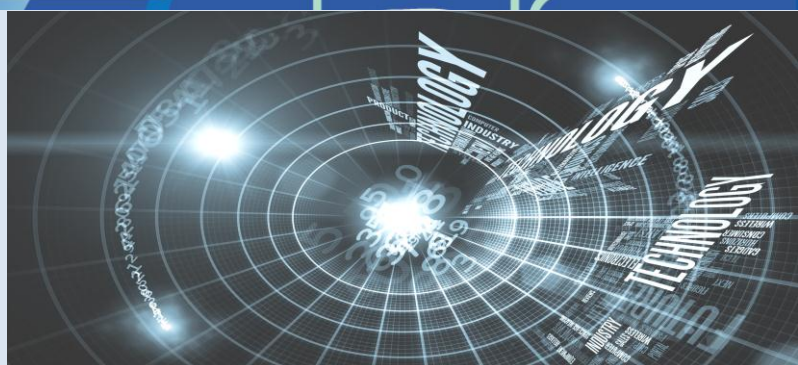
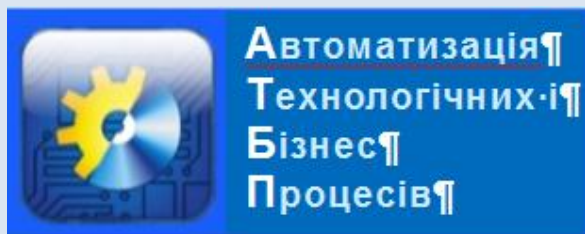




Автоматизація технологічних і бізнес-процесів Automation of technological and business-processes

-  Implementation of forecast control based on cubic spline in closed ACS: concept of construction and preliminary analysis of work
-  Development of autonomous web-services for real-time simulation of dynamic systems based on MATLAB Simulink models
-  Methods of high-performance human-machine interface
-  Research of computer Vision Methods for use in automated systems
-  Information and analytical system of the ONAF admissions committee as the basis for automated management of student contingent formation





Volume 12, Issue 4 /2020

ISSN 2312-3125 (print), ISSN 2312-931X (online)

УДК 681.5+66-933.6+338.364

Головний редактор:

Артеменко С.В., д.т.н., проф. (Одеса)

Почесний науковий консультант:

Хобін В.А., д.т.н., проф. (Одеса)

Заступники головного редактора:

Котлик С.В., к.т.н., доц. (Одеса)

Єгоров В.Б., к.т.н. (Одеса)

Редакційна колегія:

Монтік П.М., проф. (Одеса, Україна)

Князева Н.О., проф. (Одеса, Україна)

Плотніков В.М., проф. (Одеса, Україна)

Дорофєєв Ю.І., доц. (Харків, Україна)

Тимченко В.Л., проф. (Миколаїв, Україна)

Жученко А.І., проф. (Київ, Україна)

Кіріченко Л.О., проф. (Харків, Україна)

Радівілова Т.А., доц. (Харків, Україна)

Коновенко Н.Г., доц. (Одеса, Україна)

Apostolos Korlos, prof. (Thessaloniki, Greece)



Журнал «Автоматизація технологічних і бізнес-процесів» було ініційовано до видання рішенням складу І Всеукраїнської науково – практичної конференції «Інформаційні технології та автоматизація – 2008», що відбулась у стінах Одеської національної академії харчових технологій та продовжує проводитися щорічно:

www.itia.com.ua

Журнал зареєстровано
Міністерством Юстиції України
Серія KB №15895-4367P від 16.10.2009 р.

Щоквартальний науково-виробничий журнал «Автоматизація технологічних і бізнес-процесів» включено до категорії "Б" Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата наук згідно наказу МОН України від 11.07.2019 № 975 «Про затвердження рішень атестаційної колегії міністерства щодо діяльності спеціалізованих вчених рад від 23 квітня 2019 року» за спеціальностями 122, 123, 151.

www.atbp.onaft.edu.ua

Відповідальний редактор:

Бодюл О.С., к.т.н. (Одеса)

Комп'ютерний дизайн та верстка:

Козуб О.О. (Одеса)

Засновник:

Одеська національна академія харчових технологій

Адреса редакції:

м. Одеса, вул. Канатна, 112, м. Одеса, 65039,
тел.: (048)712-42-54, e-mail: journal.atbp@gmail.com

Підписано до друку 08.12.2020 р.

Рекомендовано до друку та розташування в мережі
Інтернет Вченою радою Одеської національної
академії харчових технологій

08 грудня 2020 р., протокол № 6

Відповідальність за достовірність інформації несе
автор публікації.

Матеріали друкуються мовою оригінала.
Передрукування матеріалів журналу дозволяється
лише за згодою редакції. Ліцензія CC-BY.

Надруковано у видавництві «Diol Print»

Тираж 101 прим.



AUTOMATION OF TECHNOLOGICAL AND BUSINESS-PROCESSES

Volume 12, Issue 4 /2020

ISSN 2312-3125 (print), ISSN 2312-931X (online)

Editor-in-chief:

Artemenko S., prof. (Odessa, Ukraine)

Honorary scientific consultant:

Khobin V., prof. (Odessa, Ukraine)

Deputy chief editors:

Kotlyk S., PhD (Odessa, Ukraine)

Yehorov V., PhD (Odessa, Ukraine)

Editorial Board:

Montik P., prof. (Odessa, Ukraine)

Kniazieva N., prof. (Odessa, Ukraine)

Plotnikov V., prof. (Odessa, Ukraine)

Dorofiev Y., PhD (Kharkiv, Ukraine)

Tymchenko V., prof. (Mykolayiv, Ukraine)

Zhuchenko A., prof. (Kiev, Ukraine)

Kirichenko L., prof. (Kharkiv, Ukraine)

Radivilova T., PhD (Kharkiv, Ukraine)

Konovenko N., PhD (Odessa, Ukraine)

Apostolos Korlos, prof. (Thessaloniki, Greece)

Executive editor:

Bodiul O.S., PhD (Odessa, Ukraine)

Computer design and imposition:

Kozub O.O. (Odessa, Ukraine)

Founder:

Odessa National Academy of Food Technologies

Address of the editorial office:

Ukraine, Odessa, Kanatnaya str, 112, 65039,

tel.: (048)712-42-54, e-mail:

journal.atbp@gmail.com

It's sent for the press 14.12.2020

It's recommended for printing and publishing online by academic council of Odessa National Academy of Food Technologies

08 December 2020, protocol № 6

Articles are printed in original language

It's allowed to use materials from the journal according to the Creative Commons license: CC-BY.

It's printed in publishing house «Diol Print» (Odessa).
Circulation is 101 copies.

Журнал реферується і індексується 14-ма провідними Світовими базами даних і індексними системами: EBSCO Information Services, ISSN, CrossRef, NBUV, Directory of Open Access Journals (DOAJ), ROAD, Google Scholar, OCLC WorldCat, Bielefeld Academic Search Engine (BASE) ULRICHS WEB Global Serials Directory, Index Copernicus International.

Журналу присвоєно Міжнародний стандарт якості ISO 26324: 2012. Стандарт свідчить про те, що кожній статті опублікованій в нашому журналі в обов'язковому порядку присвоюється Міжнародний цифровий ідентифікатор DOI номер (Digital Object Identifier). Це в свою чергу гарантує, що доступ до матеріалу статті буде доступний для читачів незалежно від будь-якого роду форс-мажорних ситуацій.

Згідно з Міжнародними нормами журналу присвоєна ліцензія Creative Commons CC-BY Attribution, яка фіксує звіт правил щодо захисту авторських прав авторів. Для збільшення рівня цитованості статей, опублікованих в нашому журналі вченими з усього Світу, і, як наслідок, збільшення індексу цитування наших авторів, редакція журналу АТБП оформляє всі пристатейні списки літератури відповідно до IEEE Citation Style (стиль цитування Міжнародної асоціації Institute of Electrical and Electronics Engineers).



ЗМІСТ

Степанов М. Т. Реалізація керування за прогнозом на основі кубічного сплайну в замкнутих САР: концепція побудови і попередній аналіз роботи	4
Волощук В. А., Некрашевич О. В., Ханко А. О. Методи високоефективного людино-машинного інтерфейсу	12
Стопакевич А. О., Стопакевич О. А., Тігарєв А. М., Тігарєва Т. Г. Розробка автономних web-сервісів для імітаційного моделювання динамічних систем в реальному часі на базі моделей MATLAB Simulink	21
Голубєв Л. П., Ківа І. Л., Матяш О. В. Дослідження методів Computer Vision для використання в автоматизованих системах	30
Мороз А. М., Похлебін Н. О., Хобін В. А. Інформаційно-аналітична система приймальної комісії ОНАХТ як основа автоматизованого управління формування контингенту студентів	36
Левінський М. В., Левінський В. М. Моделювання системи регулювання в Step 7 TIA Portal з використанням функціональних блоків бібліотеки «LSim»	43
Антонова А. Р., Помогасв Д. Аналіз основних переваг та недоліків різноманітних фреймворків автоматизації тестування	48

CONTENT

Stepanov M. T. Implementation of forecast control based on cubic spline in closed ACS: concept of construction and preliminary analysis of work	4
Voloshchuk V., Nekrashevych O., Khanko A. Methods of high-performance human-machine interface	12
Stopakevych A., Stopakevych O., Tigariyev A., Tigariyeva T. Development of autonomous web-services for real time simulation of dynamic systems based on MATLAB Simulink models	21
Golubev L., Kiva I., Matyash O. Research of computer Vision Methods for use in automated systems	30
Moroz A., Pokhlebina N., Khobin V. Information and analytical system of the ONAFT admissions committee as the basis for automated management of student contingent formation	36
Levinskyi M., Levinskyi V. Control systems modeling in Step 7 TIA Portal with “LSim” library functional blocks	43
Antonova A., Pomogaiev D. Analysis of the main advantages and disadvantages of various testing automation frameworks	48



УДК [621.867.3:622.612]:658.5

РЕАЛІЗАЦІЯ КЕРУВАННЯ ЗА ПРОГНОЗОМ НА ОСНОВІ КУБІЧНОГО СПЛАЙНУ В ЗАМКНУТИХ САР: КОНЦЕПЦІЯ ПОБУДОВИ І ПОПЕРЕДНІЙ АНАЛІЗ РОБОТИ

Степанов М. Т.

Одеська національна академія харчових технологій, м. Одеса, Україна

ORCID: <https://orcid.org/0000-0003-1297-5537>E-mail: stepanov197818@gmail.com

Copyright © 2020 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>DOI: <https://doi.org/10.15673/atbp.v12i4.1929>

Анотація. У статті розглядається система автоматичного регулювання, що реалізує принцип керування за прогнозом в замкнутому контурі. Об'єкти технологічного типу досить часто мають велику інерційність в каналах регулювання яка пов'язаною не тільки з чистим запізненням, але, більшою мірою з акумуляцією речовини і енергії, так званим ємнісним запізненням. Повна або часткова компенсація цієї інерційності може в значній мірі поліпшити якість регулювання для таких об'єктів. Тому в замкнутий контур системи регулювання пропонується ввести прогнозування на основі кубічного сплайну. Проведений аналіз показує, що окремі ділянки траєкторії руху регульованої змінної можуть бути представлені безперервними, багаторазово диференційованими функціями (наприклад, кубічним сплайном). Для моделі кубічного сплайну знайдені співвідношення, що дозволяють оцінювати його параметри і вести розрахунок прогнозного значення змінної в реальному часі. Проведено структурний та оптимальний параметричний синтез альтернативних варіантів систем автоматичного регулювання з типовим ПІД-регулятором, що реалізують принцип керування за прогнозом і принцип керування за станом. Порівняльний аналіз оптимальних систем, проведений в часовій і частотних областях, показав значну перевагу системи регулювання, що реалізує принцип керування за прогнозом на основі кубічного сплайну. У порівнянні з системою, що реалізує принцип керування за станом, система регулювання з прогнозуванням по кубічному сплайну забезпечує зниження інтегрального і прямих показників якості перехідних процесів до 40%. Перевірка на грубість систем автоматичного регулювання показала, що система автоматичного регулювання з прогнозуванням регульованої змінної за кубічним сплайном має більший запас стійкості, ніж система регулювання що реалізує принцип керування за станом.

Abstract. The article discusses an automatic control system that implements the principle of predictive control in a closed loop. Objects of a technological type quite often have a large inertia in the control channels associated not only with a pure delay, but, to a greater extent, with the accumulation of matter and energy, the so-called capacitive delay. Full or partial compensation of this inertia can significantly improve the quality of regulation for such objects. Therefore, it is proposed to introduce prediction based on a cubic spline into the closed loop of the control system. The performed analysis shows that individual sections of the trajectory of the controlled variable can be represented by continuous, multiply differentiable functions (for example, a cubic spline). For the model of cubic splines, relations are found that allow estimating their parameters and calculating the predicted value of the variable in real time. Structural and optimal parametric synthesis of alternative variants of automatic control systems with a typical PID controller, which implement the principle of control by forecast and the principle of control by state, has been carried out. A comparative analysis of optimal systems, carried out in the time and frequency domains, showed a significant advantage of the control system that implements the principle of predictive control based on a cubic spline. In comparison with a system that implements the principle of state control, a control system with predictive cubic spline provides a decrease in the integral and direct quality indicators of transient processes by up to 40%. Checking for the roughness of automatic control systems showed that an automatic control system with predicting an adjustable variable along a cubic spline has a greater margin of stability than a control system that implements the principle of state control.

Ключові слова: прогнозування, кубічний сплайн, помилка прогнозування, система автоматичного регулювання, запас стійкості.

Keywords: prediction, cubic spline, prediction error, automatic control system, robust control.



Вступ

Основними характеристиками замкнутих САР є якість регулювання і стійкість. Якість регулювання визначається динамічною та статичною точністю САР. Максимально досяжна динамічна точність САР багато в чому обмежується запізненням і інерційністю каналів об'єкта керування. Об'єкти технологічного типу досить часто мають велику інерційність в каналах регулювання, пов'язаної не тільки з чистим запізненням, але, більшою мірою з акумуляцією речовини і енергії, так званим ємнісним запізненням. Повна або часткова компенсація цієї інерційності може в значній мірі поліпшити якість регулювання для таких об'єктів. Тому в замкнутий контур системи регулювання пропонується ввести прогнозування на основі кубічного сплайну і перейти від загальноприйнятого принципу регулювання по поточному значенню вихідної величини $y(t)$ до принципу регулювання за прогнозом $y(t + \tau_{\text{пр}})$. Це означає що обчислення керуючого впливу (положення регулюючого органу) $U(t)$ проводиться не за поточним значенням регульованої величини $y(t)$, а по прогнозованому значенню $y(t + \tau_{\text{пр}})$ у майбутньому.

Аналіз літературних даних і постановка проблеми

Аналіз сучасних підходів [2,3] підтверджує актуальність використання принципу управління за прогнозом в сучасних САР. Прогнозування в САР може здійснюватися за різними законами. В [2, 3] розглянуті варіанти САР, що реалізують принцип замкнутого керування за лінійним прогнозом $y(t)$. Ідеальне (фізично не реалізується) прогнозування має передатну функцію

$$W(p) = e^{+p \tau_{\text{пр}}} \quad (1)$$

Якщо розкласти експонентну функцію (1) в ряд Тейлора

$$e^{+p \tau_{\text{пр}}} = 1 + p \tau_{\text{пр}} + \frac{(p \tau_{\text{пр}})^2}{2!} + \frac{(p \tau_{\text{пр}})^3}{3!} + \dots \quad (2)$$

і обмежитися в (2) першими двома складовими, то отримаємо передавальну функцію найпростішого – лінійного прогнозу:

$$W_{\text{пр}}(p) = 1 + p \tau_{\text{пр}} \quad (3)$$

Передаюча функція (3) представляє ідеальну форсуючу ланку. Її включення в контур зворотного зв'язку для прогнозування $y(t)$ дозволило підвищити динамічну точність САР. Одночасно проведений в [1] аналіз показує, що окремі ділянки траєкторії руху регульованої змінної можуть бути представлені безперервними, багаторазово диференційованими функціями (сплайнами). В якості найбільш ефективного розглядається кубічний сплайн. Для моделей кубічного сплайна знайдені співвідношення, що дозволяють оцінювати їх параметри в реальному часі і вести розрахунок прогнозного значення змінної. Результати досліджень показали, що використання кубічного сплайна дозволяє досить точно прогнозувати значення гармонійних сигналів на інтервалах менших півперіода, а випадкових гауссовських процесів на інтервалах менших одного, двох інтервалів кореляції, при цьому великі інтервали прогнозування відповідають вузькосмуговим, близьким до квазігармонійного випадковим процесам. Отримані результати підтверджують можливість включити обраний сплайн для прогнозування відфільтрованих від шумів регульованих змінних без великих похибок на час порівняння з часом запізнення в каналі керування. Тому в якості альтернативи лінійному прогнозу пропонується вести прогноз на основі кубічного сплайну (полінома третього ступеня). Але це вимагає додаткових досліджень розроблених алгоритмів в ланцюгах САР.

Мета і завдання дослідження

Метою дослідження є синтез і попередній аналіз підвищення динамічної точності та запасу стійкості САР, що реалізує принцип управління за прогнозом на основі кубічного сплайна для об'єктів технологічного типу, коли регульовану змінну можна розглядати як багаторазово диференційовану функцію часу в порівнянні з САР, що реалізує замкнутий принцип управління станом. Для цього, в контур зворотного зв'язку замкнутої САР вводиться математична модель прогнозування регульованої змінної на основі кубічного сплайна. Для досягнення поставленої мети необхідно провести структурний і оптимальний параметричний синтез САР, що реалізує принцип управління станом і САР, що реалізує принцип управління за прогнозом. Після цього провести порівняльний аналіз цих систем.

Методи і матеріали досліджень

Для проведення досліджень в якості основного методу будемо використовувати метод імітаційного моделювання в середовищі Simulink системи Matlab. Для цього на першому етапі розробимо структурну схему САР, визначимо регулятор, віртуальний тестовий ОК (ВТОК), критерій оцінки якості роботи САР і проведемо для моделі кубічного сплайна розрахунок співвідношень, що дозволяють оцінювати його параметри в реальному часі і вести розрахунок прогнозного значення регульованої змінної. Структурна схема САР яка реалізує принцип керування за прогнозом представлена на рисунку 1.

В якості регулятора будемо використовувати ПІД-регулятор з реальним диференціатором. Для проведення оптимального параметричного синтезу САР і оцінки якості її роботи в перехідних режимах скористаємося інтегральним квадратичним показником якості:



$$I = \int_{t_1}^{t_1 + T_{\text{ммо}}} (y^{\text{зд}}(t) - y(t))^2 dt, \quad (4)$$

де $T_{\text{мод}}$ – час моделювання, t_1 – час початку відліку.

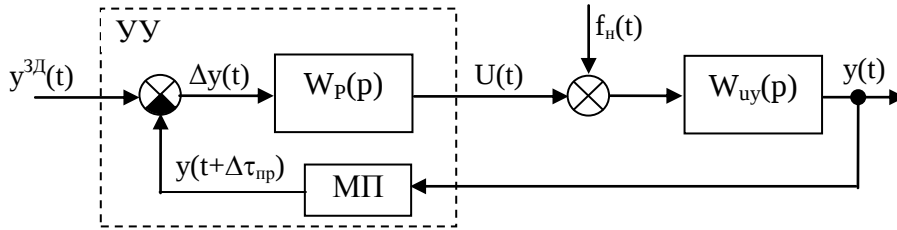


Рис. 1 – Структурна схема САР з модулем прогнозування в її контурі зворотного зв'язку

Модель об'єкта керування розглянемо зі статичними властивостями за каналом керування. Передаточна функція віртуального тестового ОУ (ВТОУ) має наступний вигляд.

$$W_{\text{уу}}^o(p) = \frac{k_o}{(T_1 p + 1)^6 (T_2 p + 1)} \quad (5)$$

де T_1 - постійна часу що характеризує ємнісне запізнення, T_2 - постійна часу, k_o - коефіцієнт передачі ОК. При цьому постійна часу $T_1 \ll T_2$. Для проведення досліджень були прийняті наступні базові значення параметрів ВТОУ: $T_1 = 0.5$, $T_2 = 7$, $k_o = 1.2$.

На рисунку 2 представлена нормована перехідна характеристика ВТОУ по каналу керування з ілюстрацією процедури параметричної ідентифікації моделі першого порядку за методикою Мініної.

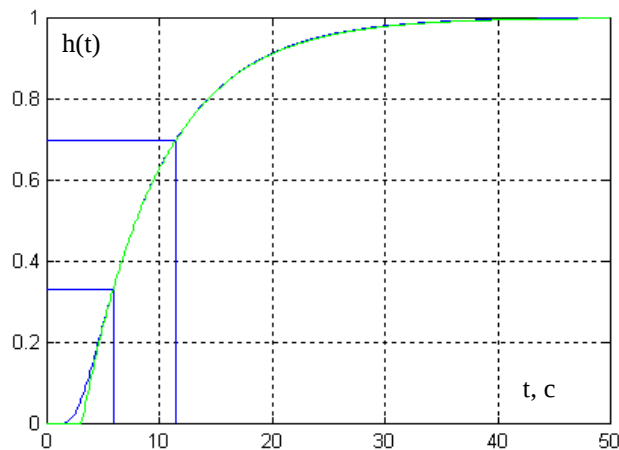


Рис. 2 – Ілюстрація процедури параметричної ідентифікації моделі ОК (методика Мініної) за каналом керування

Передаточна функція моделі першого порядку по каналу керування.

$$W_{\text{уу}}(p) = \frac{1.2e^{-3.1p}}{7p + 1} \quad (6)$$

Модуль прогнозування по кубічному сплайну будемо реалізовувати на основі наступного виразу:

$$y(t + \Delta\tau_{\text{пр}}) = \bar{y}(t) + d(t)(t + \Delta\tau_{\text{пр}})^3 + c(t)(t + \Delta\tau_{\text{пр}})^2 + b(t)(t + \Delta\tau_{\text{пр}}) + a(t), \quad (7)$$

$$\Delta\tau_{\text{пр}} \in [0, \tau_{\text{пр}}].$$

Тоді, якщо вдасться знайти значення трьох похідних цих оцінок в момент часу $t = t_0$, то оцінки значень коефіцієнтів в (7) визначаються з простих і зручних для розрахунку в реальному часі співвідношень:

$$d(t_0) = \ddot{y}(t_0)/6,$$

$$c(t_0) = (\ddot{y}(t_0) - \ddot{y}(t_0)t_0)/2,$$



$$b(t_0) = \dot{y}(t_0) - \ddot{y}(t_0)t_0 + \ddot{y}(t_0)t_0^2/2,$$

$$a(t_0) + \bar{y}(t_0) = y(t_0) - \dot{y}(t_0)t_0 + \ddot{y}(t_0)t_0^2/2 - \ddot{y}(t_0)t_0^3/6, \quad (8)$$

а прогнозовані на τ_{np} вперед значення необхідних оцінок розраховуються з виразів:

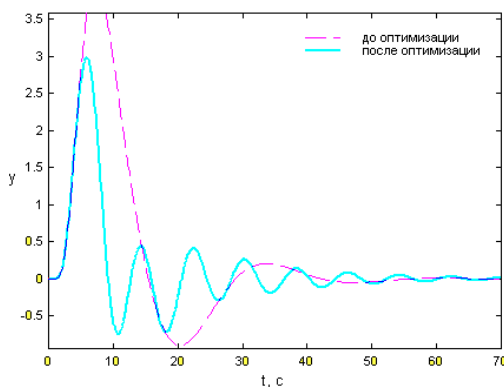
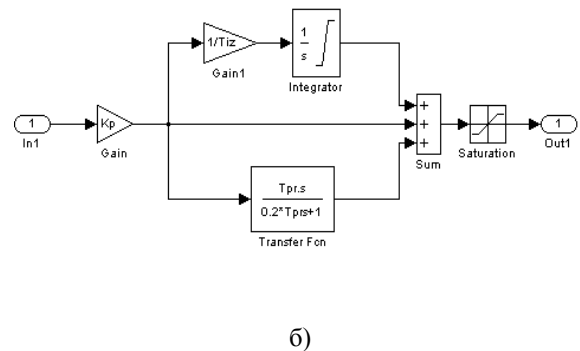
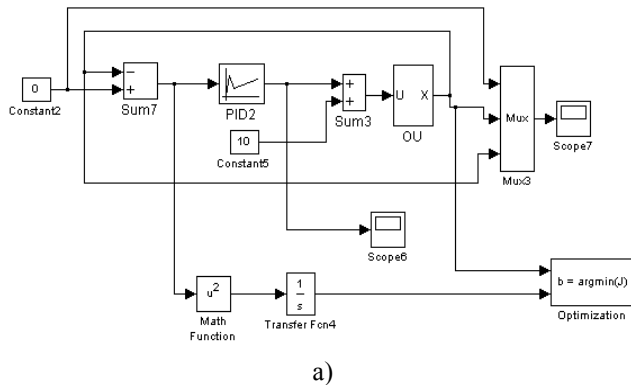
$$y(t_0 + \tau_{np}) = d(t_0)(t_0 + \tau_{np})^3 + c(t_0)(t_0 + \tau_{np})^2 + b(t_0)(t_0 + \tau_{np}) + a(t_0) + \bar{y}(t_0), \quad (9)$$

Необхідною умовою роботи алгоритму, є щоб $y(t)$ була триразово диференційованою. Ця умова цілком здійсненна, так як змінні на виході ОК, відфільтровані від шумів, як правило, є багато разів диференційованими.

У розглянутому сплайні розрахунок коефіцієнтів ведеться в припущенні, що в момент t_0 відомі значення самої змінної і всіх необхідних похідних. У разі цифрової реалізації алгоритму керування, а саме вона передбачається як основна, похідні обчислюються за значеннями решітчастої функції $f_k(nT_{KB})$, $n = \overline{0, \infty}$, з кроком квантування T_{KB} .

Для проведення порівняльного аналізу роботи САР введемо такі позначення: САРП_Л - САР реалізує принцип управління за прогнозом на основі лінійного прогнозу (3); САРП_К - САР реалізує принцип управління за прогнозом на основі кубічного сплайну (9).

На першому етапі досліджень проведемо оптимальний параметричний синтез (ОПС) САР за критерієм (4) при фіксованих параметрах моделі ОУ без модулів прогнозування. А потім в замкнутий контур отриманої САР з оптимальними параметрами регулятора введемо модулі прогнозування з лінійним прогнозом і прогнозом по кубічному сплайну. Схеми моделювання САР і результати ОПС представлені на рисунку 3.



Имя переменной	Оптимальное значение	Нижний предел	Начальное приближение	Верхний предел	Значение критерия оптимальности:
kr1	3.5382	0.2	1.5383	20	до - 92.0435
Tiz1	3.6038	0.2	3.6076	25	после - 30.9073
Tpr1	1.7325	0.5	1.7323	20	
					Количество шагов процедуры оптимизации:
					максимальное - 700
					фактическое - 98

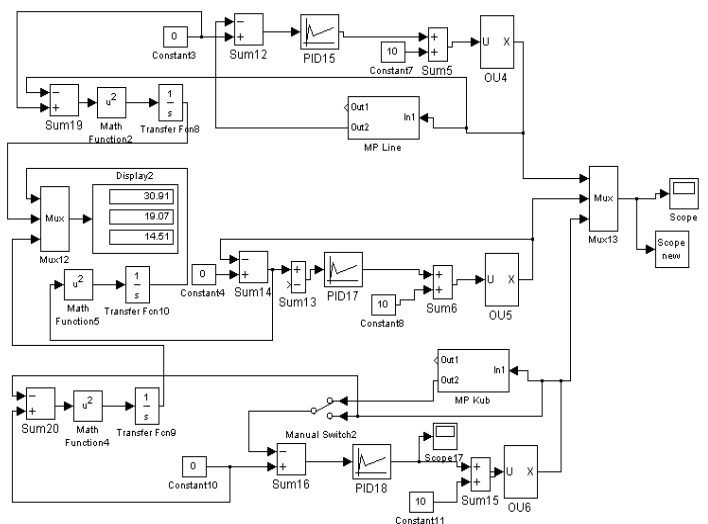
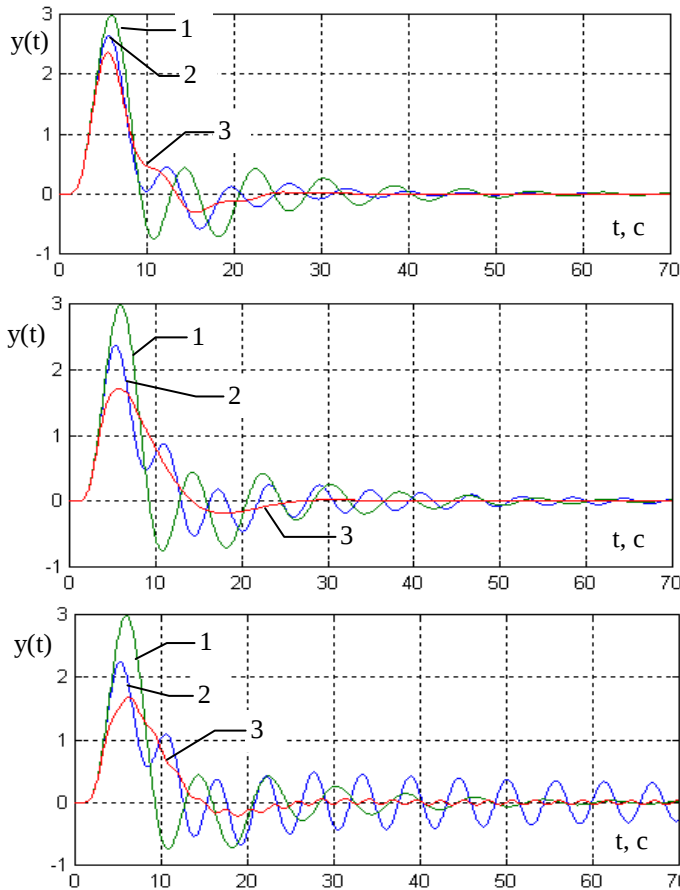


Рис. 3 – Схема моделювання САР (а) з розкритою підсистемою ПІД регулятора (б), результати її оптимального параметричного синтезу (в) при фіксованих значеннях параметром моделі ВТОУ і схема моделювання САР з доданими модулями прогнозування (г)

**Результати досліджень та їх обговорення**

Введення в замкнутий контур САР оптимальної за критерієм (4) модуля з лінійним законом прогнозування підвищує динамічну точність САР, але не забезпечує значного підвищення запасу її стійкості в порівнянні з прогнозуванням по кубічному сплайну. На рисунку 4 представлені перехідні характеристики САР для різних значень інтервалу прогнозування $\Delta t_{\text{пр}}$. З рисунків видно, що збільшення інтервалу прогнозування (до певної межі) в системі з кубічним сплайном призводить до підвищення динамічної точності (зменшується Δy^{max}) і підвищення запасу стійкості в порівнянні з базовою САР і САР з лінійним законом прогнозування.



Вариант САР	Δy^{max}	I
1. САР	2.98	30.9
2. САРП_Л ($\Delta t_{\text{пр}}=0.5$)	2.64	22.1
3. САРП_К ($\Delta t_{\text{пр}}=0.7$)	2.36	18.8

Вариант САР	Δy^{max}	I
1. САР	2.98	30.9
2. САРП_Л ($\Delta t_{\text{пр}}=0.9$)	2.37	19.1
3. САРП_К ($\Delta t_{\text{пр}}=1.4$)	1.71	14.5

Вариант САР	Δy^{max}	I
1. САР	2.98	30.9
2. САРП_Л ($\Delta t_{\text{пр}}=1.1$)	2.25	21.9
3. САРП_К ($\Delta t_{\text{пр}}=1.6$)	1.68	13.9

Рис. 4 – Перехідні характеристики САР при різних значеннях інтервалу прогнозування (1 - САР, 2-САРП_Л, 3 – САРП_К)

Приклад розрахунку запасів стійкості за коефіцієнтом передачі і часом запізнення в моделі ОУ для випадку коли інтервал прогнозування «квазіоптимальний» за критерієм (4) представлений на рисунку 5. Для системи оптимальної за критерієм (4) з коливальними перехідними процесами введення лінійного алгоритму прогнозування дає великі помилки, і фактично, модуль прогнозування стає підсилювачем сигналу, підвищуючи коефіцієнт передачі в замкнутому контурі, що підтверджується частотними характеристиками, представленими на рисунку 6.

Зі збільшенням часу прогнозування ситуація тільки погіршується і запас стійкості САР по коефіцієнту передачі ОК знижується. Доводиться знижувати інтервал прогнозу, що одночасно призводить до зниження динамічної точності САР. В цілому можна відзначити, що введення в контур зворотного зв'язку САР з високою частотою коливань власного руху (≈ 0.9 рад / сек) лінійного модуля прогнозування не забезпечує одночасного підвищення динамічної точності САР і підвищення її запасу стійкості (зокрема за коефіцієнтом передачі об'єкта) в порівнянні з використанням модуля прогнозування за кубічним сплайном. Слід також зазначити, що в системі з прогнозуванням по кубічному сплайну збільшення інтервалу прогнозу спочатку підвищує запас стійкості системи і за коефіцієнтом передачі і за часом запізнення, а при подальшому його збільшенні запас стійкості за коефіцієнтом передачі зменшується аж до нестійкості системи (рис.7). Цю особливість потрібно враховувати при введенні модуля прогнозування в систему без її перенастроювання.

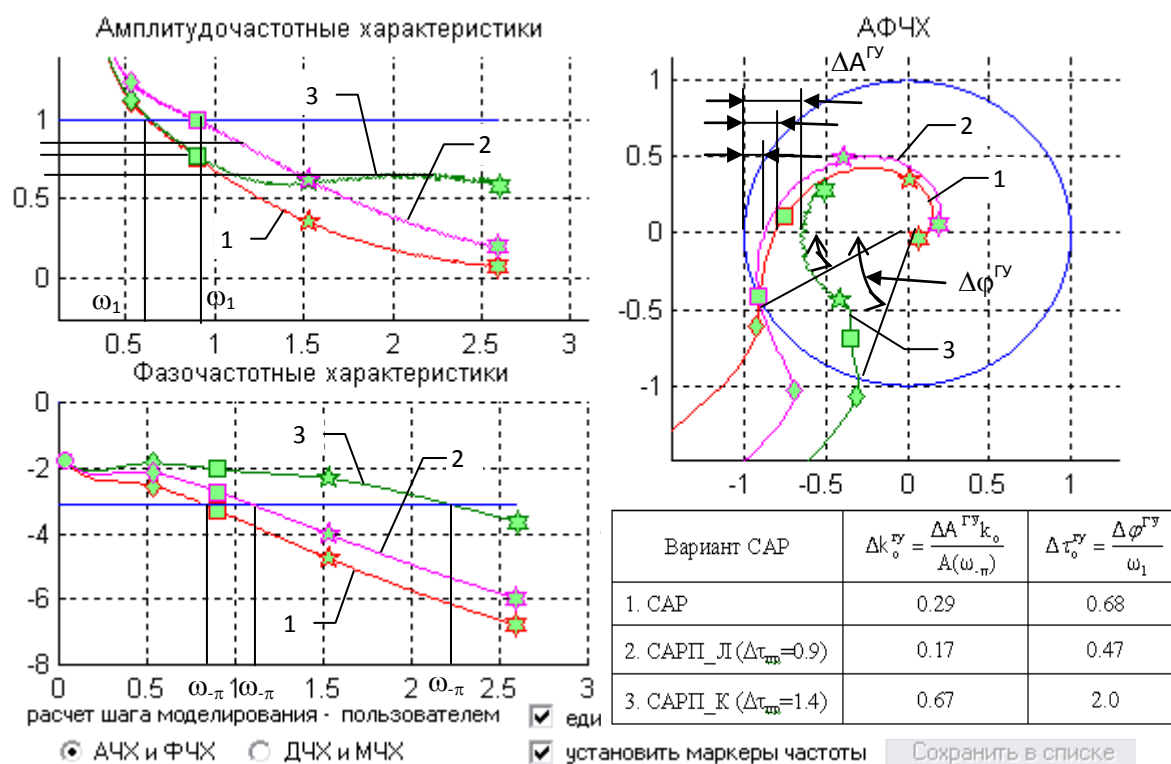


Рис. 5 – Частотні характеристики розімкнутих САР з розрахунком значень їх запасів стійкості за коефіцієнтом передачі $\Delta k_o^{\gamma y}$ і часом запізнення $\Delta \tau_o^{\gamma y}$ в моделі ОК (1 - САР, 2-САРП_Л, 3 – САРП_К)

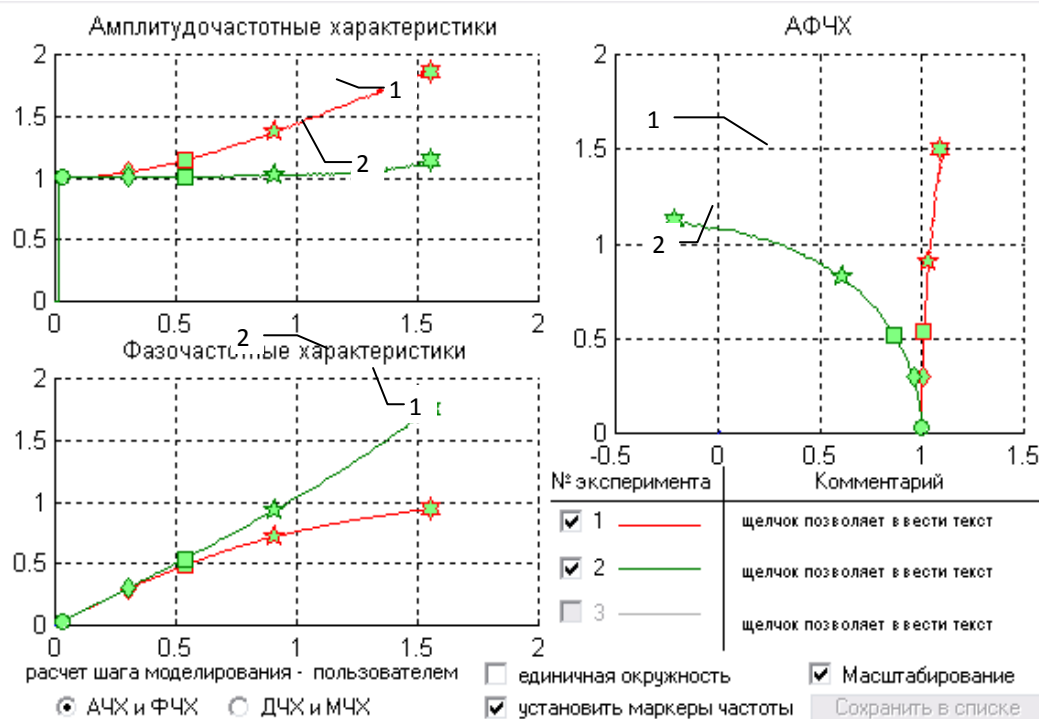


Рис. 6 – Частотні характеристики модулів прогнозування лінійного (1) і по кубічному сплайну (2) при однакових інтервалах прогнозування 1 с

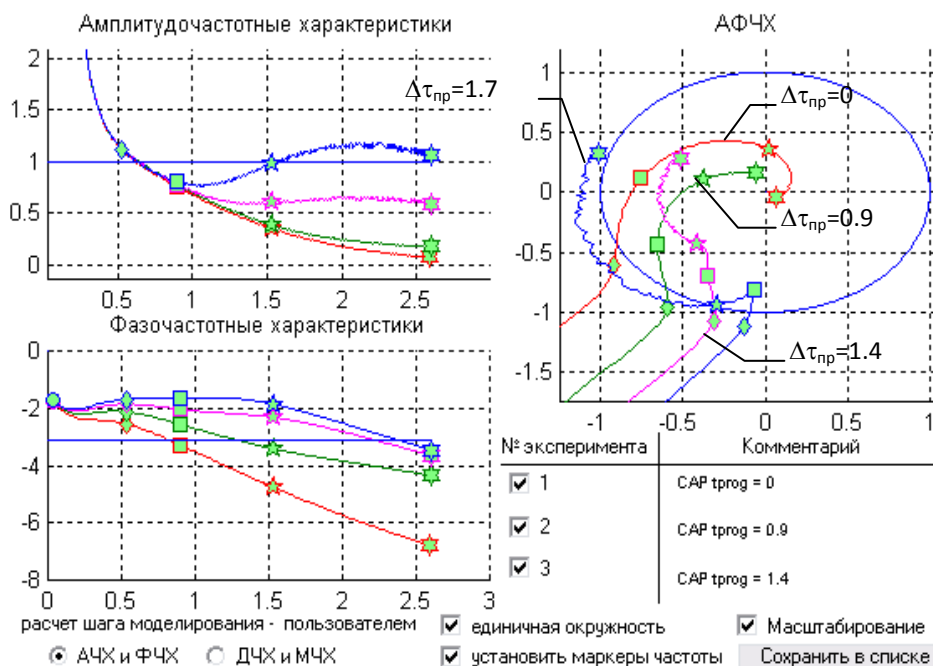


Рис. 7 – Частотні характеристики розімкнутої САР з прогнозуванням по кубічному сплайну, у які ілюструють залежність запасу стійкості САР по амплітуді і фазі в залежності від заданого інтервалу прогнозу $\Delta\tau_{пр}$

Отримані результати є попередніми і не дозволяють зробити висновок про перевагу одного з варіантів систем. Для більш точної оцінки ефективності роботи розглянутих варіантів систем проведемо їх ОПС в однакових умовах і проаналізуємо їх роботу.

Об'єкти технологічного типу мають значний рівнем невизначеності і нестационарності їх параметрів. Тому оптимальний параметричний синтез САР будемо проводити в умовах параметричної невизначеності моделі ОК. В цьому випадку параметри моделі ОУ будуть задано не фіксованими значеннями, а діапазонами їх зміни $k_0 \in [0.96...1.44]$, $T_1 \in [0.4...0.6]$. ОПС в умовах параметричної невизначеності моделі ОК дозволить забезпечити стійкість САР для заданого діапазону зміни параметрів ОУ і приблизно однаковий запас стійкості САР за коефіцієнтом передачі і часу запізнювання. На рисунках 8-10 представлені результати ОПС САР без прогнозування і з прогнозуванням, а також результати перевірки цих систем на грубість, що підтверджують їх стійкість і працездатність в заданому діапазоні зміни параметрів ОК.

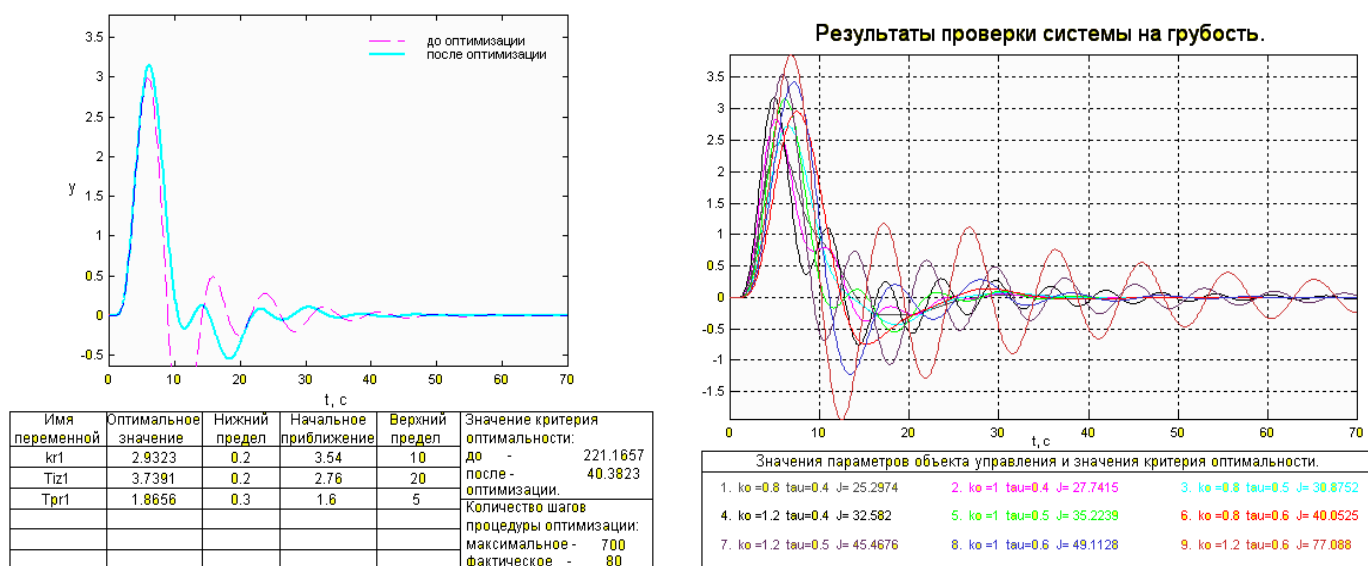


Рис. 8 – Результати оптимального параметричного синтезу і перевірки грубості САР з ПІД регулятором, яка реалізує принцип управління за відхиленням

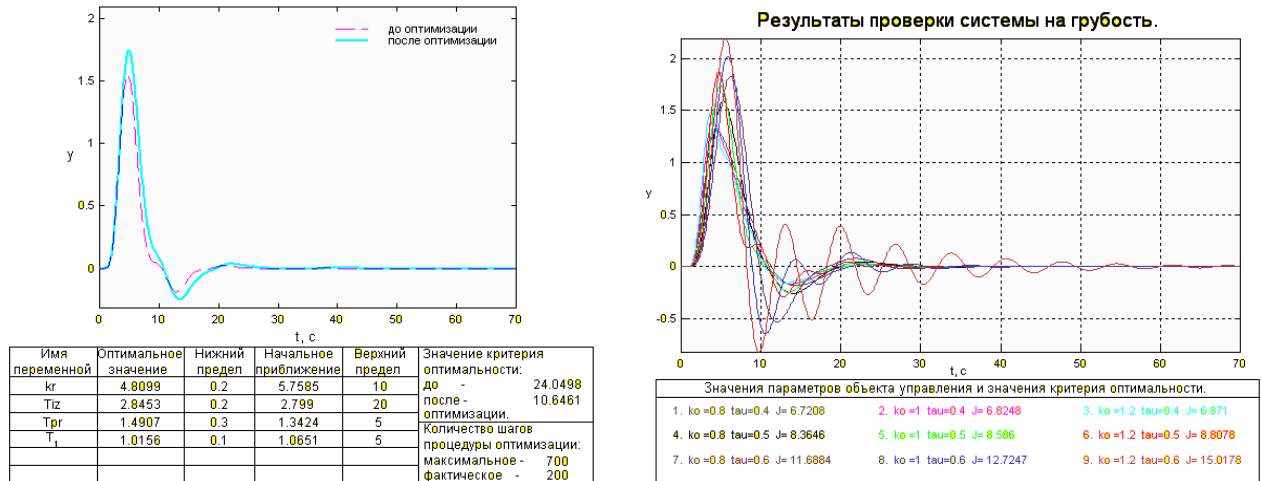


Рис. 9 – Результаты оптимального параметричного синтеза і перевірки грубості САР з ПІД регулятором, яка реалізує принцип управління за прогнозом на основі кубічного сплайну

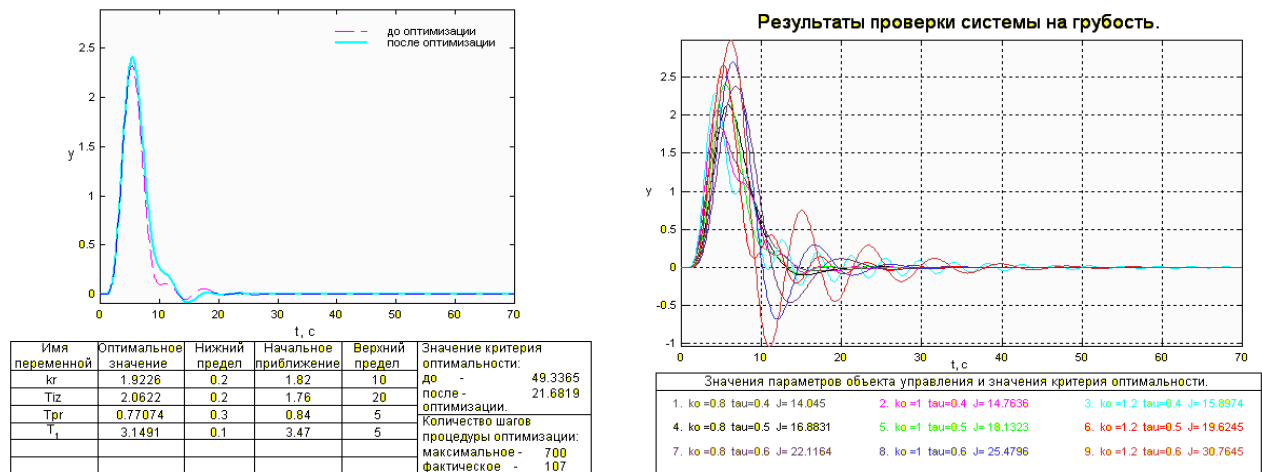


Рис. 10 – Результати оптимального параметричного синтезу і перевірки грубості САР з ПІД регулятором, яка реалізує принцип управління за прогнозом на основі лінійного прогнозування

На рисунку 11 наведено перехідні характеристики оптимальних САР для номінальних значень параметрів моделі ОК і значень параметрів, при яких інтегральний квадратичний критерій приймає мінімальне та максимальне значення. У всіх випадках САРП_К працює краще забезпечуючи зниження максимального динамічного відхилення більш ніж на 40% та інтегрального показника якості у 3-4 рази в порівнянні з базовою САР, яка реалізує принцип керування за відхиленням.

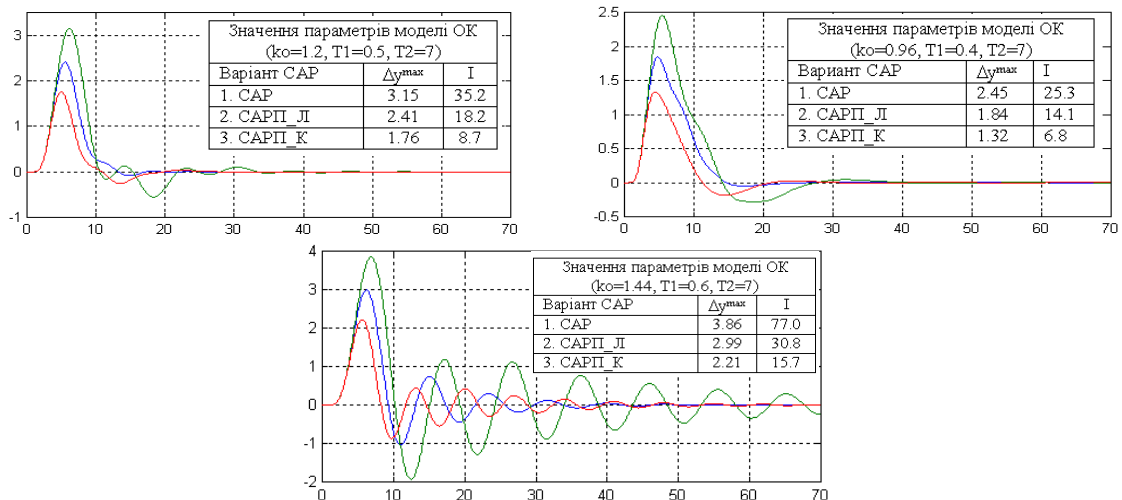


Рис. 11 – Перехідні характеристики САР при різних значеннях параметрів моделі ОК



Висновки

1. Введення в замкнуту контур регулювання модулю прогнозування за кубічним сплайном забезпечує підвищення динамічної точності САР та її запасу стійкості за коефіцієнтом передачі та часом запізнення. Для тестової САР, яка оптимальна за інтегральним квадратичним показником якості, при збільшенні інтервалу прогнозування в замкнутому контурі поступово збільшується динамічна точність та запас стійкості, при подальшому збільшенні інтервалу прогнозування відбувається збільшення еквівалентного коефіцієнта передачі модуля прогнозування, підвищення динамічної точності уповільнюється, а запас стійкості системи за коефіцієнтом передачі зменшується і система втрачає стійкість. Це потрібно враховувати при налаштуванні інтервалу прогнозування і реалізувати його само налагодження у системі. Також потрібно враховувати можливість зниження часу запізнення в каналі управління і розраховувати інтервал прогнозування виходячи з його мінімальних значень. Результати моделювання показали що введення модулю прогнозування в САР без пере налаштування параметрів регулятора дозволяє знизити максимальне динамічне відхилення на 30-35%, інтегральний показник якості до 2 разів при значному підвищенні запасу стійкості.

2. Введення в контур регулювання САР, яка оптимальна за інтегральним квадратичним показником якості модулю прогнозування за лінійним законом значних покращень не забезпечило. Підвищення динамічної точності при збільшенні інтервалу прогнозування супроводжувалось швидким зниженням запасу стійкості САР. Частково це можна пояснити властивостями АЧХ (рис.6) модулю прогнозування який на частоті власного руху системи посилює сигнал значно більше ніж модуль прогнозування на основі кубічного сплайну. Тому налаштування інтервалу прогнозування потрібно проводити у сукупності з параметрами регулятора.

3. Результати оптимального параметричного синтезу проведеного в умовах параметричної невизначеності ОК підтвердили перевагу САР з прогнозуванням за кубічним сплайном перед САР з лінійним прогнозуванням та базовою САР. САР з прогнозуванням за кубічним сплайном забезпечує зниження максимального динамічного відхилення більш ніж на 40% та інтегрального показника якості у 3-4 рази в порівнянні з базовою САР яка реалізує принцип керування за відхиленням.

Список використаних джерел

- [1] Степанов М.Т., Хобин В.А. Прогнозирование вынужденного движения и его применение в системах гарантирующего управления // Автоматизация технологических и бизнес-процесов. – Одеса, 2011. – № 5-6. – С. 20-25
- [2] Пикина Г.А., Кузнецов М.С. Синтез линейных прогностических алгоритмов регулирования // Новое в российской электроэнергетике. – 2009. – № 10. – С. 40-44.
- [3] Пикина Г.А., Кузнецов М.С. Прогностические типовые алгоритмы регулирования // Теплоэнергетика. – 2011. – № 4. – С. 61-66.
- [4] Степанов М.Т., Хобин В.А. Система автоматического регулирования инвариантна до контролируемых возмущений з прогнозуванням сигналу корекції по кубічному сплайну // Автоматизация технологических и бизнес-процесов. – Одеса, 2020. — Т. 12, № 1. – С. 67-74.

References

- [1] M.T. Stepanov et al., "Prognostirovanie vyinuzhdennoho dvizheniya i ego primeneniye v sistemah garantiruyushchego upravleniya," *Automation of technological and business processes*, no.5-6, pp.20-25, 2011.
- [2] G.A. Pikina et al. "Sintez lineynykh prognosticheskikh algoritmov regulirovaniya," *Novoe v rossiyskoy elektroenergetike*, vol. 10, pp. 40-44, 2009.
- [3] G.A. Pikina et al. "Prognosticheskie tipovyye algoritmy regulirovaniya," *Teploenergetika*, vol. 4, pp. 61-66, 2011.
- [4] M.T. Stepanov et al. "Automatic control system invariant to controlled perturbations with forecasting of correction signal by cubic spline," *Automation of technological and business processes*, vol 12, iss.1, pp. 64-70, 2020.

УДК 621.577.4

МЕТОДИ ВИСОКОЕФЕКТИВНОГО ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ

Волощук В. А.¹, Некрашевич О. В.², Ханко А. О.³

^{1,2,3} Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

ORCID: ¹ <http://orcid.org/0000-0003-0687-8968>, ² <http://orcid.org/0000-0003-2263-3549>,

³ <http://orcid.org/0000-0002-8682-9084>

E-mail: ¹ vl.volodya@gmail.com, ² olena.nekrashevych@gmail.com, ³ bismarck195@gmail.com

Copyright © 2020 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i4.1930>



Анотація. Дизайн людино-машинного інтерфейсу відіграє вирішальну роль у визначенні здатності оператора ефективно керувати процесом, особливо у відповідь на нештатні ситуації. По ряду причин поточна архітектура та можливості більшості людино-машинних інтерфейсів далеко не оптимальні для роботи з операторами. Розробка класичної системи диспетчерського управління та збору даних, як правило, починалась з детальної діаграми трубопроводів та контрольно-вимірювальних приладів, як основи дисплеїв операторів при створенні людино-машинного інтерфейсу. Як виявилось існує багато процесів для яких такий підхід не бажаний, а в деяких випадках взагалі є неприпустимим і призводить до дуже перевантаженого та заплутаного зображення без достатнього акценту на інформації, необхідній операторам для ефективного виконання своїх завдань. Як результат, галузі, з сучасними системами управління, зараз використовують примітивний людино-машинний інтерфейс, створений десятиліття тому, у той час, коли було мало знань про належні практики та принципи розробки. Метою цієї роботи є висвітлення принципів правильного та ефективного дизайну графіки, що використовується в сучасних системах диспетчерського управління та збору даних, а також виявлення неприпустимих практик, що згубно впливають на сприйняття інформації оператором. Як виявилось для розробки ефективної графіки елементи інтерфейсу не варто використовувати на власний розсуд, необхідно слідувати деяким правилам, які тлумачить, що є корисним та потрібним, а що може перешкодити правильному сприйняттю інформації. Натомість в порівняльному аналізі використання високопродуктивної графіки доводить свої переваги. Концепція високопродуктивної графіки довела свою актуальність, її принципи та можливості проектування слугують для створення людино-машинного інтерфейсу, що забезпечує чітке та зрозуміле усвідомлення ситуації операторами. Ефективний людино-машинний інтерфейс повинен надавати операторам необхідну інформацію у формі, яку вони можуть швидко зрозуміти, щоб приймати правильні рішення. Новий стандарт ISA-101 описує весь життєвий цикл людино-машинного інтерфейсу від задуму до експлуатації, що допомагає інтеграторам краще зрозуміти, що означають ці концепції та як їх застосовувати на практиці. Подальший розвиток стандарту та його імплементація на працюючих підприємствах є перспективним завданням для вчених та інженерів. Стаття носить оглядовий характер та призначена для систематизації знань, по даній тематиці та висвітлення нових перспективних напрямків досліджень.

Abstract. The design of the human-machine interface plays a crucial role in determining the operator's ability to effectively manage the process, especially in response to abnormal situations. For a number of reasons, the current architecture and capabilities of most human-machine interfaces are far from optimal for working with operators. The development of the classical system of dispatch control and data collection, as a rule, began with a detailed diagram of pipelines and control and measuring devices, as the basis of operators' displays when creating a human-machine interface. As it turned out, there are many processes for which this approach is not desirable, and in some cases is unacceptable and leads to a very overloaded and confusing image without sufficient emphasis on the information needed by operators to effectively perform their tasks. As a result, industries with modern control systems now use a primitive human-machine interface created decades ago, at a time when there was little knowledge of good practices and design principles. The purpose of this work is to highlight the principles of correct and efficient graphics design used in modern control systems and data collection, as well as to identify unacceptable practices that adversely affect the perception of information by the operator. As it turned out, to develop effective graphics, the interface elements should not be used at its discretion, it is necessary to follow some rules that explain what is useful and necessary, and what may interfere with the correct perception of information. Instead, in comparative analysis, the use of high-performance graphics proves its advantages. The concept of high-performance graphics has proven its relevance, its principles and design capabilities are used to create a human-machine interface that provides a clear and understandable understanding of the situation by operators. An efficient human-machine interface must provide operators with the necessary information in a form that they can quickly understand in order to make the right decisions. The new ISA-101 standard describes the entire lifecycle of the human-machine interface from design to operation, helping integrators better understand what these concepts mean and how to apply them in practice. Further development of the standard and its implementation in operating enterprises is a promising task for scientists and engineers. The article is an overview and is designed to systematize knowledge on this topic and highlight new promising areas of research.

Ключові слова: високоефективний HMI, SCADA, інтерфейс, ISA-101, людино-машинний інтерфейс

Keywords: high-performance HMI, SCADA, interface, ISA-101, human-machine interface

Вступ

Людино-машинний інтерфейс (HMI) – це сукупність екранів, мнемосхем, кнопок, слайдерів та інших технологій, що використовуються оператором для моніторингу технологічного процесу та взаємодії з системами диспетчерського управління та збору даних (SCADA). Саме дизайн HMI відіграє вирішальну роль у визначенні здатності оператора ефективно керувати процесом, особливо у відповідь на нештатні ситуації [1].

З деяких причин поточна архітектура та можливості більшості HMI далеко не оптимальні для виконання складних операцій. Більшість із них складаються просто із схематичної графіки, що супроводжується цифрами. Такі дисплеї надають великі обсяги необроблених даних, майже не мають потрібної інформації та забезпечують неадекватне усвідомлення ситуації оператором.



Екскурс в історію появи систем візуалізації та супервізорного контролю допоможе краще зрозуміти коріння проблеми. До появи складних цифрових систем управління НМІ оператора зазвичай представляв собою щит керування, який забезпечував повний огляд всіх показників роботи об'єкта та навчений оператор міг бачити всю обстановку майже з першого погляду.

Недоліками цих систем було те, що їх було дуже важко модифікувати. Було проблематичним додавання додаткових можливостей та майже не існувала можливість передавати та аналізувати їх данні. Саме тому їх замінили сучасні SCADA-системи, які включали можливість створювати та відображати графіку, що полегшило контроль над об'єктом управління. Однак на той час не було чітких вказівок щодо того, як правильно створювати ефективну графіку. Перші розробники створювали графіку, що імітувала схематичні мнемосхеми, насамперед тому, що вони були легко доступні. Як результат, галузі з сучасними системами управління, зараз використовують примітивний НМІ, створений десятиліття тому.

У міру розвитку виробниками апаратного забезпечення SCADA-графіки почали застосовуватися дуже кричущі практики дизайну. Результатом стали мнемосхеми, які є неоптимальними для операторів через обмеженість колірної палітри, що використовувалась непослідовно та переповненість цифрами дисплеїв. Тому можна виділити наступні конкретні елементи графічного інтерфейсу, які несприятливо впливають на сприйняття інформації [2, с. 8]:

- виключно схематичне зображення технологічного процесу;
- велика кількість необроблених числових даних;
- замала кількість трендів;
- надмірне використання анімацій та подібних відволікаючих елементів;
- яскраві тривимірні об'єкти;
- висока деталізація зображень обладнання;
- спроби розфарбувати вміст трубопроводів;
- велика виноска в одиницях вимірювання;
- яскраві кольорові зображення рівня рідини, що відображаються на всю ширину резервуару;
- багато ліній перетину та невідповідний напрямок потоку;
- невідповідне кольорове кодування та неправильне використання кольорів, пов'язаних із сигналізацією;
- обмежена та заплутана навігація по сторінках;
- відсутність ієрархії відображення інформації.

Натомість високопродуктивна графіка базується на таких методах побудови інтерфейсу:

- видалення схематичного зображення процесу, за винятком випадків, коли це функціонально важливо;
- обмежене використання кольору (колір використовується дуже конкретно та послідовно);
- сірий фон для мінімізації відблисків;
- відсутність анімації, за винятком специфічної графічної поведінки, пов'язаної з сигналізацією;
- вбудовані в мнемосхему, правильно відформатовані тренди важливих параметрів;
- аналогове представлення вимірювань важливих технологічних параметрів із відображенням їх значень відносно нормальних та нештатних діапазонів;
- правильна ієрархія відображуваного вмісту, що забезпечує поступове викладення детальної інформації;
- малоконтрастні зображення в 2D, а не в 3D;
- логічні та послідовні методи навігації;
- послідовне зображення та компонування потоків для мінімізації перетину ліній;
- методи мінімізації помилок введення даних оператора;
- заходи перевірки та безпеки.

Мета роботи

Метою роботи є дослідження принципів побудови людино-машинного інтерфейсу, що передбачає обґрунтування переваг високопродуктивної графіки та систематизації знань по даній тематиці з висвітленням нових перспективних напрямів досліджень.

Викладення основного матеріалу

Основною відмінністю високопродуктивної графіки є принцип того, що, де це можливо, робочі значення відображаються в інформаційному контексті, а не просто як необроблені числа, розкидані по екрану. Параметри можуть бути представлені набором згрупованих аналогових показників, як на рисунку 1.

Аналоговий показник – це інструмент візуалізації, інтуїтивно зрозумілий для людей, що відображає поточні значення параметра та межі його допустимих значень. Серією коротких поглядів оператор може бути в повній мірі ознайомлений з поточним станом усього технологічного процесу.

Знання про те, що є нормальним, вбудовується в сам НМІ, що полегшує навчання та виявлення нештатної ситуації – до того моменту, як спрацює сигналізація, що є вкрай бажано [2, с. 10].

Багато помилкових рішень в інтерфейсі пов'язано з кольором. Не всі люди здатні однаково добре сприймати яскраві кольори. З цієї причини існує добре відомий принцип використання кольору – колір сам по собі не використовується як єдиний розрізнявач важливої умови або статусу [3, с. 7]. Більшість графічних елементів у всьому

світі порушують цей принцип, тому бажано використовувати додаткове кодування інформації за допомогою методів, відмінних від кольору. Але якщо говорити в першу чергу про колір, то необхідно розробити кольорову палітру з послідовним використанням обмеженої кількості кольорів, що добре розрізняються оператором.

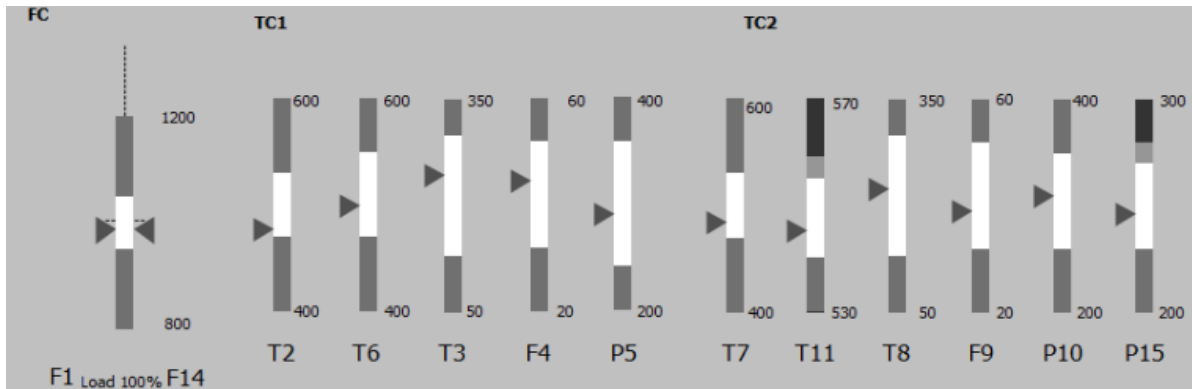


Рис. 1 – Аналогові індикатори
Fig. 1 – Analog indicators

Яскраві кольори в основному повинні використовуватися для акцентування уваги виключно до нештатних ситуацій. Екрани, що відображають нормальну роботу, не повинні покриватися яскраво насиченими кольорами, такими як червоний або зелений. Відносна яскравість об'єкта на схемі показує його статус, разом із словом біля нього. Об'єкти яскравіші від фону - працюють, а ті що темніші від фону – вимкнені, як на рисунку 2 [2, с. 11].

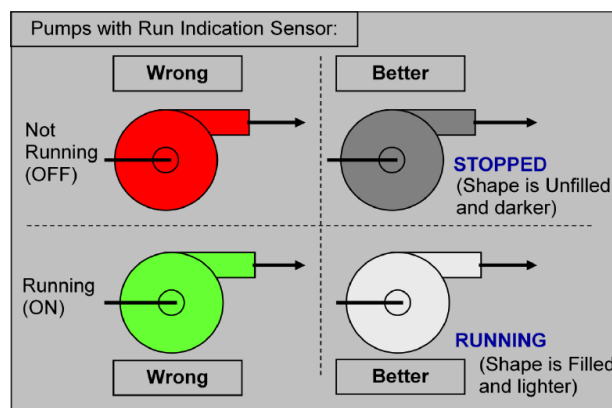


Рис. 2 – Надмірне кодування та правильне використання кольорів [2]
Fig. 2 – Excessive coding and correct use of colors

Правильне відображення сигналізації також повинно бути закодовано по пріоритету (колір / форма / текст). Кольори сигналізації не слід використовувати для функцій, не пов'язаних із сигналізацією [2, с. 12]. Коли значення виходить за нормальний діапазон, поряд з ним з'являється окремий індикатор сигналізації (див. рис. 3). Індикатор миготить, поки сигнал не підтверджено та перестає після підтвердження, але залишається відображеним, доки діє сигнал тривоги [3, с. 9].



Рис. 3 – Зображення сигналізації
Fig. 3 – Image alarms



Люди погано виявляють зміну кольору в периферійному зорі, але рух, такий як миготіння, легко виявляється. Таким чином, сигналізація легко виділяється на графіці та виявляється з першого погляду.

Сьогодні найбільш нагальною проблемою НМІ є загальна відсутність належним чином реалізованих трендів (див. рис. 4). Як правило, за замовчуванням графіки є прихованими та надаються операторові на вимогу, тобто для правильного масштабування діапазону тренду може знадобитися від 10 до 20 натискань / виділень, що сильно впливає на час реакції оператора. Тому тренди важливих параметрів повинні бути вбудованими в мнемосхему та відображатися щоразу, коли мнемосхема викликається.

В свою чергу, рівень в резервуарах не повинен відображатися, як великий об'єм контрастного кольору. Натомість, кращою ідеєю є зображення індикатора, що показує близькість до меж сигналізації. Проте саме комбінація зображень, трендів та аналогових індикаторів є найкращим варіантом [4, с. 23] (див. рис. 5).

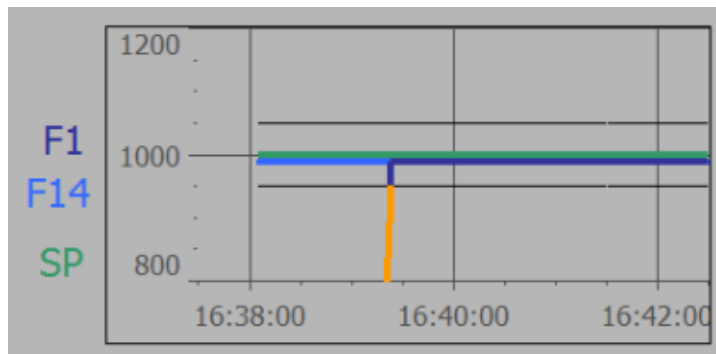


Рис. 4 – Тренди із зображенням допустимих діапазонів
Fig. 4 – Trends with the image of acceptable ranges

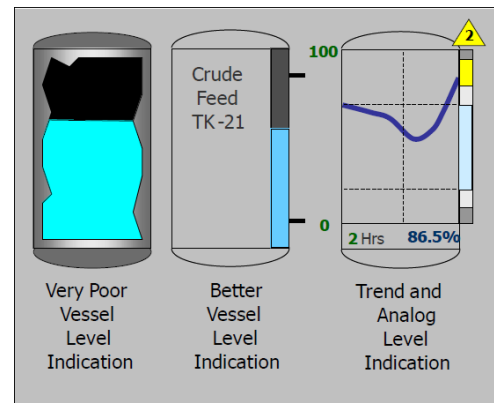


Рис. 5 – Відображення рівня в резервуарах [2]
Fig. 5 – Level display in tanks [2]

Людське око краще виявляє наявність чогось, ніж його відсутність [2, с. 16]. Це твердження знаходить підтвердження при відображенні відносних положень виконавчих механізмів. При цьому аналогове представлення краще, ніж просто цифри. Такий спосіб відображення поступається використанню рухомих елементів з вказівником, оскільки коли значення стовпчика стає низьким, представлення положення на шкалі зникає. Приклад наведено на рисунку 6.

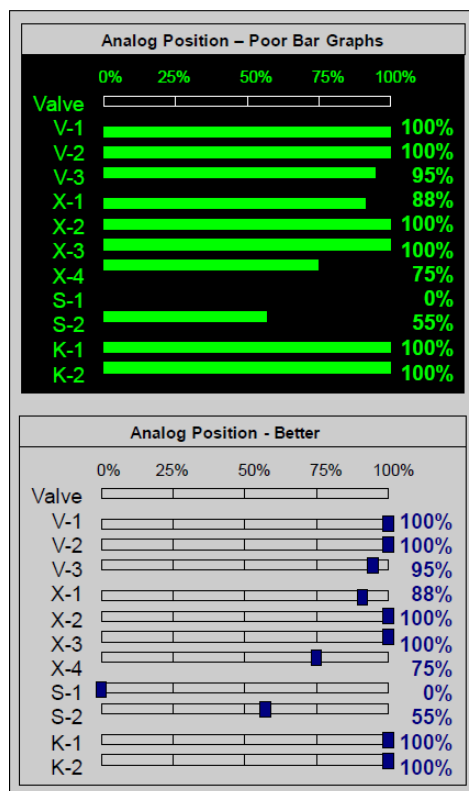


Рис. 6 – Використання шкали з вказівником замість суцільної заливки [2]
Fig. 6 – Using a scale with a pointer instead of a solid fill [2]



Для забезпечення оптимального усвідомлення ситуації та контролю дисплеї повинні мати ієрархічну структуру (див. рис. 7), яка складається з чотирьох рівнів, що забезпечують поступове висвітлення деталей. Від загального (показників ефективності) до конкретного (обладнання/діагностична інформація) [3, с. 6].

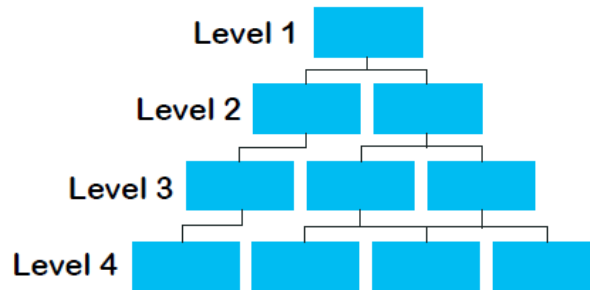


Рис. 7 – Ієрархічна структура вікон з графікою [3]
Fig. 7 – The hierarchical structure of the graphics window [3]

Дисплей 1-го рівня (вікно огляду підприємства/виробництва) використовується для огляду основних параметрів процесу та сигналізацій [5, с. 2]. Як правило він забезпечує відображення ключових показників ефективності (див. рис. 8).

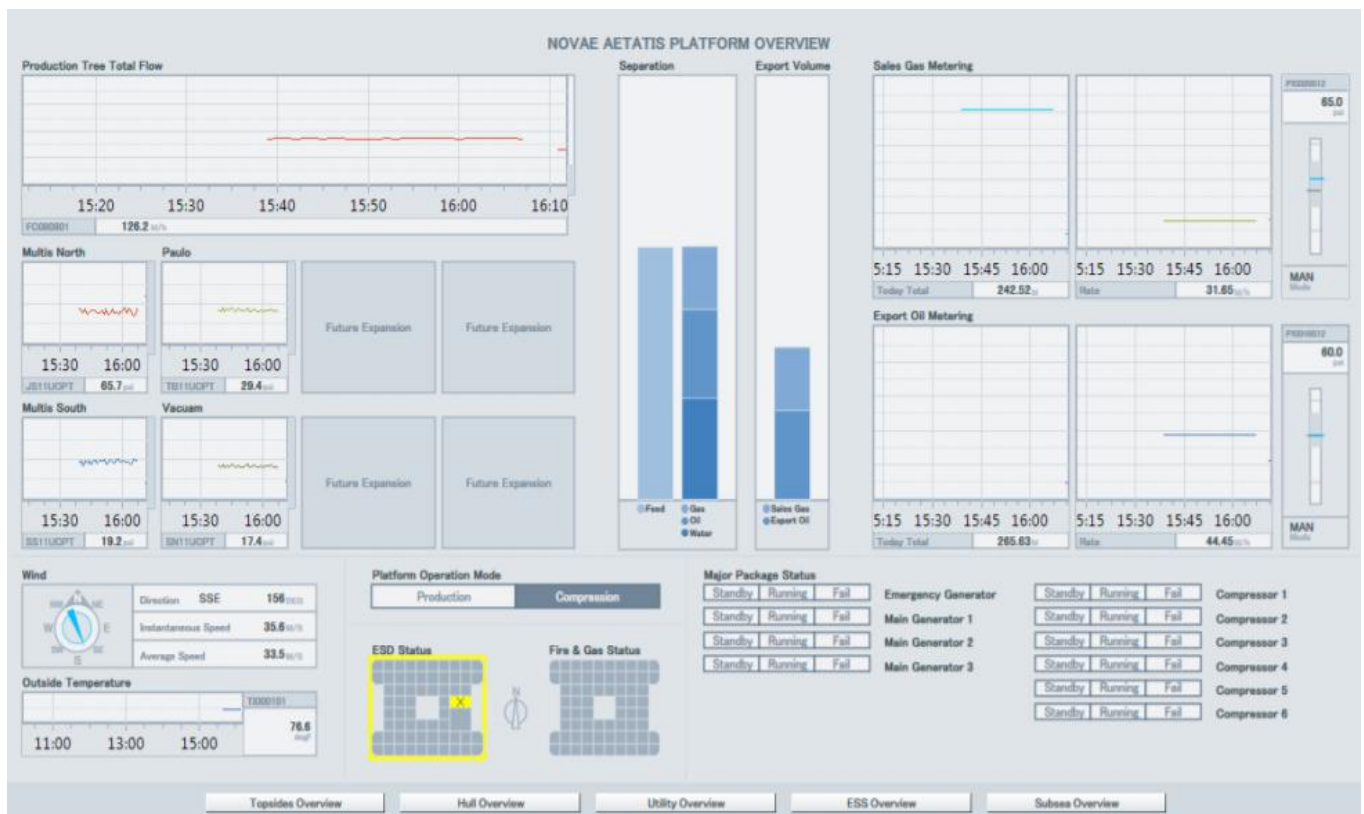


Рис. 8 – Структура вікна першого рівня [6]
Fig. 8 – First level window structure [6]

Дисплей 2-го рівня (вікно технологічного процесу) містити всю інформацію та засоби управління, які необхідні для виконання більшості завдань оператора, пов'язаних з конкретною виробничою одиницею, з одного вікна [6, с. 48]. Він повинен містити більше деталей ніж дисплей 1-го рівня та забезпечувати навігацію до більш детальних відомостей на дисплеях рівня 3 та 4 (див. рис. 9).

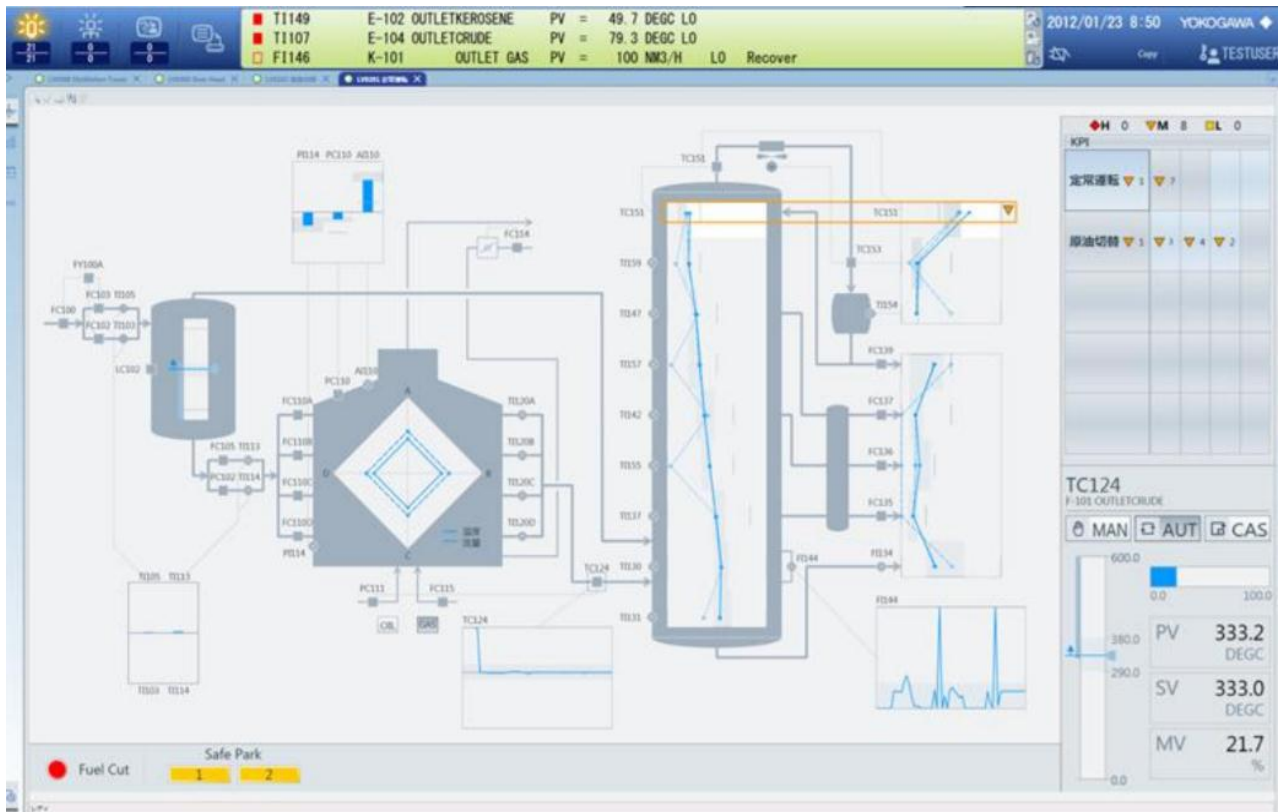


Рис. 9 – Структура вікна 2-го рівня [8]

Fig. 9 – Level 2 window structure [8]

Дисплей 3-го рівня (вікно контуру/підсистеми процесу) забезпечує всю детальну інформацію про конкретну систему або підсистему процесу та використовується для детальної діагностики проблем [7, с. 39]. На ньому зображуються всі параметри, статуси, сигнализації та блокування. Ієрархічно, ця інформація може бути об'єднана в рівень 2 або рівень 4. Приклад вікна для дисплеїв рівня 3 наведено на рисунку 10.

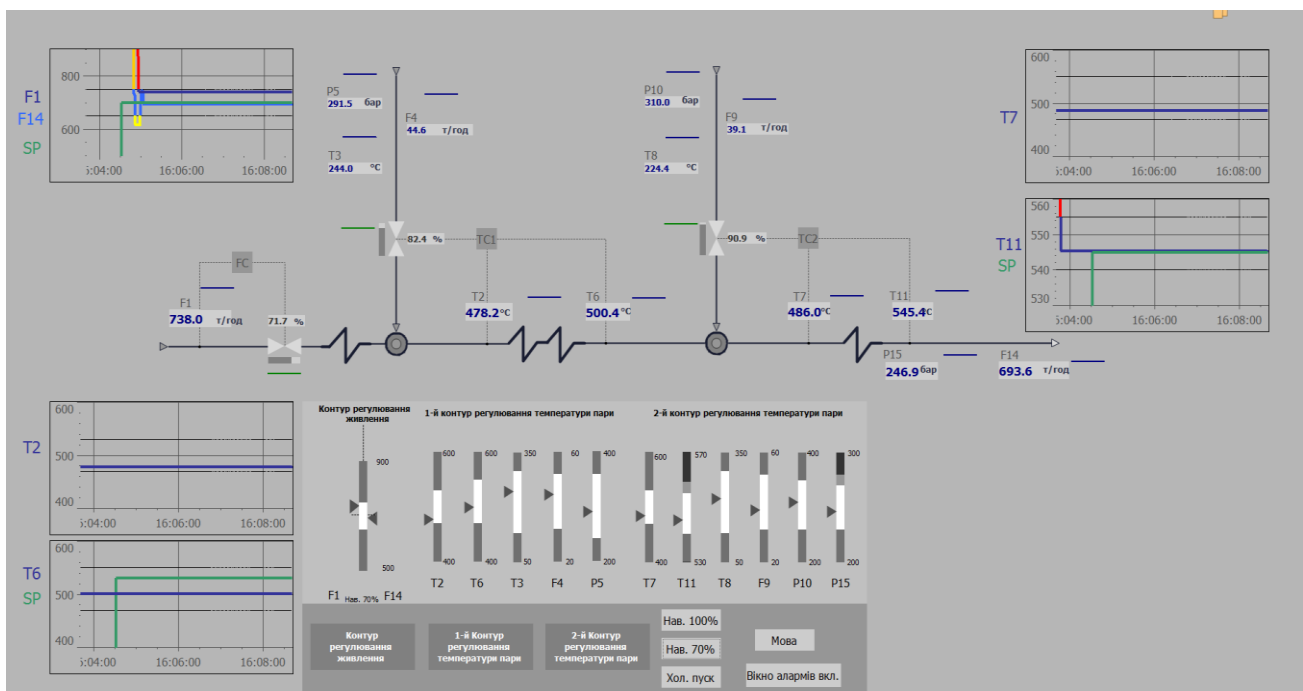


Рис. 10 – Структура вікна 3-го рівня

Fig. 10 – Structure window level 3



Дисплеї 4-го рівня (вікна діагностики) забезпечують найбільш деталізовану інформацію про підсистеми, окремі датчики або компоненти. Вони показують максимально детальну діагностичну або іншу інформацію, яка необхідна оператору в певний момент часу (див. рис. 11). Лінія розподілу між дисплеями рівня 3 та рівня 4 може бути дещо розмитою [8, с. 47].

RD	+0.0		X
SM	500.4	SP	530.0
ERR	29.6	Kp	1.0
OUT	0.0	Ti	100.0
ST	0	Kd	0.6
Er	0	Td	100.0
Eb	0	MV	0.0
RES	☐	MEn	☐

Рис. 11 – Структура вікна 4-го рівня
Fig. 11 – Structure window Level 4

Створення та підтримка високоефективного HMI - це багатоетапний процес, який може тривати десятки років від початкових концепцій до виведення системи з експлуатації. Стандарт ISA-101 розкриває підхід життєвого циклу для ефективного управління HMI та намагається виявити і вирішити різні проблеми протягом всього цього періоду часу, починаючи від розробки стандартів та дизайну HMI, закінчуючи налагодженням та обслуговуванням HMI [9]. На рисунку 12 показані етапи, які охоплює цей стандарт.

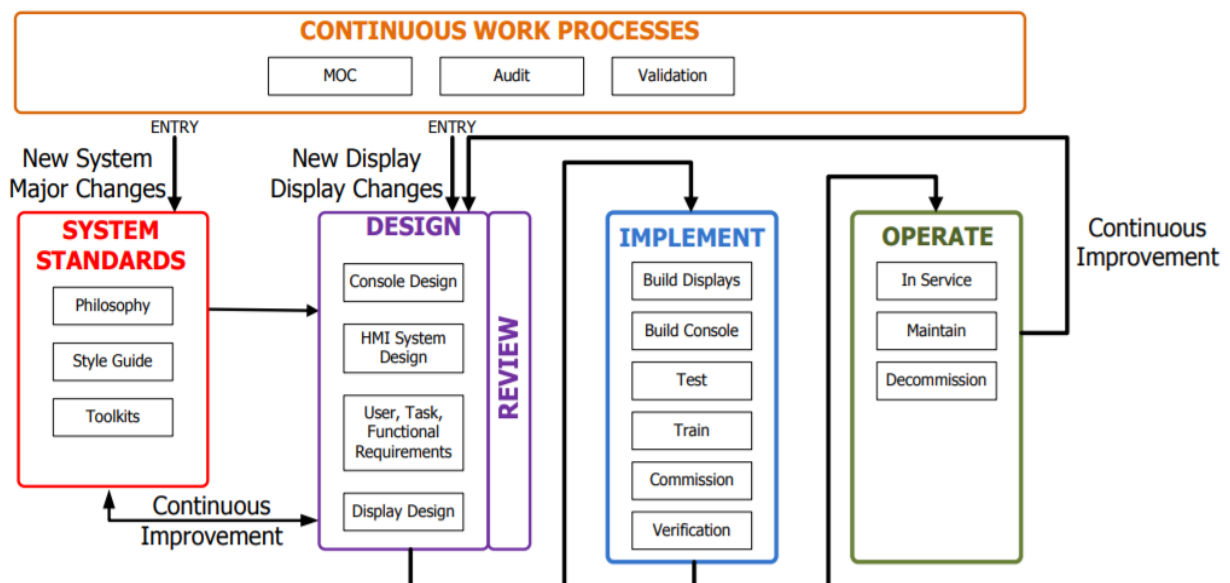


Рис. 12 – Етапи життєвого циклу HMI згідно ISA-101 [9]
Fig. 12 – HMI life cycle stages according to ISA-101 [9]

Стандарт надає вказівки щодо проектування, побудови та обслуговування HMI, що призводить до більш ефективного контролю процесу. Він також спрямований на поліпшення здатності оператора своєчасно та ефективно виявляти, діагностувати та реагувати на нештатні ситуації [10]. Практики, визначені в ISA-101, можуть бути застосованими до різних типів процесів: безперервних, періодичних та дискретних. Таким чином цей стандарт надає багато користі для кінцевих споживачів, включаючи можливості зменшити вартість HMI протягом усього життєвого циклу, підвищити ефективність роботи оператора та зменшити кількість аварій на підприємстві.

Висновки

В роботі наведено, що більшість систем автоматизації, що знаходяться на сьогоднішній день в експлуатації мають малоефективний та непрактичний HMI, що був розроблений без належних знань та призводить до зниження



швидкості сприйняття інформації оператором, підвищення зорового навантаження, і як результат збільшення часу реакції в екстрених ситуаціях та надмірна втома зорового апарату оператора. Протягом багатьох років експлуатуючий персонал працює з такими системами і навіть не підозрює, що правильно організований інтерфейс здатен впливати на ефективність роботи та самопочуття. Функціональність та спостережливість цих систем можна значно підвищити, якщо переробити їх відповідно до принципів високопродуктивної графіки, які базуються на практичності та досвіді.

Продемонстровано ефективність високопродуктивної графіки, яка є основою для подальших досліджень з метою вдосконалення взаємодії між людиною та машиною на підприємствах в кожному конкретному випадку. Це дасть змогу виключити суб'єктивність сприйняття інформаційної обстановки персоналом та дасть можливість значно поліпшити рівень спостережливості за технологічним процесом. Імплементація описаних методів проектування НМІ на діючих промислових виробництвах та експериментальна перевірка ефективності впровадження подібних технологій у повній мірі задовольнить усі необхідні інформаційні потреби операторів, та в той же час виключатиме можливість виникнення небезпечної аварійної ситуації на підприємстві внаслідок втоми або невірного трактування інформації обслуговуючим персоналом.

Показано, що одним з перспективних напрямків є аналіз глобального ефекту від впровадження високопродуктивної графіки, а саме оцінка ефективності виробничих процесів підприємства, аналіз ефективності інновацій, таких як впровадження подібного роду графіки на мобільних пристроях обслуговуючого персоналу, оцінка взаємозв'язку кількості аварійних ситуацій від якості графіки, комплексний аналіз повноти та ефективності від використання високопродуктивної графіки на підприємствах різних типів, переважно там де вагому роль грає людський фактор.

Розвиток стандарту ISA-101, пошук нових методів та дослідження в цій галузі є особливо перспективними в епоху Industry 4.0. Впровадження передових практик на діючих технологічних підприємствах є важливим завданням для молодих вчених та інженерів.

Подяка. Стаття підготовлена в рамках виконання проекту «Розроблення техніко-технологічних схем та систем керування теплозабезпечення населених пунктів на основі термодинамічних підходів» (номер державної реєстрації НДР 0120U102168).

Список використаних джерел

- [1] R. Lew, R. L. Boring and T. A. Ulrich, "A prototyping environment for research on human-machine interfaces in process control use of Microsoft WPF for microworld and distributed control system development," 2014 7th International Symposium on Resilient Control Systems (ISRCs), Denver, CO, 2014, pp. 1-6, doi: 10.1109/ISRCs.2014.6900111.
- [2] B. Hollifield, "A High Performance HMI. Better Graphics for Operations Effectiveness", 2012 ISA Water & Wastewater and Automatic Controls Symposium, Orlando, 2012, pp. 1-27.
- [3] Situational Awareness – The Next Leap in Industrial Human Machine Interface Design. Whitepaper. Schneider Electric Software, LLC., 2015. [Online]. Available: <https://www.controlglobal.com/assets/15WPpdf/150813-Schneider-SituationalAwareness.pdf>. Accessed on: Nov. 19, 2020.
- [4] Next-generation SCADA high performance human-machine interfaces. Presentation. Arcadis, 2018. [Online]. Available: https://onewaterohio.org/docs/NextGenSCADA_HPHMI_Final_020817_-_OneWater.pdf. Accessed on: Nov. 11, 2020.
- [5] D. Reising, and P. Bullemer, "Operator situation awareness & delivering ASM-compliant high performance HMIs: Failure modes and success factors", American Fuel & Petrochemical Manufacturers Annual Meeting, San Antonio, 2015, pp. 1-10.
- [6] ISA101: What It Is and Why It Is Needed – Overview and Current Activities. Presentation. The International Society of Automation, 2017. [Online]. Available: <https://isaneworleans.files.wordpress.com/2017/08/isa-new-orleans-sept-2017-meeting-isa101-presentation.pdf>. Accessed on: Nov. 14, 2020.
- [7] Using ISA-101 & High Performance HMIs for More Effective Operations. Presentation. WEAO Intelligent Wastewater Systems Seminar, 2017. [Online]. Available: http://www.grahamnasby.com/files_publications/NasbyG_2017_HighPerformanceHMIs_IntelligentWastewaterSeminar_WEAO_sept14-2017_slides-public.pdf. Accessed on: Nov. 11, 2020.
- [8] ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems. Presentation. The International Society of Automation, 2019. [Online]. Available: https://www.instmc.org/Portals/0/ISA101%202019%20Tech%20Lecture%20%281%29_1_1.pdf. Accessed on: Nov. 18, 2020.
- [9] ISA-101 Standard Addresses the HMI Lifecycle. Advisory group blog of industry trends. The International Society of Automation, 2018. [Online]. Available: <https://www.arcweb.com/blog/isa-101-standard-addresses-hmi-lifecycle>. Accessed on: Nov. 16, 2020.
- [10] ISA-101: Toward a More Effective HMI Strategy. A subsidiary of the International Society of Automation, 2015. [Online]. Available: <https://www.automation.com/en-us/articles/2015-1/isa-101-toward-a-more-effective-hmi-strategy>. Accessed on: Nov. 19, 2020.

References

- [1] R. Lew, R. L. Boring and T. A. Ulrich, "A prototyping environment for research on human-machine interfaces in process control use of Microsoft WPF for microworld and distributed control system development," 2014 7th International Symposium on Resilient Control Systems (ISRCs), Denver, CO, 2014, pp. 1-6, doi: 10.1109/ISRCs.2014.6900111.



- [2] B. Hollifield, "A High Performance HMI. Better Graphics for Operations Effectiveness", 2012 ISA Water & Wastewater and Automatic Controls Symposium, Orlando, 2012, pp. 1-27.
- [3] Situational Awareness – The Next Leap in Industrial Human Machine Interface Design. Whitepaper. Schneider Electric Software, LLC., 2015. [Online]. Available: <https://www.controlglobal.com/assets/15WPpdf/150813-Schneider-SituationalAwareness.pdf>. Accessed on: Nov. 19, 2020.
- [4] Next-generation SCADA high performance human-machine interfaces. Presentation. Arcadis, 2018. [Online]. Available: https://onewaterohio.org/docs/NextGenSCADA_HPHMI_Final_020817_-_OneWater.pdf. Accessed on: Nov. 11, 2020.
- [5] D. Reising, and P. Bullemer, "Operator situation awareness & delivering ASM-compliant high performance HMIs: Failure modes and success factors", American Fuel & Petrochemical Manufacturers Annual Meeting, San Antonio, 2015, pp. 1-10.
- [6] ISA101: What It Is and Why It Is Needed – Overview and Current Activities. Presentation. The International Society of Automation, 2017. [Online]. Available: <https://isaneworleans.files.wordpress.com/2017/08/isa-new-orleans-sept-2017-meeting-isa101-presentation.pdf>. Accessed on: Nov. 14, 2020.
- [7] Using ISA-101 & High Performance HMIs for More Effective Operations. Presentation. WEAO Intelligent Wastewater Systems Seminar, 2017. [Online]. Available: http://www.grahamnasby.com/files_publications/NasbyG_2017_High_PerformanceHMIs_IntelligentWastewaterSeminar_WEAO_sept14-2017_slides-public.pdf. Accessed on: Nov. 11, 2020.
- [8] ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems. Presentation. The International Society of Automation, 2019. [Online]. Available: https://www.instmc.org/Portals/0/ISA101%202019%20Tech%20Lecture%20%281%29_1_1.pdf. Accessed on: Nov. 18, 2020.
- [9] ISA-101 Standard Addresses the HMI Lifecycle. Advisory group blog of industry trends. The International Society of Automation, 2018. [Online]. Available: <https://www.arcweb.com/blog/isa-101-standard-addresses-hmi-lifecycle>. Accessed on: Nov. 16, 2020.
- [10] ISA-101: Toward a More Effective HMI Strategy. A subsidiary of the International Society of Automation, 2015. [Online]. Available: <https://www.automation.com/en-us/articles/2015-1/isa-101-toward-a-more-effective-hmi-strategy>. Accessed on: Nov. 19, 2020.

УДК 681.51

РОЗРОБКА АВТОНОМНИХ WEB-СЕРВІСІВ ДЛЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ДИНАМІЧНИХ СИСТЕМ В РЕАЛЬНОМУ ЧАСІ НА БАЗІ МОДЕЛЕЙ MATLAB SIMULINK

Стопакевич А. О.¹, Стопакевич О. А.², Тігарєв А. М.³, Тігарєва Т. Г.⁴

^{1,3} Одеська національна академія зв'язку імені О.С. Попова, м. Одеса, Україна

² Одеський національний політехнічний університет, м. Одеса, Україна

⁴ Одеська державна академія будівництва та архітектури, м. Одеса, Україна

ORCID: ¹ <https://orcid.org/0000-0003-1719-9071>, ² <https://orcid.org/0000-0002-8318-6853>,

³ <https://orcid.org/0000-0002-3394-2526>, ⁴ <https://orcid.org/0000-0003-4453-279X>

E-mail: ¹ stopakevich@gmail.com

Copyright © 2020 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i4.1931>

Анотація. У статті розглянута задача розробки автономного web-сервісу на базі Simulink моделей. Запропоновано технологію, яка полягає в розробці комплексу з двох пов'язаних окремих додатків: консольного застосування на мові Сі на базі коду, який генерує Simulink Coder і додатки на мові Python, яке реалізує функціонал web-сервісу. Принциповим недоліком вже запропонованих рішень по розробці web-сервісів і web-додатків на базі Simulink моделей є те, що Simulink модель виконується за допомогою Matlab. Через це виникають нерозв'язні проблеми, а саме: надмірність і ресурсовитратність, ненадійність виконання моделі Simulink в реальному часі, складність взаємодії з моделлю під час виконання, фактична однопоточність Matlab. Останнє означає, що в один момент часу в пакеті може виконуватися тільки одна модель Simulink. Головною відмінністю запропонованої



технології є те, що розроблений web-сервіс, на відміну від раніше запропонованих рішень, є автономним, тобто не вимагає для своєї роботи Matlab. Це дозволяє досягти суттєвої економії ресурсів і підвищує надійність роботи в цілому. Переваги запропонованої технології: швидка розробка; перенесення, тобто можливість працювати як в операційних системах загального призначення, реальному часу і в операційних системах для одноплатних комп'ютерів; низькі вимоги до ресурсів; стандартний програмний інтерфейс віддаленого доступу. Сфера застосування технології: розробка «цифрових двійників» для налагодження компонентів систем автоматизації, швидка розробка регуляторів складної структури (за умови, що вона ефективно може бути представлена у вигляді окремих блоків), дистанційне навчання. Подальший напрямок досліджень планується в області удосконалення запропонованого підходу в області створення цифрових двійників для складних технологічних об'єктів, розробки регуляторів з неklasичними підходами в області відновлення змінних стану для об'єктів з істотними перехресними зв'язками, а також в галузі дистанційного навчання в рамках LMS платформи.

Abstract. The article is devoted to the task of developing an autonomous web-service based on Simulink models. The technology, which consists in the development of a complex of two linked separate applications: console application in C language on the basis of code, which generates Simulink Coder and applications in Python language, which implements the functionality of web-service, is proposed. The principal disadvantage of the already proposed solutions for the development of web services and web applications based on Simulink models is that the Simulink model is performed using Matlab. Because of this there are unsolvable problems, namely: redundancy and resource consumption, unreliability of Simulink model execution in real time, complexity of interaction with the model at runtime, the actual single threading Matlab. The latter means that only one Simulink model can run in a package at a time. The main difference of the proposed technology is that the developed web-service, in contrast to previously proposed solutions, is standalone, which means it does not require Matlab for its work. This allows the achievement of significant savings in resources and increases the reliability of the system as a whole. Advantages of the proposed technology: 1) fast development; 2) portability, i.e. possibility to work both in general-purpose OS, and in real time OS and in OS for single board computers; 3) low requirements to resources; 4) standard software interface for remote access. Scope of application of the technology: development of "digital twins" for debugging of automation system components, fast development of complex structure regulators (assuming that it can be effectively represented as separate blocks), distance learning.

Ключові слова: Matlab, Simulink, web-сервіс, автономність, цифровий двійник, відладка, регулятор, блок, мова C, Python

Keywords: Matlab, Simulink, web-service, autonomous, digital twin, debugging, controller, block, C language, Python

Введение

Simulink – один из самых известных пакетов Matlab, который позволяет представлять динамические системы в удобной для восприятия человека форме и проводить их имитационное моделирование. Кроме того, пакет имеет и дополнительные функции, среди которых обратим внимание на функцию генерации кода для выполнения моделируемой системы в среде реального времени. Проблемой программ, которые генерируются Matlab Simulink Coder из моделей Simulink является то, что они требуют значительной доработки для обеспечения возможности взаимодействия с ними в реальном времени с помощью стандартных программных интерфейсов. Программы, которые генерируются Matlab Compiler для использования на Desktop или с пакетом Web App для реализации web-приложений частично лишены указанных недостатков, но, по сути, требуют для своего выполнения Matlab (или его ограниченную версию, поставляемую в бесплатном Runtime). Такие программы требуют избыточное количество ресурсов, долго загружаются и могут быть запущены в ограниченном количестве программных окружений.

Указанные обстоятельства вынуждают разработчиков искать пути выхода из ситуации. В работе предлагается технология, которая позволяет быстро преобразовать модель Simulink в автономную динамическую систему, работающую в реальном времени в качестве web-сервиса в большом количестве программных окружений, в т.ч. в окружениях с ограниченными ресурсами. Такие системы полезны для отладки систем автоматизации в реальном времени как аналоги реальных объектов управления (т.н. цифровые двойники), для реализации управляющих алгоритмов, а также в образовательных целях – для реализации лабораторных работ с возможностью параллельного выполнения их студентами удаленно.

Анализ литературных источников

Задача разработки удаленного web-интерфейса для моделирования динамических систем при проведении лабораторных работ рассмотрена в работе [1]. Студент выбирает среди списка модель, которая реализует управляющее устройство, вводит его параметры и запускает на управление реальным двигателем в реальном времени. С помощью камеры студент может наблюдать за поведением объекта управления. После выполнения работы строится график. Технически это реализуется так: на сервере с платой АЦП устанавливается Matlab и web-сервер Apache. Для каждой модели из списка разрабатывается отдельный скрипт для ввода данных. Задача скрипта – сформировать mat файл и передать его по стандартным интерфейсам межпрограммного взаимодействия в Matlab, который вызовет функцию sim для одной из известных моделей с данными из сформированного файла. После выполнения модели в течении заданного времени моделирования результаты возвращаются PHP программе и отображаются пользователю.



В каждый момент времени с системой может работать только один студент. Задача составления схемы модели для удаленного моделирования рассматривается в работе [2]. Редактор с web-интерфейсом позволяет составить схему моделирования, которую можно затем экспортировать в форматы схем Simulink и Xcos. Указанные две идеи и разработка редактора схем с web-интерфейсом, который генерирует Simulink модели и выполняет их в Matlab, объединены в работе [3]. В качестве серверного языка используется PHP. Задача взаимодействия с выполняемой Simulink-моделью в реальном времени решается в работе [4]. Поскольку получить состояние модели при её выполнении в Matlab без остановки моделирования невозможно, то разработан специальный MEX блок для Simulink, который реализует информационный обмен в реальном времени. Для информационного обмена используются механизмы сокетов и технология построения XML web-сервисов ASP.NET на базе Microsoft WCF. Такой же подход предложен в более современной работе [5], однако используется язык Python и web-сервисы JSON. Web-сервер реализуется прямо в Matlab в работе [6]. Web-сервер приостанавливает моделирование, обновляет параметры модели при получении данных от удаленного пользователя и запускает моделирование снова, что позволяет избежать необходимости разрабатывать отдельный блок для коммуникации модели, однако является более ресурсозатратным.

Все указанные работы ориентируются на то, что Simulink модель выполняется с помощью Matlab. Из-за этого возникают неразрешимые проблемы, а именно: избыточность и ресурсозатратность, ненадежность выполнения модели Simulink в реальном времени, сложность взаимодействия с моделью во время выполнения, принципиальная однопоточность Matlab. Последнее означает, что в один момент времени в пакете может выполняться только одна модель Simulink.

Таким образом, реализовать полноценную многопользовательскую клиент-серверную систему имитационного моделирования динамических систем в реальном времени с таким подходом невозможно. Кроме того, идея разработки альтернативного редактора для моделей Simulink в предложенных решениях сводится к тому, что генерируется файл Simulink-модели программой, а не средствами Matlab. Практически, это может приводить к проблемам совместимости и ошибкам, т.к. форматы Simulink файлов меняются каждый выпуск Matlab. Также не рассматривается задача отображения в web-интерфейсе состояния блоков в реальном времени, что во многом нивелирует преимущество наглядности схематического моделирования перед моделированием программным.

Целью статьи является разработка технологии проектирования автономного web-сервиса, который не требует для своей работы Matlab. Кроме того, программное решение должно соответствовать следующим требованиям: 1) быстрая разработка; 2) переносимость, т.е. возможность работать как в ОС общего назначения, в ОС реального времени и в ОС для одноплатных компьютеров; 3) низкие требования к ресурсам; 4) стандартный программный интерфейс удаленного доступа.

Выбор средств и стратегии для реализации требований

С помощью инструментов Matlab возможно преобразовать большинство Simulink-моделей в код на языках Си и Си++. Собственно, сам Simulink этим активно пользуется, применяя генерацию и компиляцию кода для отдельных блоков при симуляции. Более того, режим ускорения является, по сути, выполнением скомпилированного кода программы на языке Си, сгенерированного по модели. Поэтому первый этап состоит в генерации кода на одном из этих языков. Хотя языки Си с Си++ формально являются кроссплатформенными, на практике программы, которые реализуют работу с потоками, сетью, драйверами будут сильно отличаться по причине несовместимых программных интерфейсов различных ОС. Поэтому, доработка сгенерированного кода программы до полноценного web-сервиса на языках Си/Си++, является задачей достаточно затратной, следовательно, не соответствует требованию 1. Возможно также решение в виде реализации дополнительной библиотеки или модуля к web-серверам Apache или nginx, что является более простым, но не соответствует требованию 2. Кроме того, программные интерфейсы web-серверов постоянно меняются и в них выявляются ошибки уязвимости, что вынудит постоянно обновлять код программы. Поэтому предлагается для реализации функционала web-сервиса написать дополнительную программу на языке программирования, который оптимизирован для этой цели

Среди языков, оптимизированных для разработки web-сервисов, следует выделить C#, Java, Go и Python. Первые два языка требуют наличия виртуальной машины, которая не доступна для всех платформ, поэтому их использование не соответствует требованию 2. По большому счету соответствию требованию 1 язык Python выигрывает, т.к. сам по себе язык Python более простой чем Go, функциональный и расширяемый за счет большого количества готовых библиотек. Кроме того, программы на Python не требуют компиляции для выполнения. С другой стороны язык Go, который по сути является более развитым языком Си, оптимизированным не только для системного программирования, а и для web-приложений выигрывает по производительности полученных скомпилированных программ. Хотя разница по скорости между программами, написанными на Python и Go, может отличаться в десятки раз, в рассмотренном случае «узким местом» является программа, которая выполняет модель в реальном времени. Поэтому при больших нагрузках система будет отказывать отнюдь не из-за медленного web-сервиса, а из-за невозможности параллельного выполнения множества динамических моделей в реальном времени. Следовательно, остановим свой выбор на Python.

В качестве механизма взаимодействия между программой на Python и программой на Си/Си++ можно рассмотреть следующие механизмы: 1 – реализации библиотеки на Си/Си++ с последующим вызовом её из программы на Python; 2 – обмена с помощью сокетов между приложениями; 3 – однонаправленного информационного обмена с помощью интерфейса командной строки. С точки зрения требования 2 стоит выбрать интерфейс командной строки. Хотя



первый механизм в общем случае является наиболее корректным, однако для Python он нецелесообразен по причине использования языком Python блокировок GIL в многопоточных программах, которая вызывает проблемы со скоростью и надежностью работы программы. Второй механизм делает программу на Си сильно зависимой от платформы. Поэтому, следует выбрать третий механизм, т.к. Python имеет библиотеки для exрест-подобного управления интерактивными приложениями, работающими в режиме командной строки, как для POSIX-совместимых систем, так и для Windows (для Go, например, на данный момент такой библиотеки нет).

Поскольку объектно-ориентированная функциональность в данном случае не требуется и снижает скорость работы, то для генерации кода выберем язык Си. Последующая доработка кода будет вестись в соответствии со спецификацией языка C99.

Существуют две основные спецификации web-сервисов: REST и SOAP. Первая более популярна в последнее время и больше соответствует требованиям 3 и 4. В качестве языка сообщений в RESTful сервисах используются JSON-схемы данных, передающиеся по протоколу HTTP.

Структурная схема предлагаемого программного решения для реализации автономного web-сервиса показана на рисунке 1.



Рис. 1 – Структурная схема предлагаемого программного решения
Fig. 1 – Schematic diagram of the proposed program solution

Этапы разработки автономного web-сервиса

Предлагаемый нами алгоритм выполняется в 6 этапов:

- 1) Преобразование имеющейся математической модели в соответствии с требованиями Simulink Coder;
- 2) Генерация или ручная сборка Simulink модели;
- 3) Реализация решателя в реальном времени в виде консольного приложения на базе программного кода, сгенерированного по Simulink-модели;
- 4) Реализация RESTful web-сервиса на языке Python;
- 5) Реализация интерфейса для удаленной работы с web-сервисом;
- 6) Интеграция RESTful web-сервиса в имеющееся программно-техническое окружение.

Пример для иллюстрации

В качестве иллюстративного примера рассмотрим задачу моделирования объекта управления в программном обеспечении системы автоматизации по имеющейся линейной математической модели динамики.

Модель динамики объекта управления зададим в виде матрицы передаточных функций:

$$W(s) = \begin{bmatrix} W_{11}(s) & W_{12}(s) \\ W_{21}(s) & W_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{11}{11s+1}e^{-100s} & \frac{12}{33s+1}e^{-200s} \\ \frac{21}{22s+1}e^{-300s} & \frac{22}{44s+1}e^{-400s} \end{bmatrix},$$

где s – оператор Лапласа.

Этап 1 – Адаптация к требованиям Simulink Coder. Требования к моделям в Simulink Coder следующие:

1. Модель должна быть реализована с одним из решателей в дискретном времени. Это означает что ряд блоков не применим, в частности все блоки пакета Simscape. В документации по каждому пакету Matlab есть сводная таблица совместимости.

2. Модель может содержать Matlab код только из подмножества Embedded Matlab.



3. Модель не должна содержать МЕХ и прочие платформенно зависимые блоки, в том числе, отвечающие за доступ к основным программным интерфейсам.

Все параметры модели, которые пользователь должен менять или за которыми следует наблюдать надо четко выделить, поскольку при генерации кода Matlab склонен сокращать код, отвечающий за расчеты.

Также, по возможности, с целью увеличения наглядности модели желательно заменять блоки с комплексом параметров на элементарные блоки Simulink.

С целью упрощения представим передаточные функции (ПФ) в виде дифференциальных уравнений вида:

$$\dot{x}(t) = \frac{-1}{T} \cdot x(t) + \frac{k}{T} \cdot u(t - \tau),$$

где x – переменная состояния, T – постоянная времени звена ПФ, k – коэффициент передачи звена ПФ, τ – время запаздывания.

Кроме того, пользователь должен видеть значения выходов системы управления (y_1, y_2), а также выходы основных блоков модели (интеграторов, запаздываний и блоков вычисления отношения k/T)

В качестве решателя выберем ode5 – дискретный аналог используемого Matlab по умолчанию решателя ode45. Шаг дискретности выберем равным – 0,2.

Этап 2 – Генерация или ручная сборка Simulink модели. Стандартным способом разработки схемы математической модели в Simulink является её ручная сборка. Для последующей генерации каждому блоку необходимо дать понятный идентификатор латинскими буквами с целью последующей локализации отвечающего за него кода.

Задача генерации математической модели стандартными средствами Matlab решена на данный момент неудовлетворительно. Для добавления блоков на схему с заданными параметрами используется функция `add_block`, а для связывания входов между блоками – `add_line`. Функционал связанный с автоматическим размещением блоков фактически отсутствует, что требует при генерации вручную вводить координаты блоков. Это сильно усложняет проблему.

Интересное решение указанной проблемы найдено в библиотеке Matsim [7]. Эта библиотека имеет две уникальные возможности: построение системы Simulink из уравнений и автоматическое размещение элементов схемы. Последнее реализуется следующим образом: измеряются размеры блоков Simulink схемы, формируется задача на языке DOT, которая решается путем поиска оптимального размещения блоков с минимумом пересечений с учетом их порядка и приоритета, решение этой задачи переводится в координаты блоков Simulink. Схематическое представление решения задачи для элемента (1,1) матрицы передаточной функции проведенное с помощью интерпретатора Graphviz языка DOT, показано на рисунке 2.

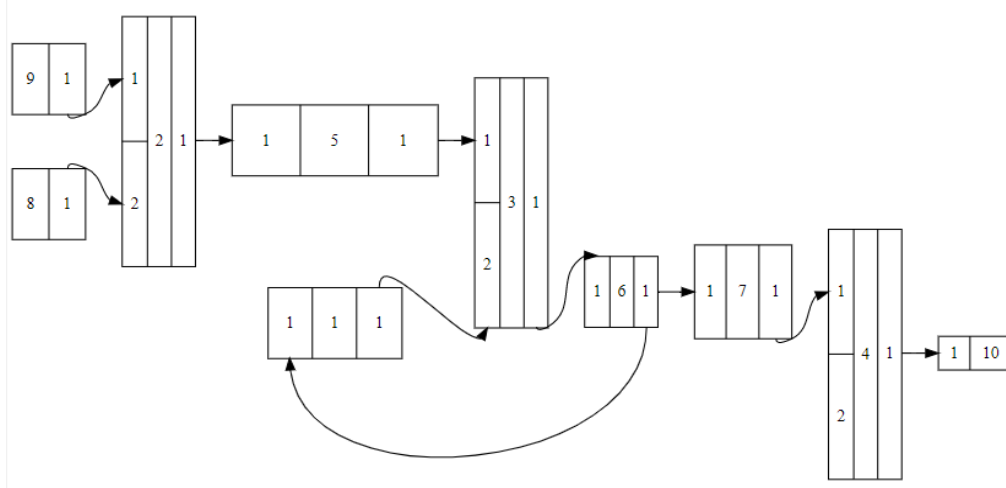


Рис. 2 – Оптимальное расположение блоков, рассчитанное пакетом Graphviz

Fig. 2 – Optimal location of blocks, calculated by Graphviz package

При представлении результата схематически каждый блок содержит три столбца. Первый содержит номер(а) входов, второй – уникальный числовой идентификатор блока, третий – номер(а) выходов. Для входных блоков первый столбец опускается, для выходных опускается последний столбец.

Сигнал управления (блок 8) суммируется (блок 2) с сигналом возмущения ко входу (блок 9). Далее величина умножается на результат вычисления $\frac{k}{T}$ (блок 5). Результат вычисления блока 5 суммируется в блоке 1 с результатом

вычисления блока 1 ($-\frac{1}{T}$). Далее сигнал проходит через интегратор (блок 6), звено запаздывания (блок 7), суммируется в блоке (4) и записывается в выходную переменную y_1 (блок 10).



Этап 3 – Реализация решателя в реальном времени в виде консольного приложения на базе программного кода, сгенерированного по Simulink-модели. Для генерации кода в Matlab необходимо выбрать цель, язык программирования и компилятор. В предлагаемом алгоритме следует выбрать цель ert (встраиваемые системы) и язык Си.

Необходимые настройки: генерировать только код, динамическое размещение памяти для инициализации модели, представлять входы и выходы корневого уровня в качестве структуры данных модели, поведение параметров – настраиваемые, оставлять порядок операндов в выражении неизменным. Рекомендуется также задать цели: эффективность и возможность трассировки.

После генерации Simulink Coder формирует папку с программой, которая проводит бесконечный вычислительный цикл без таймера. В отличие от программ, сгенерированных обычным Matlab Coder, программа содержит обработчики цикла, что позволяет модифицировать код для симуляции в реальном времени.

Ключевой файл сгенерированного кода – это файл с именем модели Simulink и с расширением .c. В этом файле (для примера он составляет порядка 30 кб кода) реализуется численное дифференцирование с помощью алгоритма ode5 и он не нуждается в модификации. Файл с окончанием _data.c содержит в себе структуру данных, которая содержит параметры модели, которые возможно модифицировать во время выполнения программы.

Файл, который необходимо модифицировать называется ert_main.c. В этом файле необходимо реализовать две задачи: 1) возможность модификации параметров модели 2) выполнение в реальном времени или с заданным ускорением.

Для решения первой задачи наиболее простым и переносимым является реализация интерфейса командной строки, рассмотренная, например в [8]. Минимальный интерфейс командной строки должен иметь следующие команды: 1) start – запуск симуляции 2) pause – пауза 3) restart – сброс настроек и запуск симуляции 4) time – вывод текущего времени симуляции / переход к заданному времени 5) acc – коэффициент ускорения симуляции 6) set – установка параметра 7) get – получение значения параметра, группы параметров или всех блоков 8) status – текущее состояние (пауза, выполнение, ошибка) 9) stepsize – получение шага моделирования 10) quit – прекращение работы программы. При запуске программы возникает приглашение командной строки, после ввода команды надо нажать ENTER и результат выведется. Для повышения надежности работы с экранным началом и концом результата выполнения команды рекомендуется выделять специальными ASCII символами (№2 и 3 в десятичной системе). Кроме того, в результате должен быть символ, который указывает на ошибку (21) и успешный результат выполнения (6).

Решение второй задачи во многом зависит от возможностей платформы. Для кроссплатформенного решения нужно реализовать как-минимум работу с помощью потоков POSIX, так и работу с помощью потоков Windows. Поскольку задача жесткого реального времени не стоит, а в Python программе и не может стоять (в т.ч. из-за наличия сборщика мусора), то это не является сложной задачей.

Этап 4 – Реализация RESTful web-сервиса на языке Python

4.1 Определение программного интерфейса. В первую очередь для реализации любого web-сервиса необходимо определить программный интерфейс. В нашем случае имеет смысл использовать только 2 класса команд: GET (считывание параметров) и PUT (изменение параметров). Сериализация данных проходит по правилам JSON.

Команды GET – команды для считывания данных:

1. /get/acc – текущий коэффициент ускорения симуляции. Пример вывода: 1.000000.
2. /get/blocks – текущие состояния блоков. Пример вывода: {"A11": -0.045103, "A12": -0.0, "A21": -0.024769, "A22": -0.0, "Add": 1.0, "Add1": 0.0, "Add4": 0.053996, "Add5": 0.069825, "Add6": 0.0, "Add7": 0.0, "G11": 0.099099, "G12": 0.0, "G21": 0.094595, "G22": 0.0, "Time": "67.800000", "W11": 5.006425, "W11delay": 0.0, "W12": 0.0, "W12delay": 0.0, "W21": 5.498778, "W21delay": 0.0, "W22": 0.0, "W22delay": 0.0}.
3. /get/model/[k, T или tau] – текущие коэффициенты блоков. Пример вывода для k: {"k11": 11.0, "k12": 12.0, "k21": 21.0, "k22": 22.0}.
4. /get/status – состояние симуляции. Примеры вывода: RUNNING, PAUSED, ERROR.
5. /get/stepsize – шаг моделирования. Пример вывода: 0.2.
6. /get/time – время моделирования. Пример вывода: 789.200000.
7. /get/variable/[u, f или y] – текущие значения основных переменных с указанием времени моделирования. Пример вывода для y: {"y1": 10.990272, "y2": 19.569536, "Time": "880.400000"}.

Команды PUT – команды для считывания данных:

1. /set/acc – новые коэффициент ускорения симуляции. Пример запроса: {"new_acc": 10}
2. /set/model – новые параметры модели. Пример запроса: {"k11": 2, "T12": 4}
3. /set/status – изменение состояния. Пример запроса: {"new_status": "pause"}
4. /set/time – ускоренно промоделировать до нового времени. Пример запроса: {"new_time": "500"}
5. /set/time – новые переменные. Пример запроса {"u1": 1, "f2": 5}

4.2 Определение формата файла конфигурации. Конфигурацию предлагается хранить в виде текстового файла формате INI. Этот формат оптимален для хранения небольшого количества настроек и удобен для ручного редактирования пользователем. Конфигурация должна содержать следующие параметры: 1) сетевые – сетевой адрес и порт, 2) имя и кодировка исполняемого файла программы-решателя 3) режим запуска по умолчанию – фоновая программа или программа с интерфейсом командной строки, требование автоматического запуска моделирования 4)



параметры безопасности (наличие гостевого доступа, необходимость авторизации для записи данных, разрешение на удаленную перезагрузку и приостановку моделирования) 5) параметры отчетности – имя файла для отчета, степень детальности отчета (отладка, нормальный, только ошибки) 6) параметры модели, вводимые перед началом моделирования (для того, чтобы не менять исполняемый файл решателя).

4.3 Выбор программных библиотек. Язык Python – лидер по количеству качественных и бесплатных библиотек для многих типовых задач программирования.

Наиболее популярными библиотеками для реализации web-сервисов и приложений на Python являются Django и Flask. Последняя библиотека минималистична при достаточном для поставленной задачи функционале, поэтому следует использовать именно Flask. Для реализации публичного программного интерфейса необходима библиотека для управления доступом с других ресурсов – Swagger и flask-cors. Также рекомендуется использовать для реализации функционала авторизации – flask-login и для кеширования – lru-cache. Поскольку информация имеет дискретный характер, то кеширование GET запросов может значительно снизить нагрузку.

Для взаимодействия с исполняемой программой решателя необходимо использовать библиотеку wexrest для Windows и rexrest для POSIX-совместимых платформ. Программный интерфейс у этих библиотек практически одинаков, однако внутренняя реализация полностью отлична.

Для реализации многопоточности необходимо использовать функционал библиотек os, threading, signal.

Для формирования приоритетной очереди рекомендуется использовать функционал, реализуемый с помощью библиотек queue, itertools и TypeVar.

Для обработки конфигурационных файлов рекомендуется использовать библиотеку configparser (преимущество этой библиотеки также в обеспечении возможности менять параметры конфигурации с помощью передачи аргументов командной строки программе), для ведения отчетов – logger, для записи текущей даты и времени в отчет по формату – datetime.

4.4 Программная реализация. Предлагаемая структура программы на Python запускает четыре процесса: процесс основной программы, процесс сервера Flask, процессы AppThread и TimerThread.

Процесс основной программы проводит запуск остальных процессов, реализует работу с системой ввода-вывода I/O и обработку стандартных сигналов ОС (SIGTERM, SIGINT и т.п.).

Процесс сервера Flask запускается в многопоточном режиме и обрабатывает адреса (т. н. маршруты, routes) API, описанные в разделе 4.1. С помощью декоратора, который содержит шаблон адреса, к адресу привязывается функция-обработчик. В случае GET запроса в функциях-обработчиках должны возвращаться данные из кэша, реализуемого с помощью функций с декоратором lru_cache с указанием периода обновления кэша. В случае PUT запроса входные данные проверяются и при их корректности добавляются в очередь с приоритетом 1.

Процесс, реализуемый с помощью класса TimerThread периодически добавляет в очередь команды обновления переменных (приоритет 2) и обновления параметров блоков (приоритет 3). Кроме того, в процессе адаптируется время цикла добавления в очередь по результатам отслеживания времени возвращения результата выполнения команд от решателя с целью недопущения переполнения очереди при высокой загрузке компьютера.

Процесс, реализуемый с помощью класса AppThread обеспечивает постоянную работу с программой-решателем с помощью expect-подобного управления: выбирает команду из очереди, посылает команды программе-решателю (sendline), дожидается возврата результата (expect) и считывает результаты (readline). Структура и основные алгоритмы класса представлены на рисунке 3.

4.5 Распространение. Исполняемый файл программы-решателя при сборке с помощью компилятора MINGW64 занимает ~1.4 МБ места на диске, в рабочем режиме процесс занимает ~14 МБ + память библиотек. Переносимая сборка программы на Python под разные платформы может быть реализована с помощью утилиты cx_Freeze. Под Windows путем тонкой настройки удалось добиться размера исполняемого файла ~13 МБ + ~24 МБ библиотек. Альтернативная возможность – использование такого инструмента как Docker. Кроме того, можно удовлетвориться стандартной установкой Python с установкой при необходимости требуемых библиотек в виртуальном окружении.

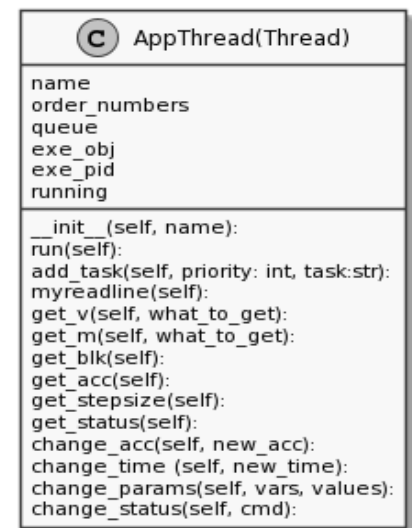
Этап 5 – Реализация интерфейса для удаленной работы с web-сервисом. Наиболее простым средством общения с RESTful web-сервисом является консольный клиент curl. Для ряда задач такой интерфейс может быть достаточен.

Типовым пользовательским интерфейсом для web-сервисов является web-интерфейс. Технология разработки такого интерфейса сводится к следующему 1) С помощью команды Matlab print модель Simulink экспортируется в формат SVG. 2) По рассчитанным с помощью языка DOT координатам на схему накладываются необходимые текстовые элементы. 3) Надписи связываются по таймеру с функцией периодического обмена с web-сервисом (при использовании библиотеки JQuery, например, обмен сообщениями реализуется с помощью функции \$.getJSON) 4) создаются функциональные элементы для изменения параметров с помощью вызовов put (в JQuery функция \$.ajax). 5) При необходимости строятся графики.

Пример работающего интерфейса показан на рисунке 4. На рис. показан момент после изменения значения с u1 с 3 на 2. При нажатии на параметр, который можно менять, в интерфейсе возникает диалог с текстовым полем. При подтверждении данные передаются web-сервису. Построение графика в реальном времени производится с помощью библиотеки CanvasJs. В желтых прямоугольных рамках отображаются выходные сигналы блоков.



Структура класса



Алгоритм метода run()

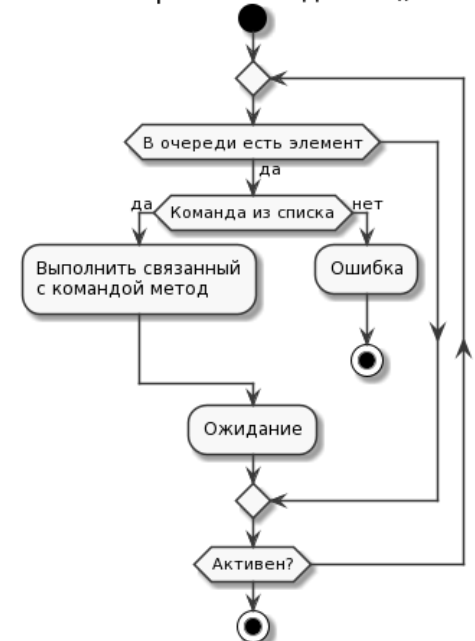


Рис. 3 – Структура и основные алгоритмы класса AppThread

Fig. 3 – Structure and main algorithms of AppThread class

Этап 6 – Интеграция RESTful web-сервиса в имеющееся программно-техническое окружение. Этот этап сложно формализовать, поскольку многое зависит от задачи. Для интеграции в типовую современную промышленную систему автоматизации необходимо использовать шлюз JSON-OPC UA. Такой шлюз возможно как разработать отдельно, так и использовать стандартный, например, реализованный в пакете OPC Router. Для интеграции с MQTT возможно использовать функционал IoT платформ.

Для образовательных целей целесообразна разработка дополнительной программы, которая будет авторизовать пользователей на выполнение заданной работы. После авторизации на сервере должен запускаться для конкретного пользователя отдельный web-сервис с выделенным для него сетевым портом. После выполнения работы сервис должен прекращать работу и освобождать порт при сохранении отчета о действиях пользователя.

Выводы

Разработка web-сервисов на базе Simulink моделей является удобным инструментом для реализации т.н. «цифровых двойников» с целью отладки работы систем автоматизации, для реализации систем регулирования и для дистанционного обучения. Анализ литературных источников показал, что предлагаемые решения ориентируются на выполнение Simulink моделей с помощью Matlab, создавая удаленный интерфейс к Matlab, что ресурсозатратно и в отдельных случаях небезопасно.



Show API

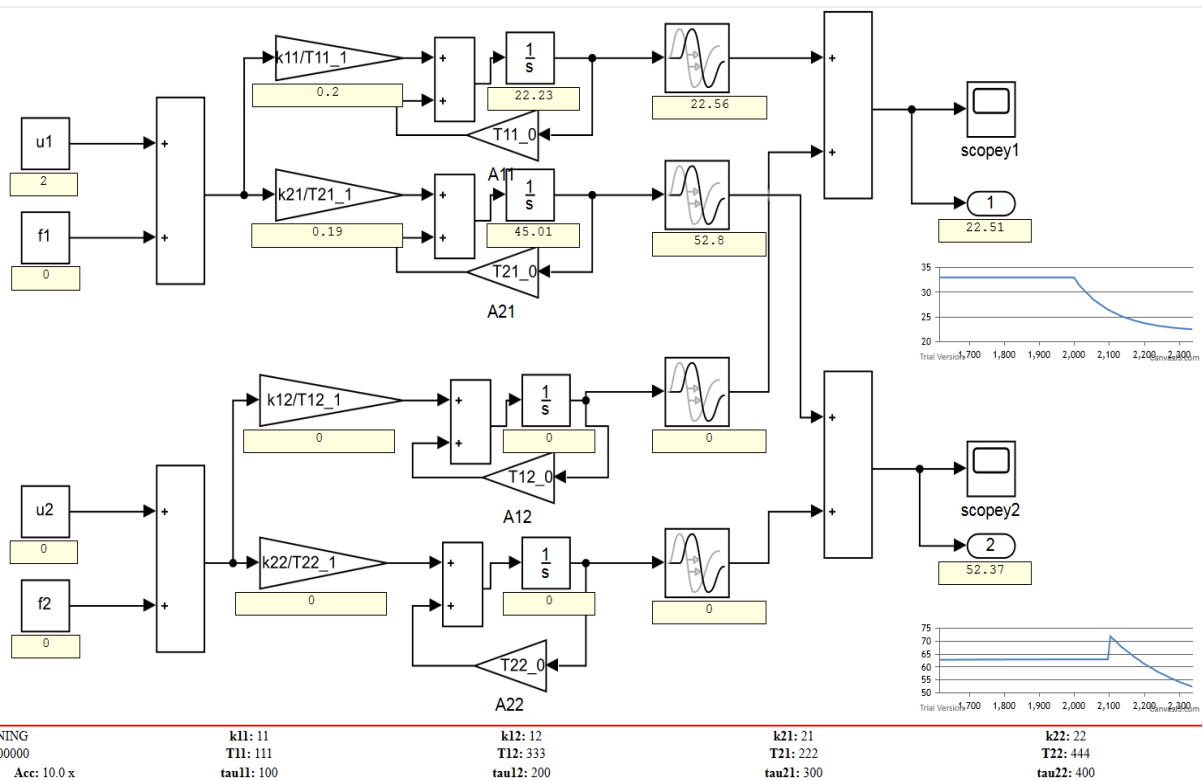


Рис. 4 – Пример работающего web-интерфейса
Fig. 4 – Example of a working web-interface

В статье предложена технология проектирования автономного web-сервиса, который реализует Simulink модель в реальном времени и не требует для своей работы Matlab. Технология заключается в разработке комплекса из двух связанных отдельных приложений: консольного приложения на языке Си на базе кода, который генерирует Simulink Coder и приложения на языке Python, которое реализует функционал web-сервиса.

Технология разработки программного решения исходит из соответствия следующим требованиям:

1) быстрая разработка; 2) переносимость, т.е. возможность работать как в ОС общего назначения, реального времени и в ОС для одноплатных компьютеров; 3) низкие требования к ресурсам; 4) стандартный программный интерфейс удаленного доступа.

Рассмотрены этапы проектирования на примере разработки автономного web-сервиса для простого объекта, описываемого матрицей передаточных функций 2×2 .

Результаты исследования рекомендуется использовать для: 1) разработки «цифровых двойников» для отладки компонентов систем автоматизации; 2) быстрой разработки регуляторов сложной структуры (при условии, что она эффективно может быть представлена в виде отдельных блоков); 3) дистанционного обучения (для удаленного многопоточного доступа студентов к Simulink моделям).

Дальнейшее направление исследований планируется в области усовершенствования предложенного подхода в области создания цифровых двойников для сложных технологических объектов, таких как спиртовая БРУ [9], разработки регуляторов с неклассическими подходами в области восстановления переменных состояния для объектов с существенными перекрестными связями, таких как атмосферная нефтяная ректификационная колонна [10, а также в области дистанционного обучения в рамках LMS платформы.

Литература

- [1] Puerto R., Jiménez L.M., Reinoso O. Remote control laboratory via Internet using Matlab and Simulink. *Computer Applications in Engineering Education*, 2010. T. 18, № 4. С. 694-702.
- [2] Janík Z., Žáková K. Online design of Matlab/Simulink and SciLab/Xcos block schemes, *14th International Conference on Interactive Collaborative Learning*. Piastany, 2011. С. 241-247.
- [3] Cirka L'uboš, Kalúz Martin. A Web-based Tool for Design of Simulink Models, *21st International Conference on Process Control*, 2017. С. 92-97.
- [4] Sysel Martin, Vaclavsky Michal. Remote Control and Monitoring in the Simulink. *International Journal of Circuits, Systems and Signal Processing*, 2013. T. 7. № 2. С. 83-90.



- [5] Bailey Bill. Connecting Simulink models to REST APIs. URL: <https://billbailey.io/2020/07/10/connecting-simulink-models-to-rest-apis/>.
- [6] Cirka L', Kal'uz M., Fikar M. On-line remote control of MATLAB simulations based on asynchronous communication model. *Matlab conference in Institute of Chemical Technology, Prague, 2013*. URL: http://dsp.vscht.cz/konference_matlab/MATLAB13/prispevky/014_cirka.pdf
- [7] Matsim: A sleek, intuitive interface for building Simulink models from a Matlab script. URL: <https://github.com/gave92/Matsim>.
- [8] Brennan Stephan. Write a Shell in C. URL: <https://brennan.io/2015/01/16/write-a-shell-in-c/>
- [9] Стопакевич А. А. Многомерная модель брагоректификационной установки косвенного действия. *Наукові праці ОНАХТ*, 2010. Т. 1., №38. С. 360-365.
- [10] Стопакевич А. А. Разработка робастной системы управления колонной атмосферной перегонки нефти *Восточно-европейский журнал передовых технологий*, 2015. Т. 5., №2 (77). С. 49-57.

References

- [1] R. Puerto, L.M. Jimenez and O. Reinoso, "Remote control laboratory via Internet using Matlab and Simulink," *Computer Applications in Engineering Education*, vol. 18, iss. 4, pp. 694-702, 2010.
- [2] Z. Janík and K. Zhakova, "Online design of Matlab/Simulink and SciLab/Xcos block schemes," *14th International Conference on Interactive Collaborative Learning, Piestany*, pp. 241-247, 2011.
- [3] L. Cirka and M. A. Kaluz, "Web-based Tool for Design of Simulink Models", *21st International Conference on Process Control*, pp. 92-97, 2017.
- [4] M. Sysel and M. Vaclavsky, "Remote Control and Monitoring in the Simulink," *International Journal of Circuits, Systems and Signal Processing*, vol. 7. iss. 2, pp. 83-90, 2013.
- [5] B. Bailey, *Connecting Simulink models to REST APIs*. URL: <https://billbailey.io/2020/07/10/connecting-simulink-models-to-rest-apis>.
- [6] L. Cirka, M. A. Kaluz and M. Fikar, "On-line remote control of MATLAB simulations based on asynchronous communication model," *Matlab conference in Institute of Chemical Technology, 2013, Prague*. URL: http://dsp.vscht.cz/konference_matlab/MATLAB13/prispevky/014_cirka.pdf
- [7] Matsim: a sleek, intuitive interface for building Simulink models from a Matlab script. URL: <https://github.com/gave92/Matsim>.
- [8] Stephan Brennan, *Write a Shell in C*. URL: <https://brennan.io/2015/01/16/write-a-shell-in-c/>
- [9] A. A. Stopakevich, "Mnogomernaja model' bragorektifikacionnoj ustanovki kosvennogo dejstvija," *Naukovi pratsi ONAKHT*, vol. 1, iss. 38, pp. 360-365, 2010.
- [10] A. A. Stopakevich, "Robust control system design of crude oil atmospheric distillation column," *Eastern-European Journal of Enterprise Technologies*, vol. 5., no. 2(77), pp. 49-57, 2015, DOI: <https://doi.org/10.15587/1729-4061.2015.50964>

УДК 004.42

ДОСЛІДЖЕННЯ МЕТОДІВ COMPUTER VISION ДЛЯ ВИКОРИСТАННЯ В АВТОМАТИЗОВАНИХ СИСТЕМАХ

Голубєв Л. П.¹, Ківа І. Л.², Матяш О. В.³

¹ Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

^{2,3} Київський національний університет технологій та дизайну, м. Київ, Україна

ORCID: ¹ <http://orcid.org/0000-0002-2980-8017>, ² <http://orcid.org/0000-0002-2920-5312>

E-mail: ¹golubevl@ukr.net, ²leonidovich1259@gmail.com, ³matasaleksandr97@gmail.com

Copyright © 2020 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i4.1932>



Анотація. У статті виконано дослідження методів комп'ютерного стерео-зору за допомогою засобів бібліотеки OpenCV. Розглянуто також методи калібрування камер стерео-зору і побудови карти глибин дальності. На виробництві в технологічних процесах, в процесах контролю якості виробів, а також в роботизованих системах часто виникають завдання визначення відстані до об'єкта. Традиційні методи визначення відстані побудовані на базі використання ультразвукових і інфрачервоних датчиків. Однак, застосування цих пристроїв не забезпечує можливості розпізнавання образів. Тому в статті використано новий підхід до вирішення завдання визначення відстані до об'єкта на основі застосування стерео-зору. В якості візуальних елементів стерео-зору були використані WEB-камери Logitech C-170. Для визначення відстані до об'єкта за допомогою стерео-зору необхідно з початку виконати побудову карти глибини об'єктів. В якості програмних засобів, що забезпечують побудову карти глибини була обрана бібліотека з відкритим програмним кодом OpenCV. Використання цієї бібліотеки в некомерційних цілях є безкоштовною, проте при її використанні у розробників часто виникають труднощі у зв'язку неповним описом використовуваних функцій. Алгоритм побудови карти глибини складається з наступних кроків: калібрування камер відео-пари, усунення дисторсії зображень, ректифікація зображень і побудова карти глибини, на основі зображення з лівої і правої камер стерео-пари. Розроблена на базі міні-комп'ютера Orange-PI з використанням методів Computer Vision система побудови карти глибини сцени показала високу ефективність і дозволяє визначити відстань до об'єктів сцени.

Abstract. The article examines the methods of computer stereo vision using the OpenCV library. Methods of stereo cameras calibration and construction of a map of depths of range are also considered. In production in technological processes, in the processes of product quality control, as well as in automated systems, the task of determining the distance to the object often arises. Traditional methods of distance determination are based on the use of ultrasonic and infrared sensors. However, the use of these devices does not provide the ability to recognize images. Therefore, the article uses a new approach to solving the problem of determining the distance to the object based on the use of stereo vision. Logitech C-170 WEB cameras were used as visual elements of stereo vision. To determine the distance to the object using stereo vision, it is necessary to build a map of the depth of objects from the beginning. The OpenCV open source library was chosen as the software to build the depth map. Use of this library for non-commercial purposes is free, but when using it, developers often have difficulties due to the incomplete description of the functions used. The algorithm for constructing a depth map consists of the following steps: calibration of video pair cameras, image distortion elimination, image rectification and construction of a depth map, based on images from the left and right stereo pair cameras. Developed on the basis of the Orange-PI mini-computer using Computer Vision methods, the system of plotting the depth of the scene showed high efficiency and allows you to determine the distance to the objects of the scene.

Ключові слова: стерео-зір, Computer Vision, веб-камера, карта глибини

Keywords: stereo vision, Computer Vision, web-camera, range map

Вступ

Стерео-зір – створення зорового образу об'єкта, що спостерігається за рахунок обробки зображення об'єкта з різних сторін в предметному просторі. В реальний час воно широко використовується в технологічних процесах промислового виробництва, при контролі якості, а також в ігровій індустрії [3-5]. Однак особлива необхідність у використанні стерео-зору виникає в роботі мобільних роботизованих систем. Тому, що мобільний робот постійно повинен вирішувати задачу визначення свого положення в просторі, розпізнавання об'єктів, що знаходяться в його оточенні і визначення відстані до цих об'єктів.

Тому виникає завдання дослідження використання методів стерео-зору для вирішення перерахованих вище завдань.

Аналіз літературних даних і постановка проблеми

У 2001 р дослідники корпорації Intel оголосили про створення бібліотеки Open Source Computer Vision (OpenCV) для реалізації функцій стереоскопічного комп'ютерного зору [6, 7]. Вона являє собою інструментарій, який налічує понад 500 функцій обробки і аналізу зображень для створення додатків комп'ютерного зору, в тому числі засоби взаємодії людини з комп'ютером. Однак процедури стерео-зору, представлені в цій бібліотеці описані в літературі, на наш погляд, недостатньо повно. У роботах [1, 2] присутній короткий опис основних команд системи стерео-зору, однак, не описані докладно параметри використовуваних функцій, а вони є визначальними в якості роботи системи відео-зору мобільного робота.

Мета і завдання дослідження

Раніше традиційно для визначення дистанції мобільного пристрою до перешкоди в мобільних роботизованих пристроях використовувалися ультразвукові та інфрачервоні датчики. Вони досить добре справляються із завданням визначення відстані до об'єкта в діапазоні до 5м. Однак мобільні роботи оснащені цими датчиками зовсім позбавлені можливості розпізнавати об'єкти і, отже, адекватно оцінювати виникає ситуацію або слідувати за заздалегідь за програмованим маршрутом. Тому виникає завдання дослідження можливостей computer vision для використання в мобільних роботів. Метою дослідження є побудова карти глибин, використовуючи засоби бібліотеки OpenCV.

**Результати досліджень**

Карта глибин (depth map) - це зображення, на якому для кожного пікселя, замість кольору зберігається його відстань до камери.

В якості стереосистеми для мобільного робота використовувалися дві Web камери Logitech C-170, закріплені на твердій рівній поверхні і на фіксованій відстані одна від одної. Система стерео-зору була підключена до міні комп'ютера OrangePi, що працює під управлінням ОС Debian 10.

Принцип роботи системи заснований на побудові карти глибин представлений на рисунку 1.

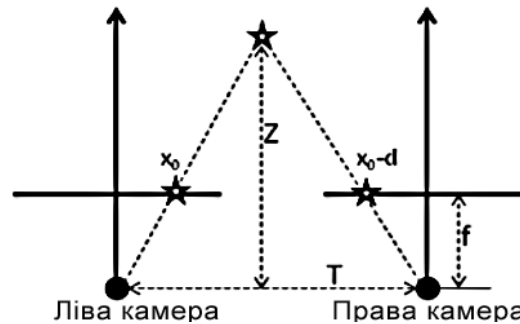


Рис. 1 – Система стерео-зору на базі двох відео камер

Fig. 1 – Stereo vision system based on two video cameras

Для кожного пікселю картинки з лівої камери з координатами (x_0, y_0) здійснюється пошук відповідного пікселя на зображенні з правої камери. При цьому координати картинки з правої камери будуть відрізнятися від координат з лівої камери на величину зміщення d тобто (x_0-d, y_0) . Пошук відповідного пікселя виконується шляхом обчислення максимуму функції відгуку, в якості якої зазвичай використовують кореляцію околиць пікселів.

Тоді залежність між зміщенням і глибиною пікселів можна виразити наступною формулою:

$$Z = \frac{f \cdot T}{d}, \quad (1)$$

де Z – відстань до пікселя; f – фокусна відстань камери; T – відстань між камерами; d м зміщення пікселя.

Для отримання даних про глибину об'єкта на сцені за допомогою зображень, одержуваних при обробці відео-потoku від двох камер необхідно виконати калібрування камер і далі побудувати карту глибин на основі формули (1).

У OpenCV для калібрування використовується метод шахівниці (ChessBoards) (рис.2). Як об'єкт калібрування в цьому методі використовується лист з чорними і білими квадратами відомого розміру. Перевагою методу є: відсутність пріоритету до горизонтальної або вертикальної осі [6].



Рис. 2 – Приклад шаблону шахівниці

Fig. 2 – Example of a chessboard template

Цей метод реалізований в OpenCV функцією:

```
int cvFindChessboardCorners (const void* image, CvSize pattern_size, CvPoint2D32f* corners),
```

де *image* – зображення, що містить шахову дошку; *pattern_size* – показує, скільки кутів у кожному рядку і стовпці дошки; *corners* – покажчик на масив, в якому можуть бути записані розташування кутів.



У разі успішного виконання (функція успішно знайшла всі кути), повертається значення що буде кількістю кутів шахівниці, в іншому випадку - як результат видається 0. Далі два зображення зберігаються в окремих змінних структури `IplImage`. Результатом виконання даного кроку є усунення будь-яких дефектів у зображеннях, які виходять з камер.

Далі виконується калібрування камер стереопари за допомогою бібліотечної функції OpenCV `calibrate Camera ()`. Після цього проводиться калібрування всієї стереопари. Дані, отримані на попередньому кроці подаються на вхід бібліотечної функції OpenCV `stereo Calibrate ()`, яка розраховує внутрішні параметри стереопари. Після процедури стерео калібрування ми отримуємо фокусну відстань камер в пікселях і координати принципів точок камер.

Розглянемо більш докладно побудову карти диспарантностей.

Для кожного пікселя лівої картини з координатами (x_0, y_0) виконується пошук пікселя на правій картинці. При цьому передбачається, що піксель на правій зображенні повинен мати координати (x_0-d, y_0) , де d - невідповідність / зміщення (disparity) (рис. 3).

Disparity задає кількість точок, сегментованих по площинах. Площині розташовані на різній відстані від камер (Cameras). Пошук відповідного пікселя виконується шляхом обчислення максимуму функції відгуку, в якості якої може виступати, наприклад, кореляція околиць пікселів. В результаті виходить карта зсувів (disparity map).

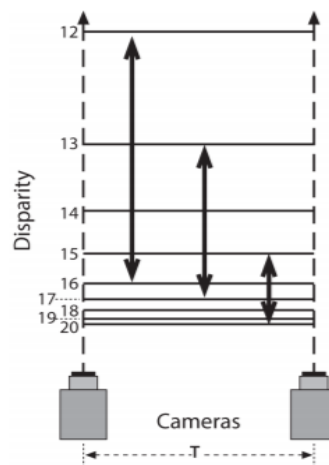


Рис. 3 – Побудова карти зсувів
Fig. 3 – Construction of a map of landslides

Далі по парі відповідних точок, знайдених раніше, можна визначити координати образу в тривимірному просторі. Для цього використовуємо функцію:

`cvReprojectImageTo3D (const CvArr* disparity, CvArr* _3dImage, const, CvMat* Q),`

де `disparity` – карта відмінностей; `_3dImage` – масив, що містить тривимірне зображення; `Q` – матриця перебудовувань, яка виходить після калібрування матриці і містить параметри:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -cx \\ 0 & 1 & 0 & -cy \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T} & (cx - cx')/T \end{bmatrix} \quad (2)$$

де T – відстань між камерами по осі X (передбачається, що координати Y однакові); f – фокусна відстань; cx, cy – координати точки на головному (центральному) промені лівої камери; cx' – x координата на правому зображенні лівої точки (це робиться для того, що уточнити, як пов'язана відстань між камерами і кількість пікселів)

Для ілюстрації програмної реалізації побудови матриці диспарантності далі наводиться фрагмент програми мовою C++:

```
// створення об'єкта для лівої камери
VideoCapture leftCam(0); //lets say 0 is left, 1 is right
if (leftCam.isOpened() == false){cout << "error: Webcam connect unsuccessful\n";
return(0); }
// створення об'єкта для правої камери
VideoCapture rightCam(1); //lets say 0 is left, 1 is right
if (rightCam.isOpened() == false){cout << "error: Webcam connect unsuccessful\n";
return(0); }
```



```
// створення матриці зображень лівої і правої камери
Mat left, right;
Mat leftClone, rightClone;
char charCheckForEscKey = 0;
while ( charCheckForEscKey != 27 && leftCam.isOpened() )
{
// читання зображення з лівої камери
leftCam.read(left);
if (left.empty()){cout << "No frame to read" << endl; break;}
leftClone = left.clone(); //copy from the left camera
// запис зображення з лівої камери в файл
imwrite("left.png", leftClone); // write it to screenshot.png in this
directory
// читання зображення з правої камери
rightCam.read(right);
if (right.empty()){cout << "No frame to read" << endl; break;}
rightClone = right.clone(); //copy from the left camera
// запис зображення з правої камери в файл
imwrite("right.png", rightClone); // write it to screenshot.png in this
directory
// формування матриці зображення лівої камери
Mat im_left = imread("left.png"); //left cam picture
// формування матриці зображення правої камери
Mat im_right = imread("right.png"); // right cam picture
Size imagesize = im_left.size();
// формування матриці диспарантності лівої камери
Mat disparity_left= Mat(imagesize.height,imagesize.width,CV_16S);
// формування матриці диспарантності правої камери
Mat disparity_right=Mat(imagesize.height,imagesize.width,CV_16S);
Mat g1,g2,disp,disp8;
// перетворення зображення лівої камери в градації сірого
cvtColor(im_left,g1, COLOR_BGR2GRAY);
// перетворення зображення правої камери в градації сірого
cvtColor(im_right,g2, COLOR_BGR2GRAY);
// створення об'єкта стерео-зору
Ptr<cv::StereoBM> sbm = StereoBM::create(0,21);
// установка параметрів об'єкта StereoBM
sbm->setDispl2MaxDiff(1);
sbm->setSpeckleRange(8);
sbm->setSpeckleWindowSize(9);
sbm->setUniquenessRatio(0);
sbm->setTextureThreshold(507);
sbm->setMinDisparity(-39);
sbm->setPreFilterCap(61);
sbm->setPreFilterSize(5);
sbm->compute(g1,g2,disparity_left);
normalize(disparity_left, disp8, 0, 255, NORM_MINMAX, CV_8U);
// виведення зображення лівої камери у вікні «Left»
namedWindow("Left", WINDOW_AUTOSIZE);
imshow("Left", im_left);
// виведення зображення правої камери у вікні «Right»
namedWindow("Right", WINDOW_AUTOSIZE);
imshow("Right", im_right);
// виведення зображення карти глибини в вікні «Depth map»
namedWindow("Depth map", WINDOW_AUTOSIZE);
imshow("Depth map", disp8);
// виведення клонованого зображення лівої камери у вікні «Left Cloned»
namedWindow("Left Cloned", WINDOW_FREERATIO);
imshow("Left Cloned", leftClone); // left is the left pic taken from camera 0
charCheckForEscKey = waitKey(1);
}
return(0);
}
```



Процес побудови карти глибин з використанням стерео-зору наведено на рисунках 4-6.

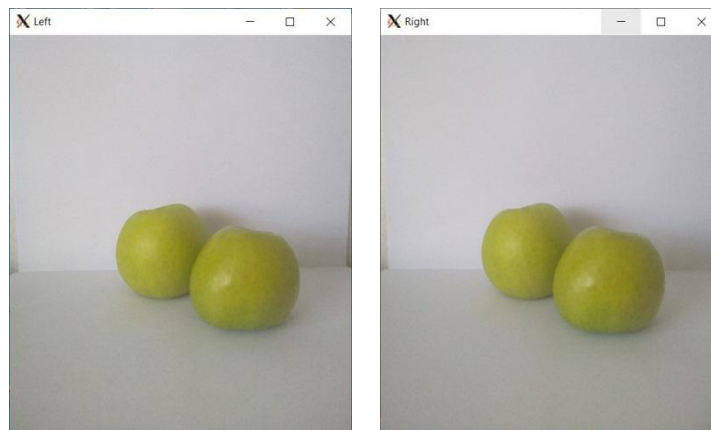


Рис. 4 – Вихідні зображення з лівої і правої камер відповідно
Fig. 4 – Input images from the left and right cameras, respectively

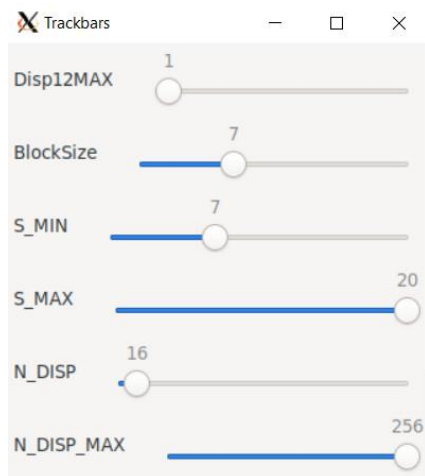


Рис. 5 – Слайдери для управління параметрами об'єкту StereoBM

Fig. 5 – Sliders to control the parameters of the StereoBM object

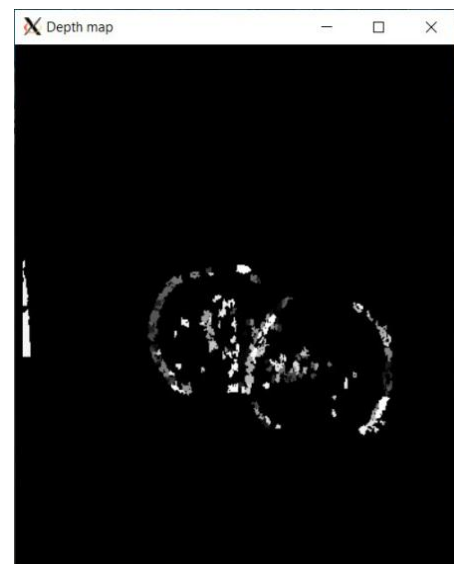


Рис. 6 – Побудована карта глибини
Fig. 6 – Constructed depth map

Основні параметри, об'єкта StereoBM, що налаштовуються:

minDisparity – мінімально можливе значення диспаратності. Зазвичай цей коефіцієнт дорівнює нулю, але іноді алгоритми випрямлення можуть зрушувати зображення, тому цей параметр необхідно відповідним чином налаштувати;

numDisparities – різниця між максимальною і мінімальною невідповідністю. Значення цього параметра завжди більше нуля. У поточній реалізації цей параметр повинен бути кратний 16;

blockSize – розмір блоку. Це повинно бути непарне число ≥ 1 . Зазвичай цей параметр повинен бути в діапазоні від 3 до 11;

P1 – перший параметр, керуючий гладкістю диспаратності;

P2 – другий параметр, керуючий гладкістю диспаратності. Чим більше ці значення, тим більше гладким є невідповідність;

disp12MaxDiff – максимально допустима різниця (в цілочисельних піксельних одиницях) при перевірці невідповідності зліва направо. Для відключення перевірки треба встановити негативне значення;

preFilterCap – значення усічення для попередньо відфільтрованих пікселів зображення. Алгоритм спочатку обчислює x-похідну в кожному пікселі і обрізає її значення з інтервалом $[-preFilterCap, preFilterCap]$. Отримані значення передаються в функцію вартості пікселів Берчфілд-Томасі.

З огляду на величину освітленості об'єктів в кадрі, завантаженості сцени і її зашумленості для отримання найкращого результату необхідно виконати настройку основних параметрів об'єкта StereoBM, змінюючи положення відповідних слайдерів.



Висновки

Розроблена стереосистема здатна визначати відстань до об'єктів, використовуючи стереоскопічний ефект в режимі реального часу. Застосування такої системи на мобільному роботизованому пристрої дозволяє підвищити безпеку його пересування. Стереосистема складається з блоку управління і двох ширококутних Web- відеокамер. Для визначення відстані в більш широкому куті огляду можна використовувати кілька стереопар, розташовуючи їх, наприклад, таким чином, щоб отримати повний тілесний кут.

Список використаних джерел

- [1] Bradski Gary, Learning OpenCV 3: Computer vision in C++ with the OpenCV library. Adrian Kaehler, Gary Bradski. O'Reilly Media -.2017, 1018 p.
- [2] Глория Буэно Гарсия Обработка изображений с помощью OpenCV//Глория Буэно Гарсия, Оскар Дениз Суарес, Хосе Луис Эспиноса Аранда, Хесус Салидо Терсеро, Исмаэль Серрано Грасиа, Ноэлия Валлез Энано, пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2016. – 210 с.: ил.
- [3] OpenCV Open Source Computer Vision URL: <http://opencv.org> (дата обращения: 28 мая 2016)
- [4] Gary Bradski, Adrian Kaehler Learning OpenCV // O'Reilly Media, 2008, P. 580
- [5] Batavia P., Singh S. Obstacle detection using adaptive color segmentation and color stereo homography // IEEE International Conference on Robotics and Automation. - 2001. - Vol. 1. - P. 705-710.
- [6] Tsai L., Hsieh J., Fan K. Vehicle detection using normalized color and edge map // 18th IPPR Conference on Computer Vision, Graphics and Image Processing. - Taipei, 2005. - P. 849-856.
- [7] Fleyeh H. Road and traffic sign color detection and segmentation - A Fuzzy Approach // IAPR Conference on Machine Vision Applications. - Tsukuba Science City, 2005. - P. 124-127.
- [8] Fuzzy-based algorithm for color recognition of license plates / F. Wang, L. Man, B. Wang, Y. Xiao, W. Pan, X. Lu // Pattern Recognition Letters. - 2008. - Vol. 29. - P. 1007-1020.

References

- [1] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer vision in C++ with the OpenCV library*. O'Reilly Media, 2017, 1018 p.
- [2] G. Bueno Garsiya et al, *Obrabotka izobrazheniy s pomoshchyu OpenCV*, per. s angl. Slinkin A., Moscow, DMK Press, 2016, 210 p.
- [3] *OpenCV Open Source Computer Vision* (2016, May 28) [Online]. Available: <http://opencv.org>
- [4] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly Media, 2008, 580 p.
- [5] P. Batavia and S. Singh, "Obstacle detection using adaptive color segmentation and color stereo homography," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2001, pp. 705-710.
- [6] L. Tsai, J.Hsieh and K. Fan, "Vehicle detection using normalized color and edge map," in *18th IPPR Conference on Computer Vision, Graphics and Image Processing*, Taipei, 2005, pp. 849-856.
- [7] H. Fleyeh, "Road and traffic sign color detection and segmentation - A Fuzzy Approach," in *IAPR Conference on Machine Vision Applications*, Tsukuba Science City, 2005, pp. 124-127.
- [8] F. Wang et al, "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognition Letters*, vol. 29, pp. 1007-1020, 2008.

УДК 004.41

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА ПРИЙМАЛЬНОЇ КОМІСІЇ ОНАХТ ЯК ОСНОВА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ФОРМУВАННЯ КОНТИНГЕНТУ СТУДЕНТІВ

Мороз А. М.¹, Похлебіна Н. О.², Хобін В. А.³

Одеська національна академія харчових технологій, м. Одеса, Україна

ORCID: ¹ <http://orcid.org/0000-0002-6655-3578>, ² <http://orcid.org/0000-0002-9789-608X>,

³ <http://orcid.org/0000-0003-0238-8371>

E-mail: ¹andrey.work2001@gmail.com, ²pohlebina.odessa@gmail.com, ³khobin@onaft.edu.ua

Copyright © 2020 by author and the journal "Automation of technological and business-processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i4.1933>



Анотація. Стаття присвячена дослідженню автоматизації процесу формування контингенту студентів (ФКС) у ході поточної вступної кампанії у Одеській національній академії харчових технологій. Розглянуто основні цілі та завдання, що виконує приймальна комісія закладу вищої освіти. Розглянуто практичне застосування автоматизованої інформаційно-аналітичної системи (ІАС). Наведено процеси, що відбуваються в інформаційно-аналітичній системі. Виділені особливості системи як об'єкта управління. Представлена розроблена інформаційна система, що забезпечує наскрізну автоматизацію управління роботою приймальної комісії на прикладі Одеської національної академії харчових технологій. Описано функціонування автоматизованої системи приймальної комісії (ПК) закладу вищої освіти. Наведено повний алгоритм дій, що виконує секретар відбіркової комісії з моменту прийняття заяви або документа до оголошення рейтингу та формування списків на зарахування. Представлено зовнішній вигляд основних інтерфейсів автоматизованого робочого місця оператора, його функції та задачі. Створено систему логування та контролю доступу до функцій автоматизованої інформаційно-аналітичної системи, що складається з декількох рівнів допуску, що забезпечує безпеку і цілісність даних, розділяє користувачів на операторів і адміністраторів. Інформаційно-аналітична система управління розглянута як система з автоматизованою технологією отримання результатів організації, документального оформлення. Наведено структурні схеми інформаційної системи та її підсистем, що дозволяє детально розглянути кожну розроблену функцію та її можливості.

Abstract. The article is devoted to the study of the automation of the process of forming a contingent of students (FCC) during the current admission campaign of ONAFT. The main goals and objectives are considered, performed by the selection committee of a higher educational institution. The practical application of the automated information and analytical system (IAS) is considered. The processes occurring in the information-analytical system (IAS) are presented. Highlighted features of the system as a control object. The developed information system is presented, which provides end-to-end automation of management of the work of the selection committee on the example of the Odessa National Academy of Food Technologies. The functioning of the automated system of the selection committee (PC) of the university is described. The full algorithm of actions is given, which is done by the secretary of the selection committee from the moment the application or document is accepted for the announcement of the rating and the formation of lists for enrollment. The appearance of the main interfaces of the automated workstation of the operator, its functions and tasks is presented. The created system of logging and access control to the functions of an automated information and analytical system, consisting of several levels of access, which ensures the safety and integrity of data, separates users into operators and administrators. IAS management is considered as a system with an automated technology for obtaining the results of the organization, documentation. Structural diagrams of the IC and its subsystems are given, allowing a detailed examination of each developed function and their capabilities.

Ключові слова: приймальна комісія, автоматизована система, інформаційна система управління приймальною комісією закладу, реєстрація абітурієнта, бізнес-процеси, інформаційні технології.

Keywords: admissions office, automated system, information management system of the institution's admissions office, applicant registration, business processes, information technology.

Вступ

Метою роботи приймальної комісії (ПК) всіх ВНЗ є формування контингенту студентів (ФКС) у ході поточної вступної кампанії. Ефективність досягнення цієї мети визначається, в кінцевому рахунку, кількістю зарахованих абітурієнтів та рівнем їх підготовки. В умовах, коли число потенційних кандидатів на зарахування до ВНЗ обчислюється тисячами, то, очевидно, що підвищення ефективності ФКС потребує автоматизації цих процесів.

Перший етап автоматизації передбачає створення інформаційно-аналітичної системи (ІАС), котра включає в себе мережу розподілених автоматизованих робочих місць всіх членів ПК, включаючи її голову і відповідального секретаря, що мають різні права доступу до ІАС. Основне завдання ІАС – забезпечити для членів ПК повне уявлення про кожного абітурієнта для прийняття обґрунтованого рішення про нього. При цьому до ІАС пред'являються дві взаємно суперечливі вимоги: а) неприпустимість помилок в ході зарахування; б) забезпечити високу швидкість при реалізації цих процедур.

Основою "зняття" протиріччя між зазначеними вище вимогами є забезпечення ІАС максимально якісним прийомом документів абітурієнта, їх реєстрацію та супровід, можливість участі у рейтингу. Можливість і ефективність вирішення цих завдань значною мірою визначає ефективність досягнення мети роботи ПК - високих показників ФКС. Відзначимо, що при функціонуванні ІАС відповідальність за прийняті рішення зберігається за ПК.

Аналіз варіантів ІАС [11,12], інформація про яких виявилася нам доступною, показав, що не одна з них не дозволяє ефективно вирішити поставлені завдання через їх малу функціональність. При цьому вартість цих ІАС, була неприйнятно висока. Внаслідок було прийнято рішення про розробку власної автоматизованої ІАС, яка б відповідала вимогам приймальної комісії та вирішувала б всі поточні завдання та мала б можливість розширення своїх функціональних можливостей при змінах нормативної бази.

Мета і завдання дослідження

Головною метою проведених досліджень є розробка автоматизованої системи управління (АСУ) ФКС. Як вже зазначалося основною і вихідною складовою АСУ ФКС повинна стати ІАС. Її виробництво не одноразовий акт, а процес «насичення» системи актуальними функціями, установки взаємозв'язків між ними, що відбуваються в ході експлуатації існуючих «перехідних» варіантів ІАС.

Початковий варіант ІАС, названий «Абітурієнт-ОНАХТ», було розроблено та впроваджено в ОНАХТ у 2010 році. Він призначений для проведення вступної кампанії та ведення документації на всіх етапах вступної кампанії закладу.



На рисунку 1 представлена структурна технічна схема взаємозв'язків між зовнішньою ІАС, сервером ІАС та автоматизованим робочим місцем (АРМ).

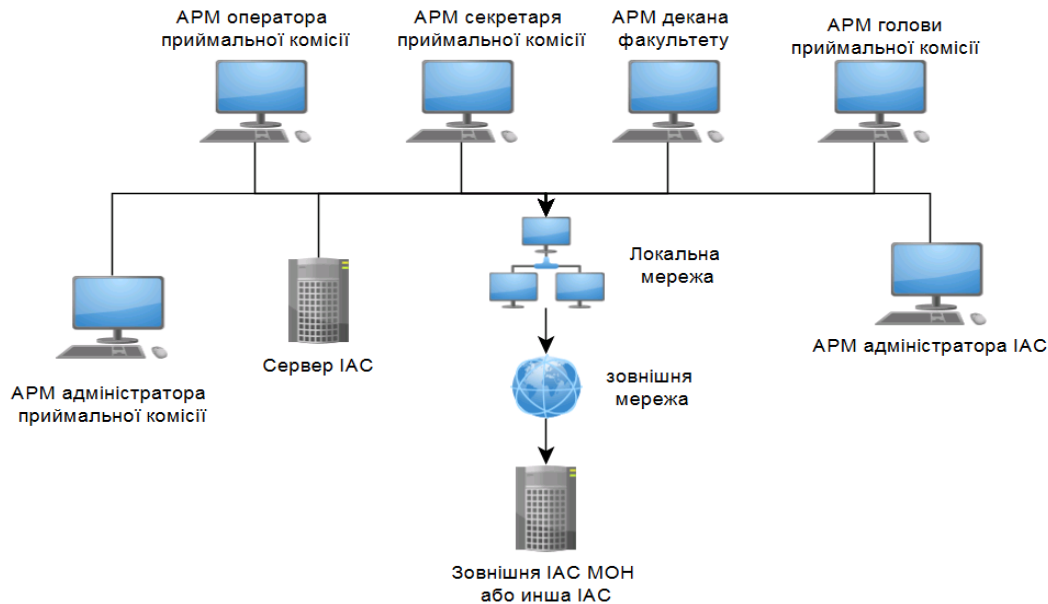


Рис.1 – Структурна технічна схема взаємозв'язків між зовнішньою ІАС, сервером ІАС та автоматизованим робочим місцем (АРМ)

ІАС умовно можна поділити на декілька підсистем - збирання даних, зарахування, аналітика та статистика. Вони взаємопов'язані між собою та утворюють цілісну систему представляючи необхідний функціонал для ефективного рішення задач.

Підсистема «збирання даних» реалізує наступні функції:

- введення даних абітурієнта оператором, їх перегляд та редагування;
- архівувати, коригувати й переглядати екзаменаційні оцінки абітурієнтів;
- імпорт даних із зовнішніх джерел (excel, word та інше), зовнішньої ІАС МОН та інших ІАС;
- генерування всіх необхідних документів (договори, протоколи, екзаменаційний лист, описаніє особової справи, розписка і інше) та заяв;
- експорт даних у форматі csv, xml, json або у зовнішні ІАС за допомогою JSON API;
- розподіл абітурієнтів на екзаменаційні групи для проведення вступних іспитів.

На рисунку 2 представлена загальна схема об'єктів підсистеми «збирання даних».

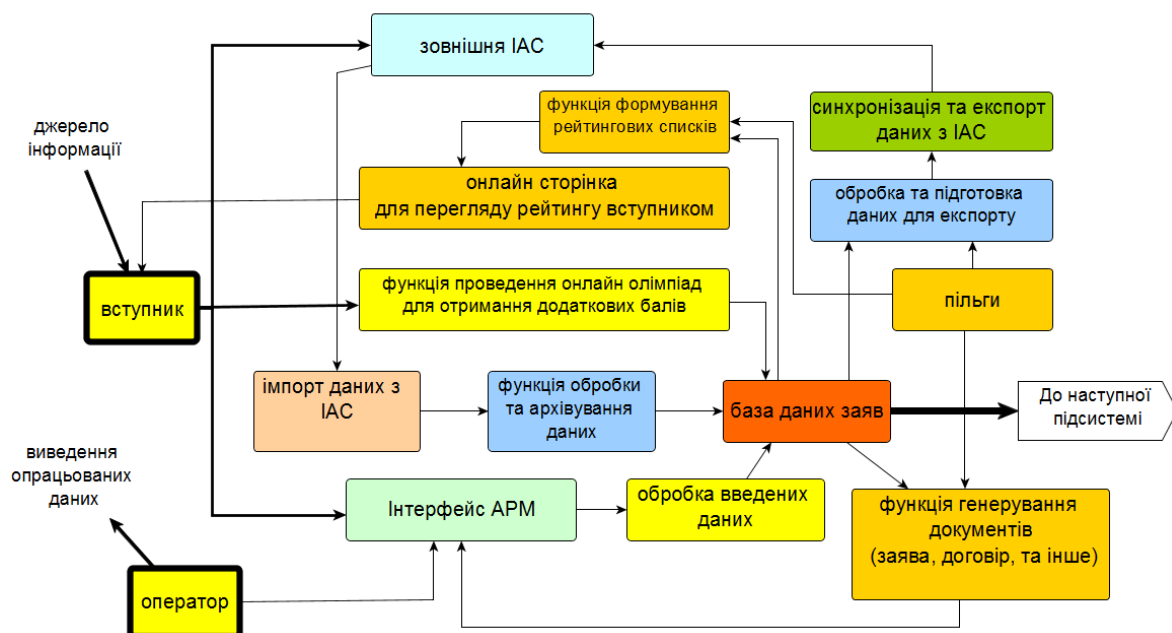


Рис.2 – Схема підсистеми «збирання даних»



Підсистема «зарахування» реалізує наступні функції:

- формування списків абітурієнтів для зарахування згідно рейтинговим списками;
- генерування наказів та інших документів (витяг та інше) для зарахування абітурієнтів;
- проводити зарахування абітурієнтів відповідно з правилами прийому до вищого навчального закладу;
- формувати екзаменаційні відомості, екзаменаційні листи, протоколи до зарахування, накази про зарахування, списки, довідки, реєстри, звіти та інші необхідні для роботи приймальної комісії (ПК) документи.

На рисунку 3 представлена загальна схема об'єктів підсистеми «зарахування».

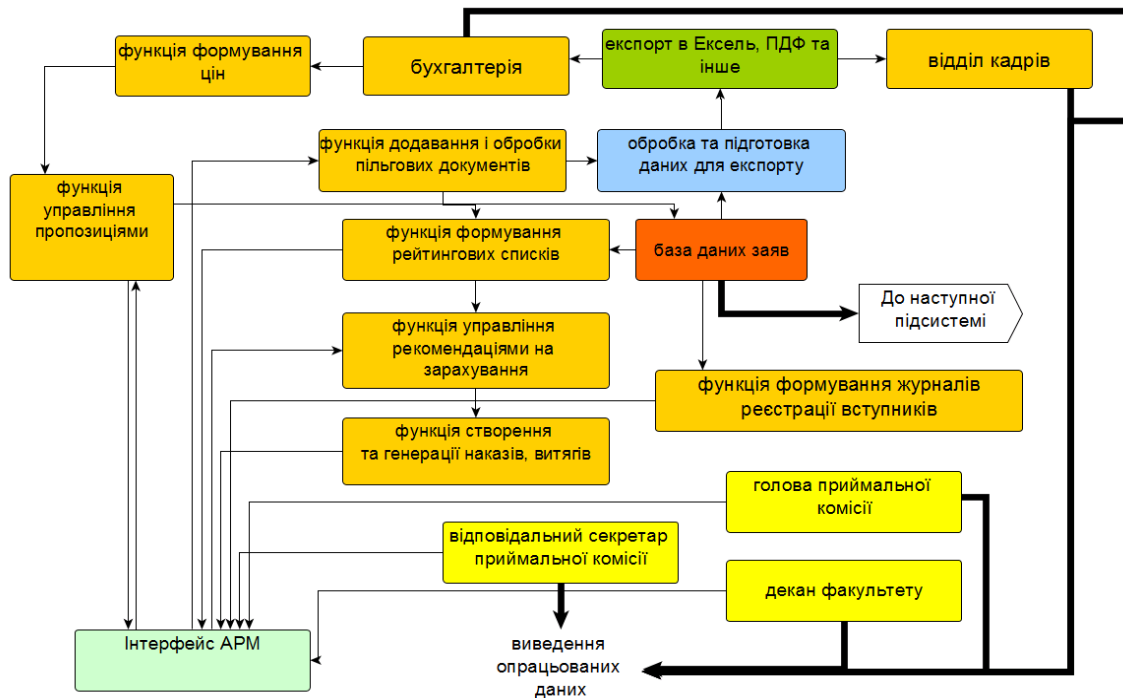


Рис.3 – Схема підсистеми «зарахування»

Підсистема «аналітика та статистика» реалізує наступні функції:

- отримання статистичних даних та їх аналітична обробка;
- генерування рейтингових списків;
- формувати статистики - географія абітурієнтів, графік вступу, статистика по факультетам та інше

На рисунку 4 представлена загальна схема об'єктів підсистеми «аналітики та статистики».

ІАС «Абітурієнт-ОНАХТ» використовується для підтримки приймальної кампанії в ОНАХТ вже 10 років і продовжує розвиватися. БД системи створена на базі реляційної СУБД (системи управління базами даних) MySQL Server. Система підтримує зарахування як за результатами вступних іспитів, що проводяться факультетами, так і з урахуванням результатів Зовнішнього незалежного оцінювання, що складається при закінченні середньої школи. Передбачена онлайн система олімпіад для отримання додаткових балів для вступу.

Найважливішим механізмом, що зачіпає основні напрямки освітньої діяльності, є інформаційна система ПК, яка розглядається як інструмент управління вузом. Саме вирішення цих питань присвячена ця стаття.

Результати досліджень

ІАС "Абітурієнт-ОНАХТ" - забезпечує організацію та супроводження діяльності приймальної комісії на всіх етапах, починаючи від аналізу плану набору абітурієнтів до формування контингенту студентів. Основними її завданнями є:

- формування плану набору абітурієнтів на новий навчальний рік;
- прийом і оформлення документів абітурієнта за спеціальностями;
- формування різних звітів про хід роботи приймальної комісії;
- підготовка документів до рекомендації, підтвердження і зарахування абітурієнтів за спеціальностями;
- формування контингенту студентів першого курсу;
- керування користувачами;
- звірка даних операторів і даних системи;
- створення рейтингового списку для зарахування;
- отримання статистичних даних;
- обмін інформацією із зовнішньою ІАС МОН.

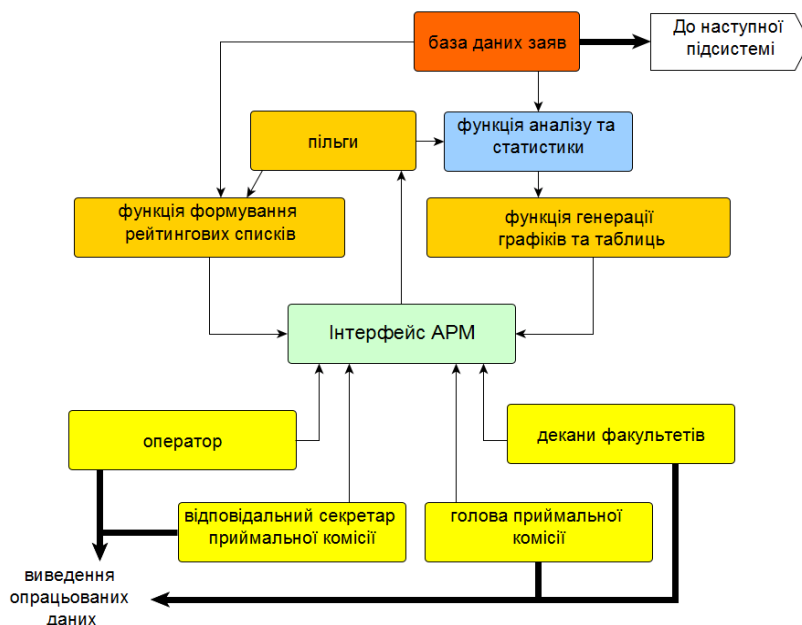


Рис.4 – Схема підсистеми «аналітика та статистика»

У системі є кілька рівнів допуску з відповідними правами та можливостями:

- оператор (може переглядати дані по своєму факультету та працювати тільки з ним);
- секретар приймальної комісії (має доступ до всіх факультетів і можливість редагування);
- декан факультету (може тільки переглядати дані за своїм факультету).

Таким чином кожен факультет має свій персональний логін та пароль для авторизації у системі, а також має певний набір функцій та можливостей. Також ведеться логування дій користувачів з функцією скасування, що дозволяє уникати помилок при введенні та редагуванні даних. Система розподілена на три групи функцій: повністю автоматичні, ті, у які вводяться дані, та ті, що виводять дані.

Структурна схема ІАС представлена на рисунку 5.

Для кожного абітурієнта створюється новий запис, в рамках якого проводяться операції:

- перевірка необхідних документів;
- рекомендація, підтвердження на зарахування за фахом;
- підтвердження проведення платежу за навчання в касі;
- розподіл за формою навчання;
- скасування та виправлення помилок;
- формальне видалення записів.

Функції генерації документів (заяв, договору ті інше), формування рейтингових списків, формування журналів реєстрації вступників взаємодіє із базою даних до якої інформація надходить в результаті синхронізації із базою зовнішнього ІАС МОН. Таким чином наша ІАС отримує усю необхідну інформація на вступника, що дозволяє прискорити процес роботи.

Функції управління рекомендаціями на зарахування та створення і генерації наказів/витягів дозволяє скоротити час на формування рейтингів та бачити усіх рекомендованих на зарахування в режимі реального часу.

Заключаючи угоду із вступником про надання освітніх послуг договір повинен містити ціни та обрану освітню програму. У системі розроблена підсистема з взаємодією із відділом кадрів та бухгалтерією, що дозволяє генерувати вартість послуг у договір.

На базі ОНАХТ вступник має можливість записатись на додаткові підготовчі курси та брати участь в онлайн олімпіаді та отримати додаткові бали при зарахуванні до закладу. Онлайн олімпіади інтегровані з ІАС це дає можливість бачити додаткові бали абітурієнтів, що їх проходили та враховувати при розрахунку конкурсного балу.

Маючи високий рівень кількості заяв важливою функцією є аналіз та статистика, що допомагають секретарям ПК слідкувати та виділяти певних вступників та побудувати статистику відносно до кількості заяв, пріоритетності.

Для реалізації всіх цих функцій було створено автоматизоване робоче місце за допомогою WEB-технологій, реалізація інтерактивного режиму роботи користувача з виконанням необхідних перевірок безпосередньо у процесі введення даних до бази. ІС представляє собою єдиний WEB-інтерфейс для оператора, що дозволяє обслуговувати абітурієнта.

Реєстрація абітурієнта проводиться оператором шляхом заповнення анкетних даних абітурієнта, окрім ПЗСО дані яких приходять до бази автоматично з зовнішньої системи ІАС МОН.

На рисунку 6 представлені деякі інтерфейси АРМ.

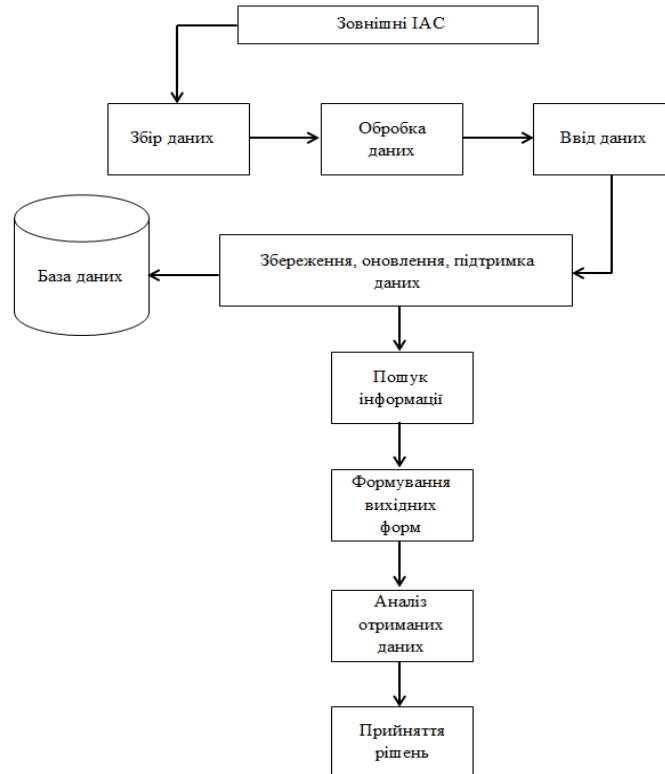


Рис. 5 – Структурна схема ІАС

The screenshot displays the BD-ONAXT web application interface. It features a top navigation bar with various icons and a main content area divided into three panels. The left panel shows a 'Заяви' (Applications) section with a table of applications. The middle panel shows a 'Статистика' (Statistics) section with a bar chart. The right panel shows a 'Додати абитуриєнта' (Add Applicant) form with various input fields for personal and academic data. Below the panels, there is a filter bar with dropdown menus for year, program, and form type.

ФК	Назва	Курс	Скор.	Рівень	Форма	Тип	Лич/Контр	Кільк.	Ориг.	Заяви	Рейтинг	Накази	Журнал
КС та А	141 Бакалавр (Електромеханічні системи з інтелектуальним керуванням) - акредитовано в 2020 році	1		Бакалавр	Денна	Відкрита	12/4	32	9				
КС та А	141 Бакалавр (Електромеханічні системи з інтелектуальним керуванням) - 2- тільки контракт - акредитовано в 2020 році	1		Бакалавр	Денна	Небюджетна	4/4	17	2				
КС та А	151 Бакалавр (Комп'ютерні системи та програмна інженерія в автоматизації) - акредитовано в 2020 році	1		Бакалавр	Денна	Відкрита	15/6	77	15				
КС та А	151 Бакалавр (Комп'ютерні системи та програмна інженерія в автоматизації) - 2- тільки контракт - акредитовано в 2020 році	1		Бакалавр	Денна	Небюджетна	10/10	32	6				
КС та А	151 Бакалавр (Штучний інтелект та автоматизація робототехнічних систем) - акредитовано в 2020 році	1		Бакалавр	Денна	Відкрита	11/6	47	5				
КС та А	151 Бакалавр (Штучний інтелект та автоматизація робототехнічних систем) - 2- тільки контракт - акредитовано в 2020 році	1		Бакалавр	Денна	Небюджетна	10/10	14	2				

Рис. 6 – Автоматизоване робоче місце оператора



Рекомендація на зарахування проводиться шляхом формування списку абітурієнтів використовуючи існуючі форми через web-браузер. Інформація заноситься в базу даних.

Підтвердження на зарахування проводиться оператором, після того як абітурієнт зробить оплату за навчання і здасть необхідні оригінали документів. Після підтвердження абітурієнтів на зарахування їм ставиться рекомендація на зарахування.

Зарахування проводиться оператором, після видання наказу про зарахування абітурієнтів на відповідні спеціальності. Необхідні документи можна сформувати і роздрукувати у вигляді рангового списку абітурієнтів у звітній системі через web-браузер.

Висновки

В ОНАХТ розроблена та введена у роботу ІАС "Абітурієнт-ОНАХТ" з управління ПК. ІАС забезпечує організацію та супроводження діяльності приймальної комісії на всіх етапах, починаючи від аналізу плану набору абітурієнтів до формування контингенту студентів. Ця система прискорила та спростила процес роботи з абітурієнтами, процес отримання персональних даних, формування звітів, створення рейтингових списків та оформлення справ. У результаті це дозволило створити ефективну систему управління приймальною комісією, значно збільшити потенціал та конкурентоздатність.

Список використаних джерел

- [1] Інтернет джерело: https://uk.wikipedia.org/wiki/Заклад_вищої_освіти.
- [2] Інтернет джерело: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/95/95.html>.
- [3] Зайцев А.С., Емельянов В.Ю., Воронин И.В. Автоматизация как основа оптимального планирования учебного процесса университета // *Мехатроника, автоматизация, управление*, 2007. № 5. С. 45-48.
- [4] Ларсон Б. Разработка бизнес-аналитики в Microsoft SQL Server 2005. Питер: Издательство, 2008.
- [5] Астахова И.Ф., Толстобров А.П. SQL в примерах и задачах: Учеб. пособие / Мн.: Новое знание, 2002.
- [6] Крюков В.В., Шахгельдян К.И. Корпоративная информационная среда вуза: методология, модели, решения: Монография. Владивосток: Дальнаука, 2007. 301 с.
- [7] Торобеков Б.Т., Таштобаева Б.Э., Савина Н.А. Особенности построения информационной системы «Офис регистратора» // *Проблемы автоматизации и управления. НАН КР. Бишкек: Илим*, 2007. С. 157-161.
- [8] Торобеков Б.Т., Сaitов Н.Ж. Информационное пространство образовательной деятельности ВУЗа. Совершенствование содержания и технологии учебного процесса: сб. трудов научно-методической конф. / Из-во ТПУ. Томск, 2010 г. С. 142-143.
- [9] Торобеков Б.Т. Автоматизация процессов приемной комиссии вузов на основе информационной системы // *Наука, техника и образование*. 2017. №4 (34), с. 30-34.
- [10] Інтернет джерело: 1С:Колледж управління середнім спеціальним навчальним закладом http://www.1c-college.ru/1c_college/price_list.php
- [11] Пакет програм "ПС-Абітурієнт" <http://www.politek-soft.kiev.ua>
- [12] Автоматизация информационных процессов - <http://infis.narod.ru/is/is-n3.htm>

References

- [1] Zaklad vyshchoi osvity. [Online] Available: <https://uk.wikipedia.org/wiki>
- [2] Information systems, their types; information system hardware and software [Online] Available: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/95/95.html>.
- [3] A.S. Zaitsev, V.Y. Emelyanov, I.V. Voronin, "Automation as the basis of optimal planning of the educational process of the university," *Mechatronics, automation, management*, no. 5, pp. 45-48, 2007.
- [4] B. Larson, *Development of business intelligence in Microsoft SQL Server 2005*. Peter, Publishing House, 2008.
- [5] I.F. Astakhova, A.P. Tolstobrov, *SQL in examples and problems: Textbook. Manual*, M.: New knowledge, 2002.
- [6] V.V. Kryukov, K.I. Shahgeldyan, *Corporate information environment of the university: methodology, models, solutions: Monograph*. Vladivostok: Dalnauka, 2007, 301 p.
- [7] B.T. Torobekov, B.E. Tashtobaeva, N.A. Savina, "Features of building an information system "Registrar's Office"," *Problems of automation and control*, NAS KR. Bishkek: Ilim, 2007. pp. 157-161.
- [8] B.T. Torobekov, N.J. Saitov, "Information space of educational activity of the university. Improving the content and technology of the educational process: SAT," *Proc. of the scientific and methodological conference*, TPU, Tomsk, 2010, pp. 142-143.
- [9] B.T. Torobekov, "Automation of processes of the reception commission of higher educational institutions on the basis of information system," *Science, technology and education*, no. 4(34), pp. 30-34, 2017, URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-protseessov-priemnnoy-komissii-vuzov-na-osnove-informatsionnoy-sistemy>
- [10] 1С:College of Secondary Special Education Management [Online]. Available: http://www.1c-college.ru/1c_college/price_list.php
- [11] Software package "PS-Abiturient" [Online], Politek-SOFT. Available: <http://www.politek-soft.kiev.ua>
- [12] Automation of information processes [Online]. Available: <http://infis.narod.ru/is/is-n3.htm>



УДК 681.51

МОДЕЛЮВАННЯ СИСТЕМИ РЕГУЛЮВАННЯ В STEP 7 TIA PORTAL З ВИКОРИСТАННЯМ ФУНКЦІОНАЛЬНИХ БЛОКІВ БІБЛІОТЕКИ «LSIM»

Левінський М.В.¹, Левінський В.М.²¹ Національний університет «Одеська морська академія», м. Одеса, Україна² Одеська національна академія харчових технологій, м. Одеса, УкраїнаORCID: ¹ <http://orcid.org/0000-0002-6544-5110>, ² <http://orcid.org/0000-0002-3563-528X>E-mail: ¹ maxlevinskyi@gmail.com, ² valerylevinskyi@gmail.com

Copyright © 2020 by author and the journal "Automation of technological and business-processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>DOI: <https://doi.org/10.15673/atbp.v12i4.1934>

Анотація. Актуальність. Моделювання систем регулювання в середовищі розробки програмного забезпечення контролерів, наприклад в Step 7 TIA Portal, дозволяє провести налагодження параметрів функціональних блоків ПІД-регуляторів, перевірку грубості САР до варіації параметрів об'єкта керування та безударність переходів між ручним та автоматичним режимом роботи. Мета. Показати приклади корекції програмного коду функціональних блоків бібліотеки "LSim", які фірма Siemens рекомендує для моделювання об'єкта керування, що забезпечують встановлення довільних початкових умов в цифрових аналогах диференціальних рівнянь динамічних ланок. Метод. Аналіз оригінального коду функціональних блоків бібліотеки "LSim" та його подальша корекція, моделювання скоригованих функціональних блоків в середовищі Step 7 TIA Portal. Результати. Наведені фрагменти кодів функціональних блоків, які дозволяють моделювати ланку запізнення, інерційну ланку першого порядку, інтегруючу ланку з довільними початковими умовами. Проведено моделювання скоригованих функціональних блоків в середовищі Step 7 TIA Portal та співставлення результатів із результатами моделювання аналогічних динамічних ланок в середовищі Simulink/MATLAB. Проведено моделювання САР із використанням зазначених блоків. Висновки. Запропоновані зміни коду функціональних блоків бібліотеки "LSim", які реалізують ланку запізнення, інерційну ланку першого порядку, інтегруючу ланку, дозволяють врахувати довільні початкові умови в цих ланках, що забезпечує моделювання динамічно сталих режимів САР в середовищі Step 7 TIA Portal.

Abstract. Topicality. Control systems modeling in PLC integrated development environment, for example in Step 7 TIA Portal, allows tuning of PID-controller blocks parameters, automatic control system robustness check when control object parameters change and "soft" change from manual to automatic operation modes. Goal. To show examples of "LSim" library function blocks code correction, which Siemens company recommends for control object modeling and which allow to set up arbitrary initial conditions in digital analogues of dynamic links differential equations. Means. "LSim" library function blocks original code analysis and its further correction, modeling of modified function blocks in Step 7 Tia Portal. Results. Fragments of function block codes, which allow to simulate delay link, first order aperiodic link, integrator link with modifiable initial conditions are presented. Corrected function blocks modeling in Step 7 TIA Portal environment was carried out as well as comparing results with modeling of the same function blocks in Simulink/MatLab environment. Automatic control system modeling with abovementioned function blocks was carried out. Conclusions. Proposed code changes of "LSim" library function blocks, which implement delay link, first order aperiodic link, integrator link allow to take into account arbitrary initial conditions in these links, which allow to model dynamically steady automatic control system modes in Step 7 TIA Portal.

Ключові слова: Step 7 TIA Portal, функціональні блоки бібліотеки "LSim", ПІД-регулятор**Keywords:** Step 7 TIA Portal, "LSim" library function blocks, PID-controller

Вступ

Розробка алгоритмів керування технологічними процесами проходить етапи імітаційного моделювання, наприклад в середовищі Simulink/MATLAB, де в системі взаємодіють моделі алгоритмів і об'єкта керування (ОК), моделювання в середовищі програмування контролера, наприклад в Step 7 TIA Portal, де в системі взаємодіють реальні програми керування і модель ОК, і етап напівнатурного моделювання, де програми керування завантажують в контролер, який по фізичним каналам зв'язку взаємодіє з комп'ютером, і в якому імітаційна модель ОК функціонує в реальному часі.



Аналіз літературних даних і постановка проблеми

В даній статті розглядається моделювання системи регулювання в середовищі Step 7 TIA Portal. Фірма Siemens рекомендує використовувати при цьому бібліотеку "LSim" Simulation Library, до складу якої входять шістнадцять функціональних блоків [1]. Вони імітують прості динамічні ланки, поєднуючи які між собою можна створити достатньо складні моделі ОК та коригуючих зв'язків.

Бібліотека "LSim" може застосовуватись при програмуванні контролерів Simatic S7-1200/S7-1500. При програмуванні попередніх серій контролерів S7-300/S7-400 використовувався аналогічний за призначенням продукт "Modular PID-Control" [2]. Порівняльний аналіз та практика використання цих продуктів виявили достатньо серйозний недолік функціональних блоків, які входять до складу бібліотеки "LSim". А саме – блоки, які програмно реалізують динамічні ланки і базуються на цифрових аналогах диференційних рівнянь, не дозволяють встановити довільні початкові умови в рівняннях, а лише нульові. Це суттєво звужує коло задач при їх використанні до найпростіших навчальних, оскільки опис моделей практично всіх технологічних об'єктів керування вимагає врахування ненульових початкових умов. Це пов'язано з накопиченням маси та енергії в агрегатах та установках, що здійснюють технологічний процес.

Мета і завдання дослідження

Відповідно мета даної статті – показати приклади корекції програмного коду функціональних блоків бібліотеки "LSim", які усувають існуючий недолік;

завдання – провести перевірку зроблених корекцій коду шляхом співставлення результатів моделювання системи регулювання в середовищах Step 7 TIA Portal та Simulink/MATLAB.

Методи і матеріали досліджень

Оскільки до складу бібліотеки "LSim" входять блоки з відкритим програмним кодом, це дало можливість провести аналіз кодів і визначити шляхи їх корекції. На рис. 1 в якості прикладу наведені фрагменти кодів функціональних блоків, які реалізують ланку запізнення FB "Lag", інерційну ланку першого порядку FB "PT1" та інтегруючу ланку FB "Integ". Вони базуються відповідно на бібліотечних блоках FB "LSim_Lagging", FB "LSim_PT1" та FB "LSim_I".

```

1
2 IF NOT #statOverfirst THEN
3   FOR #i := 1 TO #MAX BY 1 DO
4     #statDelayStore[#i] := #initstate;
5   END_FOR;
6   #statOverfirst := 1;
7 END_IF;
8
9 // Check Parameters
10 IF #delayCycles > #MAX THEN
11   #delayCycles := INT_TO_UINT(#MAX);
12 ELSE
13   #delayCycles := #delayCycles;
14 END_IF;

```

a)

```

1 // check parameter
2 IF #tmLag1 <= 0 THEN
3   #error := TRUE;
4   #status := W#16#8001; // Parameter Error
5   #statOutput := 0;
6   #output := 0;
7   RETURN;
8 ELSE
9   #error := FALSE;
10  #status := W#16#0000; // no error
11 END_IF;
12 IF #calcParam = True OR NOT #statOverfirst THEN
13   #statOutput := #initstate;
14   #statOverfirst := 1;
15   // 2-parameters For Laplace PT1-System
16   #statRel := - EXP(- #cycle / #tmLag1);
17   #statRf1 := #gain * (1 + #statRel);
18 END_IF;
19 // Differential Equation of PT1-System

```

б)

```

1 // check parameter
2 IF #tmLag1 <= 0 THEN
3   #error := True;
4   #status := W#16#8001; // Parameter Error
5   #statRxn1 := 0.0;
6   #statRyn1 := 0.0;
7   #statOutput := 0;
8   #output := 0;
9   RETURN;
10 ELSE
11   #error := FALSE;
12   #status := W#16#0000; // no error
13 END_IF;
14 // calculates the z-parameter for the difference equation
15 IF #calcParam = True OR NOT #statOverfirst THEN
16   #statOutput := #initstate;
17   #statOverfirst := 1;
18   #statRb1 := #cycle / #tmLag1;
19   #statRal := -1;
20 END_IF;
21 IF #reset = True THEN
22   #statRxn1 := 0.0;
23   #statRyn1 := 0.0;
24   #statOutput := 0;
25 END_IF;
26 // save delayed output-signals

```

в)

Рис. 1 – Фрагменти кодів скоригованих функціональних блоків: а) FB "Lag"; б) FB "PT1"; в) FB "Integ"



У всіх скоригованих функціональних блоках додана вхідна змінна `initstate`, яка визначає початкові умови. У випадку FB “Lag” до оригінального коду FB “LSim_Lagging” додані строки коду з порядковими номерами 2...7. В них спочатку аналізується значення змінної `#statOverfirst` першого звертання до блоку і потім елементи масиву `#statDelayStore`, який забезпечує затримку у часі, заповнюються значенням змінної `#initstate`, що визначає початкові умови на виході з блоку. У FB “PT1” та FB “Integ” зміна оригінального коду полягала в додаванні відповідно строчок 13 та 16.

Аналогічним чином можуть бути скориговані і інші функціональні блоки з бібліотеки “LSim”.

Для перевірки правильності здійснених корекцій коду проведено моделювання FB “Lag”, FB “PT1” та FB “Integ” в середовищі Step 7 TIA Portal, результати якого порівнювались з результатами моделювання аналогічних ланок в середовищі Simulink/MATLAB.

Результати досліджень

На рисунку 2 приведені результати моделювання у середовищі Step 7 TIA Portal об'єктів керування зі статичними властивостями у складі ланки запізнення FB “Lag”, інерційної ланки FB “PT1” та з астатичними властивостями у складі ланки запізнення FB “Lag”, інтегруючої ланки FB “Integ” при подачі на їх вхід постійного сигналу. Реакція об'єкта, отримана з використанням інструменту Trace, при однакових початкових умовах (ПУ) в ланках FB “Lag” і FB “PT1”, відображає лише його вимушений рух (рис. 2а). Коли початкові умови FB “Lag” нульові, а початкові умови FB “PT1” ненульові, то в цьому випадку спостерігається як власний рух об'єкта з ненульових початкових умов, так і його вимушений рух, спричинений вхідним сигналом (рис. 2б).

Коли в складі ОК є ланка запізнення FB “Lag” та інтегруюча ланка FB “Integ”, ненульові початкові умови на FB “Lag” також спричиняють власний рух об'єкта (рис. 2.г).

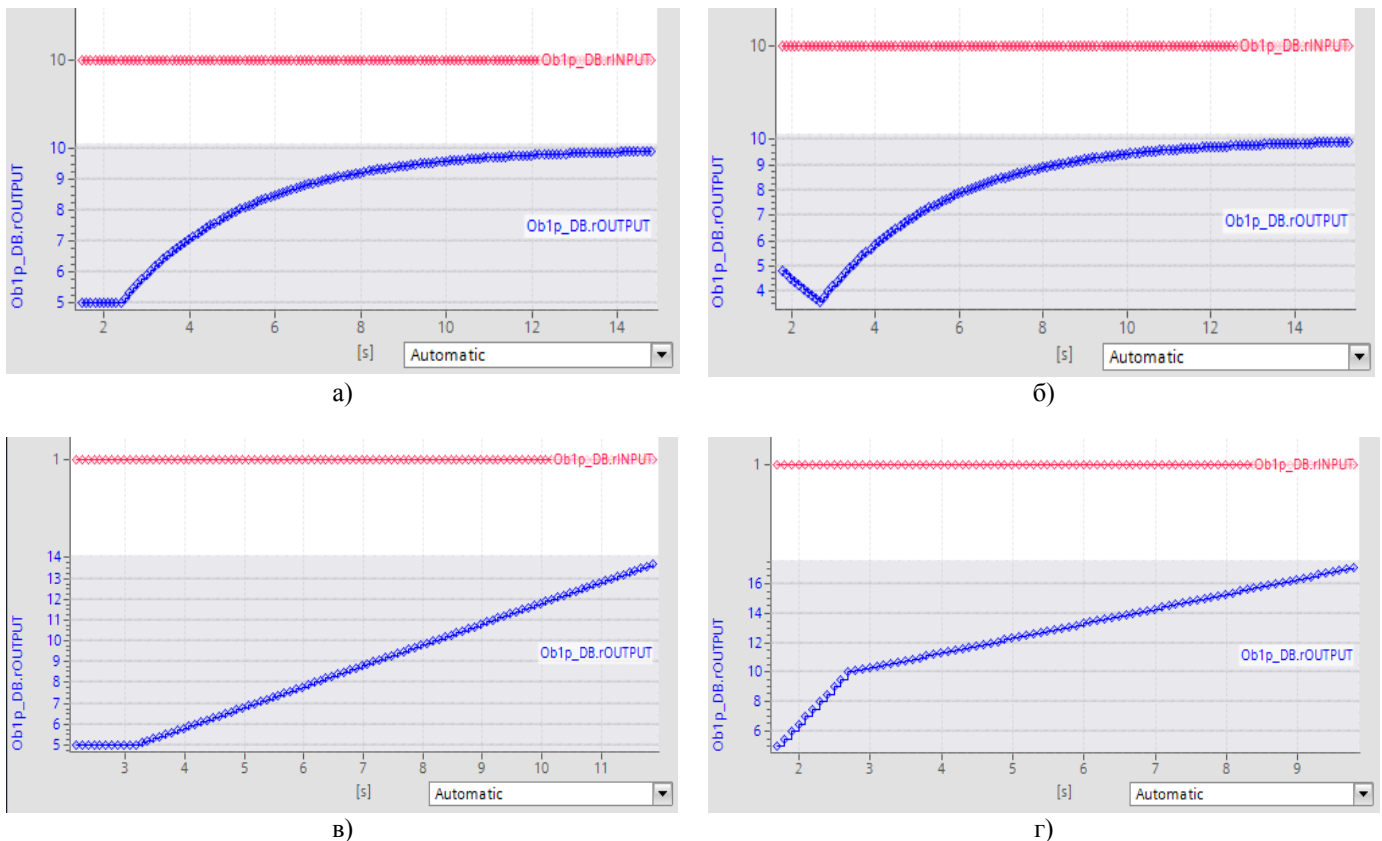


Рис. 2 – Результати моделювання об'єкту керування в середовищі Step 7 TIA Portal:

- а) ПУ FB “Lag” = 5, ПУ FB “PT1” = 5; б) ПУ FB “Lag” = 0, ПУ FB “PT1” = 5;**
в) ПУ FB “Lag” = 0, ПУ FB “Integ” = 5; г) ПУ FB “Lag” = 5, ПУ FB “Integ” = 5;

На рисунку 3 приведені результати моделювання у середовищі Simulink/MATLAB об'єктів керування з аналогічними параметрами та початковими умовами при дії на них постійного сигналу.

Порівняння результатів, наведених на рисунках 2 і 3, переконують в коректному функціонуванні блоків FB “Lag”, FB “PT1”, FB “Integ”, які дозволяють враховувати ненульові початкові умови.

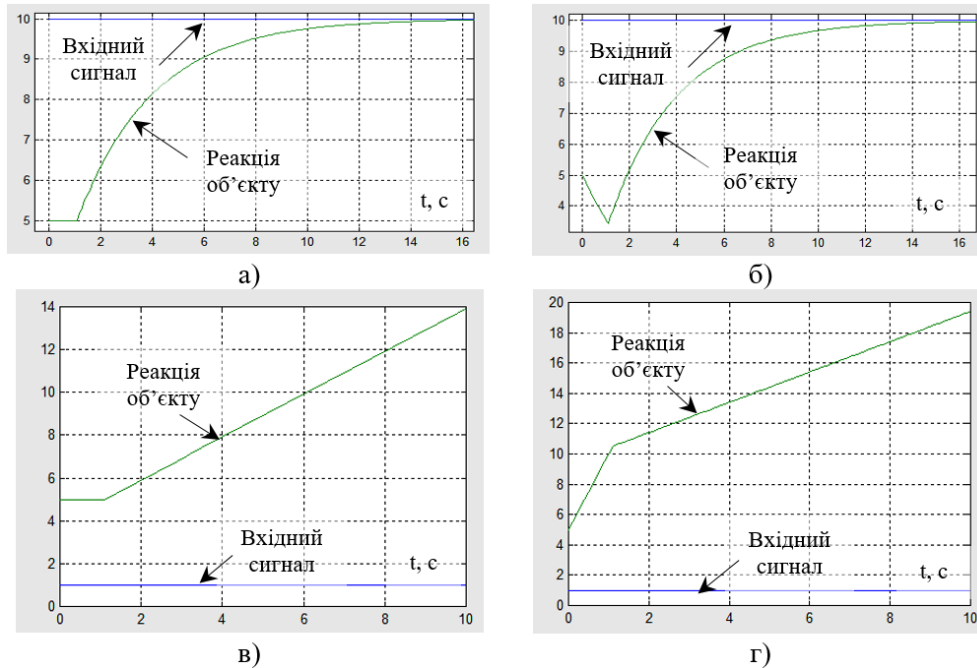


Рис. 3 – Результати моделювання об'єкту керування в середовищі Simulink/MATLAB за однакових умов із моделюванням в середовищі Step 7 TIA Portal

Практичну доцільність розробки таких блоків зручно продемонструвати на прикладі моделювання системи автоматичного регулювання (САР) в складі ПІД-регулятора та ОК з астатичними властивостями.

При її моделюванні в середовищі Simulink/MATLAB (рис. 4) встановлення початкових умов на інтеграторах регулятора та ОК на рівні заданого значення $z=30$ та нульових початкових умов в ланці запізнення дозволяють отримати динамічну рівновагу в САР, яка порушується під дією зовнішнього збурення $f_n(t=5c)=0 \rightarrow 10$.

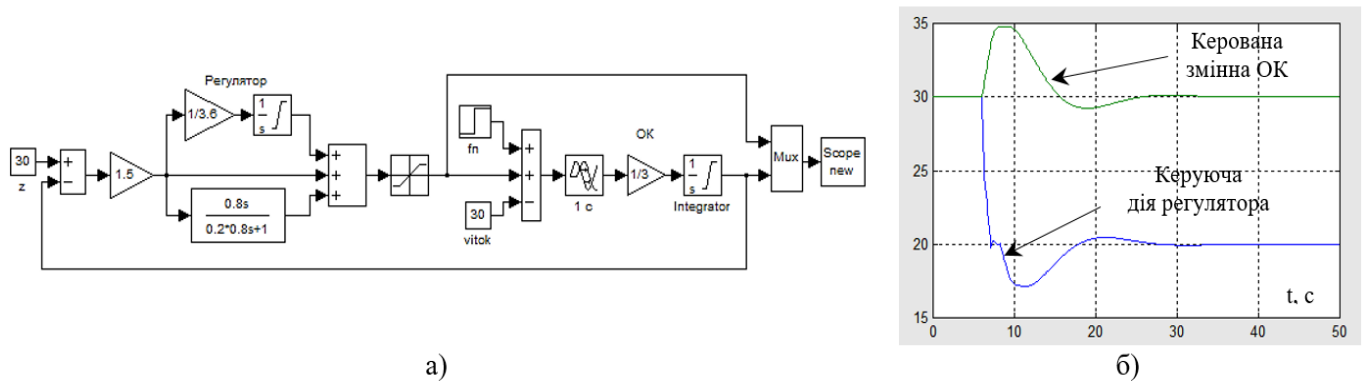


Рис. 4 – Моделювання САР в середовищі Simulink/MATLAB: а) схема; б) результати

Ставиться задача провести моделювання такої САР в середовищі Step 7 вже з використанням функціонального блоку FB PID_Compact, який входить до складу програмного забезпечення TIA Portal і рекомендується фірмою Siemens для застосування на практиці в якості ПІД-регулятора. Відповідна програма представлена на рис. 5.

Загальний блок даних GlobalDB1 служить в проекті для встановлення параметрів регулятора та ОК, а також для організації обміну з HMI-тегами.

Результати моделювання САР в середовищі TIA Portal представлені на рис. 6. Динамічна рівновага в системі забезпечується встановленням початкових умов на інтеграторі ОК (FB "Integ") на рівні заданого значення Setpoint=30. Якщо встановити аналогічне значення змінної ManualValue для FB PID_Compact, то можна досягти практично безударного переходу між ручним та автоматичним режимом роботи регулятора. При першому виклику його функціонального блоку початкові умови на інтеграторі регулятора встановлюються також на рівні заданого значення, що забезпечується значенням статичного тегу IntegralResetMode=4 [3].

Порівнюючи результати моделювання САР в двох середовищах (рис. 4 і 6), можна зробити висновок про практично ідентичну реакцію системи на дію зовнішнього збурення, що додатково переконує в коректності змін коду функціональних блоків бібліотеки "LSim", яка забезпечила можливість встановлення довільних початкових умов.



```

1  "PID_Compact_2"(Setpoint:="GlobalDB1".PID2.setpoint,
2  Input:="GlobalDB1".Obip.output,
3  ManualValue:="GlobalDB1".PID2.manualvalue,
4  ModeActivate:="GlobalDB1".PID2.modeActivate,
5  Output=>"GlobalDB1".PID2.output,
6  State=>"GlobalDB1".PID2.state,
7  Error=>"GlobalDB1".PID2.error,
8  ErrorBits=>"GlobalDB1".PID2.errorBits,
9  Mode:="GlobalDB1".PID2.mode);
10
11 IF NOT "GlobalDB1".PID2.disturbOn THEN
12   "GlobalDB1".Obip.input := "GlobalDB1".PID2.output - #zVitik;
13 END_IF;
14
15 IF "GlobalDB1".PID2.disturbOn THEN
16   "GlobalDB1".Obip.input := "GlobalDB1".PID2.output - #zVitik + #zDist;
17 END_IF;
18
19 "Object1p_DB"(input:="GlobalDB1".Obip.input,
20 gain:="GlobalDB1".Obip.gain,
21 tmLag1:="GlobalDB1".Obip.tmLag1,
22 initstate:="GlobalDB1".Obip.initstate,
23 initstatein:="GlobalDB1".Obip.initstatein,
24 cycle:="GlobalDB1".Obip.cycle,
25 delayCycles:="GlobalDB1".Obip."delayCycles",
26 output=>"GlobalDB1".Obip."output");
27
28 IF "GlobalDB1".PID2.mode="PID_Compact_2".State THEN
29   "GlobalDB1".PID2.modeActivate:=0;
30 END_IF;
31
32 IF #Initial_Call THEN
33   "GlobalDB1".Global.calcParam:=0;
34 END_IF;

```

а)

Object1p			
	Name	Data type	Default value
1	Input		
2	input	Real	0.0
3	gain	Real	0.0
4	tmLag1	Real	0.0
5	initstate	Real	0.0
6	initstatein	Real	0.0
7	cycle	Real	0.1
8	maxOut	Real	100.0
9	minOut	Real	0.0
10	calcParam	Bool	false
11	reset	Bool	false
12	Output		
13	output	Real	0.0
14	InOut		
15	delayCycles	UInt	0
16	Static		
17	zap	"Lag"	
18	apz	"PT1"	
19	inz	"Integ"	
20	Temp		
21	rtemp	Real	

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*)...	REGION
1					#zap(input:=input, initstate:=initstate, delayCycles:=#delayCycles);
2					
3					
4					
5					#apz(input:=#zap.output, tmLag1:=tmLag1, gain:=gain, cycle:=cycle, initstate:=#initstate, output=>#rtemp);
6					
7					
8					
9					
10					
11					
12					#inz(input:=#zap.output, tmLag1:=tmLag1, cycle:=cycle, initstate:=#initstatein, output=>#output);
13					
14					
15					
16					

б)

Рис. 5 – Моделювання в середовищі Step 7 TIA Portal: а) програма CAP; б) програма ОК



Рис. 6 – Результати моделювання CAP в середовищі TIA Portal:
 1 – момент вмикавання ручного режиму керування; 2 – момент переходу в автоматичний режим;
 3 – момент початку компенсації впливу зовнішнього збурення



Обговорення результатів

В настановах [3, 4] спеціалісти фірми Siemens дають рекомендації по визначенню параметрів об'єкта керування зі статичними властивостями за результатами активного експерименту та на їх основі приводять залежності по визначенню параметрів регулятора. Однак там нема вказівок, як обирати налаштування регулятора в системі з ОК, у якого астатичні властивості і в моделі якого є інтегратор. Це змушує звертатись до інших літературних джерел, наприклад [5].

З певною обережністю слід застосовувати функцію самоналаштування, яка вбудована в алгоритм FB PID_Compact.

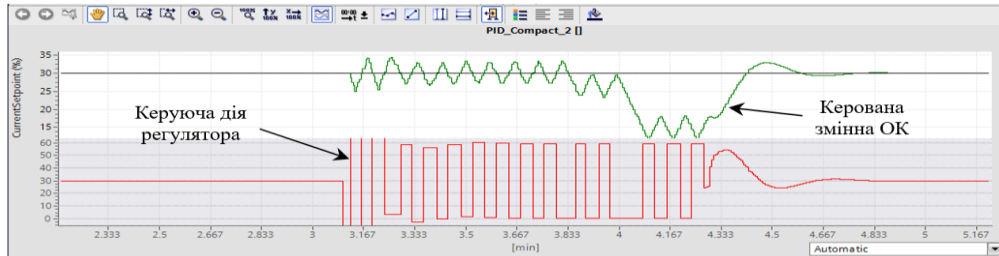


Рис. 7 – Приклад дії функції самоналаштування FB PID_Compact

Як видно з рисунку 7, ця функція передбачає активні пошукові зміни керуючої дії регулятора, що приводить до значних коливань керованої змінної ОК, а це допустимо далеко не для всіх технологічних процесів.

Автори цілком свідомі того, що матеріал статті не містить наукової новизни, але сподіваються, що вона принесе практичну користь спеціалістам, які використовують функціональні блоки ПІД-регулювання для контролерів Simatic S7-1200/S7-1500.

Висновки

Запропоновані зміни коду функціональних блоків бібліотеки “LSim”, які реалізують ланку запізнення, інерційну ланку першого порядку, інтегруючу ланку, дозволяють врахувати довільні початкові умови в цих ланках, що забезпечує моделювання динамічно сталих режимів САР в середовищі Step 7 TIA Portal.

Список використаних джерел

- [1] Library for Controlled System Simulation with STEP7 (TIA Portal). ID: 79047707, V3.0.0. Siemens AG, 03/2017. 46 p.
- [2] Modular PID Control (Модульное ПИД-управление). Руководство пользователя. C79000-G7076-C121-01. Siemens AG, 1997. 258 c.
- [3] S7-1200, S7-1500 PID control Function Manual. A5E35300227-AC. Siemens AG, 12/2017. 551 p.
- [4] Closed-Loop Control with "PID_Compact" V2.2. Entry-ID: 79047707. Siemens AG, 03/2017. 43 p.
- [5] Наладка средств автоматизации и автоматических систем регулирования. Справочное пособие. Под ред. А.С. Ключева. М.: Энергоатомиздат, 1989. 368 c.

References

- [1] Library for Controlled System Simulation with STEP7 (TIA Portal). ID: 79047707, V3.0.0. Siemens AG, 03/2017. 46 p.
- [2] Modular PID Control. Rukovodstvo polzovatelya. C79000-G7076-C121-01. Siemens AG, 1997. 258 p.
- [3] S7-1200, S7-1500 PID control Function Manual. A5E35300227-AC. Siemens AG, 12/2017. 551 p.
- [4] Closed-Loop Control with "PID_Compact" V2.2. Entry-ID: 79047707. Siemens AG, 03/2017. 43 p.
- [5] “Naladka sredstv avtomatizatsii i avtomaticheskikh sistem regulirovaniya. Spravochnoe posobie,”. A.S. Klyueva Ed. Moscow, Energoatomizdat, 1989. 368 p.

УДК 004.4: 004.075

АНАЛІЗ ОСНОВНИХ ПЕРЕВАГ ТА НЕДОЛІКІВ РІЗНОМАНІТНИХ ФРЕЙМВОРКІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ

Антонова А.¹, Помогасв Д.²

Одеська національна академія харчових технологій, м. Одеса, Україна

ORCID: ¹ <http://orcid.org/0000-0003-3599-0983>, ² <http://orcid.org/0000-0002-6212-4848>

E-mail: ¹ allaantonova62@gmail.com, ² dimonpom@gmail.com

Copyright © 2020 by author and the journal “Automation of technological and business-processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i4.1935>



Анотація. У статті розглядаються програмні засоби для автоматизації тестування програмних продуктів. Авторами проведено аналіз впливу фреймворків автоматизації тестування на сферу розробки програмного забезпечення, також проаналізовано низку задач тестувальників, які можна вирішити за допомогою цих засобів. Сфера ІТ проникає в усі аспекти життя людини уже давно, але тільки в останні декілька років спостерігається небувала тенденція стрімкого зростання кількості програм, сайтів, спеціалізованих програмних продуктів тощо. Цей стрімкий ріст обумовлений насамперед необхідністю спрощувати виконання рутинних трудомістких задач у багатьох сферах життєдіяльності, що донедавна виконували люди. Вже зараз різноманітне програмне забезпечення застосовується майже у всіх сферах життя і згодом повністю охопить усі не творчі напрями діяльності людини. Широке застосування комп'ютерних технологій призвело не тільки до збільшення чисельності програмного забезпечення, але й до стрімкого зростання вимог до цих програмних продуктів. Від їх коректного виконання, часом, залежить не тільки користувач, але й цілі галузі комерційного і державного правління. Тестування - один з найважливіших етапів контролю якості в процесі розробки програмного забезпечення. Автоматизоване тестування є його складовою частиною. Воно використовує програмні засоби для створення, виконання тестів і перевірки результатів їх виконання. Автоматизація тестування допомагає оптимізувати час на проведення тестування: надає тестувальникам можливість швидко і якісно проводити тести, мати можливість легко їх міняти, що спрощує сам процес тестування. Авторами проведено дослідження і порівняльний аналіз традиційної методології тестування та Agile методології та зроблені висновки о їх перевагах і недоліках. Основна мета цього дослідження – це проаналізувати процеси автоматизації тестування програмних продуктів, визначити основні переваги та недоліки різноманітних фреймворків автоматизації тестування, зробити висновки щодо доречності використання того чи іншого рішення та засобу.

Abstract. The article discusses software tools for automating testing of software products. The authors analyzed the impact of test automation frameworks on the field of software development, and also analyzed a number of tasks for testers that can be solved using these tools. The IT sphere has been penetrating all aspects of human life for a long time, but only in the last few years has there been an unprecedented trend of a rapid growth in the number of programs, websites, specialized software products, and the like. This rapid growth is primarily due to the need to simplify the performance of routine labor-intensive tasks in many spheres of life, until recently people performed. Already now, a variety of software is used in almost all spheres of life and subsequently will fully cover all non-creative areas of human activity. The widespread use of computer technology has led not only to an increase in the number of software, but also to a rapid growth in requirements for these software products. Their correct implementation, time, depends not only on the user, but also on entire branches of commercial and state government. Testing is one of the most important stages of quality control in the software development process. Automated testing is an integral part of it. It uses software to create, run tests, and check the results of their execution. Test automation helps to optimize testing time: it provides testers with the ability to quickly and efficiently conduct tests, to be able to easily change them, which simplifies the testing process itself. The authors conducted a study and comparative analysis of the traditional testing methodology and Agile methodology and made conclusions about their advantages and disadvantages. The main goal of this study is to analyze the processes of automation of testing software products, to determine the main advantages and disadvantages of various test automation frameworks, to draw conclusions regarding the appropriateness of using one or another solution and tool.

Ключові слова: автоматизація тестування, фреймворк, тестування програмного забезпечення

Keywords: test automation, framework, quality assurance

Вступ

Стрімке розповсюдження інформаційних технологій у всі сфери життєдіяльності людини спровокувало й потребу в збільшенні кількості спеціалізованих програм, що в тому чи іншому вигляді допомагали людині. Такий ріст обумовлений насамперед необхідністю спрощувати виконання рутинних трудомістких задач у багатьох сферах життєдіяльності, що донедавна виконували люди. Активне використання переваг інформаційних технологій не тільки дасть можливість людям займатися більш творчими та цікавими задачами, але й мінімізує кількість помилок та неточностей, що спричиняв людський фактор. Вже зараз різноманітне програмне забезпечення застосовується майже у всіх сферах життя і згодом повністю охопить усі не творчі напрями діяльності людини.

Аналіз етапів розробки програмного продукту

Будь-яке впровадження спеціалізованого програмного забезпечення має на увазі той факт, що перед безпосередньо використанням програмного продукту був етап його створення.

Говард Бет'єр (молодший) описує процес створення програмного забезпечення наступним чином:

Процес розробки - це процес діалогу, в рамках якого знання, які необхідно перетворити на програмне забезпечення, набуваються та реалізуються. Це ітераційний процес, в якому розробляємий продукт сам діє як засіб спілкування, завдяки якому учасники можуть отримати більше корисних знань [1].



Після майже півстоліття розробки програмного забезпечення досвід минулих проектів узагальнюється в найкращі практики з виробництва високоякісного програмного забезпечення. Встановлено, що розробка програмного забезпечення зазвичай включає такі етапи:

- аналіз попиту та вимог;
- проектування системи;
- програмування;
- тестування і налагодження;
- експлуатація, супровід і підтримка.

Згадуючи про процес розробки програмного забезпечення, неможливо ігнорувати роль тестування в ньому. Тестування програмного забезпечення можна трактувати як процес перетворення латентних дефектів на ідентифіковані. Цей вирішальний етап життєвого циклу розробки програмного забезпечення виявляє приховані дефекти розроблюваного продукту.

Дуже цікава методологія Agile Testing (Agile тестування) — це тестування програмного забезпечення із дотриманням принципів гнучкої розробки програмного забезпечення. Тестування проводиться паралельно з розробкою. Команда тестувальників працює постійно у ритмі зміни вимог, отримує часті релізи і слідкує за змінами коду команди розробників під час тестування програми.

Слід розуміти, що у Agile методології тестування не є окремим етапом розробки, як у традиційних моделях розробки програмних засобів. Часовий відрізок підготовки до тестування суттєво менший у Agile порівняно з традиційними моделями. Команда тестування під час Agile починає працювати з початкових етапів опрацювання вимог і малих частинок функціональностей, тоді як в традиційних моделях команда тестування працює на повну програму вже після розробки якогось завершеного модуля або цілої програми.

Роль тестування в процесі розробки програмного забезпечення

Абсолютно всі проекти потребують тестування, так як без тестів неможливо визначити наявність в продукті критичних помилок, його відповідність вимогам, рівень забезпечення безпеки і т. д. Тестування програмних продуктів використовує приблизно 40-50 відсотків усіх ресурсів і становить 50-60 відсотків від загальної вартості проекту [2]. Етап тестування, який вважається одним з основних випробувань на шляху розробки програмного забезпечення, можна розглядати як гарну можливість поліпшити якість, оптимізувати витрати і скоротити час виходу продукту на ринок програмного забезпечення. Така оптимізація є дуже бажаною у сучасних реаліях ринку, коли галузь програмного забезпечення стикається з жорсткою конкуренцією, та компанії намагаються водночас скоротити бюджет та строки проектів [3].

Як і раніше, ручне тестування є важливою частиною процесу перевірки програмного продукту на різних стадіях розробки. Під час виконання такого тестування людина - тестувальник користується розроблюваною програмою та проводить над нею різноманітні тести, які інтерпретують та аналізують поведінку продукту за різних входних умов [4]. Пізніше тестувальник готує звіти та надає коментарі щодо стану якості продукту, порівнюючи фактичні результати з очікуваними.

Загалом ручне тестування – це монотонний і трудомісткий процес, що вимагає великих затрат часу. Крім цього, в процесі ручного тестування наявний «людський фактор», отже частина помилок може залишитися невиявленою, а деякі результати перевірки можуть виявитися суб'єктивними [5].

З іншого боку, автоматичне тестування зводиться до використання деяких стандартних програмних рішень для контролю виконання тестів на тестовому програмному забезпеченні. Цей процес також передбачає встановлення передумов, узгодження фактичних результатів із прогнозованими, а також подальше документування спостережень згідно зі стандартизованими протоколами. Процеси, що виконує людина, автоматизуються і виконуються програмою або технікою. Такий підхід допомагає у досягненні важливої мети у розробці програмного забезпечення – збільшення продуктивності, зменшення часу виконання задачі і підвищення якості продукту.

Переваги автоматизації тестування

До основних переваг автоматизації тестування можна віднести:

- підвищення швидкості тестування без втрати якості;
- підвищення якості тестування (при використанні автоматизованого тестування, «людський фактор» не впливає на якість тестування);
- можливість виконання видів тестування, що не можуть бути виконанні ручним способом, або потребують значних витрат;
- повторне використання;
- автоматичне збереження звітів про результати роботи програм.

Також автоматизація має певні недоліки:

- написані тести завжди працюють однаково, що в деяких випадках може призвести до пропуску помилки, яку б помітила людина при ручному тестуванні;
- пропуск дрібних недоліків (тести можуть пропускати дрібні помилки, на які не були запрограмовані);
- витрати на розробку і підтримку (залежать від складності й частоти оновлень продукту).



Проте незважаючи на перераховані мінуси, при правильному підході та володінні необхідними ресурсами, можливо досягти результатів, близьких до ідеалу. Загалом автоматизація тестування, завдяки підвищенню швидкості процесу тестування, є найбільш ефективним рішенням в умовах жорстких строків готовності продукту.

Як результат, сьогодні існує багато комерційних програмних засобів, що дозволяють впроваджувати автоматизацію тестування. Такі рішення широко доступні, і чисельні організації займаються наданням послуг із забезпечення якості тестування. Але слід мати на увазі, що автоматизація тестування - це процес, ціна помилки в якому дуже велика. Перед його початком слід врахувати безліч факторів, наприклад, яка функція вимагає автоматизованого тестування, а для якої створення автотестів не принесе значних переваг [5]. Автоматизація тестування вимагає великих первинних вкладень з точки зору аналізу функцій програмного забезпечення, створення сценаріїв, закупівлі інструментів, навчання тощо але при правильному підході окупає себе.

Сучасні фреймворки автоматизації тестування

Selenium - це один з найпопулярніших фреймворків для автоматизації тестування, який складається з великої кількості інструментів і плагінів. Загалом *Selenium* - це великий проект, в рамках якого розробляються серія програмних продуктів з відкритим кодом, серед яких слід зазначити:

- *Selenium WebDriver* - програмна бібліотека, що дозволяє управляти браузером;
- *Selenium Grid* - це кластер, що складається з декількох *Selenium*-серверів. Він призначений для організації розподіленої мережі, що дозволяє паралельно запускати багато браузерів на великій кількості машин;
- *Selenium IDE* - це плагін до браузера Firefox, який може записувати дії користувача, відтворювати їх, а також генерувати код для *WebDriver*, в якому виконуються ті ж самі дії;
- *Selenium-рекордер*.

Katalon Studio - це ефективний інструмент автоматизації процесу тестування сайтів, веб - сервісів, мобільних додатків. *Katalon Studio* вважають «нащадком» таких фреймворків, як *Appium* і *Selenium*. Це пов'язано з тим, що він перейняв у них ряд плюсів, пов'язаних з інтегрованою автоматизацією тестування програмного забезпечення. Для комфортної взаємодії з цим інструментом потрібно володіти як початковими знаннями в тестуванні, так і бути експертом своєї справи. На практиці запуск власного проекту по автоматизації тестування не викликає труднощів навіть у людей, далеких від програмування. Наприклад, це можна зробити за допомогою функції *Object Spy*. Що стосується програмістів і більш досвідчених тестувальників, то для них *Katalon Studio* стане дуже корисним інструментом в плані економії часу як при написанні нових бібліотек, так і для підтримки вже існуючих скриптів.

Unified Functional Testing (UFT), перекладається, як комплексне функціональне рішення для тестування) - це популярний комерційний інструмент для автоматизації тестування. По суті, *UFT* - це набір функцій, призначених для тестування веб - сервісів, сайтів, API, графічного інтерфейсу мобільних, десктопних і веб - додатків практично на всіх платформах, що існують на сьогоднішній день. Інструмент має розширений функціонал розпізнавання об'єктів на основі їх зображень. Крім того, доступні багаторазові тестові компоненти, а також документація по автоматизації. Для роботи інструменту використовується *Visual Basic Scripting Edition*, завдяки чому є можливість управляти об'єктами або зберігати інформацію про виконане тестування. Крім того, *UFT* інтегрований з *Mercury Quality Center* і *Mercury Business Process Testing*, також він підтримує CI за допомогою інтеграції з такими CI-інструментами, як *Jenkins*.

Test Complete є ефективним інструментом для виконання тестування мобільних, десктопних і веб - додатків. Він розроблений компанією *SmartBear* і підтримує *VBScript*, *JavaScript*, *Python*, *C++ Script*. Так само як і в випадку з *Katalon Studio*, за допомогою *Test Complete* тестувальники зможуть без проблем проводити керування даними тестування, а також тестування із застосуванням ключових слів. Додатково в інструменті є зручна функція запису процесу тестування з можливістю подальшого відтворення. Якщо порівнювати *Test Complete* з *UFT*, то вони схожі принципами розпізнавання об'єктів GUI. Це дозволяє уникнути додаткових турбот по підтримці тестових скриптів при змінах AUT. Ще інструмент може інтегруватися з *Jenkins* під час CI-процесу.

Висновки

Говорячи про програмні засоби автоматизації тестування, процес вибору фреймворку, що буде являти собою основу для впровадження автотестів, є дуже важливим підготовчим етапом, що може вплинути на успішність процесів тестування на проекті. Зараз існує велика кількість різноманітних інструментів, що дозволяють автоматизувати тестування. У ході аналізу та дослідження ринку програмного забезпечення для автоматизованого тестування було обрано чотири фреймворки: *Selenium*, *Katalon Studio*, *Unified Functional Testing*, *Test Complete*.

Після ретельного, всебічного аналізу систем контролю та створення автоматичного тестування було прийнято рішення в новому проекті використовувати фреймворк *Selenium* в якості програмного забезпечення для автоматизованого тестування, так як недоліки цього фреймворку не є такими, що принципово заважають його використуванню, а переваги, такі як величезна спільнота й відсутність обмежень для інтеграцій, є суттєвими.

Список використаних джерел

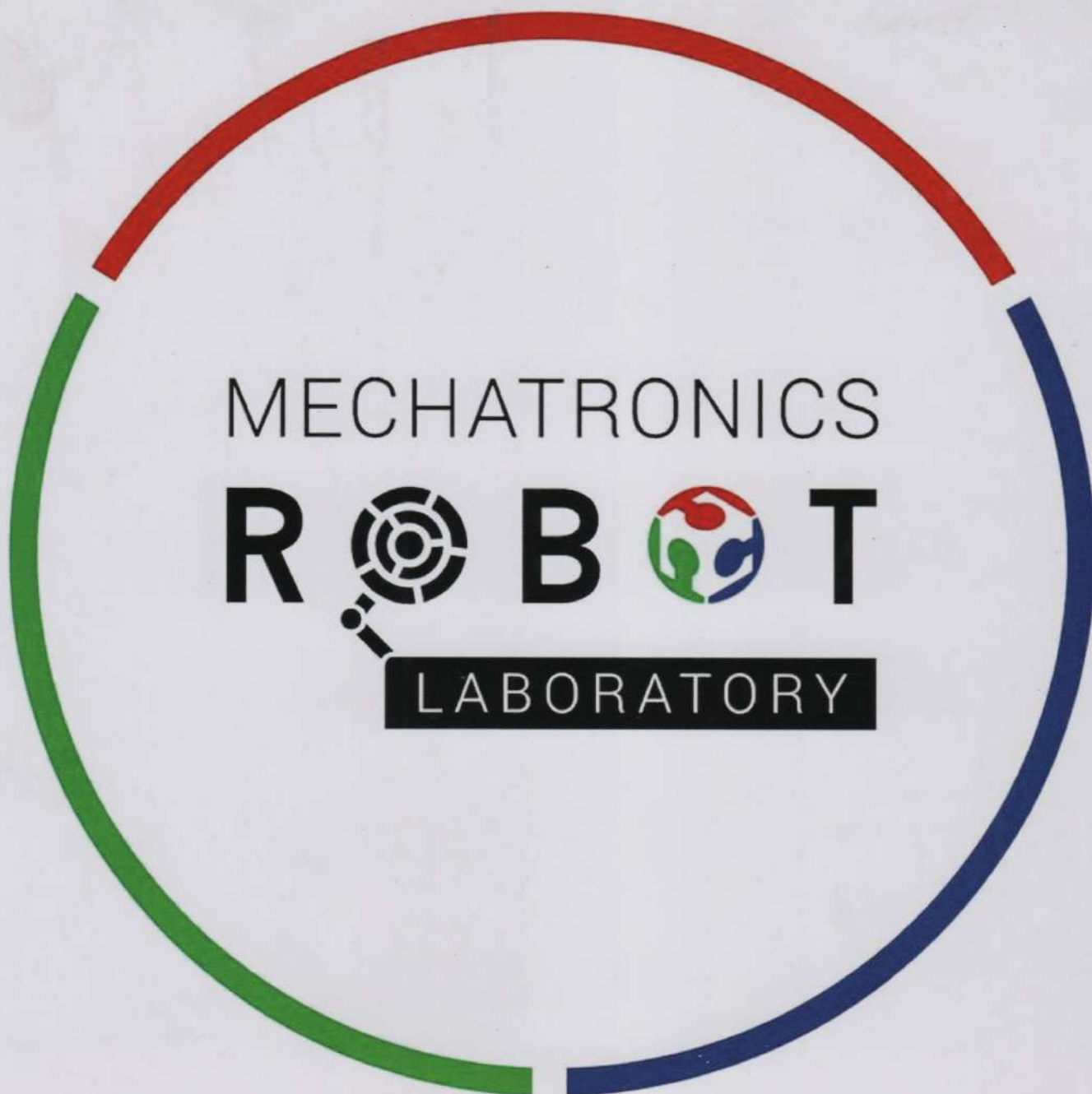
- [1] Liangliang H. Development Trend of Software Process Model / H. Liangliang, Z. Jinsheng, M. Xiaoyu. // IOP Conference Series: Materials Science and Engineering. - 2019. - №569.
- [2] Divya Kumar. The Impacts of Test Automation on Software's Cost, Quality and Time to Market / Divya Kumar, Krishn Kumar Mishra. // Procedia Computer Science. - 2016. - №79. - С. 8-15.



- [3] Rudolf Ramler. Economic perspectives in test automation: balancing automated and manual testing with opportunity cost / Rudolf Ramler, Klaus Wolfmaier. // Proceedings of the 2006 International Workshop on Automation of Software Test. – 2006. – С. 85–91.
- [4] Glenford J. Myers. The Art of Software Testing / Glenford J. Myers, Corey Sandler, Tom Badgett., 2011. – (John Wiley & Sons).
- [5] A Antonova, B Shanovskiy, Research and analysis of application of automated testing in web applications.// Automation of technological and business processes, 10 (1), 2018 – С. 55-57.
- [6] E M Abdul Rauf. Software Test Automation: An Algorithm for Solving System Management Automation Problems / E M Abdul Rauf, E Madhusudhana Reddy. // Procedia Computer Science. – 2015. – №46. – С. 949–956.
- [7] Norbert Oster. Automatic Test Data Generation by Multi-objective Optimisation / Norbert Oster, Francesca Saglietti. // SAFECOMP. – 2006. – №4166. – С. 426–438.
- [8] Jalote Pankaj. An Integrated Approach to Software Engineering / Jalote Pankaj., 2012. – (Springer Science & Business Media).
- [9] Alan Page. How We Test Software at Microsoft / Alan Page, Ken Johnston, Bj Rollison., 2008. – 364 с. – (Microsoft Press).
- [10] Antonia Bertolino. Software Testing Research: Achievements, Challenges, Dreams / Antonia Bertolino. // Future of Software Engineering. – 2007. – С. 85–103.
- [11] Rex Black. Critical Testing Processes: Plan, Prepare, Perform, Perfect / Rex Black., 2003. – 608 с. – (Addison-Wesley Professional).

References

- [1] H. Liangliang, Z. Jinsheng, M. Xiaoyu, “Development Trend of Software Process Model,” *IOP Conference Series: Materials Science and Engineering*, no. 569, 2019.
- [2] Divya Kumar, Krishn Kumar Mishra, “The Impacts of Test Automation on Software's Cost, Quality and Time to Market,” *Procedia Computer Science*, no.79, pp C. 8–15, 2016.
- [3] Rudolf Ramler, Klaus Wolfmaier, “Economic perspectives in test automation: balancing automated and manual testing with opportunity cost,” *Proc. of the 2006 International Workshop on Automation of Software Test*, 2006, pp. 85–91.
- [4] Glenford J. Myers, Corey Sandler, Tom Badgett, “The Art of Software Testing,” John Wiley & Sons, 2011.
- [5] A. Antonova, B. Shanovskiy, “Research and analysis of application of automated testing in web applications,” *Automation of technological and business processes*, 10 (1), pp. 55-57, 2018.
- [6] E. M. Abdul Rauf, E. Madhusudhana Reddy, “Software Test Automation: An Algorithm for Solving System Management Automation Problems,” *Procedia Computer Science*, no. 46, pp. 949–956, 2015.
- [7] Norbert Oster, Francesca Saglietti, “Automatic Test Data Generation by Multi-objective Optimisation,” *SAFECOMP*,. No. 4166, pp. 426–438, 2006.
- [8] Jalote Pankaj, “An Integrated Approach to Software Engineering,” Springer Science & Business Media, 2012.
- [9] Alan Page, Ken Johnston, Bj Rollison, “How We Test Software at Microsoft,” Microsoft Press, 2008, 364 p.
- [10] Antonia Bertolino, “Software Testing Research: Achievements, Challenges, Dreams,” *Future of Software Engineering*, pp. 85–103, 2007.
- [11] Rex Black, “Critical Testing Processes: Plan, Prepare, Perform, Perfect,” Addison-Wesley Professional, 2003, 608 p.



Laboratory of Mechatronics and Robotics in Odessa National Academy of Food Technologies was established in 2014. It is the first co-work space with the widely developed material base. The main aim of Laboratory is to popularize technical education in general and robotics in particular. In this regard Laboratory provides open access for everyone (conditions for access – 20 % of the time should be voluntarily devoted to social events that are valuable for Laboratory). Laboratory of Mechatronics and Robotics consists of eight auditoriums: workroom for prototyping, specialized auditorium of industrial pneumatic robotics, specialized auditorium of industrial electrical robotics, specialized auditorium of portable robotics, specialized auditorium of industrial hydraulic robotics, recreation zone for relaxation, a head office and a small storeroom. All the auditoriums have all the necessary stands and equipment for students' education process, for work on the projects and conducting of research.

Kanatnaya St, 112, Odessa, 65039, Ukraine
+380487124254

mechatronics.robotlab@gmail.com
<http://www.robot.onaft.edu.ua/>

