

# МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ПЕРЕМЕННЫМ ПОТОКОМ ДАННЫХ

В.Г. Промыслов

Рассмотрены вопросы моделирования и расчета вычислительных распределенных систем. Для моделирования применен метод «Network calculus», базирующийся на мини-(макси-) плюс алгебре. Основное внимание уделено вычислительным системам с неравным на длительном интервале наблюдения объемом принимаемых и передаваемых компонентом данных. Предложено применять обобщенное преобразование Лежандра для расчета конверта выходного потока системы. Приведены примеры моделирования вычислительной системы с суммированием двух потоков, где основной поток имеет вид T-SPEC и моделирования фрагмента вычислительной системы с линейным преобразованием потока и пакетированием.

**Ключевые слова:** распределенная вычислительная система, метод «network calculus», преобразование Лежандра.

## ВВЕДЕНИЕ

Современная система автоматизации технологических процессов в большинстве случаев реализуется как распределенная по функциям и средствам вычислительная система и представляет собой совокупность аппаратных и программных средств, обеспечивающих сбор, накопление, асинхронную обработку, представление и передачу информации. Ее компоненты могут быть распределены пространственно и (или) функционально. Моделирование таких систем и расчет их характеристик представляет несомненный интерес для пользователей и разработчиков. В последнее время в качестве средства расчета характеристик потоков данных, циркулирующих между компонентами компьютерной сети, применяется метод анализа детерминированных систем «Network calculus» [1]. Он базируется на мини-плюс алгебре и привлекателен тем, что во многих случаях позволяет рассматривать в качестве линейных системы, являющиеся нелинейными в обычной алгебре. Метод успешно применяется для расчета параметров как больших (WAN), так и малых (LAN) компьютерных сетей, базирующихся на протоколе TCP/IP [2–4].

Однако методу свойственны и недостатки, особенно очевидные при переходе от моделирования компьютерных сетей передачи данных к моделированию вычислительных систем, реализующих системы автоматизации технологических процессов. Классический подход метода «Network calculus»

разработан для ациклических систем без потерь и существенного изменения объема данных после их обработки в компоненте системы [5].

Ограничение в виде требований ациклическости и равенства на длительном интервале наблюдения объема принимаемых и передаваемых компонентом данных, видимо, несущественно для компьютерных систем передачи информации, где обычно логически присутствует единственный ациклический канал обработки, а алгоритмы обработки состоят в преобразовании формата и частично фильтрации данных, но затрудняет применение «Network calculus» для моделирования реальных распределенных вычислительных систем реализующих системы управления, со сложными алгоритмами обработки и циклическими связями.

В целях применения аппарата «Network calculus» в распределенных вычислительных системах такого типа в работах [6, 7] предложен подход, позволяющий использовать аппарат «Network calculus» для моделирования вычислительных систем с циклической зависимостью данных и линейным изменением объема данных после их обработки в компоненте системы. Однако, в последнем случае, на наш взгляд, простая линейная зависимость не отражает всего многообразия зависимостей между входным и выходным потоками данных, встречающихся в реальных системах. Нами предлагается моделирование зависимости входного и выходного потоков данных на длительном интервале наблюдения не только в виде линейного оператора, но и в виде комбинации основного потока и не-



которого вспомогательного потока на входе компонента. Для расчета конверта выходного (суммарного) потока после обработки потока в компоненте рассмотрено применение обобщенного преобразования Лежандра.

Схожие примеры применения преобразования Лежандра для моделирования вычислительных систем методом «Network calculus» можно найти в работах [8–10]. Работы в основном, концентрируются на описании свойств преобразования Лежандра и вычислительных аспектах расчета таких параметров системы, как максимальная задержка прохождения сигнала и максимальный размер буфера в системе, и не уделяют должного внимания расчету конверта выходного потока. Знание потока на выходе компонента нужно при условии разбиения системы на части для их отдельного анализа, с последующим суммированием полученных по отдельным частям результатов, для расчета характеристик всей системы [4]. Это необходимо как раз в случае, когда нарушаются предположения о равенстве входных и выходных потоков для компонента на длительных интервалах наблюдения. В работах [8–10] используются несколько различные виды записи преобразования Лежандра, соответственно имеются различия в свойствах преобразования. В настоящей работе представлена форма преобразований Лежандра, идентичная представленной в работе [10] и упрощающая операцию «обращения свертки».

### 1. «NETWORK CALCULUS»: ОСНОВНЫЕ ПОЛОЖЕНИЯ

«Network calculus» — относительно новый метод, применяемый для анализа детерминированных систем с очередью, базирующийся на понятиях мини-плюс алгебры. Основы метода заложены в работах [5, 11], которые, в свою очередь, базировались на более ранних работах [12, 13]. Далее изложение материала параграфа следует в основном работам [1, 14].

Введем некоторые полезные функции, дадим основные определения и докажем некоторые важные утверждения для «Network calculus». Обозначим функции:

$$\gamma_{r, b}(t) := \begin{cases} rt + b, & t > 0, \\ 0, & t \leq 0, \end{cases}$$

$$\beta_{R, T}(t) := \begin{cases} R(t - T), & t > T, \\ 0, & t \leq T, \end{cases}$$

$$TSPEC_{p, M, q, L} = \min(\gamma_{p, M}, \gamma_{q, L}),$$

где параметры  $r, b, T$  и  $R$  часто интерпретируются в приложениях к вычислительным и коммуникационным системам как скорость потока, его неравномерность, время задержки и производительность соответственно (см. далее § 4 и 5). Легко видеть, что функция  $TSPEC_{p, M, q, L}$  вогнутая (при условии  $M < L, p > q$ ), а функции  $\gamma_{r, b}(t)$  и  $\beta_{R, T}(t)$  выпуклые, для  $t \geq 0$ .

**Определение 1.** Функция потока есть неотрицательная, неубывающая функция времени:

$$A: \mathbb{R} \rightarrow \mathbb{R}_+ \cup \{+\infty\}, \quad A(t) \leq A(s), \quad \forall t < s. \quad \blacklozenge$$

С практической точки зрения функция потока может быть рассмотрена как счетчик  $A = \int_{t_1}^{t_2} R(t) dt$ .  $R(t) \geq 0, t_2 \geq t_1 \geq 0$ , считающий входящие в компонент вычислительной системы данные, описываемые неотрицательной функцией  $R(t)$ , или выходящие из него, тогда говорят о входной или выходной функции потока соответственно.

Будем называть функцию потока каузальной, если  $A(t) = 0, \forall t < 0$ .

Определим операции «свертки» и «обращения свертки».

**Определение 2.** Пусть даны две каузальные функции потока  $A$  и  $S$ , «свертка»  $A$  и  $S$  есть функция  $B = A \otimes S: \mathbb{R} \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ , такая что

$$A(t) \otimes S(t) := \wedge_{\tau \in \mathbb{R}} \{A(t - \tau) + S(\tau)\},$$

где  $\wedge \equiv \text{infimum}$ .  $\blacklozenge$

Основные свойства операции «свертки» приведены в Приложении 1.

Далее, если нет необходимости, аргумент  $t$  будет опускаться при записи уравнений.

**Утверждение 1.** Функция  $B$ , полученная в результате операции «свертки» над каузальными функциями потока  $A$  и  $S$ , также является каузальной функцией потока.  $\blacklozenge$

**Доказательство.** Докажем, что  $B(t_1) \leq B(t_2), \forall t_1 < t_2$ . Функция  $A$  неубывающая, следовательно,  $A(t_1 - s) + S(s) \leq A(t_2 - s) + S(s), \forall 0 \leq t_1 < t_2, s \in \mathbb{R}$ , тогда

$$\wedge_{s \in \mathbb{R}} \{A(t_1 - s) + S(s)\} \leq \wedge_{s \in \mathbb{R}} \{A(t_2 - s) + S(s)\}. \quad (1)$$

Так как  $A(t) = 0, S(t) = 0, \forall t < 0$ , то выражение (1) можно записать как  $\wedge_{0 \leq s \leq t_1} \{A(t_1 - s) + S(s)\} \leq \wedge_{0 \leq s \leq t_2} \{A(t_2 - s) + S(s)\}$ , т. е.  $B(t_1) \leq B(t_2), \forall t_1 < t_2$ .  $\blacklozenge$

Очевидно, что  $B(t) = 0, \forall t < 0$  и  $B$  — неотрицательная функция, так как  $A$  и  $S$  неотрицательны и каузальны.

**Определение 3.** Пусть даны две функции потока  $A$  и  $S$ , и  $S$  каузальна, операция «обращение свертки» есть функция  $H = A \oslash S$ , такая что

$$A(t) \oslash S(t) := \bigwedge_{\tau \in \mathbb{R}} \{A(t + \tau) - S(\tau)\}, \quad (2)$$

где  $\bigvee \equiv \text{supremum}$  и  $H$  есть наименьший поток, такой что  $H \oslash S \geq A$ . ♦

**Утверждение 2.** Функция  $H$ , полученная в результате операции «обращение свертки» над функциями потока  $A$  и  $S$ , где  $S$  каузальна, также является функцией потока. ♦

**Доказательство.** Докажем, что  $H(t_1) \leq H(t_2)$ ,  $\forall t_1 < t_2$ . Функция  $A$  неубывающая, следовательно,  $A(t_1 - s) - S(s) \leq A(t_2 + s) - S(s)$ ,  $\forall 0 \leq t_1 < t_2, s \in \mathbb{R}$ , тогда

$$\bigvee_{s \in \mathbb{R}} \{A(t_1 + s) - S(s)\} \leq \bigvee_{s \in \mathbb{R}} \{A(t_2 + s) - S(s)\},$$

так как  $S(t) = 0, \forall t < 0$ , то выражение (2) можно записать как:

$$\bigvee_{s \geq 0} \{A(t_1 + s) - S(s)\} \leq \bigvee_{s \geq 0} \{A(t_2 + s) - S(s)\},$$

т. е.  $H(t_1) \leq H(t_2), \forall t_1 < t_2$ .

Очевидно, что  $H$  — неотрицательная функция, так как  $S$  каузальна. ♦

Основные свойства «обращения свертки» приведены в Приложении 1.

**Определение 4.** Сетевой элемент имеет минимальную функцию обслуживания  $S$ , если она является каузальной функцией потока, и выходной поток системы  $B$  удовлетворяет условию  $B \geq A \oslash S$ . ♦

В рамках «Network calculus» обычно для описания потоков данных в системе применяют не функцию потока, а некоторую «производную» функцию — конверт потока, характеризующую его масштаб и которая была предложена в работе [5].

**Определение 5.** Функция  $E$  является конвертом потока  $A$ , если  $A \geq A \oslash E$ . ♦

Дополнение: будем полагать, что конверт потока удовлетворяет свойству субаддитивности т. е.  $E \oslash E \geq E$ , данное свойство почти всегда выполняется для реальных систем [15].

**Утверждение 3.** Для линейных систем с входным потоком  $A$ , выходным потоком  $B$ ,  $A(t) \geq B(t)$ , справедливо равенство:

$$E_B = E_A \oslash S, \quad (3)$$

где  $E_A$  — конверт входного потока,  $S$  — минимальная функция обслуживания  $E_B$  — конверт выходного потока. ♦

Доказательство см. в работе [1].

Требование равенства на длительном интервале наблюдения объема принимаемых и передаваемых компонентом данных, сформулированное в работе [5], можно переформулировать как требование, чтобы конверты входного и выходного потоков с

учетом задержки  $T$  на время обработки в компоненте были близки:

$$E_B(t + T) \approx E_A(t), \quad t \geq 0, \quad T \geq 0. \quad (3a)$$

Уравнение (3) используется для моделирования вычислительных систем и связывает конверт входного потока, функцию системы или ее отдельных компонентов обслуживания с конвертом выходного потока. Однако если в каком-либо из компонентов системы существенно изменяется объем данных на выходе компонента (один из примеров — это сжатие входного потока), т. е. нарушается условие (3a), то конверт выходного потока не может быть определен в рамках прямого применения метода «Network calculus», и уравнение (3) не может быть использовано. Далее мы рассмотрим методы, облегчающие расчет выходного потока в таких случаях.

## 2. МОДЕЛИРОВАНИЕ НЕОДНОРОДНОГО ПОТОКА В «NETWORK CALCULUS»

Как указывалось ранее, «Network calculus» не может быть применен для систем, где существует значительное отличие в длительном интервале наблюдения между объемом принятых и переданных компонентом моделируемой системы данных. Однако данная ситуация нормальная для большинства вычислительных систем. Поэтому в работе [6] предложен подход, позволяющий реализовать линейное изменение объема выходного потока по отношению к входному потоку (рис. 1), где оператор  $M$  на алгоритмическом уровне моделирующей программы обеспечивает линейное преобразование конверта входного потока. Предполагается, что при линейном изменении потока конверт потока также меняется линейно, однако доказательство этого отсутствует. Поэтому, докажем

**Утверждение 4.** Если  $B = lA + b, l \geq 0, b \geq 0$ , то конверт  $E_B$  выходного потока  $A$  можно представить в виде:

$$E_B = lE, \quad (4)$$

где  $E$  — конверт входного потока  $A$ .

**Доказательство.** Докажем, что  $E_B$  — конверт потока  $B$ , т. е. выполняется неравенство:

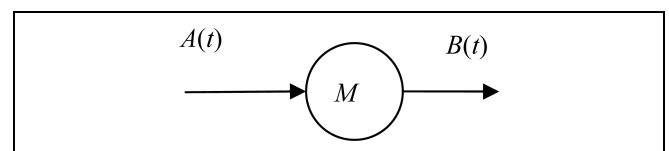


Рис. 1. Система с линейно зависимым потоком

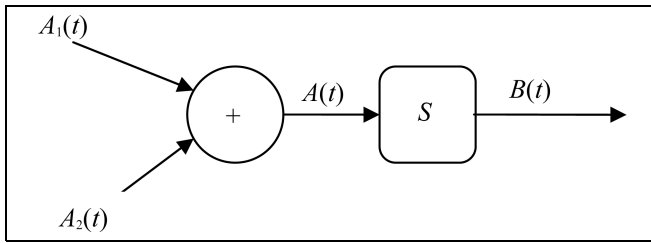


Рис. 2. Система с суммированием потоков

$IA + b \leq (IA + b) \otimes EI$ , для чего, применив свойство 4 «свертки» (см. Приложение 1), перепишем:

$$IA + b \leq (IA + b) \otimes EI = (IA) \otimes (EI) + b = I(\wedge_{\tau \in \mathbb{R}} \{A(\tau) + E(t - \tau)\}) + b. \blacklozenge$$

Подход, изложенный в работе [6], и уравнение (4) позволяют расширить применимость метода для анализа реальных вычислительных систем (пример расчета реальной системы приведен в работе [4]), но имеет ограничения. Существуют вычислительные системы, где конверт выходного потока меняется более сложным образом по отношению к конверту входного потока. Тогда описание конверта измененного потока возможно представить в виде суммы конвертов двух потоков (рис. 2) — основного потока и некоторого вспомогательного (рис. 2). Препятствие состоит в том, что решение задачи определения конверта выходного потока для системы с суммированием потоков

$$E_B = (E_{A_1} + E_{A_2}) \otimes S, \quad (5)$$

в рамках «Network calculus» нетривиально и в большинстве случаев требует применения численных алгоритмов и компьютеров. Данная проблема аналогична проблеме умножения сигналов для обычной линейной системы. Один из путей решения проблемы суммирования потоков (5) и определения конверта выходного потока заключается в подходе к суммированию как к агрегации [1, 7]. Основная идея состоит в раздельном рассмотрении влияния каждого из потоков на производительность сервера. Это позволяет удовлетворительно решить проблему нахождения выходного конверта основного потока в присутствии фонового потока, однако не позволяет определить конверт суммарного потока на выходе сервера. Рассмотрим возможность применения обобщенного преобразования Лежандра для расчета конверта выходного потока в случае суммирования входных потоков общего вида, но основное внимание сосредоточим на важном для практики частном случае, когда конверт одного из потоков описывается функцией  $\gamma_{r, b}(t)$ , также покажем расчет конверта выходного потока с помощью обобщенного пре-

образования Лежандра для общего случая уравнения (4) (линейное преобразование потока и пакетирование [1]):

$$E_B = kE_A + \rho, \quad k > 0, \quad \rho \geq 0. \quad (6)$$

### 3. ПРИМЕНЕНИЕ ОБОБЩЕННОГО ПРЕОБРАЗОВАНИЯ ЛЕЖАНДРА ДЛЯ РАСЧЕТА КОНВЕРТА ВЫХОДНОГО ПОТОКА

Преобразование Лежандра в мини-плюс алгебре, на которой базируется «Network calculus», играет ту же роль, что и преобразование Фурье для обычных линейных систем.

**Определение 6.** Преобразование Лежандра для вогнутой (выпуклой) функции  $x(t)$ , для которой существует функция, обратная ее производной, задается уравнением

$$X_L(\alpha) = a[(x')^{-1}(a)] - x((x')^{-1}(\alpha)),$$

где  $f^{-1}$  — функция, обратная функции  $f$ ,  $f' = df/dt$  — производная функции. Функция  $x(t)$  может быть восстановлена из своего образа повторным применением преобразования Лежандра к образу. Применение преобразования Лежандра для системы типа (5) могло бы упростить, как увидим далее, решение задачи нахождения конверта выходного потока. Как известно, применение преобразования Лежандра для не вогнутых (не выпуклых) функций приводит к появлению неоднозначности в образе функции, что делает невозможным выполнение обратного преобразования для анализа мини-плюс систем с конвертом потока общего вида. В целях устранения неоднозначности применяется предложенное в работе [8] обобщенное преобразование Лежандра (slope transform).

**Определение 7.** Для произвольной функции  $x(t)$  обобщенное преобразование Лежандра, базирующееся на операции supremum, определено как

$$X(\alpha) = \Lambda_{\vee, \alpha}(x) = \vee_{t \in R}(at - x(t)). \quad \blacklozenge \quad (7)$$

**Определение 8.** Для произвольной функции  $x(t)$  обобщенное преобразование Лежандра, базирующееся на операции infimum определено как

$$X(\alpha) = \Lambda_{\wedge, \alpha}(x) = \wedge_{t \in R}(at - x(t)). \quad (8)$$

Обозначим:  $\tilde{x}(t) = \Lambda_{\wedge, t}(\Lambda_{\vee, \alpha}(x(t)))$  и  $\tilde{x}(t) = \Lambda_{\wedge, t}(\Lambda_{\wedge, \alpha}(x(t)))$ .  $\blacklozenge$

Обобщенное преобразование обеспечивает однозначность в образе функции и набор свойств, сходных со свойствами преобразования Лежандра (см. Приложение 2). Обобщенное преобразование Лежандра (7) и (8) за исключением, может быть, некоторых граничных точек идентично обычному



преобразованию Лежандра для выпуклых и вогнутых функций соответственно.

Примеры обобщенного преобразования Лежандра некоторых простых функций [10]:

$$\Lambda_{\wedge, \alpha}(\gamma_{r, b}(t)) = \begin{cases} -\infty, \alpha < r, \\ -b, \alpha \geq r; \end{cases}$$

$$\Lambda_{\vee, \alpha}(\beta_{R, T}(t)) = \begin{cases} +\infty, \alpha < 0, \\ \alpha T, 0 \leq \alpha \leq R, \\ +\infty, \alpha > R; \end{cases}$$

$$\Lambda_{\vee, \alpha}(TSPEC_{p, M, q, L}) = \begin{cases} -\infty, \alpha < q, \\ \frac{L-M}{p-q}(\alpha - q) - L, q \leq \alpha \leq p, \\ -M, \alpha > p. \end{cases}$$

Теперь покажем, как с помощью обобщенного преобразования Лежандра в некоторых случаях можно упростить анализ систем типа (5) и (6) в рамках «Network calculus». Рассмотрим систему, изображенную на рис. 2. Пусть потоки  $A, A_1, A_2, B$  имеют конверты  $E, E_1, E_2, E_B$  и минимальная функция обслуживания равна  $S$ . Докажем полезные свойства:

**Утверждение 5.** Если  $E$  является конвертом для потока  $A$ , то и его аппроксимация  $\tilde{E}$ , найденная в классе вогнутых функций, также будет являться конвертом для данного потока. ♦

Доказательство. Пусть  $A(t-s) + E(s) \leq A(t-s) + \tilde{E}(s), \forall t \geq 0, s \in \mathbb{R}$ , или  $\wedge_{s \in \mathbb{R}} \{A(t-s) + E(s)\} \leq \wedge_{s \in \mathbb{R}} \{A(t-s) + \tilde{E}(s)\}$ .

**Утверждение 6.** Если  $E$  является процессом, то функция  $\tilde{E}$  также процесс. ♦

Доказательство. Очевидно, что  $\tilde{E}(s) \geq 0$ . Докажем от противного, что  $\tilde{E}(t_1) \leq \tilde{E}(t_2), \forall t_1 < t_2$ . Пусть  $E(t_1) \leq E(t_2), \tilde{E}(t_1) \geq \tilde{E}(t_1), \tilde{E}(t_2) \geq \tilde{E}(t_2) \forall t_1 < t_2$  по условию и пусть  $\exists t_1, \tilde{E}(t_1) > \tilde{E}(t_2), t_1 < t_2$ .

Пусть в некоторых двух точках  $t_1 < t_2, \tilde{E}(t) = E(t)$  (Приложение 2 свойство 6), тогда должно выполняться  $E(t_1) > E(t_2), t_1 < t_2$ , что противоречит условию утверждения. ♦

Утверждения, аналогичные утверждениям 5 и 6, можно сформулировать для минимальной функции обслуживания.

Запишем выражение для конверта выходного потока через функцию обслуживания и конверт входного потока, полагая, что функция обслуживания каузальна. Как указывалось ранее, уравне-

ние (5) не является линейным относительно операции сложения, однако, применяя обобщенное преобразование Лежандра и свойство 8 (см. Приложение 2), можно перейти от операции обращения свертки к операции вычитания образов:

$$\Lambda_{\wedge, \alpha}(E_B) = \Lambda_{\wedge, \alpha}(E_{A_1} + E_{A_2}) - \Lambda_{\vee, \alpha}(E_S). \quad (9)$$

Для произвольных функций, описывающих конверты потоков  $A_1$  и  $A_2$  вычисление обобщенного преобразования Лежандра для конверта суммарного потока на входе компонента

$$\Lambda_{\wedge, \alpha}(E) = \Lambda_{\wedge, \alpha}(E_{A_1} + E_{A_2}). \quad (10)$$

может оказаться вычислительно сложной процедурой, но для специального случая, часто встречающегося на практике, когда конверт вспомогательного потока  $E_2 = \gamma_{r, b}(t)$ , используя свойства 4 и 5 (см. Приложение 2), уравнение (10) можно переписать в виде:

$$\Lambda_{\wedge, \alpha}(E_{A_1} + \gamma_{r, b}) = \Lambda_{\wedge, \alpha-r}(E_{A_1}) - b. \quad (11)$$

Последовательно применяя уравнения (9) и (11) к уравнению (5), получим:

$$\tilde{E}_B = \Lambda_{\wedge, r}(\Lambda_{\wedge, \alpha-r}(E_{A_1}) + \Lambda_{\wedge, \alpha}(E_S)) + b, \quad (12)$$

где (см. утверждения 5 и 6)  $\tilde{E}_B$  является наименьшим вогнутым конвертом выходного потока, и если  $E_B$  является процессом, то  $\tilde{E}_B$  тоже процесс.

Для системы вида (6), уравнение (5) при использовании обобщенного преобразования Лагранжа и учете свойств 5 и 10 (см. Приложение 2), приобретет вид:

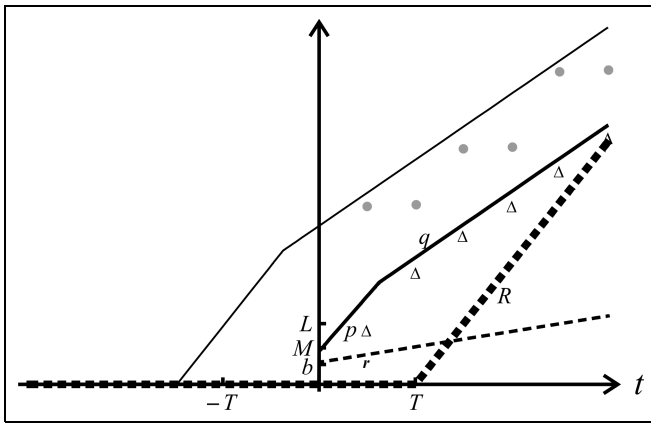
$$E_B \leq \tilde{E}_B = \Lambda_{\wedge, r}(\Lambda_{\wedge, \alpha/r}(E) - \rho - \Lambda_{\vee, \alpha/r}(S)). \quad (13)$$

Уравнения (9)–(13) могут быть применены для моделирования широкого класса вычислительных систем с неравным на длительном интервале наблюдении объемом принимаемых и передаваемых компонентом системы данных.

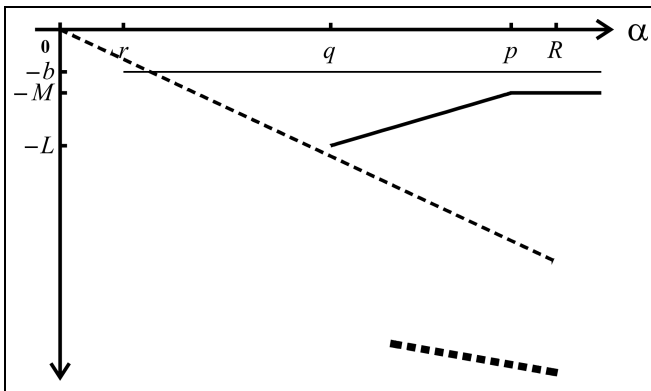
#### 4. ПРИМЕРЫ ПРИМЕНЕНИЯ ОБОБЩЕННОГО ПРЕОБРАЗОВАНИЯ ЛЕЖАНДРА ДЛЯ РАСЧЕТА СИСТЕМ С ИЗМЕНЕНИЕМ МАСШТАБА ПОТОКА

##### 4.1. Моделирование службы IntServ системы с дополнительным входным потоком

Рассмотрим стандартную модель службы IntServ [16] системы с дополнительным входным потоком (см. рис. 2). Реализация службы IntServ — одна из распространенных мер по обеспечению гарантированного качества обслуживания (QoS) в авто-



**Рис. 3. Конверт потока:**  $E_{A_1} = TSPEC$  — толстая линия;  $E_{A_2} = \gamma_{r,b}(t)$  — тонкая штриховая;  $S = \beta_{R,T}(t)$  — толстая штриховая;  $E_B$  — тонкая линия;  $\Delta$  — экспериментальный конверт входного потока,  $\bullet$  — экспериментальный конверт выходного потока



**Рис. 4. Обобщенное преобразование Лежандра:**  $\Lambda_{\wedge, \alpha}(\gamma_{r,b})$  — тонкая линия;  $\Lambda_{\wedge, \alpha}(TSPEC_{p,M,q,L})$  — толстая;  $-\Lambda_{\wedge, \alpha}(\beta_{R,T})$  — тонкая штриховая;  $\Lambda_{\wedge, \alpha-r}(TSPEC_{p,M,q,L}) - b - \Lambda_{\vee, \alpha}(\beta_{R,T})$  — толстая штриховая линия

матизированных системах управления реального времени.

Характеристики системы:  $E_{A_1} = TSPEC_{p,M,q,L}$ ,  $E_{A_2} = \gamma_{r,b}$ ,  $S = \beta_{R,T}$ , применив уравнения (5) и (12), просуммировав образы и выполнив повторно преобразование Лежандра для результирующей функции, вычислим конверт  $\tilde{E}_B$  (рис. 3 и 4):

$$E_B \leq \tilde{E}_B = \Lambda_{\wedge, \alpha}^{-1}(\Lambda_{\wedge, \alpha-r}(TSPEC_{p,M,q,L}) - b - \Lambda_{\vee, \alpha}(\beta_{R,T})).$$

Для модели параллельно получены результаты (см. рис. 3) на реальной вычислительной системе. Была собрана и измерена вычислительная система

с алгоритмом функционирования и структурой, соответствующей модели. Измерялись потоки на входе и выходе компонента и оценивался минимальный конверт потока по формуле  $\bar{E} = A \oslash A$  (см. работы [1, 3]), где  $\bar{E}$  — минимальный конверт потока,  $\bar{E} \leq \alpha$ ,  $A$  — поток. Определение параметров функции обслуживания  $\beta_{R,T}$  затрудняется тем, что прямые измерения выходного потока с последующим вычислением функции обслуживания по формуле  $S = A \oslash B$  невозможны, так как система, как уже отмечалось, нелинейная. Для того чтобы обойти данное ограничение, на идентичных технических средствах измерялись параметры функции обслуживания для системы с равным на длительном интервале наблюдения объемом принимаемых и передаваемых компонентом системы данных и вычисленные затем характеристики производительности использовались для расчета. Применение данного приема оправдано для систем, где существует избыток производительности и, в определенных пределах, изменения алгоритма функционирования или характеристик потока не приводят к изменению параметров компонентов системы.

#### 4.2. Моделирование фрагмента вычислительной системы с линейным преобразованием потока и пакетированием

Рассмотрим фрагмент системы (см. рис. 1), где поток  $A(t)$ , проходящий через выделенный компонент, претерпевает существенное долговременное изменение масштаба по следующему алгоритму (уравнение (6)):

- входной поток обрабатывается компонентом;
- проходя через компонент, объем потока уменьшается пропорционально коэффициенту  $l$ ,  $l \in \mathbb{R}_+$ ;
- выходные данные упаковываются в пакеты размером  $\rho > b$ , где  $b$  — неравномерность входного потока,  $b \in \mathbb{R}_+$ ,  $\rho \in \mathbb{R}_+$ .

Входной поток для компонента формируется посредством опроса источника данных с периодом  $T_\gamma$ . На поступивший запрос источник формирует пакет данных размером  $k$ . В результате неравномерности временных характеристик сервера и времени прохождения запроса по сети период опроса выдерживается с некоторой точностью  $\tau \ll T_\gamma$ . Данная модель описывает вычислительную систему вида «источник данных — сервер — клиент», часто реализуемую на практике в автоматизированных системах управления [4]. Известно [1], что конверт потока, поступающий от источника, в этом случае описывается функцией  $\gamma_{r,b}(t)$  с параметрами  $r = k/T_\gamma$ ,  $b = k(\tau + T_\gamma)/T_\gamma$ , эффект сжатия

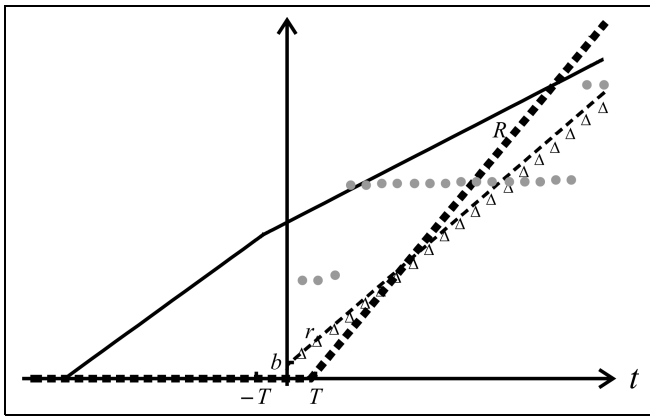


Рис. 5. Конверт потока:  $E = \gamma_{r,b}(t)$  — тонкая штриховая линия;  $S = \beta_{R,T}(t)$  — толстая штриховая;  $E_B$  — сплошная линия,  $\Delta$  — экспериментальный конверт входного потока;  $\bullet$  — экспериментальный конверт выходного потока

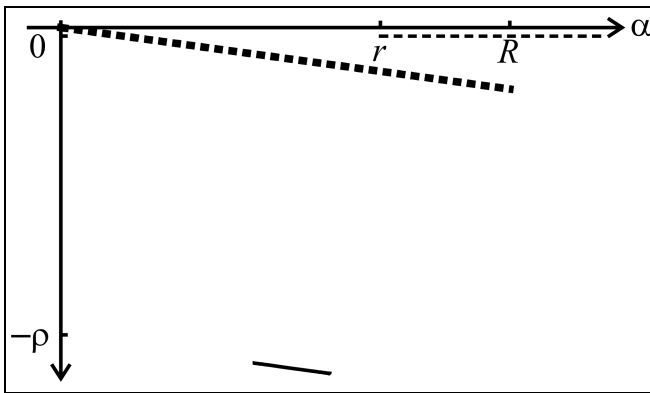


Рис. 6. Обобщенное преобразование Лежандра для конверта потока:  $\Lambda_{\alpha}(r,b)$  — тонкая штриховая линия,  $-\Lambda_{\alpha}(R,T)$  — толстая штриховая,  $l\Lambda_{\alpha}(r,b) - \rho - \Lambda_{\alpha}(R,T)$  — сплошная линия

данных описывается линейным преобразованием потока с коэффициентом  $l$ .

Алгоритм работы серверов, входящих в исследуемую систему, относится к типу алгоритмов с гарантированной скоростью обработки (guaranteed rate server), т. е.

$$h_n \leq f_n + T,$$

$$\text{где } \begin{cases} f_0 = 0, \\ f_n = \max(a_n, f_{n-1}) + s/R, \end{cases}$$

где  $s$ ,  $h_n$  и  $a_n$  — размер пакета данных, время окончания обработки и время появления информации для входа  $n$  соответственно. Для сервера указанного типа известна применимость функции  $\beta_{R,T}(t)$  для задания его функции обслуживания [4].

Следовательно, система имеет следующие характеристики:  $E = \gamma_{r,b}$ ,  $S = \beta_{R,T}$ . Тогда применяя уравнение (13), получим:

$$E_B \leq \tilde{E}_B = \Lambda_{\alpha}(l(\Lambda_{\alpha}(r,b) - \rho - \Lambda_{\alpha}(R,T))).$$

Просуммировав образы и выполнив повторно преобразование Лежандра для результирующей функции, вычислим конверт  $\tilde{E}_B$  (рис. 5 и 6).

Для модели получены результаты (аналогично, как в п. 4.1) на реальной вычислительной системе, приведенные на рис. 5.

## ЗАКЛЮЧЕНИЕ. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Как показало моделирование (§ 4), набор уравнений (см. § 3 и свойства, указанные в Приложении 2, позволяют в некоторых специальных случаях эффективно оценивать конверт выходного потока. Несмотря на то, что функционально оба примера (п. 4.1 и 4.2) очень похожи, они алгоритмически различны. Пример п. 4.1 предполагает, что изменение потока происходит непосредственно перед обработкой его сервером, а п. 4.2 иллюстрирует ситуацию, когда поток меняется после его обработки сервером. Данные алгоритмические отличия отражаются на методах вычислений.

В примерах использовались вогнутые и выпуклые функции для задания конвертов входных потоков и функции обслуживания. Это позволило получить точные значения для конверта выходного потока после восстановления его из образа.

Эффективность применения обобщенного преобразования Лежандра для решения уравнений вида (5) и (6) по сравнению с прямым применением уравнения (2) зависит от эффективности алгоритмов, как в том, так и другом случае. Мы использовали в своих расчетах алгоритм, опубликованный в работе [9], и результаты работы [10] и считаем их удобными и наглядными для анализа класса систем, сходных с теми, что рассмотрены в § 4. Алгоритмы для прямого решения уравнений вида (5) и (6) можно найти в работе [17].

Тесты, проведенные на реальных фрагментах вычислительных систем, показали хорошее согласие измеренных и модельных результатов (рис. 3 и 5) с точностью, определяемой корректностью предположений о характерах и параметрах входных потоков и функций обслуживания.

Остается открытым вопрос о точности расчета, когда конверты потоков описываются не вогнутыми функциями и результирующий конверт потока на выходе компонента системы является аппроксимацией реального конверта потока. Данный вопрос заслуживает отдельного исследования.



### ПРИЛОЖЕНИЕ 1

**Свойства операции «свертки».** Пусть функции  $A, B, C \in F$ , где  $F$  множество неотрицательных, неубывающих, каузальных функций, тогда:

- 1)  $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ ;
- 2)  $(A \otimes B) = B \otimes A$ ;
- 3)  $A, B, B', A' \in F$ , если  $A \leq B, A' \leq B'$ , то  $A \otimes A' \leq B \otimes B'$ ;
- 4)  $K \in \mathfrak{R}^+, (A + K) \otimes B = (B \otimes A) + K$ .

**Свойство операции «обращения свертки».** Пусть  $A, B, C \in F$ , тогда:

- 1)  $(A \oslash B) \oslash C = A \oslash (B \otimes C)$ ;
- 2)  $(A \otimes C) \oslash C \leq A \otimes (C \oslash C)$ ;
- 3)  $A \oslash B \leq C \Leftrightarrow A \leq B \otimes C$ .

Доказательства свойств приведены в работе [1].

### ПРИЛОЖЕНИЕ 2

**Обобщенное преобразование Лежандра.** Обозначим:  $\tilde{x}(t)$  — функция, полученная применением любой пары преобразований Лежандра к функции  $x(t)$  и знак  $\wedge$  означает любое из обобщенных преобразований Лежандра;  $X(\alpha) = \wedge(x)$  — образ функции,  $r \in \mathfrak{R}_+, b \in \mathfrak{R}_+$ .

Основные свойства:

- 1) функция  $\wedge_\vee(x)$  — выпуклая (convex);
- 2) функция  $\wedge_\wedge(x)$  — вогнутая (concave);
- 3)  $\forall t, \tilde{\tilde{x}}(t) \geq x(t)$ , где  $\tilde{\tilde{x}}(t)$  является наименьшей вогнутой функцией, большей чем  $x(t)$ . Аналогично  $\forall t, \tilde{\tilde{x}}(t) \leq x(t)$ , где  $\tilde{\tilde{x}}(t)$  является наибольшей выпуклой функцией; меньшей чем  $x(t)$ ;
- 4)  $\wedge(x) = X(\alpha) \Rightarrow \wedge(x(t) + rt) = X(\alpha - r)$ ;
- 5)  $\wedge(x(t) + b) = \wedge(x) - b$ ;
- 6)  $\forall t, \tilde{\tilde{x}}(t) = x(t) \Leftrightarrow x(t) \geq \frac{px(t-q) + qx(t+p)}{p+q} \forall p, q > 0$ ;
- 7)  $\forall t, \tilde{\tilde{x}}(t) = x(t) \Leftrightarrow x(t) \leq \frac{px(t-q) + qx(t+p)}{p+q} \forall p, q > 0$ ;
- 8)  $x(t) \otimes y(t) = \wedge_{\wedge, \alpha}(x(t)) + \wedge_{\wedge, \alpha}(y(t))$ ;
- 9)  $x(t) \oslash y(t) = \wedge_{\wedge, \alpha}(x(t)) - \wedge_{\wedge, \alpha}(y(t))$ ;
- 10)  $\wedge(x) = X(\alpha) \Rightarrow \wedge(rx(t)) = rX(\alpha/r)$ .

Доказательства свойств приведены в работах [15, 17].

### ЛИТЕРАТУРА

1. Le Boudec J.-Y., Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. Online Version of the Book Springer Verlag. — LNCS 2050. Version May 10, 2004.
2. Vantanski N., et al. Compensating the transmission delay in networked control systems // 14<sup>th</sup> Nordic process control workshop, NPCW07, Espoo, Finland, 2007.

3. Hwangnam Kim Hou, J.C. Network calculus based simulation for TCP congestion control: theorems, implementation and evaluation // INFOCOM 2004. Twenty-third Annual Joint Conf. of the IEEE Computer and Communications Societies. Publication Date: 7–11 March, 2004. — Vol. 4. — P. 2844–2855.
4. Масолкин С.И., Промыслов В.Г. Расчет некоторых параметров промышленной вычислительной сети объектов повышенного риска эксплуатации на примере АСУТП АЭС // Проблемы управления. — 2010. — № 1. — С. 47–52.
5. Cruz R.L. A Calculus for Network Delay. Part I: Network Elements in Isolation // IEEE Trans. on Information Theory. — Jan. 1991. — Vol. 37. — P. 114–131.
6. Schioler H., Jessen J., Dalsgaard J., and Larsen K. Network calculus for real time analysis of embedded systems with cyclic task dependencies. In G. Hu, editor // Proc. 20th Intern. Conf. on Computers and Their Applications, CATA 2005, March 16–18, 2005, Louisiana. — ISCA, 2005. — P. 326–332.
7. Lothar Thiele, Samarjit Chakraborty, Matthias Gries, Alexander Maxiaguine, Jonas Greutert. Embedded Software in Network Processors — Models and Algorithms. Lecture Notes In Computer Science; Vol. 2211 Proc. of the First International Workshop on Embedded Software table of contents. — 2001. — P. 416–434.
8. Maragos P. Slope transforms: Theory and application to non-linear signal processing // IEEE Trans. on Signal Processing. — 1995. — Vol. 43. — N 4. — P. 864–877.
9. Takashi Hisakado and Kohshi Okumara. Characterization of simple network using Legendre transform // IEEE Symposium ISCAS 2003. — P. 738–741.
10. Markus Fidler, Stephan Recker. Conjugate network calculus: A dual approach applying the Legendre transform // Computer Networks. — 2006. — 50 (8). — P. 1026–1039.
11. Cruz R.L. A calculus for network delay. II. Network analysis Information Theory, IEEE Trans. on Information Theory. — Jan. 1991. — Vol. 37. — P. 132–141.
12. Turner J. New Directions in Communications (or which way to the information age? // IEEE Communications. — 1986. — Vol. 24. — N 10. — P. 8–16.
13. Sidi M., Liu, W.-Z., Cidon, I., Gopal I. Congestion control through input rate regulation // IEEE Trans. on Communications. — 1993. — Vol. 41. — Iss. 3. — P. 471–477.
14. Agrawal R., Cruz R.L., Okin C., Rajan R. Performance bounds for flow control protocols // IEEE ACM Trans. on Networking. — Jun 1999. — Vol. 7. — Iss. 3. — P. 310–323.
15. Cheng-Shang Chang. Stability, queue length, and delay of deterministic and stochastic queuing networks // Automatic Control, IEEE Transactions on Issue Date. — May 1994. — Vol. 39. — Iss. 5. — P. 913–931.
16. Braden R., Clark D., Shenker S. Network Working Group Request for Comments: 1633, Integrated Services in the Internet Architecture: an Overview June 1994.
17. Nicos Zdarsky, Frank A., Martinovic, Ivan Schmitt, Jens B. The DISCO Network Calculator // Proceedings:MMB Conference: 14th GI/ITG Conference — Measurement, Modelling and Evaluation of Computer and Communication Systems Dortmund, 2008.

Статья представлена к публикации членом редколлегии В.Г. Лебедевым.

**Виталий Георгиевич Промыслов** — канд. физ.-мат. наук, вед. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ☎ (495) 334-75-71, ✉ v1925@mail.ru.



# МЕТОД ДЕФОРМАЦИИ ПУТИ В ЗАДАЧАХ ПЛАНИРОВАНИЯ ДВИЖЕНИЯ РОБОТОВ ПРИ НАЛИЧИИ ПРЕПЯТСТВИЙ<sup>1</sup>

Р.Ф. Гилимьянов, Л.Б. Рапопорт

Рассмотрена следующая задача планирования пути мобильного робота. Задан путь, проложенный в обход препятствий. При изменении положения препятствий или появлении новых требуется деформировать ранее спланированный путь таким образом, чтобы он огибал эти препятствия. Разработан метод решения, основанный на покомпонентном методе сглаживания кривизны пути и методе потенциалов и позволяющий не только огибать препятствия, но и сглаживать путь и его кривизну, как в двухмерном, так и в трехмерном случаях.

**Ключевые слова:** планирование пути, обход препятствий, метод потенциалов, деформация пути, мобильные роботы, сглаживание кривизны, В-сплайны.

## ВВЕДЕНИЕ

Задача планирования движения — одна из классических, возникающих при управлении различными робототехническими системами (роботами) [1–4]. Она заключается в следующем: для заданной модели робота в среде с неподвижными препятствиями, заданными начальной и конечной точками в конфигурационном пространстве требуется найти путь между этими точками, исключающий столкновения робота с препятствиями. Данная задача известна также как задача планирования пути.

Большинство методов планирования пути решают задачу обхода препятствий, не уделяя внимания гладкости и плавности пути. Обычно найденный путь представляется в виде последовательности точек, которые можно соединить отрезками прямых линий. При переходе с одного отрезка на другой угол наклона касательной к пути изменяется скачком. В автоматическом режиме робот вдоль такого пути движется менее точно и плавно и с большими энергетическими затратами. Поэто-

му актуальна задача сглаживания спланированного пути при наличии препятствий.

Другая проблема возникает, когда путь уже спланирован, а расположение препятствий изменилось либо появились новые препятствия. Тогда спланированный путь становится непригодным для движения робота без столкновений с препятствиями. В этом случае нужно перепланировать весь путь заново, что является весьма трудоемкой задачей. Либо уже во время движения робота применить один из локальных методов обхода препятствий или реактивное управление [1–5]. Это обеспечит обход препятствия, но не гарантирует, что робот вернется на ранее спланированный путь и достигнет цели. Существует другой подход к решению данной проблемы [3, 6–10], согласно которому ранее спланированный путь деформируется таким образом, что огибает препятствия. При этом полученный путь, как и ранее спланированный, соединяет начальную и конечную точки.

В настоящей работе предлагается метод в рамках данного подхода. Особое внимание уделяется гладкости полученного пути и плавности изменения его кривизны. В основу предлагаемого метода положены покомпонентный метод сглаживания кривизны пути [11] и метод потенциалов, широко применяемый для планирования пути и обхода

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (грант № 09-08-00064) и комплексной программы № 15 ОЭММПУ РАН.