основних параметрів стану робота в систему SCADA з можливістю аварійного останову при виникненні нештатних ситуацій.

Застосування коботів для відбирання проб матеріалу з обладнання, що приведено в дію, та його інтеграція в загальну систему автоматизованого управління надасть можливість суттєво знизити небезпеку для спеціаліста під час виконання службових обов'язків.

Список використаних джерел

1. Collaborative and Humanoid Robots [Working Title]. (2021). IntechOpen. doi: https://doi.org/10.5772/intechopen.91603

2. M. Javaid, A. Haleem, R. P. Singh, Significant applications of Cobots in the field of manufacturing, Cognitive Robotics, Volume 2, 2022, Pages 222-233, https://doi.org/10.1016/j.cogr.2022.10.001

3. UR-30. URL: https://www.universal-robots.com/products/ur30-robot/

Іванов А. О., *здобувач першого (бакалаврського) рівня вищої освіти*
Житомирський державний університет імені Івана Франка, Житомир

СИМУЛЯЦІЯ СЛІДУВАННЯ РОЮ ДРОНІВ ЗА ВАТАЖКОМ ЗАСОБАМИ МОВИ ПРОГРАМУВАННЯ JAVASCRIPT

Мурмурація (flocking) – складний процес скоординованого руху великої зграї птахів. Комп'ютерне моделювання дозволяє аналізувати цю поведінку та шукати способи використовувати це явище. В сучасному світі робототехніки з'являється необхідність імітувати природні стратегії груп для покращення ефективності автономних систем. В тому числі вивчається задача мурмурації, яка може використовуватись в низці напрямів робототехніки і напрямів, де тим чи іншим чином доводиться керувати групами автоматизованих юнітів.

Використання багатьох менших окремих роботів, які злагоджено працюють, має певні переваги над використанням одного керованого різнофункціонального робота. Ці переваги полягають в наступному: при використанні багатьох дрібних юнітів, вихід з ладу одного з них не є таким критичним по працездатності системи і по ресурсам на ремонт, ніж вихід з ладу головного робота, який паралізує всю систему; використання багатьох юнітів дозволяє забезпечити розпаралелювання виконання роботи між ними, що покращує ефективність системи.

Таким чином, впровадження зграйної поведінки гарантує стійкість системи до пошкоджень, і більшу її ефективність.

Один з напрямів впровадження рою роботів, які розглядаються – автономне управління дронами, при якому деяка кількість автоматизованих одиниць слідують за лідером, керування яким здійснюється вручну або автономно. Також такий варіант часто зустрічається в комп'ютерних іграх у вигляді, наприклад, загону NPC. Але на відміну від комп'ютерних ігор, в реальному світі юніти переживають дефіцит отримуваної ззовні інформації, а також обмеженість у можливості обмінюватись інформацією один з одним. Тому важливим аспектом моделювання ситуацій слідування рою дронів за ватажком є орієнтація і прийняття рішень юнітами, маючи мінімум інформації про інших юнітів та лідера.

Наша симуляція відбуватиметься у 2-вимірному просторі – вважається, що дрони літають на одній висоті. Також всі юніти загону є анонімними, що значить, що їх не можна відрізнити зовні. В нашій симуляції на юнітів були накладені наступні обмеження: єдина зовнішня інформація, якою вони володіють – це знання їх взаємного розміщення з іншими юнітами та лідером, а також знання, хто є лідером, знання його орієнтації на площині а також формацію, яку вони мають дотримуватись. Також важливим обмеженням є відсутність у юнітів довготривалої пам'яті, що унеможливлює їм здатність запам'ятовувати інформацію окрім характеристик юнітів і знання їхніх ролей в загоні.

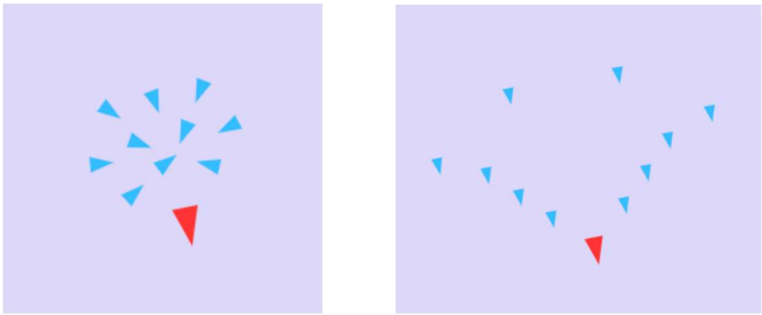

Рисунок 1. Приклади формацій юнітів: а) невпорядковане, б) впорядковане

Реалізовано декілька варіантів слідування (див. рис. 1): невпорядковане, яке дозволяє імітувати зграї птахів, які тримаються разом, створюючи «хмару»; і впорядковане, за яким всі юніти тримають

чіткий стрій, цей варіант дозволяє імітувати клини птахів (на рисунку лідер позначений червоним, а інші юніти – синім). З урахуванням дефіциту інформації, перший варіант слідування реалізований з алгоритмічною складністю O(1) на ітерацію, або O(n) з урахуванням уникнення зіткнення між юнітами. Другий алгоритм слідування працює за $O(n^2)$, з урахуванням колізії між юнітами $O(n^2 + n) = O(n^2)$. Важливо зазначити, що ватажок рою абсолютно вільний в своїх діях – юніти в будь-якому разі автономно уникатимуть зіткнення з ним.

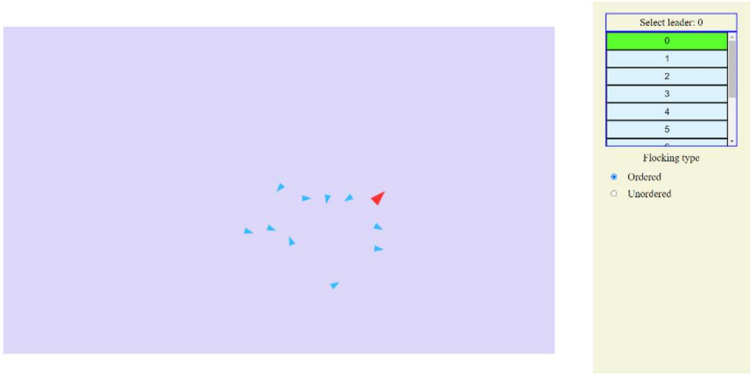

Рисунок 2. Зовнішній вигляд веб-додатку.

Програма розроблена на мові програмування JavaScript дає можливість (див. рис. 2) обирати тип слідування, і обирати ватажка, причому весь загін переформовуватиметься згідно встановлених змін. Цю програму можна використовувати для вивчення і дослідження алгоритмів слідування, доповнюючи її функціонал.

Список використаних джерел

1. *Vincenzo G., Giuseppe P. Coordination without communication: the case of the flocking problem. Italy, 2003*
2. *Housheng Su. A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements. "International Journal of Control" 82(7), 2009, pp. 1334-1343*
3. *Craig W. Reynolds. Steering Behaviors For Autonomous Characters. California, USA, 2002*
4. *Mat B. Programming Game AI by Example. Texas, USA, ISBN 1-55622-078-2, 2005, pp. 133-189*
5. *Fatih G., Erol Ş. The pros and cons of flocking in the long-range "migration" of mobile robot swarms. "Theoretical Computer Science", Ankara, Turkey, 2010, pp. 2140-2154*
6. *Understanding Steering Behaviors. URL:*
7. *https://code.tutsplus.com/series/understanding-steering-behaviors--gamedev-12732*