

## МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОЕКТУВАННЯ АДАПТИВНОГО РЕГУЛЯТОРА ЗАСОБАМИ LABVIEW

*В.А. Жеребко, Т.Й. Лукіна, А.С. Ляковський*

Національний технічний університет України «КПІ», 03056, Київ, проспект Перемоги, 37,  
корпус 18, к. 522,  
тел. +380(44) 406 8346, [zherebko@acts.kiev.ua](mailto:zherebko@acts.kiev.ua)

Представлена методика синтезу й моделювання адаптивної системи управління з еталонною моделлю та приклад програмної реалізації регулятора на платформі реального часу. Об'єктом керування обрано маятниковий механізм із пропелерним виконавчим пристроєм. Синтез та моделювання системи управління виконані в середовищі MATLAB. Проектування та програмна реалізація регулятора проведені мовою графічного програмування G, засобами середовища LabVIEW та додаткового модуля Control Design. Регулятор реалізовано на апаратно-програмній платформі CompactRIO.

This paper presented methods of synthesis and simulation of model reference adaptive control and an example of controller software implementation on real-time platform. As a control object is selected pendulum mechanism with propeller execution units. Synthesis and simulation of control systems implemented in the environment of MATLAB. Design and software implementation made graphical programming language G and LabVIEW environment by means of Control Design Toolkit. Regulator implemented in hardware and software CompactRIO platform.

### Вступ

Повний цикл проектування комп'ютеризованих інтелектуальних систем управління на базі освітніх технологій вищих навчальних закладів включає засвоєння теоретичного та практичного матеріалу навчально-методичного, програмного, системного, технічного та інформаційно-правового забезпечення. Теоретичне та практичне забезпечення навчальних дисциплін надається з урахуванням наскрізної системи навчання коли предметна область усіх дисциплін з певного напрямку доповнюється та поглиблюється від молодших до старших курсів навчання з плином часу. За такою системою надаються знання щодо теорії та практики проектування систем інтелектуального управління. Для проведення наукових досліджень і навчального процесу в області систем автоматичного управління розробляється лабораторне обладнання, навчальні стенди, дослідні макети, методичне, алгоритмічне та програмне забезпечення.

Адаптивні системи управління є складовою частиною інтелектуальних систем управління. Навчальний матеріал щодо синтезу та моделювання адаптивних систем управління викладаються в курсах «Теорія автоматичного управління» та «Цифрові системи». Програмні та системно-технічні навички відпрацьовуються в навчальних циклах «Проектування елементів систем управління» та «Проектування комп'ютеризованих систем управління».

У межах різних навчальних дисциплін з напрямку «Інтелектуальні системи управління» для синтезу та моделювання адаптивних систем управління здебільшого використовується програмний пакет MATLAB. А для проектування та практичної (схемно-технічної) реалізації окремих синтезованих регуляторних елементів системи управління використовується середовище графічного програмування LabVIEW. Тому постає важлива задача об'єднання навичок, що отримані при вивченні теорії синтезу, моделювання адаптивних систем управління (в MATLAB) та практичної програмно-апаратної реалізації синтезованих регуляторних елементів системи управління (в LabVIEW).

У даній роботі пропонується покрокове методичне забезпечення циклу проектування адаптивної системи управління та програмної реалізації адаптивного регулятора на простому прикладі, що включає об'єкт керування із виконавчим пристроєм, контролер управління та датчик зворотного зв'язку. Методика орієнтована на впровадження та використання в навчальному процесі з напрямку проектування систем управління.

### 1. Підхід до проектування та синтезу адаптивних систем управління

Програмне проектування адаптивних систем управління [1] з кінцевою реалізацією регуляторів на цільових керуючих пристроях можна умовно поділити на такі етапи:

- ідентифікація елементів системи управління у вигляді математичних моделей (об'єкт або процес управління, датчики, виконавчі пристрої, фільтри тощо);
- синтез структури та параметрів регуляторних елементів;
- імітаційне моделювання;
- реалізація програми регулятора та інших блоків системи;
- компіляція програмного коду у вигляді виконавчих файлів програми;
- завантаження програмного коду безпосередньо до цільової платформи;
- перевірка на працездатність спроектованої системи.

Загальна структура адаптивної системи управління складається з наступних складових елементів (рис. 1). Умовні позначення (рис. 1) наступні:

$U_c$  – вхідний сигнал завдання;

$y_{ем}$  – вихідний сигнал еталонної моделі;

$U$  – вхідний сигнал об'єкта керування й вихідний сигнал регулятора;

$y_{ок}$  – вихідний сигнал об'єкта керування.

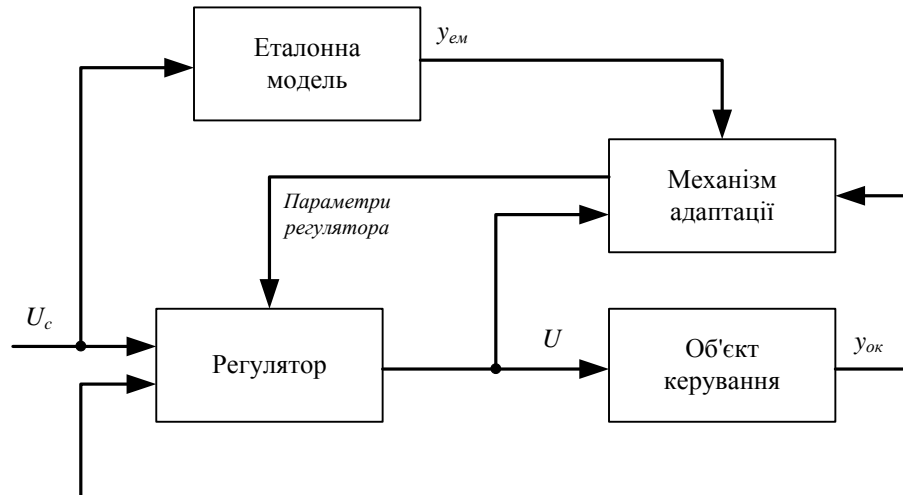


Рис 1. Структурна схема адаптивної системи управління з еталонною моделлю

Існує багато різноманітних методів для отримання бажаної перехідної характеристики відпрацювання завдання об'єктом керування, наприклад за допомогою регуляторного елемента. При проектуванні адаптивної системи управління з використанням еталонної моделі, необхідно безпосередньо вибрати еталонну модель об'єкта керування, структуру регулятора та елементи механізму адаптації (рис. 1).

Синтез адаптивної системи управління згідно з еталонною моделлю починається з визначення динамічної помилки  $e$ . Це різниця між сигналами виходів об'єкта керування та еталонної моделі:

$$e = y_{ок} - y_{ем}. \quad (1)$$

Виходячи з (1) формується вагова функція  $J(\theta)$ .  $J$  – функція від  $\theta$ .  $\theta$  – параметр, що буде налаштовуватись у регуляторі.

$$J(\theta) = \frac{1}{2} e^2(\theta). \quad (2)$$

Для того щоб визначити як змінюється параметр  $\theta$ , рівняння має бути побудоване за змінною  $\theta$ .

$$\frac{d\theta}{dt} = -\gamma \frac{\delta J}{\delta \theta} = -\gamma e \frac{\delta e}{\delta \theta}. \quad (3)$$

В'язок між змінною  $\theta$  та ваговою функцією виглядає наступним чином:

$$J(\theta) = |e(\theta)|, \quad \frac{d\theta}{dt} = -\gamma \frac{\delta e}{\delta \theta} \text{sign}(e), \quad (4)$$

де  $\text{sign}(e) = \begin{cases} 1, e > 0 \\ 0, e = 0 \\ -1, e < 0 \end{cases}$ .

Використавши визначення вагової функції (2), виведемо закон управління.

Для системи  $\frac{Y(s)}{U(s)} = kG(s)$ , де  $k$  невідоме. Потрібно, щоб вираз набув вигляду  $\frac{Y(s)}{U_c(s)} = k_0 G(s)$ ,

використовуючи об'єкт керування  $G_{ем}(s) = k_0 G(s)$ .

Далі вибираємо вагову функцію

$$J(\theta) = \frac{1}{2} e^2(\theta) \longrightarrow \frac{d\theta}{dt} = -\gamma e \frac{\delta e}{\delta \theta}.$$

Запишемо рівняння для помилки

$$e = y - y_{em} = kGU - G_{em}U_c = kG\theta U_c - k_0GU_c.$$

Підрахунок похідних «чутливості» виглядає наступним чином:

$$\frac{\partial e}{\partial \theta} = kGU_c = \frac{k}{k_0} y_{em}.$$

Застосуємо (3) та отримаємо алгоритм адаптивного управління:

$$\frac{d\theta}{dt} = -\gamma' \frac{k}{k_0} y_{em} e = -\gamma' y_{em}.$$

## 2. Опис конструкції об'єкта керування

Об'єктом керування виступає маятникова система (рис. 2), яка складається з бруска, що закріплений у вертикальній площині на краю нерухомого стола. Точка кріплення з'єднана з датчиком положення, виходячи з чого можуть бути виміряні кут та кутова швидкість. На нижньому кінці бруска прикріплений двигун постійного струму й пропелер. При подачі напруги на двигун, пропелер починає обертатися та піднімати брусок. Завдання полягає в тому, щоб підняти брусок на необхідний кут.

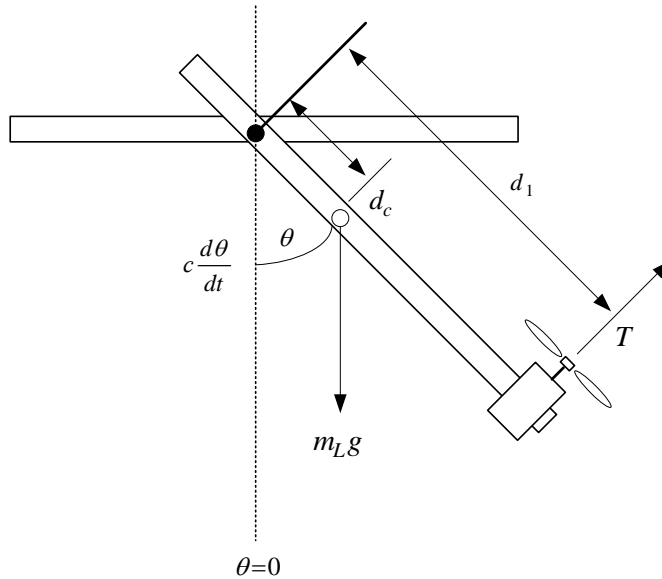


Рис. 2. Схема об'єкта керування з маятниковим механізмом

Умовні позначення (рис. 2) наступні:

$T$  – сила тяги пропелера;  $d_1$  – відстань від точки кріплення до осі пропелера;  $d_c$  – відстань від точки кріплення до центру хитання;  $\theta$  – кут відхилення бруска;  $m_L g$  – сила тяжіння, що діє в центрі хитання;  $c$  – добуток коефіцієнта тертя, що діє в точці кріплення бруска на величину  $d_1$ .

## 3. Математична модель об'єкта керування

Для опису диференційних рівнянь кутового руху маятника скористаємось законом Ньютона та законом збереження кутового моменту [2]:

$$J \ddot{\theta} + c \dot{\theta} + mgd_c \sin \theta = (d_1)T, \quad (5)$$

де  $mgd_c$  – момент сили тяжіння щодо осі коливання маятника.

Відповідно, безперервна передатна функція об'єкта керування набуде вигляду:

$$\frac{\theta(s)}{T(s)} = \frac{d_1}{Js^2 + cs + mgd_c}. \quad (6)$$

Якщо підставити в (6) фізичні розміри бруска (довжина  $d_1=1,89$  м,  $d_c=0,63$  м) та номінали (вага  $m=1,74$  кг, коефіцієнт тертя – 0,02), тоді результуюча передатна функція об'єкта керування виглядатиме наступним чином:

$$G_p = \frac{\theta(s)}{T(s)} = \frac{1.89}{s^2 + 0.0389s + 10.77}. \quad (7)$$

Запишемо математичний вираз для сигналу управління, що подається на об'єкт керування:

$$\begin{aligned} u &= \theta_1 u_c - \theta_2 y_{ок}, \\ e &= y_{ок} - y_m = G_p u - G_m u_c, \\ y_{ок} &= G_p u = \left( \frac{1.89\theta_1}{s^2 + 0.0389s + 10.77} \right) (\theta_1 u_c - \theta_2 y_{ок}), \\ y_{ок} &= \frac{1.89\theta_1}{s^2 + 0.0389s + 10.77 + 1.89\theta_2} u_c. \end{aligned} \quad (8)$$

Виходячи з (8) помилка виглядає наступним чином:

$$e = \frac{1.89\theta_1}{s^2 + 0.0389s + 10.77 + 1.89\theta_2} u_c - G_m U_c. \quad (9)$$

Взявши часткову похідну від помилки за  $\theta_1$  та  $\theta_2$ , отримаємо похідні «чутливості»:

$$\begin{aligned} \frac{\partial e}{\partial \theta_1} &= \frac{1.89}{s^2 + 0.0389s + 10.77 + 1.89\theta_2} u_c, \\ \frac{\partial e}{\partial \theta_2} &= -\frac{1.89^2 \theta_1}{(s^2 + 0.0389s + 10.77 + 1.89\theta_2)^2} u_c, \\ \frac{\partial e}{\partial \theta_2} &= -\frac{1.89\theta_1}{s^2 + 0.0389s + 10.77 + 1.89\theta_2} y_{ок}. \end{aligned} \quad (10)$$

Вищеотримані похідні «чутливості» містять у собі параметри об'єкта керування, але необхідно пам'ятати, що сигнал завдання  $U_c$  не включає цих параметрів. За умовами моделювання передбачено що характеристики об'єкта керування невідомі. Завдання полягає у наближенні характеристик об'єкта керування до його еталонної моделі. Якщо еталонна модель близька до об'єкта керування, її характеристики можуть бути описані згідно об'єкта керування. Внаслідок отримаємо наступні похідні «чутливості»:

$$s^2 + 0.0389s + 10.77 + 1.89\theta_2 \approx s^2 + a_{1m}s + a_{0m}, \quad (11)$$

$$\frac{\partial e}{\partial \theta_1} = \frac{a_{1m}s + a_{0m}}{s^2 + a_{1m}s + a_{0m}} u_c, \quad (12)$$

$$\frac{\partial e}{\partial \theta_2} = \frac{a_{1m}s + a_{0m}}{s^2 + a_{1m}s + a_{0m}} y_{ок}. \quad (13)$$

Математична модель алгоритмів адаптації виглядає наступним чином:

$$\frac{d\theta_1}{dt} = -\gamma \frac{\partial e}{\partial \theta_1} e = -\gamma \left( \frac{a_{1m}s + a_{0m}}{s^2 + a_{1m}s + a_{0m}} u_c \right) e, \quad (14)$$

$$\frac{d\theta_2}{dt} = \gamma \frac{\partial e}{\partial \theta_2} e = \gamma \left( \frac{a_{1m}s + a_{0m}}{s^2 + a_{1m}s + a_{0m}} y_{ок} \right) e. \quad (15)$$

Необхідно підкреслити, що вхідні та вихідні сигнали об'єкта керування попередньо фільтруються. Передавальна функція еталонної моделі має вигляд:

$$G_m = \frac{a_{0m}}{s^2 + a_{1m}s + a_{0m}}. \quad (16)$$

#### 4. Моделювання та синтез адаптивної системи управління з еталонною моделлю

Моделювання систем управління виконується в програмному середовищі MATLAB. Рекомендації та практичні приклади використання цього пакету наведені в [3].

Для моделювання системи адаптивного керування вибрано коефіцієнт  $\gamma = 0.0001$  у передавальних функціях адаптивного алгоритму. Передавальна функція еталонної моделі вибрана зі врахуванням наступних параметрів: час встановлення  $T_s = 3$  с та коефіцієнт згасання  $\zeta = 0.707$ .

Розрахуємо параметри передавальної функції еталонної моделі:

$$a_{0m} = \left( \frac{4}{T_s \zeta} \right)^2 = 3.56, \quad a_{1m} = 2\zeta \left( \frac{4}{T_s \zeta} \right) = 2.67.$$

Підставивши ці дані в (15), (16), отримаємо безперервні передавальні функції, що наведені в табл. 1. «Реальним» об'єктом керування вибрана передавальна функція зі змінним коефіцієнтом підсилення (в нашому випадку 4.41), що відрізняється від наведеного в (7). Він враховує перетворення завдання кута (в градусах) в напругу на двигун пропелера та перетворення значення датчика кута з радіан на градуси.

Таблиця 1. Неперервні передатні функції адаптивної системи управління

№	Елементи системи	Неперервна передатна функція
1	Еталонна модель маятника	$\frac{3.56}{s^2 + 2.67s + 3.56}$
2	Фільтр (1, 2)	$\frac{2.67s + 3.56}{s^2 + 2.67s + 3.56}$
3	Алгоритм адаптації (Тета1, Тета 2)	$-\frac{0.0001}{s}, \frac{0.0001}{s}$
4	ПД-регулятор	$1 + 1.5s$
5	Об'єкт керування	$\frac{4.41}{s^2 + 0.039s + 10.77}$

На рис. 3 представлена схема адаптивної системи управління, яка була змодельована в програмному пакеті MATLAB (Simulink) з передатних функцій (табл. 1). Дана схема має два основних контури: контур з еталонною моделлю і адаптивний контур. Для завдання тактових імпульсів слугує генератор тактових імпульсів, що позначено на схемі як «ІМПУЛЬСИ». Далі в системі йде блок «Gain» (позначений на схемі «КУТ\_3(град.)»), який необхідний для завдання потрібного кута відхилення бруска маятника.

В контур з еталонною моделлю входить блок «Transfer function», названий в схемі «ЕТАЛОННА МОДЕЛЬ». Він слугує для завдання еталонних характеристик моделі, які в кінці будуть порівнюватись з реальною передатною характеристикою об'єкта керування. До регуляторного контура входить ПД-регулятор, позначений на рис. 3 сірим кольором.

Адаптивний контур складається з блоків «Transfer function», позначені в схемі TETA1 і TETA2, відповідно. Для переведення значення градуса в напругу слугує блок «Gain», позначено (рис. 3) «град. -> V». Для обмеження напруги слугує блок «Saturation», в схемі позначений – «обмеження 0-10 V».

Об'єкт керування в схемі описано за допомогою одного блоку «Transfer function», який у схемі має назву «ОБ'ЄКТ КЕРУВАННЯ». Для переведення градусів в радіани в системі присутній блок «Gain», який названий «180/3.14». Також у схемі присутні фільтри для зменшення паразитних завад сигналу від датчика кута.

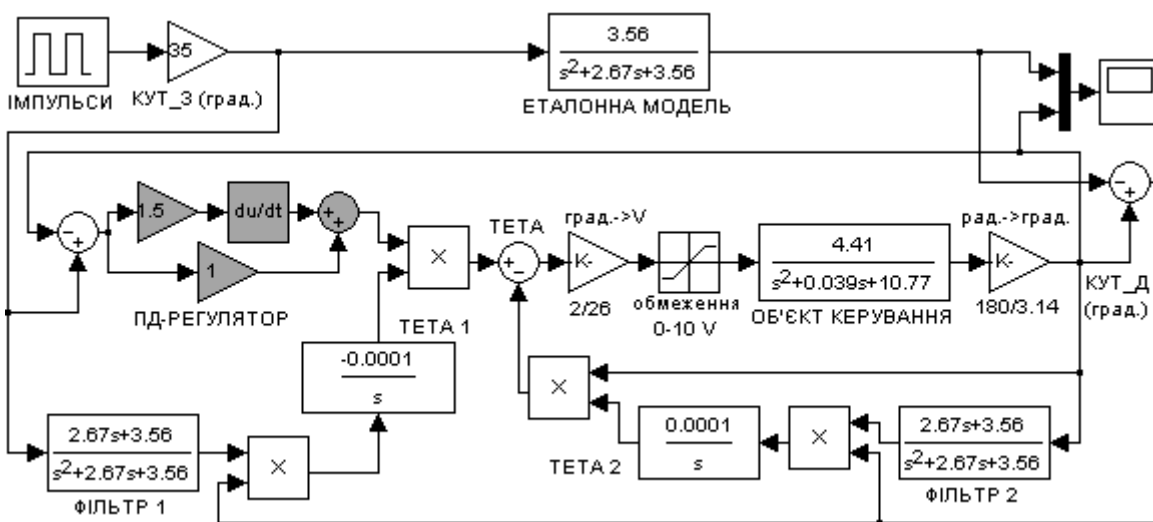


Рис. 3. Адаптивна система управління з еталонною моделлю в MATLAB (Simulink)

Результати імітаційного моделювання системи управління виводяться на осцилограф «Scope» (рис. 4). На рис. 4, а) можна спостерігати часові характеристики відпрацювання установки завдання об'єктом керування (вихід об'єкта керування містить нестійкі коливання) та еталонною моделлю (крива зі стійкою перехідною характеристикою) без регулятора. На рис. 4, б) наведені результати моделювання з ПД-регулятором (вихід об'єкта керування має стійку характеристику).

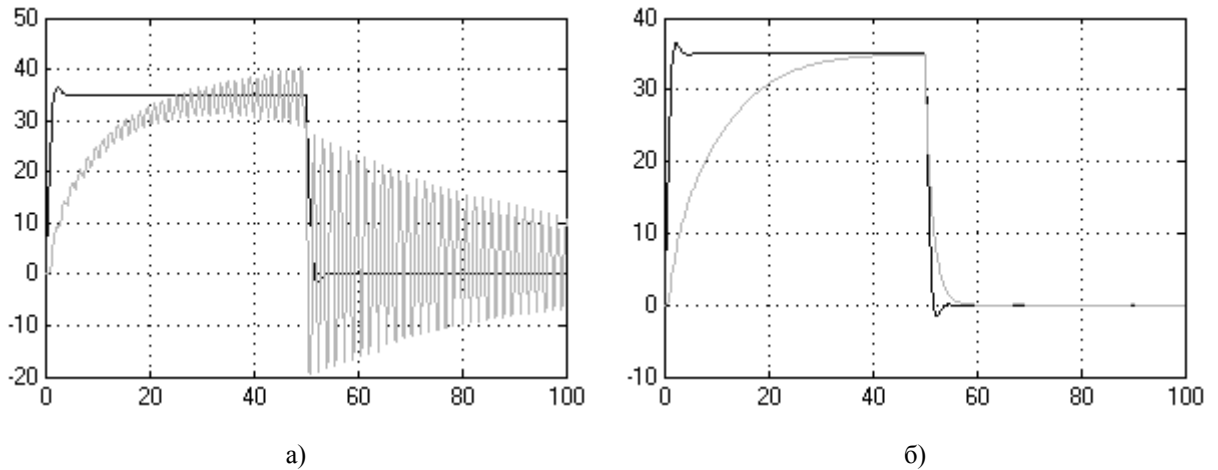


Рис. 4. Характеристики відпрацювання завдання (35 град.) об'єктом керування:  
а) – без ПД-регулятора; б) – з ПД-регулятором

### 5. Програмна реалізація адаптивного регулятора на цільовій апаратній платформі

Результати імітаційного моделювання адаптивної системи управління використані для подальшої програмної реалізації регуляторного пристрою на цільовій апаратній платформі: В межах даної роботи цільовою апаратною платформою обрано вільно-програмований контролер автоматизації реального часу CompactRIO (виробник National Instruments) з операційною системою реального часу (ОСРЧ) Phar Lap ETS.

Програмування платформи CompactRIO не відрізняється від програмування стандартної ПК-платформи. Різниця полягає у завантаженні виконавчих файлів безпосередньо на віддалену платформу CompactRIO мережевим інтерфейсом LAN для подальшого запуску на виконання. Методичні та практичні рекомендації щодо програмування платформи CompactRIO наведені в [4].

Модель системи управління (рис. 3) складається з передатних функцій неперервного типу. Цільова апаратна платформа, що є цифровим обчислювальним пристроєм реального часу, виконує будь-які програми користувача в дискретному часі. Тому необхідно передатні функції синтезованих елементів регуляторного пристрою (інших допоміжних елементів) представити у вигляді дискретних передатних функцій.

Всі операції перетворення неперервних передатних функцій в дискретні виконані за допомогою середовища MATLAB та вбудованих стандартних функцій  $tf()$  та  $c2d()$ . Функція  $tf()$  слугує для ініціалізації неперервної передатної функції, а  $c2d()$  – для перетворення неперервної передатної функції в дискретну форму. Всі перетворення виконувались з часом дискретизації – 0,01 сек. Вигляд усіх передатних функцій та їх аналогів у дискретній формі наведені в табл. 2.

Таблиця 2. Розрахунок дискретних передатних функцій адаптивної системи управління

№	Елементи системи	Команда в MATLAB	Дискретна передатна функція
1	Еталонна модель маятника	<code>sys1 = tf([0 3.56],[1 2.67 3.56])</code> <code>c2d(sys1,0.01)</code>	$\frac{0.0001764z + 0.0001749}{z^2 - 1.973z + 0.9737}$
2	Фільтр (1, 2)	<code>sys2 = tf([2.67 3.56],[1 2.67 3.56])</code> <code>c2d(sys2,0.01)</code>	$\frac{0.02652z + 0.02167}{z^2 - 1.973z + 0.9737}$
3	Алгоритм адаптації (Тета1, Тета2)	<code>sys3 = tf([0.0001],[1 0])</code> <code>c2d(sys3,0.01)</code>	$-\frac{1 \cdot 10^{-6}}{z-1}, \frac{1 \cdot 10^{-6}}{z-1}$
4	ПД-регулятор	<code>sys4 = tf([1.5 1],[0 1])</code> <code>c2d(sys4,0.01,'tustin')</code>	$\frac{301z - 299}{z + 1}$

Реалізацію програми регуляторного пристрою виконано в середовищі графічного програмування LabVIEW мовою G у вигляді віртуального пристрою VI. Програмний інструментарій LabVIEW є розробкою компанії National Instruments, що дозволяє структурно та інтуїтивно зрозуміло проектувати інженерні рішення широкого класу задач, від систем вимірювання та діагностики до систем промислового управління та автоматизації надскладними технічними об'єктами. Опис мови та середовища програмування LabVIEW викладені в [5].

Складовою частиною платформи CompactRIO є реконфігуроване шасі на базі FPGA-контролера. Всі модулі введення-виведення підключені в напрямі до входів FPGA-контролера. Для зчитування апаратних (фізичних) сигналів з модулів введення-виведення необхідне окреме програмування FPGA-контролера.

Зчитування сигналів можна виконувати в режимі «Scan Mode», коли до FPGA завантажуються стандартні функції зчитування (реалізовано лише в ОСРЧ VxWorks для CompactRIO). У нашому випадку ОСРЧ не підтримує такої можливості, тому в FPGA-контролер завантажено двійковий код скомпільованої програми (створена засобами LabVIEW) зчитування / запису даних з модулів контролера CompactRIO.

Програма адаптивного регулятора засобами LabVIEW, як і будь-яка прикладна програма, складається з типових елементів мови програмування високого рівня, а саме: змінних, математичних операцій, операторів вибору, циклів тощо. Для роботи з елементами систем управління (наприклад, дискретна передатна функція), окрім стандартного пакету LabVIEW, необхідна також наявність додаткового програмного модуля Control Design [6], що безпосередньо інтегрується в середовище LabVIEW. Окрім «коду» на мові G в LabVIEW також створюється так звана «Лицьова панель» для візуалізації складових елементів та роботи програми в цілому.

Програмний код адаптивного регулятора (рис. 5) складається з безкінечного циклу, який зупиняється кнопкою «СТОП». У тілі циклу розміщені та поєднані між собою блоки, що відповідають дискретним передатним функціям синтезованих елементів адаптивного регулятора.

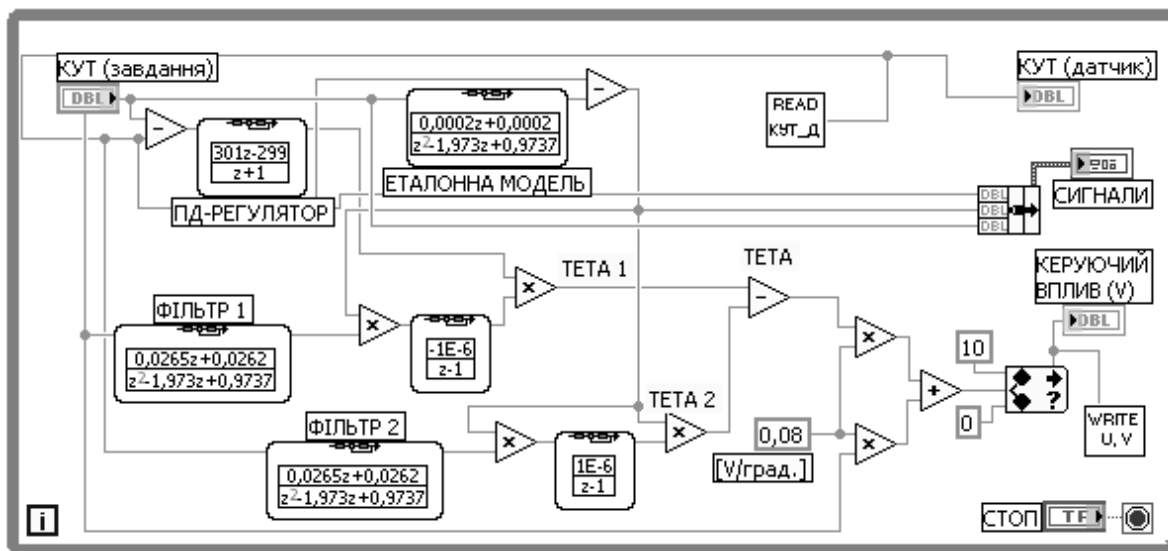


Рис. 5. Блок-діаграма реалізації адаптивного регулятора в LabVIEW (на CompactRIO)

Миттєве значення кута (з датчика) зчитується з дискретного входу (лічильника) модуля введення та перетворюється на градуси (блок «READ КУТ\_Д», рис. 5). Керуючий вплив «ТЕТА» переводиться з градусів в напругу (шляхом множення на коефіцієнт  $2/26=0.08$ ) і записується до модуля виведення аналогових сигналів в діапазоні 0..10В, а далі подається на фізичний об'єкт керування (блок «WRITE U,V»; рис. 5).

Візуалізація процесу роботи адаптивної системи управління маятниковим механізмом із застосуванням адаптивного регулятора з еталонною моделлю виконано на «Лицьовій панелі» віртуального пристрою VI. Ручне завдання (користувачем) кута, миттєве значення кута (з датчика), а також графіки результату роботи системи показані на рис. 6.

Запроектований адаптивний регулятор функціонує у реальному масштабі часу в контролері CompactRIO під керівництвом ОСРЧ та LabVIEW REAL-TIME. Тестування роботи виконувалося шляхом зміни початкового кута положення маятника (бруска) щодо осі та завдання необхідного кута положення.

На рис. 6 можна спостерігати позитивне відпрацювання маятником заданого кута. Завдяки адаптивному регулятору управління проходить достатньо стійко без паразитних та хаотичних коливань.

## Висновки

У роботі представлено покрокову методику синтезу, моделювання, проектування та практичної програмної реалізації адаптивного регулятора з еталонною моделлю. Моделювання, синтез та проміжні математичні розрахунки проводилися засобами пакета MATLAB. Програмна реалізація адаптивного алгоритма та допоміжних фільтрів виконані засобами графічної мови програмування G в середовищі LabVIEW.

У розглянутій нестійкій системі управління маятниковим механізмом адаптивний алгоритм було синтезовано й реалізовано з застосуванням еталонної моделі об'єкта та за допомогою ПД-регулятора, що дозволив виключити паразитні коливання при стабілізації заданого кута положення маятника.

Адаптивний регулятор інтегровано на цільову платформу вільно-програмованого контролера автоматизації реального часу CompactRIO та проведено тестування роботи адаптивної системи управління кутом положення маятника.

Результати роботи можуть бути впроваджені в навчальний процес з напрямку «Інтелектуальні системи управління» в наступні навчальні дисципліни: «Моделювання систем», «Теорія автоматичного управління», «Проектування комп'ютеризованих систем управління».

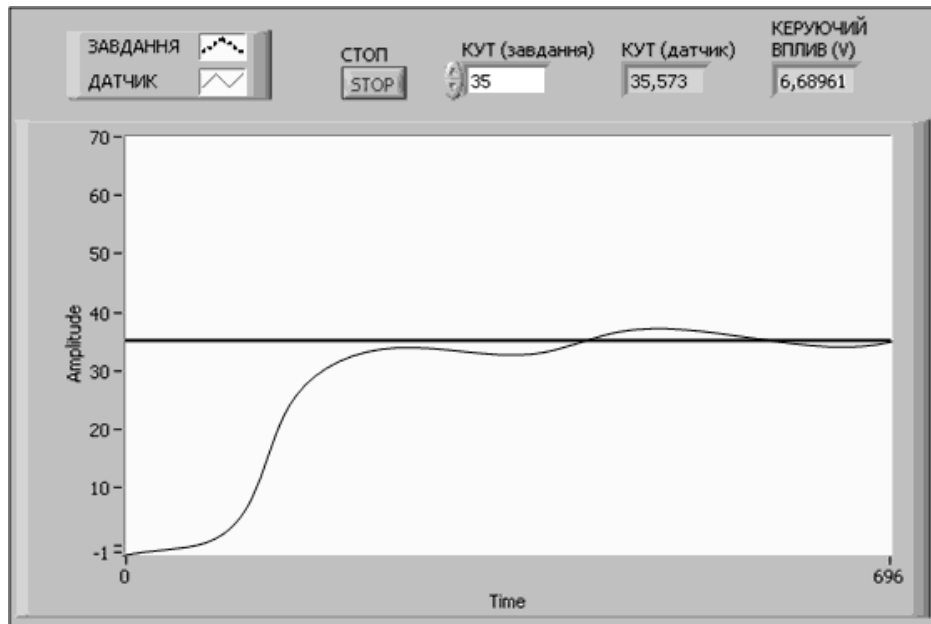


Рис. 6. Демонстрація роботи адаптивної системи управління об'єктом в LabVIEW

1. *Александров А. Г.* Оптимальные и адаптивные системы: учеб. пособие для вузов по спец. «Автоматика и упр. в техн. системах». – 2-е изд. – М.: Высш. школа, 2003. – 262 с.
2. *Control the Position of a Robot Arm Naturally Vibrates by Model Reference Adaptive Control (MRAC)* [Електронний ресурс]. – Режим доступу: <http://www.pages.drexel.edu/~kws23/tutorials/MRAC/MRAC.html>.
3. *Дьяконов В. П.* MATLAB 7.\*/R2006/R2007: Самоучитель. – М.: ДМК Пресс, 2008. – 768 с.
4. *LabVIEW и CompactRIO: основы разработки приложений.* / National Instruments, 2008. – 305 с.
5. *Трэвис Дж., Кринг Дж.* LabVIEW для всех [Текст] / Дж. Трэвис. – 3-е изд. – М.: ДМК Пресс, 2008. – 880 с.
6. *NI LabVIEW Control Design Toolkit. User Manual* [Електронний ресурс]. – Режим доступу: <http://www.ni.com/pdf/manuals/371057g.pdf>.