

## ПРОГРАМНА РЕАЛІЗАЦІЯ ІМПЕДАНСНОГО PI-РЕГУЛЯТОРА СИСТЕМИ КЕРУВАННЯ ДУГОВОЇ СТАЛЕПЛАВИЛЬНОЇ ПЕЧІ НА МОВІ GO

**Анотація.** Керування електродами дугової сталеплавильної печі (ДСП) залишається складною інженерною задачею через нелінійну та стохастичну природу електричної дуги. Нестабільність дуги безпосередньо впливає на питоме енергоспоживання та рівномірність проплавлення шихти, що критично для металургійних підприємств з безперервним циклом виробництва. У цій роботі розглянуто підхід до програмної реалізації імпедансного PI-регулятора системи керування електродами ДСП із застосуванням мови програмування Go.

В основу алгоритму покладено дискретну форму PI-регулятора з обмеженням інтегратора (*anti-windup*), що дозволяє уникнути насичення при різких змінах завдання. Математична модель дуги побудована на базі рівнянь Кассі-Майра, адаптованих для дискретного часового кроку. Програмна реалізація оформлена у вигляді модульної структури мовою Go з окремими компонентами для розрахунку регулятора, циклу моделювання та збереження результатів у форматі CSV.

За результатами моделювання встановлено, що система виходить на заданий імпеданс протягом 0,15-0,2 с при перерегулюванні, що не перевищує допустимих меж. Зафіксована залишкова статична похибка в межах 2-3% пояснюється спрощеннями прийнятої моделі дуги та може бути усунена шляхом введення адаптивного коригування коефіцієнтів. Отримані результати підтверджують придатність Go як інструменту для реалізації алгоритмів керування у складі промислових програмних систем.

**Ключові слова:** дугова сталеплавильна піч, імпедансне регулювання, PI-регулятор, математичне моделювання, електрична дуга, мова Go, система керування, автоматизація.

**Постановка проблеми.** Підвищення енергоефективності дугових сталеплавильних печей залишається актуальною задачею для металургійного виробництва, що зумовлено насамперед нестабільною поведінкою електричної

дуги. Дуга чутлива до зміни складу шихти, геометрії міжелектродного проміжку та температурних умов – її параметри можуть суттєво змінюватися протягом короткого часу. Наслідком цього є коливання струму і напруги, зростання питомих витрат електроенергії, підвищений знос електродів та погіршення технологічних показників плавки.

Регулювання положення електродів є основним інструментом стабілізації режиму горіння дуги. Однак між електричною підсистемою дуги та механічною підсистемою електрогідравлічного приводу існує принципова різниця в динаміці: електрична дуга реагує на збурення за мілісекунди, тоді як привід переміщення електрода має значно більшу інерційність. Ця асиметрія часових характеристик ускладнює синтез регулятора і вимагає обґрунтованого вибору його структури та параметрів.

У промисловій практиці поширеним підходом є імпедансне регулювання, при якому керуючий сигнал формується на основі повного опору дуги. На відміну від регулювання за струмом або напругою окремо, такий підхід враховує обидва електричних параметри одночасно, що забезпечує вищу точність і стійкість керування [8]. Додатковим чинником, що ускладнює задачу, є виражена нелінійність об'єкта керування: динаміка дуги нестаціонарна, містить швидкоплинні перехідні процеси, а її параметри суттєво залежать від поточного режиму роботи печі – зокрема від стадії плавки та ступеня проплавлення шихти. Саме ця нелінійність унеможлиблює пряме застосування класичних методів синтезу і вимагає використання математичних моделей, що адекватно відтворюють реальну поведінку дуги.

У більшості наявних досліджень задача синтезу та верифікації регуляторів для ДСП вирішується із застосуванням спеціалізованих пакетів – зокрема MATLAB/Simulink. Такі засоби є ефективними на етапі аналізу, однак мають суттєве практичне обмеження: отримані моделі складно перенести у реальне промислове середовище без повного переписування коду. Це стримує впровадження розроблених алгоритмів керування у виробничі системи автоматизації.

Зазначена проблема визначає доцільність використання універсальних мов програмування для реалізації алгоритмів керування. Мова Go у цьому контексті має низку практичних переваг: компіляція у нативний виконуваний

код, відсутність залежності від зовнішніх середовищ виконання, підтримка модульної архітектури та висока продуктивність [18, 19]. Незважаючи на це, питання реалізації імпедансних PI-регуляторів для систем керування ДСП засобами Go залишається практично не дослідженим у відкритих джерелах, що й визначає мету даної роботи.

**Аналіз останніх досліджень і публікацій.** Інтерес до математичного моделювання та автоматичного керування дуговими сталеплавильними печами не вщухає вже кілька десятиліть, що зумовлено рядом причин: зростання цін на електроенергію, посилення вимог до якості сталі та нестабільна поведінка електричної дуги, яка досі не піддається повністю детермінованому опису.

Відправною точкою для більшості досліджень у цій галузі слугує фундаментальна праця Fruehan [1], де систематизовано фізико-хімічні основи сталеплавильних процесів і показано, що якість металу безпосередньо залежить від того, наскільки точно підтримуються електричні режими: напруга, струм і потужність дуги. По суті, саме ця робота сформулювала вимоги до систем керування, які інші автори намагалися реалізувати практично.

Огляд Hay, Visuri, Aula та Echterhof [2] дає уявлення про те, наскільки різноманітними стали підходи до моделювання ДСП на сьогодні: термодинамічні, електричні, теплообмінні та гібридні моделі вирішують різні підзадачі, однак жодна з них окремо не охоплює повну картину процесу. Автори прямо вказують: адекватна модель має враховувати взаємодію електричної, теплової та механічної підсистем одночасно. Цю думку розвивають Abadi, Tang та Rashidi [3], які зосередились на чисельних методах – зокрема CFD та методі скінченних елементів – для відтворення електромагнітних і теплових полів усередині печі.

Окремо варто зупинитися на моделях динаміки самої дуги. Класична робота Cassie [4] запропонувала рівняння для опису зміни провідності дуги – підхід, який, попри свою давність, досі використовується як основа. Golestani та Samet [5] пішли далі й поєднали модель Кассі з моделлю Майра в єдину гібридну конструкцію, яка точніше відтворює поведінку дуги як у сталих, так і в перехідних режимах. Практичне застосування цих ідей знайшло відображення у роботі Blažič, Škrjanc та Logar [6]: автори побудували трифазну

модель ДСП і запропонували індекс якості дуги – числовий показник її стабільності, що обчислюється в реальному часі за електричними вимірюваннями.

Щодо безпосередньо керування електродами, тут картина неоднорідна. Balan, Hancu та Luru [7] розробили методи адаптивного керування, які враховують нелінійність об'єкта – зміну характеристик шихти в ході плавки. Wang, Mao, Tian, Li та Yuan [8] переконливо показали переваги імпедансного регулювання: підтримання заданого співвідношення напруги до струму дуги дає стабільніший результат, ніж регулювання за окремими електричними величинами. Logar та Škrjanc [9] підійшли до питання з боку енергетики – їх цікавило, як налаштування регулятора впливає не лише на процес плавки, але й на електромагнітну сумісність із живильною мережею. Математичну базу для формалізації всіх цих задач заклали ще Billings, Boland та Nicholson [10] – їх підхід до побудови моделі об'єкта і вибору структури регулятора залишається актуальним і в сучасних системах автоматизації.

Нарешті, Boulet, Lalli та Ajersch [11] зосередились на тому, що нерідко лишається поза увагою теоретиків – на практичній реалізації алгоритмів керування. І тут виявляється характерна прогалина: переважна більшість публікацій або обмежується теоретичним аналізом, або прив'язана до спеціалізованих середовищ на кшталт MATLAB/Simulink, що суттєво ускладнює перенесення розроблених рішень у реальні промислові системи.

Саме цим і пояснюється постановка завдання в даній роботі: розробити програмну реалізацію імпедансного PI-регулятора засобами універсальної мови програмування Go – такої, що одразу придатна для інтеграції у виробниче середовище без прив'язки до дорогих чи вузькоспеціалізованих платформ.

**Мета дослідження.** Метою даного дослідження є розробка та програмна реалізація імпедансного PI-регулятора системи керування переміщенням електродів дугової сталеплавильної печі з використанням мови програмування Go. У межах роботи передбачається використання математичної моделі як основи для формування алгоритму керування, спрямованого на стабілізацію імпедансу дуги з урахуванням динаміки електричної дуги та особливостей електромеханічної системи.

Особлива увага приділяється дослідженню впливу параметрів PI-регулятора на динамічні характеристики системи, зокрема точність регулювання, швидкодію та наявність статичної похибки. У процесі дослідження передбачається проведення чисельних експериментів, отримання часових залежностей основних параметрів системи та їх подальший аналіз.

Досягнення поставленої мети має забезпечити обґрунтування ефективності використання мови Go для реалізації алгоритмів керування та створення програмних засобів, придатних для інтеграції в системи автоматизації та побудови цифрових двійників дугових сталеплавильних печей.

**Виклад основного матеріалу.** У даному дослідженні розглядається задача побудови алгоритму керування переміщенням електродів дугової сталеплавильної печі (ДСП) на основі імпедансного принципу. Основною метою керування є забезпечення стабільного режиму горіння електричної дуги шляхом підтримання заданого значення її повного опору, що є одним із ключових факторів ефективності процесу плавлення сталі [1], [2].

Імпеданс дуги є узагальненим параметром, що враховує як напругу, так і струм, і визначається співвідношенням:

$$Z(t) = \frac{U(t)}{I(t)} \quad (1)$$

де  $U(t)$  – напруга дуги,  $I(t)$  – струм дуги.

Застосування імпедансного підходу до керування дозволяє більш точно враховувати фізичний стан електричної дуги порівняно з методами, що базуються лише на окремих електричних параметрах. Як показано в роботах [8], [12], підтримання стабільного імпедансу забезпечує підвищення енергоефективності процесу та зниження коливань електричних параметрів.

У системі керування задається еталонне значення імпедансу  $Z_{set}$ , яке відповідає оптимальному технологічному режиму роботи печі. Задача керування полягає у формуванні такого сигналу впливу на електрогідравлічний привід переміщення електродів, який забезпечує мінімізацію відхилення поточного значення імпедансу від заданого.

Таким чином, система керування реалізує замкнений контур регулювання, в якому параметри електричної дуги безпосередньо впливають на положення електродів і, відповідно, на довжину дуги. Взаємозв’язок між електричними та механічними параметрами системи визначає складність задачі керування та потребує застосування ефективних алгоритмів регулювання [3], [7].

Ключовим елементом системи автоматичного керування є формування сигналу похибки, який визначає різницю між заданим і поточним значенням регульованої величини. У випадку імпедансного регулювання похибка визначається як:

$$e(t) = Z_{set} - Z(t) \quad (2)$$

де  $Z_{set}$  – задане значення імпедансу,  $Z(t)$  – поточне значення імпедансу дуги.

Сигнал похибки відображає ступінь відхилення режиму роботи печі від оптимального та є основою для формування керуючого впливу. Як показано у дослідженнях динаміки електричної дуги [4], [5], навіть незначні відхилення імпедансу можуть призводити до суттєвих змін електричних параметрів процесу, що підкреслює важливість точного формування сигналу похибки.

При позитивному значенні похибки фактичний імпеданс менший за заданий, що відповідає зменшенню довжини дуги та потребує підйому електрода. При від’ємному значенні похибки, навпаки, імпеданс перевищує задане значення, що вимагає опускання електрода для відновлення робочого режиму. Така інтерпретація сигналу похибки узгоджується з принципами імпедансного керування електродами, наведеними у роботах [8], [9].

Таким чином, сигнал похибки безпосередньо визначає напрямок і інтенсивність керуючого впливу на електромеханічну систему. Його коректне формування є критично важливим для забезпечення стійкості та швидкодії системи керування, особливо з урахуванням нелінійного та нестационарного характеру електричної дуги [2], [6].

Для забезпечення стабілізації імпедансу електричної дуги у роботі використано пропорційно-інтегральний (PI) закон керування, який є одним із найбільш поширених у системах автоматичного регулювання технологічних

процесів та широко застосовується у задачах керування дуговими сталеплавильними печами [10].

Керуючий сигнал формується на основі поточного значення похибки та її інтегральної складової за наступним співвідношенням:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (3)$$

де  $K_p$  – коефіцієнт пропорційної складової,  $K_i$  – коефіцієнт інтегральної складової,  $e(\tau)$  – сигнал похибки регулювання.

Пропорційна складова забезпечує швидку реакцію системи на зміну режиму роботи дуги, формуючи керуючий вплив, пропорційний величині відхилення. Інтегральна складова, у свою чергу, виконує накопичення похибки у часі та дозволяє усунути статичну похибку регулювання, що є особливо важливим для забезпечення точного підтримання заданого імпедансу [8].

Використання саме PI-регулятора обумовлено компромісом між складністю реалізації та ефективністю керування. Додавання диференціальної складової у даному випадку є недоцільним через високу чутливість до шумів вимірювання струму та напруги, а також до швидкоплинних флуктуацій електричної дуги, що підтверджується дослідженнями динаміки дуги [4], [5].

З урахуванням фізичних обмежень системи керування, інтегральна складова регулятора обмежується в допустимих межах:

$$I_{min} \leq I(t) \leq I_{max} \quad (4)$$

що дозволяє уникнути ефекту накопичення інтегратора (integral windup) та забезпечити стабільну роботу регулятора при значних відхиленнях від заданого режиму.

Таким чином, сформований PI-регулятор забезпечує необхідну швидкодію, стійкість та точність системи керування електродами дугової сталеплавильної печі, що узгоджується з сучасними підходами до автоматизації електротехнологічних процесів [7], [9].

Оскільки програмна реалізація системи керування виконується у цифровому середовищі, безперервний закон PI-регулювання потребує переходу до дискретної форми. Такий підхід широко застосовується при

реалізації систем керування у програмному середовищі та відповідає сучасним методам чисельного моделювання [11].

Інтегральна складова регулятора у дискретному вигляді обчислюється за рекурентною формулою:

$$I_k = I_{k-1} + e_k \cdot dt \quad (5)$$

де  $I_k$  – поточне значення інтегральної складової,  $e_k$  – значення похибки на кроці  $k$ ,  $dt$  – крок дискретизації.

Відповідно, керуючий сигнал визначається як:

$$u_k = K_p e_k + K_i I_k \quad (6)$$

Така форма запису дозволяє ефективно реалізувати алгоритм регулювання у програмному кодї з використанням простих обчислювальних операцій.

Вибір кроку дискретизації  $dt$  є важливим фактором, що впливає на точність і стійкість моделювання. Занадто великий крок може призвести до втрати точності та виникнення чисельної нестійкості, тоді як надто малий крок збільшує обчислювальні витрати. У задачах моделювання електродугових печей це питання розглядається у сучасних роботах, присвячених чисельним методам моделювання [2], [3].

Дискретна форма PI-регулятора добре узгоджується з модульною архітектурою програмної реалізації та дозволяє інтегрувати алгоритм у загальний цикл моделювання, де на кожному кроці виконується обчислення похибки, оновлення інтегральної складової та формування керуючого впливу.

Таким чином, перехід до дискретної форми забезпечує можливість ефективної програмної реалізації алгоритму регулювання та створює основу для подальшої інтеграції системи керування у цифрові платформи, що відповідає сучасним тенденціям розвитку систем автоматизації [6].

Програмна реалізація алгоритму імпедансного PI-регулювання виконана з використанням мови програмування Go, що дозволяє створювати ефективні та масштабовані обчислювальні системи. На відміну від спеціалізованих середовищ моделювання, таких як MATLAB/Simulink, використання універсальної мови програмування забезпечує більшу гнучкість, можливість інтеграції у промислові системи та підвищену продуктивність обчислень.

Архітектура програмної системи побудована за модульним принципом і включає окремі компоненти, що відповідають за:

- обчислення електричних параметрів дуги;
- реалізацію алгоритму регулювання;
- чисельне інтегрування;
- збереження та візуалізацію результатів моделювання.

Центральним елементом системи є модуль PI-регулятора, який реалізовано у вигляді окремої структури даних:

```
type PIController struct {  
    Kp float64  
    Ki float64  
    I float64  
    Imin float64  
    Imax float64  
}
```

У процесі моделювання на кожному кроці часу виконується обчислення сигналу похибки, оновлення інтегральної складової та формування керуючого сигналу. Алгоритм реалізації PI-регулятора має наступний вигляд:

```
func (c *PIController) Update(e float64, dt float64) float64 {  
    c.I += e * dt  
    if c.I > c.Imax {  
        c.I = c.Imax  
    }  
    if c.I < c.Imin {  
        c.I = c.Imin  
    }  
    u := c.Kp*e + c.Ki*c.I  
    return u  
}
```

Застосування обмеження інтегральної складової дозволяє уникнути ефекту перенасичення (integral windup), що особливо важливо при значних відхиленнях імпедансу на початкових етапах роботи системи.

Використання мови Go забезпечує ряд переваг:

- високу швидкість виконання обчислень;
- ефективне управління пам'яттю;
- простоту реалізації модульної архітектури;
- можливість подальшої інтеграції у системи реального часу.

Таким чином, запропонована програмна реалізація дозволяє створити універсальний інструмент для моделювання та дослідження систем керування дуговими сталеплавильними печами.

Чисельне моделювання роботи системи керування виконується у вигляді ітераційного процесу з фіксованим кроком дискретизації  $dt$ . На кожному кроці моделювання здійснюється послідовне обчислення параметрів системи, що дозволяє відтворити її динамічну поведінку у часі.

Загальна структура обчислювального циклу включає такі етапи:

1. обчислення миттєвих значень напруги та струму дуги;
2. визначення поточного значення імпедансу;
3. обчислення сигналу похибки;
4. формування керуючого сигналу за допомогою PI-регулятора;
5. оновлення положення електрода.

У спрощеному вигляді цикл моделювання може бути представлений наступним чином:

```
for t := 0.0; t < T; t += dt {  
  Z := U / I  
  e := Zset - Z  
  u := controller.Update(e, dt)  
  electrodePosition += u * dt  
}
```

Наведений фрагмент коду ілюструє базовий принцип чисельної реалізації замкненої системи керування, у якій на кожному кроці часу здійснюється обчислення поточного значення імпедансу, формування похибки регулювання та визначення керуючого впливу. Такий підхід відповідає класичній схемі дискретного керування зі зворотним зв'язком і дозволяє адаптувати модель до програмної реалізації у середовищах реального часу.

Крок дискретизації  $dt$  обирається з урахуванням характерних часових параметрів процесу. Він повинен бути достатньо малим для коректного відтворення динаміки системи, але при цьому не призводити до надмірного збільшення обчислювальних витрат. Занадто великий крок може спричинити втрату точності або навіть нестійкість чисельного розв'язку, тоді як надто малий – суттєво збільшує час обчислень. У проведених дослідженнях значення

*dt* обирається експериментально та забезпечує компроміс між точністю та швидкодією моделювання.

Додатково слід зазначити, що вибір кроку дискретизації безпосередньо впливає на якість відтворення перехідних процесів, зокрема швидкість реакції системи та амплітуду коливань. У системах керування електродуговими печами, де характерні швидкі зміни струму та напруги, це має особливе значення.

Особливістю реалізації є синхронізація роботи всіх модулів у межах єдиного обчислювального циклу, що дозволяє забезпечити узгодженість між електричною та механічною підсистемами. Такий підхід забезпечує коректне відображення взаємодії між процесами різної фізичної природи та відповідає сучасним принципам побудови імітаційних моделей складних технічних систем [2], [3].

Для подальшого аналізу результатів моделювання у програмі передбачено збереження основних параметрів системи у файл формату CSV. Такий підхід забезпечує можливість подальшої обробки даних, побудови графіків та повторного аналізу отриманих результатів без повторного запуску моделі.

```
import (  
    "encoding/csv"  
    "fmt"  
)  
type SimulationResult struct {  
    Time float64  
    Z float64  
    Zset float64  
    Err float64  
    Control float64  
    Prop float64  
    Integ float64  
}  
func saveResult(w *csv.Writer, r SimulationResult) {  
    w.Write([]string{  
        fmt.Sprintf("%.6f", r.Time),  
        fmt.Sprintf("%.6f", r.Z),  
        fmt.Sprintf("%.6f", r.Zset),  
        fmt.Sprintf("%.6f", r.Err),  
        fmt.Sprintf("%.6f", r.Control),  
    })  
}
```

```
    fmt.Sprintf("%.6f", r.Prop),  
    fmt.Sprintf("%.6f", r.Integ),  
  })  
}
```

Наведений фрагмент демонструє принцип формування структурованого набору даних, який містить час моделювання, фактичне та задане значення імпедансу, похибку регулювання, керуючий сигнал, а також пропорційну та інтегральну складові регулятора. Збереження результатів у табличному форматі, що забезпечує розділення етапів обчислення та візуалізації результатів і підвищує відтворюваність дослідження та спрощує перевірку отриманих результатів.

У результаті виконання моделювання сформовано набір часових залежностей основних параметрів системи, зокрема імпедансу дуги  $Z(t)$ , сигналу керування  $u(t)$ , та похибки регулювання  $e(t)$ . Це дозволяє здійснити комплексний аналіз динамічних властивостей системи керування та оцінити ефективність запропонованого алгоритму.

Для оцінки ефективності імпедансного PI-регулятора проведено чисельне моделювання системи керування переміщенням електродів дугової сталеплавильної печі. У ході моделювання отримано часові залежності імпедансу дуги, сигналу похибки, керуючого впливу, а також пропорційної та інтегральної складових регулятора.

На рис. 1 представлено зміну імпедансу дуги у часі. Аналіз отриманої залежності показує, що система забезпечує вихід регульованої величини на заданий рівень протягом короткого перехідного процесу тривалістю приблизно 0.15-0.2 с. На початковому етапі спостерігається відхилення імпедансу, зумовлене початковими умовами та накопиченням інтегральної складової регулятора.

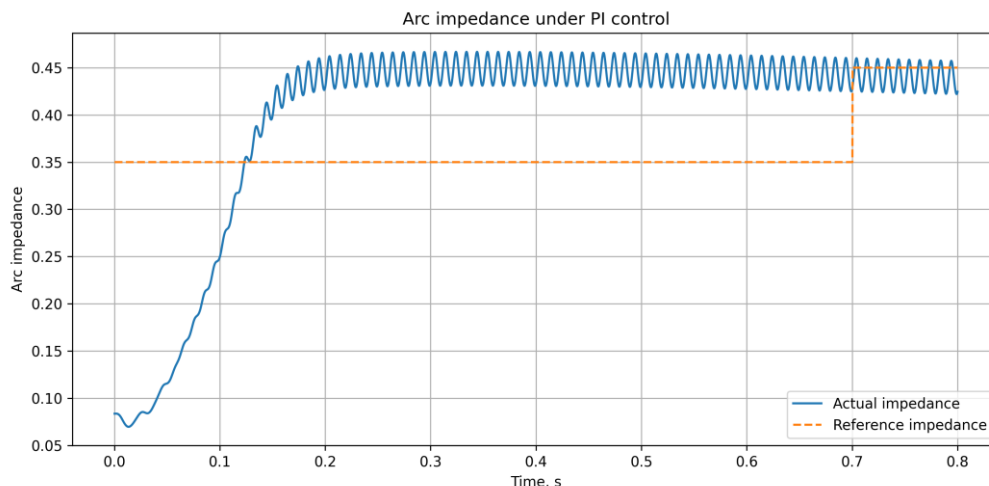


Рисунок 1 – Часова залежність фактичного та заданого імпедансу дуги при роботі PI-регулятора

Після завершення перехідного процесу імпеданс стабілізується поблизу заданого значення  $Z(t)$ . Відносна амплітуда коливань у стаціонарному режимі не перевищує 3-5% від номінального значення, що свідчить про достатній рівень демпфування та стійкість системи.

На рисунку 2 наведено часову залежність похибки регулювання. Максимальне значення похибки спостерігається на початковому етапі, після чого вона швидко зменшується та наближається до нуля. У стаціонарному режимі залишкова похибка не перевищує 2-3%, що підтверджує ефективність інтегральної складової регулятора у компенсації статичних відхилень.

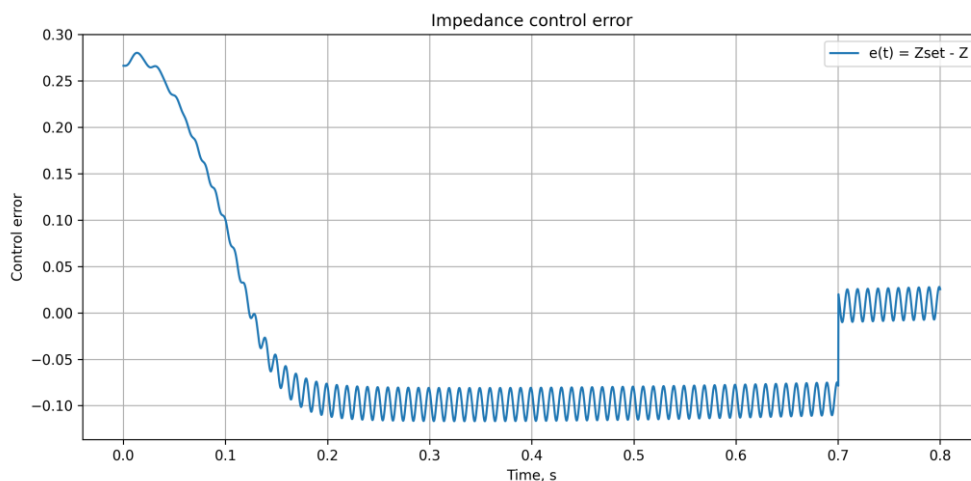


Рисунок 2 – Часова залежність похибки регулювання

На рисунку 3 представлено зміну керуючого сигналу. На початковому етапі спостерігається значний рівень керуючого впливу, що відповідає великій величині похибки. Надалі сигнал плавно змінюється та стабілізується без різких стрибків, що є важливим для запобігання перевантаженню виконавчих механізмів системи переміщення електродів.

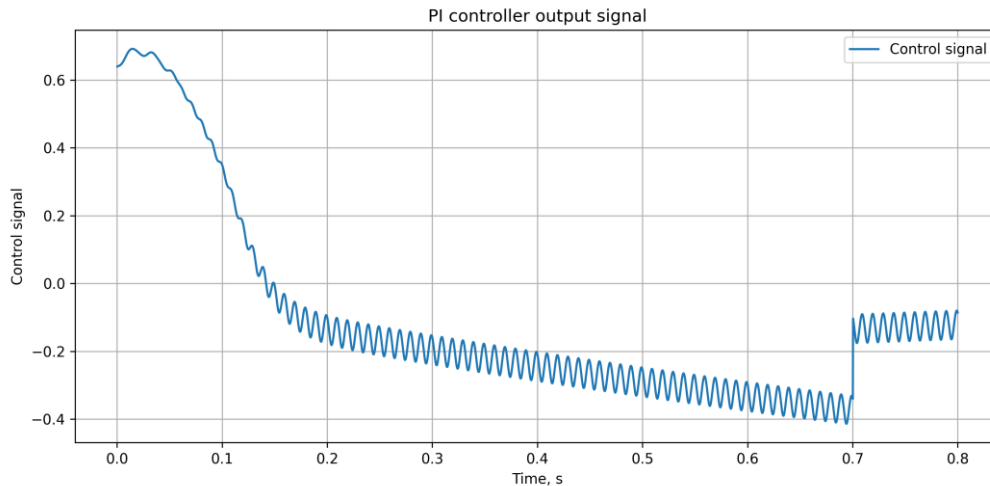


Рисунок 3 – Часова залежність керуючого сигналу PI-регулятора

На рисунку 4 наведено пропорційну та інтегральну складові сигналу керування. Пропорційна складова забезпечує швидку реакцію системи на зміну похибки, тоді як інтегральна складова поступово накопичується та відіграє ключову роль у забезпеченні нульової статичної похибки. Використання обмеження інтегратора дозволяє уникнути його надмірного зростання та запобігає виникненню нестійких режимів.

Обмеження інтегральної складової дозволяє уникнути її надмірного зростання та запобігає виникненню нестійкості.

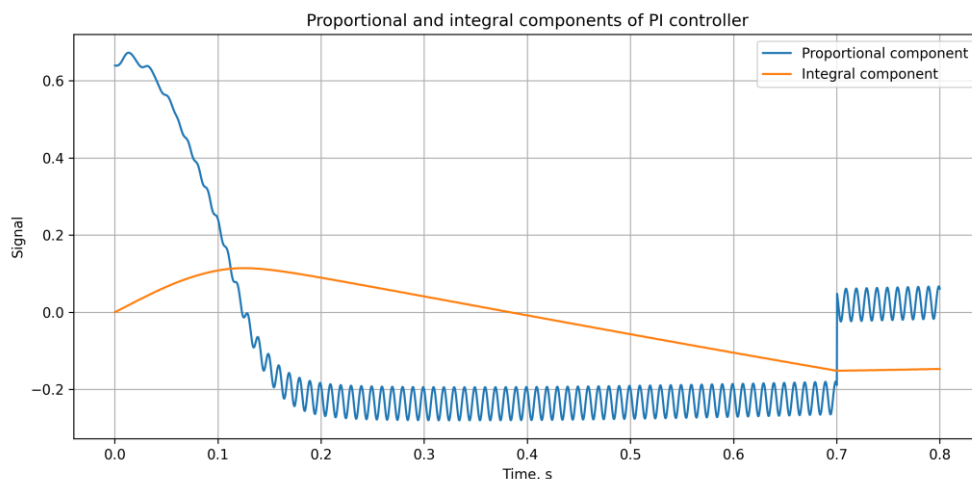


Рисунок 4 – Пропорційна та інтегральна складові сигналу керування

Сучасні підходи до обробки та візуалізації даних у наукових дослідженнях широко базуються на використанні відкритих програмних бібліотек, що інтегруються з мовами загального призначення [13]. Це забезпечує гнучкість у побудові графіків, можливість автоматизації обчислень і адаптації програмних рішень до задач інженерного моделювання.

Для візуалізації результатів моделювання використано засоби побудови графіків на основі мови програмування Python із застосуванням бібліотеки Matplotlib, яка широко використовується у задачах наукових обчислень та аналізу даних. Використання такого підходу дозволяє автоматизувати обробку результатів моделювання та формувати часові залежності параметрів системи. Це підвищує інформативність отриманих результатів, спрощує їх інтерпретацію та забезпечує відтворюваність проведених досліджень [12], [13].

Узагальнюючи результати моделювання, можна зробити висновок, що запропонований PI-регулятор забезпечує:

- стійку роботу системи керування;
- швидкий вихід на заданий режим (до 0.2 с);
- низький рівень статичної похибки (до 3%);
- плавний характер керуючого впливу без перевантаження виконавчих механізмів.

Отримані результати узгоджуються з відомими підходами до керування дуговими сталеплавильними печами [8], [9] та підтверджують ефективність використання імпедансного принципу регулювання.

Разом з тим слід зазначити, що розроблена модель містить певні спрощення, зокрема не враховує випадкові збурення, шум вимірювання та змінність параметрів дуги у часі. Це може впливати на точність відтворення реальних умов експлуатації. Подальші дослідження доцільно спрямувати на ускладнення моделі та врахування стохастичних факторів, що дозволить підвищити її адекватність реальним технологічним процесам.

**Висновки.** Виконана робота вирішує конкретну інженерну задачу: створити працездатну програмну реалізацію імпедансного PI-регулятора для системи керування електродами ДСП таку, що не потребує спеціалізованих середовищ і може бути безпосередньо інтегрована у промислове програмне забезпечення.

Вибір імпедансного принципу регулювання не випадковий. Аналіз існуючих підходів [8, 9] показав, що підтримання заданого співвідношення напруги і струму дуги забезпечує стійкішу роботу печі, ніж регулювання за окремими електричними величинами, – особливо в умовах різких змін властивостей шихти на початку плавки.

Розроблена математична модель враховує взаємозв'язок електричних і механічних параметрів системи. Алгоритм PI-регулювання реалізовано мовою Go у вигляді модульної структури з окремими компонентами для розрахунку керуючого сигналу, виконання циклу симуляції та збереження результатів. Така архітектура дозволяє змінювати параметри регулятора без переписування решти коду, що важливо як для налагодження, так і для подальшої адаптації до реальних умов експлуатації.

Результати чисельного моделювання підтвердили працездатність підходу: система виходить на заданий імпеданс за 0,15-0,2 с, перерегулювання залишається в допустимих межах, а статична похибка не перевищує 2-3%. Не менш важливою виявилася плавність керуючого сигналу – різкі зміни напруги на приводі електрода неприпустимі з точки зору механічного ресурсу конструкції, і запропонований алгоритм цю умову виконує.

Залишкова статична похибка є очікуваним наслідком спрощень у моделі дуги, а не вадю регулятора. Її усунення – завдання наступного етапу, де доцільно розглянути адаптивне коригування коефіцієнтів залежно від

поточного режиму роботи печі та ввести компенсацію випадкових збурень і шумів вимірювання.

Отримані результати мають практичну цінність для розробників цифрових систем керування ДСП, а запропонований підхід до програмної реалізації може бути використаний і ширше – зокрема при створенні цифрових двійників електротехнологічних об'єктів.

#### ЛІТЕРАТУРА/REFERENCES

1. Fruehan, R. J. (1998). *The making, shaping and treating of steel* (11th ed.). AISE Steel Foundation.
2. Hay, T., Visuri, V.-V., Aula, M., & Echterhof, T. (2021). A review of mathematical process models for the electric arc furnace process. *Steel Research International*, 92(1), 2000395. <https://doi.org/10.1002/srin.202000395>
3. Abadi, M., Tang, L., & Rashidi, M. (2024). A review of simulation and numerical modeling of electric arc furnaces. *Results in Engineering*. <https://doi.org/10.1016/j.rineng.2024.102473>
4. Cassie, A. M. (1939). Arc rupture and circuit severity: A new theory. *CIGRE Report*, (102), 1–14.
5. Golestani, S., & Samet, H. (2016). Generalised Cassie–Mayr electric arc furnace models. *IET Generation, Transmission & Distribution*, 10(13), 3364–3373. <https://doi.org/10.1049/iet-gtd.2016.0141>
6. Blažič, B., Škrjanc, I., & Logar, V. (2024). Arc quality index based on three-phase Cassie–Mayr electric arc furnace model. *IEEE Transactions on Industrial Electronics*. <https://doi.org/10.1109/TIE.2024.3352119>
7. Balan, R., Hancu, O., & Lupu, C. (2008). Adaptive control of electric arc furnace. In *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (pp. 19–24).
8. Wang, Z., Mao, J., Tian, X., Li, J., & Yuan, P. (2020). Impedance regulation based electrode control system for electric arc furnace. *ISIJ International*, 60(5), 982–990. <https://doi.org/10.2355/isijinternational.ISIJINT-2019-459>
9. Logar, V., & Škrjanc, I. (2012). Modeling and control of electric arc furnace using a neural network approach. *IEEE Transactions on Industrial Electronics*, 59(3), 1630–1637. <https://doi.org/10.1109/TIE.2011.2161655>
10. Billings, S. A., Boland, F. M., & Nicholson, H. (1979). Electric arc furnace modeling, simulation and control. *IEE Proceedings – Control Theory and Applications*, 126(6), 567–574. <https://doi.org/10.1049/piee.1979.0126>
11. Boulet, B., Lalli, G., & Ajersch, M. (2003). Modeling and control of an electric arc furnace. In *Proceedings of the American Control Conference (ACC)* (Vol. 5, pp. 3862–3867).
12. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
13. Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3), 10–20. <https://doi.org/10.1109/MCSE.2007.58>
14. Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design, and tuning* (2nd ed.). ISA.
15. Ogata, K. (2010). *Modern control engineering* (5th ed.). Prentice Hall.
16. Dorf, R. C., & Bishop, R. H. (2016). *Modern control systems* (13th ed.). Pearson.
17. Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). *Feedback control of dynamic systems* (7th ed.). Pearson.

18. Donovan, A. A., & Kernighan, B. W. (2015). *The Go programming language*. Addison-Wesley.  
19. Pike, R. (2012). Go at Google: Language design in the service of software engineering. In *Proceedings of the ACM Conference*. <https://doi.org/10.1145/2384716.2384720>  
20. Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Received 15.03.2025.

Accepted 10.04.2025.

Published 30.04.2026

UDC 621.365:681.5

Dmytro Kartamyshev

## SOFTWARE IMPLEMENTATION OF AN IMPEDANCE PI CONTROLLER FOR AN ELECTRIC ARC FURNACE CONTROL SYSTEM USING THE GO PROGRAMMING LANGUAGE

**Abstract.** *Electrode control in electric arc furnace (EAF) operation represents one of the most technically demanding problems in steelmaking. The inherently unstable behaviour of the electric arc – its sensitivity to scrap composition, inter-electrode gap geometry, and thermal conditions – results in current and voltage fluctuations, increased specific energy consumption, and accelerated electrode wear. Despite a considerable body of research on mathematical modelling and controller design for EAFs, most published work relies on specialised software environments such as MATLAB/Simulink and does not address the direct deployment of control algorithms in industrial automation systems. The question of implementing control logic using general-purpose programming languages suitable for production environments remains insufficiently explored.*

*The objective of this work is to develop an impedance-based PI control algorithm for the EAF electrode positioning system and to implement it in the Go programming language, followed by analysis of the closed-loop system's dynamic behaviour.*

*The mathematical model is based on the Cassie-Mayr arc equations, which describe arc conductance dynamics under both transient and steady-state conditions. The controller is implemented in discrete time with a fixed integration step to ensure numerical stability. An anti-windup mechanism – clamping the accumulated error sum within defined bounds – is applied to prevent integrator saturation under control signal constraints. The software is structured as a modular Go application comprising dedicated components for control signal computation, simulation loop execution, and CSV-based result logging. The choice of Go is justified by its practical advantages for industrial automation: compilation to native executable code, absence of runtime dependencies, support for modular architecture, and high computational performance. Simulation*

*results are visualised using Python/Matplotlib, ensuring reproducibility and clarity of the obtained data.*

*Numerical simulation results show that the developed system achieves a stable transient response with a settling time of 0.15-0.2 s and overshoot within acceptable limits. A residual steady-state error of 2-3% is attributed to simplifications in the adopted arc model and can be reduced by introducing adaptive gain correction. The smooth control signal profile, free from abrupt changes, confirms the algorithm's suitability for real-world application with respect to mechanical loading on the electrode drive. The findings demonstrate the viability of Go as an implementation language for control algorithms within digital EAF automation systems, as well as for the development of digital twins of electrotechnological processes.*

**Keywords:** *electric arc furnace, impedance control, PI controller, mathematical modeling, electric arc, Go programming language, control system, automation*

**Картамишев Дмитро Олександрович** – к.т.н., асистент кафедри автоматизації виробничих процесів Донбаської державної машинобудівної академії, Краматорськ – Тернопіль – Хуст, Україна. ORCID: <https://orcid.org/0000-0003-3240-8919>

**Dmytro Kartamyshev** – candidate of engineering sciences, assistant lecturer of the department of automation of production processes, Donbas State Engineering Academy, Kramatorsk – Ternopil – Khust, Ukraine. ORCID: <https://orcid.org/0000-0003-3240-8919>