathematical modeling. *ZDM*, 2006, Vol. 38, No. 2, P. 96–112.
3. Sovetov V., Yakovlev S. *Modelirovanie sistem* [Simulation systems]. Moscow, Vysshaya shkola Publ., 2001, P. 343.
4. Arnold V. *Teoriya katastrof* [Catastrophe Theory]. Moscow, Nauka Publ., 1990, 128 p.
5. Medvedev A. V. [Some notes on H-models for non-inertis systems with a delay] *Vestnik SibGAU*. 2014, No. 5 (54), P. 24–34 (In Russ.).
6. Medvedev A. V. [Analysis of the data in the problem identification]. *Komp'yuternyy analiz dannykh modelirovaniya* [Computer analysis of simulation data]. 1995. Vol. 2, P. 201–206.
7. Medvedev A. V. [H-model for non-inertia systems with delay]. *Vestnik SibGAU*. 2012, No. 5 (54), P. 84–89 (In Russ.).
8. Zipkin Ya. *Adaptatsiya i obuchenie v avtomaticheskikh sistemakh* [Adaptation and learning in automatic systems], Moscow, Nauka Publ., 1968, 400 p.
9. Feldbaum A. *Osnovy teorii optimal'nykh avtomaticheskikh sistem* [Fundamentals of the theory of optimal automatic systems]. Moscow, Fizmatgiz Publ., 1963, 552 p.
10. Ruban A. I. *Metody analiza dannykh* [Methods of Data Analysis]. Krasnoyarsk, CPI KSTU Publ., 2004, 319 p.
11. Marco A., Rodolphe Le Riche. Globalized Nelder-Mead methods for engineering optimization. *ELSIVIER Science direct*, 2004, Vol. 1, P. 2–10.
12. Prayoth Kumsawat. A Genetic Algorithm Optimization Technique for Multiwavelet - Based Digital Audio Watermarking. *EURASIP Journal on Advances in Signal Processing*, 2010, Vol. 1, P. 15–25.
13. Colin R. Reeves. Genetic Algorithms for the Operations Researcher. *INFORMS Journal on Computing*, 1997, Vol. 9, No. 3, P. 231–250.
14. Jeffrey J. The application of genetic algorithm in GIS network analysis. *International Archives of Photogrammetry and Remote Sensing*, 2000, Vol. 33, Part B4, P. 1184–1191.
15. Raymond C., Ooi Koon B. A Comparison between Genetic Algorithms and Evolutionary Programming based on Cutting Stock Problem. *Engineering Letters* 2007, 115 p.