Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель самовідновлювальної кіберфізичної системи

(тема)

Виконав:

студент II курсу, групи КСМм-20-1

Левцун Д.К.
(прізвище, ініціали)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерні системи та мережі
(повна назва освітньої програми)

Керівник: проф. Рубан І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Левцуну Денису Костянтиновичу
(прізвище, ім’я, по батькові)

1. Тема роботи Модель самовідновлювальної кіберфізичної системи

затверджена наказом по університету від “ 05 ” листопада 2021 р. № 1656 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.

3. Вхідні дані до роботи Архітектура системи

4. Перелік питань, що потрібно опрацювати у роботі

1) Аналіз предметної області
2) Узагальнена теоретико-множинна модель КФС
3) Розробка моделі CPS
4) Реалізація моделі СКФС

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 13 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	09.11.21-16.11.21	
2	Розробка моделі КФС	17.11.21-21.11.21	
3	Реалізація .моделі КФС	22.11.21-02.12.21	
4	Оформлення результатів кваліфікаційної роботи	03.12.21-12.12.21	
5	Подання кваліфікаційної роботи керівникові та	12.12.21-13.12.21	
	її попередній захист		
6	Подання кваліфікаційної роботи на рецензу-	13.12.21-14.12.21	
	вання		

Дата видачі завдання 08 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Рубан І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 67 с., 15 рис., 1 дод., 18 джерел.

CYBER-PHISICAL SYSTEM, SELF-HEALING, SELF-AWERE, SHSA, MODEL.

Метою кваліфікаційної роботи є розробка рішення для самовідновлюваних CPS. У ході виконання кваліфікаційної роботи проведено огляд сучасних кіберфізичних.

За результатами проведених досліджень були проаналізовані методи та технології моделювання самовідновлювальних кіберфізичних систем.

Як результат кваліфікаційної роботи було створено модель самовідновлювальної кіберфізичної системи, програмний продукт, що реалізує цю модель. У цій роботі розробленно модель самовідновлення КФС, використовуючи надмірність інформації у комунікаційної мережі. Інформація або сигнали фізичних об’єктів – змінних CPS можуть бути закодовані на основі знань відносин між цими змінними КФС.

ABSTRACT

Master’s thesis: 67 pages, 15 figures, 1 appendice, 18 sources.

CYBER-PHISICAL SYSTEM, SELF-HEALING, SELF-AWERE, SHSA, MODEL.

The purpose of the qualification work is to develop a model of self-healing CPS. In the course of the qualification work a review of modern cyber-physical.

According to the results of the research, methods and technologies for modeling self-healing cyber-physical systems were analyzed.

As a result of the qualification work a model of self-healing cyber-physical system was created, a software product that implements this model. In this paper, a model of self-healing CPS was developed using redundancy of information in the communication network. Information or signals of physical objects – CPS variables can be encoded based on the knowledge of the relationships between these CPS variables.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КФС – кіберфізична система

АС – автономні обчислення (англ., Autonomic Computing)

AI – штучний інтелект (англ., Artificial Intelligence)

API – інтерфейс прикладного програмування (англ., Application Programming Interface)

CPS – кіберфізична система (англ.,Cyber-Physical System)

IoT – інтернет речей (англ.,Internet of Things)

ORR – реконфігурація часу виконання на основі онтології (англ., Ontology-based Runtime Reconfiguration)

QoS – якість обслуговування (англ., Quality-of-Service)

SH-PGSA – самовідновлення шляхом структурної адаптації, що керується властивостями (англ., Self-Healing by Property-Guided Structural Adaptation)

SHSA – самолікування шляхом структурної адаптації (англ., Self-Healing by Structural Adaptation)

SOA – сервісно-орієнтована архітектура (англ., Service-Oriented Architecture)

ВСТУП

Особливість розвитку сучасної техніки полягає в тому, що вона набуває самоврядність за рахунок використання інтелектуальних програмно-апаратних структур. Таким чином, реалізується перехід управлінських функцій від людини до самої техніки. Це веде до глобальних змін – набуття штучного інтелекту техносферою, що стає локомотивом земної еволюції.

Основним фактором, що стимулює появу розумної техніки, є психофізична обмеженість людини. Вважається «…що у сьогоднішньому світі у процесі трансформації сучасної техніки в психології людини відбулася зміна традиційного поняття щастя. Під щастям сучасної людини у всеосяжному уявленні стала розуміти комфортність його життя, яка забезпечується цифровізацією його життя буття. У сферу людської життєдіяльності, як у величезну всепоглинаючу вирву, втягуються не лише певні об’єкти забезпечення комфортності індивіда (перелік яких експоненційно зростає), а й різноманітних послуги, а також енергія та інформація. В результаті спостерігається вибухоподібний процес ускладнення техніки, що оточує людину».

Крім того, розвиток складної техніки породжує труднощі її експлуатації, технічного обслуговування. Внаслідок чого через неправильне обслуговування або неправильних проектних рішень відбуваються аварії та техногенні катастрофи, що тягнуть за собою не тільки руйнування та забруднення навколишнього середовища, а й людські жертви.

Проте розумні сутності сучасна техніка забезпечує еволюційний прорив людства у майбутнє. Подальша реалізація цієї тенденції неминуче веде до всеосяжної цифровізації середовища, потенціал якої буде порівняти з можливостями людського розуму, а згодом мабуть значно їх перевищить.

У кіберфізичних системах виробництво та виробничі процеси відбуваються у фізичному просторі, а їхня віртуальна реальність протікає в

комп’ютерні системи, в інформаційному просторі. У комп’ютерному світі КФС здійснюється перетворення виробничих проблем, що відбуваються в фізичний простір у своїх цифрових двійників. Взаємодія віртуальних сутностей для досягнення заданих цілей дозволяє через дзеркальне відображення фізичних сутностей керувати фізичним світом за допомогою зворотного перетворення інформації на фізичні дії

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття

Національний інститут стандартів та технологій США (NIST), визначає кіберфізичні системи (Cyber-Physical Systems, CPS) як сукупність цифрових, аналогових, фізичних та людських компонентів, функціонуючих за допомогою інтегрованої фізичної технології та логіки [1].

Термін CPS запропонований для позначення автоматизованих комплексів, що складаються з різних природних об’єктів, штучних систем та керуючих контролерів, що об’єднуються в єдине ціле засобу інтернет (Інтранет).

Технічні причини виникнення CPS [1]:

- зростання числа пристроїв із вбудованими процесорами та засобами зберігання даних: сенсорні мережі, медичне обладнання; розумні будинки і т.д;

- інтеграція, що дозволяє досягти найбільшого ефекту шляхом об’єднання окремих компонентів у великі системи: Інтернет речей (IoT), World Wide Sensor Net, розумні довкілля (Smart Building Environment), оборонні системи майбутнього;

- обмеження когнітивних здібностей людини, що еволюціонують повільніше, ніж машини. Люди не можуть впоратися з обсягом інформації, необхідним прийняття рішень, і передають частину функцій CPS, виводячи себе з контуру управління (human out of loop). Кіберфізичні системи можуть посилити аналітичні здібності людини.

Існує потреба у створенні інтерактивних систем нового рівня, що зберігають людину в контурі управління (human in the loop).

Кіберфізична система – організаційно-технічна концепція управління інформаційними потоками, інтеграція обчислювальних ресурсів у фізичні

процеси виробництва. У такій системі датчики, контролери та інформаційні системи об’єднані у єдину мережу. По суті, CPS є промисловим інтернетом речей (IOT) з розподіленою автоматизованою системою управління з урахуванням TCP/IP-сети інтелектуальних агентів. Основою CPS є закладена в систему управління комп’ютерну модель (цифрового або («тіньового») двійника фізичного активу), що забезпечує ефективну працездатність активу загалом.

Кіберфізичні системи (CPS) або (КФС) – розумні системи, які включають інтерактивні інженерні мережі з фізичних та комунікаційних компонент. CPS та пов’язані з ним системи є загальновизнаними інструментами, що мають великий потенціал впровадження, що створює шляхи реалізації інноваційних додатків, які мають величезний вплив на безліч секторів світової економіки [2]. Серед цих секторів NIST вказує насамперед на промисловість, транспорт, енергетику та охорону здоров’я.

Поняття КФС є всеосяжним. За певних умов Значну частину явищ сучасного світу можна назвати КФС. Тому до категорії КФС можна віднести величезну кількість пристроїв та концепції. До них відносяться, як мінімум такі складові сучасної техніки:

- роботи, у тому числі розумні роботи;
- Інтернет речей (включаючи промисловий (індустріальний) Інтернет);
- високоавтоматизовані автомобілі та високоавтоматизований;
- транспорт;
- безпілотні літальні апарати;
- «розумне місто»;
- «розумні» медичні пристрої.

Під цифровізацією сучасної техніки розуміють поетапне розширене впровадження в технічні системи/об’єкти та різні технології (інформаційні) компонентів, що мають інтелектуальні властивості. на сьогодні ця тенденція спостерігається повсюдно: дитячі іграшки, побутові прилади, оснащені

мікрочіпами, найскладніші технологічні комплекси нафтохімічних та хімічних виробництв.

Концепцію побудови КФС вважатимуться розвитком з одного боку технічної кібернетики, яка орієнтована на застосування кібернетичних моделей дослідження проблем керованості, стійкості, безпеки та ін. З іншого боку концепція кіберфізики істотно розширює поняття фізики не тільки пов’язаних з її новими сучасними розділами, а й поняття штучно створених систем, які використовують принципи живої природи.

Концепція кіберфізичних систем полягає в інтеграції обчислювальних ресурсів та фізичних процесів з метою досягнення емерджентності, появи принципово нових можливостей у створюваних CPS. Метою такої інтеграції стає досягнення синергетичного ефекту за рахунок нових системних зв’язків, одержання особливих властивостей, що не притаманні її елементам.

За застосовністю пристроїв CPS можуть бути класифіковані за галузевому принципу: у НУО, у нафтохімії, у промисловості, в енергетиці тощо.

За форматом пристроїв CPS можуть включати глобальні «системи систем», їх окремі складові, датчики та засоби вимірювань, об’єкти будь-якого розміру та масштабу (від нано роботів до розумних фабрик).

CPS включає як апаратну, так і віртуальну (обчислювальну) частини. Кожен із цих компонентів, у свою чергу, може взаємодіяти з більшістю сучасних технологічних трендів, адитивним виробництвом, блокчейн-технологією, віртуальною та доповненою реальностями (VR та AR), хмарними обчисленнями тощо.

Сам термін CPS може використовуватися як для опису конкретного пристрої, так і для опису системи або концепції (маючи на увазі інтеграцію обчислювального компонента у фізичний процес).

З технічного погляду CPS мають багато спільного зі структурами типу GRID, цільовими завданнями яких є інтеграція, віртуалізація та управління ресурсами та послугами у розподілених, гетерогенних, динамічних

«віртуальних організаціях». Вирішення цих завдань у CPS виконується за допомогою промислового інтернету речей (IoT), міжмашинного взаємодії (Machine-to-Machine, M2M), та бланкових технологій (Fog та Cloud Computing). Очікується, що на новому щаблі розвитку хмарних обчислень будуть знижуватися затримки, що виникають під час передачі даних, і будуть забезпечені нові можливості розвитку інтелектуальних механізмів [3].

У той же час CPS – це інструмент для подолання таких проблем автоматизованих систем управління, як розподіленість, надійність, відмовостійкість, безпека, масштабованість та автономність роботи в важкодоступні місця.

На польовому рівні CPS ефективними є сенсорні мережі (Sensor NetWorks). Сенсорні мережі, розгорнуті на основі багатьох недорогих датчиків, дозволяють отримувати необхідні обсяги інформації для подальшої аналітики.

Відмінними рисами CPS є [1, 3]:

- орієнтація послуги. Сервіси CPS реалізуються на основі сервісно-орієнтованої еталонної архітектури;
- інтелектуальна самоорганізація. CPS забезпечує здатність приймати рішення самостійно;
- здатність людей і CPS об’єднуватися для вирішення спільних проблем та спілкуватися один з одним;
- віртуалізація світу виробництва на різних рівнях деталізації, від датчиків та виконавчих механізмів для всього CPS;
- формування бази знань на основі технологічної та технічної інформації;
- забезпечення міждисциплінарної модульності, гнучкої адаптації до змін, що змінюються шляхом заміни або розширення окремих модулів;
- можливості у реальному часі. Алгоритми та технології великих даних, що надаються в режимі реального часу;
- оптимізація виробничого процесу з використанням алгоритмів та

великих даних підвищення загальної ефективності устаткування (Overall Equipment Effectiveness, OEE);

- інтеграція даних з дисциплін і за життєвим циклом;
- доступ до даних, що надійно зберігаються у хмарі або інтрамережі.

На рисунку 1.1 показана архітектура кіберфізичної системи, названа 5C (connection, conversion, cyber, cognition, configuration) (комунікаційне середовище, конверсія даних в інформацію, кіберрівень цифрових двійників, амопізнання своєї сутності, самоконфігурація).

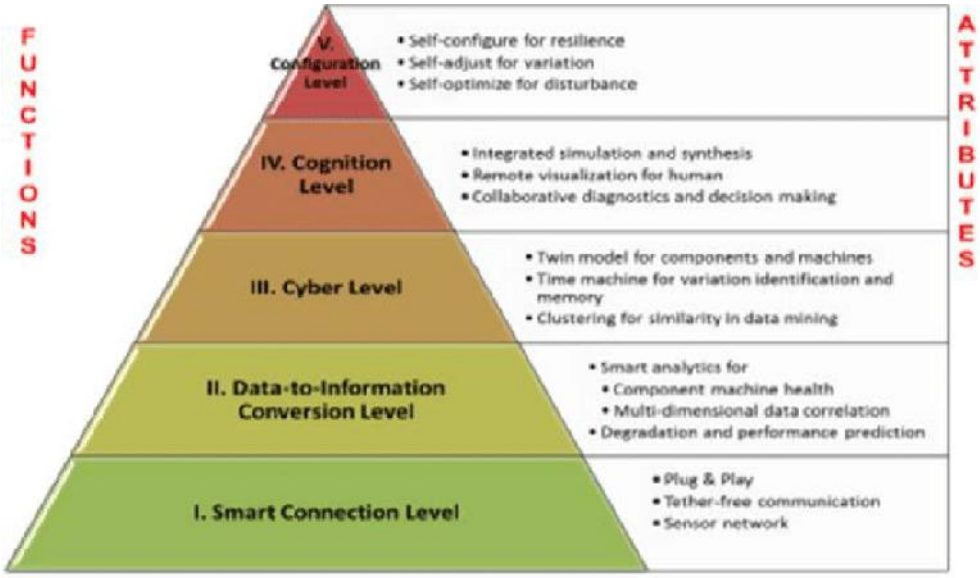

Рисунок 1.1 – 5C архітектура кіберфізичних систем

На рівні «Комунікаційне середовище» пристрої можуть виконувати самопідключення та самоконтроль своєї поведінки. Цей рівень є найнижчий шар CPS і реалізує перший крок до досягнення інтелектуалізації активу з використанням таких елементів, як розумні датчики, розумні виконавчі механізми та розумні протоколи комунікаційного середовища потоку даних. Розумні КВП реалізують самоналаштування і самоконтроль польового рівня. Східні та низхідні канали зв’язку цього рівня стають особливо значущими при інтеграції польового та виробничого рівнів при вирішенні таких ключових бізнес-процесів, як системи планування ресурсів підприємства (ERP),

управління відносинами з клієнтами (CRM) та управління ланцюжками поставок (SCM).

«Конверсія» – шар CPS, на якому багато типів обурень та невідповідності виключаються з інформації, отриманої з польового рівня, завдяки використанню таких інструментів як глибока аналітика даних та хмарні обчислення. Ці аналітичні інструменти допомагають підготовці вирішення таких проблем, як обчислення терміну служби активу, що залишився, і виявлення причин незапланованих зупинок технологічного обладнання. На цьому рівні на основі вимірювань пристроїв та датчиків виконується оцінка критичних проблем із самосвідомістю. Дані від пристроїв з автономним та дротовим підключенням та розумними КВП стають основою для отримання інформації від активів. Їхні дані стають основою для самовизначення критичних проблем фізичних активів.

На рівні «Кіберрівень цифрових двійників» кожна фізична сутність технологічного простору створює свій «двійник», використовуючи інструментальні функції, які в подальшому контролюють стан активів, заснованих на методології «Time-Machine» (стандартної функції безперервного автоматичного копіювання фізичної сутності на віртуальному рівні). Віртуальний «двійник» активу в технологічному кіберпросторі виконує самоаналіз та здійснює обчислювальну підготовку управління фізичною сутністю.

Рівень «Самопізнання своєї сутності» представляє результати самооцінки на підставі «інфографічного» (infographic) значення з метою вказівки користувачам контексту потенційних проблем. Рівень виконавчих дій виконує перетворення інтелектуальних рішень на дію лише на рівні активів. На підставі критеріїв пріоритету, ризику для досягнення необхідного рівня продуктивності або стійкості до відмови може бути переконфігурована. На цьому етапі використовуються алгоритми, програмне забезпечення та комп’ютерна інфраструктура для аналізу стану активів та прогнозування їхньої поведінки. Це комплекс програмно-алгоритмічного забезпечення побудови

обчислювальної архітектури, алгоритмів та забезпечення інформаційної безпеки.

Рівень "Самоконфігурація" перетворення потенціалу інтелекту в дію, що виконує рух від кіберпростору до фізичного простору. Цей етап дозволяє кіберфізичній системі переводити рішення в реальні події. Методологія кібернетичної фізики заснована на добре розроблених методах теорії управління: методах лінійного та нелінійного програмування, оптимального, адаптивного керування; методи ідентифікації параметрів, методи фільтрації та оцінювання станів, методи оптимізації систем.

1.2 Обчислювальна самосвідомість

CPS значно виграли б від вищого ступеня автономії, включаючи всебічне розуміння їхньої ситуації в навколишньому середовищі, їхнього власного стану, а також їхньої цілісності та ефективності для прийняття найкращих рішень щодо їх обмежених ресурсів на найбільш вигідні дії. В цілому основні проблеми, з якими стикаються CPS, зводяться до наступного:

Динаміка довкілля. Навколишнє середовище постійно змінюється і відома лише частково відома лише частково. У результаті наслідки дій системи невизначені, її плани мають бути постійно переглядати та переосмислювати.

Ситуативність та суб’єктивність. Перспективи кожної системи різні, що з різним контекстом і позиції, з якою відбувається зондування, і навіть через те, що самі датчики унікальні. Ці фактори мають бути враховані під час аналізу сенсорних даних.

Фізичні обмеження та надійність. Усі ресурси обмежені через розмір, вартість, обчислювальну обмеження за розміром, вартістю, обчислювальною потужністю та енергопостачання. Крім того, якість зондування, обчислень та зв’язку часто обмежена і ще більше погіршується протягом терміну служби через несправності, відмови, знос і старіння.

У контексті обчислювальної техніки поняття самосвідомості використовувалося для характеристики систем, здатних створювати та використовувати знання про аспекти, пов’язані з ними, наприклад, власний стан, ситуацію, поведінку та цілі. Це також можна застосовувати рекурсивно, щоб самосвідома обчислювальна система могла міркувати про свої власні процеси для виконання цих речей [4]. Самосвідомість була запропонована як ключова властивість для все більш складних систем [4].

Дослідження самосвідомих комп’ютерів було натхнене психологічними концепціями та процесами[5]. Однак воно спрямований на розробку обчислювальних інтерпретацій властивостей і можливостей, пов’язаних з самосвідомістю, які є цінними для технічних систем. Оскільки це відкритий і складний набір характеристик, обчислювальну самосвідомість зазвичай визначають у термінах рівнів самосвідомості [5, 6], які можуть бути присутніми або бажаними в різному ступені. Важливою відмінністю самосвідомих обчислювальних систем є те, що вони відчувають своє середовище та свій стан під час виконання, щоб отримати знання, які потім можуть бути використані для міркувань та прийняття рішень під час виконання. Це передбачає іншу нову процедуру проектування, в якій рішення, що залежать від контексту та часу виконання, не інтегруються в систему під час проектування дизайнером (оскільки це неможливо зробити належним чином), а приймаються під час виконання самою системою.

Протягом останніх двох десятиліть дослідницьке співтовариство розробило ряд еталонних архітектур для контролерів агентів як частини само систем з різними акцентами. Ключові приклади включають спостереження-вирішення-дія (ODA) [7] і моніторинг-аналіз-планування-виконання через спільну базу знань (MAPE-K) [8]. Однак, хоча системи, побудовані на цих моделях, часто мають певний рівень самосвідомості, моделі явно не відображають проблем самосвідомості. Зовсім недавно були розроблені еталонні архітектури, щоб чітко підкреслити ці проблеми та забезпечити архітектурний погляд на (можливі) можливості системи самосвідомості. Дві

широко прийняті еталонні архітектури включають еталонну архітектуру EPiCS [9] і Learn-Reason-Act[10].

Дослідження самосвідомих обчислювальних систем було підтримано DARPA [11], а також Європейською комісією через програму Future & Emerging Technologies [12]. На сьогоднішній день можна виділити ряд ключових тематичних властивостей самосвідомих обчислювальних систем.

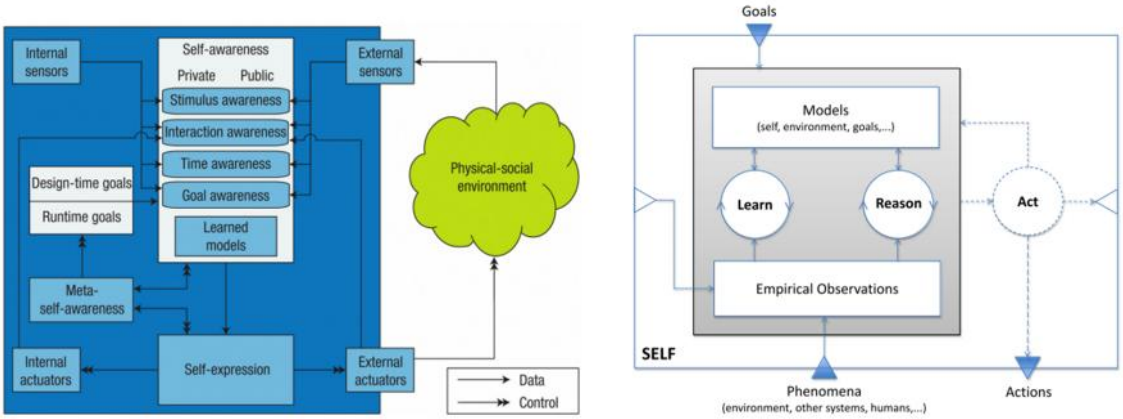

Рисунок 1.2 – Приклад еталонних архітектур самовідновлювальних систем

Самомоніторинг дозволяє швидше і точніше дізнатися, як на систему впливає операційне середовище і наскільки добре досягаються її цілі в зовнішньому середовищі.

Самомоделювання дозволяє системі мати доступ до чіткої інформації про обмеження, найкращого використання та практики всіх її компонентів, а також до актуальної інформації про стан системи.

Навчання дозволяє системі не тільки вивчати закономірності зовнішнього світу, але й те, наскільки добре її власні компоненти працюють у різних режимах роботи та в різних ситуаціях. У поєднанні з процесами міркування система також може навчитися постійно вдосконалювати процеси планування та поведінки.

Самоаналіз дозволяє системі використовувати різноманітні процеси міркування (включаючи спілкування з іншими системами та користувачами)

для діагностики будь-яких проблем чи аномалій, оптимізації чи покращення її продуктивності, планування адаптивної реакції на нову ситуацію тощо. Йому не потрібно чекати, поки інші системи чи користувачі скажуть йому, як інтерпретувати інформацію чи моделі, що відстежуються, і він може самостійно адаптувати свої реакції чи поведінку. Самоаналіз також включає самотестування та перевірку.

Самозвітування дозволяє системі інформувати користувача-людину та інші системи про результати свого моніторингу, обробки, навчання та аргументації. У складних системах це критична можливість, оскільки жоден зовнішній зонд не може бути настільки швидким або потенційно настільки точним.

Важливо відзначити, що самосвідомість не є бінарною властивістю.

1.3 Самовідновлювальні кіберфізичні системи

Концепція самосвідомості та можливості виконання самосвідомих обчислювальних систем можуть вирішити проблеми автономності для майбутніх CPS. Наприклад, можливість підтримувати розуміння власного стану та навколишнього середовища може дозволити марсоходу оцінити наслідки добре або частково працюючої хімічної камери або майбутнього шторму на Марсі. Крім того, виявлення інформації протягом більш тривалого періоду часу дозволяє створювати та вивчати знання, які можна використовувати для міркувань і прийняття рішень, наприклад, для конкретного робочого навантаження, яке завжди призводить до пікових температур на HMPSoC. З цією метою дані датчиків мають бути проаналізовані та абстраговані до рівня, на якому вони можуть бути пов’язані з контекстом. Контекст в кінцевому підсумку визначається цілями системи. Отже, система повинна прийти до розуміння того, як її поточний стан і середовище, враховуючи знання, отримані з досвіду, заважають або просувають її цілі.

В результаті об’єднання двох сфер кіберсистем і фізичних систем ми уявляємо самосвідомі CPS [13] з такими характеристиками:

- самосвідомий CPS розташовується у фізичному середовищі за допомогою взаємодії через датчики та виконавчі механізми та враховує своє конкретне фізичне розташування та перспективу в навколишньому середовищі;
- самосвідома CPS обмежена у ресурсах і оптимізує під час виконання для кількох і часто суперечливих цілей, таких як розмір, час, потужність обробки, енергія, вартість і температура;
- самосвідома CPS підтримує розуміння свого стану, включаючи функції та продуктивність своїх компонентів, таких як датчики та виконавчі механізми. Система знає відповідні внутрішні процеси, які визначають її поведінку, її реакції та її можливості. На основі цих знань він здатний виявляти недоліки та відхилення від очікуваної поведінки та продуктивності;
- самосвідома CPS підтримує розуміння свого середовища, включаючи навколишні об’єкти та інші системи, з якими вона взаємодіє. Система здатна виявляти відхилення від очікуваної поведінки та продуктивності середовища, і вона постійно вивчає та покращує своє розуміння середовища;
- самосвідома CPS підтримує відповідні знання щодо своїх цілей і цілей і використовує ці знання як ефективну основу для розподілу ресурсів та динамічного прийняття рішень. Система є історичною в тому сенсі, що вона відстежує поведінку, продуктивність і відхилення себе та навколишнього середовища з часом, щоб побудувати оцінку щодо недавнього минулого, довгострокового минулого або навіть усього життя;
- самосвідома CPS самозвітує, тобто може відстежувати та пояснювати свої рішення та дії.

Незважаючи на жорсткі обмеження ресурсів, багато CPS мали б значну обчислювальну потужність для реалізації деяких або всіх цих функцій, якщо б це було виконано ефективно. Навіть якщо поточні системи не повністю реалізують ці функції, вони все одно можуть кваліфікуватися як частково

самосвідомі. Фактично, навіть у майбутніх CPS очікується, що конкретні системи будуть лише частково самосвідомі залежно від потреб та обмежень програми. Тим не менш, здібності в цьому списку сприяють самосвідомості і повинні враховуватися в процесі проектування систем самосвідомості. У наступних розділах ми докладніше обговорюємо деякі з пов’язаних проблем і проблем, пов’язаних із проектуванням для самосвідомості.

1.4 Проблеми впровадження самосвідомлення у кіберфізичні системи

Самосвідомості в обчислювальних системах загалом і CPS зокрема є перспективним баченням, цей клас систем має певні унікальні характеристики, які роблять застосування концепцій самосвідомості нетривіальним. Є п’ять ключових проблем самосвідомісті:

- розвиток ресурсочутливої самосвідомості. Очевидно, що самосвідомість не приходить безкоштовно, але додаткові функції навчання та поведінки можуть заощадити більше загальних ресурсів, ніж потрібно для їх реалізації. З цієї причини важливо завжди включати вартість самих процесів самосвідомості в емпіричні оцінки самосвідомих систем (як приклад див. нещодавню роботу з самосвідомості в мобільних роботах). В інших випадках самосвідомість не призводить до зменшення загального використання ресурсів, а натомість отримує інші якості, такі як гнучкість або пояснення. У цих випадках важливо провести оцінку системи в контексті діапазону визначених вимог. Звичайно, для систем з вищим рівнем самосвідомості ми хочемо, щоб система сама збалансувала такі компроміси, приймаючи рішення щодо того, як розгорнути свої кінцеві ресурси для досягнення ефективного результату в компромісі між такими вимогами. Це призводить до першого завдання для дослідження самосвідомих CPS: формування чутливої до ресурсів самосвідомості;

- визнання ситуативності та суб’єктивності. Середовища змінюються як між системами, так і в певній системі з часом через такі фактори, як

мобільність, зміни клімату тощо. Знання CPS обов’язково є суб’єктивними, оскільки вони отримані з його власних сенсорів, пам’яті та процесів міркування. Це важливо в контексті як єдиної CPS, так і інтеграції кількох систем. Наприклад, огляд марсохода може бути заблокований пагорбом, і тому він може не знати про цікаві моменти. Однак вертоліт, що супроводжує, може мати безперешкодний і інший огляд. Це призводить до різного сприйняття через різні місця розташування. Це також стосується тимчасово та в межах окремого вузла. Наприклад, система, здатна міркувати не тільки про те, як її дані змінюються з часом, але й про перспективу, пов’язану з кожним читанням, означає, що вона може робити подібні інтегруючі судження. Це дозволяє системі робити висновки про свій стан у певний період часу. Більш довгострокові заяви цього типу можуть також ідентифікувати збої в сенсорному апараті, що спонукає людину до ремонту або реконфігурації системи у відповідь. Це також допомагає вирішити фізичні обмеження та вимоги до надійності. Тому наша друга проблема для самосвідомих CPS полягає в тому, що самосвідомість є суб’єктивною; його знання залежить від власної точки зору системи;

- створення інфраструктури для процесів самосвідомості. Під інфраструктурою маються на увазі механізми та засоби всередині CPS для забезпечення динамічної та адаптивної поведінки. Самосвідомий агент обмірковує компроміси та розподіляє ресурси там, де вони найбільше потрібні. Однак це значною мірою залежить від цілей і поточного стану системи. Оскільки важливість і пріоритети цілей можуть змінюватися залежно від ситуації, важливо, щоб система могла побудувати моделі цілей, які відображають ці цілі вищого рівня в поточному контексті та використовують різні типи міркувань і процесів навчання. Це також дозволяє «програмувати» системи зовсім по-іншому, оскільки система має здатність конструювати цілі нижчого рівня, можливо, проміжні цілі з вищого рівня. Це дозволяє описати бажану поведінку системи з точки зору цілей високого рівня, можливо, на рівні програми або стійкості системи. Звичайно, подібна діяльність сама по

собі буде коштувати певних ресурсів, тому важливе питання полягає в тому, які інфраструктурні процеси необхідні, щоб забезпечити систему такими можливостями;

- перевірка самосвідомості та встановлення гарантій. Можна очікувати, що довічна адаптація з боку CPS стане проблемою для їх прийняття. Як мінімум, користувачі вимагатимуть, щоб ці системи з процесами адаптації мали мінімальний набір поведінкових гарантій. Розробка CPS, усвідомлююча чи ні, для якої не можуть бути надані жодні гарантії, є невдачею. На щастя, самосвідомість і тут може допомогти. Оскільки система контролює та обмірковує свою поведінку та процеси адаптації, вона також може генерувати пояснення і, отже, зміцнювати довіру. Тим не менш, ця дискусія підкреслює притаманну проблему, яка існує у збалансуванні конкуруючих вимог постійного навчання та адаптації, а також встановлення передбачуваності та надійності.

1.5 Чутливі до ресурсів самовідновлювальні системи

Системи зазвичай обмежені у плані наявних у їхньому розпорядженні обчислювальних ресурсів, пам’яті, датчиків та комунікаційних ресурсів. Крім того, ці ресурси повинні підтримувати баланс із вимогами системи щодо її продуктивності, безпеки, захисту та гнучкості. Edge-IoT і медичні CPS, що носяться, більш сприйнятливі до таких обмежень ресурсів, і знаходження правильного балансу з їхніми вимогами може бути крихким. Тому самоусвідомлююча CPS повинна знати про свої загальні ресурсні обмеження, і зокрема про частку ресурсів, які можуть бути виділені для забезпечення властивостей самосвідомості. Один із підходів, який допомагає подолати цю ресурсну проблему, полягає у розвантаженні ресурсомістких процесів, таких як навчання чи міркування, на системи з великою кількістю доступних ресурсів, таких як хмарні сервери (наприклад, як у засланні [14]). Однак у деяких випадках це неможливо через інфраструктурні обмеження (наприклад,

відсутність постійного або досить тісного з’єднання з хмарними системами), а також через вимоги безпеки та захисту.

Іншим підходом до вирішення проблеми обмеженості ресурсів є наближені обчислення [15]. Наближені обчислення, які були широко вивчені в області обробки зображень [15] та ML [16], коли точні обчислення не потрібні або можна допустити певний рівень неточності [15]. Рішення наближених обчислень покликані перевершити свої точні аналоги за швидкістю, енергоспоживанням або розміром кремнієвої матриці. Деякі самоусвідомлюючі рішення, такі як References [16], відстежують навколишнє середовище (наприклад, характеристики каналу зв’язку) та його зміни та адаптують свій інтерфейс (в даному випадку його внутрішній опір) під час виконання, щоб відповідати навколишньому середовищу. Покращене узгодження імпедансу призводить до збільшення потужності, що подається в мережу, і більш сильного виявлення сигналу на стороні. Це покращує продуктивність усієї системи (тобто відношення сигнал/шум (SNR) та якість зв’язку). На відміну від цих рішень, де система намагається економити енергію, у статті [17] на основі обізнаності системи щодо поведінки користувача (передбачуваний час включення) система може дозволити нормальне або підвищене енергоспоживання, незважаючи на (щодо) низький заряд батареї.

Нині існує значний простір розробки мінімальних, але значних і багаторазово використовуваних форм обчислювального самосвідомості, які потрібні для систем, дуже обмежених ресурсах. Справді, цікаво запитати, як може виглядати «мінімальна» самосвідомість. Інше важливе і досить відкрите питання дослідження в цьому відношенні – як виміряти обмеження системи, а також вартість різних процесів і функціональних можливостей, включаючи самосвідомість. Навіть якщо в деяких випадках механізми вимірювання можуть бути дуже простими та зрозумілими, у деяких випадках вони становлять значні труднощі.

2 УЗАГАЛЬНЕНА ТЕОРЕТИКО-МНОЖИННА МОДЕЛЬ КФС

Інтернет речей і кіберфізичні системи включають взаємодіючі: людини (в різних ролях), а також логічні, фізичні, перетворювальні компоненти, розроблені для функціонування за допомогою інтегрованої логіки та фізики. Таким чином, уніфікована модель компонентів передбачає 4 категорії компонентів КФС та IoT: логічні, фізичні, перетворювачі та компоненти за участю людини.

На логічному рівні КФС процеси описуються мовою формальної логіки та реалізуються за допомогою стандартних технологій збору, перетворення та зберігання інформації в інформаційно-комунікаційних системах. Фізично, як правило, розглядається реалізація КФС, розроблена або адаптована для взаємодії з очікуваним операційним середовищем для досягнення однієї або кількох передбачуваних цілей при дотриманні обмежень, що накладаються на систему. Сполучною ланкою між логічним і фізичним рівнями є перетворювачі – різні сенсорні пристрої, датчики, що збирають дані про фізичний стан кіберфізичного оточення, інтерпретація яких може використовуватися для зміни логічного стану системи, а також виконавчі механізми, здатні впливати на фізичний стан такого середовища. Саме перетворювачі займають у КФС центральну роль, забезпечуючи взаємодію фізичних та логічних компонентів. Зазначимо також, що присутня на малюнку фігура людини може представляти різні класи користувачів (стейкхолдери, системні адміністратори, користувачі, обслуговуючий персонал тощо), що взаємодіють із системою як на інформаційному, так і фізичному рівні. Таким чином, представлені елементи відображають чотири основні категорії компонентів КФС: логічні компоненти, фізичні компоненти, користувачі, а також перетворювачі, що включають датчики і виконавчі механізми.

Відповідно до наведеного вище визначення, кіберфізична система CPS може бути укрупнено представлена наступним кортежем параметрів

$$C \quad =< Ph, L \ , S \quad , A \ , H \quad >, \tag{2.1}$$

де Ph – множина фізичних компонентів;

Lg – множина логічних компонентів;

Sens – множина датчиків;

Act – множина виконавчих механізмів;

Hum – множина осіб, залучених у процеси функціонування КФС або що знаходяться всередині кіберфізичного оточення та є кінцевими користувачами системи.

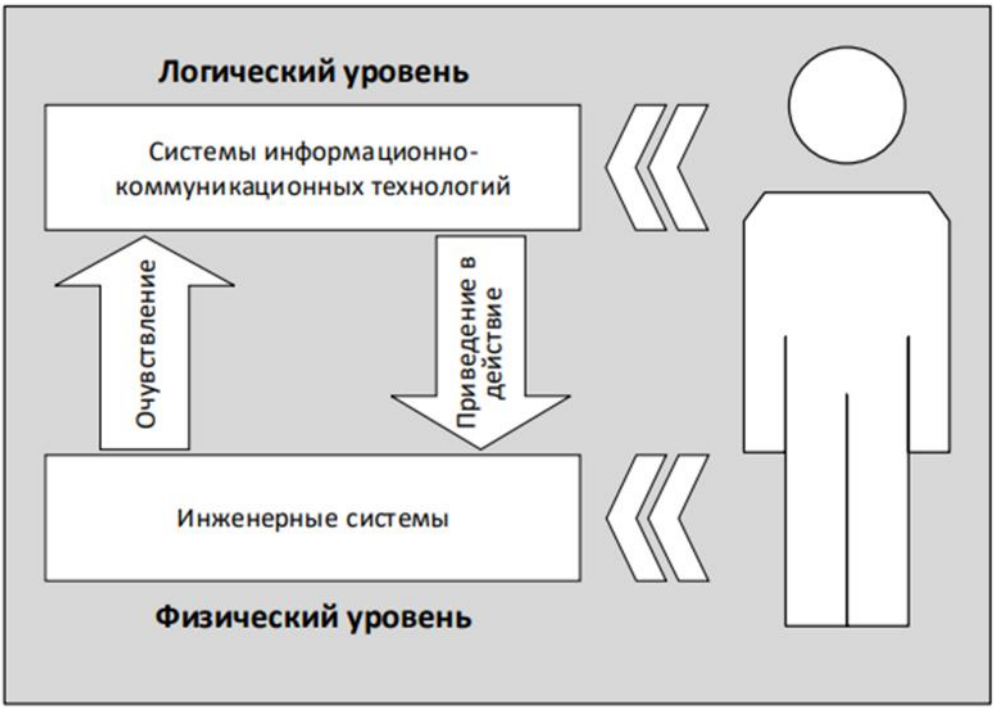

Рисунок 2.1 – Узагальнена модель компонентів КФС

До фізичних компонентів Ph відносяться різноманітні інженерні модулі, розміщені в середовищі функціонування КФС, апаратне забезпечення та його компоненти, а також мережна інфраструктура фізичного рівня. Зазначимо також, що протікають у даних компонентах процеси мають виключно фізичний характер і включають різноманітні процеси з перетворення і передачі енергії.

Логічні компоненти Lg включають всі рівні програмного забезпечення (вбудоване ПЗ, обробники, драйвери, операційні системи та програми), мережі та мережеве оточення логічного рівня, а також інформацію на рівні даних, інформації та знань. Відповідно протікають у даних компонентах процеси є переважно інформаційними і включають різні процеси по збору, агрегації, обробці та аналізу даних, а також процеси створення, отримання, зберігання, пошуку та поширення інформації.

Датчики Sens та виконавчі механізми Act, відповідно, є вхідними та вихідними перетворювачами відповідно. Датчики збирають інформацію про фізичний стан середовища для подальшої обробки на рівні логічних компонентів, у той час як виконавчі механізми змінюють його в результаті отримання керуючих сигналів з боку Lg.

Модель функціонування датчиків Sens може включати опис процесів передобробки сигналу, аналого-цифрового перетворення, калібрування, дискретизації і зіставлення метаданих. Актуатори можуть бути представлені широким рядом механічних пристроїв, починаючи від вібромотора в смартфоні або фітнес-пристрої до складних промислових маніпуляторів.

Взаємодія людей з компонентами КФС включає проектування, експлуатацію, забезпечення систем, а також взаємодію з ними та їх безпосереднє використання. Так, людина може бути як користувачем системи Hum_U, так і її компонентом Hum_C або елементом зовнішнього середовища Hum_{Env}. Таким чином, безліч Hum включає три інші множини різної потужності, що характеризують все різноманіття користувачів

$$H = \{H_C, H_U, H_E\}. \quad (2.2)$$

У контексті функціонування КФС визначимо безліч функціональних компонентів КФС F як сукупність логічних та фізичних компонентів системи, які при взаємодії формують базову одиницю функціональності

$$F = \{\langle Ph_i, L_j \rangle | Ph_i \in Ph, L_j \in L, i = \overline{1,n}, j = \overline{1,m}\}. \quad (2.3)$$

Передбачається, що кожен функціональний компонент КФС повинен мати наступні можливості:

- наявність обчислювальних потужностей та програмного забезпечення, необхідних для реалізації власних функцій;
- наявність достатнього об’єму пам’яті для зберігання всіх необхідних
- для власної діяльності даних;
- здатність встановлювати мережеві з’єднання з іншими компонентами системи та забезпечувати цільову передачу даних;
- здатність до отримання та збору необхідної інформації про стан навколишнього середовища та інших компонентів системи;
- здатність до самодіагностики в контексті виявлення власної несправності, а також до інформування пов’язаних компонентів у разі виникнення такої несправності.

Кожен функціональний компонент асоційований з певним набором датчиків, виконавчих механізмів та людей. Такі набори разом утворюють безліч перетворювачів функціональних компонентів Tr:

$$T = \{S_i, A_j, C_k | S_i \in S, A_j \in A, C_k \in C, i = \overline{1,n}, j = \overline{1,m}, k = \overline{1,l}, \quad (2.4)$$

що дозволяє функціональним компонентам здійснювати різного роду перетворення на фізичному та логічному рівнях у контексті взаємодії із зовнішнім середовищем та користувачами КФС, включаючи як обмін та перетворення енергії (механічної, електричної тощо) – на фізичному рівні, так і обмін, перетворення та синтез інформації – на логічному рівні. Таким чином, для будь-якої КФС може бути визначено відображення

$$f \square F \rightarrow T \ , \tag{2.5}$$

а КФС відповідно може бути охарактеризована множиною упорядкованих пар виду

$$CP = \{\langle F_i | T_i \rangle | F_i \square F, T_i = f(F_i), i = \overline{1,n}\}. \tag{2.6}$$

Слід зазначити, що елементи будь-якого перетворювача потенційно можуть бути інтероперабельними, як наслідок, набори елементарних компонентів КФС, що входять до складу деякого перетворювача, можуть входити до складу необмеженої кількості перетворювачів інших функціональних компонентів.

Узгоджена взаємодія функціональних компонентів КФС дозволяє забезпечити надання сервісів КФС S_{CPS} споживачам Cons, які можуть виступати як інші функціональні компоненти, так і люди:

$$C = Fl \ , \tag{2.7}$$

Варто також відзначити, що багато споживачів у загальному випадку можуть відрізнятися для кожного окремого сервісу КФС:

$$C_S = \{C_i | C_i \square C, i = \overline{1,n}\}, \tag{2.8}$$

Введемо відображення g:

$$g: S_S \rightarrow C \ , \tag{2.9}$$

Оскільки послуги КФС реалізуються за допомогою узгодженої взаємодії різних наборів функціональних компонентів, то також може бути визначено наступне відображення h:

$$h: C \quad \rightarrow S_C \quad , \tag{2.10}$$

Таким чином, композиція відображень h∘g дозволяє встановити відповідність між споживачами сервісів (чи користувач або інший функціональний компонент КФС) і функціональними компонентами, необхідними для реалізації відповідних сервісів КФС.

Розглянемо узагальнену динамічну модель функціонування КФС. Введемо, відповідно, узагальнений вектор стану КФС та вектор вихідних характеристик КФС, а також узагальнений вектор відповідно:

$$\begin{aligned} \overline{x}(t) &= \| x_F^{-T}, x_T^{-T} \|^T, \\ \overline{y}(t) &= \| y_F^{-T}, y_T^{-T} \|^T. \end{aligned} \tag{2.11}$$

Допустим, некоторое изменение физической среды $\xi_{Ph}(t)$ приводит к изменению n различных физических параметров P_t :

$$P_{t,} = \{p_{t,} \in P_t | \xi_{Ph}(t) \rightarrow P_t, 1 = \overline{1,n}\}, \tag{2.12}$$

і реєструється датчиками, що сукупно формують набір $Sens_k$:

$$S_{t} = \{S_{t} \in S_{i} | P_t \rightarrow S_{t}, j = \overline{1,m}\}. \tag{2.13}$$

Оскільки кожен датчик $Senst_j$ з набору датчиків $Sens_t$, що реєструють зміну параметра p_{tk}, може входити в одне або відразу кілька підмножин виду $Sens_i$, таких що $Sens_i \in Tr_i$, i = 1,n, відповідно реєстрація змін датчик $Sens_{tj}$ призводить до зміни станів всіх функціональних компонентів $F_{tj} \in F$ з перетворювачами $Tr_{tj} \in Tr$ асоційований даний датчик.

Враховуючи те, що з кожним із перетворювачів функціональних компонентів Tr_i, i = 1,n, , асоційований цілий набір датчиків $Sens_i$, то

оновлений стан $\bar{x}_F(t_1)$, кожного з функціональних компонентів множини F визначається виключно на основі лише тих змін фізичних параметрів середовища P_{tk}, які були зареєстровані датчиками, що належать до набору $Senst_i$. При цьому набір $Senst_i$ входять лише ті датчики, які:

- зареєстрували зміну зовнішнього середовища ($Senst_i$ □ $Sens_t$);
- належать до набору датчиків, що входить до складу відповідного перетворювача функціональних компонентів Tr_i ($Sens_{ti}$ □ $Sens_i$):

$$S_{t_i} = S_t | S_i, \tag{2.14}$$

Відповідь системи зміни інформаційного оточення $\xi_{Lg}(t)$ багато в чому аналогічна відповіді зміни у фізичному середовищі. Допустимо, зміна інформаційного оточення $\xi_{Lg}(t)$ призводить до зміни p логічних параметрів L_t:

$$L_t = \{l_t \in L_t | \xi_L(t) \to L_t, \ j = \overline{1,p}\}, \tag{2.15}$$

та реєструється логічними компонентами КФС Lg_t □ Lg. Оскільки кожен логічний компонент Lg_{tj} з набору Lg_t може входити в одне або відразу кілька підмножин виду Lg_i, таких що Lg_i □ Fi, i = 1,n, відповідно реєстрація змін компонентом Lg_{tj} призводить до зміни станів всіх функціональних компонентів F_{tj} □ F з якими асоційований даний компонент.

Враховуючи те, що з кожним з функціональних компонентів F_i асоційований цілий набір логічних компонентів Lg_i, то оновлений стан $\bar{x}_F(t1)$ кожного з функціональних компонентів множини F визначається виключно на основі змін логічних параметрів L_{ti}, зареєстрованих компонентами, що належать до набору Lg_{ti}, до якого входять тільки ті логічні компоненти, які:

- зареєстрували зміну логічних параметрів (Lg_{ti} □ Lg_t);
- належать до набору логічних компонентів, що входить до складу відповідного функціонального компонента F_i (Lg_{ti} □ Lg_i):

$$L_{t_1} = L_{t}|L_{i}, \quad (2.16)$$

Зазначимо, що в обох випадках зміна стану і-го функціонального компонента F_i призводить до опосередкованої зміни станів фізичних та логічних компонентів системи, що належать до наборів Ph_i та Lg_i відповідно. Крім того, перехід іго функціонального компонента стану $\bar{x}(t_0)$ у стан $\bar{x}(t_1)$ може індукувати зміну вихідних характеристик системи $\bar{y}(t)$, зокрема, за допомогою активації виконавчих механізмів з набору Act_i □ Tr_i.

3 РОЗРОБКА МОДЕЛІ CPS

У цьому розділі представлені різні властивості, які є особливо важливими для систем CPS. А також розроблена модель CPS з використанням цих властивостей.

3.1 Самоадаптація

У дуже динамічному контексті CPS адаптація є важливою властивістю від фізичного до прикладного рівня. Вона дозволяє взаємодіючим вузлам, а також сервісам, що їх використовують, своєчасно реагувати на постійно мінливий контекст відповідно до високорівневих цілей, наприклад, бізнес політики або цілей продуктивності, які визначаються людьми. Однак на етапі адаптації участь людини має бути зведена до мінімуму, щоб впоратися з передбачуваним великим масштабом і реактивністю. Тому CPS повинна бути здатна розмірковувати автономно і приймати рішення, що самоадаптуються. Когнітивні радіостанції на фізичному і канальному рівнях, мережеві протоколи, що самоорганізуються, автоматичне виявлення послуг і (повторне) зв’язування на прикладний рівень є важливими факторами для самоадаптованих CPS.

3.2 Самоорганізація

У системах CPS – і особливо у WS&AN – дуже часто вузли приєднуються та залишають мережу спонтанно. Тому мережа повинна бути здатна реорганізовуватися відповідно до цієї мінливої топології. Самоорганізовані протоколи маршрутизації мають велике значення в CPS, щоб забезпечити безперебійний обмін даними в цих неоднорідних мережах. Через велику кількість вузлів переважно розглядати рішення без центральної

точки управління, наприклад, кластерні підходи. При роботі над самоорганізацією також дуже важливо враховувати енергоспоживання вузлів та виробити рішення, що максимізують термін служби системи CPS та ефективність зв’язку всередині цієї системи.

3.3 Самооптимізація

Оптимальне використання обмежених ресурсів (таких як пам’ять, пропускна здатність, процесор і, найголовніше, потужність) пристроїв CPS потрібне для стійкого та довговічного розгортання CPS. Враховуючи деякі високорівневі цілі оптимізації з точки зору продуктивності, енергоспоживання або якості обслуговування, система сама повинна виконувати необхідні дії для досягнення поставлених цілей.

3.4 Самоконфігурація

Системи CPS потенційно складаються з тисяч вузлів і пристроїв (наприклад, датчиків та виконавчих механізмів), що робить конфігурацію системи дуже складною і важкою вручну. Система CPS повинна надавати засоби віддаленої конфігурації, щоб програми для самостійного керування автоматично конфігурували необхідні параметри на основі потреб додатків і користувачів. Це включає, наприклад, налаштування параметрів пристрою та мережі, встановлення/видалення/оновлення програмного забезпечення, або налаштування параметрів продуктивності.

3.5 Самозахист

Через бездротову та повсюдну природу CPS буде вразливою для численних шкідливих атак. Оскільки CPS тісно пов’язана з фізичним світом, атаки будуть спрямовані, наприклад, на контроль фізичного середовища або

отримання приватних даних. CPS має автономно налаштовуватися на різні рівні безпеки та конфіденційності, не впливаючи при цьому на якість обслуговування та якість досвіду.

3.6 Самоопис

Пристрої та ресурси (наприклад, датчики та виконавчі механізми) повинні бути в змозі описати свої характеристики та можливості у виразним чином, щоб дозволити іншим взаємодіючим об’єктам взаємодіяти з ними. Адекватні формати та мови опису пристроїв та послуг повинні бути визначені адекватними форматами та мовами опису пристроїв та послуг, можливо на семантичному рівні. Існуючі мови повинні бути адаптовані, щоб знайти компроміс між виразністю, спільністю та розміром описів. Самоопис є фундаментальною властивістю для реалізації plug-and-play ресурсів та пристроїв.

3.7 Самопізнання

Поряд із самоописом, функція самовиявлення відіграє важливу роль для успішного розгортання CPS. Пристрої/послуги CPS повинні бути динамічно виявлені та використовуватися іншими безперешкодно та прозоро. Тільки потужні та виразні протоколи виявлення пристроїв та послуг протоколи (разом із протоколами опису) дозволять системі CPS бути повністю динамічною. CPS-система може бути повністю динамічною (з погляду топології).

3.8 Самозабезпечення енергією

Самозабезпечення енергією є надзвичайно важливою (і дуже специфічною для CPS) особливістю реалізації та впровадження стійких рішень CPS. Методи збору енергії (сонячної, теплової, вібраційної і т.д.) слід

віддати перевагу як основному джерелу живлення замість батарей, які необхідно регулярно замінювати, і які негативно впливають на навколишнє середовище.

3.9 Самовідновлення

Метою цієї властивості є виявлення та діагностика проблеми у міру їх виникнення та негайно намагатися усунути їх автономним способом. Системи CPS повинні відстежувати постійно відстежувати стан різних вузлів і виявляти, коли вони поводяться інакше, ніж очікувалося. Потім вона може виконувати дії для усунення проблем (наприклад, змінюючи параметри конфігурації або встановлення оновлення програмного забезпечення).

Самовідновлення – це процес виявлення та відновлення після збоїв у динамічних системах (на відміну від традиційної відмови стійкості, яка налаштовується під час проектування і тому не може впоратися з несподіваними збоями). Самовідновлення адаптує систему під час виконання для пом’якшення наслідків збоїв. Адаптація – складний процес, який можна узагальнити чотирма абстрактними кроками:

- збирання інформації про навколишнє середовище та визначення внутрішніх властивостей системи (оцінка стану)[18];
- аналіз спостережень (виявлення збоїв);
- вирішити, як адаптуватися, щоб досягти бажаного стану (знайти стратегію відновлення, наприклад, через ORR [6]) та, нарешті, iv) діяти (відновлювати).

У цій роботі ми фокусуємося на інтерфейсі служби адаптації (зокрема, збирання та дія), її вимогах та проектуванні архітектури системи, що дозволяє це зробити (рис. 3.1).

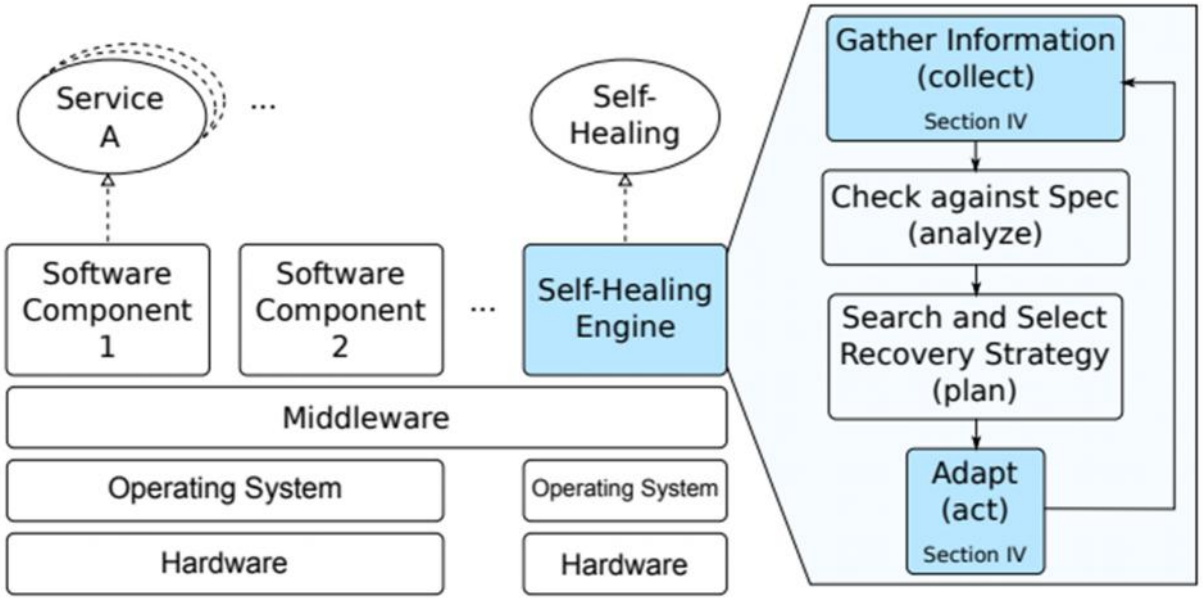

Рисунок 3.1 – Приклад архітектури системи, що самовідновлюється

Системи часто поділяються на кілька підсистем, з’єднаних комунікаційною мережею, що є самостійною підсистемою. Далі кожна підсистема може включати різні апаратні та/або програмні компоненти (див. рис. 1). Компонент розглядається як «чорна скринька» з чітко визначеними інтерфейсами. Кожен компонент надає послуги, які можуть залежати від інших компонентів. Сервіс – це передбачувана поведінка системи/компонента. На рисунку 3.1 зображено сервіс, що самовідновлюється. Він реалізований декількома компонентами виявлення збоїв (збір та аналіз) та відновлення (планування та дії) компонентів відповідно. Підсистеми можуть залежати один від одного, наприклад, операційна система потребує апаратного забезпечення для роботи.

Підсистеми та компоненти використовуються для опису ієрархічної структури системи, в основному для того, щоб упоратися зі складністю системи. Понад те, поведінка системи часто описується сервісами та його взаємодією, в порівнянні сервіс-орієнтована архітектура (SOA), де фізична структура опускається. Зазвичай, послуги взаємодіють друг з одним у вигляді повідомлення.

Відмова – це подія, що призводить до відхилення сервісу специфікації. Відмовлення зазвичай відстежуються на інтерфейсі компонента. Компоненти можуть містити механізм виявлення помилок та додатково пригнічувати помилкові виходи. Такі компоненти називаються fail-silent. Деякі компоненти можуть автоматично припиняти виконання збоїв, так звані fail-stop компоненти. Однак помилковий компонент може видавати неправильні вихідні дані, які можуть призвести до відмови інших сервісів.

Розрізняють структурну та параметричну адаптацію під час виконання. Структурна адаптація змінює структуру системи, тобто. додає, видаляє чи переставляє компоненти. Як наслідок, вона додає та видаляє послуги компонентів. Під структурною адаптацією також розуміється зміна інформаційного потоку між сервісами, тобто. ми припускаємо, що взаємодії між сервісами є частиною структури системи. На відміну від цього параметрична адаптація змінює поведінку компонента шляхом адаптації його параметрів. Ці два типи адаптації можна комбінувати. У той час як параметрична адаптація реалізується самим компонентом, структурна адаптація виконується в рамках підсистеми або всієї системи і має бути включена в архітектуру системи (підсистеми).

Системи можуть бути охарактеризовані атрибутами. Атрибути можуть бути використані для відмінності систем один від одного. Оцінений атрибут називається властивістю. Послуги надають чи споживають властивості, тобто, відбувається обмін інформацією задля встановлення функціональності чи поведінки. Отже, властивості CPS часто знаходяться по відношенню один до одного. Всі відносини властивостей системи можуть зберігатися в онтології.

ORR фокусується на структурній адаптації в сервіс-орієнтованих системах реального часу. Зокрема, коли один із сервісів системи реального часу виходить з ладу, ORR може розробити заміну сервісу, що відмовив. Алгоритм заміни використовує базу знань базу знань, яка визначає взаємозв’язок властивостей у CPS, що називається онтологією, та додаткову інформацію про час виконання CPS. Динамічні зміни, такі як додавання,

видалення або адаптація сервісів під час виконання, моделюються за допомогою таблиці Serviceto-Ontology-Mapping (SOM), яка пов’язує властивості, оброблювані сервісом кіберчастини (наприклад, значення датчика) з властивостями, визначеними в онтології. З іншого боку, таблиця містить тимчасову специфікацію взаємодії сервісів (наприклад, швидкість, з якою компонент датчика оновлює свої виміри). Працюючі послуги та його опис, включаючи вхідні і вихідні характеристики, можна отримати з реєстру сервісів. SOM повинна оновлюватися щоразу, коли сервіс додається, видаляється або змінюється.

Щоб упоратися з непередбачуваністю та еволюційністю таких систем, забезпечуючи при цьому їхню працездатність, сама система має бути адаптивною. Традиційна стійкість до відмови, як правило, обмежується несправностями і загрозами, врахованими під час проектування. Наприклад, простою стійкістю до відмови може бути голосування за результат резервування компонентів або скидання завдання при помилці. Проте, щоб залишатися надійною та безпечною навіть при виникненні несправностей та загроз, які не були враховані або передбачені під час проектування, але можуть виникнути під час виконання системи необхідні засоби для самовідновлення. Самовідновлення є перспективним підходом для боротьби з непередбаченими несправностями, що потребує поняття самосвідомості та самоадаптації.

Самовідновлення розвинулося з традиційної стійкості до відмови шляхом додавання автономності та штучного інтелекту. Тому процес є більш складним, ніж традиційна стійкість до відмов, і може бути розділений на кілька частин. Наприклад, MAPE-K – це загальна структура для самоадаптації, спочатку розроблена для автоматизації (будь-яких) завдань. Вона виражає цикл зворотного зв’язку – моніторинг, аналіз, планування і виконання з використанням деяких знань – системи, що самоадаптується. У суміжних роботах етапи можуть бути перейменовані (наприклад, збір, аналіз, планування та виконання у дорожніх картах самоадаптації) або узагальнені

(наприклад, моніторинг, діагностика та відновлення у дослідженнях самовідновлення).

Більшість сучасної літератури з самовідновлення можна розділити на огляди з виявлення аномалій, моніторингу під час виконання, отказоустойчивости, динамічної реконфігурації і самоадаптації. Питання застосування цих методів до CPS може звузити коло потенційних підходів. Деякі властивості CPS, зокрема, продуктивність у реальному часі та критичність до безпеки, вимагають особливого підходу та обережності. Проте, можливості підключення, що забезпечуються IoT, також можуть відкрити нові можливості (наприклад, більше речей, більше інформації). Дуже перспективним є самовідновлення шляхом структурної адаптації (SHSA). SHSA замінює сервіс, що відмовив, на інший, використовуючи неявне резервування. На відміну від явної надмірності, яка досягається шляхом дублювання критично важливих системних сервісів та голосування за виходи (рисунок 3.2), частина інформації, що відмовила, може бути отримана і зі зв'язаної інформації (рисунок 3.3). Наприклад, інформація a, що надається службою A, може бути замінена на a 0 яка виходить з комбінації b і c.

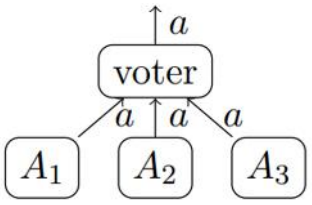

Рисунок 3.2 – Явне резервування

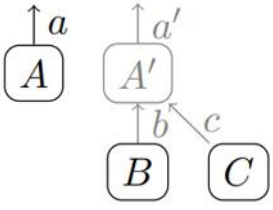

Рисунок 3.3 – Неявне резервування

Система, що проектується, може не включати (неявне) резервування відразу (якщо це так, система буде використовувати об’єднання інформації для використання резервування), проте при складанні (під)систем від різних постачальників резервування часто стає доступним. Зазвичай це стосується розподіленої системи або системи систем, наприклад, (підключеного) транспортного засобу або в цілому до CPS, підключеного через IoT. Основи цієї роботи закладено Хьофтбергером[19], який розробив основу знань (у вигляді онтології), що описує неявне резервування у системі. Дана робота застосовує та покращує роботу Хьофтбергера з самовідновлення з використанням надмірності.

SHSA – це загальносистемний підхід, тобто. він обмежується одним компонентом. Він використовує доступну інформацію, що передається по комунікаційній мережі системи, для створення заміни сервісу, що відмовив під час виконання. Отже, можливості мережі та адаптивність системи відіграють вирішальну роль.

Компоненти CPS часто припускають, що спостереження проводяться у загальний час, тобто синхронно. Наприклад, компоненти об’єднання датчиків та оцінки стану періодично відбирають та об’єднують вимірювання, щоб покращити передбачуваний стан фізичної системи. Представлені в цій роботі замінники, а також традиційні монітори та виборці (наприклад, потрійна модульна надмірність (TMR)) також покладаються на синхронні спостереження. Однак великі та розподілені мережі датчиків або IoT часто використовують асинхронні парадигми зв’язку (тобто вимірювання передаються через повідомлення, які не синхронізовані за часом) і можуть страждати від затримки, джиттера або навіть втрати пакетів. Більш того, різні датчики надають свої вимірювання з різною швидкістю.

SHSA вимагає запуску з боку монітора або пристрою для виявлення збоїв. Збої зазвичай виявляються шляхом вказівки бажаної поведінки у вигляді моделей, сигнатури або специфікації. Простий метод, пов’язаний з SHSA, для виявлення значення несправного датчика у різних сценаріях відмови полягає

у перевірці за пов’язаними джерелами інформації. На жаль, інформація, що передається в системі, страждає від невизначеності у просторі (наприклад, шум) та часу (наприклад, затримки). Тому підписку та отриману інформацію не можна порівнювати безпосередньо.

Динаміка CPS зазвичай вимагає швидкості вибірки та обробки ≤ 100 мс (наприклад двигуни, автономні транспортні засоби). Тому бажано пом’якшити наслідки збою протягом такого часу. Хефтбергер [19] застосував простий глибинний пошук (DFS) для пошуку пов’язаної інформації у базі знань у разі, якщо якась інформація не працює. Перший знайдений можливий зв’язок використовується для створення заміни. Однак велика кількість доступної інформації в IoT може уповільнити пошук заміни. Крім того, створення заміни або відключення існуючої служби може тривати значний час. Тому бажано швидко знайти і відразу ж запустити найкращу чи оптимальну заміну.

У разі більш ніж однієї можливої заміни, SHSA потрібні засоби для оцінки та вибору оптимального рішення для програми. Крім того, SHSA має подбати про те, щоб рішення задовольняло вимогам програми, наприклад, щодо безпеки.

Оскільки термін «самовідновлевлення» відносно новий, термінологія була уточнена спочатку шляхом вивчення робіт про надійність і стійкість, а потім про відмову стійкість, самовідновлення та самоадаптації в CPS та програмної інженерії для отримання пошукових рядків та пов’язаних робіт з цієї теми. Урок, отриманий у ході безперервного пошуку відповідних робіт, полягає в тому, що самовідновлення – це широка область, що включає виявлення, діагностику та відновлення, у кожній з яких можуть використовуватись різні моделі, інструменти та підходи. Наприклад, виявлення несправності може бути досягнуто шляхом вказівки допустимої сигнатури, моделювання аномалії або перевірки надмірність. Крім того, підходи різняться залежно від мети (надійність проти безпеки), моделі збою (шум проти стратегічної атаки) або області (CPS проти хмарних сервісів). Грунтуючись на роботі Хефтбергера[19], подальші сучасні дослідження були

зосереджені на моделях та методах для SHSA на основі надмірності в CPS. Оскільки моделі та методи самовідновлення зазвичай виходять з II або автономних обчислень (AC), вимоги CPS були враховані в ході досліджень і розробок, представлених у даній роботі. Наприклад, CPS може бути обмежена в ресурсах і реалізовувати програми реального часу та критично важливі для безпеки. Однією з цілей даної роботи було застосування та інтеграція SHSA у CPS та демонстрація застосовності SHSA у CPS. Для цього була потрібна підходяща платформа і як мінімум один комплексний приклад використання.

Щоб створити відповідну платформу для SHSA у CPS, було чітко визначено вимоги SHSA до архітектури. На основі цих вимог було проведено пошук платформ та застосовних основ, включаючи управління процесами та комунікаціями (проміжне ПЗ, протоколи тощо). Іншим важливим аспектом при виборі платформи, що дозволяє використовувати SHSA, був випадок використання, як обраний мобільний робот. Тому в прототипах використовувалася Robot Operating System (ROS), система, що широко використовується в робототехніці, яка відповідає всім вимогам SHSA і надає необхідні функції для створення прототипів і подальшого розвитку (наприклад, інструменти для налагодження та візуалізації програми, інтерфейси до інших систем). Архітектура механізму була обрана на основі суміжних робіт у галузі самоадаптації, AC, II або CPS. Наприклад, процес і компоненти механізму самовідновлення були розроблені відповідно до Monitor, Analyze, Plan and Execute using a Knowledge base (MAPE-K), що добре зарекомендувала себе структурою автоматизації завдань і широко використовуваної у співтоваристві AC.

Відправною точкою для моделі надмірності послужила база знань, визначена Хефтбергер [19]. База знань моделює відносини між змінними CPS. Оскільки змінні CPS є випадковими величинами, були розглянуті ймовірні моделі, наприклад, байєсовські або марковські мережі. Проте підстановка складає основи правил, визначених відносинами між змінними. Щоб вибрати правильну підстановку, підстановка та її показники ефективності мають бути

формально визначені. Моделі та методи були досліджені на предмет їх придатності для розширення бази знань, також з урахуванням різних угруповань, наприклад, верифікації (наприклад, підходи, засновані на контрактах), прийняття рішень та штучного інтелекту (наприклад, теорія корисності). Для методів була застосована та сама процедура. Підходи з використанням правил, графів, ймовірнісних моделей, штучного інтелекту та верифікації були досліджені, відфільтровані та об’єднані для застосування самовідновлення у CPS.

Тестування і оцінка самовідновлення CPS вимагає даних від фізичних об’єктів, тобто. наборів даних (наприклад, отриманих у мережі датчиків), симуляторів або прототипів, що надають зразки з часом. Тому було проведено онлайн-пошук відповідних симуляторів або еталонів та наборів даних на таких платформах, як UCI Machine Learning Repository або Kaggle. На жаль, у доступних наборах даних на різних платформах відсутні дані датчиків, що вимірюють різні фізичні об’єкти. Наявні набори даних часто є дискретними даними (наприклад, для класифікації), витягують характеристики з одного датчика (наприклад, датчики зображень) або збирають дані реплік датчиків (наприклад, температура на різних ділянках). Наскільки мені відомо, також немає (відкритих або безкоштовних) симуляторів, що надають відповідні сигнали, які можна було б використовувати для оцінки. Більш того, основна частина цієї роботи включає розробку архітектури та реалізацію самовідновлення спостережних можливостей CPS на основі резервування, що краще оцінювати на реальному прототипі CPS. Тому в цій дисертації проводиться оцінка розроблених рішень на реальному прототипі CPS мобільного робота (замість того, щоб вкладати ресурси у реалізацію симулятора). Робот складається з різних датчиків та обчислювальних блоків, що утворюють гетерогенну систему систем, подібну до IoT. Крім того, там, де це застосовно та/або необхідно, були створені штучні дані, наприклад, для перевірки складності у просторі та часі реалізацій.

Представлені основи знань у прикладах використання можуть бути

досить невеликими. З одного боку, це забезпечує швидкий результат підстановки, з іншого боку, SHSA може бути додатковими накладними витратами. Однак SHSA надає спосіб моделювання та використання надмірності, який може бути розширений, адаптований та повторно використаний. Також зазначимо, що представлені бази знань орієнтовані на додаток та обмежені прототипами. У цій роботі представлені вимоги, рекомендації, формальні визначення та алгоритми SHSA. Вибір платформи для SHSA або окремих її реалізацій може вимагати перегляду з появою нових архітектур та фреймворків, наприклад, вибір ROS як платформи для SHSA або обробки асинхронних, багатошвидкісних або відкладених вимірювань відповідно. У той же час є друга версія ROS (ROS 2.0), і промисловість також працює над проміжними програмами та стандартами для надійної передачі повідомлень у CPS (наприклад, Industrial IoT або Time-Synchronized Network (TSN) у стандарті Ethernet). Тим не менш, в IoT та бездротових сенсорних мережах робота з тимчасового вирівнювання, як і раніше, актуальна. SHSA була розроблена для пом’якшення відмов служб спостереження. Відмови контролерів та виконавчих механізмів у цій роботі не розглядаються. Збої виявляються на виході сервісів. За наявності відповідного опису або семантики виходів контролерів база знань SHSA може обробляти надмірність контролерів. Більшість CPS реалізують один контролер, специфічний їх застосування (виняток є область авіоніки, де застосовується TMR чи явне резервування). У IoT, яка включає кілька додатків CPS, кожна з яких оснащена окремим контролером, моделювання надмірності виходів контролерів може забезпечити відмовостійкість і цих компонентів. Це вимагає подальших досліджень та стандартизації семантичного опису інтерфейсів сервісів. Можливою відправною точкою є онтологія IoT, наприклад, SensorML. Вихід виконавчого механізму управляє фізичним об’єктом і більше не є інформацією, що передається у комунікаційній мережі. Оскільки SHSA замінює інформацію, що відмовила в мережі, відмови приводів не можуть бути пом’якшені SHSA.

CPS зазвичай збираються з кількох підсистем, з’єднаних комунікаційною мережею (наприклад, IoT), яка сама собою є підсистемою. Кожна підсистема включає різні апаратні та/або програмні компоненти. Компонент – це підсистема, що розглядається як «чорна скринька» з чітко визначеними інтерфейсами. Кожен компонент надає послуги, які можуть залежати від інших компонентів. Сервіс – це передбачувана поведінка системи/компонента. Як правило, послуги взаємодіють один з одним за допомогою повідомлень. Сервіси CPS реалізують цикл зворотного зв’язку, спостерігаючи та контролюючи своє оточення. Як наслідок, повідомлення, які передаються в рамках CPS, зазвичай містять інформацію, обчислену контролерами або зібрану датчиками, що вимірюють фізичні об’єкти – змінні CPS. Зокрема, служби спостереження публікують (зазвичай періодично) значення та тимчасову мітку змінної v CPS – спостереження або знімок змінної у певний момент часу – через мережу зв’язку. Сигнал v(t) або короткий v являє значення змінної v CPS з часом. Особливо в IoT для CPS при складанні (суб-) систем різних постачальників часто виникає явна і неявна надмірність змінних CPS. Наприклад, два сервіси виводять або надають сигнали va1 і va2 одного й того ж виду (явна надмірність). Неявна форма надмірності пов’язує інформацію різного виду друг з одним з допомогою функції. Така надмірність інформації може бути використана для порівняння пов’язаних спостережень або заміни служб спостереження, що вийшли з ладу (наприклад, датчиків або оцінок стану), що називається самовідновленням шляхом структурної адаптації (SHSA).

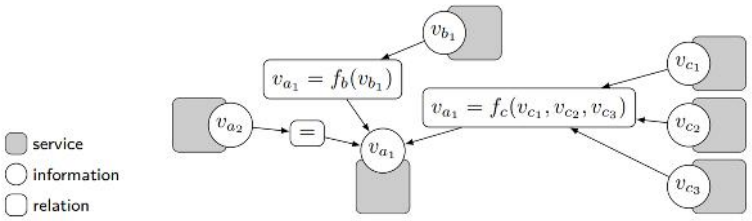

Рисунок 3.4 – Сервіси, що надають пов’язану інформацію через свої інтерфейси

На відміну від параметричної адаптації, що змінює поведінку самого сервісу, SHSA адаптує структуру системи. Проте більшість CPS спроектовані як статичні, тобто їх архітектура не передбачає засобів для додавання, видалення або перепідключення програмних сервісів.

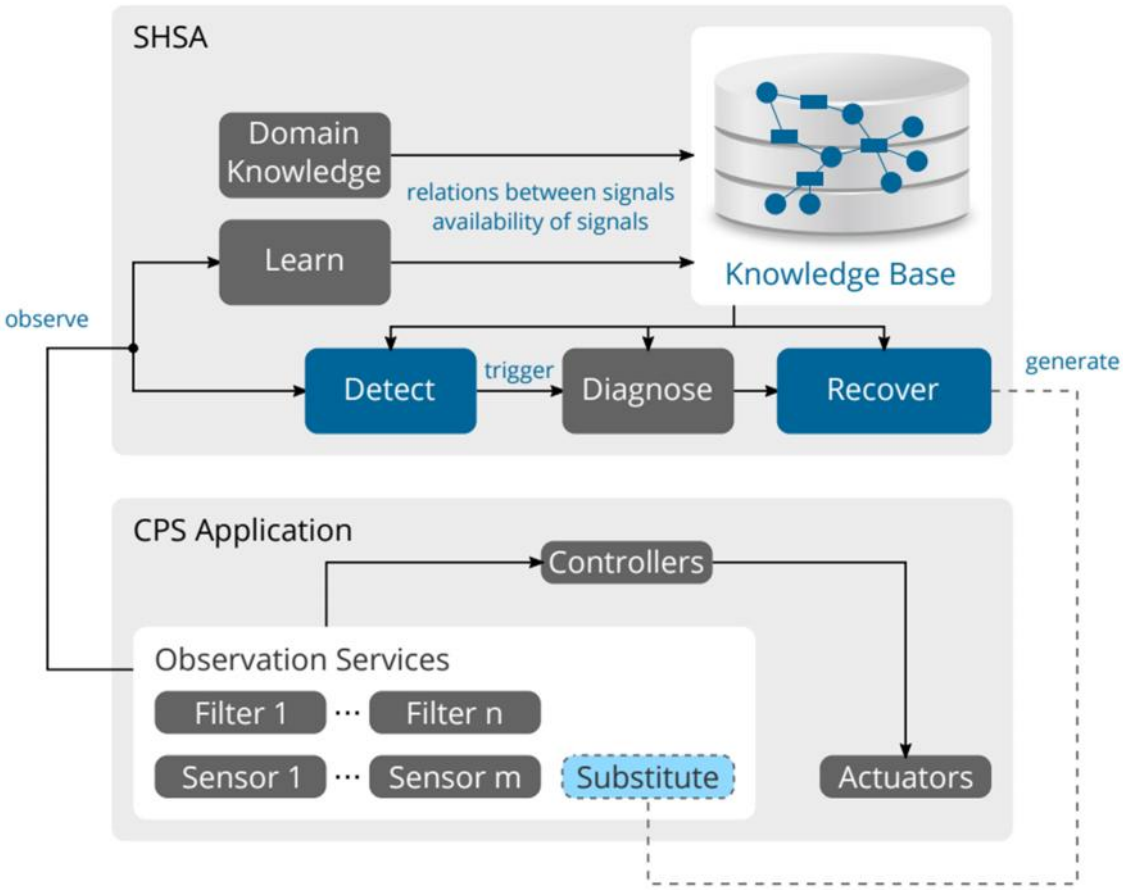

Рисунок 3.5 – Будівельні блоки або послуги (ящики) та інформаційний потік (стрілки) SHSA

Як наслідок, SHSA повинна бути включена та полегшена не лише спеціальною службою SHSA сервіс, але й базова платформа, що включає апаратне забезпечення, системне програмне забезпечення та проміжне програмне забезпечення.

4 РЕАЛІЗАЦІЯ МОДЕЛІ СКФС

Для уявлення про те, як працює SHSA, розробимо симуляційний засосунок, оснащений декількома датчиками для об'їзду перешкод (помічник екстреного гальмування), паркування або розпізнавання навколишніх об'єктів (помічник зміни смуги руху). Як і в сучасному автомобілі, деякі функції або програми (наприклад, помічник зміни смуги руху) або доступні, або ні. Три програми мобільного робота реалізуються декількома сервісами. На рисунку 4.1 показані послуги та потік інформації. Робот зупиняється перед перешкодами, коли мінімальне значення вимірювань відстані за допомогою системи Light Detection and Ranging (LIDAR) падає нижче за певний поріг, паркується за допомогою масиву гідролокаційних датчиків і розпізнає об'єкти за допомогою камери Time-of-Flight (ToF). Зокрема, LIDAR забезпечує вимірювання відстані v_{lidar}. Оцінювач перешкод визначає мінімальну відстань v_{dmin} від LIDAR. Система запобігання зіткненням вирішує, чи безпечно продовжувати рух чи робот має бути зупинено. Він видає бажану швидкість та направлення v_{safe_cmd} службовому роботу, який керує двигунами коліс.

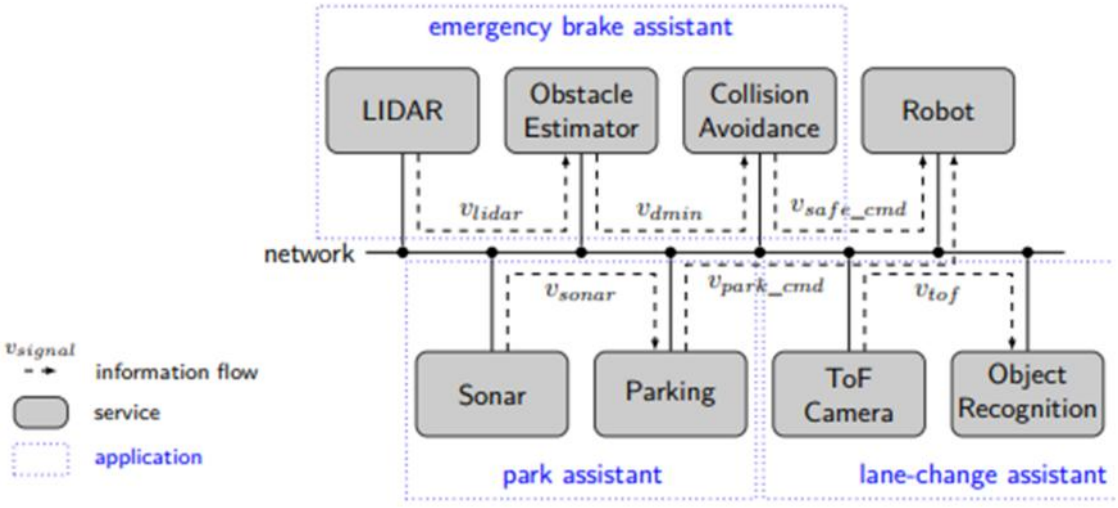

Рисунок 4.1 – Схема застосунка

Ці три програми працюють незалежно один від одного. Тим не менш, всі три використовують інформацію про відстань до перешкод, хоч і за допомогою різних датчиків (оптимізованих для конкретної програми). Така надмірність інформації змодельована в базі знань, що використовується SHSA для виявлення та відновлення після збоїв. Наприклад, служба SHSA, підключена до комунікаційної мережі, визначає коли LIDAR виходить з ладу, порівнюючи дані LIDAR з даними гідролокатора і камери. У разі відмови LIDAR SHSA генерує заміну, яка надає необхідну інформацію, отриману з виходів гідролокатора, контролерам. виходу LIDAR у порівнянні з виходом гідролокатора та камери (рисунок 4.2). У разі відмови LIDAR SHSA генерує заміну, яка надає необхідну інформацію, отриману з виходів гідролокатора, контролерам (рисунок 4.3).

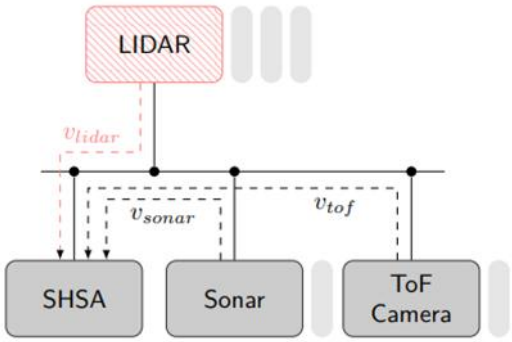

Рисунок 4.2 – Сценарій виявлення несправностей SHSA

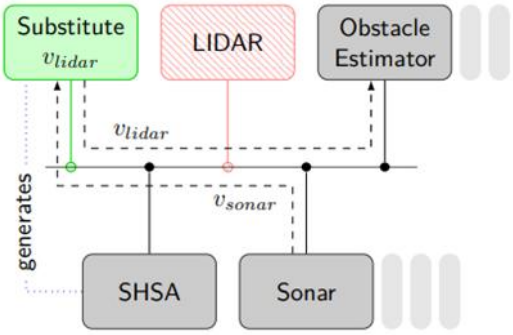

Рисунок 4.3 – Сценарій відновлення SHSA

Дані можуть передаватися безперервним потоком (синхронні), плануватися для відправлення та отримання у певні моменти часу (тригерні), передаватися за подіями або надсилатися через нерегулярні інтервали (асинхронні). У мережах зв'язку, керованих подіями або асинхронних мережах, точний час отримання повідомлення невідомий. Такі служби, як блоки об'єднання датчиків, мажоритарні виборці або замінник, створений SHSA, повинні поєднувати інформацію з різних джерел. Тому інтерфейс таких служб повинен обробляти можливі асинхронні, багатошвидкісні вимірювання та недетермінований зв'язок із найкращими зусиллями в мережі IoT або CPS. І тому послуги повинні виконувати так зване тимчасове вирівнювання входів перед виконанням обчислень.

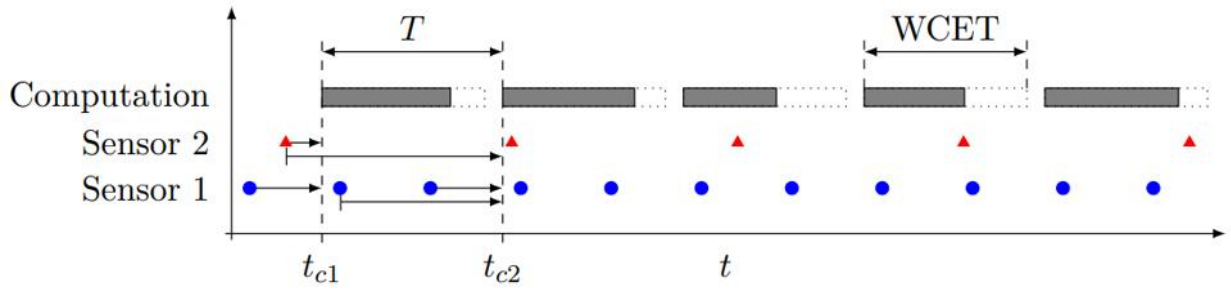

Рисунок 4.4 – Типова тимчасова шкала обчислення, що періодично виконується, і його асинхронних і багатошвидкісних входів (датчик 1 і 2)

Отримані входи вирівняні по початковій точці обчислень (показані стрілками від отриманих спостережень до перших двох обчислень, що починаються t_{c1} і t_{c2}).

4.1 База знань про резервування

База знань SHSA включає:

- відносини між змінними CPS, кожне з яких виражається у вигляді функції r : $v_o = f(V_I)$ для обчислення вихідної змінної v_o з набору вхідних

змінних V_I;

- наявність сигналів та їх відображення на змінні;
- за бажанням користувальницькі властивості змінних, відносин і сигналів для вимірювання їх ефективності.

Відправною точкою до створення бази знань є граф інформаційних потоків докладання. Наприклад, на рисунку 4.1 показані сервіси та інформаційні потоки, реалізовані на мобільному роботі, який ми використовували як прототип SHSA. Співвідношення сигналів один з одним та об'єднання сигналів, що дають ту саму змінну (наприклад, відстань до перешкод), дає базу знань, представлену на рисунку 4.5.

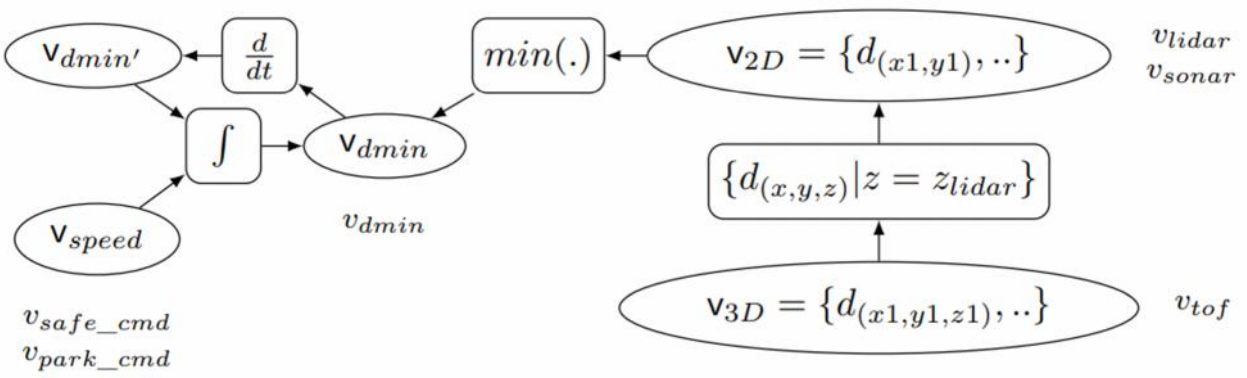

Рисунок 4.5 – База знань мобільного робота, що використовується для запобігання зіткнень, що самовідновлюється. Фігури – відносини. Еліпси – змінні

Доступні сигнали v_{signal} збираються з потоку інформації рисунку 4.1. Деякі сигнали захоплюють той самий вид інформації, наприклад, змінна CPS v_{2D} – набір вимірювань відстані до перешкод навколо робота на фіксованій висоті – яка надається v_{lidar} і v_{sonar}. Мінімальна відстань v_{dmin} може бути отримана шляхом взяття min(.) спостережень з v_{2D} (у додатку, який виконує служба «Оцінювач перешкод»).

ToF-камера надає спостереження v_{tof} як тривимірних зображень глибини v_{3D}. З цього зображення можна витягти двовимірні відстані, вибравши спостереження на висоті LIDAR. Останнє спостереження мінімальної відстані v_{dmin0} може бути збережено та інтегровано зі швидкістю робота v_{speed} (включаючи напрямок) для оцінки поточної мінімальної відстані vdmin.

База знань K = (V, R, E) – це двосторонній спрямований граф (який також може містити цикли) з незалежними наборами змінних V і відносин R CPS. V та R є вузлами графа. Ребра E визначають вхідний/вихідний інтерфейс відношення. Зокрема, vi є вхідною змінною для r, якщо □(v_i ,r) □ E. vo є вихідною змінною r, якщо □(r, v_o) □ E.

Відновлення SHSA замінює необхідний, але непрацюючий сигнал vfailed семантично еквівалентними сигналами (сигнал v необхідний, якщо v є вхідним для контролера, наприклад, vdmin у наведеному вище прикладі). Для цього спочатку невдалий сигнал зіставляється з відповідною змінною v_{sink} в основі знань. Потім SHSA перевіряє базу знань на надмірність. Якщо змінна не забезпечується безпосередньо іншим сигналом, SHSA шукає можливу заміну. Підстановка поєднує кілька функцій зв’язку змінних друг з одним. Зокрема, підстановка s v_{sink} – це зв’язний ациклічний підграф бази знань із такими властивостями:

- вихідна змінна є єдиним стоком підстановки;
- кожна змінна має нуль або одне відношення як попередника;
- усі вхідні змінні відносини мають бути включені (з цього випливає, що джерелами графа підстановки є лише змінні).

Підстановка s допустима, якщо всі вихідні змінні забезпечені сигналами, інакше підстановка неприпустима. Багато допустимих підстановок змінної v називається S(v). Тільки допустима підстановка може бути інстанційована (наприклад, заміни непрацюючого сигналу) шляхом конкатенації відносин у функцію, яка приймає вибрані сигнали як вхідні. Наприклад, сигнал ToF-камери v_{tof} може бути використаний для отримання мінімальної відстані до перешкоди v_{dmin}.

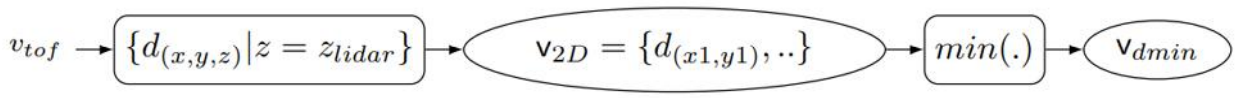

Рисунок 4.6 – Правильна заміна для v_{dmin}

Монітор періодично перевіряє правдоподібність. При кожному викликі монітор порівнює спостереження, зібрані з моменту останнього виклику монітора, з буфером (настроюваного розміру) спостережень за попередніми виконаннями. Буфер використовується для компенсації пізніх прийомів чи високих затримок зв’язку.

Один виклик монітора складається з кількох кроків:

- нові та буферизовані спостереження збираються у комбінації входів для підстановок монітора таким чином, щоб їх часові інтервали перекривалися;
- кожна комбінація входів виконується, тобто. інтервали значень входів переносяться через відповідні відносини підстановки шляхом застосування інтервальної арифметики. Тимчасовий інтервал виходу – це перетин тимчасових інтервалів входів (час, протягом якого вихід вважається дійсним);
- далі монітор попарно порівнює виходи останнього кроку. Помилка на підстановку агрегується та ранжується;
- монітор повертає підстановку з найбільшою помилкою, або жодної, якщо всі помилки дорівнюють 0. Наступний фільтр ковзаючого середнього або ковзаючої медіани може компенсувати провали вхідних сигналів.

Результат моніторингу – невдала заміна s_{failed} – використовується для запуску відновлення. Крім того, необхідно провести аналіз першопричини, щоб визначити, який саме вхідний сигнал привів до несправності на виході s_{failed}. Для простоти у наведених прикладах підстановка мала лише один вхід, щоб можна було обійтися без діагностики. Понад те, у поточної реалізації передбачається, що входи підстановок незалежні друг від друга.

4.2 Відновлення

Відновлення сигналу, що відмовив, запускається монітором, наприклад, простим сторожовим тайм-аутом, заданим тайм-аутом для сигналу або виявленням несправності, представленим в останньому розділі. Блок відновлення шукає допустиму заміну та інстанцує її. Замінник забезпечує сигнал, що раніше відмовив, шляхом передачі зв’язаних сигналів через допустиму заміну. Базу знань можна переглянути, наприклад, за допомогою глибинного пошуку (DFS) [19], або запросити для пошуку допустимих замін. Однак налаштування та запуск замінника (інстанціювання та підключення) може зайняти значно більше часу, ніж сам пошук. Тому бажано шукати найкращу заміну одразу.

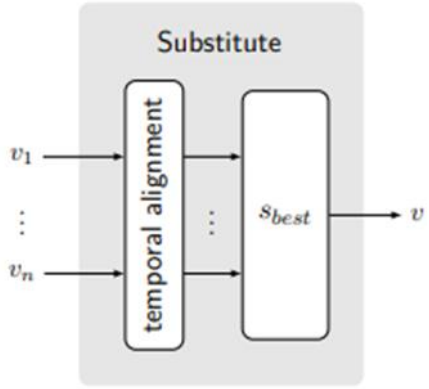

Рисунок 4.7 – Абстрактний вигляд замінника. Вхідні сигнали v_i вирівнюються та передаються через підстановку s_i

Спрямований пошук, оцінюючи кожну заміну за корисністю, зменшує час пошуку найкращого варіанту відновлення. Більш того, запропонований алгоритм – Self-Healing by Property-Guided Structural Adaptation (SH-PGSA) – може бути розділений на потоки, які можуть виконуватися паралельно. Спочатку несправний сигнал зіставляється з його змінною, скажімо v_{sink}, в основі знань. v_{sink} є коренем кожного можливого дерева підстановок. Спочатку

SH-PGSA створює робочого для підстановки v_{sink}. Робітник виявляє відносини, пов’язані з v_{sink}. Він вибирає найкраще ставлення та повертає додаткових робітників для інших можливостей. Кожен робітник є ще однією можливою заміною для v_{sink}. Робітники також зберігають корисність своєї заміни. Робітникові з найбільшою корисністю дозволяється продовжити на один крок, тобто він перевіряє такі пов’язані стосунки ненаданих змінних. Всі змінні обраного відношення повинні бути доступні або згодом замінені. Якщо корисність підстановки монотонно зменшується (за рахунок додавання відношення чи змінної як наслідок), то спочатку повертається оптимальна (з погляду функції корисності) підстановка. Робочий також може перервати роботу, якщо підстановка є недійсною і не надані змінні не можуть бути замінені далі.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є аналіз предметної області та розробка моделі самовідновлювальної CPS. Було проаналізовано стан кіберфізичних систем, стан самовідновлювальних кіберфізичниз систем, проблеми впровадження самосвідомлення у кіберфізичній системі, чутливі до ресурсів самовідновлювальні системи

Проаналізовані дані дозволяють конкретизувати проблеми та питання, що були розглянуті в кваліфікаційной роботі, а саме розробка модель самовідновлювальної кіберфізичної системи.

Науковою новизною кваліфікаційной роботи є розробка методу самовідновлювальної кіберфізичної системи.

Як результат кваліфікаційної роботи було створено модель самовідновлювальної кіберфізичної системи, програмний продукт, що реалізує цю модель. У цій роботі розробленно модель самовідновлення КФС, використовуючи надмірність інформації у комунікаційної мережі. Інформація або сигнали фізичних об’єктів – змінних CPS можуть бути закодовані на основі знань відносин між цими змінними КФС.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Куприяновский В.П., Д.Е. Намиот, С.А. Синягов. Киберфизические системы как основа цифровой экономики // International Journal of Open Information Technologies. – 2019. – № 4. – С. 31–42.

2. Кудж С.А., Цветков В.Я. Сетецентрическое управление и киберфизические процессы // Образовательные ресурсы и технологии. – 2017. – № 2. – С. 86–92.

3. Афанасьев М. Я., Федосов Ю. В., Крылова А. А., Шорохов С. А. Организация киберфизических производственных систем с использованием технологий блокчейн и смарт-контрактов // Изв. вузов. Приборостроение. – 2019. – Т 62. – № 3. – С. 226–236.

4. Цветков В.Я. КИБЕР ФИЗИЧЕСКИЕ СИСТЕМЫ // Международный журнал прикладных и фундаментальных исследований. – 2017. – № 6. – С. 64–65.

5. Куприяновский В.П. Умная инфраструктура, физические и информационные активы, Smart Cities, BIM, GIS и IoT // International Journal of Open Information Technologies. – 2020. – № 5. – С. 1–32

6. Росляков А.В. Интернет вещей: учебное пособие / А.В. Росляков, С.В. Ваняшин, А.Ю. Гребешков. – Самара: ПГУТИ, 2015. – 200 с.

7. H. Hoffmann et al. 2013. A generalized software framework for accurate and efficient management of performance goals. In Proceedings of the International Conference on Embedded Software. 1–10

8. IBM Corporation. 2006. An Architectural Blueprint for Autonomic Computing. IBM White Paper

9. Peter R. Lewis, Arjun Chandra, Funmilade Faniyi, Kyrre Glette, Tao Chen, Rami Bahsoon, Jim Torresen, and Xin Yao. 2015. Architectural aspects of self-aware and self-expressive computing systems. IEEE Comput. 48, 8 (2015), 62–70.

10. Samuel Kounev, Peter Lewis, Kirstie Bellman, Nelly Bencomo, Javier Camara, Ada Diaconescu, Lukas Esterle, Kurt Geihs, Holger Giese, Sebastian Götz, Paola Inverardi, Jeffrey Kephart, and Andrea Zisman. 2017. The notion of selfaware computing. In Self-Aware Computing Systems, Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu (Eds.). Springer, 3–16.

11. K. Neshatpour, F. Behnia, H. Homayoun, and A. Sasan. 2018. ICNN: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation. In Proceedings of the 2018 Design, Automation Test in Europe Conference Exhibition (DATE'18). 551–556.

12. Farnaz Forooghifar, Amir Aminifar, and David Atienza Alonso. 2018. Self-aware wearable systems in epileptic seizure detection. In 21st Euromicro Conference on Digital System Design (DSD'18). IEEE, 426–432.

13. H. A. Kholerdi, N. TaheriNejad, and A. Jantsch. 2018. Enhancement of classification of small data sets using selfawareness; an iris flower case-study. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS'18). 1–5.

14. E. P. Kim and N. R. Shanbhag. 2014. Energy-efficient accelerator architecture for stereo image matching using approximate computing and statistical error compensation. In Proceedings of the IEEE Global Conference Signal and Information Processing (GlobalSIP'14).

15. Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Xiaobing Feng, Yunji Chen, and Olivier Temam. 2015. ShiDianNao: Shifting vision processing closer to the sensor. In Proceedings of IEEE/ACM International Symposium on Computer Architecture.

16. Jiachao Deng, Yuntan Fang, Zidong Du, Ying Wang, and Huawei L. 2015. Retraining-based timing error mitigation for hardware neural networks. In Proceedings of the Conference on Design, Automation and Test in Europe. 593–596.

17. E. Shamsa, A. Kanduri, N. Taherinejad, A. Proebstl, S. Chakraborty, A. M. Rahmani, and P. Liljeberg. 2020. Usercentric resource management for

embedded multi-core processors. In Proceedings of 33rd IEEE Internation Conference on VLSI Design and Embedded Systems (VLSID’20). 1–6.

18. Левцун Д. К. Модель самовідновлювальної системи / Д. К. Левцун, І. В. Рубан. // InterConf. – 2021. – №89. – С. 530.

19. O. Höftberger. Knowledge-Based Dynamic Reconfiguration for Embedded Real-Time Systems. PhD thesis, TU Wien, Institute of Computer Engineering, Wien, 2015.