

МІНІСТЕРСТВО ОСВІТУ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ

МАТЕРІАЛИ ДЕВ'ЯТОЇ
МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ ТА МОЛОДИХ ВЧЕНІХ



ПРИСВЯЧЕНА 55-РІЧЧЮ
ІНСТИТУТУ КОМП'ЮТЕРНИХ СИСТЕМ

“Сучасні інформаційні технології 2019”

“Modern Information Technology 2019”



NetCracker®



23-24 травня

Одеса
«Екологія»
2019

УДК 62-526

**ІНФОРМАТИЗАЦІЯ АВТОМОБІЛЬНИХ БОРТОВИХ СИСТЕМ
КОМП'ЮТЕРНОЇ ДІАГНОСТИКИ**

А.О.Азаренков, Д.О.Кесов, М.Ю.Батура, К.В.Чебан
д.т.н., професор, зав. кафедрою інформаційних технологій Вичужанін В.В.
Одеський Національний Політехнічний Університет, УКРАЇНА

АНОТАЦІЯ. Дана стаття є описом інформатизації і алгоритмізації проекту нашої команди, за яким ми розроблюємо автоматизовану бортову систему діагностики для симуляторної моделі автомобіля.

Розробка електронного обладнання є перспективним напрямком на сьогоднішній час. Мається на увазі: системи дистанційної діагностики, системи управління двигуном, системи радіонавігації, бортові комп'ютери для надання комунікаційних послуг. Розповсюдженна практика пізнання цих систем за допомогою когнітивних імітаційних моделей. Адаптація роботи студентів з реальними автомобілями пов'язана з фінансовими проблемами і необхідністю наявності стартових знань електроніки й інженерії.

Вихід – ми вирішили спробувати створити свою бортову систему на основі машинки для українських змагань *Roborace* [1].

Отож, ціль даної роботи – створити алгоритм діагностики параметрів руху і розробки вирішення проблеми при виникненні екстремальних ситуацій.

Ми плануємо зробити симулятор машинки, щоб уникнути зайвої роботи. Для цього нам знадобиться мікроконтролер [2] esp32 і датчики відстані [3], на яких будуть імітуватися перешкоди для машинки. Програмне забезпечення буде написано у *Arduino IDE* [4].

Стандартний алгоритм роботи машинки можна описати так.

Визначаються відстані з правого, лівого, переднього та заднього датчиків. Сервопривід [5] передається кут повороту, розрахований за формулою.

$$A = \text{rotateMiddle} + Krot * (\text{left} - \text{right}) / \text{forward} \quad (1)$$

де $\text{rotateMiddle} = 85$ (середній поворот), $Krot = 12.5$ (регулюючий коефіцієнт повороту), left – відстань до лівої стінки траси, right – відстань до правої стінки траси, forward – відстань до передньої перешкоди.

Визначається швидкість для машинки. Розраховуємо зміну розположення машинки з зміною часу, за формулою.

$$sp = (\text{forward} - \text{prevDist}) / (\text{millis}() - \text{prevTime}) \quad (2)$$

де prevDist – попередня відстань до передньої перешкоди, $\text{millis}()$ – функція програмної оболонки, яка повертає кількість мілісекунд з моменту початку виконання поточної програми, prevTime – попередній результат функції $\text{millis}()$.

Після цього переназначаються:

$\text{prevDist} = \text{forward}$

$\text{prevTime} = \text{millis}()$

для послідуєчих обчислень.

Обчислюємо регулюючий показник швидкості, за формулою.

$$\text{speedSHIM} = \text{MaxSpeed} * sp / 1.7 \quad (3)$$

де $\text{MaxSpeed} = 120$ (максимальна швидкість).

В результаті обчислюється оптимальна швидкість, за формулою.

$$\text{OptSpeed} = \text{Kspeed} / (\text{speedSHIM} - \text{MaxSpeed}) + \text{MaxSpeed} \quad (4)$$

де Kspeed = 2400 (коефіцієнт швидкості).

Тепер, ми можемо приступити до розробки алгоритму системи діагностики нашого симулятора. В реальному житті автомобілі піддаються великій кількості інформаційних та фізичних навантажень. З приводу нашого симулятора, навіть робочий алгоритм може працювати не коректно через несправності у технічному обладнанні, а також через зміни параметрів траси. Ми розглянемо приклад, при якому покриття дороги виявилось занадто слизьким для легкої машинки, і тому на поворотах її заносить. На програмному коді це означає, що одне з перешкод наближається занадто швидко, і алгоритм задає критично великий кут повороту для стабілізації руху. Для виявлення заносу ми використовуємо програмну функцію constrain().

$$\text{constrain}(A, 55, 115) \quad (5)$$

Функція перевіряє і якщо треба задає нове значення, так щоб воно було в області допустимих значень, заданої параметрами. Якщо кут повороту менше 55 або більше 115, то це означає, що відмувається внештатна ситуація, треба зупинити машинку і порекомендувати програмісту коректувати програмний код. А саме – зменшити показник максимальної швидкості. Таким чином за формулою (4) розрахована швидкість буде зменшуватися, що забезпечить зниження фактору заносу на рух машинки.

Планується експериментально визначити, в яких пропорціях необхідно зменшувати максимальну швидкість, розробивши функцію залежності максимальної швидкості від показника заносу.

Також слід зазначити, що у проектуванні розглядається можливість дистанційного перекодування симулятора за допомогою спеціального обладнання, а не тільки визначання порад для користувача і локальне перекодування.

Ми плануємо розробити повноцінну систему діагностики з покриттям більшої кількості нештатних ситуацій. Це дозволить на основі простих алгоритмів і не затратного технічного обладнання здобити навички у розробці бортових систем діагностики для реальних автомобілей.

Таким чином, ми розробили алгоритм системи бортової діагностики симулятора автомобіля, виявили напрямок подальших дій проекту і виклали дані у статті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Roborace гонки автономних роботів [Електронний ресурс]. - <http://roborace.org/>
2. ESP32 [Електронний ресурс]. - <https://ru.wikipedia.org/wiki/ESP32>
3. Инфракрасный дальномер Sharp [Електронний ресурс]. - <http://amperka.ru/product/infrared-range-meter-150>
4. Arduino IDE [Електронний ресурс]. - https://en.wikipedia.org/wiki/Arduino_IDE
5. Будова та принцип роботи сервоприводу [Електронний ресурс]. - <https://radiomodel.in.ua/budova-ta-printsip-roboti-servoprivodu/>
6. Гонки Roborace [Електронний ресурс]. - <https://habr.com/ru/sandbox/72384/>