

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА ГЕНЕТИЧНА ОПТИМІЗАЦІЯ СИСТЕМ КЕРУВАННЯ ЗАСОБАМИ ПРОГРАМУВАННЯ LABVIEW

В.А. Жеребко, О.А. Писаренко, В.П. Драбинко

Процедури проектування, синтезу, імітаційного моделювання та цільової реалізації алгоритмів вирішення оптимізаційних задач у ієрархічних автоматизованих системах керування складними об'єктами вимагає використання різноманітного математичного та програмно-апаратного забезпечення впродовж науково-дослідних та проектних робіт. У даній статті пропонується уніфікація математичного та алгоритмічного забезпечення у вигляді єдиної концепції вирішення задач оптимального керування. А у якості уніфікації програмно-апаратних засобів реалізації математично-алгоритмічних напрацювань пропонується використання програмного забезпечення LabVIEW та промислового контролера CompactRIO, на базі яких розроблено програмний інструментарій вирішення задач оптимізації із використання генетичного алгоритму.

Ключові слова: імітаційне моделювання, автоматизація систем керування, оптимізація систем керування, LabVIEW, генетичний алгоритм, генетичний оптимізатор.

Процедуры проектирования, синтеза, имитационного моделирования и целевой реализации алгоритмов решения оптимизационных задач в иерархических автоматизированных системах управления сложными объектами требует использования разнообразного математического и программно-аппаратного обеспечения в течение научно-исследовательских и проектных работ. В данной статье предлагается унификация математического и алгоритмического обеспечения в виде единой концепции решения задач оптимального управления. А в качестве унификации программно-аппаратных средств реализации математически-алгоритмических разработок предлагается использование программного обеспечения LabVIEW и промышленного контроллера CompactRIO, на базе которых разработан программный инструментальный решения задач оптимизации по использованию генетического алгоритма.

Ключевые слова: имитационное моделирование, автоматизация систем управления, оптимизация систем управления, LabVIEW, генетический алгоритм, генетический оптимизатор.

The procedures for the engineering, synthesis, simulation and targeted implementation of algorithms for solving optimization problems in hierarchical automated control systems require complex objects using a variety of mathematical and software and hardware for research and engineering work. In this article the unification of mathematical and algorithmic support in the form of a unified concept of solving optimal control is proposed. We propose to use LabVIEW software and CompactRIO industrial controller as the unification of software and hardware implementation of mathematical-algorithmic developments, on which developed software tools to solve optimization problems using genetic algorithm.

Key words: simulation, automation of control systems, optimization of control systems, LabVIEW, genetic algorithm, genetic optimizer.

Вступ

Вирішення оптимізаційних задач керування в складних системах не мають традиційного простого рішення і в кожному конкретному випадку потребують трудомістких наукових розробок [1]. Це пояснюється тим, що оптимізаційні задачі – вкрай складні процедури, що передбачають вивчення та імітаційне моделювання складних об'єктів керування (ОК), стратегій та ситуацій, створення альтернатив та пошук варіантів рішень на альтернативах, виходячи з конкретного ситуаційного стану ОК та навколишнього середовища. Можливість реалізації таких процедур обмежується з однієї сторони реальною динамікою ОК і швидкодією технічних засобів реалізації ієрархічних автоматизованих систем керування (АСК), з іншої – різноплановістю, складністю та неефективністю алгоритмів вирішення таких задач, особливо в умовах невизначеності, зміни параметрів ОК, збурень, зміни критеріїв функціонування та інше. Все це призводить до того, що оптимізаційні задачі в АСК в повній мірі практично не реалізуються, хоч повинні вирішуватися на кожному рівні ієрархічної АСК. При вирішенні оптимізаційних задач в ієрархічних АСК на всіх рівнях можна застосовувати, як основу, методику вирішення двоетапної задачі оптимального керування (ЗОК).

Суть єдиної концепції вирішення ЗОК полягає в уніфікованому підході до вирішення оптимізаційних задач на всіх рівнях ієрархічної АСК. Відповідно до цієї концепції оптимізаційні задачі на всіх рівнях представляються узагальноною моделлю у вигляді кортежу, що об'єднує модель ОК, критерії, які представляють оптимізаційну задачу, обмеження на керуючі впливи і стани ОК та проміжок часу на якому вирішується оптимізаційна задача. Оптимізаційні задачі вирішуються шляхом генетичного алгоритму на моделях АСК, як задачі параметричної оптимізації регуляторів. Моделі ОК представляються лінгвістичними або нейромережевими структурами [2], параметри яких також налагоджуються генетичним алгоритмом [3], як задачі параметричної оптимізації. Такий підхід до вирішення оптимізаційних задач керування зводить весь процес до евристики і дозволяє уникнути використання складних математичних процедур, як при побудові моделей ОК, так і при вирішенні оптимізаційних задач.

Єдина концепція реалізує методику вирішення двоетапної задачі оптимізації на всіх рівнях ієрархічної АСК у такій узагальненій послідовності:

- представити ЗОК яка включає в себе кортеж, що об'єднує модель ОК, критерії, які представляють оптимізаційну задачу, обмеження на керуючі впливи і стани ОК та проміжок часу на якому вирішується оптимізаційна задача;
- розробити модель ОК, виходячи із апріорних та експериментальних даних;
- виходячи з моделі ОК синтезувати структуру регулятора і побудувати модель системи керування;
- знайти оптимальні параметри налагодження регулятора, що вирішують оптимізаційну задачу на моделі системи керування;
- шляхом моделювання роботи моделі системи керування знайти множину оптимальних програм керування ОК для різних допустимих умов і станів його функціонування, створити необхідний банк оптимальних програм керування;
- реалізувати замкнуту систему керування ОК з оптимально налаштованим регулятором;
- для вирішення оптимізаційної задачі використовувати відповідну оптимальну програму керування із банку оптимальних програм та керування оптимального регулятора;
- одержану додаткову інформацію про якість вирішення оптимізаційної задачі в процесі функціонування системи використати для адаптації системи керування.

При вирішенні багатокритеріальних оптимізаційних задач використовуються технології скаляризації багатокритеріальних задач, зокрема, це вибір основного критерію і зведення інших до рівня обмежень, скаляризація на основі компромісів, або пріоритетів, послідовна оптимізація критеріїв із області Парето, згортка критеріїв шляхом їх «розмивання» на нечітких множинах та інші.

Інтелектуальні системи керування на основі лінгвістичних і нейромережових моделей з генетичним пошуком оптимальних параметрів налагодження моделей і регуляторів можуть бути основою для побудови структур систем керування, що вирішують оптимізаційні задачі на всіх рівнях ієрархічної системи керування. Виходячи з цього розроблено єдину концепцію вирішення оптимізаційних задач для всіх рівнів ієрархічної системи керування та технологію її реалізації. Технологія реалізації єдиної концепції потребує програмних середовищ для створення та моделювання лінгвістичних і нейромережових систем керування і програмних засобів для реалізації оптимізаційних процедур генетичного пошуку оптимальних параметрів моделей об'єктів і регуляторів.

Математичне вирішення оптимізаційних задач це достатньо довготривалий та витратний обчислювальний процес. Зазвичай вирішення таких задач відбувається на етапі дослідження ОК, проектуванні АСК, імітаційного моделювання системи в цілому тощо тобто поза режимом експлуатації АСК на реальному ОК. Сучасний стан автоматизації ОК та побудови АСК вимагає від систем здатності до вирішення ЗОК у реальному часі. Часто густо процедури моделювання, проектування, програмування, цільового виконання виконуються на зовсім різних програмно-апаратних платформах, різноманітними програмними засобами та технічними пристроями [4]. Тож окрім уніфікації математичного апарату щодо вирішення ЗОК в складних АСК постає також питання у виборі універсального програмно-апаратного забезпечення для подальшої практичної реалізації оптимізаційних концепцій. Такого єдиного готового рішення серед загалу програмних засобів на ринку не існує. Тож на базі програмного комплексу LabVIEW компанії National Instruments було розроблено програмний інструментарій для проектування і моделювання лінгвістичних та нейромережових моделей ОК і систем керування та програмних засобів для реалізації генетичного алгоритму (ГА). У якості апаратної виконавчої платформи для проведення наукових досліджень використовується промисловий контролер автоматизації CompactRIO компанії National Instruments. Використання такого комплексу програмно-апаратних засобів значно спрощує проведення досліджень, а саме процедур моделювання, налагодження та програмної реалізації синтезованих структур в АСК.

У даній статті наведений узагальнений опис програмного інструментарію що націлений на вирішення пошукових задач оптимізації в ієрархічних АСК на базі ГА.

1. Програмний засіб для моделювання й оптимізації систем керування

Програмний засіб (ПЗ) для оптимізації систем керування реалізовано у вигляді бібліотеки модулів «Генетичний оптимізатор» (GenOp) так званих віртуальних інструментів VI (програм і даних), які розширюють бібліотеку стандартних програмних модулів та засобів автоматизованого проектування, що реалізовані виробником в LabVIEW. Базова задача GenOp – це віднайти стратегію оптимального керування технологічного ОК з точки зору критеріїв якісного та енергозберігаючого керування, зокрема налаштування оптимального регулятора. Базовий принцип роботи GenOp полягає у математичних розрахунках щодо пошуку оптимальних законів керування ОК на базі класичних, нейронечітких та еволюційних підходів, зокрема генетичного ГА.

Загальна структурна системи GenOp наведена на рис. 1. Перелік основних функцій інструментальної системи GenOp:

- створення, збереження математичних моделей, що описують загальну структуру ОК класичними методами із теорії керування;
- створення, збереження математичних моделей, що описують загальну структуру ОК із

використанням інтелектуальних технологій, зокрема нечіткого та нейромережевого математичного апарату;

- створення моделей регуляторів класичного (наприклад ПД) та інтелектуального (нечіткі, нейромережеві) типів;
- об'єднання моделей ОК, регуляторів й критеріїв оптимізації (КО) у так звані цільові структури оптимізації (ЦСО);
- вибір КО та створення власних критеріїв користувача;
- налаштування параметрів оптимізації керування ОК.

Цільові структури оптимізації (ЦСО) – у загальному сенсі це закінчені структурні елементи системи GenOp, що складаються з математичних описів та потребують пошуку власних оптимальних параметрів за певним критерієм, наприклад: глобальний мінімум функції багатьох змінних, навчання штучної нейромережі із невідомим вектором ваг, побудова нечіткої моделі із невідомими правилами та функціями приналежності тощо. Тобто ЦСО – це така математична структура, що обов'язково містить невідомий вектор (кластер, масив) параметрів, пошук яких виконує інтелектуальний оптимізаційний ГА. Оскільки системі GenOp функції оптимізаційного пошуку покладаються на ГА, тому в термінах еволюційного апарату ЦСО – це не що інше як фітнес-функція прилаштованості індивідів.

Система GenOp спроектовано на базі програмної платформи LabVIEW із використання графічної мови програмування G та допоміжного модулю Control Design & Simulation Toolkit (CDS). LabVIEW – повноцінний інструментальний пакет із широким математичним апаратом та редактором візуальних операторських панелей. Модуль CDS розширює стандартний набір функцій LabVIEW для проектування та моделювання автоматичних систем керування. GenOp складається із наступних головних частин (рис. 1): інтерфейсна підсистема (лицьова панель) та підсистема реалізацій (блок-діаграми). Лицьова панель GenOp дозволяє вводити наступні налаштування: початкові дані моделі ОК та її складових елементів, обирати алгоритм, структуру й параметри оптимізації (це ЦСО), обирати робочий критерій оптимізації, а також відобразити результати обрахунків у вигляді таблиць та графіків. Блок-діаграми підсистеми реалізації містять всі відкриті вихідні коди системи GenOp, а саме: модуль генетичної оптимізації, модуль критеріїв оптимізації та набір шаблонів із прикладами ЦСО. Відкритість вихідних кодів всіх блок-діаграм не обмежує функцій відлагодження та подальшого вдосконалення й модернізації GenOp. Зазвичай будь-яке ПЗ поділяється на окремі функціонально закінчені модулі, кожен з яких виконує окремий блок завдань та оперує лише певним набором даних, визначеного типу та об'єму для збереження у оперативній або постійній пам'яті. Модулі інтерфейсу та введення даних відокремлюються в окрему групу. Система GenOp має у складі наступні модулі:

- модуль «Генетична оптимізація»;
- модуль «Обрахунок критеріїв оптимізації»;
- модуль «Цільова структура оптимізації».

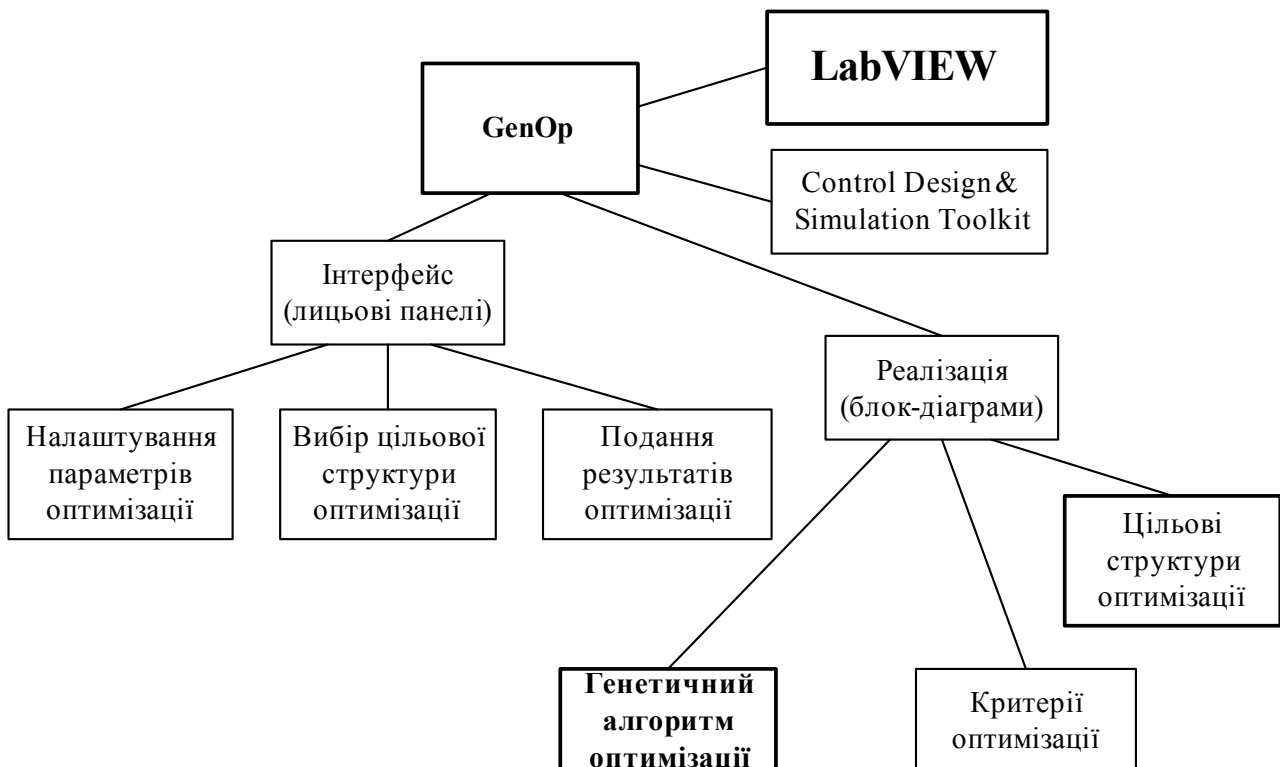


Рис. 1. Структура системи GenOp

Інтерфейс системи GenOp надає користувачу наступні можливості:

- введення початкових даних та налаштувань, необхідних для роботи системи;
- спостереження за динамікою оптимізаційних обрахунків на цифрових індикаторах та віртуальному осцилографі;
- запуск системи на виконання та його зупинка.

Програмний модуль «Генетична оптимізація» (ГО) спроектовано із використання відкритого некомерційного допоміжного модулю WARTIA, написаний в середовищі LabVIEW мовою G. Бібліотека WARTIA містить базові елементи реалізації ГА, необхідні для повноцінного функціонування системи GenOp. За замовчування WARTIA виконує оптимізаційний пошук глобального максимуму фітнес-функції, а GenOp розроблено для оптимізаційного пошуку глобального мінімуму.

Програмний модуль «Цільова структура оптимізації» (ЦСО) виконує обрахунок математичної моделі регулятора та моделі ОК об'єднаних у контур керування за умови наявності вхідних та вихідних даних. Програмний модуль «Критерій оптимізації» (КО) обраховує оптимізаційний алгоритм пошуку оптимуму (глобальний та/або локальні, мінімуму чи максимуму) враховуючи початкові дані про стан ОК та граничні обмеження.

Програмний модуль «Генетична оптимізація» складається із наступних частин: лицьова панель модулю та блок-діаграми ГА. Лицьова панель дозволяє вводити налаштування параметрів ГА, вибір режимів роботи ГА, а також відображати результати оптимізаційних обрахунків у вигляді таблиць та графіків. Блок-діаграми підсистеми реалізації містять всі відкриті вихідні коди програмного модулю «Генетична оптимізація», а саме: модуль ініціалізації індивідів, модуль обчислення фітнес-функції ЦСО, модуль селекції та схрещування а також модуль мутації індивідів. Відкритість вихідних кодів всіх блок-діаграм не обмежує функцій відлагодження та подальшого вдосконалення й модернізації програмного модулю «Генетична оптимізація».

До модулю ГО перед початком роботи системи GenOp підключається тільки один із модулів ЦСО, а модуль КО може бути додано і більше, що забезпечить подальший пошук оптимальних рішень за декількома критеріями одразу. ЦСО програмно відокремлені від головного модулю ГО, а модулі КО інтегровані безпосередньо у модуль ГО.

Узагальнений алгоритм роботи системи GenOp складається з кроків.

Крок 1. Початок роботи системи GenOp. Вибір критерію оптимізації. Серед реалізованих програмно КО обираються необхідні блок-діаграми та формується програмний фрагмент, який надалі використовуватиметься у ГО. Вводяться необхідні налаштування КО.

Крок 2. Обирається режим роботи системи GenOp, що здатна функціонувати у двох режимах: режим моделювання (статичний) та режим реального часу (динамічний). У випадку вибору режиму реального часу – перехід до кроку 6, інакше – крок 3.

Крок 3. Вибір ЦСО, що складається із регулятора та моделі ОК. Вибір виконується шляхом введення шляху та імені файлу віртуального інструмента із відповідним ЦСО.

Крок 4. Виконання модулю ГО. Перед початком роботи вводяться необхідні налаштування ГО, додається КО та підключається зовнішній файл із ЦСО.

Крок 5. Виконується перевірка завершення роботи ГО. У випадку досягнення цільового КО перехід до кроку 8, інакше – знову до кроку 4 доти, доки не буде досягнуто цільового КО, або доки не скінчиться лічильник ітерацій пошукового ГА.

Крок 6. Вибір ЦСО, що складається із регулятора та каналів апаратного зчитування та запису даних про стан реального ОК через плати/модулі введення/виведення. Вибір виконується шляхом введення шляху та імені файлу віртуального інструмента із відповідним ЦСО.

Крок 7. Виконується перевірка завершеності роботи ГО на кроці 5. У випадку завершеності оптимізації – перехід до кроку 9, інакше – до кроку 4 доти, доки не буде досягнуто цільового КО, або доки не скінчиться лічильник ітерацій пошукового ГА.

Крок 8. Збереження результатів ГО для подальшого використання в оптимальному керуванні у вигляді вектору оптимальних керувань та(або) оптимальних налаштувань регулятора. Кінець.

Крок 9. Для режиму роботи системи GenOp у реальному часі виконується постійне збереження поточного стану ОК в буферний стек. Розмір такого буферу обирається користувачем. У випадку досягнення межі переповнення, разом із записом стану ОК стеку, також починається зчитування найперших даних, що дозволяє організувати зсув історії (кадр) про поточний стан ОК у реальному часі.

Крок 10. Виконується перевірка якості керування шляхом порівняння поточної помилки моделі ОУ та відповідної помилки реального ОУ ГО. У випадку коли керування задовольняє – Кінець роботи GenOp, інакше – перехід до кроку 11.

Крок 11. Виконується зчитування поточного стану (кадру) ОК із стеку та знову перехід до кроку 4.

Параметрів ГА оптимізації досить багато. Звичайно, від значень цих параметрів, залежить ефективність цільового пошуку в залежності від типу розв'язуваних завдань. Існують деякі середні значення, за яких успішно вирішуються практично всі типи завдань оптимізаційного характеру.

Керуючі параметри модулю ГО:

- режим мутації: кожен біт інвертується з певною ймовірністю; інвертуються ті біти що збігаються з батьківськими хромосомами (координатами) з певною ймовірністю; мутація відсутня;
- обсяг буфера «розгону» для розміщення вихідної популяції;
- обсяги наступних популяцій (масивів);
- число ітерацій (поколінь, кроків оптимізації).

Робота ГА у модулі ГО полягає у наступну. Перед першим кроком потрібно випадковим чином створити початкову популяцію індивідів; навіть якщо вона виявиться абсолютно неконкурентоздатною, ймовірно, що ГА все одно достатньо швидко переведе її в життєздатну популяцію. Тож підсумком першого кроку алгоритму є популяція N , що складається з N особин. Розмноження в ГА зазвичай статеве – щоб створити нащадка (біологічний принцип), потрібні декілька батьків, зазвичай двоє. Розмноження в різних ГА визначається по-різному та іноді залежить від представлення даних. Головна вимога до розмноження – щоб нащадок чи нащадки мали можливість успадкувати риси обох батьків, «змішавши» їх будь-яким способом. До мутацій відноситься все те ж саме, що й до розмноження: є деяка частка мутантів m , що є параметром ГА, і на кроці мутацій потрібно вибрати $m*N$ особин, а потім змінити їх відповідно до заздалегідь визначених операцій (режимів) мутації. На заключному етапі відбору потрібно з усієї популяції вибрати певну її частку, яка залишиться «в живих» на цьому етапі еволюції. Є різні способи проводити відбір. Ймовірність виживання особини h повинна залежати від значення функції пристосованості *Fitness* (h). Сама частка тих, що вижили s зазвичай є параметром ГА, і її просто задають заздалегідь. За підсумками відбору з N особин популяції N повинні залишитися $s*N$ особин, які увійдуть до підсумкової популяції H . Решта особин гине.

Критерій оптимізації (КО) – деякий показник, значення якого визначає якість тестованого набору вхідних параметрів системи (ЦСО). Чим більше значення КО, тим краще оцінюється результат тестування з даними набором параметрів.

Тут необхідно зробити важливе зауваження: КО може використовуватися тільки в режимі тестування (статичний режим) з використанням ГА. Зрозуміло, що в режимі повного перебору всіх комбінацій значень параметрів, будь-який КО параметрів експерта в тестері відсутній. Інше питання, що ми можемо зберегти результати тестування і потім вже обробити їх на предмет підбору найкращої комбінації параметрів.

Реалізація КО на алгоритмічному рівні зводиться до забезпечення введення в ПЗ показників якості й обмежень, а також, за умови багатокритеріального пошуку, жорстку паритетність для випадку коли оптимальні обрахунки підкорятимуться принципу Парето. Прикладами КО, реалізованих у системі GenOp є критерій мінімізації помилки керування та критерій мінімізації енерговитрат на усіх рівнях ієрархії АСК.

Цільові структури оптимізації (ЦСО) – це завершені віртуальні інструменти (VI-файли), розроблені в LabVIEW, що містять у собі моделі замкнених контурів керування. ЦСО складаються здебільшого із блоку моделі ОК та регулятора для різного рівня складності моделей ОК та законів керування ними.

Можливість пакету LabVIEW дозволяють побудувати наступні види ЦСО як на моделях ОК так і на реальних об'єктах. Це досягається завдяки інтеграції у LabVIEW програмних засобів підтримки апаратного забезпечення (модулів введення-виведення) головного контролера CompactRIO із засобами вимірювання, регулювання та керування, встановлених безпосередньо на цих ОК.


Наведемо стислий перелік окремих елементів ЦСО, підтримка яких реалізована в системі GenOp:

- модель ОК й регулятор визначені аналітично та задані у формах: диференційного рівняння (системи) із початковими та граничними умовами, у вигляді передатної функції, матрицею простору станів (Control Design Toolkit);
- модель ОК й регулятор не визначені аналітично та побудована емпіричним способом різними методами ідентифікації (System Identification);
- модель ОК й регулятор побудовані у вигляді лінгвістичних моделей методами нечіткої логіки (Fuzzy Logic);
- модель ОК й регулятор побудовані інтелектуальними методами нейромережевого апарату (Neural Net);
- модель ОК й регулятор реалізовані в ЦСО засобами вбудованих функцій LabVIEW для зчитування-запису фізичних каналів введення-виведення контролеру CompactRIO.

Іншими словами, широкі можливості пакету LabVIEW й допоміжних Toolkit-ів дозволяють реалізувати ЦСО будь-якої складності для їх подальшого використання на апаратній платформі CompactRIO.

В межах роботи для моделювання, практичної реалізації та перевірки працездатності системи GenOp було використано та опрацьовано два типи ЦСО, а саме: на прикладі АСК із моделлю ОК коливального типу й класичного ПД-регулятора [4], а також із нейромережевим регулятором [5, 6]. Для цих ЦСО оптимізаційний алгоритм відшукує оптимальні параметри ПД- та нейромережевого регуляторів заданої структури.

2. Програмна реалізація генетичного оптимізатора

Відповідно до розробленого алгоритму функціонування ГО побудовано віртуальний інструмент ГО (рис. 2), що отримує із лицьової панелі системи GenOp початкові параметри налаштувань та шлях до файлу із ЦСО у вигляді кластеру (структури) об'єднаних даних. Розділені окремо кластерні дані подаються до блоку ГА – . Початкові масиви розв'язку та популяцій обнулено. Модуль ГО також містить цикл для оновлення даних на графіку, що відображає процес оптимізаційного пошуку за допомогою ГА. ГО здебільшого базується на підпрограмах віртуальних інструментів реалізації ГА. Відповідно до класичної теорії та алгоритму роботи ГА розроблено наступні модулі ГА блоку ГО для системи GenOp.

Модуль «Ініціалізація популяцій» реалізує створення (генерування) початкового набору даних про популяції індивідів для подальших генетичних мутацій в роботі ГА. Розмірність масиву задається на лицьовій панелі системи GenOp, закладка «Parameters».

Модуль «Сортування популяцій» виконує процедуру сортування поточних популяцій індивідів відповідно до постановки задачі й напряму оптимізаційного пошуку екстремуму. В системі GenOp реалізація оптимального пошуку відповідає задачі мінімізації. Тож сортування популяцій виконується за спаданням. Після процедури сортування популяції ранжуються. Модуль «Вибір батьківських хромосом» (селекція) займається вибором (по розрахованим на другому етапі значеннями фітнес-функції пристосованості) тих хромосом, які братимуть участь у створенні нащадків для наступної популяції, тобто для чергового покоління. Такий вибір здійснюється відповідно до принципу природного відбору, за яким найбільші шанси на участь у створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості. Існують різні методи селекції. В системі GenOp використовується метод рулетки (roulette wheel selection), який свою назву отримав за аналогією з відомою азартною грою. Кожній хромосомі може бути співставлений сектор колеса рулетки, величина якого встановлюється пропорційною значенню функції пристосованості даної хромосоми. Тому, чим більше значення функції пристосованості, тим більше сектор на колесі рулетки. Все колесо рулетки відповідає сумі значень функції пристосованості всіх хромосом даної популяції. Модуль «Кодування хромосом фенотипу у генотип» призначений для перетворення двійкових записів хромосом до еквівалентних числових значень. У класичному ГА застосовуються два основних генетичних оператора: оператор схрещування (crossover) і оператор мутації (mutation). Функцію crossover виконує модуль «Схрещення батьківських хромосом», а mutation – модуль «Мутації». Однак слід зазначити, що оператор мутації грає явно другорядну роль порівняно з оператором схрещування. Це означає, що схрещування в класичному ГА здійснюється практично завжди, тоді як мутація – досить рідко. Ймовірність схрещування, як правило, досить велика (зазвичай), тоді як ймовірність мутації встановлюється дуже малою (найчастіше). Це впливає з аналогії зі світом живих організмів, де мутації відбуваються надзвичайно рідко.

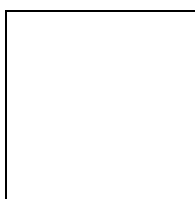


Рис. 2. Блок-діаграма віртуального інструменту генетичного оптимізатора

У ГА мутація хромосом може виконуватися на популяції батьків перед схрещуванням або на популяції нащадків, утворених в результаті схрещування.

На першому етапі схрещування вибираються пари хромосом з батьківської популяції (батьківського пулу). Це тимчасова популяція, що складається з хромосом, відібраних в результаті селекції і призначених для подальших перетворень операторами схрещування і мутації з метою формування нової популяції нащадків.

На даному етапі хромосоми з батьківської популяції об'єднуються в пари. Це проводиться випадковим способом відповідно до ймовірності схрещування. Далі для кожної пари відібраних таким чином батьків розігрується позиція гена (локус) у хромосомі, що визначає так звану точку схрещування. У результаті схрещування пари батьківських хромосом виходить наступна пара нащадків. Модуль «Декодування хромосом генотипу у фенотип» призначений для зворотного перетворення еквівалентних числових значень хромосом до їх двійкових записів (бінарний вигляд).

Модуль ГО у статичному режимі використовується для пошуку оптимальних керувань, оптимальних параметрів моделей ОК, регуляторів тощо без урахування реальної поведінки реального ОК в реальному часі.

Навіть отримавши оптимальний набір параметрів керування та налаштувань регулятора не існує стовідсоткової гарантії що у реальному часі на реальному ОК бездоганно виконуватиметься стратегія оптимального керування. Це зазвичай пов'язано із тим фактом, що з плином часу ОК переходять у певний життєвий цикл функціонування, в осередку якого оптимізаційний пошук та власне синтез оптимального керування взагалі не виконувався. Ще складніша ситуація виникає коли ОК апіорі є слабоформалізованим та не ідентифікованим з певними властивостями невизначеного характеру. Тож необхідність виконувати ГО у реальному часі в системах керування ОК є досить актуальною та нагальною проблемою.

В системі GenOp функціонал, що реалізує режим роботи ГО в реальному масштабі часу реалізований на базі стекової структури FIFO, яка зберігає фрагмент життєдіяльності ОК (кадр) за певний попередній проміжок часу. Глибина стеку – змінний параметр та налаштовується окремо. В процесі функціонування системи керування, у разі виходу стану ОК за межі оптимальних керувань, вмикається зчитування кадру ОК та перерахунок оптимальних параметрів ЦСО засобами модулю ГО.

Висновки

Розроблені концепція і програмний інструментарій засобами LabVIEW для реалізації та методики синтезу оптимальних програмних керувань GenOp дозволяють довести складність побудови систем керування, що вирішують оптимізаційні задачі в складних АСК до рівня інженера – розробника АСК, технолога чи оператора АСК. Система GenOp відкрита розробнику для подальшого вдосконалення та розширення функціональних можливостей. Наступні етапи розробки планується направити на реалізацію квантових алгоритмів для вирішення оптимізаційних задач в умовах нечіткої, невизначеної або взагалі відсутньої інформації про поточний стан ОК.

Література

1. Методы классической и современной теории автоматического управления: Учебник в 3-х т. Т.2: Оптимизация многообъектных многокритериальных систем. Под ред. Н.Д. Егупова. М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. 748 с.
2. Хайкин С. Нейронные сети: полный курс.: 2-е издание [пер. с англ.]. М.: Издательский дом «Вильямс», 2006. 1104 с.
3. Курейчик В.М. Генетические алгоритмы и их применение. Издательство ТРТУ, издание второе, 2002. 242 с.
4. Жеребко В.А., Кравець П.І., Василевська Х.С., Степанчук О.О. Підхід до побудови штучних нейронних мереж та їх реалізація на апаратній платформі CompactRIO. *Вісник вінницького політехнічного інституту*. 2009. № 3(84). С. 66–70.
5. Жеребко В.А., Онацький А.В., Соломонюк Ю.Р. Дослідження витрат ресурсів FPGA при побудові нейромережевої логічної функції XOR засобами LabVIEW. *Обчислювальний інтелект ОІ-2011: І Міжнар. наук. конф., 10-13 травня 2011 р.:* зб. пр. ЧДТУ, 2011. С. 165.
6. Жеребко В.А. Програмна модель генетичного оптимізатора ваг штучної нейронної мережі в LabVIEW. *Проблеми програмування*. 2013. № 2. С. 100–108.

References

1. Methods of classical and modern theory of automatic control: A textbook in 3 volumes. T.2: Optimization of multi-object multicriteria systems. Egupov N.D. // Moscow: MSTU them. N.E. Bauman, 2000. 748 p. (in Russian)
2. Haykin S. Neural Networks: A Comprehensive Foundation. Moscow: 2006. 1104 p. (in Russian)
3. Kureichik V.M. Genetic algorithms and their application // Publishing house of TRTU, second edition, 2002. 242 p. (in Russian)
4. Zhrebko V.A., Kravets P.I., Vasilevsky H.C., Stepanchuk O.O. An approach to the construction of artificial neural networks and their implementation on the hardware platform CompactRIO // Vinnytsia Polytechnic Institute. 2009. N 3 (84). P. 66–70. (in Ukrainian)
5. Zhrebko V.A., Onatsky A.V., Solomonyuk Yu.R. Investigation of the cost of FPGA resources when constructing the neural network logic function of XOR using LabVIEW // Computing Intellect, 2011: ChDTU, 2011. P. 165. (in Ukrainian)
6. Zhrebko V.A. Software model of the genetic optimizer of artificial neural network weights in LabVIEW. *Problems in programming*. (2). P. 100–108. (in Ukrainian).

Про авторів:

Жеребко Валерій Анатолійович,

старший викладач кафедри автоматичного управління в технічних системах Національного технічного університету України “КПІ імені Ігоря Сікорського”.
Кількість наукових публікацій в українських виданнях – понад 20.
Кількість наукових публікацій в зарубіжних виданнях – 2.
<http://orcid.org/0000-0001-6163-0258>,

Писаренко Олег Анатолійович,

студент кафедри автоматичного управління в технічних системах Національного технічного університету України “КПІ імені Ігоря Сікорського”.

Драбинко Віталій Павлович,

студент кафедри автоматики і управління в технічних системах
Національного технічного університету України “КПІ імені Ігоря Сікорського”.

Місце роботи авторів:

Національний технічний університет України “КПІ імені Ігоря Сікорського”.

03056, м. Київ-56, проспект Перемоги, 37, корпус 18.

Тел.: +38 (044) 236 7989.

E-mail: zherebko@kpi.ua,

oa.pisarenko@gmail.com,

v.drabynko@gmail.com.

