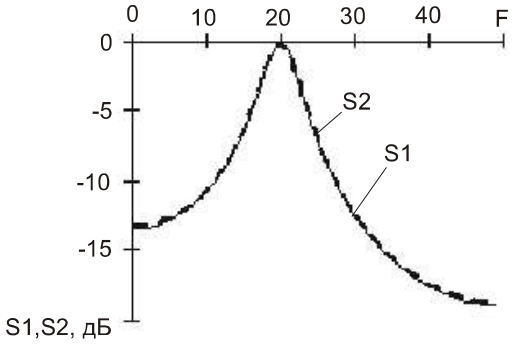

Рис. 1

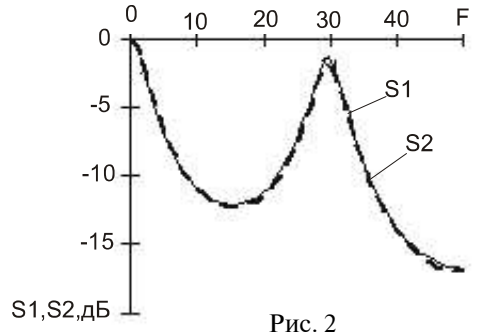

Рис. 2

$\Delta f_0 = 5$; $\Delta f_1 = 5$; $f_1 = 30$. Значения рассчитанных коэффициентов АР формирующих фильтров составляли:

$$\Phi_{3,1} = 0{,}326; \Phi_{3,2} = -0{,}280; \Phi_{3,3} = 0{,}624.$$

Анализ графиков показывает, что отклонения между задаваемой СПМ и его оценкой незначительны. Данные показатели являются приемлемыми для большинства инженерных приложений. Предложенный метод генерации имитационных коррелированных случайных процессов превосходит существующие аналоги [4].

Литература: 1. *Бокс Дж. , Дженкинс Г.* Анализ временных рядов: Пер.с англ. М.: Мир.1974. Вып.1. 406 с. 2. *Марпл.мл. С.Л.* Цифровой спектральный анализ и его приложения: Пер. с англ.М.: Мир, 1990. 584 с. 3. *Кармалита В.А.* Цифровая обработка случайных колебаний. М: Машиносроение,1986. 80 с. 4. *Быков В.В.* Цифровое моделирование в статистической радиотехнике. М: Сов. радио, 1971. 326 с.

Поступила в редколлегию 11.10.99

Рецензент: д-р техн. наук Кравченко Н.И.

Тихонов Вячеслав Анатольевич, канд. техн. наук, доцент кафедры РТС ХТУРЭ. Научные интересы: радиолокация, распознавание образов, статистические модели. Адрес: Украина, 61078, Харьков, ул. Каразина, 7/9, кв. 9, тел. 40-95-87, 47-03-91.

Русановский Дмитрий Евгеньевич, студент гр. АРТ-95-1 ХТУРЭ. Научные интересы: цифровая обработка речи – кодирование, сжатие речи, статистическое моделирование. Хобби: программирование, Web-дизайн. Адрес: Украина, 61118, г.Южный, ул. Освобождения-2.

e-mail: d_rusanovsky@aport.ru

Тихонов Дмитрий Вячеславович, студент гр. ИСПР-98-1 ХТУРЭ. Научные интересы: искусственный интеллект, распознавание речи, программирование. Адрес: Украина, 61078, Харьков, ул. Каразина 7/9, кв. 9, тел. 47-03-91.

УДК 519.713

ЭФФЕКТИВНАЯ РЕАЛИЗАЦИЯ ОПЕРАЦИИ ВОЗВЕДЕНИЯ В СТЕПЕНЬ БОЛЬШИХ ЦЕЛЫХ ЧИСЕЛ В КРИПТОГРАФИЧЕСКИХ СИСТЕМАХ МЕТОДОМ ПРЕДВЫЧИСЛЕНИЙ ПО ФИКСИРОВАННОМУ ОСНОВАНИЮ

ГОРБЕНКО И.Д., ЛАВРИНЕНКО Д.И.

Проводится сравнительный анализ алгоритмов для операции возведения в степень по модулю целых чисел большой разрядности. Описывается алгоритм с использованием набора предвычисленных значений по фиксированному основанию. Предлагается параллельная модификация такого алгоритма. Приводятся результаты применения данных алгоритмов для наиболее распространённых схем и алгоритмов в криптографии.

1. Введение

В ряде несимметричных криптографических алгоритмов [1], а также в алгоритмах и средствах криптоанализа такого класса [2] арифметические операции выполняются над числами x, разрядность которых L_x значительно превышает разрядную сетку L_p современных вычислительных средств. Такие числа известны как числа многократной точности. Основные операции, которые выполняются над ними, это сложение, вычитание, умножение, деление и возведение в степень по модулю N. Проведенные исследования показали, что наиболее часто в криптопреобразованиях используется операция возведения в степень по модулю. Вычислительная сложность данной операции носит полиномиальный характер, причём порядок и коэффициенты полиномов, описывающих вычислительную сложность, зависят от применяемых математических методов и алгоритмов умножения, деления и возведения в степень по модулю. Достаточно подробно методы и алгоритмы возведения в степень по модулю рассмотрены в [3].

Применение при выполнении операции умножения по модулю преобразования Монтгомери или преобразования Барретта [3], а также использование при возведении в степень по модулю блочного метода [4] позволили уменьшить вычислительную сложность этих операций. Такое уменьшение эквивалентно повышению скорости преобразований. Несмотря на определённое уменьшение вычислительной сложности операции возведения в степень, её величина остаётся всё ещё значительной, а выполнение требований по увеличению скорости криптографических преобразований – проблематичным. В связи с этим дальнейшее уменьшение вычислительной сложности операции возведения в степень по модулю чисел большой разрядности остаётся актуальной задачей.

Анализ классических методов уменьшения вычислительной сложности показывает, что они базируются на уменьшении временной сложности. В то же время вычислительная сложность может быть уменьшена за счёт увеличения пространственной сложности, требующей, например, увеличения объема памяти.

В данной статье рассматриваются методы возведения в степень больших целых чисел, базирующиеся на выполнении предвычислений. Для каждого метода рассчитываются временная и пространственная вычислительная сложность, т.е. затрачиваемые время и память. Даётся оценка выигрыша для широко применяемых цифровых подписей класса Эль-Гамаля [5] и криптографического протокола (схемы) Диффи-Хеллмана [6].

2. Анализ основных методов

В общей постановке задачу формируем следующим образом. Пусть g – элемент группы Z/qZ, где q – большое целое. Необходимо вычислить $g^n(\mathrm{mod}\,N)$, где n – также большое и равномерно распределено на {0,...,N}, с минимальной вычислительной сложностью, применяя для этого методы пространственно-временной минимизации вычислительной сложности операции возведения в степень по модулю.

Рассмотрим и проанализируем основные методы выполнения таких вычислений.

Применяя схему квадрата и умножения, использующую двоичное представление n, $g^n(\mathrm{mod}\,N)$ можно вычислить в среднем за $3/2\cdot\lceil\log_2 n\rceil$ умножений и в худшем случае – за $2\cdot\lceil\log_2 n\rceil-2$ умножений [3].

Блочный метод, или метод скользящего окна, обеспечивает более эффективное выполнение операции возведения в степень за счет сравнительно малого увеличения пространственной вычислительной сложности. Сущность его заключается в выполнении для каждого вектора предвычислений, последующего запоминания и использования предвычисленных значений. Показано [4], что блочный метод возводит в степень в среднем за

$$\lceil\log_2 n\rceil-1+(2^k-1)+\lceil\log_2 n/k\rceil$$

умножений. Здесь k – длина блока (окна) в битах. Минимум выражения достигается для L_x=512 битов при k=5, для L_x=1024 битов при k=6. Порядок расчёта оптимального значения k изложен в [4].

Рассмотренные выше методы являются универсальными в том смысле, что не предъявляют требований к своим входным данным. Показано [3, 7], что наложение определённых ограничений снижает количество степеней свободы и, как следствие, уменьшает вычислительную сложность операции возведения в степень по модулю. В качестве такого ограничения примем требование фиксированного основания g.

Относительно простым методом такого уменьшения вычислительной сложности является вычисление g^n посредством представления n в виде 2^i, $i=\overline{1,\lceil\log_2 n\rceil-1}$ [3]. Вычислив члены ряда

$$\left\{g^{2^i}\,\middle|\,i=1,\ldots,\lceil\log_2 n\rceil-1\right\}$$

и перемножив те его элементы, разряды которых в двоичном представлении n равны 1, получим значение g^n. Данный метод позволяет вычислить $g^n(\mathrm{mod}\,N)$ за $v(n)$-1 умножений, где $v(n)$ – число единиц в двоичном представлении n, используя память $\lceil\log_2 n\rceil$ значений [3].

Дальнейшим обобщением этого метода является представление n по произвольному основанию x. В этом случае необходимо предварительно вычислять и сохранять значения $g^{x^0},\ldots,g^{x^{m-1}}$, где x – основание системы счисления представления n, а m равно $\lceil\log_x n\rceil$. Если найти разложение

$$n=\sum_{i=1}^{m-1}a_i x^i,$$

где $0\le a_i\le x$-1 для $0\le i<m$, то можно также определить

$$g^n=\prod_{d=1}^{h}c_d^d,\ \text{где}\ c_d=\prod_{a_i=d}g^{x^i}. \qquad (1)$$

Анализ этого алгоритма показывает, что вычисление $g^n(\mathrm{mod}\,N)$ можно выполнить за $m+O(h\cdot\log h)$ умножений. При этом за счёт пространственной сложности временную сложность метода можно уменьшить. Одним из алгоритмов решения этой задачи является следующий [7]:

$b = 1$;

$r = 1$;

for ($d = h$; d i 1; d—)

{

  for ($i|a_i = d$)

    $b = b\cdot g^{x^i}$;

  $r = r\cdot b$;

}

return ***r***.

После выполнения цикла i раз получим $b=c_h\cdot c_{h-1}\cdot\ldots\cdot c_{h-i+1}$ и $r=c_h^i\cdot c_{h-1}^{i-1}\cdot\ldots\cdot c_{h-i+1}$. После выполнения цикла h раз вычислим $r=\prod_{d=1}^{h}c_d^d$ [7].

Определим временную вычислительную сложность алгоритма (для случая n≠0). Так как в показателе степени m цифр по основанию x, то вычисление $b=b\cdot g^{x^i}$ выполняется самое большое m раз. Учитывая то, что эта строка выполняется только для $a_i\neq0$, получим среднее количество умножений в этой строке как $\frac{h}{x}\cdot m$ или $\frac{x-1}{x}\cdot\lceil\log_x n\rceil$. Строка $r=r\cdot b$ выполняется точно h или $x-1$ раз. Таким образом, суммарная временная вычислительная сложность этого алгоритма составит

$$I_{fge}=x-1+\frac{x-1}{x}\cdot\lceil\log_x n\rceil \qquad (2)$$

модульных умножений. Оценка пространственной вычислительной сложности этого алгоритма определяется легко – она составляет m или $\lceil\log_x n\rceil$ значений.

В операции возведения в степень по модулю используются модульные вычисления. Наименьшей вычислительной сложностью, как правило, обладают методы Монтгомери и Барретта [3]. Выбор метода модульного приведения определяется проектировщиком криптосистемы, поскольку на разных ЭВМ эти приведения могут иметь различную вычислительную сложность. Следует лишь заметить, что для варианта с применением метода Барретта первые умножения в строках $b = b \cdot g^{x^i}$ и $r = r \cdot b$ можно заменить на операции присваивания, поскольку первоначальные значения r и b равны 1.

Операнды криптографических преобразований имеют, как правило, фиксированную длину 512 или 1024 бит [3]. Из выражения (2) следует, что вычислительная сложность зависит от основания x. Определим наиболее выгодное основание x для представления n. Для этого найдём минимум функции

$$f(x) = x - 1 + \frac{x-1}{x} \cdot \lceil \log_x n \rceil$$

для L_n=512 бит и L_n=1024 бит, где $x \in N$. Результаты исследований представлены на рис. 1 и 2. Как следует из рис. 1, для L_n=512 функция достигает минимума при x=26, а для L_n=1024 – при x=42. Приведенные кривые позволяют оценить проигрыш, если x выбирается без учёта требований минимизации вычислительной сложности.

Из приведенного выше вытекает необходимость решения задачи получения представления n по базе x. Для большого и равномерно распределенного на $\{0,...,N\}$ n и произвольного x можно использовать операции деления "длинного" числа на "короткое". Для вычисления представления n по базе x потребуется не менее m операций [3]. Учитывая то, что операция деления имеет большую вычислительную сложность, чем операция модульного умножения [3], нахождение представления n по базе x будет вносить заметную долю в общую временную вычислительную сложность алгоритма. Но если $x=2^i$, то для определения представления n можно воспользоваться операциями сдвига, имеющими заметно меньшую вычислительную сложность. При L_n=512 бит, для которого оптимальным является x=26, значение x близко к $32=2^5$. Поэтому можно принять x=32. Проведенный вычислительный эксперимент (табл. 1) подтвердил обоснованность такого подхода (здесь и далее по тексту для тестирования использовался ПК с процессором Pentium-MMX 200MHz). При x=32 выигрыш по сравнению с x=26 достигается из-за применения операций сдвига, хотя x=32 и не является оптимальным значением.

Таблица 1

Основание представления n	Время выполнения операции, с	Количество предвычисленных значений
x = 26	0,0069	109
x = 32	0,0067	103

Результаты вычислительного эксперимента для L_n=1024 представлены в табл. 2. Минимальное время выполнения операции достигается также при x=32. При x=64 выигрыша по сравнению с x=42 почти нет.

Таблица 2

Основание представления n	Время выполнения операции, с	Количество предвычисленных значений
x = 42	0,0446	190
x = 32	0,0429	205
x = 64	0,0444	171

Анализ результатов эксперимента показывает, что варианты с $x=2^i$ имеют меньшую временную вычислительную сложность. Но окончательный выбор варианта можно предоставить проектировщику криптоалгоритма. Это объясняется тем, что для варианта x=32 достигается минимальная временная вычислительная сложность, а для x=64 – пространственная вычислительная сложность.

3. Сравнение метода предвычислений с известными методами

Оценим эффективность применения этого алгоритма по сравнению с традиционными методами [3]. Для этого выберем наиболее эффективный метод блочного возведения в степень. Оба алгоритма будут выполнять модульное возведение в степень с использованием арифметики Монтгомери. Результаты экспериментальных исследований приведены в табл. 3.

Из табл. 3 следует, что метод с предвычислениями при L_n=512 даёт выигрыш в ≈3,8 раза, а при L_n=1024 – в ≈4,3 раза.

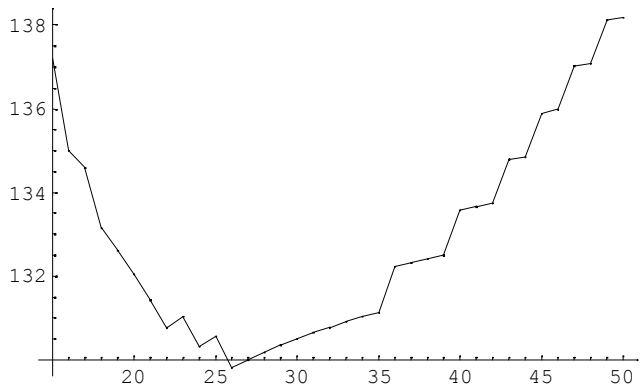

Рис.1. Временная вычислительная сложность операции модульного возведения в степень в зависимости от основания представления экспоненты для L_n=512 бит

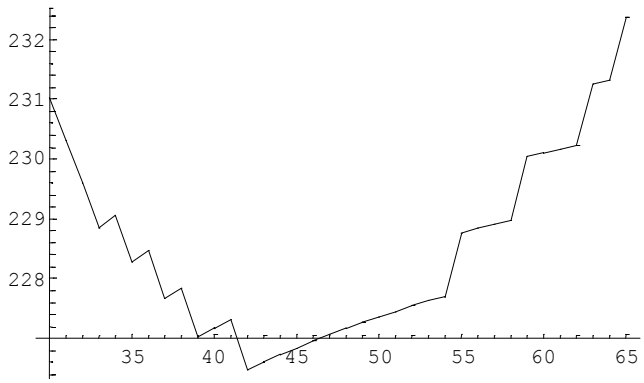

Рис.2. Временная вычислительная сложность операции модульного возведения в степень в зависимости от основания представления экспоненты для L_n=1024 бит

Таблица 3

	Время выполнения операции, с	
	512 бит	1024 бит
Блочный метод по Монтгомери	0,0262	0,192
Метод с предвычислениями	0,0069	0,0446

В проверке цифровой подписи Эль-Гамаля [5] используется следующая схема вычислений:

$$g_1^{n_1} \cdot g_2^{n_2} \pmod M. \qquad (3)$$

Она вычисляется за три операции: два модульных возведения в степень и модульное умножение. Эти операции можно совместить путем параллельного выполнения возведения в степень, используя следующий алгоритм:

$b = 1;$

$r = 1;$

for $(d = h;\ d \ge 1;\ d—)$

{

for $(i | a_{1i} = d)$

$b = b \cdot g_1^{x^i}$;

for **(** $i | a_{2i} = d$ **)**

$b = b \cdot g_2^{x^i}$;

$r = r \cdot b$

}

return ***r.***

Здесь

$$n_1 = \sum_{i=1}^{m-1} a_{1i} x^i, \quad n_2 = \sum_{i=1}^{m-1} a_{2i} x^i .$$

В данном алгоритме, в отличие от приведенного выше алгоритма возведения в степень по фиксированному основанию, для каждого основания g накопление компонент c_i (1) происходит в одной переменной b, что позволяет использовать в цикле только одну строку $r = r \cdot b$.

Пространственная вычислительная сложность для приведенного алгоритма будет вдвое больше по сравнению с нераспараллеленным и составит

$$\lceil \log_x n_1 \rceil + \lceil \log_x n_2 \rceil$$

значений. Определим число умножений, выполняемых в алгоритме (для случая $n_1 \ne 0$ и $n_2 \ne 0$). Получим, что суммарное количество умножений в нем равно

$$x - 1 + 2 \cdot \frac{x-1}{x} \cdot \lceil \log_x n \rceil, \qquad (4)$$

где по сравнению с (2) в (4) удвоение второго члена сделано в связи с двумя возведениями в степень.

Применяя данную схему вместо классической для вычисления (3), получаем выигрыш в x операций модульного умножения. Это соответствует выигрышу в ≈20% для L_n=512 бит и ≈18,5% для L_n=1024 бит. Пространственная вычислительная сложность для обоих вариантов алгоритма проверки цифровой подписи эквивалентна и равна

$$\lceil \log_x n_1 \rceil + \lceil \log_x n_2 \rceil$$

значений. Результаты вычислительного эксперимента по проверке цифровой подписи представлены в табл. 4.

Таблица 4

	Время выполнения операции, с	
	512 бит	1024 бит
"Классическая" схема	0,010	0,033
Параллельный алгоритм	0,008	0,026

Здесь оба варианта используют метод предвычислений по фиксированному основанию.

4. Заключение

Полученные результаты позволяют утверждать, что применение метода с предвычислениями по фиксированному основанию способно существенно (в 3-4 раза) уменьшить временную вычислительную сложность выработки и проверки цифровой подписи по Эль-Гамалю, а также выработки сеансовых ключей по схеме Диффи-Хеллмана. Однако данные методы не подходят для алгоритмов и схем, базирующихся на шифровании по RSA из-за невозможности наложить требование фиксированного основания при выполнении возведения в степень. Одним из решений данной проблемы является применение так называемых аддитивных цепочек [8]. Но и здесь, чтобы добиться эффективной реализации, существует множество задач, подлежащих решению.

Литература: 1.*Диффи У.* Первые десять лет криптографии с открытым ключом // ТИИЭР. 1998. Т.76, №5. С.54-74. 2. *Lenstra A.K., Manasse M.S.* Factoring with two large primes // Advances in cryptology – Eurocrypt'90. Berlin,1991. P. 72-82. 3.*Menezes A., Oorschot P., Vanstone S.* Handbook of Applied Cryptography. CRC Press, 1996. 816p. 4. *Горбенко И.Д., Качко Е.Г., Свинарев А.В.* Оптимизация алгоритмов арифметики многократной точности // Тр. 2 межд. конф. "Теория и техника передачи, приема и обработки информации" – Туапсе, 1996. 5. *ElGamal T.* A public-key cryptosystem and a signature scheme based on discrete logarithms // IEEE trans. on Information Theory. 1985. Vol.IT-31. №4. P. 469-472. 6.*Diffie W., Hellman M.E.* New directions in cryptography // IEEE trans. on Information Theory. 1976. Vol.IT-22. №6. P. 644-654. 7.*Ernest F. Brickell, Daniel M. Gordon, Kevin S. McCurley and David B. Wilson.* Fast Exponentiation with Precomputation (Extended Abstract) 8.*J. Bos and M. Coster.* Addition Chain Heuristics, in Advances in Cryptology, 1990.

Поступила в редколлегию 02.10.99

Рецензент: д-р техн. наук Долгов В.И.

Горбенко Иван Дмитриевич, д-р техн. наук, профессор, проректор по научной работе ХТУРЭ. Научные интересы: защита информации в компьютерных системах и сетях. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 30-24-50, 37-56-39.

Лавриненко Дмитрий Иванович, аспирант кафедры ЭВМ ХТУРЭ. Научные интересы: криптографические методы защиты информации в компьютерных системах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел.40-94-25.

УДК 681.513

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СТОХАСТИЧЕСКИМИ НЕСТАЦИОНАРНЫМИ ОБЪЕКТАМИ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ С АКТИВНЫМ НАКОПЛЕНИЕМ ИНФОРМАЦИИ

I. ДОСТОВЕРНО-ЭКВИВАЛЕНТНЫЙ ПОДХОД

АДОНИН О.В., БОДЯНСКИЙ Е.В., КОТЛЯРЕВСКИЙ С.В.

Рассматривается задача синтеза адаптивного регулятора для динамического стохастического нестационарного объекта, функционирующего в условиях априорной неопределенности. В основу синтеза положен принцип достоверной эквивалентности и методика локальной оптимизации. Введение в контур идентификации настраиваемого упредителя позволило получить более простую структуру закона управления по сравнению с известным регулятором Кларка-Гофтропа.

1. Постановка задачи

Пусть управляемый динамический объект описывается системой уравнений в пространстве состояний

$$\begin{cases} x(t+1) = A(\theta)x(t) + B(\theta)u(t) + w(t), \\ y(t) = C(\theta)x(t) + v(t), \end{cases} \qquad (1)$$

где $x(t) - (n \times 1)$ – вектор состояний объекта в дискретный момент времени t ($t = 0, 1, 2, ...$); $u(t)$ – скалярное управляющее воздействие; $y(t) - (p \times 1)$ – вектор наблюдаемых сигналов объекта; $A(\theta)$, $B(\theta)$, $C(\theta) - (n \times n), (n \times 1), (p \times n)$ - матрицы коэффициентов объекта, зависящие в общем случае от неизвестного и нестационарного вектора параметров θ; $w(t)$, $v(t) - (n \times 1), (p \times 1)$ – векторы случайных возмущений (помех) такие, что $M\{v(t)|F_t\} = 0$,

$$M\{w(t)w^T(t)|F_t\} = P_w(\theta) > 0,$$

$$M\{v(t)v^T(t)|F_t\} = P_v(\theta) > 0,$$

$$M\{w(t)v^T(t)|F_t\} = P_{wv}(\theta) > 0,$$

F_t – σ-алгебра согласования с процессом $\{x(0), x(1), ..., x(t), u(0), ..., u(t)\}$; вектор θ в общем случае имеет $n^2 + n + np + \frac{1}{2}n(n+1) + \frac{1}{2}p(p+1) + np = \frac{3}{2}n^2 + \frac{1}{2}p^2 + \frac{3}{2}n + \frac{1}{2}p + 2np$ неизвестных параметров.

Цель управления задается одношаговым критерием вида

$$I_t^{ce} = M\left\{\|x(t+1)\|_Q^2 + ru^2(t)|F_t\right\}, \qquad (2)$$

где $Q = \mathrm{diag}(q_1, q_2, ..., q_n)$ – положительно определенная весовая матрица управления; r – неотрицательный весовой множитель.

Задача состоит в нахождении на каждом такте управления t управляющего сигнала $u(t)$, доставляющего минимум критерию оптимизации (2).

2. Преобразование структуры модели

Традиционный подход к синтезу управляющего устройства базируется на принципе стохастической эквивалентности. При этом задача решается в два этапа: оценивание состояния с помощью фильтра Калмана:

$$\begin{cases} \hat{x}(t+1,\theta) = A(\theta)\hat{x}(t,\theta) + B(\theta)u(t) + K(t,\theta) * \\ * (y(t) - C(\theta)\hat{x}(t,\theta)), \\ \hat{y}(t,\theta) = C(\theta)\hat{x}(t,\theta), \\ K(t,\theta) = (A(\theta)P_x(t,\theta)C^T(\theta) + P_{wv}(\theta)) * \\ * (C(\theta)P_x(t,\theta)C^T(\theta) + P_v(\theta))^{-1}, \\ P_x(t+1,\theta) = A(\theta)P_x(t,\theta)A^T(\theta) + P_w(\theta) - \\ - K(t,\theta)(C(\theta)P_x(t,\theta)C^T(\theta) + P_v(\theta))K^T(t,\theta) = \\ = A(\theta)P_x(t,\theta)A^T(\theta) + P_w(\theta) - (A(\theta)P(t,\theta) * \\ * C^T(\theta) + P_{wv}(\theta))(C(\theta)P_x(t,\theta)C^T(\theta) + \\ + P_v(\theta))^{-1}(A(\theta)P_x(t,\theta)C^T(\theta) + P_{wv}(\theta))^T \end{cases} \qquad (3)$$

и оптимизации критерия

$$I_t^{ce} = M\left\{\|\hat{x}(t+1)\|_Q^2 + ru^2(t)|F_t\right\} \qquad (4)$$

вместо целевой функции (2).

Использование такого подхода возможно только в случае, когда вектор параметров θ априорно известен. Если же это не так, то наряду с оцениванием состояний необходимо идентифицировать в реальном времени вектор параметров θ, что значительно усложняет задачу.

Введем в рассмотрение обновляющую последовательность

$$e(t) = y(t) - C(\theta)\hat{x}(t,\theta) = C(\theta)(x(t) - \hat{x}(t,\theta)) + v(t) \qquad (5)$$

и запишем соотношения (3) в обновляющей:

$$\begin{cases} \hat{x}(t+1,\theta) = A(\theta)\hat{x}(t,\theta) + B(\theta)u(t) + K(t,\theta)e(t), \\ y(t) = c(\theta)\hat{x}(t,\theta) + e(t), \\ M\{e(t)e^T(t)|F_t\} = P_e(t,\theta) = C(\theta)P_X(t,\theta)c^T(\theta) + \\ + P_V(\theta) \end{cases} \qquad (6)$$

и прогнозирующей формах: