

УДК 004.021

О.С. Усік, Л.М. Маркіна
Луцький національний технічний університет
СЕРЕДОВИЩЕ ДЛЯ МОДЕЛЮВАННЯ СИСТЕМ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ

В даній роботі описано мету, вигляд та можливості додатку «Середовище для моделювання систем автоматичного регулювання».

Було детально описано принцип його роботи, логіко-математичну базу, інтерфейс та деякі загальні аспекти створення подібних застосунків з використанням мови програмування JavaScript., тому що побудова схем і дослідження систем автоматичного регулювання є однією з задач автоматизації. В масштабах прикладного та навчального проектування цей процес займає тривалий час та є трудомістким. Для розробки інтерфейсу використано технологію HTML5, тому середовище орієнтоване на використання не лише в мережі, а й у програмах під операційну систему Windows та у мобільних засобах

Ключові слова: САР, САК, середовище, JavaScript, HTML5.

А.С. Усик, Л.М. Маркіна
СРЕДА ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ

В данной работе описано цель, вид и возможности приложения «Среда для моделирования систем автоматического регулирования».

Детально описан принцип его работы, логико-математическую базу, интерфейс и некоторые общие аспекты создания подобных приложений с использованием языка программирования JavaScript., Так как построение схем и исследования систем автоматического регулирования является одной из задач автоматизации. В масштабах прикладного и учебного проектирования этот процесс занимает длительное время и является трудоемким. Для разработки интерфейса использована технология HTML5, поэтому среда ориентирована на использование не только в сети, но и в программах под операционную систему Windows и мобильных средствах

Ключевые слова: САР, САК, средовище, JavaScript, HTML5.

A.S. Usik, L.M. Markina
ENVIRONMENT FOR AUTOMATIC CONTROL SYSTEMS MODELING

This paper describes the purpose, appearance and application possibilities "environment for simulation of automatic control systems."

It describes in detail how it works, logical-mathematical basis, and some common interface aspects of these applications using the programming language JavaScript., Because the construction of schemes and study of automatic control systems is a task automation. In the scale of applied and academic design process takes a long time and is labor intensive. For the development of interface technology used HTML5, so the use of environment-oriented not only online but also in applications under the Windows operating system and mobile vehicles. existing methods of solving this problem was the calculation using MATLAB gear and output graphs on the screen. This option is not convenient and effective. It takes time and requires additional steps and skills from the user.

JavaScript has to implement all the methods we need components and combining them into an operating system which can work comfortably user. This programming language is not compiled, interpreted and therefore can change the required program elements directly during its operation. It is very important to calculate mathematical expressions, and work with an interface based on HTML5.

Keywords: SAC, SAR, modeling, environment, JavaScript, HTML5.

Постановка проблеми. Побудова схем і дослідження систем автоматичного регулювання є однією з задач автоматизації. В масштабах прикладного та навчального проектування цей процес займає тривалий час та є трудомістким.

Аналіз останніх досліджень і публікацій. Одним із існуючих методів розв'язання цієї проблеми було обчислення за допомогою програми MATLAB передаточних та вивід графіків на екран. Цей варіант не є зручним та ефективним. Він займає багато часу та потребує додаткових дій та навиків від користувача.

Відсутність єдиного середовища для одночасного задання схеми, параметрів ланок, обчислення передаточних функцій та виведення обрахунків у потрібному вигляді зумовило зародження ідеї для створення описуваного застосунку.

Постановка завдань. Мета роботи — створити середовище для зручного моделювання та дослідження САР та САК.

Завдання дослідження:

- Опис інтерфейсу середовища, демонстрація роботи з кожним елементом.

- Опис логіко-математичної бази та створення додатку за допомогою JavaScript

Предмет дослідження: середовище для моделювання систем автоматичного регулювання.

Об'єкт дослідження: процес розробки додатку «Середовище для моделювання систем автоматичного регулювання»

Виклад основного матеріалу. Першим кроком до розробки було вивчення основних понять та термінів, принципів обрахування характеристик та передаточних.

Тому ми виділили три основні частин які слід було продумати:

- інтерфейс
- логіко-математична база
- вивід результатів

JavaScript має усі методи для реалізації потрібних нам компонентів і об'єднання їх в діючу систему з якою зручно зможе працювати користувач. Це мова програмування, яка не компілюється, а інтерпретується тому може змінювати потрібні елементи програми прямо під час її роботи. Це дуже важливо для обчислення математичних виразів, та роботи з інтерфейсом на базі HTML5.

Перша частина: інтерфейс.

Для розробки інтерфейсу використано технологію HTML5, тому середовище орієнтоване на використання не лише в мережі, а й у програмах під операційну систему Windows та у мобільних засобах. Наявність потужних засобів візуалізації, простота розробки, поширеність документації та швидкодія є найвагомішими аргументами на використання саме її.

HTML5 - не продовжувачем мови розмітки гіпертексту, а є новою відкритою платформою, призначеною для створення веб - додатків використовують аудіо, відео, графіку, анімацію та багато іншого.

В новому стандарті добавлено багато нових тегів серед яких буде використано такі DOM елементи, як article, canvas, figure та section.

Інтерфейс середовища на базі HTML стилізоване CSS та логіко-математичне ядро на базі JS підключаються в одному .html файлі.

Опис основних частин.

Після запуску програми ми побачимо початкові елементи (Рис.1).

Основна частина має клас «section» та містить у собі CAP – схему, полотна для малювання графіка та меню налаштувань.

CAP-схема, відображається тегом article та зображена у вигляді послідовних блоків зворотного зв'язку(верхні блоки) та розімкненої системи(нижні).

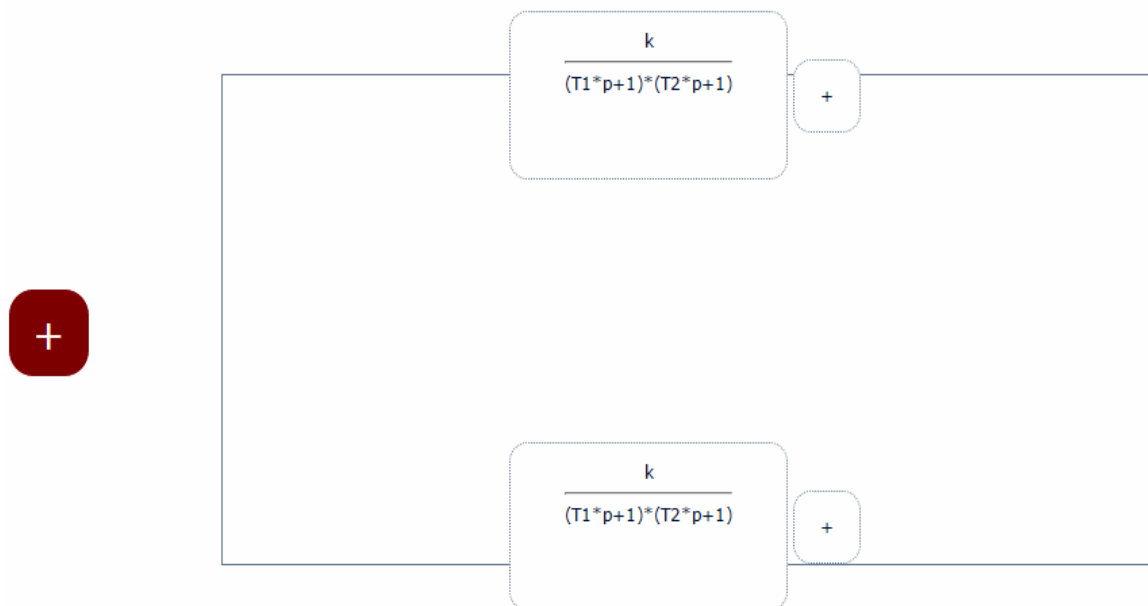


Рис. 1. Головне меню

Другу частину відображає тег section, що має клас «lSide». Вона розміщена зліва та відкривається після наведення на «+» в червоному. Відкриття працює по виклику CSS як реакція на: hover. Тут усі доступні ланки для проектування САР (Рис.2).

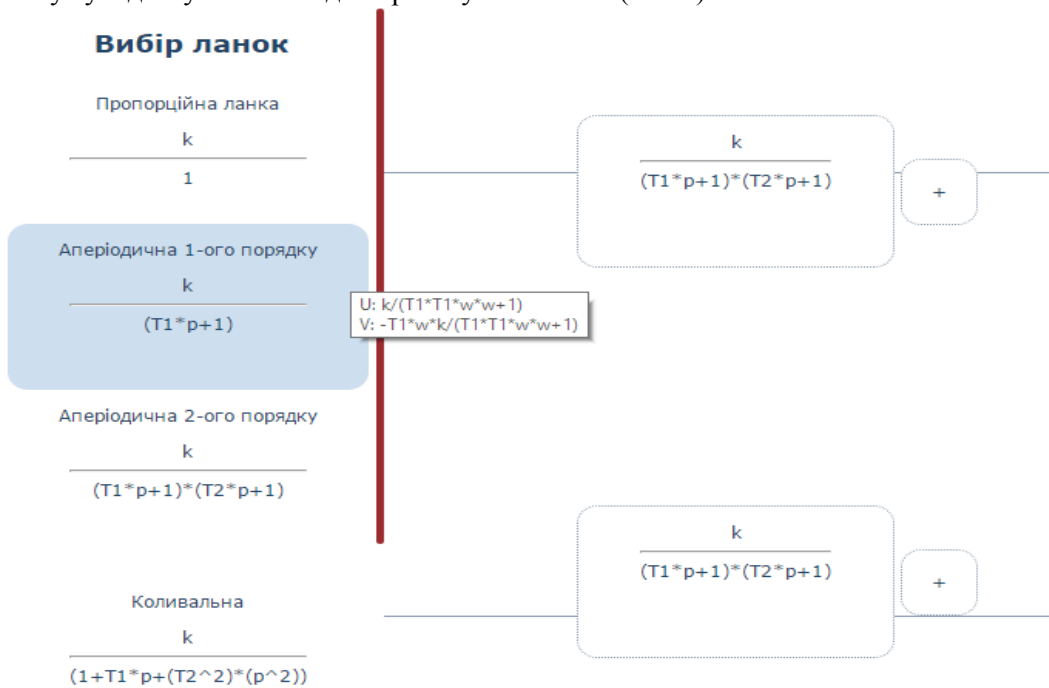


Рис. 2 Панель вибору ланок

Головною особливістю інтерфейсу є можливість перетягнути ланку з лівого меню вибору ланок та встановити її на САР-схему. Це реалізується javascript-функцією Drag and Drop, що запозичена з сайту learn.javascript.ru, що знаходиться у вільному доступі. Функцію адаптовано під потреби проекту.

Щоб додати нову ланку на САР або зворотній зв'язок перетягніть ланку та відпустіть кнопку мишки над елементом з символом «+» (Рис. 3).

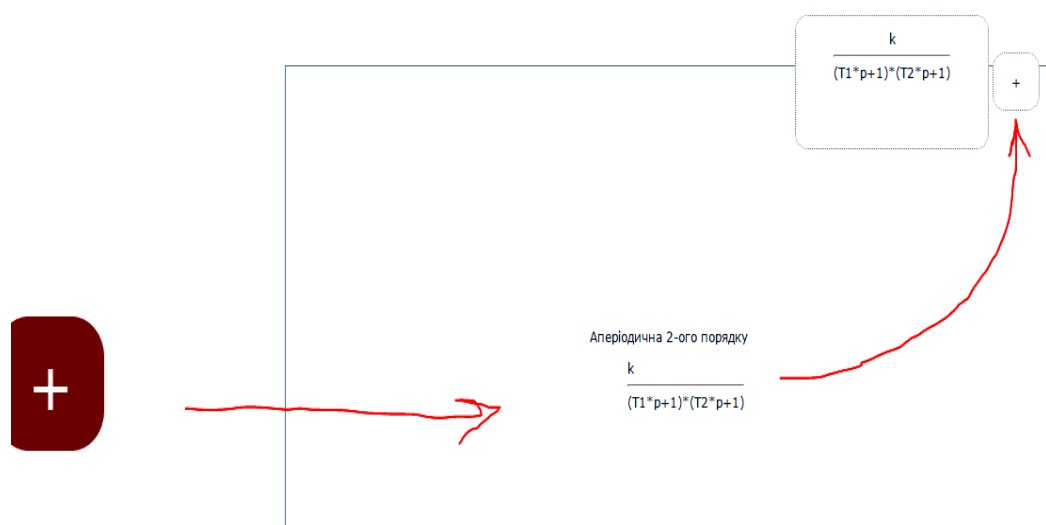


Рис. 3. Drag and Drop

Щоб замінити ланку потрібно перетягнути і «покласти» на неї нову ланку з області вибору ланок.

Щоб видалити слід навести на неї мишкою та натиснути «X».

Сама ланка складається з тега `p`, `figure`, `label`, `input` та `br` (див. Рис.4). В специфікації HTML5 тег `figure` використовується для опису в собі якогось тексто-логічного елементу. В даному випадку це – формули передаточних.

Three panels showing the state of a web form for an aperiodic second-order system. Each panel has the title "Аперіодична 2-ого порядку" and the formula $(T_1 * p + 1) * (T_2 * p + 1)$. The input fields for k , T_1 , and T_2 all contain the value "1". At the bottom of each panel are "Go" and "X" buttons.

Рис. 4. Загальний вигляд ланки при наведенні 1, введення даних 2, та видалення ланки 3

Нижче знаходиться меню налаштувань.

Тут нам доступно:

- масштабування графіків по кожній з осей у вигляді коефіцієнтів, що вводяться в меню `Ox zoom` та `Oy zoom`. Для прикладу - початкове значення 0 означає, що до автоматично-вибраного масштабу додано 0 (Масштабування проводиться автоматично, так щоб графік в середньому займав 6/8 області полотна.).

- введення нижньої межі до верхньої границі змінної часу w .
- задання кроку змінної w (збільшення кроку пришвидшує обчислення та зменшує візуальну якість нарисованих характеристик)
- Вибір потрібних характеристик
- Встановлення типу системи (розімкнена, замкнена, по збуренню)
- головна кнопка «Показати характеристики» малює потрібні нам графіки на області графіків.

Все це зображено на рис. 5

Змінити графік

Всі

АФХ

АЧХ

Амплітудо-фазова

ЛФЧХ + Фазочастотна

годограф Михайлова

критерій Найквіста

критерій Найквіста в логарифмічному масштабі

Розімкнена система

Замкнута

- / +

Збурення

f

Ox zoom = 1

Oy zoom = 1

w = 0 до 12

step = 0,01

Показати характеристики

Рис. 5. Меню налаштувань

Друга частина: логіко-математична база.

Для обчислення всіх величин було створено окрему JavaScript бібліотеку, яка реалізує функції розрахунків передаточних розімкненої, замкненої та по збуренню і потрібних для характеристик параметрів, перетворення формул, та виводу в консоль.

В бібліотеці доступні функції та методи для розрахунку формул передаточних та характеристик для АФЧХ, ЛАФЧХ, ФЧХ, годографа Михайлова, критерія Найквіста, та Найквіста в логарифмічному масштабі.

Середовище підключається як ядро на базі JS до html документу за допомогою тегу:

```
<script src='SAR_processor.js'></script>
```

Усе це веб технології, тому процесор можна розмістити на сайті, легко доробити, запускати на будь-якому комп'ютері відкриваючи браузером Google Chrome, або зберігати незалежний файл html з бібліотекою та працювати офлайн.

Для створення Win32 додатку з цією бібліотекою можна використовувати Node.js як основний двигун. Для підключення бібліотеки можна скористатися наступним методом:

В файлі бібліотеки SAR_processor.js на початку вставити строки до потрібної нам функції exports.SAR_processor = " Назва_функції що експортується"; //(count, Cooker, ConsumParts ...)

В файлі програми дописати

```
var SAR_processor = require('./lib/SAR_processor.js');
var Obj = new SAR_processor.Назва_функції(a, b, c);
```

В бібліотеці містяться функції count, setter, getElementData, Cooker, ConsumParts та getSystemPoints.

count(s,j) – обраховує у виразі «s» кількість комплексних множників позначених символом «j», щоб знати який множник поставити перед ними($\sqrt{-1}$, -1, $-\sqrt{-1}$, +1).

setter(p) – обробляє об'єкт «p», що видає функція Cooker створюючи в ньому фазову $\phi(w)$, $A(w)$ та $L(w)$

getElementData(e) – виводить всі доступні ланки з об'єкту DATA в DOM елемент «e» задаючи їх кінцевий вигляд тегами <div>, <input>, <label> та <p>.

ConsumParts(ElemenClass) – організовує обробку введених даних та переводить все в глобальний вигляд, рахує значення дійсної та уявної значень САК та зворотного зв'язку; готує данні для функції Cooker та інше.

Аргумент ElemenClass може приймати два значення: 'ones' та 'under' САК та зворотній зв'язок відповідно.

Cooker(W) –перетворює масив ланок W у готовий до обрахунку символічний вираз. Множники з комплексними значеннями відносить до уявної частини інші до дійсної. В результаті роботи повертає дійсну та уявну частини розімкненої системи чи зворотного зв'язку.

Математичні формули, що використовувалися при побудові характеристик:

$$W(p) = W_1(p) \times W_2(p) \times \dots \times W_n(p) , \quad (1)$$

де $W(p)$ – передаточна функція розімкненої системи

$$W(p) = \frac{W_p(p)}{1 + W_p(p) \times W_3(p)} , \quad (2)$$

де $W(p)$ – передаточна функція системи зі зворотнім зв'язком

$$W(p) = \frac{W^f(p)}{1 + W_p(p)} , \quad (3)$$

де $W(p)$ – передаточна функція системи зі збуренням

$$A(w) = \sqrt{U^2(w) + V^2(w)} , \quad (4)$$

де $A(w)$ – амплітудно-частотна характеристика

$$\varphi(w) = \arctg\left(\frac{V(w)}{U(w)}\right) \times \frac{180}{\pi} , \quad (5)$$

де $\varphi(w)$ – фазо-частотна характеристика

$$L(w) = 20 \log(A(w)) , \quad (6)$$

де $L(w)$ - логарифмічно-частотна характеристика.

Після звернення схеми ми отримуємо об'єкт в якому міститься дійсна та уявна частина, фаза системи, її амплітуда, амплітуда в логарифмічному масштабі, дійсна та уявна частини характеристичного рівняння.

Усі вони обраховуються методом eval() який виконує потрібну нам формулу. Все що порібно це задавати значення частоти w відносно якої формули зводяться.

Третя частина: вивід результатів.

Усі проведені розрахунки та формули можна досліджувати в консолі. При натисканні клавіші «F12», або «правою кнопкою мишки -> Посмотреть код» ми побачимо середовище перегляду коду. Тут у верхньому меню обираємо вкладку «Console» або «Консоль» та бачимо, усі проведені розрахунки або їх помилки.

Область графіків - найбільша область. Початково вона прихована та після натискання на кнопку «Показати характеристики» вона з'являється. Кожен підпис позначає, яка характеристика і де буде нарисована (Рис. 6).

Намальований графік можна зберегти у вигляді звичайного зображення в форматах *.jpg або *.png. Клацнувши правою кнопкою миші на графіку в меню оберіть «Зберегти як» та вкажіть шлях куди хочете файл зберегти. Це зроблено стандартними методами браузера.

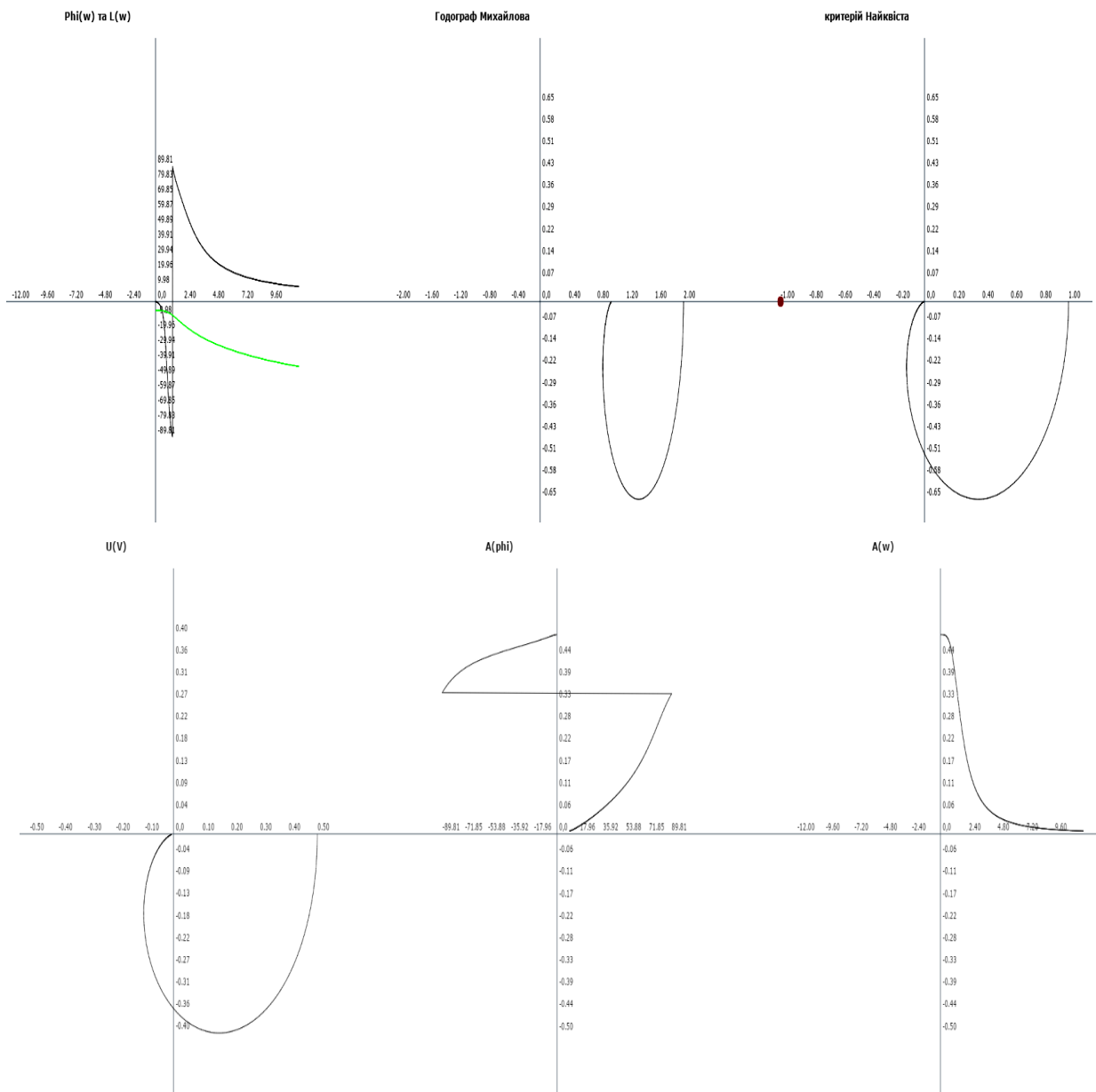


Рис. 6 Вигляд графіків

Висновок:

Таким чином, по завершенню розробки результатом, є готовий до роботи веб-додаток, що ефективно виконує поставлені перед ним задачі, є зручним у користуванні та може легко розширюватися в майбутньому. Середовище повертає результати в потрібній нам формі та дає доступ до перебігу обрахунків.

Цей веб-додаток можна легко впровадити наприклад, в WIN RT програмі або на сайті університету, що дасть онлайн доступ до даного ресурсу. Встановлення на сервері Node.js дасть змогу проводити обчислення на сервері повертаючи нам лише результат, що пришвидшить роботу з додатком. Поширення програми у вигляді збірки за html-файлу та логіко-математичної бібліотеки дає змогу використовувати середовище offline на комп'ютерах, смартфонах, терміналах та ін.

Програма може бути запущена на усіх комп'ютерах з встановленим браузером на базі WebKit, тому є доступною кожному студенту, що значно піднімає рівень обізнаності, та економить час при виконанні багатьох навчальних задач.

Додаток буде актуальним тривалий час, а стаття матиме повчальний характер для тих хто починає вивчати створення веб-додатків з використанням JavaScript та вивчає теорію автоматичного керування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Попович М. Г. Теорія автоматичного керування : підручник / М. Г. Попович, О. В. Ковальчук. – К. : Либідь, 1997.
2. Воронов А. А. Теория автоматического управления. Ч. 2. Теория нелинейных и специальных систем автоматического управления. – М. : Высш. шк., 1986. – 504 с.
3. Воронов А. А. Основы теории автоматического управления. – М. : Энергия, 1980.
4. Костнюк Л. Д. Моделирование электроприводов : навч. посіб. / Л. Д. Костнюк, В. І. Мороз, Я. С. Паранчук. – Львів : Видавництво Національного університету «Львівська політехніка», 2004. – 404 с.
5. Говорохин В. Комп'ютер в математическом исследовании : учебный курс / В. Говорохин, В. Цибулин – СПб. : Питер, 2001. – 624 с.

Рецензенти:

Марчук Віктор Іванович доктор технічних наук, професор.

Хвещук Олександр директор ТМ 'Забіяка'.

Стаття надійшла до редакції 25.04.2016.