

СТВОРЕННЯ РОЗПОДІЛЕНИХ МОДЕЛЕЙ ДИНАМІЧНИХ СИСТЕМ У МОДЕЛЮЮЧОМУ СЕРЕДОВИЩІ SIMULINK

Мілохов О.М., Забровський С.В.
Кафедра ЕОМ ДонНТУ
amilohov@ami.ua

Abstract

Milokhov A.N., Zabrovskiy S.V. Distributed dynamic system model building in Simulink simulation environment. There is considered the methods of distributed models building in Simulink environment. The method of distribution is chosen and implemented on base of simple loop-test.

Вступ

До динамічних систем тієї чи іншої складності можна віднести технологічні установки і лінії, процеси, що відбуваються в навколишньому середовищі, хімічні реакції, процеси в електричних мережах і взагалі будь-яку фізичну систему, параметри якої змінюються в часі відповідно визначеного закону. Моделювання динамічних систем є головним (і в ряді предметних галузей єдиним) методом їхнього дослідження і перевірки проектних рішень при створенні й удосконалюванні [1].

MATLAB є одним з перших широко розповсюджених математичних пакетів. У рамках даного пакета поставляється моделююче середовище Simulink, що має широкі можливості для моделювання процесів різної складності. Універсальність і широкі можливості по розширенню і програмуванню цього середовища уможлиблює її застосування до розподіленого моделювання динамічних систем.

Можливості організації розподілених обчислень у середовищі Simulink

Simulink є блочно-орієнтованою мовою моделювання дискретних і безперервних процесів [2]. Для нього існує бібліотека стандартних блоків, а також реалізована можливість створювати власні блоки. Методику створення блоків можна розділити на три основні частини:

- Компонування нового блоку з існуючих блоків. Операція нагадує угруповання об'єктів.
- Написання лінійних рівнянь мовою MATLAB (MATLAB function block)
- Написання власних блоків за допомогою S-функцій.

Останній метод є самим складним, але одночасно й самим гнучким методом створення нових блоків. Дані функції одержують повний контроль над ходом виконання моделі. Блок, створений такою функцією, може мати довільну кількість входів і виходів, свої внутрішні (локальні) змінні і що найважливіше - при виклику даного блоку можливе використання змінної часу (t). S-функції можна писати практично на будь-якій мові програмування, починаючи від вбудованої М-мови і закінчуючи С++ [2].

При створенні розподіленого моделюючого середовища на базі Simulink виникають наступні питання:

- Як розбити модель на підмоделі
- Як здійснити взаємодію між розподіленими компонентами

Перше питання відноситься переважно до проблеми балансування завантаження і не торкає суті проблеми організації розподілених обчислень. На існуючому етапі розробки можна розглядати статичне балансування завантаження з ручною або автоматичною розбивкою на підмоделі. Розподіл цих підмоделей між цільовими моделюючими серверами буде здійснюватися за допомогою "диспетчера", через який відбуватиметься синхронізація процесів моделювання. Процес розподілу моделі торкається підсистеми діалогу і топологічного аналізатора. При цьому використовуються алгоритми розподілу, мінімізуючі зовнішні зв'язки й урівнюючі обчислювальне навантаження серверів.

Друге питання пов'язане з сутністю розподілених обчислень. Вбудованої підтримки таких обчислень (моделювання) ні Simulink, ні MATLAB, як базова його платформа, не мають. Більш того, MATLAB-мова не має засобів, що дозволяли б організовувати обмін даними з іншими підмоделями (цільовими серверами). Однак дану проблему можна вирішити з використанням S-функцій. З їхньою допомогою можна організувати як передачу вхідних, так і вихідних параметрів підмоделі.

Основна риса моделювання безперервних процесів у Simulink- це те, що моделювання відбувається з перемінним кроком інтегрування. Таким чином, у різних підмоделях крок інтегрування може бути різний, у зв'язку з чим точки синхронізації (передачі вхідних/вихідних аргументів) можуть відрізнятися. Як один з варіантів реалізації обміну в такому "гетеро-часовому" середовищі можна запропонувати наступну структуру.

У даній структурі користувач направляє запити до сервера синхронізації, що виконує всі операції по ініціалізації, запуску моделювання, контролю процесу обміну даними, передачі результатів моделювання в клієнтське програмне забезпечення.

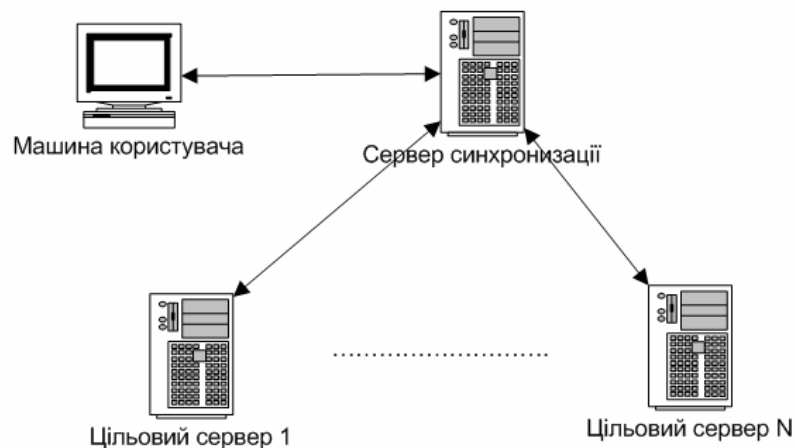


Рис. 1. Узагальнена схема обміну даними

Розглянемо принцип розподілу моделі на підмоделі для подальшого виконання на цільових машинах. Візьмемо як приклад схему, зображену на рис. 2. Для здійснення розподілених обчислень пропонується створити блоки, що представляють собою "віртуальні" входи-виходи і здійснюють передачу даних по мережі. Таким чином, модель буде мати структуру, показану на рис. 3.

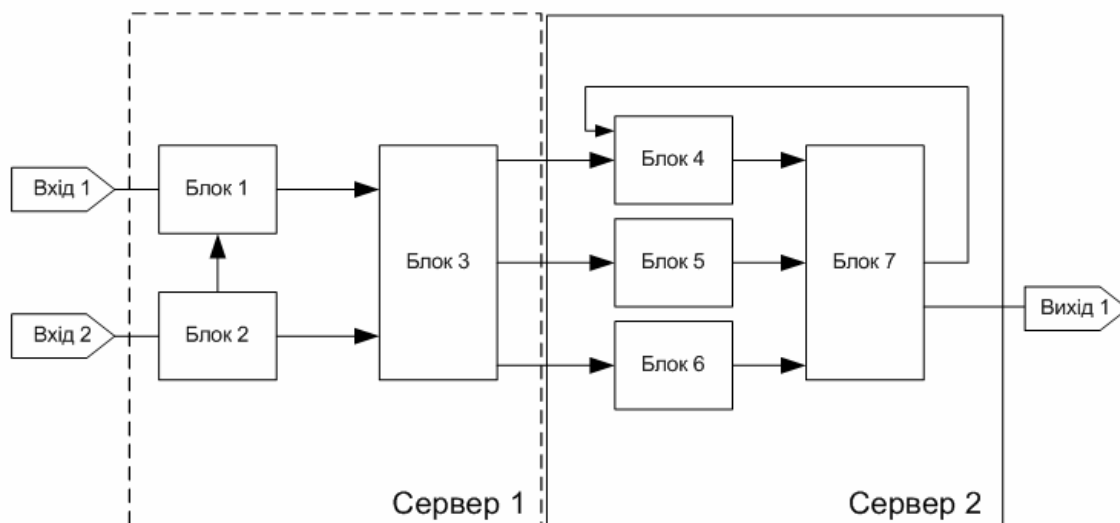


Рис. 2. Розподіл моделі на підмоделі

Особливої уваги заслуговує структура і функціонування "сервера синхронізації" та блоків, що здійснюють взаємодію з ним по мережі. Сервер синхронізації за допомогою топологічного аналізатора одержує блоки, для яких відомі вхідні значення і запускає їхнє моделювання. Залежні блоки направляють запит до сервера синхронізації про значення

параметрів у момент часу t . Сервер синхронізації обробляє інформацію, що надійшла від вихідних блоків і аналізує можливість представлення цієї інформації вхідним блокам, що очікують. Так, наприклад, якщо є інформація від вихідного блоку за моменти часу $(t-a)$ і $(t+b)$, то значення в момент часу t може бути отримане шляхом інтерполяції. Дана схема може застосовуватися, як досить універсальна для розподіленої системи моделювання.

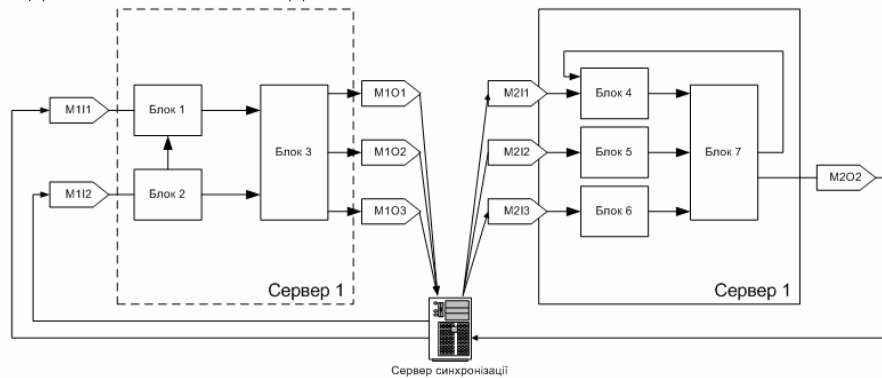


Рис. 3. Взаємодія підмоделей через "сервер синхронізації"

Реалізація розподіленої моделі кільцевого тесту

Для перевірки працездатності такої архітектури був обраний кільцевий тест, тому що він досить простий, легко перевіряється і дозволяє виявити помилку інтегрування, що накопичується. У нотифікації середовища моделювання Simulink даний тест має блок-схему показану на рис. 4. Система рівнянь, що описує його:

$$\begin{cases} \frac{dy}{dt} + \omega^2 x = 0 \\ \frac{dx}{dt} = y \end{cases}$$

Аналітичне рішення цієї системи: $x = A \cdot \sin(\omega t + \varphi)$, де A – амплітуда, ω – кругова частота, φ – початкова фаза. Початкові умови: $x(0) = A \cdot \sin \varphi$; $y(0) = A \cdot \omega \cdot \cos \varphi$.

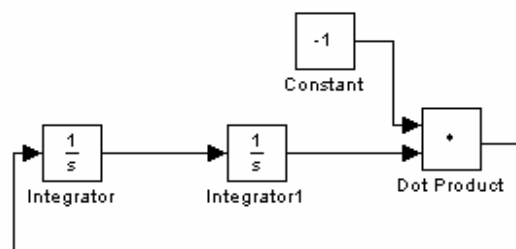


Рис.4. Кільцевий тест у MATLAB

Дана модель була розбита на дві підмоделі, як показано на рис. 5а і 5б.

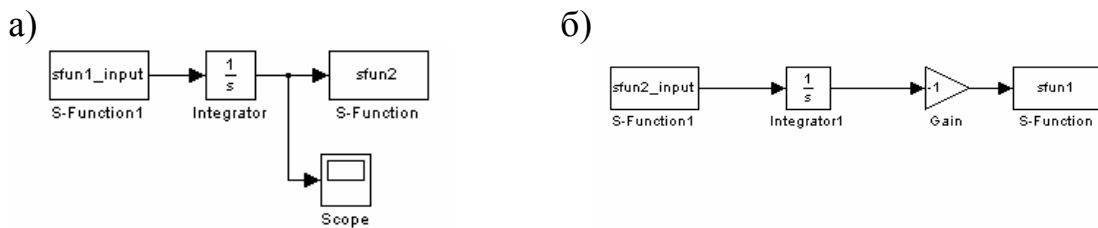


Рис. 5. Перша (а) та друга (б) підмоделі

Підмоделі взаємодіють в мережі Ethernet з центральним сервером синхронізації, що забезпечує збір і розподіл даних. При запиті даних сервер синхронізації проводить пошук найближчого наявного значення, інтерполює його методом кусочно-лінійної інтерполяції [3], а потім відправляє це значення до входу, який його запросив. Результат роботи показаний на рис. 6. Як засіб, що надає сервіси передачі даних по мережі Ethernet, був використаний "tcp/udp/ip Toolbox". З показаного нижче рисунка видно, що результат задовольняє аналітичному рішенню $x = A \cdot \sin(\omega t + \varphi)$. Для оцінки похилок інтегрування при різних частотах ω був проведений експеримент, результат якого показаний у табл. 1. Кінцевий час інтегрування - 10 сек. Використаний метод інтегрування - Рунге-Кута.

Таблиця 1

Максимальні похибки моделювання кільцевого тесту

f , Гц	ω , c^{-1}	Δx_{max} ($h=0,01c$)	Δx_{max} ($h=0,001c$)	Δx_{max} ($h=0,0001c$)	Δx_{max} ($h=0,00001c$)
1	6,2832	$7,7 \cdot 10^{-6}$	$7,8 \cdot 10^{-10}$	$8 \cdot 10^{-14}$	$7,2 \cdot 10^{-14}$
10	62,832	0,64	$8,2 \cdot 10^{-5}$	$8,2 \cdot 10^{-9}$	$1 \cdot 10^{-10}$
100	628,32	-	1,18	$8,2 \cdot 10^{-4}$	$8,2 \cdot 10^{-8}$
1000	6283,2	-	-	1,18	$8,2 \cdot 10^{-3}$

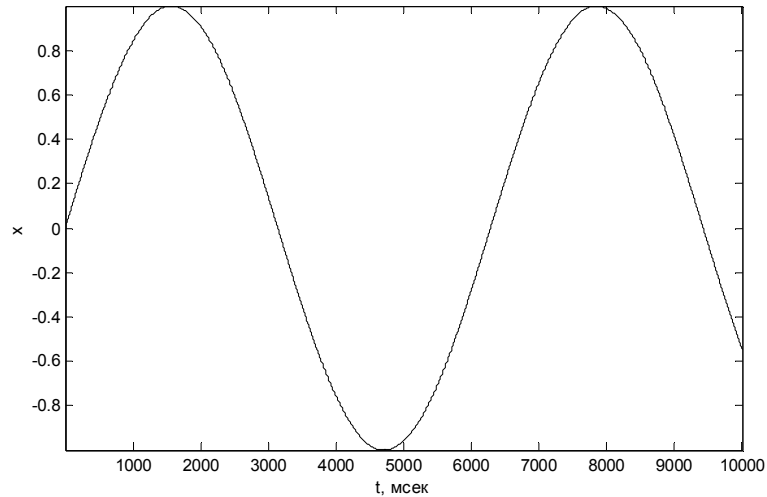


Рис. 6. Результат виконання моделювання
(рішення щодо невідомої x)

Висновки

У даній статті розглянуто спосіб розподіленого моделювання в середовищі Simulink. Як середовище обміну даними використана локальна мережа Ethernet з протоколом спілкування TCP/IP. Розроблений кільцевий тест показав працездатність запропонованого способу моделювання в необхідному діапазоні частот. Він може застосовуватися при моделюванні складних динамічних систем.

Література

1. В.А. Святний. Проблеми паралельного моделювання складних динамічних систем. - Наукові праці ДонДТУ, серія ІКОТ, вип..6, Донецьк, 1999, с. 6-14.
2. Online Simulink user's guide. Copyright 1984-2000 The MathWorks Inc.
3. В.А. Святний. Гибридные вычислительные системы. К.:«Вища школа», 1980, 248 с.

Дата надходження до редколегії: 10.11.2003 р.