

УДК 62.50 (575.2) (04)

**ПРИМЕНЕНИЕ ПАКЕТА MATLAB/SIMULINK  
ДЛЯ СИНТЕЗА КООРДИНИРОВАННЫХ ДЕЦЕНТРАЛИЗОВАННЫХ  
ЗАКОНОВ УПРАВЛЕНИЯ И МОДЕЛИРОВАНИЯ  
ВЗАИМОСВЯЗАННЫХ СИСТЕМ**

*Н.М. Лыченко* – канд. техн. наук, доц.

*П.С. Бастов* – инженер-программист

*П.В. Семин* – инженер-программист

---

Some tools for designing and investigation of decentralized control of continuous and discrete systems have been developed on the basis of MATLAB and SIMULINK package.

Для моделирования, исследования и синтеза систем автоматического управления наибольшую популярность приобрела среда MATLAB, в основном благодаря специализированной библиотеке Control Toolbox и возможностям своего приложения – пакета Simulink. В последние годы Simulink стал наиболее широко используемым средством для моделирования и анализа динамических систем. При моделировании с использованием пакета Simulink реализуется принцип визуального программирования, в соответствии с которым пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

Блок Simulink'a состоит из множества входов, множества состояний и множества выходов. Выход блока является функцией времени, входов и состояний блока. Выполнение модели происходит за несколько этапов. Сначала выполняется инициализация модели. На этом этапе Simulink собирает блоки в модель, проверяет размерность сигналов, типы данных, время квантования, а также вычисляет параметры блоков и определяет порядок выполнения блоков. Затем Simulink входит в цикл моделирования, где каждая итерация называется шагом моделирования. На каждом

шаге моделирования Simulink выполняет каждый блок в порядке, определенном на этапе инициализации. Для каждого блока Simulink вызывает функции, которые вычисляют состояния, производные и выходы блока в текущий момент времени. Это происходит, пока не завершится весь цикл моделирования.

Решение различных задач синтеза управления децентрализованными системами сводится к достаточно трудоемким вычислительным процедурам. Библиотека пакета Simulink содержит более 200 блоков, которые описывают модели различных элементов систем, включая генераторы сигналов, динамические блоки, блоки математических и нелинейных функций, блоки вывода сигналов, а также многочисленные элементы для сборки сложных моделей из простых блоков. Однако блоков для решения задач синтеза и исследования систем децентрализованного управления в пакете Simulink нет. В то же время Simulink позволяет расширять свои функциональные возможности с помощью механизма Simulink (S)-функций.

S-функция представляет собой описание блока Simulink'a на каком-либо компьютерном языке. S-функции используют особый синтаксис вызовов, который позволяет им взаимо-

действовать с решающими устройствами Simulink'a. Это взаимодействие очень похоже на то, которое имеет место между встроенными блоками и Simulink'ом. S-функция имеет обобщенную структуру и может быть использована для моделирования непрерывных, дискретных и гибридных систем. Результатом разработки S-функций являются файлы динамически загружаемой библиотеки (.dll), называемые MEX-файлами (Matlab Extension file).

Simulink взаимодействует с S-функцией посредством методов обратного вызова, которые реализованы в конкретной S-функции. Каждый метод выполняет предопределенную задачу, конкретная реализация метода зависит от той функциональности, которую необходимо реализовать в блоке.

**Проблема синтеза координированных децентрализованных законов оптимального управления взаимосвязанными системами**

Рассматривается задача синтеза управления взаимосвязанными системами большой размерности, состоящими из совокупности  $M$  подсистем, каждая из которых (для дискретных систем) описывается уравнением

$$x_i[k+1] = A_i x_i[k] + B_i u_i[k] + \varphi_i(x, u, k),$$

$$x_i(t_0) = x_{i0}, \quad \forall i = 1, \dots, M,$$

показатель качества всей системы может быть задан в самом общем виде:

$$2J = \|x[N]\|_{Q_1}^2 + \sum_{k=0}^{N-1} (\|x[k]\|_{Q_x} + \|u[k]\|_R + \psi(x, u, k)).$$

Здесь  $x_i[k] \in \mathbb{R}^{n_i}$ ,  $u_i[k] \in \mathbb{R}^{m_i}$  – векторы состояний и управлений, функции  $\varphi_i(x, u, k)$  – нелинейные взаимосвязи и нелинейные части динамики подсистем,  $\psi(x, u, k)$  – некоторая выпуклая функция.

Оптимизационная задача (ОЗ) решается посредством перехода к эквивалентной оптимизационной задаче (ЭОЗ) методом декомпозиционно-координационной оптимизации с адаптацией критерия [1] с помощью двухуровневой вычислительной процедуры и последовательной схемой координации, либо с параллельно-последовательной схемой координации [2]. Вычислительные процедуры имеют две отличительные особенности: на верхнем уровне формируется вектор координирующих пере-

менных  $[\alpha', \bar{x}', \beta', \bar{u}']$ , обеспечивающий сходимость процедуры к оптимальному для всей системы решению; на нижнем уровне независимо решаются ОЗ для каждой подсистемы при фиксированных координирующих переменных. Результатом решения оптимизационных задач являются переменные состояния  $x_i[k]$ , управляющие воздействия  $u_i[k]$  и вспомогательная функция  $f_i[k]$ . Оптимальное решение определяется итеративно, в результате взаимодействия информацией между координирующим (верхним) и нижним уровнями.

На рис. 1 представлена блок-схема вычислительной процедуры с последовательной схемой координации для решения задачи оптимального управления линейной взаимосвязанной системой

$$x_i[k+1] = A_i x_i[k] + B_i u_i[k] +$$

$$+ \sum_{j=1}^M (A_{ij} x_j[k] + B_{ij}[k] u_j[k]),$$

$$x_i(t_0) = x_{i0}, \quad \forall i = 1, \dots, M$$

с квадратичным несепабельным критерием

$$2J = \|x[N]\|_{Q_1}^2 + \sum_{k=0}^{N-1} (\|x[k]\|_{Q_x} + \|u[k]\|_R).$$

На этой блок-схеме:  $Q_{1x} = Q_{1xd} + Q_{1xof}$ ,  $R = R_d + R_{of}$ ,  $Q_1 = Q_{1d} + Q_{1of}$ ;  $Q_{2d}, Q_{2xd}, Q_{2ud}$  – блочно-диагональные матрицы в критерии ЭОЗ;  $D_i = (R_i + Q_{2ui} + B_i^T P_i[k+1] B_i)^{-1}$ . Специальный выбор матриц  $Q_{2d}$  позволяет решать на нижнем уровне стационарные уравнения Риккати вместо нестационарных, что значительно облегчает вычисления.

Анализ вычислительных процедур для решения задач синтеза координированных децентрализованных систем оптимального управления в различных постановках [3–5] показал, что при решении на нижнем уровне ОЗ можно выделить общие для всех постановок базовые слагаемые в правых частях разностных уравнений для вычисления вспомогательной функции  $f[k]$  и для вычисления состояний  $x[k]$ , а также для вычисления координирующих управляющих воздействий  $u_{ki}[k]$ . Эти базовые части соответствуют представленным на рис. 1 в

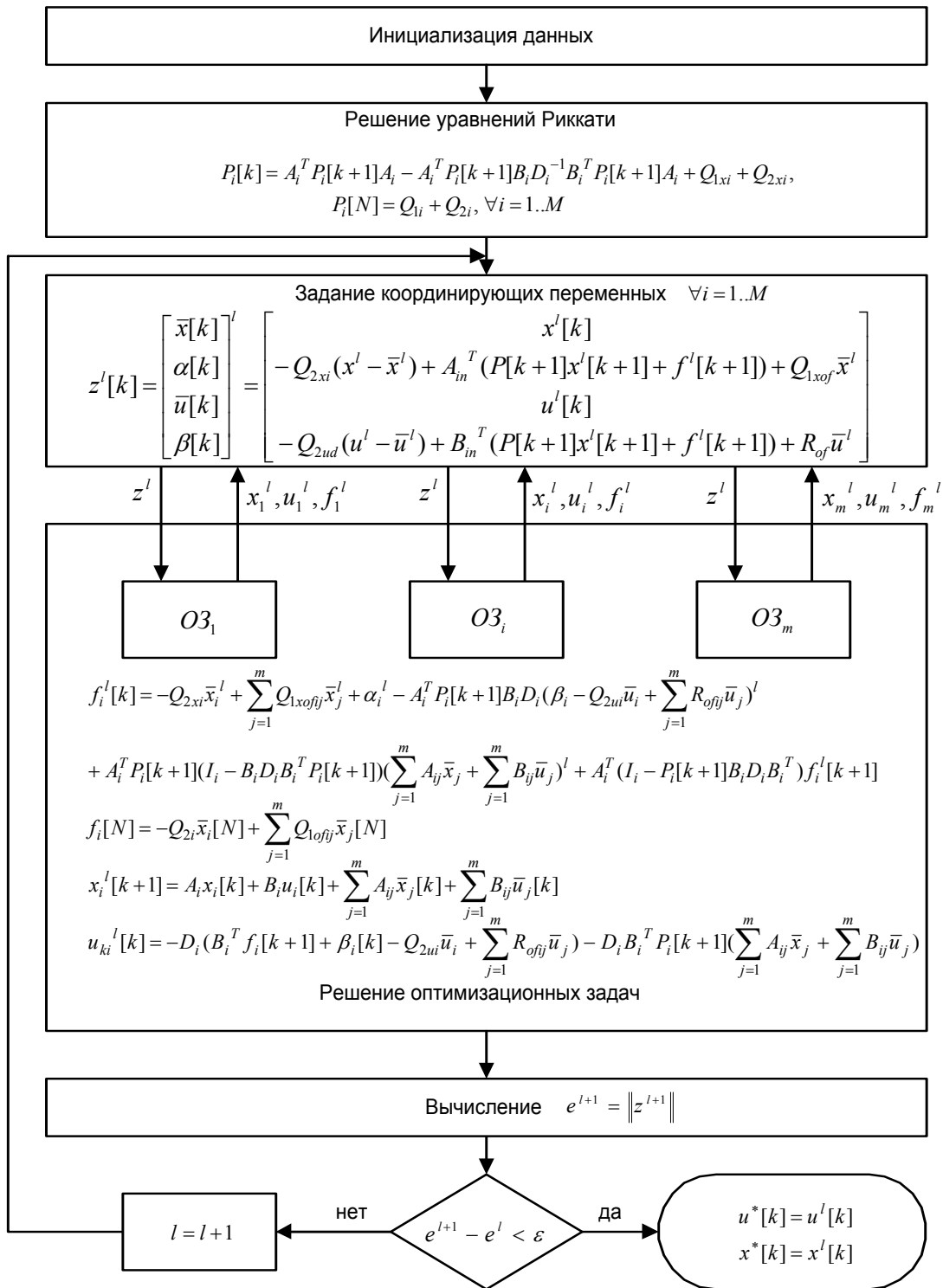


Рис. 1. Блок-схема вычислительной процедуры с последовательной схемой координации.

блоке Решение оптимизационных задач уравнениям. Кроме базовых, можно выделить дополнительные слагаемые, соответствующие различным постановкам оптимизационных задач. Примеры этих слагаемых для некоторых постановок приведены в таблице.

Учитывая возможность пакета Simulink создавать блоки с переменным числом входов, представляется рациональным "вводить" эти дополнительные слагаемые в правых частях разностных уравнений в качестве дополнительных входов новых, создаваемых с помощью механизма S-функций блоков. Кроме того, рационально совместить достоинства Simulink-моделирования с функцией программной (автоматической) декомпозиции полной системы на совокупность подсистем согласно указанной структуре декомпозиции.

#### Разработка программных средств

Программные средства представляют собой набор программ и Simulink-моделей, разделенных на два уровня. На верхнем уровне находится Matlab-скрипт, реализующий двухуровневую вычислительную процедуру (рис. 1), на нижнем – Simulink-модели для решения нестационарных уравнений Риккати, для расчета координирующих управлений, полных управлений и состояний по подсистемам, которые вызываются в процессе выполнения вычислительной процедуры. Модели для расчета координирующих управлений и состояний получены в результате разработки S-функций. Они обладают возможностью задавать дополнительные входы для решения задач в различ-

ных постановках. S- функции разработаны на языке программирования С.

**Разработка S-функций для вычисления координирующего управления и вычисления состояний** свелась к реализации определенного подмножества методов обратного вызова, необходимых для корректного взаимодействия со средой Simulink'a: инициализация Simulink-блока, задание времени моделирования, задание начальных условий, возвращение выходов Simulink-блока, возвращение производных состояний Simulink-блока, очистка выделенной памяти.

Структура программы S-функции для вычисления координирующего управления (рис. 2) несложная.

Функция mdlInitializeSizes() включает задание и проверку параметров (матрицы, необходимые для вычисления координирующего управления, вектор количества и размерностей подсистем, индекс подсистемы, конечные условия, количество необязательных входов, а также наличие возмущающего воздействия); установку количества непрерывных либо дискретных состояний (оно равно размеру подсистемы); установку количества входных портов (оно складывается из количества обязательных и необязательных входов); установку количества выходных портов; задание размера рабочих векторов, хранящих указатели на необходимые в ходе моделирования данные. Функция mdlInitializeSampleTimes() устанавливает время моделирования (непрерывное или дискретное).

Оптимизационная задача	Дополнение к вычислению		
	$f_i[k+1]$	$u_{ki}[k]$	$x_i[k]$
С возмущением и заданием в критерии	$-Q_{2xi}x_i^0 + A_i^T P_i[k+1] \times (I_i - B_i D_i B_i^T P_i[k+1]) \mu_i[k]$	$-D_i B_i^T P_i[k+1] \mu_i[k]$	$\mu_i[k]$
Для нелинейных систем с возмущениями	$A_i^T P_i[k+1](I_i - B_i D_i B_i^T P_i[k+1]) \times (\varphi(\bar{x}, \bar{u}, k) + \mu_i[k])$	$-D_i B_i^T P_i[k+1] \times (\varphi(\bar{x}, \bar{u}, k) + \mu_i[k])$	$\varphi(\bar{x}, \bar{u}, k) + \mu_i[k]$
Координированное децентрализованное слежение	$A_i^T P_i[k+1](I_i - B_i D_i^{-1} B_i^T P_i[k+1]) \times \mu_i[k] - G_i y_i^*[k]$	$-D_i B_i^T P_i[k+1] \mu_i[k]$	$\mu_i[k]$

В функции mdlInitializeConditions() происходит задание начальных условий (или конечных условий, если моделирование происходит из конечного момента времени в начальный – для вычисления функции  $f(t)$ , например). Кроме того, в этой функции инициализируется рабочий вектор указателей.

После того, как отработаны функции инициализации модели, моделирование переходит на второй этап – цикл моделирования. Он представлен двумя функциями – mdlOutputs() и mdlUpdate(), именно на них ложится вся вычислительная нагрузка на протяжении всего процесса моделирования. По окончании цикла моделирования вызывается функция mdlTerminate(), которая производит работу по очистке ранее зарезервированной памяти.

На основе разработанных S-функций были созданы *Simulink-модели (блоки), реализующие*

*вычисление координирующего управления и состояний системы.*

На рис. 3а изображен Simulink-блок для вычисления координирующего управления. Входами являются координирующие переменные  $\alpha[k], \bar{x}[k], \beta[k], \bar{u}[k]$ , дополнительные входы.

Выходами блока являются вспомогательная функция  $f_i[k]$  и координирующее управление  $u_{k_i}[k]$ . Входные значения хранятся в виде переменных рабочего пространства MATLAB'a. Выходные значения также сохраняются в рабочем пространстве. Для блока, вычисляющего состояния системы (рис. 3б), входами являются координирующее управление и координирующие переменные, а выходом – состояния  $x_i[k]$  и полные управления подсистемы  $u_i[k]$ . Параметры блоков вводятся в разработанном окне параметров.

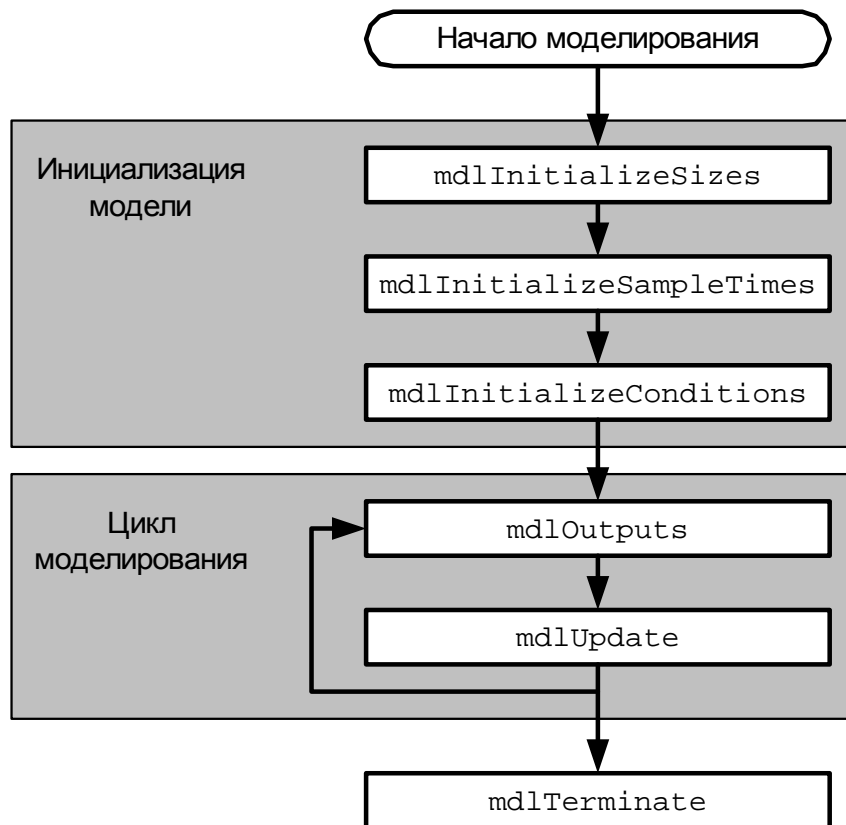


Рис. 2. Структура S-функции для вычисления координирующего управления.

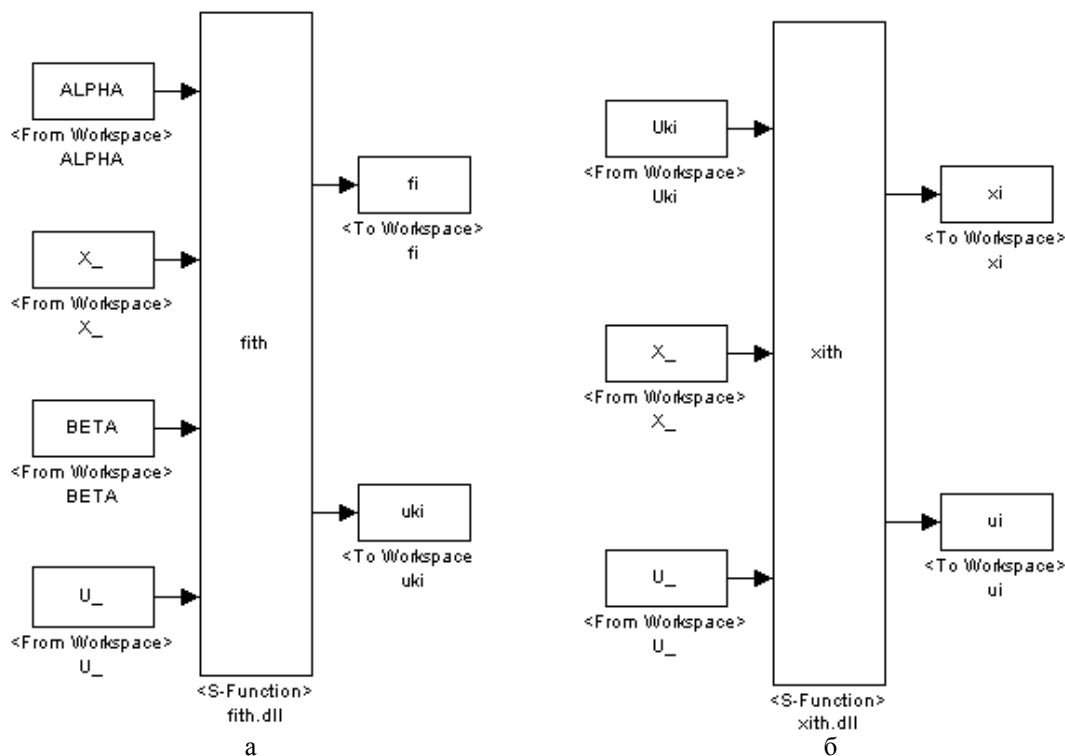


Рис. 3. Simulink-модели (блоки) вычисления для  $i$ -той подсистемы.  
 а – координирующего управления и б – полного управления и состояния.

**Дополнительные входы.** Количество дополнительных входов задается в качестве параметра блока, содержащего S-функцию. Далее на этапе инициализации модели этот параметр извлекается и в зависимости от его значения устанавливается соответствующее количество входов. Размерности векторов – дополнительных слагаемых в правых частях разностных уравнений, подаваемых на дополнительные входы, должны быть равны размерности подсистемы. Эти слагаемые учитываются только на итеративно вычисляемом выходе  $fi$ . Дополнительные слагаемые к  $uki$  добавляются в модели после выхода из блока. Примеры формирования дополнительных входов представлены на рис. 4–6.

Поскольку возмущения часто присутствуют в исследуемых системах и дополнительные слагаемые, связанные с ними, достаточно громоздки, для их учета сделано исключение. Для того чтобы добавить порт для ввода возмущений, достаточно отметить соответствующую опцию в параметрах S-функции. При

этом на подсистему подается полный вектор возмущений (подвектор возмущений выделяется внутри S-функции). Преобразования, связанные с присутствием в системе возмущений, жестко прописаны внутри S-функции, а не выполняются вне ее, как это подразумевается для других дополнительных входов.

Разработанные программные средства позволяют проектировать и исследовать законы координированного децентрализованного оптимального управления для непрерывных и дискретных систем с произвольной структурой декомпозиции и для задач в различных постановках. Для выполнения этих целей разработаны S-функции и скрипт-файлы, реализующие двухуровневые вычислительные процедуры. Разработанные программные блоки могут применяться инженерами-проектировщиками систем управления при решении задач синтеза оптимального управления взаимосвязанными системами большой размерности, а также при моделировании и исследовании таких систем в среде Matlab.

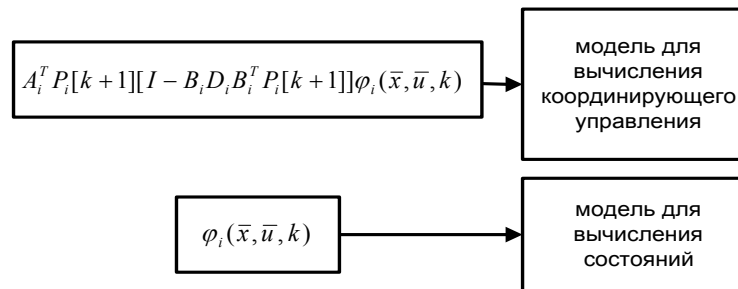


Рис. 4. Дополнительные входы для нелинейных систем.

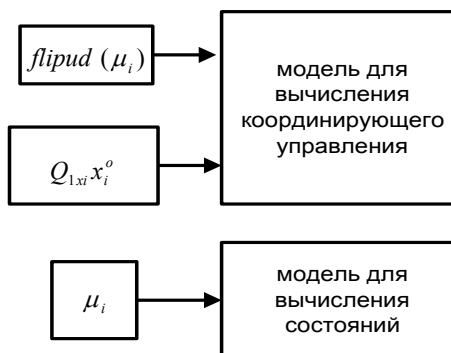


Рис. 5. Дополнительные входы для систем с заданием в критерии и возмущением.

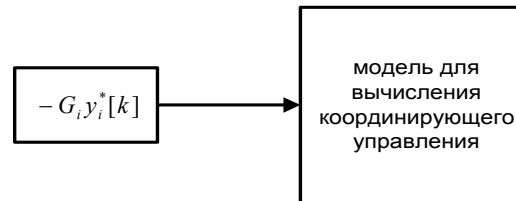


Рис. 6. Дополнительные входы для систем с эталонными траекториями.

### Литература

1. Миркин Б.М. Декомпозиционно-координационная оптимизация динамических систем с адаптацией критерия // Автоматика и телемеханика. – 2001. – № 7. – С. 148–157/
2. Nataly M. Lychenko. Decomposition-coordinated optimization of large-scale discrete systems with parallel-sequential coordinated scheme // Proc. of The 7th Mediterranean Conference on Control & Automation (MED99, Haifa, Israel, June 27 – July 4, 1999), Haifa, p. 420-429.
3. Лыченко Н.М. Алгоритмы децентрализованного координированного управления дискретными системами с заданными характеристиками динамики // Проблемы автоматки и процессов управления. – 2002. – № 1. – С. 10–17.
4. Лыченко Н.М. Синтез оптимальных управлений гибридными системами с параллельной схемой вычисления координирующих переменных // Сб. тр. II Международн. конф. "Идентификация систем и задачи управления" SICPRO'03. – М.: ИПУ РАН, 2003. – С. 693–706.
5. Миркин Б.М., Лыченко Н.М. Задача координированного децентрализованного слежения выходов взаимосвязанной непрерывной системы за эталонными траекториями // Проблемы автоматки и процессов управления. – 2002. – № 1. – С. 42–48.