

УДК 681.51 : 004.942

Максим Багінський

Черкаський національний університет ім. Б. Хмельницького, Україна

ЗАБЕЗПЕЧЕННЯ ІНТЕГРАЦІЇ ЗОВНІШНІХ МОДЕЛЕЙ У СИСТЕМИ УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ

Maxim Baginskiy

ENSURING INTEGRATION OF EXTERNAL MODELS IN REAL-TIME CONTROL SYSTEMS

Одержання достовірних ефективних рішень за умов ризику, жорстких обмежень на часові та матеріальні ресурси без автоматизації процесів прийняття рішень та адаптивного управління у ряді випадків викликає суттєві труднощі. Це є наслідком того, що процеси є трудомісткими, часто виконуються за умов невизначеності, нечіткості, неточності, стохастичності процесів [1].

Вирішити частину таких проблем можна використовуючи адаптивне керування процесами та прогнозування на основі моделювання об'єктів чи процесів. Існує парадигма об'єктно-орієнтованого програмування, яка пропонує підхід до вироблення стратегії управління виробничими установками, і полягає в широкому використанні комп'ютерних моделей у контурі управління у масштабі реального часу.

Способи заміни моделі можна розділити на дві категорії. Перша – заміна параметрів моделі. Друга – заміна власне моделі. Прикладом першого випадку може бути поліном Колмогорова-Габора. При необхідності змінити поведінку моделі, необхідно змінити коефіцієнти при членах полінома. Другий підхід передбачає не тільки модифікацію параметрів моделі, а й зміну її виду. Наприклад, спочатку для прогнозування деякого параметра використовується проста модель лінійної регресії, яку потім буде замінено на поліном Колмогорова-Габора.

Якщо говорити про способи практичної реалізації, то в першому випадку модель може бути написана на мові програмування високого рівня, скомпільована під архітектуру виконавчого пристрою та зашита в ПЗП виконавця, а інформація про параметри моделі може надходити із-зовні по каналах зв'язку від керуючого пристрою до виконавчого. Другий випадок є складнішим. Різні моделі можуть мати кардинально різні способи реалізації, а їх узагальнення, для представлення різних видів моделі через однаковий інтерфейс, не завжди можливий.

Вирішення задачі організації середовища функціонування моделі може здійснюватися двома способами. Перший – реалізація власних інструментів для створення та відпрацювання моделей в масштабах реального часу. Інший варіант – використання сторонніх уже апробованих рішень. В такому випадку для інтеграції отриманих моделей у виконавчі пристрої необхідно лише створити інструменти обміну між середовищами генерації та відпрацювання моделей.

Деякі математичні середовища надають досить великі можливості для інтеграції. Так, математичний пакет Matlab можна використовувати із іншими програмами як засіб, який буде виконувати обчислення математичних моделей. Дане середовище досить потужне і надає зручний рівень абстракції для вирішення різних задач. Тому при використанні Matlab в ролі системи для створення-виконання моделі можна отримати додаткові переваги в забезпеченості засобами роботи із складними даними, гарантоване функціонування інструментів, які надаються системою. Варто зауважити, що система Matlab в цілому є досить громіздкою, тому можна поставити під сумнів можливість використання Matlab у вбудованих та системах реального часу.

Іншою пропонованою альтернативою є використання механізму, який часто застосовується при необхідності розширення функціональності деякого програмного засобу – використання скриптових мов. Такий підхід часто використовується в САД-системах, пакетах 3D графіки, комп'ютерних іграх. Одною із скриптових мов, яка заслуговує уваги, є Python. Python є стандартним

компонентом *nix систем і використовується як засіб для розробки невеликих утиліт. Відповідно, для даної мови розроблено досить багато бібліотек і модулів для виконання різних задач. Завдяки проекту Sage – відкритій математичній бібліотеці для Python [2], – інколи в наукових колах Python використовується як вільний замітник Matlab. Враховуючи це, Python може бути адекватною альтернативою Matlab в плані обробки та аналізу даних, створення моделей. Разом з тим інтерпретатор Python може бути використаний як виконавчий елемент розроблених моделей. Варто відмітити можливість використання мови для вбудованих систем, оскільки доступним є вихідний код, який може бути зібраний під необхідну платформу.

Інша скриптова мова, яку можна розглядати в руслі пошуку засобу для представлення та виконання моделей – Lua. Як і Python, дана мова є відкритою, і має доступну програмну реалізацію. Для роботи Lua використовує компілятор в байткод і віртуальну машину для виконання згенерованого байткоду. Варто відмітити ще те, що для даної мови розроблено компілятор LuaJIT, який забезпечує умови для виконання програмного коду з великою швидкістю та може бути використаний для критичних задач [3]. Реалізація мови компактна і, як зазначено раніше, відкрита. Тому є можливість використання мови у вбудованих системах. На відміну від Python, для Lua немає такої великої кількості бібліотек, тому реалізацію частини математичного апарату потрібно буде реалізовувати в C/C++. Але це дозволяє використовувати для складних обрахунків оптимізований код, скомпільований під конкретну платформу.

Ще одним варіантом уваги варіантом є заміна моделей як скомпільованого програмного коду. Даний підхід повинен забезпечити найбільшу швидкість, проте вимагає додаткового дослідження в плані використання для вбудованих систем.

Таким чином, за наявності кількох можливих варіантів вирішення задачі інтеграції зовнішніх моделей в системи управління реального часу, постає питання оптимальності їх вибору для конкретної ситуації. Інструменти розробки та виконання моделей можуть бути оцінені за наступними критеріями:

- швидкість виконання моделі;
- швидкість зміни моделі;
- вимоги до розміру ОЗП виконавчого пристрою;
- можливість інтеграції у вбудовані системи;
- тип, математичний апарат та складність моделей;
- складність створення моделі;
- складність та збитковість моделі;
- об'єм моделі та форма представлення.

На основі комплексної оцінки за запропонованими критеріями можуть бути отримані ефективні рішення у реалізації систем адаптивного управління за модельно-орієнтованим підходом.

Література

1. Кучеренко В.С. Автоматизоване управління складними об'єктами за умов невизначеності на основі гібридних імітаційних моделей: Автореф. дис... канд. техн. наук: 05.13.06 / Харк. нац. ун-т радіоелектрон. – Х., 2007. – 19 с. – укр.
2. Sage: Open Source Mathematics Software [Електронний ресурс]. – Режим доступу: <http://www.sagemath.org/> (Перевірено 15.11.2012)
3. Performance Comparison [Електронний ресурс]. – Режим доступу: <http://luajit.org/performance.html> (Перевірено 20.11.2012)