

---

“АСАУ” – 17(37) 2010

---

УДК 004.896

В.Г. Писаренко, Ю.В. Писаренко, К.Ю. Мелкумян, О.С. Коваль

## **ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДБОРУ ПАРАМЕТРІВ МАНІПУЛЯЦІЙНОЇ СИСТЕМИ РОБОТА ДЛЯ РОБОТИ В СТИСЛИХ ПРОСТОРОВИХ УМОВАХ ШАХТИ**

*Анотація:* Розглянуто задачу адаптації конструкції маніпулятора робота для проникнення у вузький прохід завалів на основі відео розвідки заданого тунелю шахти. Створено алгоритмічне та програмне забезпечення для вибору на рівні імітаційного моделювання числа ланок маніпулятора та кутів між ланками для системи управління маніпуляційною системою інтелектуального мобільного робота.

*Ключові слова:* маніпулятор робота, адаптація конструкції, імітаційне моделювання.

Розробки даної роботи призначені забезпеченню підбору оптимальних параметрів на рівні комп'ютерного імітаційного моделювання для роботи робото-технічних пристроїв в стислих просторових умовах шахти (типу виконання розвідувальних функцій в шахтному забої чи ремонтних робіт у вузькому приміщенні інженерної споруди).

В цих умовах доцільно застосовувати інтелектуальний мобільний робот (ІМР), якому поставлено чотири взаємопов'язані задачі [1]:

1) зорієнтуватись для обстеження нової конфігурації стислого приміщення, що утворилася після аварії з обвалом ґрунту, і проникнути до нього в середину;

2) обстежити і розпізнати нову 3D- конфігурацію приміщення;

3) оперативно виконати задачу у зв'язку з програмним завданням;

4) розрахувати оптимальну послідовність кінематичних операцій багатоланцюгового маніпулятора ІМР для цілей подальшого виконання головної задачі;

5) вирішивши проблеми - 1, - 2, - 3, - 4, реалізувати центральну частину заданої програми: знайти і захопити маніпулятором головний елемент всієї “сцени”. Прикладом заданої програми може бути пошук засипаних обвалом ґрунту гірників, або вимкнення/ввімкнення певних електроприладів, чи відновлення зруйнованих аварією комунікації для нормалізації ситуації, тощо.

### **Постановка задачі**

Реалізація програмної системи для імітаційного моделювання передбачає вирішення таких задач:

- моделювання зовнішнього середовища;
- побудова траєкторії захвата маніпулятора по центрам мас зрізів, якщо конфігурація тунелю наперед відома. Під час переміщення ланцюгів маніпулятора та його захвату потрібно дотримуватись цієї

---

© В.Г. Писаренко, Ю.В. Писаренко, К.Ю. Мелкумян, О.С. Коваль, 2010

траєкторії, яка проходить в центрі тунелю. Для цього потрібно обрати кращій метод побудови якомога точнішої траєкторії за допомогою кубічного сплайну;

- реалізація руху захвату маніпулятора за побудованим сплайном;
- реалізація руху захвату маніпулятора за методом перевірки зіткнень “сфери безпеки” з поверхнями тунелю на кожному кроці. Поворот сегменту маніпулятора здійснюється якщо зіткнення відбулося. Цей метод використовується коли конфігурація тунелю наперед невідома.

### **Алгоритм створення моделі тунелю з набору зрізів**

Для того щоб створити модель тунелю з набору зрізів з різною кількістю точок у зрізах потрібно провести триангуляцію і отримати з набору точок набір трикутників (полігонів) без додавання нових точок, при цьому модель не повинна бути суцільною.

Для проведення триангуляції авторами запропоновано наступний алгоритм. Потрібно пройти по всім зрізам тунелю і, беручи по два зрізи (верхній і нижній), знаходити полігони між ними. Для цього потрібно спроекувати обидва зрізи на площину, сумістити їх центри і з центру провести промінь у будь-якому напрямку таким чином, щоб він перетнув по одному ребру у обох зрізах. Далі слід визначити параметри обходу точок зрізів, обхід повинен проходити у обох зрізах в одному напрямку, починаючи від знайдених ребер. Потім необхідно занести послідовно всі точки верхнього зрізу у список згідно вибраного напрямку, те ж саме провести з другим зрізом. Далі провести розрахунок відстані між першими і останніми елементами отриманих списків, знайти найбільшу з них, визначити у якому списку більше точок, і з того списку, у якому точок більше, видалити крайню точку (відстань з якої є найбільшою), побудувавши спочатку з її участю полігон. Якщо у обох списках рівна кількість точок, взяти перші елементи списків і підрахувати відстані від першого елемента одного списку до другого елемента другого списку. Нульовий елемент, відстань з якого є найбільшою, видаляється зі списку після побудови полігону з його участю. Так продовжувати поки у списках не залишиться по одній точці. Аналогічним чином потрібно опрацювати координатами ребер, які було знайдено спочатку. Для побудови полігону слід взяти 2 точки, між якими розраховано відстань, третьою точкою буде точка, що йде наступною у списку, з якого ми видаляли точку.

Алгоритм побудови тунелю за зрізами наведено на рисунку 1.

### **Алгоритм пошуку оптимального оптимальної довжини сегменту маніпулятора**

Для розрахунку оптимальної довжини сегменту маніпулятора потрібно в якості вхідних даних тунель, заданий зрізами, координати полігонів, з яких він складається. Маючи координати полігонів і параметри маніпулятора, а також побудувавши сплайн по центрам зрізів, якими

заданий тунель, можна розрахувати положення сегменту маніпулятора у деякий момент часу (маніпулятор рухається за сплайном) і визначити перетинається сегмент з площинами тунелю чи ні. Потрібно взяти спочатку велику довжину сегменту і поступово, знаходячи перетини сегменту з полігонами тунелю, зменшити її до оптимальної, тобто до максимальної довжини, при якій сегмент маніпулятора зможе пройти тунелем без зіткнень з його полігонами. За результатами перевірки можливості зіткнення сегменту “можливої” довжини маніпулятора з полігонами тунелю потрібно перевизначити мінімальну і максимальну довжинами сегменту.

Алгоритм визначення довжини ланок маніпулятора наведено на рис. 2.

### **Алгоритм руху маніпулятора в тунелі**

Для того щоб маніпулятор мав змогу пройти по тунелю потрібно дати на його шарніри між ланками кути повороту у полярній системі координат. Ці кути є різними для кожного шарніру і змінюються по мірі просування маніпулятора углиб тунелю. Тобто створюється масив кутів (2 кути для кожного шарніру маніпулятора), і у цей масив у кожний момент часу заносяться кути повороту сегментів. При цьому якщо рух є поступальним достатньо розраховувати тільки кути для шарніру сегменту, який проходить по тунелю першим, інші сегменти після досягнення позиції в якій знаходився перший сегмент будуть повторювати його маневр і їм можна дати кути розраховані для першого сегменту. Якщо рух маніпулятора не є поступальним ці кути потрібно розраховувати послідовно для всіх сегментів.

Для розрахунку кутів потрібно мати координати позицій, в якій знаходяться кінці сегменту маніпулятора. Координати одного кінця сегменту маніпулятора завжди відомі, для розрахування положення іншого кінця можна скористатися одним з двох методів. Перший передбачає, що маніпулятор рухається по сплайну, побудованому по центрам зрізів, якими заданий тунель, тобто кінці сегменту знаходяться на сплайні, і, якщо відомі координати першого кінця, довжина сегменту і кути повороту сегменту у попередній позиції, координати другого кінця можна легко розрахувати за допомогою кубічного сплайну. Інший метод передбачає, що маніпулятор рухається по тунелю у одному напрямку поки може, після чого здійснює поворот, у цьому випадку визначається зіткнення, і якщо його нема, координати другого кінця сегменту розраховуються легко, вони лежать на вже відомій прямій, якщо ж є зіткнення розраховується зсув за допомогою якого можна уникнути зіткнення, зсув дає нову пряму з якої можна розрахувати положення другого кінця сегменту.

Є стартове положення у якому маніпулятор прикріплений до носія, є початкові кути, масиви кутів та поточних положень сегментів. Послідовно перебираємо усі сегменти маніпулятора. Розраховуємо нове положення сегменту маніпулятора і кути повороту сегменту за допомогою сплайну за алгоритмом, блок-схему якого наведено на рисунку 3.

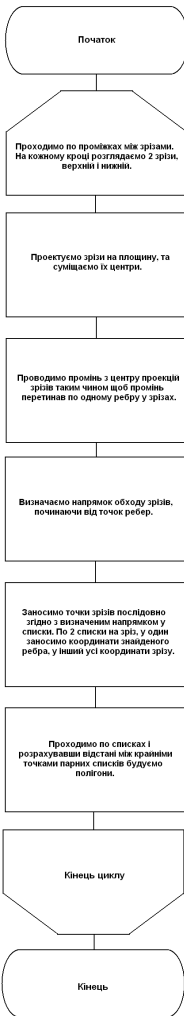


Рис. 1 – Блок-схема алгоритму побудови тунелю за зрізами з різною кількістю точок.

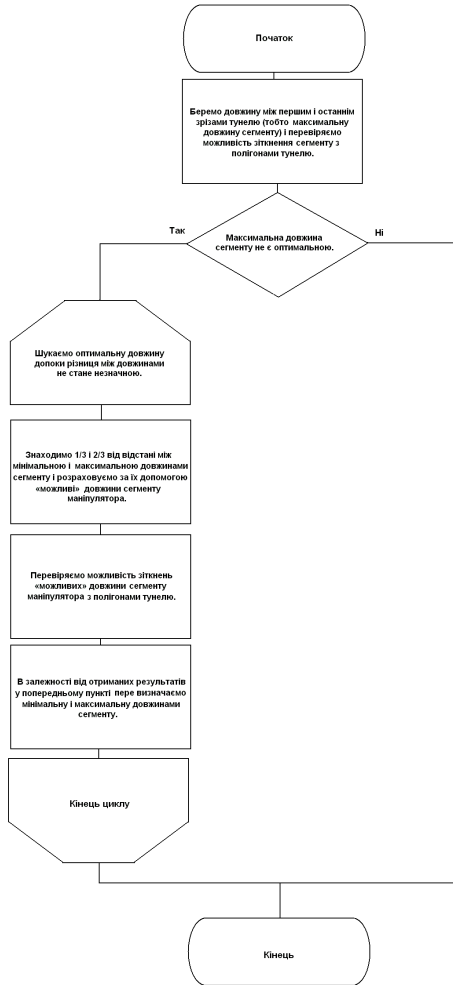


Рис. 2 – Блок-схема алгоритму визначення довжини ланок манипулятора.

## **Опис програмного модуля відлагодження параметрів керування маніпулятором робота**

Для програмного модуля задаються наступні вхідні дані:

1. Характеристики тунелю:

- план зрізів;
- координати точок кожного зрізу;

2. Маніпулятор:

- ланцюги як складові маніпулятора, які є в наявності у технопарку.

Вихідні дані:

- довжина ланок маніпулятора для тунелю з наявних в технопарку;
- значення кутів повороту сегменту маніпулятора на кожному кроці руху маніпулятора вглиб тунелю.

Функціональність програми показано діаграмою прецедентів на рисунку 4.

Після задання тунелю (Меню “Задать тунель”) головна форма модуля розрахунку траєкторії маніпулятора відображає 3D сцену шахти і маніпулятора. Сцена містить траєкторії захвата маніпулятора, побудовані за допомогою кубічних сплайнів 3 методами [2-6]. Маніпулятор здійснює поступовий рух. На кожному кроці ланцюги маніпулятора не повинні зіткнутися зі стінами тунелю. Захват маніпулятора рухається за розрахованою траєкторією – сплайном (а саме третій спосіб - найкращій).

Програма має наступну функціональність:

- завдання параметрів тунелю (рис. 5);
- візуалізація 3D сцени (маніпулятор, тунель, шлях маніпулятора) (рис. 6.);
- зміна положення та швидкості руху камери;
- переключення між режимами відображення сцени (точки, точки і лінії, зображення з текстурами та шейдерами, показ або ховання частин зображення);
- тестування сплайна, за яким будується шлях крізь тунель;
- відображення кутів всіх між всіма ланками маніпулятора, значення кутів змінюється щосекунди по мірі руху маніпулятора вглиб тунелю (рис. 7.);
- можливість змінювати метод, за яким ведеться розрахунок шляху маніпулятора. Передбачено три методи побудови кубічного сплайну для руху по функції, яка побудована по центрам зрізів тунелю, коли параметри тунелю відомі з урахуванням отриманих відеоданих попередньої розвідки завалу (рис. 8). Для випадку, коли тунель невідомий реалізовано метод “тику”, де маніпулятор рухається вперед поки не буде знайдено перетин поверхні сфери безпеки з однією з площин тунелю. Якщо такий перетин є, відбувається поворот, а площа, з якою відбувся перетин виділяється синім кольором (рис. 9.);- можливість зупинити або поновити анімацію рух маніпулятора по тунелю та встановити його в початкове положення;

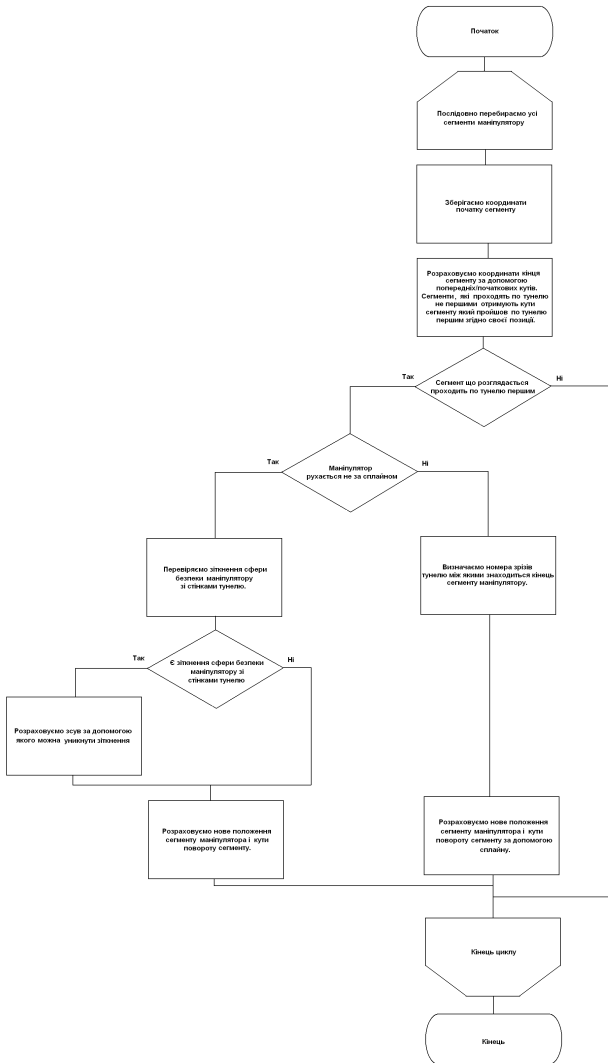


Рис. 3 – Блок-схема розрахунку нового положення сегменту маніпулятора і кутів повороту сегменту за допомогою кубічного сплайну.

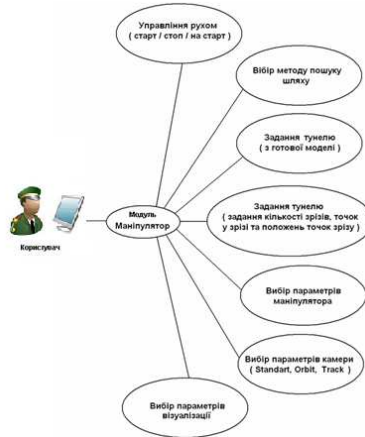


Рис. 4 – Діаграма прецедентів.

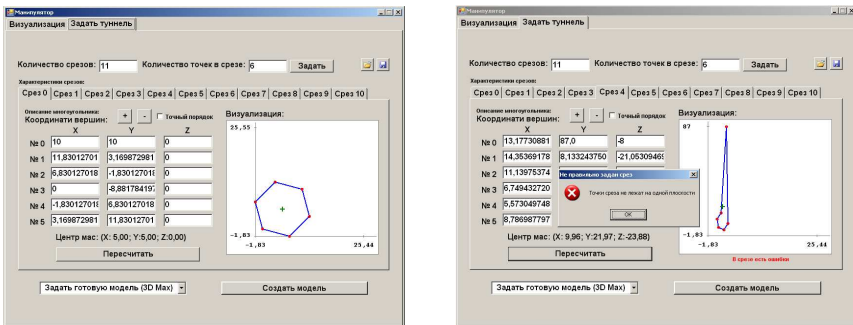


Рис. 5 – Форма задання параметрів тунелю.

- змінювати моделі готових тунелів;
- розрахунок кількості сегментів, довжини окремого сегмента, зміна його ширини, радіусу сфери безпеки та швидкості руху маніпулятора (рис. 10).

### Вибір засобів і середовища програмування

Програмне забезпечення було розроблене за допомогою мови програмування Microsoft Visual C# 2008. Вибір був обумовлений широтою можливостей даної мови і відносною простотою використання, крім того Microsoft XNA 3.0 (набір інструментів з керуванням середовищем часу виконання), створений Microsoft, що полегшує розробку й керування ком-

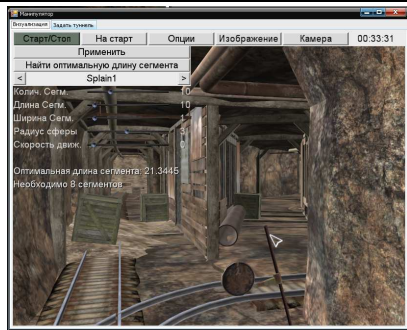
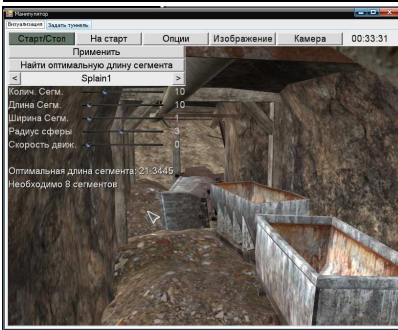


Рис. 6 – Візуалізація 3D сцени (маніпулятор, тунель).

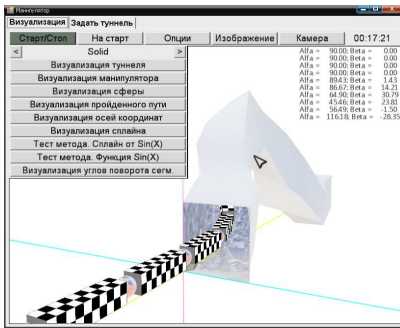


Рис. 7 – Відображення кутів між ланками маніпулятора

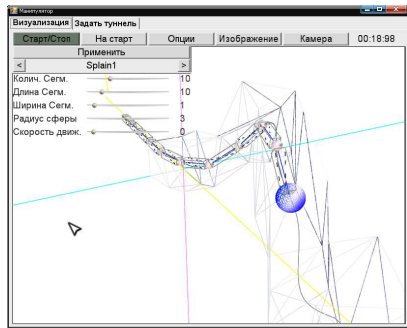


Рис. 8 – Відображення траєкторії маніпулятора, побудованої за обраним методом розрахунку кубічного сплайну

п'ютерними іграми) розроблене саме для цієї мови програмування, є їх повна сумісність [7-15].

## Висновки

В роботі було розглянуто проблему моделювання фізичних процесів і відображення результатів моделювання у реальному часі за допомогою сучасної графіки.

Створено модель маніпулятора з  $N$  сегментів, розроблено алгоритм його руху по деякому тунелю, заданому зрізами з різною кількістю точок на кожному зрізі, для проходження у стиснутих умовах до поставленої цілі. Маніпулятор рухається по тунелю за побудованим кубічним сплайном, коли параметри тунелю відомі. Якщо параметри тунелю наперед невідомі тоді використовується метод “тику”, а саме розраховуються зіткнення



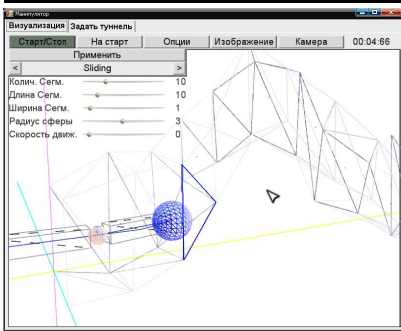


Рис. 9 – Відображення площини, з якою відбувся перетин кінцевої ланки.

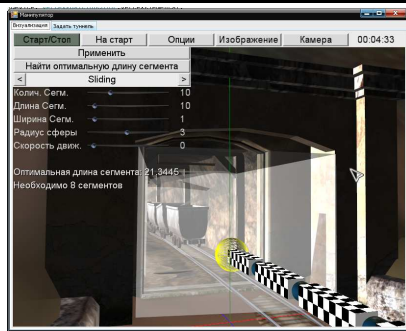


Рис. 10 – Розрахунок кількості сегментів, довжини окремого сегмента, зміна його ширини радіусу сфери безпеки та швидкість руху маніпулятора.

кінцевої ланки маніпулятора з стінками тунелю і, в разі зіткнення виконуються поворот на наступному кроці.

За допомогою отриманих результатів можна розрахувати необхідні параметри маніпулятора перед його відтворенням у фізичному просторі.

Розглянуті задачі є частиною розробки Інституту кібернетики імені В.М. Глушкова Національної академії наук України за темою “ВК.265.07.09. Розробка системи інтелектуальних роботів для дистанційного моніторингу з використанням спеціальної технології обміну оперативними даними для боротьби з пожежами на шахтах і з проявами тероризму на морі та на суходолі” [1, 11]. Користувачами розробленого програмного забезпечення, як очікується, будуть інженери-дослідники та інженерний персонал апаратно-програмного комплексу.

### Литература

1. Писаренко В.Г., Писаренко Ю.В. Актуальные направления развития интеллектуализированной робототехники для снижения аварийности на шахтах // Искусственный интеллект. – 3. – 2009. – С. 308-316.
2. Писаренко В.Г. Об интерполяции со сглаживанием выпуклыми кубическими сплайнами при оптимизации конечных условий // Сборник научных трудов “Физика и механика нелинейных явлений”. - Киев “Наукова думка”. – 1979. – С 115-131.
3. Лук’яненко С. О. Основи обчислювальних методів розв’язування диференціальних рівнянь. Видання третє, виправлене і доповнене. НТУУ “КПІ”, Київ 2004. – 153с.
4. Лук’яненко С. О. Математичне забезпечення САПР. Видання третє, виправлене і доповнене. НТУУ “КПІ”, Київ 2004. – 143с.

5. Мелкумян Е.Ю. Восстановление гладкой пространственной траектории ответственных элементов робототехнических комплексов путем сплайн-интерполяции // Адаптивные системы автоматического управления. -2009.- . 15 (35).
6. Мелкумян К.Ю. Вибір удосконаленого методу планування траєкторії руху маніпулятора // Адаптивні системи автоматического управління. -2010.- . 16 (36).
7. Карлі Уотсон, Крістіан Нейгейл. Microsoft Visual С# 2008 Базовий курс, “Діалектика”, Москва \* Санкт-Петербург \* Київ, 2009. – 1211 с.
8. Крістіан Нейгейл, Білл Івєн. С# 2005 і платформа .NET 3.0 Для професіоналів, “Діалектика”, Москва \* Санкт-Петербург \* Київ, 2008. – 1790 с.
9. Трей Неш. С# 2008 Пришвидшений курс. Для професіоналів, “Вільямс”, Москва \* Санкт-Петербург \* Київ, 2008. – 576 с.
10. Бенжамин Нитшке. XNA Game Programing for Xbox 360 and Windows Professional, 2008. – 300с.
11. Сайт <http://www.xnadev.ru/news.php>
12. Співтовариство сайту <http://www.gamedev.ru/> яке знаходиться за адресою: <http://www.gamedev.ru/community/xna/>
13. Авторське свідоцтво на наукову статтю Писаренко В.Г., Писаренко Ю.В. “Базовые компоненты интеллектуальной информационной технологии “АНТИПИРАТ” (подано у Державний департамент інтелектуальної власності України 14.12.2009 р.).

Отримано 03.12.2010 р.