

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



Наукові праці
Донецького національного технічного університету
Серія: «Проблеми моделювання
та автоматизації проектування»

№ 1 (14), 2019

Покровськ – 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАУКОВІ ПРАЦІ

**ДОНЕЦЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО
УНІВЕРСИТЕТУ**

**Серія «Проблеми моделювання та автоматизації
проектування»**

**Випуск присвячується 45-річчю співробітництва ДонНТУ і
Штутгартського університету в галузях моделювання, САПР,
автоматизації динамічних систем, паралельних і
розподілених обчислень.**

Всеукраїнський науковий збірник

Заснований у червні 1999 року

Виходить 2 рази на рік

№ 1 (14), 2019

Покровськ – 2019

Публікується згідно з рішенням Вченої ради державного вищого навчального закладу «Донецький національний технічний університет», протокол № 10 від 19.12.2019 р.

У збірнику опубліковано статті співробітників факультету комп'ютерних наук та технологій ДонНТУ, а також деяких інших університетів, які є науковими партнерами ДонНТУ і активно співпрацюють у спільних дослідженнях. В публікаціях наведено результати наукових досліджень та розробок, виконаних у 2019 році за наступними напрямками: теоретичні аспекти моделювання та автоматизації проектування, методи та засоби моделювання, методи та засоби автоматизації проектування, спеціалізовані обчислювальні системи моделювання, контроль процесів та керування динамічними системами. Матеріали збірки призначені для викладачів, наукових співробітників, інженерно-технічних працівників, аспірантів та студентів, що займаються питаннями дослідження, моделювання та розробки складних динамічних систем.

Засновник та видавець - Донецький національний технічний університет

РЕДАКЦІЙНА КОЛЕГІЯ

Професор, д. т. н., В.А. Святний (головний редактор, відп. за випуск), доцент, к. т. н., С.О. Цололо (заст. головного редактора), доцент, к. т. н., В.В. Шамаєв (заст. головного редактора), Prof. Dr. of Sc., A. Barkalov, Prof. Dr.-Ing. habil. Dr. h.c., S. Heinrich, Prof. Dr.- Ing. habil., A. Kienle, Prof. Dr.- Ing. habil., W. Ehlers, Prof. Dr. rer. nat. Dr. h. c., K. Rothermel, доцент, к. т. н., С.О. Ковальов, професор, д. т. н., О.А. Дмитрієва, професор, д. т. н., Н.М. Куцуль, доцент, к. т. н., Ю.Л. Дікова, ст. викладач, О.Г. Шевченко, аспірант, К.С. Гайдук.

Збірник зареєстрований в Державному комітеті інформаційної політики, телебачення та радіомовлення України. Свідоцтво КВ № 7377 від 03.06.2003.

Збірник «Наукові праці Донецького національного технічного університету» заснований в червні 1999 р. Випуск 211.

ЗМІСТ

Святний В.А. 45 років наукового співробітництва Донецького національного технічного університету з Штутгартським університетом (Німеччина)	5
Тягунова М.Ю., Ільїна Н.О., Голуб Т.В. Інтелектуальна система підтримки прийняття рішення щодо вибору лікарського препарату	9
Федько А.О., Зеленьова І.Я., Тягунова М.Ю. Розробка та моделювання системи «розумний будинок» з підвищеною надійністю на базі Arduino	16
Smolij V. About features of management preproduction of electronic vehicles	26
Гайдук К.С., Шевченко О.Г. Управління пам'яттю і структури даних в операційних системах FreeRTOS та Predicate OS	36
Мірошкін О.М., Маргієв Г.Е. Паралельний вирішувач диференціальних рівнянь на базі блокових обчислювальних методів	49

45 РОКІВ НАУКОВОГО СПІВРОБІТНИЦТВА ДОНЕЦЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ З ШТУТТГАРТСЬКИМ УНІВЕРСИТЕТОМ (НІМЕЧЧИНА)

На початку 70-х років минулого століття відбувся спад напруженості у відносинах тодішнього СРСР з капіталістичним світом, відомий в історії як період розрядки. В Європі розрядка ознаменувалась політичним діалогом з Федеративною Республікою Німеччиною, в результаті якого активізувались економічні відносини та було започатковано міждержавні програми в галузі науки, освіти та культури. Німецьке наукове товариство (DFG-Deutsche Forschungsgemeinschaft) та союзне Міністерство вищої освіти заключили угоду про взаємне наукове стажування викладачів вузів. Одною з найефективніших форм цієї співпраці виявилось 10-місячне стажування кандидатів наук віком до 36 років. Саме з такого стажування тодішнього декана факультету обчислювальної техніки і автоматизованих систем, доцента кафедри ЕОМ (тепер – КІ) Святного В.А. розпочалось наукове співробітництво ДонНТУ з Штутгартським університетом (ШУ). Директор інституту системної динаміки та автоматичного керування (ISR-Institut für Systemdynamik und Regelungstechnik) професор Ернст-Дітер Гіллес, будучи членом наукової ради DFG, дав згоду на прийом Святного В.А. за темою досліджень, що зосереджувались на розробленні й використанні аналого-цифрових (гібридних) моделей багатозв'язних систем автоматизованого керування мережними динамічними об'єктами з зосередженими та розподіленими параметрами. Для реалізації таких моделей в ISR була інстальована новітня на той час американська гібридна обчислювальна система PACER 600. Датою початку наукового співробітництва з ISR є 8 листопада 1973 р., коли відбулась перша бесіда В.А. Святного з Е.-Д. Гіллесом, на якій було погоджено робочий план стажування, визначено режим доступу до системи PACER 600 та коло співробітників ISR, що займались аналогічною тематикою – тодішні асистенти Міхаель Цайц, Райнер Бук, Роберт Кайнер, Клаус Штарк. План стажування було повністю виконано, результати розробок та

досліджень аналого-цифрових моделей використано автором в докторській дисертації. Набутий досвід став основою для відкриття галузевої науково-дослідної лабораторії «Структури та програмне забезпечення аналого-цифрових обчислювальних систем», що діяла на факультеті ОТІ до 1991 року. Співпраця з Штутгартськими колегами продовжувалась через листування, в 1975 році було завершено відлагодження цифрової моделі мережного динамічного об'єкта на системах CDC 6600 і CYBER. Наукове відрядження В.А. Святного за програмою DAAD в 1981 та 1987/88 р.р. слугували закріпленню контактів та розширенню тематики досліджень. Знайомство з деканом-засновником факультету інформатики та інституту паралельних і розподілених систем (IPVS) проф. Андреасом Ройтером поклало початок співробітництва в галузі побудови паралельних моделюючих середовищ та інформаційних технологій. В березні 1989 р. А. Ройтер відвідав тодішній ДПП. Результатом цього візиту був план переходу від аналого-цифрових до паралельних апаратно-програмних засобів моделювання складних динамічних систем. В 1990 р. міжурядова програма дозволила укладання прямих угод між вузами Союзу та ФРН. Символічним стало підписання трьох угод між факультетом ОТІ та інститутами ISR (Е.-Д. Гіллес), IPVS (А. Ройтер), IAS (Р. Лаубер) в дні серпневого путчу 1991 року. А. Ройтер запросив проф. Святного В.А., декана факультету ОТІ, на 6-місячну наукову роботу як гостя-професора. Саме в цей період, з 15.12.91 по 15.06.92, разом з професорами А. Ройтером, Е.-Д. Гіллесом, М. Цайтцем, Р. Лаубером, докт.-інж. Т. Бройнлем було сформульовано новий напрям наукового співробітництва – паралельне моделювання складних динамічних систем, модельна підтримка проектування систем автоматизації технологічних процесів, розробка паралельних алгоритмів чисельного рішення великорозмірних систем диференціально-алгебраїчних рівнянь. В 1992, 1993 та в 1995

роках за цією тематикою як гості-професори відвідали IPVS В.В. Лапко і Л.П. Фельдман. Треба підкреслити, що підтримка наших німецьких колег в переломні 1992-1995 роки, надання можливості провідним науковцям факультету виконувати дослідження на новітніх апаратно-програмних засобах мали вирішальне значення для збереження наукового потенціалу кафедр ЕОМ та ПМІ, для інтеграції ДонНТУ в європейський науково-освітній простір.

Однією з центральних тем співробітництва є розробка розподіленого паралельного моделюючого середовища (РПМС). Концепція середовища сформульована В.А. Святним, М. Цайтцем, Т. Бройнлем та А. Ройтером в 1992 р. в процесі розробок та реалізації SIMD-моделей мережних динамічних об'єктів (SIMD – Single Instruction Multiple Data, паралельні обчислювальні системи з керуючим процесором та полем процесорних елементів, що одночасно виконують операції над масивами даних розміром до 64*1024 чисел). На IPVS-семінарі 10.06.1992 В.А. Святним було зроблено доповідь, в якій презентувались SIMD-моделі та концепція масово паралельного моделюючого середовища. Того ж дня в газеті «Штутгартер цайтунг» опубліковано статтю «Співробітництво з Україною», в якій кореспондент виклав інтерв'ю з професорами А. Ройтером та В.А. Святним і підкреслив, що Штутгартський університет в особі IPVS надає можливість Донецькому технічному університету в особі факультету ОТІ (нині КНТ) мати теледоступ до високопродуктивної паралельної обчислювальної техніки. В умовах, коли вже не стало галузевої лабораторії, коли повністю було відсутнім фінансування наукових розробок, а в Штутгарті тільки починали впровадження Internet-технологій, організація цього міжнародного проекту мала світову актуальність за науковим змістом і відкривала хороші перспективи для молодих науковців факультету.

З появою в обчислювальному центрі ШУ (HLRS-Höchstleistungsrechenzentrum Stuttgart) високопродуктивних MIMD-систем (1993) постало питання побудови MIMD-компоненти РПМС (MIMD – Multiple Instruction Multiple Data, паралельні обчислювальні системи з повнофункціональними процесорами, кожен з яких виконує свою програму-процес, що є частиною

загального алгоритму вирішення складної задачі). В 1998 р. інститут IPVS (проф. К. Ротермель) передав кафедрі ЕОМ 12-вузлову систему Intel Paragon, через IPVS-сервер було з 1997 р. налагоджено доступ до всіх паралельних ресурсів HLRS.

В 1995 р. вчена рада нашого університету обрала почесними докторами професорів Е.-Д. Гілlesa, А. Ройтера, Р. Лаубера як вчених зі світовим ім'ям, що внесли вирішальний внесок в розбудову наукового співробітництва Штутгартського і Донецького технічного університетів.

19.01.2000 р. за поданням деканів чотирьох факультетів (інформатики, електротехніки, технологій та інженерно-геодезичного) сенат ШУ ухвалив рішення про підписання рамкової угоди про наукове співробітництво з Донецьким національним технічним університетом. Угоду підписали ректори, професори О.А. Мінаєв і Д. Фріч. Координаторами співробітництва було призначено професорів М. Цайтца і В.А. Святного, які з 1973 року займалися цією справою. 31 травня 2011 р. під час святкування 90-річчя ДонНТУ відбулось підписання оновленої угоди ректорами О.А. Мінаєвим і В. Ресселем.

В 2000 р. вчена рада ДонНТУ обрала почесними докторами професорів ШУ М. Цайтца, П. Гьонера і К. Ротермеля, що активно співробітничать з факультетом ОТІ/КНТ, сприяють організації обміну студентами й аспірантами.

З 2000 р. до активного співробітництва залучено федеральний надпотужний обчислювальний центр (HLRS) ШУ, який надає паралельні ресурси для реалізації універсальних і проблемно орієнтованих паралельних моделюючих середовищ. Надійно функціонує теледоступ до суперкомп'ютерів NEC SX9, CRAY XE6 та інших паралельних систем. Директор HLRS проф. Міхаель Реш щорічно виступає з доповідями про актуальні розробки й перспективи в галузі суперкомп'ютингу перед студентами й викладачами ДонНТУ. В 2007 р. на базі ДонНТУ пройшов міжнародний семінар «Проблеми паралельного моделювання», в якому взяли участь 6 співробітників HLRS на чолі з проф. Решем, наукові співробітники інституту проблем моделювання в енергетиці (ПМЕ) НАН України, аспіранти, студенти й викладачі ФКНТ. З 2006 р. професори Реш і Святний є

членами програмних комітетів міжнародних конференцій «Моделювання» (Київ), «Моделювання та комп'ютерна графіка» (Донецьк/Покровськ), що проводяться ІІМЕ та факультетом КНТ ДонНТУ. В 2009 р. вчена рада ДонНТУ обрала М. Реша почесним доктором. В цьому ж році в німецьких та українських виданнях опубліковано сумісну наукову доповідь з проблем паралельного моделювання «Forschungsgebiet: parallele Simulationstechnik – Область досліджень: паралельне моделювання» (L. Feldmann, M. Resch, V. Svjatnyj, M. Zeitz). На симпозіумі з моделювання ASIM'2011 (Волфенбюттель, ФРН) та київській міжнародній конференції «Моделювання – 2012» кафедрою КІ та HLRS презентовано розробки паралельних симуляторів мережних динамічних систем з розподіленими параметрами. В червні 2010 р. разом з ОЦ введено в експлуатацію 100-вузловий паралельний кластер NEC 5800, переданий кафедрі КІ штутгартським HLRS. Восени 2013 р. проведено модернізацію кластера шляхом розширення кількості обчислювальних вузлів до 140 та заміни системи зв'язку між вузлами на більш потужну. Обладнання отримано від ШУ (HLRS) як гуманітарна допомога в липні 2013 р.

З 2001 року функціонує програма DAAD «Стипендії ім. Леонарда Ейлера» за проектом «Паралельне моделювання і розподілене моделююче середовище для автоматизованих систем» (професори М. Цайтц, П. Гьонер, М. Реш, В.А. Святний). Стипендії виділяються ШУ, що має багаторічну наукову співпрацю з ДонНТУ і призначені для підтримки студентів, аспірантів та молодих викладачів на завершальному етапі їх кваліфікаційних праць – магістерських та кандидатських. За ці роки 74 стипендіати кафедри «Комп'ютерна інженерія» (КІ) пройшли наукове стажування в Штутгартському університеті. За ініціативи професора Е.-Д. Гілlesa в 2001 р. засновано наукову програму з технічної кібернетики Pro3, за якою магістрантам та аспірантам кафедри КІ виділяється дві-три стипендії на рік. Вже 40 магістрантів, частина з яких стали аспірантами та асистентами, скористались цією стипендією. Наукові дослідження за цією програмою виконуються нашими стипендіатами в Макс-Планк-Інституті динаміки складних технічних систем (МПП, Магдебург)

та в ШУ й Гамбурзькому технічному університеті (ГТУ).

Ми маємо понад 80 спільних публікацій в німецьких та українських виданнях. За тематикою співробітництва захищено дві докторські (В.А. Святний – 1986, О. А. Дмитрієва – 2013) та 16 кандидатських дисертацій, в стадії підготовки до захисту знаходяться ще 5 дисертацій. Випускники кафедри І. Степанов, П. Скворцов, М. Краснік, М. Доста захистили дисертації відповідно в ШУ (2008, 2015), МПП (2008), ГТУ (2012). Дисертація Михайла Красніка отримала почесну медаль товариства Макса Планка.

До 2014 року кафедра активно співпрацювала з англійським та німецьким технічними факультетами ДонНТУ, англо-й німецькомовні студенти виконували курсові та дипломні роботи за темами наукового співробітництва з інститутами ШУ. Студенти німецькомовних груп проходять наукову практику в інститутах ШУ. На даний час в HLRS працюють випускники кафедри С. Гоголенко, Є. Коваленко, В. Кушнарєнко, О. Щербакєв, О. Чєпцов, О. Андрущенко, які є виконавцями проектів з ефективного застосування паралельних обчислювальних систем.

З 03 по 05 листопада 2013 року в ДонНТУ відбувся науковий семінар, присвячений 40-річчю співпраці з ШУ. Штутгартську делегацію очолював ректор університету професор Вольфрам Рєссєль. З доповідями виступили професори Андреас Ройтер, Рудольф Лаубєр, Курт Ротєрмєль, Міхаєль Цайтц, Петєр Гьонєр, Міхаєль Рєш, Олівер Заводни, Ахім Кінлє, Штєфан Вєснєр, Вольфганг Елєрс. Німецькі колеги отримали почесні грамоти за активне наукове співробітництво з ДонНТУ, а Олівер Заводни був обраний почесним доктором ДонНТУ по кафедрі КІ. Учасники семінару високо оцінили стан спільних наукових розробок, окреслили перспективи розвитку досліджень в галузі паралельного моделювання складних динамічних систем, зокрема побудови розподіленого паралельного моделюючого середовища на базі об'єднання паралельних ресурсів ШУ і ДонНТУ. На щорічному зібранні ШУ 15.11.2013 року В.А. Святного нагороджено почесною медаллю Штутгартського університету за успішне наукове співробітництво.

3 жовтня 2014 року ФКНТ ДонНТУ успішно працює в Покровську. Наші німецькі колеги з розумінням та активним співчуттям поставились до ситуації з переміщенням університету. Професор Курт Ротермель передав кафедрі КІ 63 одиниці комп'ютерного обладнання, професор Міхаель Реш і його співробітники скомпонували, виконали всі необхідні формальності і забезпечили транспортування високопотужного паралельного MIMD-кластера, що має 80 обчислювальних вузлів. В співпраці з Львівською політехнікою (професор М. В. Лобур) факультет отримав обладнання для двох лабораторій в рамках ТЕМПУС-проєкту. В 2016 році продовжено на 5 наступних років діючу з 2000 року угоду про наукове співробітництво з Штутгартським університетом. На кафедрах КІ і мовної підготовки створено німецьке відділення, яке покликане готувати фахівців європейського рівня і сприяти інтеграції ДонНТУ у світову спільноту технічних університетів, відновлено ДААД-проєкт німецькомовного навчання (DSG – deutschsprachige Studiengänge), який виконується спільно з Магдебурзьким університетом та НТУ КПІ і ХПІ. Випускники німецькомовних груп успішно працюють на факультеті КНТ, на фірмах України та Німеччини. В поточному році 7 магістрантів і один аспірант кафедри КІ пройшли дослідницьку практику на фірмах Німеччини, випускники Г. Нещадіна та А. Бардаков отримали запрошення на роботу. Вчена Рада ДонНТУ обрала почесними докторами професорів Штефана Гайнріха (2015, ГТУ), Вольфганга Елерса (2016, ШУ), Штефана Веснера (2017, ун-т Ульм), Міхаеля Вайріха (2018, ШУ). Випускники аспірантури К. Теплинський і А. Масюк захистили дисертації в спецраді ДонНТУ, працюють в Мюнхені. Теплинський заснував у Покровську ІТ-фірму, де працюють випускники кафедри і проходять всі види практик наші студенти. Визначна подія 2019 року – старт спільного з HLRS Штутгартського університету дворічного проєкту «ParSimTech: Розробка й імплементація українського дослідницько-навчального центру технологій паралельного моделювання в Донецькому національному технічному університеті (м. Покровськ)». Фінансування робіт забезпечують федеральне міністерство Німеччини BMBF та МОН України в рамках міжурядової WTZ-

програми. Прискіплива незалежна експертна комісія BMBF високо оцінила запит професорів М. Реша і В. Святного на цей проєкт, визнала результати попередніх спільних розробок та досліджень такими, що відповідають світовому стану робіт в галузі паралельного моделювання та суперкомп'ютингу. Це високе досягнення наукової школи ДонНТУ «Методи та засоби паралельного моделювання складних динамічних систем», заснованої професором Л.П. Фельдманом і активно підтримуваної нашими німецькими колегами. На жаль, вже пішли з життя професори **Лев Петрович Фельдман (1923-2017)**, **Рудольф Лаубер (1930-2017)**, **Ернст-Дітер Гіллес (1935-2019)**. Їхні вдячні колеги та учні продовжують успішно розвивати цей актуальний і відносно новий науковий напрямок – Parallele Simulationstechnik, Parallel Simulation Technology (ParSimTech – технології паралельного моделювання).

Я висловлюю щирю подяку професорам Міхаелю Цайтцу, Андреасу Ройтеру, Петеру Гьонеру, Курту Ротермелю, Міхаелю Решу, Ахіму Кінле, Томасу Бройнлю, Оліверу Заводни, Вольфраму Ресселю, Штефану Веснеру, Штефану Гайнріху, Міхаелю Вайріху та їх співробітникам за активну багаторічну співпрацю, за вагомий внесок у підготовку наукових і інженерних кадрів європейського рівня на факультеті КНТ ДонНТУ.

Зав. каф. КІ проф. Святний В.А.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ЩОДО ВИБОРУ ЛІКАРСЬКОГО ПРЕПАРАТУ

Тягунова М. Ю., Ільїна Н. О., Голуб Т.В.

Пропонується інтелектуальна система підтримки прийняття рішення щодо вибору лікарського препарату, розроблена з використанням методу імовірнісного вибору. Рішення в системі приймається на основі ваги відповіді користувача про симптоми захворювання. Розроблене програмне забезпечення дозволяє оцінити ефективність використання методу при створенні систем підтримки прийняття рішення.

Ключові слова: підтримка прийняття рішення, інтелектуальна система, імовірнісний вибір, нейронні мережі, обробка інформації, автоматизація медицини, захворювання, лікарські препарати.

Вступ. В сучасному світі інформаційні технології вдало проникають у всі сфери діяльності людини. Вони допомагають полегшити життя сучасного суспільства, автоматизувати і прискорити вирішення різного роду проблем. Для вчених основною задачею є синтез автоматичних структур, пов'язаних з моделюванням інтелектуальних можливостей людини, що здатні вирішувати складні завдання інформаційного забезпечення різних видів діяльності. Створення подібного механізму допоможе оптимізувати роботу людини в будь-якій області, зокрема й у медицині [1].

Сьогоднішній етап розвитку інформатизації охорони здоров'я характеризується повномасштабним впровадженням медичних інформаційних систем, що охоплюють практично всі сторони діяльності, включаючи керування ресурсами, лікувальним процесом і надання медичної допомоги. Зокрема, для допомоги людині у виборі лікарських засобів, що застосовуються для лікування захворювань, пошуку дешевших ліків, або визначення, до якого спеціаліста звернутися з відповідною хворобою, було розроблено інтелектуальну систему підтримки прийняття рішення.

Реалізація цієї системи дозволяє вирішити проблеми при прийнятті медичних рішень, зокрема таких, як недостатність знань, обмеженість часових ресурсів, відсутність можливості залучення великого числа компетентних експертів і неповнота інформації про стан пацієнта.

Мета. Допомогти людині-нефахівцю обрати необхідні медикаменти для лікування

захворювання на основі симптоматики нездужання.

Матеріали та методи. Щоб досягнути визначеної мети, був проведений аналіз методів обробки інформації в інтелектуальних системах та досліджено переваги та недоліки аналогічних медичних систем. В процесі реалізації інтелектуальної системи підтримки прийняття рішення щодо вибору лікарського препарату були розроблені структура і алгоритм функціонування системи та зручний в застосуванні веб-інтерфейс користувача. Проведено тестування коректності роботи системи та відповідності до вимог, що були поставлені в технічному завданні.

Аналіз останніх досліджень та публікацій. Для побудови системи були розглянуті два методи обробки інформації – використання нейронної мережі та метод імовірнісного вибору.

Застосування нейронних мереж для обробки інформації в системі не є доцільним, тому що для побудови об'єкта на основі нейронної мережі необхідно проводити багатоциклове налаштування внутрішніх елементів та зв'язків між ними, а це призводить до значних часових витрат. Ще одним їх недоліком є те, що поведінка навченої нейронної мережі не завжди може бути однозначно передбачуваною, що збільшує ризик застосування нейронних мереж в медичній галузі [2].

На відміну від нейронних мереж, імовірнісні методи забезпечують сприятливі умови для прийняття рішення і змістовні гарантії якості вибору. За допомогою імовірнісних методів можливо зробити вибір

рішення в умовах визначеності, якщо щодо кожної дії відомо, що вона незмінно призводить до деякого результату або вибір рішення при ризику, якщо кожна дія приводить до одного з безлічі можливих приватних результатів, причому кожен результат має обчислювану або експертно оцінювану ймовірністю появи [3]. Тому для створення інтелектуальної системи підтримки прийняття рішення щодо вибору лікарського препарату був використаний метод імовірнісного вибору.

Дослідження аналогів системи. Перед розробкою системи були розглянуті аналогічні медичні системи підтримки прийняття рішення та проведено детальний аналіз їх переваг та недоліків для забезпечення розроблюваної інтелектуальної системи максимальною ефективністю та зручністю в користуванні.

За допомогою додатку «Моя аптечка: аналоги лікарств» користувач може знайти дешеві ліки та дешеві аналоги дорогих ліків. Каталог «Моєї аптечки» нараховує близько 1000 безрецептурних ліків, що продаються в Росії, з вказанням їх середньої вартості. Використовуючи даний додаток можливо: знайти необхідні ліки, аналоги ліків, та їх ціни; дешеві аналоги дорогих ліків; скористатися пошуком ліків за певними симптомами, категоріями, діючій речовині, або повним каталогом; створити декілька профілів для усіх членів родини.

До недоліків даної системи слід віднести те, що вона розрахована переважно на користувачів однієї країни, користувачі з України можуть користуватися «Моєю аптечкою», але більшість ліків не буде доступною для них. Також значними недоліками є те, що при виборі симптомів, що турбують хворого, трапляються назви захворювань без пояснень, симптоми та розшифрування значень яких звичайний користувач може не знати, а також помилки при формуванні результатів опитування – при введенні симптомів «кашель, сухий кашель», система рекомендує ліки не тільки від сухого, а й від вологого виду кашлю, що може значно нашкодити здоров'ю хворого.

Simptomus – найбільш відома медична експертна система для діагностики захворювань, її місія - допомогти людям дізнатися, чим вони хворіють. В основі діагностичної системи лежить база знань, наповнена фахівцями в різних областях

медицини. При діагностиці використовуються. Завдяки цим алгоритмам стає можливим зробити висновки про можливі захворювання навіть в умовах неповних або неточних даних про стан пацієнта.

Діагностика відбувається миттєво з використанням складних алгоритмів штучного інтелекту: немає необхідності заповнювати анкету з безліччю полів, щоб отримати результати. Список можливих захворювань відображається практично миттєво - відразу ж після вказівки чергового симптому. Кожен новий симптом оновлює список ймовірних захворювань. Знайти свої симптоми системі можливо через "розумний пошук", або відповідаючи на питання, які задає система діагностики [4].

Недоліками системи Simptomus є непрацездатне поле для введення симптому – при натисканні на яке з'являється помилка «Сбой на сайте», проте вказати симптом можливо завдяки двом альтернативним пошукам – обравши на зображенні хвору частину тіла, або автоматичні питання лікаря.

Ще одним недоліком є неточність підбору питань системою – користувачу, що відповідає на питання стосовно однієї категорії хвороб, можуть бути запропоновані питання іншої, абсолютно несумісної категорії (наприклад, користувачу, якого турбує кашель, і він відповідає на питання про нього, система починає пропонувати питання про зубний біль, які зовсім не пов'язані з кашлем, та на основі них ставити можливі діагнози).

Експертна система «Домашній доктор 2.2» - проста медична експертна система. Вона визначає характер захворювання, ґрунтуючись на відповідях користувача, отриманих в результаті діалогу. База знань, що включає близько 100 поширених захворювань, була взята з книги: К.Нейлор «Як побудувати свою експертну систему».

Для використання системи необхідно завантажити її на свій комп'ютер та запустити її. Вікно з гіпотезами завжди знаходиться в робочому стані, тобто після кожної відповіді користувача, система змінює своє припущення щодо його хвороби.

Результати слід аналізувати після відповіді на всі запропоновані питання, переглянувши всі видані системою гіпотези у вікнах "Основна гіпотеза" і "Додаткові припущення" та уважно вивчивши опис до всіх представлених хвороб. Далі слід

проаналізувати схожість спостережуваних симптомів і симптомів в описанні кожної хвороби [5].

Основними недоліками є те, що система має невелику базу знань, що не доповнюється, застарілий зовнішній інтерфейс, та найголовніше – для того, щоб отримати результат, користувач має відповісти більше ніж на 60 питань для отримання основної гіпотези захворювання.

Зробивши аналіз аналогічних медичних систем, було вирішено забезпечити в інтелектуальній системі підтримки прийняття рішення щодо вибору лікарського препарату спрямування розробки – для користувачів з України (українські ліки та ціни), високу точність формування симптомів, запитань та рекомендацій лікарських засобів, надійність та відмовостійкість, можливість розширення бази знань, та зручний і сучасний інтерфейс користувача.

Структура системи. Ядром інтелектуальної системи є база знань, що містить в собі знання експертів (лікарів) щодо предметної області. Вона складається з двох компонентів:

- фактів – статичних даних, що введені лікарем (симптоми та назви хвороб, медичні препарати, що застосовуються для їх лікування, аналоги запропонованих ліків, перелік питань та варіантів відповідей для визначення захворювання, рекомендації щодо категорії спеціаліста, до якого слід звернутися з даними симптомами);

- правил – набору інструкцій, що дозволяє отримати нові факти, на основі тих, що мають (рекомендації медичних препаратів та категорії лікаря в залежності від введених користувачем симптомів).

База даних (робоча пам'ять) призначена для тимчасового зберігання початкових та проміжних даних, що є результатом спілкування інтелектуальної системи підтримки прийняття рішення щодо вибору лікарського препарату з користувачем, що веде з нею діалог.

Підсистема спілкування забезпечує ведення діалогу з користувачем інтелектуальної системи, в ході якого система запитує в нього факти та симптоми, що його турбують, необхідні для процесу міркування та визначення категорії лікарських засобів.

Машина логічного виводу являє собою механізм міркувань, що оперує з фактами та правилами з бази знань та даними, що

отримані від користувача системи, та зберігаються в робочій пам'яті. Основною метою машини логічного виводу є реалізація стратегії вибору відповідного правила, факту. Зокрема, вона керує чотирма процесами: вибором активних правил і фактів, їх зіставленням, вирішенням конфліктів, виконанням обраного зазначеного правила.

Підсистема пояснень необхідна для того, щоб дати можливість користувачу контролювати хід міркувань системи з вибору препарату для лікування. У разі відсутності конструктивних пояснень щодо рекомендацій певного лікарського препарату, користувач не матиме довіри до системи, і вона не матиме перспектив для використання.

Підсистема придбання знань автоматизує процеси наповнення системи знаннями, здійснювані лікарем. Лікар вносить свої знання в інтелектуальну систему, а саме в компонент придбання знань. Його основними функціями є: автоматизація процесу наповнення бази знань та актуалізація бази знань. Для функціонування підсистеми придбання знань розробляються алгоритми функціонування модуля накопичень знань та веб-інтерфейси користувачів, що дозволяють здійснювати операції маніпулювання зі знаннями.

При роботі з інтелектуальною системою підтримки прийняття рішення, користувач системи за допомогою Всесвітньої мережі Інтернет заходить до веб-браузера, де вводить адресу системи підтримки прийняття рішення щодо вибору лікарського препарату. При спрацюванні запиту, він бачить перед собою інтерфейс користувача, куди може ввести необхідні для аналізу дані.

Після введення, дані за допомогою веб-серверу передаються в робочу пам'ять інтелектуальної системи, звідки потрапляють до інших компонентів системи, в результаті чого формується результат роботи інтелектуальної системи, що потрапляє в робочу пам'ять та за допомогою веб-сервера відображається в інтерфейсі користувача.

Інтелектуальна система підтримки прийняття рішення щодо вибору лікарського препарату працює в двох режимах: режимі придбання знань і в режимі вирішення задачі.

У режимі придбання знань спілкування з системою здійснює (через посередництво інженера по знаннях) експерт. В цьому

режимі лікар, використовуючи компонент придбання знань, наповнює систему знаннями, які дозволяють інтелектуальній системі в режимі вирішення самостійно (без експерта) вирішувати завдання вибору лікарського препарату.

У режимі консультації спілкування з інтелектуальною системою здійснює кінцевий користувач, якого цікавить результат та (або) спосіб його отримання. В цьому режимі дані, що задані користувачем, після обробки їх діалоговим компонентом надходять в робочу пам'ять. Машина логічного виводу на основі вхідних даних з робочої пам'яті, загальних даних предметної області і правил з бази знань формує рішення задачі.

На рисунку 1 приведено структуру інтелектуальної системи підтримки прийняття рішення щодо вибору лікарського препарату.

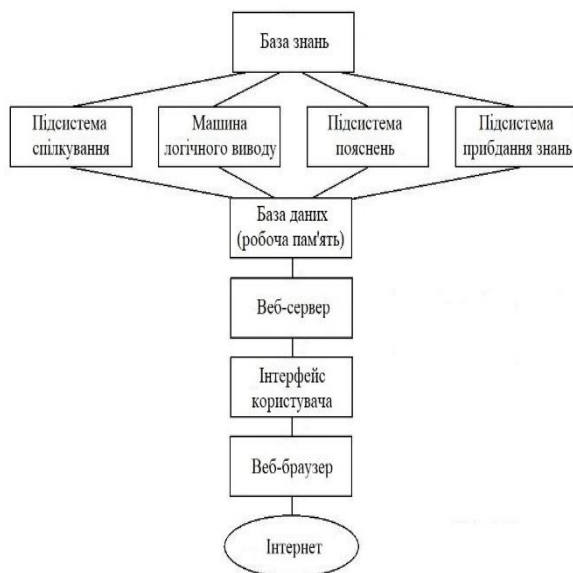


Рис. 1 – Структура інтелектуальної системи підтримки прийняття рішення щодо вибору лікарського препарату

Алгоритм роботи системи.

Узагальнений алгоритм функціонування інтелектуальної системи підтримки прийняття рішень наступний:

1) в інтелектуальну систему завантажуються список стандартних запитань, порядок і зміст яких є незмінним для кожного з користувачів;

2) користувач обирає одну з декількох відповідей на кожне запитання;

3) на основі відповідей на стандартні запитання, визначається модуль бази знань,

що містить групу наступних (уточнюючих) запитань, що належать обраній стандартній послідовності;

4) в систему завантажуються відповідна група уточнюючих запитань, визначається їх кількість та розраховується вага одного запитання;

5) користувачу відображуються запитання, він робить свій вибір:

- при відповіді «ні» вага запитання віднімається від загальної ваги, відбувається перехід до наступного запитання, обраного випадковим чином з усієї групи запитань;

- при відповіді «не знаю» або «так» до загальної ваги додається відповідно половина або повна вага запитання. Відбувається перехід до наступного запитання, або п.б.;

6) вихідні дані, введені користувачем, зберігаються, коли загальна вага відповідей становить більше 80% (0.8);

7) проводиться зіставлення вихідних фактів з базою знань, визначається ідентифікатор захворювання, що отримав найбільшу вагу;

8) результат роботи системи у вигляді переліку лікарських засобів відображається користувачеві.

Блок-схему алгоритму роботи інтелектуальної системи підтримки прийняття рішення представлено на рисунку 2.

Реалізація системи. Користувач на початку роботи з інтелектуальною системою потрапляє на головну сторінку системи, на якій він має можливість ознайомитися з основною інформацією про систему та розпочати роботу з нею.

Коли користувач натискає кнопку «Розпочати роботу», на сервер відправляється запит за допомогою Java Script (JS) для отримання ідентифікаторів стандартних запитань. Вони надходять у вигляді json-об'єктів та конвертується в масив JS. На рисунку 3 зображений результат, отриманий від сервера.

В цей час користувачеві відображається перше питання із сформованого масиву стандартних запитань. Коли користувач натискає на відповідь, з масиву стандартних запитань видаляється елемент масиву, що відповідає цьому запитанню (рисунок 4).

Останнім стандартним запитанням є перелік груп захворювань певної частини тіла, де в користувача виникають больові

відчуття. Коли користувач обирає необхідну частину тіла, розпочинається завантаження уточнюючих запитань, що належать до групи відповідей на стандартні запитання. На рисунку 5 зображено зупинку функції стандартних запитань та запуск функції уточнюючих запитань. На цьому рисунку також можна побачити отриману від сервера вагу відповіді на запитання, а також кількість питань, що належать до цієї групи.

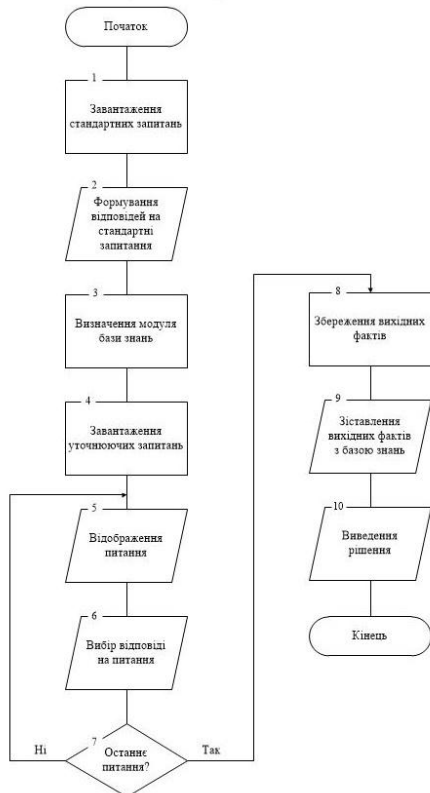


Рисунок 2 - Блок-схема алгоритму роботи інтелектуальної системи

```

Инспектор  Консоль  Отладчик  Сеть  Стили  Профайлер
Поиск в консоли
[{"id": "1"}, {"id": "2"}, {"id": "3"}, {"id": "4"}, {"id": "5"}, {"id": "6"}, {"id": "7"}]
Array(7) [ "1", "2", "3", "4", "5", "6", "7" ]
[{"id": "1", "question": "\u0414\u043e\u0431\u0440\u0430\u0442\u0430 \u0432\u0430\u0433\u043e\u0432 \u0432\u043e\u0432\u043e\u0434\u0443 \u0432\u043e\u0434\u043e\u0432\u0438", "id": "2", "question_id": "1", "answer": "\u0414\u043e\u0432\u0435\u0434\u0430\u0432\u0430\u0432"}]
1
1

```

Рисунок 3 – Отриманий від сервера результат

Початкова вага відповідей користувача дорівнює нулю. Коли користувач відповідає «так» на уточнююче запитання, то вага, що була отримана при завантаженні уточнюючих запитань, додається до поточної ваги відповідей користувача (рисунком 6).

```

Инспектор  Консоль  Отладчик  Сеть  Стили  Профайлер
Поиск в консоли
2
standard: 2,3,4,5,6,7
[{"id": "2", "question": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0443 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "3", "question": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "6", "question_id": "3", "answer": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430"}]
3
5

```

Рисунок 4 – Видалення елемента масиву

```

Инспектор  Консоль  Отладчик  Сеть  Стили  Профайлер
Поиск в консоли
end
false
start
голова
{"weight": 0.14285714285714285, "count": "28"}
28
{"id": "5", "title": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "6", "question": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "6", "question_id": "3", "answer": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430"}]

```

Рисунок 5 - Зупинка функції стандартних запитань та запуск функції уточнюючих запитань

```

Инспектор  Консоль  Отладчик  Сеть  Стили  Профайлер
Поиск в консоли
weight: 0.14285714285714285
yes: 5
no:
dont:
{"id": "6", "title": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "6", "question": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "6", "question_id": "3", "answer": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430"}]
Чи бувало у вас нудота під час головного болю?

```

Рисунок 6 – Стан ваги відповідей при значенні «так»

На рисунку також можна побачити поточні значення масивів відповідей, що стосується до кожної з них («так», «не знаю», «ні»); оскільки користувачем була натиснута кнопка «так», ідентифікатор питання був доданий до масиву відповідей «так», та змінено поточне значення ваги відповідей.

Коли користувач відповідає на питання «не знаю», то до загальної ваги відповідей додається половина значення ваги однієї відповіді (рисунком 7).

```

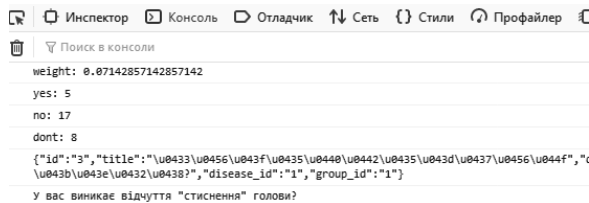
Инспектор  Консоль  Отладчик  Сеть  Стили  Профайлер
Поиск в консоли
weight: 0.21428571428571427
yes: 5
no:
dont: 8
{"id": "7", "title": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "question": "\u0414\u043e\u0432\u0430\u0432 \u0432\u043e\u0432\u043e\u0434\u0442\u0430 \u0432\u043e\u0432\u043e\u0434\u0442\u0430", "id": "7", "question_id": "3", "group_id": "1"}]
Біль в голові має пульсуючий характер?

```

Рисунок 7 - Стан ваги відповідей при значенні «не знаю»

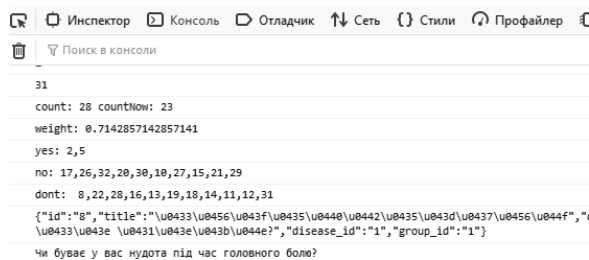
З рисунку видно, що значення поточної ваги відповідей змінилося – до загальної ваги відповідей була додана половина ваги відповіді, а також в масив відповідей «не знаю» додався ідентифікатор питання.

При відповіді на запитання «ні», від поточного значення ваги віднімається вага відповіді, а також до масиву відповідей «ні» додався ідентифікатор питання (рисунк 8).



Рисунк 8 - Стан ваги відповідей при значенні «ні»

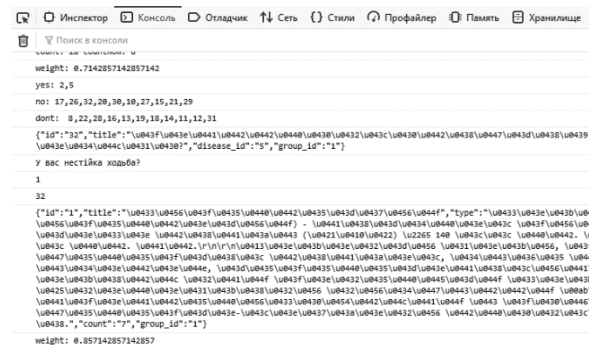
Таким чином, користувач відповідає на ряд уточнюючих запитань, поки загальна вага відповідей не перевищить значення 80% (0.8). На рисунку 9 зображений стан змінних перед тим, як система визначить результат опитування.



Рисунк 9 – Стан змінних перед визначенням результатів опитування

При наступній відповіді користувача «так», загальна вага відповідей перевищить

80% і система визначить результат, що буде відображений користувачеві (рисунк 10).



Рисунк 10 – Визначення результатів опитування

На рисунку можна побачити, що оскільки загальна вага перевищила 80%, в систему надійшли дані з результатами опитування у вигляді json-об'єкту. Після цього автоматично завантажується сторінка з інформацією щодо захворювання для користувача та лікарськими засобами, що застосовуються для його лікування.

Висновки. Таким чином, розроблена система готова до експлуатації. З її допомогою користувач зможе визначити своє захворювання та обрати лікарський препарат для його лікування. Наукове значення являє собою висока точність роботи системи завдяки методу імовірнісного вибору. Можливими варіантами для модернізації роботи системи в майбутньому є покращення зручності веб-інтерфейсу користувача на інших мовах, а також доопрацювання сторінки з результатами роботи системи так, щоб на ній відображався діапазон цін лікарського засобу у різних аптеках міста, в якому мешкає користувач.

Література

- [1] К. Сексенбаєв, Б. К. Султанова, М.К. Кісіна, Інформаційні технології в розвитку сучасного інформаційного суспільства. Казань: «Молодий вчений», 2015.
- [2] Д. С. Корнеєв, Використання апарату нейронних мереж для створення моделі оцінки та управління ризиками підприємства. Київ: "Університет банківської справи", 2007.
- [3] Інтелектуальні інформаційні системи в умовах невизначеності і ризику [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/6020505/page:26/>.
- [4] Сервіс онлайн-діагностики захворювань Symptomus [Електронний ресурс] – Режим доступу до ресурсу: <http://symptomus.ru/>.
- [5] Домашній доктор 2.2 [Електронний ресурс] – Режим доступу до ресурсу: http://www.aiportal.ru/downloads/expert-systems/home_doctor_2_2.html.

References

- [1] K. Seksenbaiev, B. K. Sultanova, M.K. Kisina, *Informatsiini tekhnolohii v rozvytku suchasnoho informatsiinoho suspilstva*. Kazan: «Molodyi vchenyi», 2015.
- [2] D. S. Kornieiev, *Vykorystannia aparatu neuronnykh merezh dlia stvorennia modeli otsinky ta upravlinnia ryzykamy pidpriemstva*. Kyiv: "Universytet bankivskoi spravy", 2007.
- [3] *Intelektualni informatsiini systemy v umovakh nevyznachenosti i ryzyku [Elektronnyi resurs]* – Rezhym dostupu do resursu: <https://studfile.net/preview/6020505/page:26/>.
- [4] *Servis onlain-diahnostyky zakhvoriuvan Simptomus [Elektronnyi resurs]* – Rezhym dostupu do resursu: <http://simptomus.ru/>.
- [5] *Domashnii doktor 2.2 [Elektronnyi resurs]* – Rezhym dostupu do resursu: http://www.aiportal.ru/downloads/expert-systems/home_doctor_2_2.html.

Надійшла до редакції 09.12.2019 р.

Тягунова Марія Юрївна – кандидат технічних наук, доцент, Національний університет «Запорізька політехніка» (вул.Жуковського, 64, м. Запоріжжя, 69063, Україна).

E-mail: mary.tyagunova@gmail.com.

Ільїна Наталія Олександрівна – студентка магістратури, Національний університет «Запорізька політехніка» (вул.Жуковського, 64, м. Запоріжжя, 69063, Україна).

E-mail: natka7natka@gmail.com.

Голуб Тетяна Василівна – асистент, Національний університет «Запорізька політехніка» (вул.Жуковського, 64, м. Запоріжжя, 69063, Україна).

E-mail: golub.tv6@gmail.com

INTELLIGENT DECISION SUPPORT SYSTEM FOR DRUG SELECTION

An important aspect of medical informatization is the development of effective expert systems that are available for use by unskilled users. Such systems are designed to help the patient in the initial diagnosis of the disease, as well as the choice of drugs, the nearest pharmacies, recommendations for the use of drugs, the choice of the doctor of the required specialty, etc.

In this paper, a description of the developed decision-making system is presented, which allows an ordinary user to independently perform the initial diagnosis of the disease and the selection of suitable drugs. The system's operation is based on assigning various weighting factors to questions, depending on the answers received from the user, and the subsequent logical conclusion using the probabilistic choice method. The use of this method is preferable, in comparison with the use of artificial neural networks, due to the greater uncertainty in the behavior of the latter.

The system includes the following main components: a knowledge base (contains facts and rules), a logical inference machine, a database (acts as a working memory, and stores current user responses), subsystems for communicating with the user, explanations and knowledge acquisition. The expert system is implemented as a web application using the Java Script programming language and the JSON data exchange format.

The tests carried out showed the system's operability and its suitability for practical use.

Keywords: *decision support, intelligent system, probabilistic choice, neural networks, information processing, automation of medicine, diseases, drugs.*

Tiahunova Mariia Yuriivna – Doctor of Philosophy, Associate Professor, National University "Zaporizhia Polytechnic" (Zhukovsky str., 64, Zaporozhye, 69063, Ukraine).

E-mail: mary.tyagunova@gmail.com.

Ilina Nataliia Oleksandrivna – Master's student, National University "Zaporizhia Polytechnic" (Zhukovsky str., 64, Zaporozhye, 69063, Ukraine).

E-mail: natka7natka@gmail.com.

Golub Tetiana Vasulivna – assistant, National University "Zaporizhia Polytechnic" (Zhukovsky str., 64, Zaporozhye, 69063, Ukraine).

E-mail: golub.tv6@gmail.com

РОЗРОБКА ТА МОДЕЛЮВАННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» З ПІДВИЩЕНОЮ НАДІЙНІСТЮ НА БАЗІ ARDUINO

Федько А.О., Зеленьова І.Я., Тягунова М.Ю.

В роботі приведено розробку та моделювання системи "розумний будинок" з підвищеною надійністю на базі популярної платформи Arduino. Запропонована система "розумного будинку" має архітектуру, в якій плата Arduino виступає у ролі пристрою-посередника, а комп'ютер сконфігурований так, що можна здійснювати віддалене керування системою зі смартфона чи планшета. Описано моделювання системи в пакеті Proteus VSM. Також запропоновано способи підвищення надійності системи та приведено результати тестування.

Ключові слова: мікроконтролер, Arduino, Proteus VSM, клієнтський додаток, підвищення надійності

Вступ. Сучасна людина приділяє велику увагу рівню комфортності середовища проживання. До умов комфорту входять, наприклад, клімат-контроль приміщення, побутові потреби, рівень безпеки помешкання. В результаті реалізації усіх цих потреб рівень складності систем інженерного оснащення квартир та приватних будинків швидко зростає, так само як і зростає кількість пристроїв, які входять до складу подібних систем. В даній роботі розглядається комплексна система «розумний будинок», яка допомагає користувачеві з вирішенням задачі керування такою складною системою, беручи на себе основні потоки даних та залишаючи за користувачем лише прийняття основних рішень.

На даний момент розробка систем «розумного будинку» стала значно простішою, ніж кілька років тому. На сучасному світовому ринку пропонується велика кількість готових рішень для реалізації подібних систем, але функціональність таких готових систем не завжди задовольняє потреби кінцевого користувача, тому одним із сучасних способів досягнення оптимального та зручного рішення є розробка власної системи «розумного будинку».

Одним з популярних варіантів для проектування схеми «розумного будинку» на сьогодні є платформа Arduino, побудована на базі мікроконтролерів Atmega [1]. Завдяки доступній вартості керуючих контролерів і великій кількості сенсорів можна побудувати абсолютно будь-яку систему.

На даний момент при розробці систем на базі Arduino в якості керуючого пристрою

використовують саму плату Arduino. В таких типах систем управління платою здійснюється за допомогою кнопок та енкодерів, які об'єднуються у керуючу панель. Якщо необхідно здійснити віддалений контроль, то використовують GSM модулі, що в епоху стрімкого поширення Інтернету вже стає недоцільним. Для реалізації віддаленого керування можна також використовувати контролери ESP, які підтримують протоколи WiFi, але такі контролери значно дорожчі за Arduino [2].

В даній роботі запропоновано систему, у якій в якості керуючого пристрою виступає звичайний користувацький комп'ютер, а Arduino являється пристроєм передачі даних. Переваги такого використання Arduino у тому, що знижується навантаження на сам мікроконтролер, а завдяки комп'ютеру можна здійснювати способи віддаленого управління системою за допомогою мережі Інтернет. Також пропонується варіант системи із керуванням з мобільного телефону.

Однак платформа Arduino є скоріше аматорським, ніж професійним рішенням. У зв'язку з цим виникає необхідність забезпечити підвищення надійності систем «розумного будинку», побудованих на базі Arduino.

Аналіз останніх досліджень та публікацій. «Розумний будинок» (англ. Smart home) – це комплекс рішень для автоматизації повсякденного догляду за житловим приміщенням, який позбавляє людину від виконання рутинних дій [3]. Тут і побутова техніка – від роботів-пилососів до приладів, керованих із смартфона, і системи,

здатні контролювати все, що відбувається в помешканні.

Основними компонентами системи «розумного будинку» є сенсори, актуатори, пристрій керування, а також програмне забезпечення [4]. У системах деяких виробників, в тому числі і від Arduino, користувач може розробити програму самостійно, в інших – використовуються готові рішення, в яких користувачеві доступні лише типові сценарії.

Сучасні системи «розумного будинку» поділяють на декілька різновидів [5]:

- оснащені власним контролером або обчислювальним сервером;

- системи, які використовують в якості процесора смартфон чи планшет користувача;

- системи, які опрацюють інформацію за допомогою віддаленого сервера, що належить компанії-розробнику (хмарний сервіс).

На сьогоднішній день, ринок систем «розумного будинку» розвивається стрімкими кроками. Розглянемо наступні готові рішення, такі як, Revogi Sence Kit, Xiaomi Smart home Suite, Fibaro Starter kit.

Revogi Sence Kit – це інтелектуальна система безпеки для житлових приміщень [6]. Даний набір дозволяє легко створити систему, яка об'єднує всі домашні пристрої в одну екосистему. Всіма пристроями у наборі можна управляти за допомогою смартфона або планшета.

Однією з переваг системи «розумного будинку» від компанії Revogi є задоволення естетичних потреб користувача, оскільки продукція має стильний та компактний вигляд. Також подібні системи досить прості для монтажу і не потребують окремих ліній живлення та ліній зв'язку, а потрібна лише WiFi точка доступу. До недоліків цього рішення можна віднести невелику кількість актуаторів, а також невеликий спектр сенсорів, які можна встановити у приміщенні.

Компанія Xiaomi – китайська компанія, заснована Леєм Цзюнем у 2010 році [7]. Спеціалізується на виробництві електронної техніки, в першу чергу смартфонів та «розумних пристроїв». На даний момент компанія має лінійку пристроїв Xiaomi Smarthome Suite, які контролюються за допомогою Xiaomi MiHome – єдиного додатку, який керує продуктами розумного

дому Xiaomi: він використовується для управління та спілкування з пристроями, для підключення їх до мережі, а також для підключення пристроїв один до одного за допомогою WiFi та Bluetooth. До складу Xiaomi MiJia Smart home Security Kit входять різноманітні сенсори, аналізатори, камера спостереження, підсистеми керування режимом роботи побутових пристроїв, а також сучасний потужний центр керування системою безпеки – комутатор MiControlHub. Це головний пристрій, який потрібно підключити в MiHome, а вже до нього приєднувати і інші пристрої. Деякі пристрої можуть працювати окремо від хаба. Також на хабі є система підсвічування, яка може відображати до 16 млн. кольорів. В налаштуваннях хаба можна обрати режим, щоб підсвічування вмикалося автоматично, якщо в приміщенні недостатньо освітлення.

Як і у випадку з продукцією компанії Revogi, продукція компанії Xiaomi має сучасний дизайн, а також володіє більшою кількістю рішень у порівнянні з конкурентом. Fibaro Starter KIT EU Z-Wave – початковий комплект для організації розумного будинку Fibaro [8]. У цей набір входять, наприклад, контролер розумного будинку Fibaro Home Center Lite Z-Wave, розумний сенсор руху Fibaro Motion Sensor 3в1 Z-Wave, сенсор відкриття дверей / вікна FIBARO Door / Window Sensor 2 Z-Wave, сенсор протікання води Fibaro Flood Sensor Z-Wave, сенсор диму Fibaro Smoke Sensor Z-Wave, розумна розетка Fibaro Wall Plug Z-Wave. Fibaro Home Center Lite - це контролер для домашньої автоматизації, який використовується для управління системою розумного будинку Fibaro. Можна з'єднати до 230 актуаторів і сенсорів за допомогою бездротового протоколу Z-Wave. Комплекс працює зі зручним графічним інтерфейсом, який дозволяє контролювати всі пристрої і створювати сценарії.

Однак, розглянуті системи є кінцевими, «закритими», оскільки ці проектні рішення є готовим продуктом і не призначені для інтеграції в умови, не передбачені паспортними даними. Для таких цілей краще використовувати популярну нині платформу Arduino. Розробляючи систему на базі Arduino можна самому визначати варіант кінцевої системи, беручи до уваги також способи монтажу та набори вихідних даних для керування цією системою.

Arduino – італійська компанія, що займається розробкою і виробництвом компонентів і програмного забезпечення для систем «розумний будинок», призначених для неспеціалістів [6]. Примітним є те, що цей розробник створив повністю відкриту архітектуру запропонованих систем, що дало можливість стороннім виробникам розробляти нові та копіювати вже існуючі Arduino-сумісні пристрої, а також випускати для них програмне забезпечення.

Апаратна частина системи складається з плати із мікроконтролером (процесорна плата) і встановлених на ній плат розширення, які в побуті називають *shield*. Підключення плат розширення до процесорної плати дозволяє додавати до «розумного будинку» нові компоненти. Зібрана система може бути як повністю автономною, так і працювати сумісно з комп'ютером через стандартний дротовий або бездротовий інтерфейс.

Платформа Arduino представлена великою кількістю плат, які базуються на різних типах мікроконтролерів: Arduino Uno, Nano, Mega, Leonardo, Mini та інші. Всі ці плати відрізняються кількістю портів вводу/виводу, а також різними типами мікроконтролерів.

Також платформа Arduino має значний набір актуаторів, серед них: електродвигуни, реле та різноманітні перемикачі, диммери для плавного змінювання інтенсивності освітлення та інше.

Комплекс на платформі Arduino надає такі переваги:

- можливість автономної роботи, обумовлена наявністю власного контролера;
- широкі можливості з налаштування роботи системи (користувач сам пише програму, в якій можуть бути передбачені сценарії будь-якої складності);
- простота процесу завантаження програми в контролер: програматор для цього не потрібен, достатньо мати USB-кабель (в мікроконтролері є прошивка завантажувача Bootloader);
- доступна вартість компонентів, обумовлена відсутністю у того чи іншого виробництва монопольних прав (архітектура є відкритою).

До недоліків можна віднести те, що контролер Atmega, встановлений на плату Arduino, є однопоточним і не може виконувати завдання паралельно.

На даний момент існує багато середовищ моделювання платформи Arduino, це зумовлено значною популярністю даної платформи, серед них виділяють наступні:

- Tinkercad;
- Virtual BreadBoard;
- Proteus VSM 8.1.

За допомогою цього класу програм можна не тільки проектувати електронні схеми, а й віртуально підключати їх до електричного кола за допомогою вбудованого симулятора. У режимі реального часу можна спостерігати за поведінкою схеми, перевіряти і налагоджувати її працездатність. Якщо в такий симулятор додати віртуальну плату Arduino, то можна відстежити поведінку схеми і в Arduino-проектах. Для налагодження скетчів у багатьох відомих сервісах присутня також можливість завантаження справжніх скетчів, які «завантажуються» в модель, і змушують вести схему з підключеними елементами так само, як і з включеною реальною платою. Таким чином, можливо симулювати роботу досить складних проектів без фізичного підключення Arduino, що істотно прискорює розробку. Дані середовища представляються як у вигляді он-лайн сервісів, так і у вигляді готових продуктів під різноманітні операційні системи.

В ході проведення аналізу існуючих систем «розумного будинку» було виявлено, що готові рішення не задовольняють усі потреби користувача, тому в якості основної платформи розробки вирішено використовувати апаратно-обчислювальну платформу Arduino, яка буде здійснювати комунікацію з програмним забезпеченням через послідовний порт комп'ютера.

Метою даної роботи є підвищення надійності складних систем «розумного будинку», спроектованих на базі популярної платформи Arduino.

Матеріали і методи. В загальному вигляді система «розумний будинок» складається з двох рівнів: перший рівень – центральний контролер, який оброблює дані які поступають з сенсорів і видає керуючі сигнали на основі отриманих даних; другий рівень складається з сенсорів та актуаторів. В такій архітектурі головний контролер являється пристроєм з кнопками керування та дисплеєм. При використанні платформи Arduino, головний контролер представляється у вигляді одної з плат Arduino (наприклад:

Arduino Nano, Arduino Leonardo та інші), яка в свою чергу базується на мікроконтролері Atmega.

В даній роботі пропонується інша архітектура, де головний контролер виступає у якості пристрою-посередника, тобто він приймає дані з сенсорів та передає ці дані на комп'ютер чи сервер. В такій системі комп'ютер чи сервер видають керуючі сигнали на головний контролер, який в свою чергу відправляє керуючі сигнали на актуатори, наприклад, по сигналу з

керуючого комп'ютера, відбувається відкриття вікон чи ввімкнення світла. При такому підході на комп'ютер чи сервер встановлюється програма, яка розроблена для системи «розумного будинку», при цьому сервер може бути сконфігурований таким чином, що можна керувати даним «розумним будинком» віддалено, через будь-який пристрій, що має вихід в мережу Інтернет. На рисунку 1 зображено запропоновану архітектуру з виділенням комп'ютером.

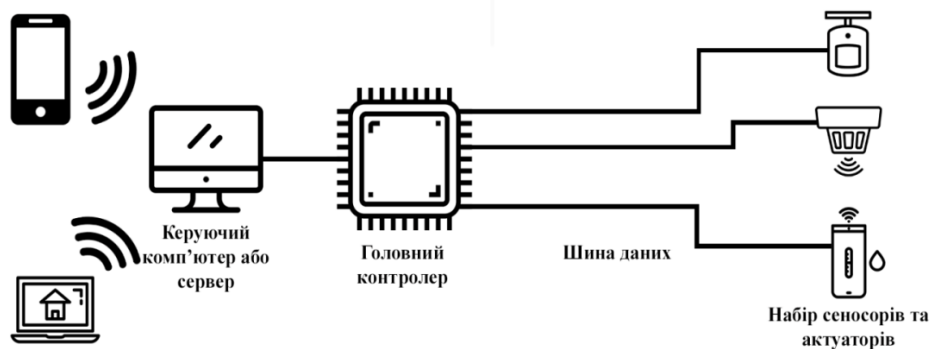


Рисунок 1 – Архітектура системи «розумний будинок» з виділенням комп'ютером

Організація віддаленого доступу, для керування системою через смартфон, забезпечується за допомогою Web-серверу, який встановлюється на керуючий комп'ютер. Вдалим варіантом для організації такої системи є використання одноплатного комп'ютера Raspberry Pi в якості пристрою керування. Оскільки Raspberry Pi працює на базі операційної системи Linux, то для віддаленого контролю необхідно лише налаштувати звичайний LAMP сервер. Всі дані у системі передаються у форматі JSON, які можна передати на спеціально розроблену web-сторінку, через яку можна віддалено керувати системами даного «розумного будинку».

Спектр обладнання, яке можна встановити в житло, вкрай широкий і різноманітний, тому кожна частина структури представляє собою складну систему, що дозволяє вирішувати практично будь-які завдання, поставлені замовником. У даній роботі враховані основні функції системи, які найбільш поширені і широко використовуються в «розумних будинках» по всьому світу.

Обравши всі необхідні компоненти, необхідно програмно описати функції, що

виконуються системою керування «розумний будинок». Програму можна розділити на смислові частини, кожна з яких відповідає одній з функцій, які виконуються «розумним домом».

Першою функцією, яку виконує система «розумний будинок», є система контролю освітлення. Дана система базується на сенсорах руху та рівня освітленості. Головний контролер системи отримує дані з цих сенсорів, і виконує необхідні дії. Другою функцією системи «розумний будинок» є контроль температури у приміщенні, що організовано наступним чином: система отримує дані з сенсору температури, після цього проходить порівняння значень з встановленими, якщо температура вище, ніж потрібно, то вмикається система охолодження, якщо нижче – система опалення.

Наступна система – це система безпеки, яка складається з двох основних підсистем, перша – безпека від проникнення, друга – протипожежна безпека.

Система контролю розумних пристроїв працює за дуже простим алгоритмом: система очікує команду від користувача, та по команді активує той чи інший пристрій. Алгоритм запускається циклічно, у

проміжках між передачею даних, спеціальна функція отримує керуючі сигнали для даної системи.

Моделювання є невід'ємною частиною розробки будь-якої системи керування та дозволяє налагодити виконання основних завдань та функцій системи. На даному етапі роботи проводиться моделювання основних функцій системи «розумний будинок» в середовищі Proteus VSM.

Пакет моделювання Proteus VSM у базовому варіанті не має компонентів, пов'язаних з платформою Arduino, тому є декілька варіантів розробки моделі для системи на базі платформи Arduino: перший – зібрати принципову схему з паспорту на пристрій на Arduino UNO, оскільки ця плата базується на мікроконтролері ATmega328, який присутній у базовому пакеті Proteus VSM, другий – скористатися вже готовою бібліотекою Arduino.

Для даного проекту нам знадобляться наступні бібліотеки для Proteus VSM:

- Arduino UNO Library – бібліотека додає плату Arduino UNO у каталог наявних пристроїв;
- Gas Sensor Library – додає сенсори газу серії MQ;
- PIR-Sensor Library – додає сенсори руху;
- DHT-11 Library – додає сенсори температури та вологості DHT-11;
- LDR Sensor Library - додає сенсори освітленості.

Всі ці бібліотеки необхідно скопіювати у папку ...\\Proteus\\LIBRARY.

Наступними кроками моделювання є:

- створення проекту в середовищі Proteus VSM;
- вибір та додавання компонентів, з яких будується система, а саме – вибір сенсорів, актуаторів та мікроконтролеру для даної системи «розумного будинку»;
- об'єднання всієї компонентів у принципову схему системи;
- налаштування конфігураційних файлів для сторонніх бібліотек;
- розробка прошивки для мікроконтролеру;
- розробка клієнтського додатку для керування системою.

Оскільки в даній системі платформа Arduino є лише приймачем/передавачем інформації, а рішення приймає головний комп'ютер, то на основі всіх даних, що

передаються, й відбувається керування системою.

Головний комп'ютер керує системою за допомогою «командних слів», посылаючи ці слова по послідовному порту, до якого підключений головний контролер системи. Контролер аналізує надіслане «керуюче слово», наприклад, слово «heatsystem», активує систему опалення та вмикає цю систему, при цьому передаючи інформацію про її стан на головний комп'ютер. При цьому кожне «керуюче слово» обов'язково повинне бути відокремлене спеціальним символом, в даному випадку це символ «@».

Як було зазначено раніше, Arduino є скоріше пристроєм передачі даних, на стороні мікроконтролера не приймаються основні рішення. Всі рішення про роботу системи приймаються на комп'ютері, до якого підключається плата Arduino. Клієнт за допомогою програмного додатку може здійснювати контроль над системою, при цьому цей додаток може бути як мобільним так і будь-яким іншим, оскільки дані надходять в універсальному форматі.

Розроблено клієнтський додаток, написаний на мові програмування Java з використанням бібліотеки Swing для створення графічного інтерфейсу системи.

Для тестування розробленої системи необхідно об'єднати середовище моделювання Proteus VSM та розроблений клієнтський додаток, це можна зробити за допомогою допоміжної програми, яка створює віртуальний послідовний порт – Virtual Serial Ports Emulator (VSPE).

Способи підвищення надійності системи. Підвищення надійності системи на базі мікроконтролерів пропонується здійснити наступними способами, а саме:

- резервування контролерів;
- здійснення резерву по живленню;
- резервування каналів зв'язку.

При «гарячому» резервуванні дві (або більше) плат платформи Arduino підключаються паралельно до джерела живлення, причому пристрої знаходяться в активному стані постійно [10]. Для реалізації цього додамо до розробленої системи ще одну плату Arduino UNO.

У схемі присутні два контролери Arduino UNO, які підключені до сенсорів паралельно. Також обидва контролери пов'язані за допомогою послідовного порту, який представлений двома контактами 0 (RX)

і 1 (RX) кожної з плат. Такий спосіб підключення дозволяє отримувати достовірні дані з сенсорів на обох контролерах одночасно, та надсилати їх по послідовному порту (реальному чи віртуальному). При цьому контролери мають однаковий програмний код, але за однієї умови: оскільки послідовний порт не може приймати дані одночасно від двох контролерів, то необхідно встановити затримку для передачі даних. Перевіряємо результат роботи системи, при симуляції використовуємо інструмент середовища Proteus VSM – «Віртуальний термінал».

Резервування платформи Arduino з використанням сигнального дроту – такий спосіб можна напевне віднести до так званого «теплого» резервування. Для здійснення резерву з використанням сигнального дроту із головного контролера на резервний поступає сигнал з п'ятивольтової лінії, резервний контролер постійно зчитує цей сигнал, і поки надходить 5В, знаходиться у режимі очікування. Після того, як на вхід резервного контролера перестає надходити сигнал в 5В, резервний контролер переходить до активного стану та ініціює виконання основного коду програми. Доки резервний контролер знаходиться в стані очікування, його порти відключені, а після переходу в активний стан всі порти починають працювати у штатному режимі. При всьому цьому один контакт контролеру зарезервованій та підключений до контакту перезавантаження контролеру, що необхідно для випадків, коли головний контролер може повернутися в робочий стан.

Для здійснення «холодного» резерву, необхідно у схему включити пристрій для безпечного перемикання сигналу живлення. При цьому необхідно враховувати умови переключення контролерів, оскільки для різних задач існують різні варіанти переключення. Наприклад, у найпростіших задачах (наприклад, в системі вмикання/вимикання світла) перемикання можна здійснити за допомогою мікросхеми CD4030BE, яка по суті є звичайним логічним елементом «виключає АБО». На входи мікросхеми CD4030BE подаються дві напруги: перша – з джерела живлення, а друга – з п'ятивольтової лінії головного контролеру. У випадку, якщо мікроконтролер головної плати відмовить і перестане подавати живлення на п'ятивольтову лінію, то з вихідного контакту

мікросхеми CD4030BE буде ввімкнено лінію живлення на резервний контролер.

Також, як механізм перемикання, можна теж використовувати мікроконтролер, наприклад Attiny85, але при цьому необхідно враховувати час перемикання, який залежить від швидкості роботи мікроконтролера. У такому випадку до входів Attiny85 підключаються п'ятивольтова лінія з головного мікроконтролера та напруга з джерела живлення. В мікроконтролер прошивається програма, котра аналізує вхідні напруги, і у випадку, якщо відсутня напруга з головного контролеру, забезпечує подачу напруги на резервний.

Резервування джерел живлення забезпечується за допомогою спеціалізованих мікросхем LTC4412 або LTC4416, які відстежують обриви лінії живлення та здійснюють плавне перемикання джерел напруги. Дані схеми підключаються до двох джерел живлення, а саме основного та резервного, а виходи підключаються до вхідної напруги платформи Arduino. При цьому необхідно враховувати, що при перемиканні живлення виникає затримка, в цей період контролер буде неактивним, але при відновленні живлення почне працювати в штатному режимі.

Резервування каналів зв'язку забезпечується за допомогою використання як дротового так і бездротового з'єднання. Для бездротового з'єднання можна використовувати плати розширення RF, які працюють на частоті 433 МГц, ці плати представлені у вигляді двох окремих плат: одна працює у режимі передачі (TX), а інша у режимі прийому (RX). Ці плати підключаються до контактів послідовного порту TX ->TX, RX->RX.

Для реалізації резерву необхідно дві плати Arduino UNO, перша з них виступає у ролі передавача інформації, до неї підключаються всі сенсори системи та модуль TX для бездротової передачі по послідовному порту. Друга плата Arduino підключається до керуючого комп'ютера та приймає дані з сенсорів через модуль RX. Додатково, для підвищення надійності, обидві плати Arduino поєднуються фізичним середовищем.

При такому підключенні, якщо виникне проблема з обривом дроту, то передача даних продовжиться по резервному бездротовому каналу зв'язку. І навпаки, якщо зникне

бездротовий зв'язок, то передача продовжиться по дротовому з'єднанню.

Для резервування лінії передачі найпростішим способом є дублювання послідовного порту. При реалізації такого способу використовуються дві плати Arduino та по чотири контакти на кожній з них. Для цього необхідно програмно створити додатковий послідовний порт на платах Arduino UNO. При фізичному розміщенні системи у приміщенні всі канали зв'язку прокладаються окремо один від іншого, що підвищує надійність при виході з ладу одного з каналів зв'язку.

Результати та їх обговорення.

Система повинна працювати як можна більше часу без відмови, для цього знайдемо основні показники надійності. Показники надійності:

а) Показник безвідмовності – властивість об'єкта безупинно зберігати працездатність протягом певного часу (напрацювання). Для початку розрахуємо час безвідмовної роботи для кожного компонента окремо [9]:

- 1) МК Arduino UNO – 175200 год.
- 2) Сенсор температури та вологості DHT-11 – 80000 год.
- 3) Сенсор диму MQ2 – 60000 год.
- 4) Сенсор освітлення ВН1750 – 65000 год.

- 5) Сенсор руху HS-SR501 – 70000 год.
- 6) Реле – 100000 год.

б) Інтенсивність відмов i -го елемента визначається за формулою (1):

$$\lambda_i = \frac{1}{T_i} \quad (1)$$

$$\lambda_{МК} = \frac{1}{T_{МК}} = \frac{1}{175200} = 5,7 * 10^{-6}$$

(1/год.)

$$\lambda_{dht11} = 12,5 * 10^{-6} \text{ (1/год.)}$$

$$\lambda_{mq2} = 16,7 * 10^{-6} \text{ (1/год.)}$$

$$\lambda_{bh1750} = 15,3 * 10^{-6} \text{ (1/год.)}$$

$$\lambda_{sr501} = 14,2 * 10^{-6} \text{ (1/год.)}$$

$$\lambda_{реле} = 10 * 10^{-6} \text{ (1/год.)}$$

в) Імовірність безвідмовної роботи одного елемента системи по експонентному закону розподілу часу, визначається:

$$P_i(t) = e^{-\lambda_i t} \quad (2)$$

де λ_i – інтенсивність відмов i -го елемента, 1/год; t – час, який повинна працювати система (для розрахунку $t=10000$ годин).

З розрахунків отримуємо:

$$P_{МК}(t) = e^{-5,7 * 10^{-6} * 10000} = 0,9445$$

$$P_{dht11}(t) = 0,8824$$

$$P_{mq2}(t) = 0,8461$$

$$P_{bh1750}(t) = 0,8581$$

$$P_{sr501}(t) = 0,8676$$

$$P_{реле}(t) = 0,9048$$

Розрахуємо показники надійності для системи в цілому.

Згідно графу надійності, імовірність безвідмовної роботи системи з послідовною структурою елементів упродовж 10000 годин визначається за формулою (3):

$$P_{системи} = \prod_{i=1}^n P_i(t) \quad (3)$$

В результаті отримуємо:

$$P_{системи} = 0,4297$$

Інтенсивність відмов системи:

$$\lambda_{системи} = -\frac{\ln 0,43}{10000} = 8,4 * 10^{-5} \text{ (1/год.)}$$

Середнє напрацювання на відмову системи, визначається за наступною формулою (4):

$$T_{системи} = \frac{1}{\lambda_{системи}} \quad (4)$$

З розрахунків отримуємо, що $T_{системи} = 11905$ годин. Виходячи з того, що система працює 24 години 365 днів у році, отримуємо середнє напрацювання на відмову у роках:

$$T_{системи / роки} = \frac{11905}{24 * 365} \approx 1,35 \text{ роки.}$$

Виходячи з отриманих даних можна побачити, що очікуваний час напрацювання на відмову складає 1,35 роки, тому необхідно розглянути способи підвищення надійності даної системи.

Розрахуємо показники надійності для «гарячого» резерву. Для початку розрахуємо показники надійності тільки для мікроконтролеру, а потім для всієї системи. На рисунку 2 зображено граф надійності для «гарячого» резерву двох мікроконтролерів.

Вірогідність безвідмовної роботи «гарячого» резерву на лінії АВ визначається за формулою (5):

$$P_{AB} = 1 - (1 - P_{MK1}) * (1 - P_{MK2}) \quad (5)$$

Згідно з формулою (5) та значеннями отриманими раніше отримуємо $P_{AB} = 0,997$. Підрахуємо інші параметри згідно формулам 3 та 4, в результаті чого отримуємо показник інтенсивності відмови системи та середнє напрацювання на відмову у роках:

$$\lambda_c = 1 * 10^{-6} \text{ 1/год.}$$

$$T_{річне} = 101.47 \text{ років.}$$

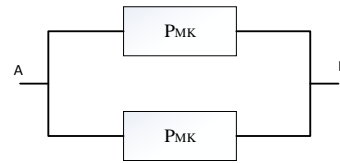


Рисунок 2 – Граф надійності «гарячого» резерву двох мікроконтролерів на умовній лінії зв'язку АВ

Тепер розрахуємо надійність для всієї системи. На рисунку 3 зображено граф надійності для системи з «гарячим» резервом. Підрахуємо параметри згідно формулам 2 та 3, та отримуємо:

$$P_{системи} = 0,4536$$

$$\lambda_{системи} = 7,9 * 10^{-5} \text{ 1/год.}$$

$$T_{системи} = 12658 \text{ год.}$$

$$T_{річне} = 2,44 \text{ років.}$$

На рисунку 4 наведено графіки, які показують ефективність використання «гарячого» резерву в системі.

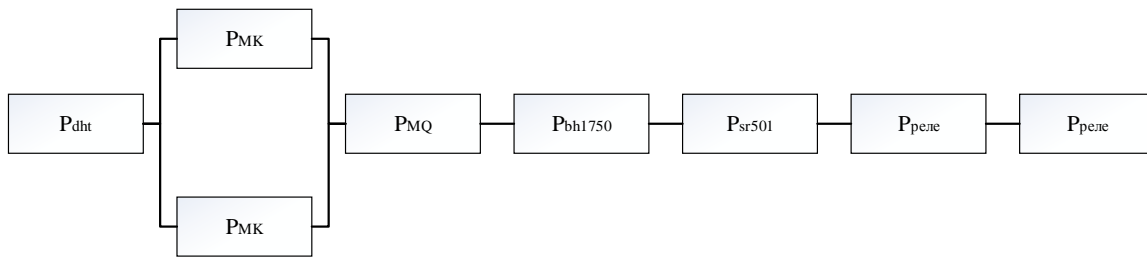


Рисунок 3 – Граф надійності для системи з «гарячим» резервом

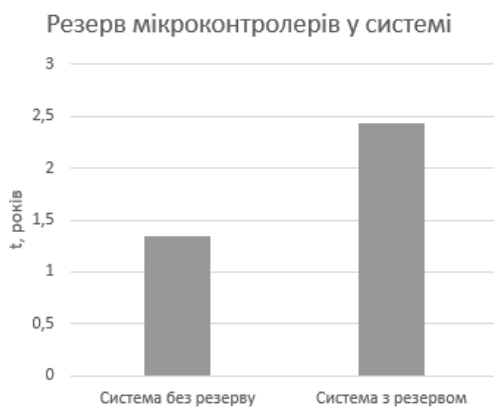


Рисунок 4 – Графіки ефективності використання «гарячого» резерву мікроконтролерів у системі «розумний будинок»

Висновки. В даній роботі наведено архітектуру для побудови системи «розумний будинок», в якій плата Arduino є посередником між сенсорами та комп'ютером. При використанні даної архітектури можливо

здійснювати віддалений контроль «розумного будинку» за допомогою смартфона чи планшету, який підключається до веб-застосунку, встановленому на керуючому комп'ютері. Дані в системі передаються у форматі JSON, це дозволяє розробити API для віддаленого керування та відстеження параметрів системи.

Модель запропонованої системи протестовано у середовищі моделювання Proteus VSM, для контролеру розроблено «прошивку», яка забезпечує функціонування системи.

Також було розглянуто задачу підвищення показників надійності системи, зокрема, вірогідності безвідмовної роботи. Запропоновано декілька способів підвищення надійності розробленої системи а саме: гарячий, холодний та теплий резерв контролерів; резерв ліній живлення; резерв ліній передачі даних.

Використання «гарячого», «теплого» та «холодного» резервування контролерів для

системи «розумний будинок» необхідно тоді, коли здійснюється віддалене керування системою. При виникненні проблеми з головним контролером системи, резервний вступає у дію та забезпечує безперебійне функціонування системи.

При використанні «гарячого» резервування система має досить високу надійність та швидкість роботи, оскільки при відключенні одного з контролерів передача даних продовжиться. Але при гарячому резервуванні обидва контролери знаходяться в активному стані, що негативно впливає на енергоефективність системи.

При використанні «теплого» резерву ситуація приблизно така сама, як і при

«гарячому», за винятком того, що при вимкненні головного контролера відбудеться затримка, яка необхідна для ввімкнення резервного контролера.

«Холодний» резерв забезпечує оптимальну енергоефективність та надійність системи, але більш складний у реалізації, оскільки має додаткові елементи, які здійснюють перемикання.

Для досягнення найбільшої ефективності можна використовувати всі наведені способи підвищення надійності як окремо, так і їх комбінацію.

Розроблена модель системи протестована та придатна для застосування у реальних умовах.

Література

- [1] AVR [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/AVR>
- [2] Arduino [Електронний ресурс] – Режим доступу: <https://www.arduino.cc/>
- [3] Розумний дім [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Home_automation
- [4] Компоненты системы умный дом своими руками [Електронний ресурс] – Режим доступу: <http://dom-automation.ru/umnyj-dom/articles/komponenty-sistemy-umnyj-dom-svoimi-rukami.html>
- [5] Рівні розумного будинку [Електронний ресурс] – Режим доступу: <https://erc.ua/erc-reviews/19643/rivni-rozumnogo-budinku/>
- [6] SMART Sense – Revogi Innovation [Електронний ресурс] – Режим доступу: <https://www.revogi.com/product/smart-sense/>
- [7] Xiaomi Smart Home [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Xiaomi_Smart_Home
- [8] HomeKit smart home device | FIBARO [Електронний ресурс] – Режим доступу: <https://www.fibaro.com/en/homekit/>
- [9] Азарсков В. М. Надежность систем управления и автоматики. Учебное пособие / В. Н. Азарсков, В. П. Стрельников — К.: НАУ, 2004. — 164с. ISBN 966-598-173-0
- [10] Способи підвищення надійності системи "розумний будинок" на платформі Arduino Nano / А.О. Федько, І.Я. Зеленьова, М.Ю. Тягунова, А.В. Тіменко, О.О. Бугасв // Міжнар. наук.-практ. конф.: «Technical sciences: history, the present time, the future, EU experience», 27-28 september, 2019: тези доп., - Wlowlawek, Republic of Poland, 2019. – Р. 49-52.

References

- [1] AVR. Available at: <https://ru.wikipedia.org/wiki/AVR>
- [2] Arduino. Available at: <https://www.arduino.cc/>
- [3] Rozumnij dim. Available at: https://en.wikipedia.org/wiki/Home_automation
- [4] Komponenty sistemy umnyj dom svoimi rukami. Available at: <http://dom-automation.ru/umnyj-dom/articles/komponenty-sistemy-umnyj-dom-svoimi-rukami.html>
- [5] Rivni rozumnogo budinku. Available at: <https://erc.ua/erc-reviews/19643/rivni-rozumnogo-budinku/>
- [6] SMART Sense – Revogi Innovation. Available at: <https://www.revogi.com/product/smart-sense/>
- [7] Xiaomi Smart Home. Available at: https://en.wikipedia.org/wiki/Xiaomi_Smart_Home
- [8] HomeKit smart home device | FIBARO. Available at: <https://www.fibaro.com/en/homekit/>

- [9] Azarskov V. M. Nadezhnost sistem upravleniya i avtomatiki. Uchebnoe posobie / V. N. Azarskov, V. P. Strelnikov — K.: NAU, 2004. — 164s. ISBN 966-598-173-0
- [10] Sposobi pidvishennya nadijnosti sistemi "rozumnij budinok" na platformi Arduino Nano / A. Fedko, I. Zelenova, M. Tyagunova, A. Timenko, O. Bugayev // Mizhnar. nauk.-prakt. konf.: «Technical sciences: history, the present time, the future, EU experience», 27-28 september, 2019: tezi dop., - Wlowlawek, Republic of Poland, 2019. – P. 49-52.

Надійшла до редакції 20.12.2019 р.

Федько Антон Олексійович – магістр кафедри комп'ютерних систем та мереж, Національний університет «Запорізька політехніка» (вулиця Жуковського, 64, Запоріжжя, Запорізька область, 69061)

E-mail: fedkoanton.a@gmail.com

Зеленьова Ірина Яківна – кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, Національний університет «Запорізька політехніка» (вулиця Жуковського, 64, Запоріжжя, Запорізька область, 69061)

E-mail: irina.zeleneva@gmail.com

Тягунова Марія Юрїївна – кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, Національний університет «Запорізька політехніка» (вулиця Жуковського, 64, Запоріжжя, Запорізька область, 69061)

Email: mary.tyagunova@gmail.com

DEVELOPMENT AND MODELING OF A SMART HOME SYSTEM WITH INCREASED RELIABILITY BASED ON ARDUINO

The article describes the architecture for designing a Smart home system in which the Arduino board is the connecting node between the sensors and the computer. Using this architecture, you can remotely control a Smart home using a smartphone or notepad – these gadgets are connected to the web service installed on the user's computer. Data in the system is transmitted in JSON format, this allows user to develop an API for remote control and monitoring of system parameters. To simulate and test the proposed system, the Proteus VSM was used. The necessary software for the system controller has been developed.

The problem of increasing the reliability of the system, namely the probability of failure-free operation, was also considered. Several methods have been proposed to increase reliability by redundancy. Hot and cold redundancy of controllers, redundancy of power supply lines, as well as data transmission lines are described. When using hot standby, the system provides a sufficiently high reliability and speed, since data transfer will continue even if one of the controllers fails. However, with this method, both controllers are turned on and require electricity.

The cold redundant system provides high energy efficiency and reliability, but the implementation of such a system is more complicated due to the additional elements for switching the reserve. All the described redundancy methods can be used both separately and in combination, which can increase the effect. The developed model of the Smart home system is tested and suitable for use.

Keywords: *microcontroller, Arduino, Proteus VSM, client application, increased reliability*

Fedko Anton – Master of Department of Computer Systems and Networks, National University «Zaporizhzhya Polytechnic» (Zhukovsky Street, 64, Zaporizhzhia, Zaporizhzhia Region, 69061).
Email: fedkoanton.a@gmail.com

Zelenova Irina – PhD, Associate Professor of Department of Computer Systems and Networks, National University «Zaporizhzhya Polytechnic» (Zhukovsky Street, 64, Zaporizhzhia, Zaporizhzhia Region, 69061)

Email: irina.zeleneva@gmail.com

Tyagunova Marya – PhD, Associate Professor of Department of Computer Systems and Networks, National University «Zaporizhzhya Polytechnic» (Zhukovsky Street, 64, Zaporizhzhia, Zaporizhzhia Region, 69061)

Email: mary.tyagunova@gmail.com

ABOUT FEATURES OF MANAGEMENT PREPRODUCTION OF ELECTRONIC VEHICLES

Victoria Smolij

As a result of undertaken studies new conception of management is worked out by designer preproduction of electronic vehicles, leaning against single informative space of arrangement of electronic vehicle, operative management designer preproduction and control system by the resources of enterprise, allowing to promote management efficiency designer preproduction of electronic vehicles.

Key words: *designer preproduction, electronic vehicle, management conception, informative space, arrangement of electronic vehicle, operative management, resources of enterprise, system of support of making decision.*

Introduction. The offered conception of management embraces the next stages of creation of electronic vehicle designer preproduction of electronic vehicles: arrangement, constructing, preproduction, test and directly production of preproduction model of good [1, 3 - 5].

Analysis of recent research and publications. The feature of the offered conception of management of designer preproduction of electronic vehicles is that, leaning against single informative space of arrangement of electronic vehicle, operative management designer preproduction and control system by the resources of enterprise, a cost and prime price of preproduction model of electronic vehicle cutout is arrived at; reduction of terms of producing of new electronic vehicles; the competitiveness of enterprise rises at upgrading of electronic vehicles, reliability, oscillation and resonant stability [2, 6, 14].

An achievement such of results is maybe by means of application of the developed system of support of making decision, realizing a management the systems of arrangement and designer preproduction and their cooperation in single informative space, that changes maintenance of designer preproduction in a root, orients a management arrangement of electronic vehicle on the resources of enterprise, providing management efficiency designer preproduction on the whole [5, 8, 16].

Purpose of research. For an enterprise - producer of electronic vehicles there is providing of management efficiency, including minimization of material production inputs new good or good with the parameters of quality and reliability, excelling analogues, reduction of duration of process of production etc., maybe by

the decision of management task exactly designer preproduction by means of choice of arrangement, parameters and properties of producible block of electronic vehicle (exactly as the completed structurally executed good).

Materials and methods of research. For realization of management conception it is necessary designer preproduction of electronic vehicles without the changes of technical equipped of production on the whole to modernize control system only. It is necessary to work out the system of support of making decision, providing with a managing personnel facilities, methods and instruments, providing possibility to realize the offered conception of management [7 – 11, 15].

Results and Discussion. The conceptual model of the automated management of designer preproduction of electronic vehicles looks like, brought around to a fig. 1. The designer preproduction implies producing of preproduction model of good with the parameters of quality, excelling analogues on condition of cost and unit cost cut out and reduction of terms of producing of good [16 - 19].

By analogy with existent classification of CASS of planning, making, tests of and other, the functions of components of the developed system of support of making decision of designer preproduction of electronic vehicles are analogical to the functions of ERP-, MES-, PDM-systems, applied on the enterprises of instrument-making industry [12, 20]. Application of the functions realized in the developed system of support of making decision analogical to the functions executable PDM - by the system, provides integration of data about arrangement of electronic vehicle, got as a result of planning and design. The constituent of the

developed system of support of making decision of designer preproduction, realizing the functions of PDM, are the systems, executes a management data about electronic vehicles, including the design of electronic vehicle as an

object of designer preproduction and management by arrangement of electronic vehicle [13 - 15].

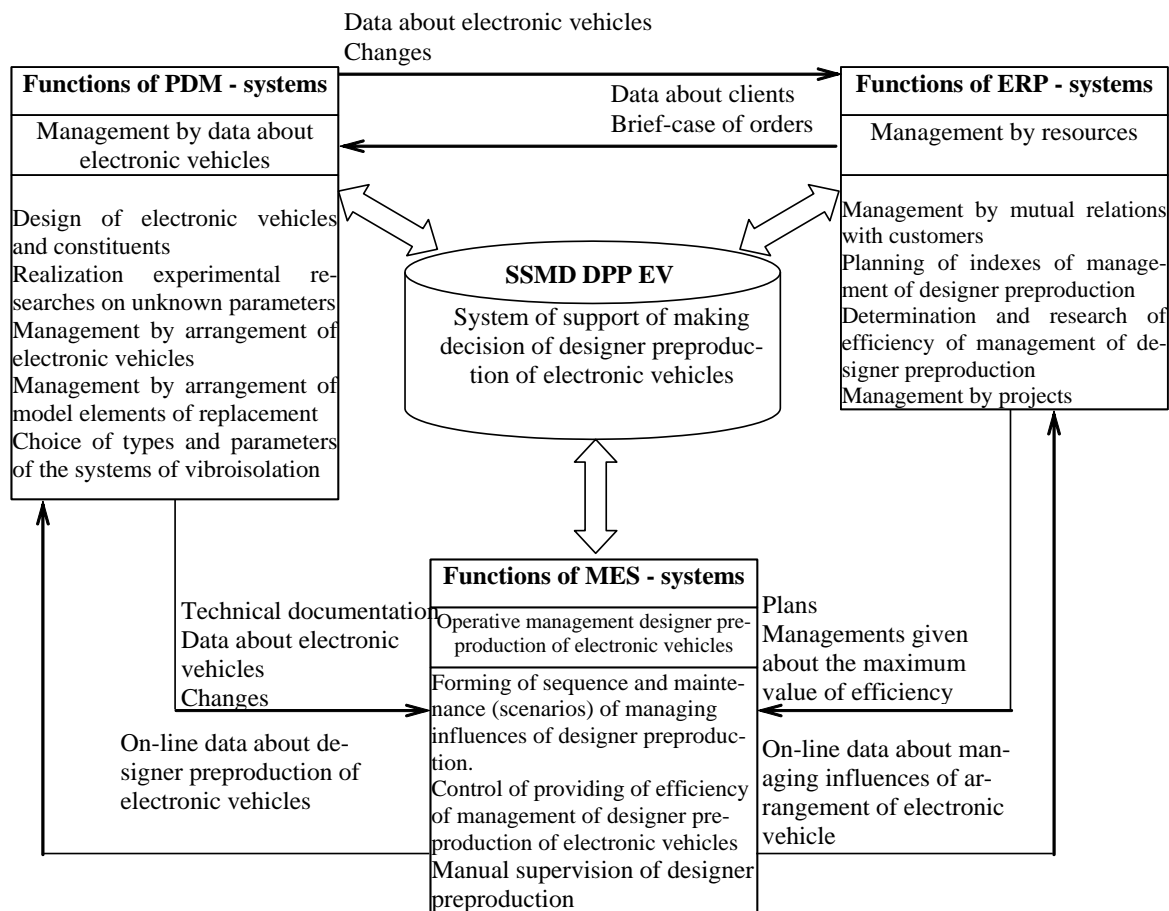


Fig. 1 - Conceptual model of the automated management of designer preproduction of electronic vehicles

Functions of MES - the systems in the developed system of support of making decision of designer preproduction of electronic vehicles suppose implementation of operative management designer preproduction of electronic vehicles on the basis of generated by the system of support of making decision of management scenarios realizing optimal maintenance and sequences of operations of arrangement on condition of observance of condition of management efficiency designer preproduction on the whole [14].

The constituent of the developed system of support of making decision of designer preproduction, realizing the functions of ERP, are the systems, organizes interrelation with the customer of company-producer of electronic vehicles, management by the developed projects and determines the possible size of management

efficiency designer preproduction of electronic vehicles, that will allow to the company to produce competitive good, save the markets of sale and get a profit at minimum material production inputs preproduction model of good.

The developed system of support of making decision of designer preproduction of electronic vehicles supposes an association and active cooperation of functions of the examined systems with the purpose of achievement of the required properties, quality, reliability, to resonant and oscillation stability of electronic vehicles on condition of minimization materially - technical production inputs, achievement of cost expended on retraining and in-plant training of personnel effectiveness, expenses on consultative services of experts etc.

The system of support of making decision designer preproduction of electronic vehicles

supposes implementation of functions of design of electronic vehicle taking into account setting and supposed external environments with the purpose of receipt of great number of effective decisions on a management by arrangement of electronic vehicle for the achievement of the required parameters of quality, reliability, oscillation and resonant stability of electronic vehicles. Realization of similar arrangement is possible by the managing affecting of designer preproduction, providing the achievement of financial viability of production of preproduction model of good, exception of tests, returns, revision of good, reduction of time, material production inputs and formalizations of knowledge and experience of experts.

In the conceptual model of the automated management the designer preproduction of electronic vehicles is distinguish next basic essence: management efficiency designer preproduction, electronic vehicle, model element of replacement. Every type of essence at the

construction of conceptual model appears as a separate rectangle with the name inwardly, thus the dependent types of essences are represented in a double scope.

The attributes of essence appear as ellipses with the name of the attributes connected by a continuous line with corresponding essence (or by a relation). Every type of relation is shown as a rhombus with the name of relation inwardly. Thus a rhombus is surrounded by a double line, if a relation is set between a dependent type to essence, from existence of that it is in dependence. The separate elements of diagram unite continuous lines (determined raising of management task designer preproduction of electronic vehicles) or dotted lines (stochastic raising of management task). Because connections (relations) of corresponding types of essence are not binary, therefore they are connected by no directional ribs.

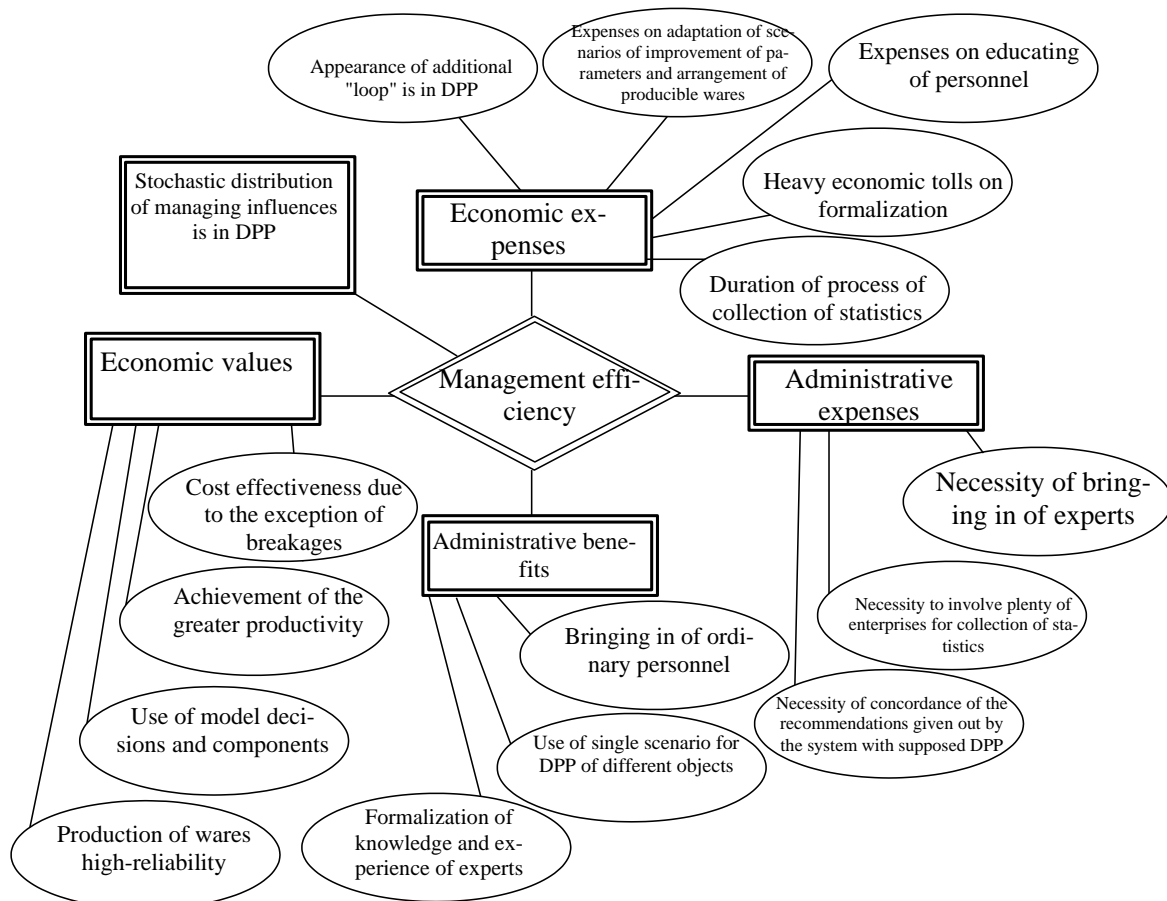


Fig. 2 - Conceptual model of management efficiency designer preproduction of electronic vehicle

It should be noted that for a management the designer preproduction of electronic vehicle is examine the technical and economic indexes

of management (management efficiency) and stochastically up-diffused selections of managing influences, divided by teaching and

verification for formalization of management process.

The conceptual model of management efficiency is brought designer preproduction of electronic vehicle around to a fig. 2. On a fig. 2 the attributes of essence are not shown, because we deal with a management designer preproduction of electronic vehicle a few levels of organization are distinguished in that, each of that includes elements from different levels, therefore in detail relations are explained in description to the conceptual model of the system of support of making decision.

Here it should be noted that for a management the designer preproduction of electronic vehicle is examine the technical and economic indexes of management (management efficiency) and stochastically up-diffused selections of managing influences, divided by teaching and verification for formalization of management process.

Management process implementation of certain sequence and maintenance of the managing influences sent both to arrangement of electronic vehicle implies designer preproduction of electronic vehicles and operative management by a personnel and resources of enterprise.

Thus, both for modern firms - producers of domestic electronic technique and for scientific and production enterprises airplane and rocket productions, a management process has identical procedure leaning against subjective presentations and skills of personnel in this area designer preproduction of pre-production model of good.

The unstructured and semistructured multicriterion tasks decide by means of the developed system of support of making decision of designer preproduction of electronic vehicles. The decision of the examined multicriterion tasks will allow not only to improve quality, reliability, resonant and oscillation stability of electronic vehicles, but also to provide financial viability of production of pre-production model of good, eliminate tests, returns on the revision of good, to shorten time, material production inputs and formalized knowledge and experience of experts [15].

In the examined automated management [16] subsystems are distinguished: creations of preliminary character of good, implementation of design of object of designer preproduction, realizations of experimental researches on authentication and research of failing

parameters, estimation of quality, reliability, oscillation and resonant stability of electronic vehicle, estimations of management efficiency designer preproduction of electronic vehicles, system of support of making decision, including the subsystems of receipt and treatment of knowledge.

It should be noted that each of them is fully independent and universal for the different sort of tasks, and also each of subsystems is characterized the high degree of noninteraction, that allows to realize the different variants of the parallel including of components of the system and organize an interface for a few users with the simultaneous processing of data.

Examining the computer integrated production of electronic vehicles, characterized by the high degree of automation and application of computer-aided designs, it should be noted that, the object of designer preproduction is characterized repetition, concurrent execution of operations of arrangement and constructing, difficult internal organization.

On an electronic vehicle as an object of designer preproduction is laid on row of requirements on the parameters of quality, reliability, oscillation and resonant stability, it is here necessary to execute the requirements of financial viability and management efficiency designer preproduction, characteristic for the electronic vehicles of the different setting and external environments. All requirements, touching both arrangement and properties of electronic vehicle and management efficiency designer preproduction of electronic vehicle, must be realized in developed system of support of making decision [15].

It should be noted presence and variants of differentiation of the developed system of support of making decision and degree of her use for different on purpose and to composition of objects of designer preproduction.

In the process of functioning the system of support of making decision carries out:

- Ground of necessity of initializing of management process;
- Task of type of object;
- Choice of setting of object;
- Analysis of external environments;
- Prognostication of the possible states of object by means of probabilistic criterion leaning against the base of knowledge and rule of support of making decision;
- Maps the state of object to his components and vice versa;

- Offers methodologies of upgrading, reliability, resonant and oscillation stability of electronic vehicle by means of the offered criteria of arrangement of block of electronic vehicle and constituents;
- Provides the selection of instruments of mathematical design;
- Contains explanations about that, how to work with the instruments of mathematical design;
- Identifies the unknown parameters of mathematical models by realization of experimental researches on a corresponding chart, methodology of the programmatic and technical providing of experiments;
- By means of the worked out criterion of management quality the designer preproduction of electronic vehicles (management efficiency) is watch rationality made alteration in a management designer preproduction of electronic vehicle and their influence is determined on the technical and economic indexes of management on the whole;
- Provides subsequent treatment of control program for a technological equipment, tuned to the same not only under a corresponding technological equipment but also under quality parameters, reliability, oscillation and resonant stability of electronic vehicle.

On the other hand this single cycle of management is mapped to some association of making electronic vehicles, subject to strict hierarchicalness, being with each other in different conceptual relations. It does not allow to talk about possibility of successive implementation of each of operations above some certain object, and supposes the presence of possibility of application of separate operation on the package (great number) of objects or great number of operations on one object, not except possibility of the simultaneous processing of great number of objects.

In the offered conception of management it is assumed designer preproduction, that the type of object for that make an electronic vehicle determines not only external environments but also possibility of bringing of modifications in a management his designer preproduction (management effect), assuming variations of technical and economic indexes and some variations on speed implementations of designer preproduction. The base of knowledge and mechanism of receipt of decision are related to

setting of object and object of exploitation, determined by the probabilistic criterion of management efficiency, that systematizes knowledge of experts and experience of specialists engaging in the problem of upgrading and reliability of electronic vehicles.

The variants of terms are possible exploitations that is taken into account in the mathematical model of object, that gives universality in-process designer preproduction the offered management. There is transformation of indexes of quality, reliability, to oscillation and resonant stability of electronic vehicle depending on the results of design on the basis of criteria of arrangement of object of designer preproduction. At changing of mathematical model of electronic vehicle management methodology is applicable designer preproduction either for other external environments or for other objects.

The hierarchy of the prospected objects of designer preproduction is characterized encapsulation, when the elements of subsequent levels consist of great number of elements of previous levels. Such the organization is possible for different objects, what universality of the offered algorithm allows to mark and to reflect his principles on a management by other systems.

Requirements to instrumental part are conditioned by realization of mathematical model as a complex of programmatic and technical facilities, supporting a receipt and transformation of informative character of electronic vehicle in accordance with the criteria of his arrangement. Also the distinguishing feature of instrumental part is methodology plugging in the arsenal of tools of the experimental setting equipped by rigging for authentication of failing parameters of object of designer preproduction.

A return to the management is needed designer preproduction, because virtual building on in form system of support of making decision, including designing complexes, brings in some changes in a structure and parameters of electronic vehicle, that must be passed on a management by designer preproduction and to watch rationality made alteration on the criterion of management quality designer preproduction of electronic vehicles (management effect).

A single algorithm taking into account all these facts differs in large universality and flexibility, because allows to reform on control system designer preproduction of any electronic

vehicles without depending on their informative filling.

In respect of task of type of object, then it the simplified enough classification and scientific and practical interest here present not classification, and standing after it components of probabilistic criterion of management efficiency. The values of components of probabilistic criterion are identified in the process of educating of the system of support of making decision determine, take into account and visualize the base of knowledge of probabilistic descriptions of frequencies of refuses, reasons of refuses, influencing active factors and correlation of technical and economic parameters for a management designer preproduction for the certain type of electronic vehicle and external environments.

The automated management designer preproduction of electronic vehicles is based mainly on dialogue instructions acting on behalf of accepting decision. In order that realization of such possibility took place, it is necessary to supply a person, a decision-making, by instruments, allowing to get, process and analyze information, and mechanism, for preparation of decision-making. To that end it is necessary to build the row of informative, programmatic, technical and intellectual subsystems of the developed system of support of making decision of designer preproduction of electronic vehicles.

Management process implementation of certain sequence and maintenance of the managing influences sent both to arrangement of electronic vehicle implies designer preproduction of electronic vehicles and operative management by a personnel and resources of enterprise. Thus, both for the modern firms of producers of domestic electronic technique and for scientific and productive enterprises airplane and rocket productions, a management process has identical procedure leaning against subjective presentations and skills of personnel in this area designer preproduction of pre-production model of good [16, 19, 20].

The unstructured and semi structured multicriterion tasks decide by means of the developed system of support of making decision of designer preproduction of electronic vehicles. The decision of the examined multicriterion tasks will allow not only to improve quality, reliability, resonant and oscillation stability of electronic vehicles, but also to provide financial viability of production of pre-production model of good, eliminate tests, returns on the revision

of good, to shorten time, material production inputs and to build an algorithm knowledge and experience of experts.

Integration of the developed system of support of making decision is arrived at by implementation of row of functions of the systems of production of electronic vehicles, presenting the results of modulating, arrangement and management designer preproduction as data, mechanisms of processing of data and receipt of knowledge making the base of knowledge of the system of support of making decision.

The managing affecting process of designer preproduction acts from outside, in particular on behalf of decision making, in order to ratify the offered scenarios of managing influences of arrangement of electronic vehicle, corresponding to the condition of management efficiency designer preproduction. In a fundamental chart the variant of the use of the developed system of support of making decision is envisaged also for the design of electronic vehicles of the different setting and external environments with the use of the experimental setting for research of failing parameters or additional research of electronic vehicle and constituents.

The offered fundamental chart of the system of support of making decision of designer preproduction of electronic vehicles will realize the mechanism of educating of the system of support of making decision, sent to adaptation of scenarios of achievement of necessary parameters, properties and arrangement of electronic vehicle under the existent terms of financial viability of production and corresponding technical and economic indexes of making of pre-production model of electronic vehicle.

Facial, decision making as the making developed system of support of making decision, it is necessary to distinguish the row of functions:

- preparation of information;
- input of information;
- implementation of design;
- preparation and input of information in the system of support of decision-making;
- analysis of the results got from the system of support of making decision;
- implementation of directive management directly designer preproduction of electronic vehicle, including of necessity executions of the given out recommendations.

For the decision of existent productive situations it is necessary formally to describe the actions of face of decision making, consisting in that his activity is presented by the task of great numbers of requests for implementation: designs, arrangements, operative management designer preproduction, research of management efficiency [14, 17 - 20].

Query on behalf of decision making can touch implementation of design, task of preliminary variant of arrangement, research of resources of enterprise on making of pre-production model of good and their variation.

Person an accepting decision, leaning against subjective knowledge and experience, can obtain the certain indexes of efficiency for some period of time and amount of heuristic iterations, however the developed system of support of making decision, leaning against the mechanisms of educating and multicriterion optimization, allows to bring down expenses and prime price of pre-production model of electronic vehicle; to reduce the terms of producing of new electronic vehicles; to promote the competitiveness of enterprise at upgrading of electronic vehicles, reliability, oscillation and resonant stability; to economize the facilities expended on retraining and in-plant training of personnel, expenses on consultative services of experts etc.

On results a design the system of support of making decision forms the components of criteria of arrangement of electronic vehicles, on results processing of statistical data are criteria of arrangement, on the basis of questioning of experts, from literary sources and statistics are management scenarios designer preproduction, coming from the analysis of technical and economic, skilled and other of indexes of production of electronic vehicles is management efficiency. Synthesizes all these indexes, the system of support of making decision prospects and processes, preparing information facial decision making [14 - 16].

Except the functions of search of optimal scenarios of achievement of the required parameters of quality, reliability, to oscillation and resonant stability of electronic vehicle in the process of designer preproduction the offered system of support of making decision, both produces variants and adapts existing before methodologies oscillation, shock and other variants of defense of electronic vehicles from external influences to the existent economic

terms and market mechanisms of development of production of electronic vehicles.

Similar family systematization is possible by means of statistical treatment of results of the expert questioning, formalization of procedure of management and forming of case frames for the certain types of electronic vehicles. Introduction of the offered innovations in the existent chart of management of designer preproduction of electronic vehicles will give an opportunity to tune the prospected management under the certain setting of the produced good and condition of exploitation, to attain optimal combinations of economic and administrative parameters of management designer preproduction of electronic vehicles.

For the achievement of the put aim it is necessary to organize questioning of experts concerning the estimations of priorities of results of application of the offered methodology of management of designer preproduction for different objects with subsequent statistical treatment of results of questioning. It is also necessary to estimate adequacy of the got experimental data, define priorities of parameters and their functional intercommunications and on them to produce the analysis of the got results with organization of feed-back for the estimation of efficiency of management of designer preproduction of electronic vehicles. As a method of decision of this task the method of analysis of hierarchies is applied.

Integration of the developed system of support of making decision is arrived at by implementation of row of functions of the systems of production of electronic vehicles, presenting the results of modulating, arrangement and management designer preproduction as data, mechanisms of processing of data and receipt of knowledge making the base of knowledge of the system of support of making decision [14 - 16].

The managing affecting process of designer preproduction acts from outside, in particular on behalf of decision making, in order to ratify the offered scenarios of managing influences of arrangement of electronic vehicle, corresponding to the condition of management efficiency designer preproduction. In a fundamental chart the variant of the use of the developed system of support of making decision is envisaged also for the design of electronic vehicles of the different setting and external environments with the use of the experimental setting for research of failing parameters or

additional research of electronic vehicle and constituents.

The offered fundamental chart of the system of support of making decision of designer preproduction of electronic vehicles will realize the mechanism of educating of the system of support of making decision, sent to adaptation of scenarios of achievement of necessary parameters, properties and arrangement of electronic vehicle under the existent terms of financial viability of production and corresponding technical and economic indexes of making of pre-production model of electronic vehicle.

Conclusions. New conception of management is worked out by designer preproduction of electronic vehicles, leaning against single informative space of arrangement of electronic vehicle, operative management designer preproduction and control system by the resources of enterprise, allowing to promote management efficiency designer preproduction of electronic vehicles.

First it offers to examine designer preproduction of electronic vehicles as system determined and stochastic constituents, allowing depending on setting of object of designer preproduction to determine management scenarios for the achievement of the required parameters and arrangement of electronic vehicles.

Decision of stochastic task of management the receipt of functional dependences supposes designer preproduction for research of management efficiency including quality and

amount of management cycles, allows to optimize the technical and economic and administrative indexes of management, that it is necessary to realize supports of making decision in the developed system.

The worked out conceptual model of the system of support of making decision, leaning against single informative space of arrangement of electronic vehicle, operative management designer preproduction and control system by the resources of enterprise, allows to provide the required properties, quality, reliability, resonant and oscillation stability of electronic vehicles on condition of minimization of material and technical production inputs, achievement of cost expended on retraining and on effectiveness.

Worked out informative and algorithmic providing of management, including the criteria of arrangement and management quality designer preproduction of electronic vehicles, designer preproduction of electronic vehicles, it is necessary to realize supports of making decision in the developed system.

Worked out algorithms of management designer preproduction, choice of types of objects and construction of model of electronic vehicle, including the stages arrangements, constructing, preproduction and tests of electronic vehicle, support intercommunication with the experimental setting with the corresponding rigging, allow to get failing information about the object of designer preproduction and estimate his parameters and arrangement without producing of pre-production model of good.

References

- [1] Aglietti G.S. Development of the MiniSIL™ Structural design/ G.S. Aglietti, A. Wicks, A.J.Barrington-Brown // Journal of Aerospace Engineering. - Vol 213 part G. - pp. 255-263. - 1999. ISSN 0954-4100.
- [2] Aglietti G.S. A Lighter Enclosure for Electronics for Space Applications/ G.S. Aglietti //Journal of Aerospace Engineering. - part G, Vol. 216-3. – pp. 131-142. 2002.
- [3] Basu K. Soft sets: an ordinal formulation of vagueness with some applications to the theory of choice/ K.Basu, R.Deb, P.K.Pattanaik // Fuzzy Sets and Systems. – №45. – P. 45 – 58. 1992.
- [4] Ivakhnenko A.G. Samoorganizatsyya of the forecasting systems / A.G. Ivakhnenko, I.F. Myuller. – Kyiv: Technique, 1985. – 223. 1985. (in Russian)
- [5] Jampolsky L.S. Flexible computerized systems: planning, design and management / L.S. Jampolsky, P.P. Melnychuk, B.B. Samotokin, M.M. Polishuk, M.M. Tkach, K.B. Ostapchenko, O.I. Lisovichenko.- Shytomyr: SNTU, - 680 P. + CD. 2005. (in Russian)
- [6] La Malfa S. Use of a Dynamic Absorber in the case of a Vibrating Printed Circuit Board of Complicated Boundary Shape / S.La Malfa, P.A.A.Laura, C.A.Rossit, O.Alvarez // Journal of Sound and Vibration. - Vol. 230(3). - pp.721-724. 2000.
- [7] Laura P.A.A. Dynamic Stiffening of a Printed Circuit Board / P.A.A.Laura, L.Ercoli, and.,//

- Acustica. - Vol. 81. – pp. 196-197. 1995.
- [8] Lim G.H. Effect of Edge and Internal Point Support of a Printed Circuit Board Under Vibration/ G.H.Lim, J.H.Ong, J.E.T.Penny // ASME Journal of Electronic Packaging. – Vol. 121, №2. - pp. 122-126. 1999.
- [9] Ong J.H. Simple Technique for Maximising the Fundamental Frequency of Vibrating Structures/ J.H. Ong, Lim G.H. // ASME Journal of Electronic Packaging. - No 4, Vol. 122. - pp. 341-349. 2000.
- [10] Royzman V. The dynamic effects and shocks in electronics / V. Royzman, E. Nester // Experience of designing and application of cad systems in microelectronics. 6th International Conference of CADSM 2001, FEB 12-17. – P. 256-259. 2001.
- [11] Saaty T. Decision-making. Method of analysis of hierarchies / T. Saaty. – M.: Radio and svyaz, 1993. – 320. 1993. (in Russian)
- [12] Saaty Thomas L. Eigenweightor an logarithmic lease squares/ Thomas L. Saaty // Eur. J. Oper. Res. — V. 48, № 1. - 156-160. 1990. (in Russian)
- [13] Saaty T.L. Multicriteria Decision Making. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation/ Thomas L. Saaty. – University of Pittsburgh, - 359. 1990. (in Russian)
- [14] Smolij V.N. Automation of processes of production of blocks of electronic vehicles: Monographija. – Lugansk: East Ukrainian National University named after V.Dal. – 124. 2006. ((in Ukrainian).
- [15] Smolij V.N. Case frame by production of electronic vehicle of military purpose // Scientific Papers of Donetsk National Technical University. Series «Informatics, Cybernetics and Computer Science» (ICCS - 2010). – Donetsk: DNTU. - Issue 11(164) – 188 - 193 . 2010. (in Russian)
- [16] Smolij V.N. Hierarchy of criteria in the operations management of electronic vehicles / V.N. Smolij // Praci Lugansk Branch of International Informatization Academy. – Lugansk: East Ukrainian National University named after V.Dal. – №1(21). - 64 - 69. 2010 (in Russian)
- [17] Steinberg D.S. Vibration Analysis for Electronic Equipment. - John Wiley & Sons. 2000.
- [18] Suhir E. 2000.: Predicted Fundamental Frequency of Vibration of a Heavy Electronic Component Mounted on a Printed Circuit Board / E. Suhir // ASME Journal of Electronic Packaging. – Vol. 122, No 1. - pp. 3-5.
- [19] Valiani A. Case Study: Malpasset Dam-Break Simulation using a Two-Dimensional Finite Volume Method/ A.Valiani, V.Caleffi, A.Zanni// Journal of Hydraulic Engineering. – May,– Vol.128, №. 5. – P. 460-472. 2002.
- [20] Ulshin V.A. Automated management by designer preparation of production of electronic vehicles/ Vitaly Ulshin, Victoria Smolij // TEKA Kom. Mot. I Energ. Roln. - 11A. - 276 – 281. 2011.
- [21] Ulshin V.A. Case-based reasoning method for diagnostic decision support system of bridge cranes/ Vitaly Ulshin, Sergey Klimchuk // TEKA Kom. Mot. I Energ. Roln. - 11A. - 266 – 275. 2011.
- [22] Wong T.-L. Experimental Modal Analysis and Dynamics Response Prediction of PC Boards With Surface Mounted Electronic Components / T.-L. Wong, K.K.Stevens, G. Wang // ASME Journal of Electronic Packaging. – Vol. 113. - pp. 244-249. 1999.

Надійшла до редакції 13.12.2019 р.

Smolij Victoria Nicolaevna – Doctor of Technical Sciences, Professor, Volodymyr Dahl East Ukrainian National University, (59 a, Central avenue, Severodonetsk, 93400, Ukraine).
E-mail: vmsmolij@ukr.net.

ПРО ОСОБЛИВОСТІ УПРАВЛІННЯ КОНСТРУКТОРСЬКОЮ ПІДГОТОВКОЮ ВИРОБЛЕННЯ ЕЛЕКТРОННИХ АПАРАТІВ

В результаті проведених досліджень розроблена концепція управління конструкторською підготовкою виробництва електронних апаратів, що спирається на єдиний інформаційний простір компоновання електронного апарату, оперативного управління конструкторською підготовкою виробництва і системи управління ресурсами підприємства, що дозволяє підвищити ефективність управління конструкторською підготовкою виробництва електронних апаратів.

Сформований комплекс технічних засобів для технологічного процесу виробництва блоків електронних апаратів, який дозволяє впровадити нові науково-дослідні результати у виробництво. Запропонований критерій якості технологічного процесу виробництва блоку електронного апарату, що зв'язує надійність виробу, що випускається, з його механічними, технічними і економічними параметрами.

Розроблені програмно-технічні комплекси для моделювання механічних навантажень блоків електронних апаратів і складових вирішують завдання моделювання електронного апарату як об'єкту конструкторської підготовки виробництва і дозволяють генерувати початкову інформацію для системи підтримки ухвалення рішень, що розробляється, і аналізувати рішення конструкторської підготовки виробництва електронних апаратів, що приймаються.

У детермінованій постановці рішення задачі синтезу електронного апарату з оптимальними вібраційними, тепловими і механічними параметрами передбачається шляхом застосування ітераційних алгоритмів, що по черзі відпрацьовують спочатку ієрархію складових електронних апаратів, потім відповідної головної групи чинників, що відповідають пріоритету, що відбиває умови експлуатації, а потім безпосередньо кожного з чинників. Стохастична постановка рішення задачі синтезу електронного апарату з оптимальними параметрами в площині техпроцесу його виробництва припускає рішення задачі оптимального управління у рамках адаптивної системи за критерієм компоновання, що зв'язує параметри електронного апарату і його складових з параметрами техпроцесу його виробництва. У детермінованій постановці рішення задачі синтезу електронного апарату з оптимальними вібраційними, тепловими і механічними параметрами передбачається шляхом застосування ітераційних алгоритмів, по черзі.

Провішив дослідження особливостей моделей опису технологічного процесу виробництва електронних апаратів, слід зазначити, що в детермінованій постановці необхідно відшукати оптимальні параметри електронного апарату, а в стохастичній - оптимальний сценарій досягнення необхідних властивостей в конкретному технологічному процесі виробництва.

Проведення експериментальних досліджень за оцінкою достовірності пропонованого методу припускає дослідження техпроцесів виробництва електронних апаратів конкретних підприємств галузі з виконанням їх порівняльного аналізу, впровадженню отриманих результатів і розробки рекомендацій і пропозицій по модернізації існуючих техпроцесів.

Сформульована мета роботи, що полягає в створенні методу синтезу електронних апаратів з оптимальними вібраційними, тепловими і механічними параметрами в детермінованій постановці і процесу проектування, що забезпечує оптимальність, в стохастичній постановці.

Ключові слова: конструкторська підготовка виробництва, електронний апарат, концепція управління, інформаційний простір, компоновка електронного апарату, оперативне управління, ресурси підприємства, система підтримки прийняття рішень.

Смолій Вікторія Миколаївна – доктор технічних наук, професор, Східноукраїнський національний університет ім.В.Даля (пр. Центральний, 59 а, м. Сєверодонецьк, 93400, Україна).
E-mail: vmsmolij@ukr.net.

УПРАВЛІННЯ ПАМ'ЯТТЮ І СТРУКТУРИ ДАНИХ В ОПЕРАЦІЙНИХ СИСТЕМАХ FREERTOS ТА PREDICATE OS

Гайдук К. С., Шевченко О. Г.

В роботі наведено результати дослідження та порівняльного аналізу алгоритмів динамічного розподілу пам'яті та базових структур даних, що використовуються у вбудованих операційних системах FreeRTOS і Predicate OS. Запропоновано поняття умовної верхньої межі часу виконання операції та коефіцієнту детермінованості, що дозволяють виконувати кількісну оцінку детермінованості роботи алгоритму. Дослідження виконувалися на базі платформи STM32 із застосуванням таймера високої роздільної здатності DWT. Отримані результати можуть бути корисними для розробників програмного забезпечення при виборі алгоритмів динамічного розподілу пам'яті та їх параметрів.

Ключові слова: алгоритми динамічного розподілу пам'яті, детермінізм алгоритму, динамічні списки, черги, накладні витрати ресурсів, операційні системи реального часу.

Вступ. Використання динамічного розподілу пам'яті (ДРП) у вбудованих системах реального часу (СРЧ) є обмеженим, оскільки знижує часовий детермінізм системи, обумовлює накладні витрати оперативної пам'яті та процесорного часу, збільшує час відгуку [1,2]. Проте, ДРП все частіше застосовується у вбудованих СРЧ, на що є кілька причин: 1 – поява алгоритмів ДРП зі сталим часом виконання; 2 – поява досить потужних апаратних платформ для вбудованих систем; 3 – використання ДРП наділяє програмне забезпечення більшою гнучкістю, ніж у разі статичного розподілу пам'яті. Прикладами операційних систем реального часу (ОСРЧ), що використовують ДРП, можуть слугувати такі, як RT-Linux GPL, QNX Neutrino, MARTE OS [1] та ін.

Окрім збільшення гнучкості програмного забезпечення, ДРП також опосередковано дозволяє досягти запобігання ресурсних взаємних блокувань [3, 4], що було використано в авторській операційній системі Predicate OS. Одним з ключових архітектурних елементів зазначеної ОС є поштові скриньки задач (черги повідомлень), реалізовані на базі динамічних списків, що робить розподільник пам'яті та черги повідомлень найбільш вузьким місцем системи в плані витрат процесорного часу і пам'яті. Для оцінки життєздатності використаної архітектури було прийнято рішення виконати порівняння Predicate OS з

такою популярною вбудованою операційною системою реального часу, як FreeRTOS [5].

Аналіз останніх досліджень та публікацій. Існує велика кількість алгоритмів ДРП [6-12], серед яких наявні як ті, що міцно увійшли в інженерну практику, так і ті, що представляють виключно теоретичний інтерес. Серед усіх алгоритмів ДРП лише деякі характеризуються сталим часом виділення та звільнення блоку пам'яті (прикладом можуть слугувати TLSF та Half-Fit), решта ж має час $O(n)$ для обох, або однієї з вказаних операцій [6, 8, 9, 11]. В [10] доводиться детермінованість за часом алгоритму DL malloc, який застосовується в багатьох операційних системах і реалізаціях стандартної бібліотеки stdlib мови програмування С, хоча він не знайшов широкого застосування в ОСРЧ.

Операційна система FreeRTOS надає п'ять схем розподілу пам'яті [13], перша з яких дозволяє лише виділяти пам'ять, і не підтримує функції звільнення блоків пам'яті. Інші чотири представлені алгоритмами Best-Fit, First-Fit, розподільником зі стандартної бібліотеки stdlib, а також модифікацією First-Fit, що дозволяє організовувати складову купу на базі кількох не суміжних областей пам'яті. Незважаючи на те, що FreeRTOS позиціонується як ОСРЧ, застосовані в ній алгоритми Best-Fit і First-Fit мають найгірший час виділення блоку, рівний $O(n)$, де під параметром n розуміється довжина списку вільних блоків.

Раніше згаданий алгоритм TLSF, використаний в Predicate OS, не підтримує функції динамічної зміни розміру купи $\text{sbrk}()$, однак це обмеження не представляється критичним для переважної більшості вбудованих систем.

Метою даної статті є дослідження і порівняльний аналіз алгоритмів динамічного розподілу пам'яті, а також базових структур даних, що використовуються в операційних системах FreeRTOS та Predicate OS.

Матеріали та методи. Для кількісної оцінки та порівняння алгоритмів ДРП використовувалися наступні показники: мінімальне, середнє і максимальне значення часу виконання операції; стандартне відхилення на основі зміщеної оцінки дисперсії; коефіцієнт коваріації. У разі дослідження операції виділення пам'яті в різних алгоритмах, при розрахунку вищезазначених показників бралися до уваги лише випадки вдалого виконання операції (функція повертає не нульовий показчик).

Незважаючи на те, що перелічені показники надають деяку характеристику алгоритму, вони є малоінформативними в контексті оцінки детермінізму, тому як в останньому випадку найбільший інтерес представляє розкид випадкової величини відносно деякої верхньої межі, а не відносно середнього значення. У зв'язку з цим в роботі введено поняття умовної верхньої межі часу виконання операції, а також коефіцієнта детермінованості, смисл і алгоритм розрахунку яких наведено нижче.

Нехай дано N результатів вимірювання часу виконання деякої операції. Множина даних відліків ділиться на рівні інтервали по d відліків в кожному (1):

$$d = \lfloor N/k \rfloor, \quad (1)$$

де k має порядок близько 10.

Очевидно, що кількість отриманих інтервалів m буде визначатися виразом (2):

$$m = \begin{cases} \lfloor N/k \rfloor, & \text{if } N \bmod k = 0 \\ \lfloor N/k \rfloor + 1, & \text{if } N \bmod k \neq 0, \end{cases} \quad (2)$$

Для кожного інтервалу знаходиться максимальне значення, що дає m локальних максимумів LM_i . Середнє арифметичне всіх локальних максимумів назовемо умовною верхньою межею часу виконання Th (3):

$$Th = \frac{1}{m} \sum_{i=1}^m LM_i. \quad (3)$$

Коефіцієнт детермінованості K_d розраховується як середнє квадратичне різниць $t_i - Th$ (4), де t_i - i -й відлік. При цьому враховуються лише ті точки, що лежать вище Th (рис. 1):

$$K_d = \frac{1}{r} \sum_{i=1}^N (t_i - Th)^2, \text{ if } t_i - Th > 0, \quad (4)$$

де r - кількість точок, для яких виконується умова $t_i - Th > 0$.

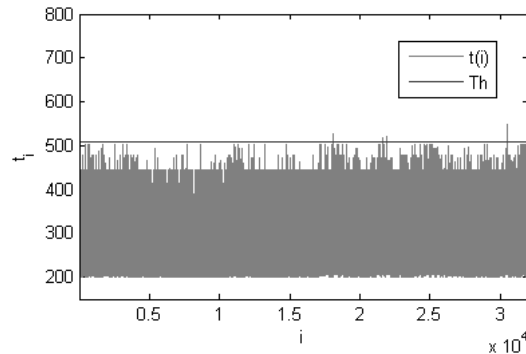


Рис. 1 – Умовна верхня межа часу виконання операції.

Для отримання кількісних характеристик різних алгоритмів ДРП було використано дві схеми проведення експериментів:

Схема 1. В межах одного кроку циклу виконується вісім запитів на виділення блоку пам'яті і стільки ж операцій звільнення. Порядок запитів на виділення і звільнення блоку пам'яті перемішаний [14].

Кількість ітерацій циклу у всіх експериментах постійна і становить 8000. Розмір купи H_s у всіх випадках дорівнює 15 Кб. Діапазон розмірів запитуваних блоків дорівнює $[0, \dots, H_s/Q - 1]$, де Q - параметр, який приймає значення 1 або 20, що відповідає двом окремим варіантам експерименту.

Функцію-перехоплювач невіддалого виділення пам'яті в FreeRTOS [2] відключено.

Схема 2. На кожній ітерації циклу виконується або випадкова кількість запитів на виділення блоку пам'яті, або випадкова кількість запитів на звільнення. Причому, запити на виділення виконуються частіше, ніж на звільнення [14]. Кількість ітерацій дорівнює 64000. Як і у разі першої схеми, проводяться окремі експерименти для різних значень параметра Q . В обох схемах ведеться підрахунок кількості відмов.

З метою більш ретельного аналізу деяких алгоритмів, використані спеціальні вставки у вихідному коді, що дозволяють

досягти «декомпозиції» одержуваних графіків і виконати їх інтерпретацію (рис. 7-8).

Операції виділення блоку пам'яті (`malloc()`) і звільнення (`free()`) є нересурсобезпечливими, і тому основний код відповідних функцій має виконуватися всередині критичних секцій (КС). Незважаючи на те, що обробники завдань-автоматів в Predicate OS [3] є атомарними і не можуть переривати виконання один одного, вони можуть перериватися обробниками апаратних переривань, тому організація КС актуальна також і для даної операційної системи.

З метою дослідження роботи розподільників із обробників апаратних переривань, використовувалися модифіковані варіанти вищеописаних схем: паралельно з циклом в основній програмі, що містить запити `malloc()` і `free()`, ті ж запити по черзі надходили також з обробника одного з апаратних таймерів системи [14].

Списки і черги. Як було зазначено раніше, в Predicate OS черги реалізовано на базі двозв'язних динамічних списків. Кожен елемент списку має наступні поля: покажчик на довільні дані; покажчики на попередній і наступний елементи; пріоритет (можлива організація черг з пріоритетами); крайній термін обробки; код повідомлення (події). Останні два поля актуальні для поштових скриньок задач. При створенні списку, вказується максимальна кількість елементів, яку він може містити (для створення списку з необмеженою кількістю елементів, необхідно вказати константу `MAX_LIST_SIZE`).

З тієї причини, що списки використовуються для організації не тільки черг без пріоритетів, а й черг з пріоритетами, створення списків на базі масивів (без виділення окремого блоку пам'яті під кожен елемент списку) є складним, і нераціональним в плані накладних витрат процесорного часу.

В FreeRTOS черги є безпріоритетними, і організовані на базі масивів. Черги суміщені із засобами синхронізації (двійкові семафори, рахункові семафори, м'ютекси або рекурсивні м'ютекси). Таким чином, черга може перебувати в блокованому або вільному стані. Ряд функцій для роботи з чергами мають параметр, що задає таймаут для виконання операції. В Predicate OS подібні механізми синхронізації не потрібні, через використання парадигми автоматного програмування [3, 15].

FreeRTOS не надає API для роботи зі списками [16], хоча використовує двозв'язний сортований (за спаданням) динамічний список при плануванні виконання задач.

Оцінка асимптотичної складності алгоритмів за часом, а також накладних витрат пам'яті, обумовлених використанням алгоритмами різних службових структур даних, виконувалася на підставі аналізу відповідних вихідних кодів. У всіх експериментах, випробування проводилися за умови відсутності в системі будь-яких задач, що забезпечує коректність одержуваних вимірів часу.

Робота виконана на базі апаратної платформи мікроконтролера STM32F103C8, що має ядро Cortex-M3, 20 Кб SRAM та 64 Кб FLASH. Тактова частота - 8 МГц. Вимірювання часових інтервалів здійснювалося за допомогою таймера високої роздільної здатності DWT (Data Watchpoint and Trace unit). Передача результатів експерименту на хост виконувалася за допомогою інтерфейсу USART. Версії операційних систем: FreeRTOS 10 і Predicate OS 1.1.

Результати та їх обговорення. Результати дослідження алгоритмів ДРП наведено в табл. 1-4 та на рис. 2-5. Було розглянуто чотири схеми розподілу пам'яті, що використовуються в операційній системі FreeRTOS (`heap2.c-heap5.c`), а також дві реалізації алгоритму TLSF, застосованого в Predicate OS: реалізація виключно на мові C, та із застосуванням асемблерних вставок (команди `CLZ` та `RBIT` з системи команд Cortex-M3). Позначення, використовувані в таблицях: F – відсоток відмов при запитах на виділення блоку пам'яті, T_{min} , T_{max} , $T_{сер}$ – відповідно, мінімальний, максимальний та середній час виконання операції (в тактах процесору), S - стандартне відхилення на основі зміщеної оцінки дисперсії, K – коефіцієнт коваріації, Th – умовна верхня межа часу виконання, K_d – коефіцієнт детермінованості.

З отриманих результатів видно, що алгоритм Best-Fit характеризується найбільшим часом виконання (серед розглянутих алгоритмів), та найгіршим детермінізмом. Час виконання операцій виділення та звільнення пам'яті для цього алгоритму пропорційний фрагментації купи (рис. 6).

Розкид часу виконання операції `malloc()` для алгоритмів DL `malloc` та First-Fit на рисунку 2 менше, ніж на рисунках 3-4, що

обумовлено відмінністю у схемах проведення експериментів: у разі першої схеми, на початок кожного кроку моделювання купа знаходиться в нефрагментованому стані. Кількість запитів malloc() фіксована, і це обумовлює обмеженість розміру структур даних (списків, дерев) для кожного з зазначених алгоритмів. У разі використання другої схеми, розміри структур даних не детерміновані алгоритмом моделювання, і розкид часу виконання операції помітно збільшується.

Як у разі імплементації алгоритму TLSF виключно мовою C, так і в разі використання асемблерних вставок, найгірший час виконання операцій malloc() і free() дорівнює $O(1)$, однак різниця між даними двома варіантами реалізації очевидна (рис. 2-5). Якщо для TLSF (C) коефіцієнт детермінованості K_d складає 1.62 (табл. 3), то для TLSF (C + ASM) $K_d = 0$. Алгоритм TLSF дещо поступається алгоритмам DL malloc та

First-Fit за середнім часом виконання (табл. 3), однак має практично ідеальний детермінізм, у разі використання асемблерних вставок ($K_d = 0$ проти $K_d = 15.75$ та $K_d = 0.83$ для DL malloc та First-Fit відповідно). Умовна верхня межа часу виконання Th операції malloc() для алгоритмів TLSF та First-Fit приблизно однакова: 373 та 359 відповідно (табл. 3).

Також помітна різниця у результатах експериментів для випадків $Q = 1$ (запитувані розміри блоків належать діапазону [0; 15359]) та $Q = 20$ (запитувані розміри блоків належать діапазону [0; 767]): кількість відмов у разі $Q = 20$ значно нижча, ніж для $Q = 1$. Так, при $Q = 1$ та другій схемі, відсоток відмов для всіх алгоритмів становить не менше 36%. У разі ж $Q = 20$, F зменшується майже до нуля, і лише в алгоритму Best-Fit лишається рівною ~38%.

Табл. 1 – Дослідження алгоритмів ДРП за першою схемою ($Q = 1$, операція виділення пам'яті).

№	Алгоритм (схема, реалізація)	$F, \%$	T_{min}	T_{max}	$T_{сер}$	S	$K, \%$
1	Best-Fit (heap2.c)	97	180	2848	1685	500	30
2	DL malloc (heap3.c)	46	197	571	241	89	37
3	First-Fit (heap4.c)	46	201	299	253	9	4
4	First-Fit (heap5.c)	53	195	308	251	18	7
5	TLSF (C)	47	287	783	583	74	13
6	TLSF (C + ASM)	47	222	373	358	25	7

Табл. 2 – Дослідження алгоритмів ДРП за другою схемою ($Q = 1$, операція виділення пам'яті).

№	Алгоритм (схема, реалізація)	$F, \%$	T_{min}	T_{max}	$T_{сер}$	S	$K, \%$
1	Best-Fit (heap2.c)	64	183	2555	1611	527	33
2	DL malloc (heap3.c)	36	199	573	287	110	38
3	First-Fit (heap4.c)	36	204	332	257	12	5
4	First-Fit (heap5.c)	40	198	326	254	20	8
5	TLSF (C)	36	290	803	587	74	13
6	TLSF (C + ASM)	36	229	374	360	26	7

Табл. 3 – Дослідження алгоритмів ДРП за другою схемою ($Q = 20$, операція виділення пам'яті).

№	Алгоритм (схема, реалізація)	$F, \%$	T_{min}	T_{max}	$T_{сер}$	S	$K, \%$	Th	K_d
1	Best-Fit (heap2.c)	38,3	183	4563	2674	764	29	3572	121.00
2	DL malloc (heap3.c)	0,0	199	573	294	95	32	509	15.75
3	First-Fit (heap4.c)	0,1	204	394	279	32	11	371	0.83
4	First-Fit (heap5.c)	0,1	198	382	272	31	11	359	0.80
5	TLSF (C)	0,1	365	818	639	80	13	806	1.62
6	TLSF (C + ASM)	0,1	225	373	352	42	12	373	0.00

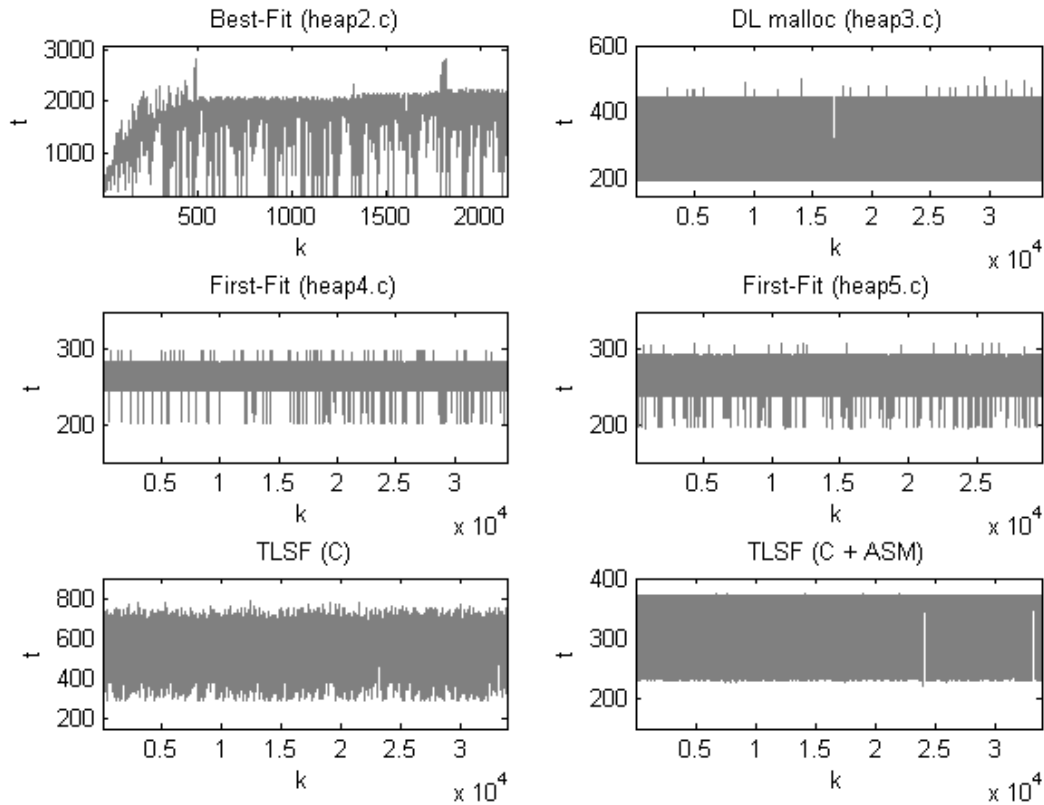


Рис. 2 – Дослідження алгоритмів ДРП за першою схемою ($Q = 1$, операція виділення пам'яті).

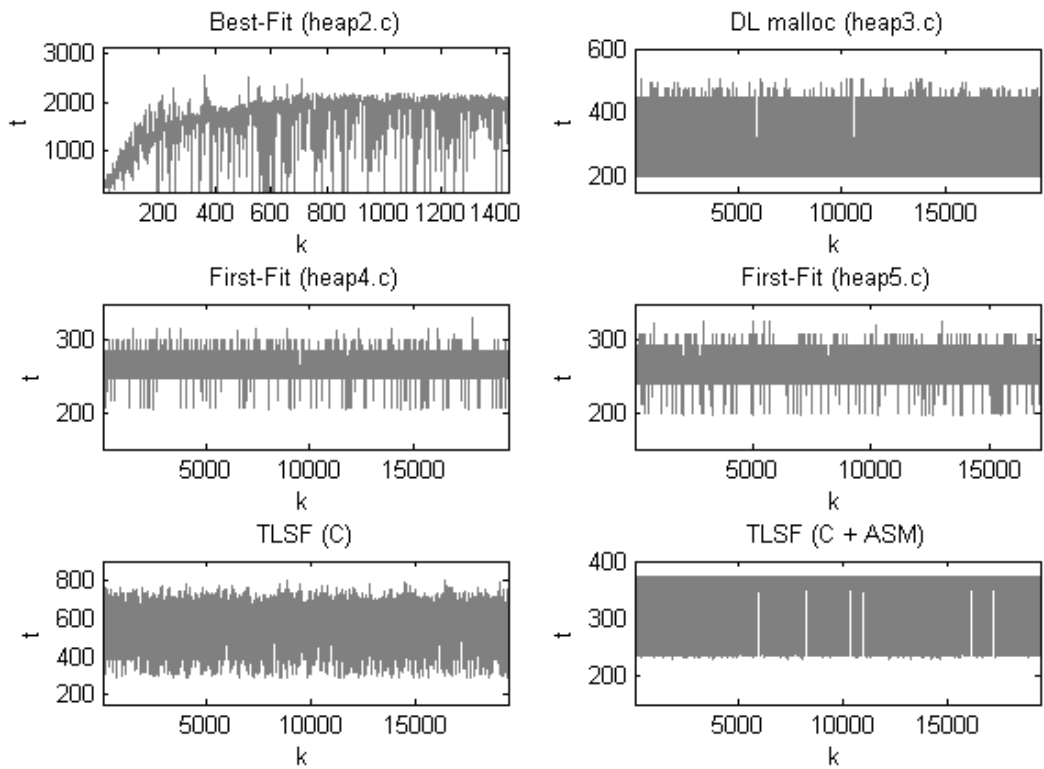


Рис. 3 – Дослідження алгоритмів ДРП за другою схемою ($Q = 1$, операція виділення пам'яті).

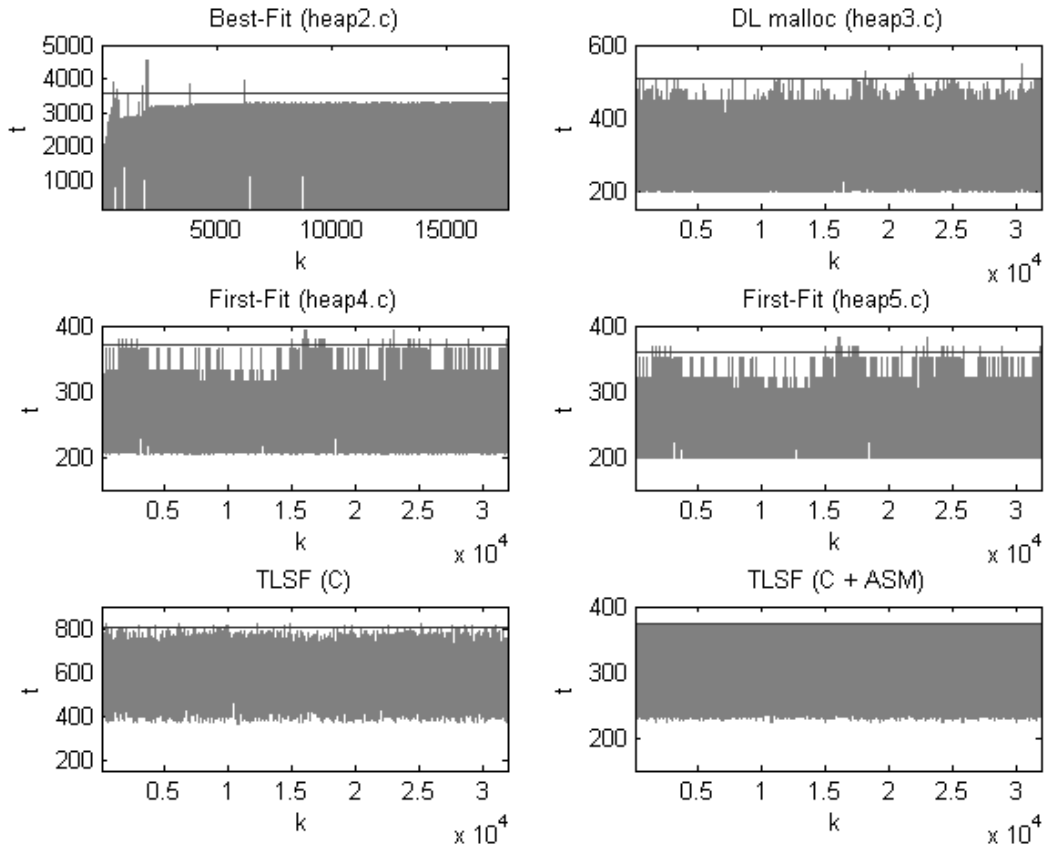


Рис. 4 – Дослідження алгоритмів ДРП за другою схемою ($Q = 20$, операція виділення пам'яті).

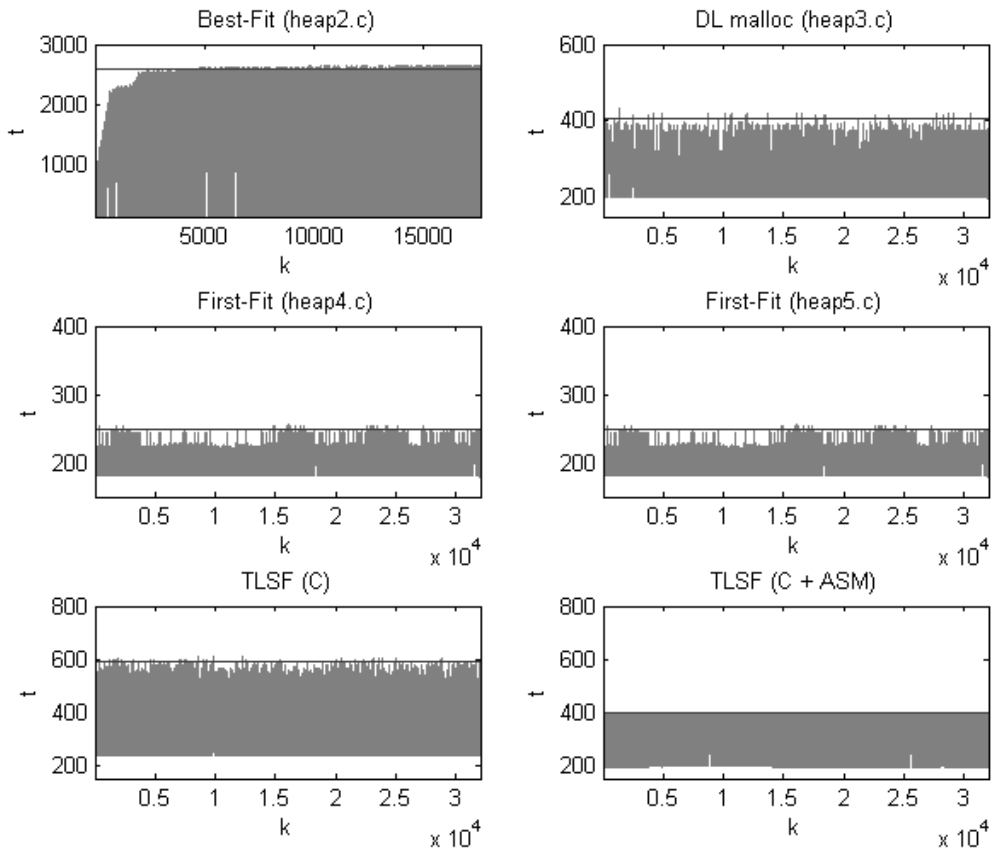


Рис. 5 – Дослідження алгоритмів ДРП за другою схемою ($Q = 20$, операція звільнення пам'яті).

Табл. 4 – Дослідження алгоритмів ДРП за другою схемою ($Q = 20$, операція звільнення пам'яті).

№	Алгоритм (схема, реалізація)	T_{min}	T_{max}	$T_{сер}$	S	$K, \%$	Th	K_d
1	Best-Fit (heap2.c)	144	2644	2135	611	29	2587	0.96
2	DL malloc (heap3.c)	196	434	248	53	21	406	2.11
3	First-Fit (heap4.c)	178	256	211	14	7	249	0.17
4	First-Fit (heap5.c)	178	256	211	14	7	249	0.17
5	TLSF (C)	236	610	369	77	21	593	1.28
6	TLSF (C + ASM)	193	398	283	50	18	397	0.00

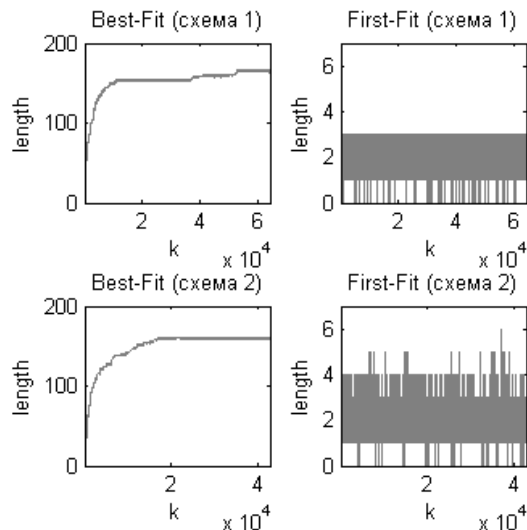


Рис. 6 – Зміна довжини списку вільних блоків в алгоритмах Best-Fit та First-Fit протягом часу моделювання, в залежності від схеми експерименту

Частотний аналіз алгоритму TLSF (операція виділення пам'яті, реалізація з асемблерними вставками, схема 1, $Q = 1$). Було виділено три окремих випадки (один випадок відповідає одній гілці або сукупності гілок алгоритму): F1 (відносна частота 0.39) – індекси f_i та s_i [1], розраховані на підставі запитаного розміру блоку, відповідають не порожньому списку вільних блоків; F2 (відносна частота 0.61) – індекси f_i та s_i вказують на порожній список, і їх необхідно корегувати, виконуючи пошук найближчого не порожнього списку з блоками більшого розміру; F3 (відносна частота 0.03) – виділення знайденого вільного блоку без відокремлення надлишкової частини (рис. 7).

Таким чином, на графіку F1 ми бачимо випадки «прямого влучання». При цьому, надлишкова частина може відділятися, або ні. Графік F2 відповідає випадкам промаху. Надлишкова частина може відділятися, або ні (звичай, відділяється, але декілька точок

внизу відповідають випадкам, коли блок одразу повертається розподільником). F3 – «пряме влучання» або промах, але без відокремлення блоку. Графіки F1-F3 дещо зсунуті з причини появи у кодї вставок трасування.

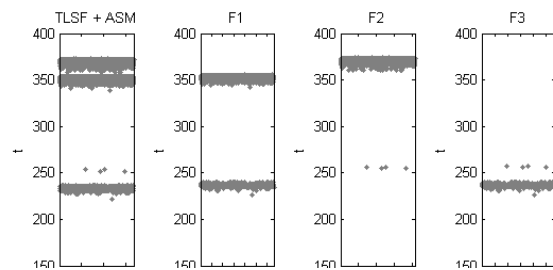


Рис. 7 - Частотний аналіз алгоритму TLSF.

Частотний аналіз алгоритму First-Fit (операція виділення пам'яті, схема 1, $Q = 1$). Виділено чотири окремих випадки (рис. 8): F1 (відносна частота 0.997) – випадок з відокремленням надлишкової частини; F2 (відносна частота 0.003) – без відокремлення; F3 (відносна частота 0.167) – з відокремленням надлишкової частини, без об'єднання з наступним вільним блоком; F4 (відносна частота 0.830) – з відокремленням надлишкової частини, з об'єднанням з наступним вільним блоком.

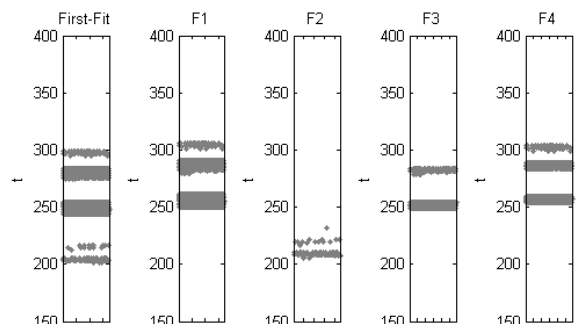


Рис. 8 - Частотний аналіз алгоритму First-Fit.

Вплив константи SLI [1] на показники алгоритму TLSF. Константа FLI визначає кількість бітових мап першого рівня (2^{FLI}), в той час як константа SLI визначає кількість бітових мап другого рівня (2^{SLI}) для кожного діапазону розмірів блоків. З урахуванням того, що використовується апаратна платформа містить лише 20 Кб SRAM, було досліджено залежність показників алгоритму лише від значення константи SLI в діапазоні [1; 4] (табл. 5).

Табл. 5 – Вплив константи SLI на показники TLSF (операція виділення пам'яті, схема 1, $Q = 1$).

SLI	F, %	T_{min}	T_{max}	$T_{сер}$	S	K, %
4	47	222	373	358	25	7
3	48	225	372	354	32	9
2	50	222	373	352	36	10
1	54	224	372	347	42	12

Бачимо, що зменшення SLI веде до збільшення коефіцієнту коваріації, дещо зменшує середнє значення часу виконання операції, та майже не впливає на T_{min} і T_{max} .

Табл. 6 – Вплив вирівнювання розміру блоку на показники алгоритмів Best-Fit та First-Fit.

Вирівнювання, байт \ Алгоритм	1			2			4		
	$T_{сер}$	T_{max}	F, %	$T_{сер}$	T_{max}	F, %	$T_{сер}$	T_{max}	F, %
Best-Fit	2477	5034	97	2141	3509	97	1937	2935	97
First-Fit	260	340	46	259	328	46	254	300	46

Табл. 6 (продовження)

Вирівнювання, байт \ Алгоритм	8			16			32		
	$T_{сер}$	T_{max}	F, %	$T_{сер}$	T_{max}	F, %	$T_{сер}$	T_{max}	F, %
Best-Fit	1601	2702	97	1108	1742	95	649	1025	94
First-Fit	254	300	46	255	301	46	254	301	46

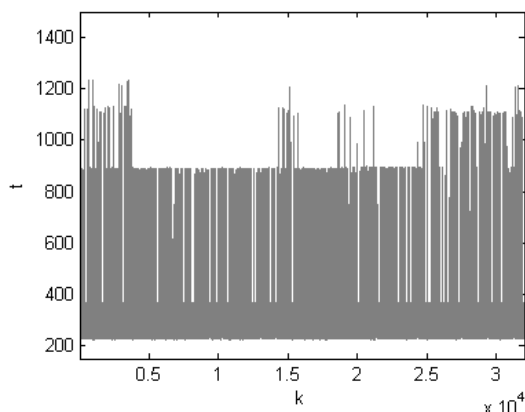


Рис. 9 – Час виклику malloc() TLSF з основної програми, за умови паралельних викликів функцій розподільника із обробника переривань таймера.

Вплив вирівнювання розміру блоку на показники алгоритмів Best-Fit та First-Fit. FreeRTOS дозволяє задавати вирівнювання розміру блоку пам'яті, рівне ступеню двійки від 1 до 32. З отриманих результатів (табл. 6) видно, що у разі Best-Fit, вирівнювання дозволяє істотно скоротити час виділення блоку, та дещо скоротити відсоток відмов. У разі First-Fit, вирівнювання дозволяє дещо скоротити час виділення блоку, і практично не впливає на кількість відмов.

Паралельний виклик функцій розподільника TLSF з основної програми та обробника переривань таймера. Результати вимірів часу виклику функції malloc() алгоритму TLSF із циклу в основній програмі (схема 2) наведено на рис. 9. Завдяки тому, що основний код функцій malloc() і free() алгоритму TLSF в операційній системі Predicate OS захищений критичними секціями, система працює коректно, хоча видно, що середній час виконання malloc() зріс майже вдвічі, і детермінізм часу її виконання погіршився.

Оцінка часової складності алгоритмів ДРП. Результати оцінки часової складності розглянутих у роботі алгоритмів ДРП наведено в табл. 7-8 (відповідно, для функцій виділення та звільнення пам'яті).

Табл. 7 – Асимптотична складність за часом функцій виділення пам'яті в різних алгоритмах ДРП

№	Алгоритм (схема)	Ω	θ	O
1	Best-Fit (heap2.c)	$\Omega(1)$	$\theta(n)$	$O(n)$
2	DL malloc (heap3.c)	$\Omega(1)$	$\theta(1)$	$O(1)$
3	First-Fit (heap4.c, heap5.c)	$\Omega(1)$	$\theta(n)$	$O(n)$
4	TLSF	$\Omega(1)$	$\theta(1)$	$O(1)$

Табл. 8 – Асимптотична складність за часом функцій звільнення пам'яті в різних алгоритмах ДРП

№	Алгоритм (схема)	Ω	θ	O
1	Best-Fit (heap2.c)	$\Omega(1)$	$\theta(n)$	$O(n)$
2	DL malloc (heap3.c)	$\Omega(1)$	$\theta(1)$	$O(1)$
3	First-Fit (heap4.c, heap5.c)	$\Omega(1)$	$\theta(n)$	$O(n)$
4	TLSF	$\Omega(1)$	$\theta(1)$	$O(1)$

Накладні витрати пам'яті на зберігання службових даних розподільника визначаються в байтах відповідно до наведених нижче виразів (в залежності від схеми / алгоритму динамічного розподілу пам'яті), в яких n – кількість блоків, на які фрагментовано купу.

Best-Fit (heap2.c): $M = 22 + 8n$.

First-Fit (heap4.c): $M = 24 + 8n$.

First-Fit (heap5.c): $M = 24 + 8(k + 1) + 8n$, де k – кількість регіонів пам'яті.

TLSF: $M = 20 + 4 \cdot 2^{FLI} \cdot (1 + 2^{SLI}) + 16n$, або $M = 1108 + 16n$, при $FLI = 4$ та $SLI = 4$.

DL malloc: $M = 2196 + 16n_1 + 32n_2$, де n_1 – кількість блоків розміром менше ніж 256 байт (smallbins) [6, 11, 17], n_2 – кількість блоків розміром 256 байт або більше (largebins).

З метою порівняння, в табл. 9 наведено значення накладних витрат пам'яті M для різних схем / алгоритмів ДРП, виходячи з припущення, що в системі наявно 10 блоків розміром 200 байт, та 10 блоків розміром 300 байт. Позначення в таблиці: $M_1, \%$ - накладні витрати пам'яті у відсотках від загального об'єму пам'яті для мікроконтролеру з 20 Кб SRAM; $M_2, \%$ - те ж саме для плати Raspberry Pi 3 з 1 Гб RAM (значення M_1 і M_2 округлені та наближені).

Списки. Функції для роботи зі списками в Predicate OS мають константний час. На рис. 10 (а, б) показано час додання нового

елементу до списку для двох випадків: (а) – список було створено з обмеженням на кількість елементів, які він може містити ($n=20$; якщо список містить максимально допустиму кількість елементів, при спробі додання ще одного, функція одразу повертає управління), (б) – без обмеження. За аналогією, на рис. 10 (г, д), наведено результати для вилучення елемента зі списку. Функції для роботи зі списками в FreeRTOS також мають константний час, за виключенням функції додання елемента до списку vListInsert(), найгірший час виконання якої становить $O(n)$ (рис. 10, в). На рис. 10 (е) показано час видалення елемента зі списку в FreeRTOS. Часові показники функцій роботи зі списками в Predicate OS та FreeRTOS наведено в табл. 10.

Табл. 9 – Накладні витрати пам'яті у разі використанні різних алгоритмів ДРП.

Алгоритм (схема)	M	$M_1, \%$	$M_2, \%$
Best-Fit (heap2.c)	182	1	10^{-5}
First-Fit (heap4.c)	180	1	10^{-5}
First-Fit (heap5.c), ($k = 2$)	200	1	10^{-5}
TLSF	1428	7	10^{-4}
DL malloc	2676	13	10^{-4}

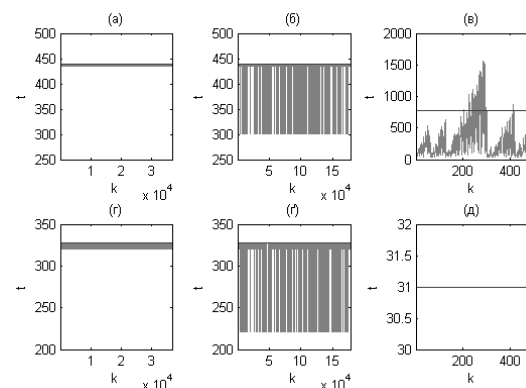


Рис. 10 – Час виконання функцій роботи зі списками в Predicate OS та FreeRTOS.

Табл. 10 – Часові показники функцій роботи зі списками в Predicate OS та FreeRTOS.

№	ОС	Операція	Функція	T_{min}	T_{max}	$T_{сер}$	Th	K_d
1	Predicate OS	add	add_list_item (з обмеженням)	435	439	439	439	0
2		add	add_list_item	302	439	439	439	0
3		delete	delete_list_item (з обмеженням)	320	328	328	328	0
4		delete	delete_list_item	221	328	328	328	0
5	FreeRTOS	add	vListInsert	45	1563	297	774	63
6		delete	uxListRemove	31	31	31	31	0

Результати оцінки накладних витрат пам'яті M в байтах на службові дані для роботи зі списками в обох досліджуваних ОС наведено в табл. 11. Позначення, використані в таблиці: n – кількість елементів в списку, $M, \%$ - відношення M до загального об'єму оперативної пам'яті, для випадку $n = 20$ та об'єму SRAM 20 Кб.

Табл. 11 – Накладні витрати пам'яті при роботі зі списками.

ОС	M	$M, n = 20$	$M, \%$
Predicate OS	$16 + 36n$	736	4
FreeRTOS	$20 + 28n$	580	3

Черги. Всі функції для роботи з чергами в обох досліджуваних ОС мають константний час. Час виконання операцій додання (enqueue) та видалення (dequeue) елемента з черги в FreeRTOS наведено на рис. 11 (в) і (д) відповідно. В Predicate OS можливе створення черг з пріоритетами або без. Час виконання операцій додання та видалення елемента з черги для випадків черги без пріоритетів та з пріоритетами наведено на рис. 11 (а, б) та (г, д) відповідно. Часові показники функцій роботи з чергами в Predicate OS та FreeRTOS наведено в табл. 12.

Накладні витрати при роботі з чергами в FreeRTOS залежать від конфігурації проекту. У разі мінімальної конфігурації (не використовується групування черг, вимкнені

функції налагодження тощо), вони становлять 72 байти на одну чергу.

Накладні витрати при роботі з чергами в Predicate OS визначаються виразом (5):

$$M = 16 \cdot tqc + 17 + 32 \cdot qc + 36 \cdot (sn + pq), \quad (5)$$

де tqc – максимально можлива кількість черг в системі, визначена у файлі конфігурації, qc – загальна кількість черг в системі, sn – сумарна кількість елементів в всіх чергах, pq – кількість черг з пріоритетами. Так, у разі наявності в системі однієї черги без пріоритетів, що містить 10 елементів, при $tqc = 32$ отримаємо 597 байт.

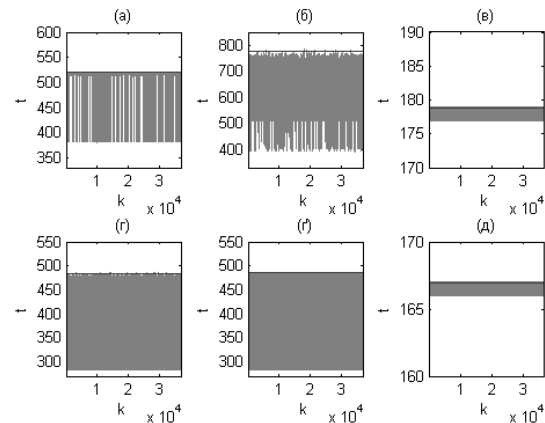


Рис. 11 – Час виконання функцій роботи з чергами в Predicate OS та FreeRTOS.

Табл. 12 – Часові показники функцій роботи з чергами в Predicate OS та FreeRTOS.

№	ОС	Операція	T_{min}	T_{max}	$T_{сеп}$	Th	K_d
1	Predicate OS	enqueue	380	520	518	520	0,00
2		enqueue (з пріоритетами)	392	784	602	777	1,36
3		dequeue	283	485	385	484	0,26
4		dequeue (з пріоритетами)	283	486	376	486	0,00
5	FreeRTOS	enqueue	177	179	177	179	0,004
6		dequeue	166	166	166	166	0,002

Висновки. Було виконано дослідження та порівняльний аналіз алгоритмів ДРП та базових структур даних, що використовуються в операційних системах FreeRTOS і Predicate OS. На підставі отриманих результатів, можна зробити наступні висновки:

1. Якщо один алгоритм характеризується часом $O(1)$, а другий часом $O(n)$, то це ще не означає, що на практиці детермінізм

першого буде кращим. Наприклад, DL malloc має найгірший час виконання $O(1)$, а First-Fit - $O(n)$. Проте, коефіцієнт детермінованості першого дорівнює 15.75, а для First-Fit – 0.83 (табл. 3).

2. Якщо алгоритм характеризується часом $O(n)$, це ще не означає, що на практиці він має поганий детермінізм. Наприклад, в реальних умовах, довжина списку вільних блоків в алгоритмі First-Fit змінюється в

досить вузькому діапазоні (рис. 6), що робить його майже детермінованим.

3. Використання асемблерних вставок із застосуванням команд CLZ та RBIT із системи команд Cortex-M3 дозволяє досягти практично ідеального детермінізму в разі алгоритму TLSF.

4. Зі збільшенням коефіцієнту SLI в алгоритмі TLSF розкид часу виділення блоку зменшується.

5. У разі використання алгоритму Best-Fit, вирівнювання розміру блоку на 32 байти дозволяє суттєво скоротити час виділення пам'яті. Однак, з огляду на великий час виконання алгоритму Best-Fit та сильну не детермінованість, його використання вбачається нераціональним (тим паче, в задачах, що потребують роботи в режимі реального часу).

6. Небажано використовувати функції динамічного управління пам'яттю всередині обробників переривань, оскільки це може суттєво погіршити реактивні властивості та детермінованість системи.

7. Використання списків та черг в Predicate OS пов'язане із значно більшими накладними витратами пам'яті, ніж у FreeRTOS. Питання про те, чи виправдані такі витрати на практиці, потребує додаткового дослідження.

Варто також відзначити різницю між алгоритмом та його імплементацією, і пам'ятати про неї при дослідженні часових властивостей алгоритмів, адже час виконання може істотно залежати не тільки від логіки алгоритму, але й від способу його реалізації.

Отримані в даній роботі результати можуть допомогти практичним програмістам більш свідомо підходити до вибору алгоритмів ДРП для своїх проектів та їх параметрів. Запропоновані поняття умовної верхньої межі часу виконання операції та коефіцієнт детермінованості мають теоретичну та практичну цінність, і дозволяють кількісно оцінювати детермінізм імплементованих алгоритмів.

Література

- [1] M. Masmano, I. Ripoll, A. Crespo, J. Real, "TLSF: новая система распределения динамической памяти для систем реального времени."
- [2] R. Barry, *Using the FreeRTOS Real Time Kernel*. 2009.
- [3] К. С. Гайдук, О. Г. Шевченко, "Предотвращение взаимоблокировок в ОС для программируемых систем на кристалле," *Sci. Educ. a New Dimens. Nat. Tech. Sci.*, т. 6, № 172, С. 25–28, 2018, doi: 10.31174/SEND-NT2018-172VI20-06.
- [4] K. S. Gaiduk, O. G. Shevchenko, "The Deadlock Problem & Approaches to Its Solution," in *Computer and information systems and technologies*, 2018, С. 59.
- [5] "FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions," 2019. [Online]. Режим доступу: <https://www.freertos.org/>. [Перевірено: 25.12.2019].
- [6] J. Hemanth, X. Fernando, P. Lafata, and Z. Baig, "International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018," 2019, т. 26, С. 1053–1060, doi: 10.1007/978-3-030-03146-6.
- [7] V. Heikkilä, "A Study on Dynamic Memory Allocation Mechanisms for Small Block Sizes in Real-Time Embedded Systems," University of Oulu, 2012.
- [8] M. Masmano, I. Ripoll, and A. Crespo, "Dynamic storage allocation for real-time embedded systems," *Real-time Syst. Symp. Work. Sess.*, С. 1–4, 2003.
- [9] S. S. Craciunas, A. Sokolova, C. M. Kirsch, H. Stadler, H. Payer, and R. Staudinger, "A Compacting Real-Time Memory Management System," *Proc. 2008 USENIX Annu. Tech. Conf. USENIX 2008*, С. 349–362, 2008.
- [10] W. Fang, "Analysis on Dynamic Memory Allocation," С. 1–17, 2012.
- [11] A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, "Emerging Technologies in Data Mining and Information Security," in *Proceedings of IEMIS 2018*, 2018, С. 889.
- [12] Э. Таненбаум, *Современные операционные системы*, 3rd ed. СПб.: Питер, 2010.
- [13] "FreeRTOS - Memory management options for the FreeRTOS small footprint, professional grade, real time kernel (scheduler)," 2019. [Online]. Режим доступу:

- <https://www.freertos.org/a00111.html>. [Перевірено: 25.12.2019].
- [14] К. С. Гайдук, "Схеми дослідів," 2019. [Online]. Режим доступу: https://github.com/ks-gayduk/dma_test. [Перевірено: 25.12.2019].
- [15] К. С. Гайдук and О. Г. Шевченко, "Аналітичний огляд парадигм подійно-орієнтованого та автоматного програмування," in *Інформаційна безпека та комп'ютерні технології: Збірник тез доповідей III Міжнародної науково-практичної конференції, 19-20 квітня 2018 року*, 2018, С. 336.
- [16] "FreeRTOS API categories," 2019. [Online]. Режим доступу: <https://www.freertos.org/a00106.html>. [Перевірено: 25.12.2019].
- [17] D. Lea, "dlmalloc." [Online]. Режим доступу: <http://gee.cs.oswego.edu/pub/misc/malloc-2.8.4.c>. [Перевірено: 25.12.2019].

References

- [1] M. Masmano, I. Ripoll, A. Crespo, and J. Real, "TLSF: A new dynamic memory allocator for real-time systems."
- [2] R. Barry, *Using the FreeRTOS Real Time Kernel*. 2009.
- [3] K. S. Gaiduk, O. G. Shevchenko, "Deadlock Prevention in OS for Programmable Systems on a Chip," *Sci. Educ. a New Dimens. Nat. Tech. Sci.*, vol. 6, no. 172, pp. 25–28, 2018, doi: 10.31174/SEND-NT2018-172VI20-06.
- [4] K. S. Gaiduk, O. G. Shevchenko, "The Deadlock Problem & Approaches to Its Solution," in *Computer and information systems and technologies*, 2018, p. 59.
- [5] "FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions," 2019. [Online]. Available: <https://www.freertos.org/>. [Accessed: 25-Dec-2019].
- [6] J. Hemanth, X. Fernando, P. Lafata, and Z. Baig, "International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018," 2019, vol. 26, pp. 1053–1060, doi: 10.1007/978-3-030-03146-6.
- [7] V. Heikkilä, "A Study on Dynamic Memory Allocation Mechanisms for Small Block Sizes in Real-Time Embedded Systems," University of Oulu, 2012.
- [8] M. Masmano, I. Ripoll, and A. Crespo, "Dynamic storage allocation for real-time embedded systems," *Real-time Syst. Symp. Work. Sess.*, pp. 1–4, 2003.
- [9] S. S. Craciunas, A. Sokolova, C. M. Kirsch, H. Stadler, H. Payer, and R. Staudinger, "A Compacting Real-Time Memory Management System," *Proc. 2008 USENIX Annu. Tech. Conf. USENIX 2008*, pp. 349–362, 2008.
- [10] W. Fang, "Analysis on Dynamic Memory Allocation," pp. 1–17, 2012.
- [11] A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, "Emerging Technologies in Data Mining and Information Security," in *Proceedings of IEMIS 2018*, 2018, p. 889.
- [12] Andrew S. Tanenbaum, *Modern Operating Systems*, 3rd ed. SPb.: Piter, 2010.
- [13] "FreeRTOS - Memory management options for the FreeRTOS small footprint, professional grade, real time kernel (scheduler)," 2019. [Online]. Available: <https://www.freertos.org/a00111.html>. [Accessed: 25-Dec-2019].
- [14] K. S. Gaiduk, "Schemes of experiments," 2019. [Online]. Available: https://github.com/ks-gayduk/dma_test. [Accessed: 25-Dec-2019].
- [15] K. S. Gaiduk, O. G. Shevchenko, An analytical overview of event-oriented and automata-based programming paradigms. Information Security and Computer Technologies: a collection of abstracts of the 3rd International Science and Practical Conference (Kropivnitsky, April 19-20, 2018), p. 336.
- [16] "FreeRTOS API categories," 2019. [Online]. Available: <https://www.freertos.org/a00106.html>. [Accessed: 25-Dec-2019].
- [17] D. Lea, "dlmalloc." [Online]. Available: <http://gee.cs.oswego.edu/pub/misc/malloc-2.8.4.c>. [Accessed: 25-Dec-2019].

Надійшла до редакції 25.12.2019 р.

Гайдук Кирило Сергійович – аспірант, Донецький національний технічний університет (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: kyrylo.haiduk@donntu.edu.ua.

Шевченко Ольга Георгіївна – старший викладач, Донецький національний технічний університет (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: olha.shevchenko@donntu.edu.ua.

DATA STRUCTURES AND MEMORY MANAGEMENT IN THE FREERTOS AND PREDICATE OS OPERATING SYSTEMS

The results of the study and comparative analysis of dynamic memory allocation algorithms (DMA) and basic data structures used in the FreeRTOS and Predicate OS embedded operating systems are presented. These systems use such algorithms of DMA as Best-Fit, First-Fit, DL malloc and TLSF. Dynamic lists and queues are the basic data structures of these operating systems. The following indicators were used to quantify and compare algorithms of DMA: minimum, average, and maximum values of operation time; standard deviation based on offset variance estimate; covariance coefficient. Although these indicators provide some characteristic of the algorithm, they are uninformative in the context of deterministic estimation, since in the latter case, the largest variation is the variation of the random variable relative to some upper bound rather than the average. In this connection, the notion of conditional upper limit of operation time, as well as the coefficient of determination, is introduced in the paper and an algorithm for their calculation is given. Two variants of TLSF algorithm implementation are considered: exclusively in C language and using assembly insertions. It is shown that the use of CLZ and RBIT commands from the Cortex-M3 command system allows to achieve practically perfect determinism of the algorithm operation. Frequency analysis and analysis of the influence of the parameters (constants) of the algorithms on their time and statistical performance were performed for individual algorithms of DMA. Additionally, algorithms of DMA and lists and queue functions have been evaluated for their asymptotic time complexity and memory overhead for storing service data. The studies were performed using the STM32F103C8 (20 Kb SRAM) platform, using the high resolution DWT timer. The results obtained can be useful for software engineers in selecting the algorithms for dynamic memory allocation and their parameters.

Keywords: dynamic memory allocation algorithms, determinism of algorithm, dynamic lists, queues, resource overhead, real-time operating systems.

Haiduk Kyrylo Serhiiiovych – PhD-student, Donetsk National Technical University, (2, Shybankova square, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: kyrylo.haiduk@donntu.edu.ua.

Shevchenko Olha Heorhiivna – Senior Lecturer, Donetsk National Technical University, (2, Shybankova square, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: olha.shevchenko@donntu.edu.ua.

ПАРАЛЕЛЬНИЙ ВИРІШУВАЧ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ НА БАЗІ БЛОКОВИХ ОБЧИСЛЮВАЛЬНИХ МЕТОДІВ

Мірошкін О.М., Маргієв Г.Е.

Більшість сучасних програмних пакетів для вирішення диференціальних рівнянь побудовані за послідовними алгоритмами, які не можуть забезпечити ефективного розпаралелювання при застосуванні на паралельних архітектурах. Одною з головних проблем при розробці розподіленого паралельного моделюючого середовища є пошук алгоритмів для побудови вирішувача, який зможе виконувати пошук рішення із заданою точністю у встановлений час, та використовуватиме архітектурно-релевантні принципи по відношенню наявних обчислювальних засобів. Робота присвячена практичному застосуванню чотириточкового блокового чисельного методу задля побудови програмного вирішувача диференціальних рівнянь з урахуванням особливостей наявних паралельних обчислювальних архітектур. Запропонований вирішувач був запрограмований мовою програмування C, формат виклику якого було обрано на основі аналізу інших вирішувачів диференціальних рівнянь та їх систем та адаптовано з урахуванням особливостей обраної мови програмування. Отриманий вирішувач було успішно перевірено на тестових задачах на основі систем рівнянь першого і другого ступенів. Тестування показало високу стійкість та точність при розв'язанні систем кількістю до 10,000 рівнянь. В статті наведено приклад формату використання розробленого вирішувача при чисельному моделюванні аеродинамічних процесів у фрагменті шахтної вентиляційної мережі.

Ключові слова: вирішувач, диференціальні рівняння, блоковий чисельний метод, розпаралелювання, архітектурно-релевантний підхід.

Вступ. Для вирішення задачі чисельного розв'язання диференціальних рівнянь зазвичай використовують програмні вирішувачі, які реалізують чисельні методи Ейлера [1], Сімпсона [2], Адамса [3], Рунге-Кути [4]. Основою цих методів є послідовний підхід до вирішення задачі, а саме пряма залежність наступного значення функції від результатів попереднього кроку алгоритму, що призводить до обмеження в можливостях використання паралельних засобів на рівні алгоритмів [5]. В протилежність до них, блокові обчислювальні методи (БОМ) були розроблені з урахуванням необхідності виконання обчислень на паралельних комп'ютерних системах. БОМ-розробки запропоновано в роботах Фельдмана Л. П., Дмитрієвої О. А., Назарової І. А. та інших [6–8]. Задача побудови розподіленого паралельного моделюючого середовища (РПМС) [9–11] для моделювання складних динамічних систем вимагає програмної реалізації чисельних методів, які б давали змогу вирішувати різноманітні диференціальні рівняння із оптимальним залученням усіх наявних паралельних

ресурсів, реалізуючи так званий архітектурно-релевантний підхід у використанні доступних обчислювальних ресурсів. Для прикладу були обрані системи рівнянь у частинних похідних, які описують аеродинамічні процеси.

Розробка вирішувача. Згідно з [12], рівняння m -крокового k -точкового блокового методу мають наступний вигляд:

$$u_{n,i} = u_{n,0} + i\tau \left(\sum_{j=1}^m b_{i,j} F_{n,j-m} + \sum_{j=1}^k a_{i,j} F_{n,j} \right) \quad (1)$$

де $u_{n,i}$ – наближені рішення в точці i блоку n в момент часу $i\tau$; $n=1, 2, \dots, N$; $i=1, 2, \dots, K$; $u_{n,0}$ – наближене рішення в нульовій точці блоку n . Для першого блоку це значення співпадає з початковою умовою задачі Коші, для всіх наступних – із значенням останньої точки попереднього блоку, тобто:

$$u_{n,0} = \begin{cases} u(0), n=0; \\ u_{n-i,K}, n>0; \end{cases} \quad (2)$$

$F_{n,j} = f(t_n + j\tau; u_{n,j})$ – значення правої частини в точці j блоку n , де $n=1, 2, \dots, N$; $F_{n-1,j} = f(t_{n-1} + j\tau; u_{n-1,j})$ – значення правої частини в точці j блоку $n-1$; $a_{i,j}$, $b_{i,j}$ – коефіцієнти Лагранжа.

При $k=4$ в роботі [13] отримано формули (3) для чотириточкового блокового методу.

Важливою особливістю цих формул є те, що обчислення точок блоку виконується незалежно, що дає змогу застосувати розпаралелювання. На основі формул (3) розроблено програмний вирішувач, загальний алгоритм якого наведено на рис. 1.

Виклик вирішувача. Формат виклику вирішувача для зручності зроблено у звичний для інших програмних вирішувачів спосіб. Далі наведені приклади викликів вирішувачів в різних математичних програмних пакетах. Система Matlab використовує наступний формат виклику функцій [14]:

`[t,y]=ode45(odefun, tspan, y0, options),`

система Mathcad поряд з іншими пропонує для використання наступну функцію [15]:

`rkfixed(init, x1, x2, intvls, D),`

а в системі Wolfram Mathematica можна вирішити диференціальне рівняння наступним чином [16]:

`DSolve[{y'[x] == z[x], z'[x] == -y[x], y[0] == 0, z[0] == 1}, {y[x], z[x]}, x].`

Виклик вирішувачів в інших математичних бібліотеках має свої особливості, але використовує ті ж основні параметри: межі діапазону пошуку рішення, початкові значення, спосіб обчислення значень правої частини.

Виходячи з аналізу вирішувачів у відомих математичних пакетах та алгоритму роботи вирішувача блокового чисельного методу, пропонується наступний формат виклику розробленого вирішувача:

`int solver(*fn, t0, tf, *y0, nf, **x, **y, *n),`

де `*fn` – посилання на функцію обчислення правих частин системи диференціальних рівнянь; `t0`, `tf` – межі діапазону пошуку

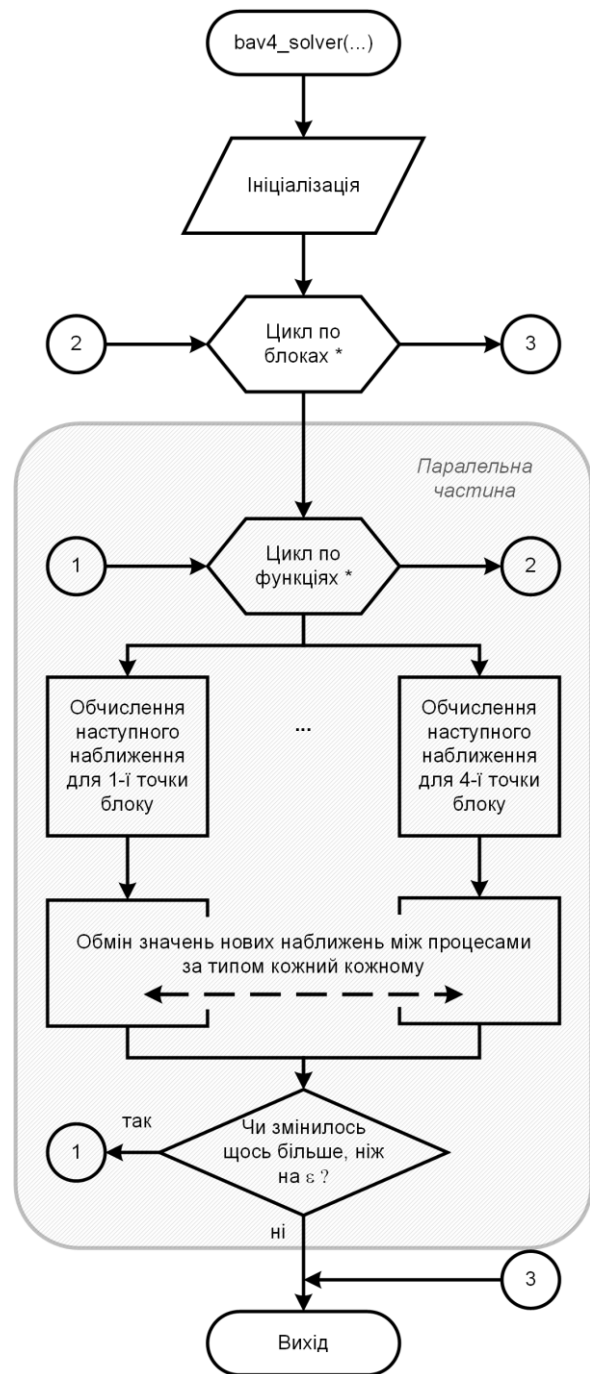


Рис. 1 – Узагальнений алгоритм роботи паралельного блокового вирішувача

рішення; `*y0` – масив початкових значень; `nf` – кількість функцій в системі; `**x`, `*n` – вихідний масив із значеннями абсцис точок, в яких знайдено рішення системи та їх кількість відповідно; `**y` – матриця знайдених значень.

$$\begin{aligned}
 u_{n,1} &= u_{n,0} + \frac{\tau}{720} (251F_{n,0} + 646F_{n,1} - 264F_{n,2} + 106F_{n,3} - 19F_{n,4}); \\
 u_{n,2} &= u_{n,0} + \frac{\tau}{90} (29F_{n,0} + 124F_{n,1} + 24F_{n,2} + 4F_{n,3} - F_{n,4}); \\
 u_{n,3} &= u_{n,0} + \frac{3\tau}{80} (9F_{n,0} + 34F_{n,1} + 24F_{n,2} + 14F_{n,3} - F_{n,4}); \\
 u_{n,4} &= u_{n,0} + \frac{2\tau}{45} (7F_{n,0} + 32F_{n,1} + 12F_{n,2} + 32F_{n,3} + 7F_{n,4});
 \end{aligned} \tag{3}$$

Збільшена кількість параметрів та певні особливості їх використання зумовлені особливостями мови програмування C, що була використана для реалізації вирішувача.

Розпаралелювання. Система формул (3) має чотири незалежних рівняння, обчислення яких доцільно виконати на чотирьох обчислювальних елементах. За умови використання процесору з кількістю обчислювальних ядер 4+, для обчислення точок блоку однієї функції доцільно використовувати переваги технології OpenMP [17], а при розв'язанні систем диференціальних рівнянь доцільне використання декількох процесорів та бібліотеки MPI [18] для обміну даними між ними.

Приклади використання.

Вихідні дані для моделювання. Однією з задач, які вирішуються за допомогою РПМС на кафедрі комп'ютерної інженерії ДонНТУ (м. Покровськ), є моделювання шахтних вентиляційних мереж (ШВМ) як складних динамічних об'єктів. Розглянемо одну з моделей ШВМ, яка задана за допомогою системи диференціальних рівнянь у частинних похідних:

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi_r, t) Q_j^2; \\ -\frac{\partial P_j}{\partial t} = \frac{\rho \alpha^2}{F_j} \frac{\partial Q_j}{\partial \xi}. \end{cases} \tag{4}$$

Докладне пояснення параметрів моделі можна знайти в роботах [10, 19]. Із застосуванням розробленого вирішувача виконаємо моделювання гілки ШВМ довжиною $L = 1000$ м з кроком за просторовою координатою $\Delta \xi = 100$ м, площею поперечного перерізу $F = 5.98$ м² та питомим аеродинамічним опором $r = 1.9e-03$. Одним кінцем гілка, що моделюється, виходить в атмосферу, вважатимемо, що

відносний тиск на цьому кінці гілки дорівнює нулю ($P_{ATM} = 0$), а на іншому кінці гілки є вентилятор, який створює депресію $H = -1$ кПа відносно атмосферного тиску.

Функція обчислення правих частин системи. Після перетворення системи для придатного для обчислення вигляду функція обчислення правих частин системи з 19 рівнянь матиме наступний вигляд:

```

int RHS(double x, double *y, int n, double *rhs)
{
/* Обчислення допоміжних коефіцієнтів *alpha,
beta та gamma
*/
double Patm = 0.0, Hvent = -1000.0;
rhs[ 0] = alpha * (Patm-y[1]) - beta * y[0] *
fabs(y[0]);
rhs[ 1] = gamma * (y[0]-y[2]);
rhs[ 2] = alpha * (y[1]-y[3]) - beta * y[2] * fabs(y[2]);
...
rhs[17] = gamma * (y[16]-y[18]);
rhs[18] = alpha * (y[17]-Hvent) - beta * y[18] *
fabs(y[18]);
}
    
```

Парні елементи масиву rhs відповідають потокам повітря у фрагментах гілки ШВМ після апроксимації моделі за просторовою координатою. Непарні елементи масиву містять значення тиску на межах між фрагментами гілки ШВМ.

Синтаксис виклику вирішувача

```

int main(int argc, char** argv)
{
int NumSections = 10;
int i, nx, nf = 2 * NumSections - 1;
double *x, *y, y0[nf];
for (i = 0; i < nf; i++)
y0[i] = 0.0;
solver_ex(RHS, 0.0, 100.0, &y0[0], nf, &x, &y, &nx,
0.1, 0);
printXY_Excel(x, y, nx, nf, 1);
return 0; }
    
```

Результати обчислень. На рис. 2 та рис. 3 наведені результати моделювання фрагменту шахтної вентиляційної мережі за допомогою розробленого вирішувача на базі

чотириточкового блокового чисельного методу з кроком за часом $t = 0.01$.

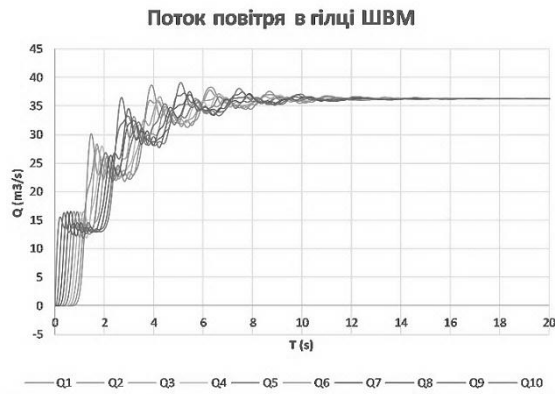


Рис. 2 – Результати моделювання. Потоки повітря в фрагментах гілки ШВМ

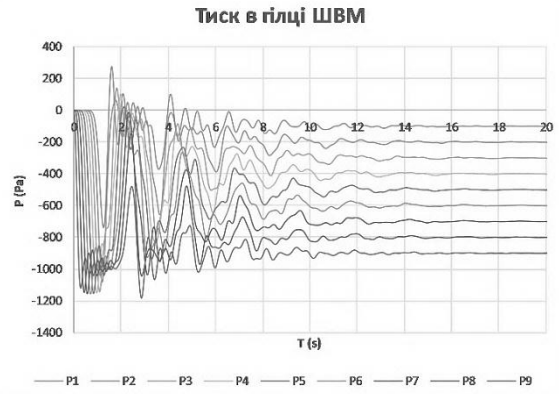


Рис. 3 – Результати моделювання. Тиск в фрагментах гілки ШВМ

Подальші дослідження. Подальша робота над вирішувачем планується в двох основних напрямках – дослідження використання різних підходів до розпаралелювання вирішувача з метою зменшення часу виконання алгоритму на паралельних архітектурах та тестування вирішувача на різних системах диференціальних рівнянь. Крім того, обов'язковим є порівняння розробленого вирішувача із засобами розв'язання диференціальних рівнянь та їх систем, що існують, з метою виявлення класів задач та особливостей апаратних обчислювальних платформ, при яких застосування розробленого вирішувача може зменшити часові показники або збільшити точність рішення.

Висновки. У результаті дослідження було розроблено паралельний вирішувач, який реалізує алгоритм чотириточкового

блокового чисельного методу з урахуванням особливостей архітектури паралельної обчислювальної системи. Вирішувач дозволяє знаходити рішення диференціальних рівнянь та/або їх систем з заданою точністю. Подальшому дослідженню підлягають використання елементів технології OpenMP, щодо оптимізації використання обчислювальних ресурсів у обчислювальних багатоядерних вузлах зі спільною пам'яттю, а також використання розробленого вирішувача на різних рівнях розпаралелювання моделей складних динамічних систем при їх моделюванні в розподіленому паралельному моделюючому середовищі, розробкою якого займаються в лабораторії надпродуктивних обчислень Донецького національного технічного університету в м. Покровськ.

Література

- [1] Метод Ейлера : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Ейлера
- [2] Метод Сімпсона : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Сімпсона
- [3] Метод Сімпсона : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Адамса
- [4] Метод Рунге — Кутти : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Рунге_—_Кутти

- [5] OpenMP topic: Loop parallelism : [Електронний ресурс] // Parallel Programming in MPI and OpenMP by Victor Eijkhout. – Режим доступу : <http://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-loop.html>
- [6] Дмитриева О. А. Разработка многошаговых параллельных коллокационных блочных методов с использованием интерполяционных полиномов Эрмита / О. А. Дмитриева // Радиоэлектронні і комп'ютерні системи. – 2013. – № 5. – С. 243–249. – Режим доступу: http://nbuv.gov.ua/UJRN/recs_2013_5_42
- [7] Устойчивые одношаговые блочные методы численного решения жестких обыкновенных дифференциальных уравнений / Л.П. Фельдман // Штучний інтелект. – 2009. – № 1. – С. 213-217. – Бібліогр.: 9 назв. – рос. – Режим доступу: <http://dspace.nbuv.gov.ua/handle/123456789/7841>
- [8] Назарова И. А. Экстраполяционные блочные одношаговые численные методы решения жестких задач Коши // Научно-теоретический журнал ИПИИ НАН Украины «Искусственный интеллект», №3, 2010. – Донецк: ИПИИ, 2010. – С. 116–126. – Режим доступу: <http://ea.donntu.edu.ua/handle/123456789/6227>
- [9] Святний В.А. Паралельне моделювання складних динамічних систем // Моделирование–2006: Международная конференция. Киев, 2006 г. – Киев, 2006. – С. 83–90.
- [10] Kushnarenko Volodymyr, Miroshkin Oleksandr, Svyatnyj Volodymyr, "Towards the simulation of dynamic network objects with distributed parameters based on block difference numerical methods". The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 21-23 September, 2017, Bucharest, Romania. P. 1073-1076.
- [11] Мірошкін О.М. Архітектура розподіленого паралельного моделюючого середовища для складних динамічних об'єктів з мережною топологією / О. М. Мірошкін, В. А. Святний, А. С. Любимов // Праці VII-ї Міжнародної науково-практичної конференції «Проблеми інформатики та комп'ютерної техніки» (ПКТ – 2018), м. Чернівці: Видавничий дім «Родовід», 2018. – С. 59-61.
- [12] Фельдман Л.П., Назарова И.А. Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений / Л.П. Фельдман, И.А. Назарова // Математическое моделирование. – 2006. – Т. 18, № 6. – С. 17–31.
- [13] Гусева А.Б. Дискретная Simulation-модель сетевого динамического объекта с распределенными параметрами на основе блочного численного метода / А. Б. Гусева, В. Г. Кушнарченко // Проблемы моделирования и автоматизации проектирования: Сб. научн. тр. ДонНТУ, вып. 9 (179), Донецк, 2011, С. 274–283.
- [14] ode45 : [Електронний ресурс] // MathWorks Matlab R2019a Online Documentation. – Режим доступу : <https://www.mathworks.com/help/matlab/ref/ode45.html>
- [15] Решатели ОДУ : [Електронний ресурс] // Справочный центр PTC Mathcad Prime 5.0.0.0. – Режим доступу : http://support.ptc.com/help/mathcad/ru/index.html#page/PTC_Mathcad_Help/ode_solvers.html
- [16] How to | Solve a Differential Equation : [Електронний ресурс] // Wolfram Language & System Documentation Centre. – Режим доступу : <https://reference.wolfram.com/language/howto/SolveADifferentialEquation.html>
- [17] OpenMP : [Електронний ресурс] // Режим доступу : <https://www.openmp.org>
- [18] MPICH | High-Performance Portable MPI : [Електронний ресурс] // Режим доступу : <https://www.mpich.org>
- [19] Svyatnyy V. A. Parallele Modellierung und Simulation der luftdynamischen Prozesse in den Grubenbewetterungsnetzen / V. A. Svyatnyy, V. G. Kushnarenko, O. M. Miroshkin / Вісці Донецького гірничого інституту, № 1 від 2018 р., С. 116–129. - Режим доступу: http://science.donntu.edu.ua/wp-content/uploads/2018/08/Svyatnyy_JDMI_1_18.pdf

References

- [1] “Euler method” 5 Dec. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Euler_method
- [2] “Simpson's rule” 29 Nov. 2019. Accessed on: Dec. 6, 2019. [Online]. Available:

- https://en.wikipedia.org/wiki/Euler_method
- [3] "Linear multistep method" 27 Sep. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Linear_multistep_method
- [4] "Runge–Kutta methods" 22 Nov. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- [5] V. Eijkhout Parallel Programming in MPI and OpenMP. Accessed on: Dec. 6, 2019. [Online]. Available: <http://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-loop.html>
- [6] Dmitrieva O. A. "Razrabotka mnogoshagovykh parallel'nykh kollokatsionnykh blochnykh metodov s ispol'zovaniem interpolatsionnykh polinomov Ehrmita", Radioelektronni i komp'uterni sistemi, vol.64, no. 5, pp. 243–249, 2013. [Online]. Available: http://nbuv.gov.ua/UJRN/recs_2013_5_42
- [7] L. P. Fel'dman, "Stable One-step Block Methods of Numeral Decision of Stiff Ordinary Differential Equations", Iskusstvennyy intellekt, no. 1, pp. 213–217, 2009. [Online]. Available: <http://dspace.nbuv.gov.ua/handle/123456789/7841>
- [8] Nazarova I. A. "Ehkstrapolyatsionnye blochnye odnoshagovye chislennye metody resheniya zhestkikh zadach Koshi", Nauchno-teoreticheskiy zhurnal IPII NAN Ukrainy «Iskusstvennyy intellekt» Donetsk, no. 3, pp. 116–126, 2010. [Online]. Available: <http://ea.donntu.edu.ua/handle/123456789/6227>
- [9] Sviatnyi V.A., "Paralelne modeliuвання skladnykh dynamichnykh system" in Mezhdunarodnaya konferentsiya: Modelirovanie–2006. Kiev, 2006, Ukraine, pp. 83–90.
- [10] Kushnarenko Volodymyr, Miroszkin Oleksandr, Svyatnyj Volodymyr, "Towards the simulation of dynamic network objects with distributed parameters based on block difference numerical methods". The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 21-23 September, 2017, Bucharest, Romania. P. 1073-1076.
- [11] O.M. Miroszkin, V. A. Sviatnyi, A. S. Liubymov, "Arkhitektura rozpodilenooho paralelnoho modeliuuchoho seredovyscha dlia skladnykh dynamichnykh ob'ektiv z merezhnoiu topolohiieiu", Pratsi VII-yi Mizhnarodnoi naukovo-praktychnoi konferentsii «Problemy informatyky ta kompiuternoï tekhniki» (PIKT – 2018), Vydavnychiy dim «Rodovid», Chernivtsi, 2018, Ukraine, pp. 59-61.
- [12] L.P. Fel'dman, I.A. Nazarova, "Paralel'nye algoritmy chislennogo resheniya zadachi Koshi dlya sistem obyknovennykh differentsial'nykh uravneniy", Matematicheskoe modelirovanie, vol. 18, no. 6 pp. 17–31, 2006.
- [13] A.B. Guseva, V. G. Kushnarenko "Diskretnaya Simulation-model' setevogo dinamicheskogo ob"ekta s raspredelennymi parametrami na osnove blochno chislennogo metoda", Problemy modelirovaniya i avtomatizatsii proektirovaniya: Sb. nauchn. tr. DonNTU, vol. 179, no. 9, Donetsk, 2011, Ukraine, pp. 274–283.
- [14] "ode45 MathWorks Matlab R2019a Online Documentation". Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.mathworks.com/help/matlab/ref/ode45.html>
- [15] "Решатели ОДУ" Accessed on: Dec. 6, 2019. [Online]. Available: http://support.ptc.com/help/mathcad/ru/index.html#page/PTC_Mathcad_Help/ode_solvers.html
- [16] "How to | Solve a Differential Equation" 2019. Accessed on: Dec. 6, 2019. [Online]. Available: <https://reference.wolfram.com/language/howto/SolveADifferentialEquation.html>
- [17] "OpenMP" Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.openmp.org>
- [18] "MPICH | High-Performance Portable MPI" Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.mpich.org>
- [19] V. A. Svyatnyy, V. G. Kushnarenko, O. M. Miroszkin, "Parallele Modellierung und Simulation der luftdynamischen Prozesse in den Grubenbewetterungsnetzen", Visti Donetskoho hirnychoho instytutu, no. 1, pp. 116–129, 2018. [Online]. Available: http://science.donntu.edu.ua/wp-content/uploads/2018/08/Svyatnyy_JDMI_1_18.pdf

Надійшла до редакції 07.12.2019 р.

Мірошкін Олександр Миколайович – кандидат технічних наук, доцент, Університет Ульма (алея Альберта Ейнштейна, 11, Ульм, 89081, Німеччина).

E-mail: miroshkinan@gmail.com

Маргієв Георгій Едуардович – аспірант, кафедра комп'ютерної інженерії, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: margievge@gmail.com

BLOCK NUMERICAL METHOD BASED PARALLEL SOLVER FOR DIFFERENTIAL EQUATIONS

Most modern software packages for solving differential equations are based on sequential algorithms that cannot provide efficient parallelization when applied to parallel architectures. One of the main problems in distributed parallel simulation environment development is search for algorithms for a solver that can give a mathematical problem solution with a given accuracy within a specified time, and will use architecturally-relevant principles with respect to existing computing resources. A separate class of problems when using such principles is in load balancing between individual computing nodes while reducing the number of exchange operations between them.

The paper is dedicated to a practical application of a four-point block numerical method for the development of the software differential equations solver, taking into account the features of parallel computing architectures to be used. The proposed solver was programmed in C programming language, the call format of which has been chosen based on analysis of other solvers for differential equation and their systems and adapted to the specific features of the selected programming language. The solver is built using two approaches for computing processes parallelizing in modern computer architectures: multi-platform shared memory multiprocessing programming in C (OpenMP) and parallel library for distributed memory systems (MPI). The resulting solver was successfully tested on different test problems based on differential equation systems of the first and second orders. The testing showed high stability and accuracy during solving systems with up to 10,000 equations. In the article an example of developed solver usage for numerical simulation of mine air network aerodynamical processes, represented by the system of differential equations in partial derivatives, is shown.

Keywords: *solver, differential equations, block numerical method, parallelization, architecturally-relevant approach.*

Miroshkin Oleksandr Mykolaiovych – Candidate of Technical Sciences, Professor, Associated Professor, Ulm University (Albert Einstein Alley, Ulm, 89081, Germany).

E-mail: miroshkinan@gmail.com

Marhiiev Heorhii Eduardovych – PhD student, Computer Engineering Department, State Higher Education Establishment "Donetsk National Technical University", (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

Наукове видання

Наукові праці Донецького національного технічного університету

Серія «Проблеми моделювання та автоматизації проектування»

Всеукраїнський науковий збірник

№ 1(14), 2019

(українською, російською, англійською, німецькою мовами)

Відповідальний за випуск: В. А. Святний

Технічний редактор: К. С. Гайдук

Підписано до друку 27.12.2019.

Формат 60 x 84 1/16. Гарнітура «Таймс»

Папір офсетний. Друк офсетний.

Умов. друк. арк. 6,6. Облік.- вид. арк. 4,2

Тираж 300 прим. Замовлення № 11/20

Адреса редакції: ДВНЗ «ДонНТУ», Україна, 85300, м. Покровськ, пл. Шибанкова, 2.

Тел.: (062) 392-03-09. E-mail: pmap@donntu.edu.ua

<http://pmap.donntu.edu.ua>

Видавець та виготовлювач: Державний вищий навчальний заклад

«Донецький національний технічний університет».

Україна, 85300, м. Покровськ, пл. Шибанкова, 2. Тел.: (062) 392-03-09

Свідоцтво про державну реєстрацію суб'єкта видавничої справи:

серія ДК № 4911 від 09.06.2015 р.