

Робот для испытания «Кегельринг»

О.С. Марченко, Д.А. Фаррахова, Н.В. Киреев, студенты 4 курса каф. КИБЭВС
Научный руководитель О.В. Пехов инженер кафедры КИБЭВС
г. Томск, ТУСУР, ФВС, marchenca@mail.ru

Робототехнические комплексы популярны в области образования как современные высокотехнологичные исследовательские инструменты в области теории автоматического управления и мехатроники. Во многих странах широко распространено обучение в школах и колледжах с использованием робототехнических комплексов таких, как набор LEGO Mindstorms. С развитием этого опыта стали популярны соревнования роботов. Одним из испытаний таких соревнований является «Кегельринг».

Суть испытания заключается в том, чтобы за кратчайшее время вытолкнуть корпусом робота все 8 кегель за пределы ринга.

Конструкция

После анализа испытания и состава набора LEGO Mindstorms, для конструирования робота были выбраны следующие детали: управляющий блок NXT 2.0 (рисунок 1а), моторы (рисунок 1б), сенсор расстояния или ультразвуковой сенсор (рисунок 1в), сенсор освещенности (рисунок 1г), колеса (рисунок 1д), балки и шестеренки (рисунок 1е).



а)



б)



в)



г)



д)



е)

Рисунок 1 – Элементы набора, использованные для конструирования робота:

- а) управляющий блок NXT 2.0; б) моторы; в) сенсор расстояния;
г) сенсор освещенности; д) колеса; е) балки и шестеренки

Из выбранных деталей была собран робот, конструкция которого изображена на рисунке 2.

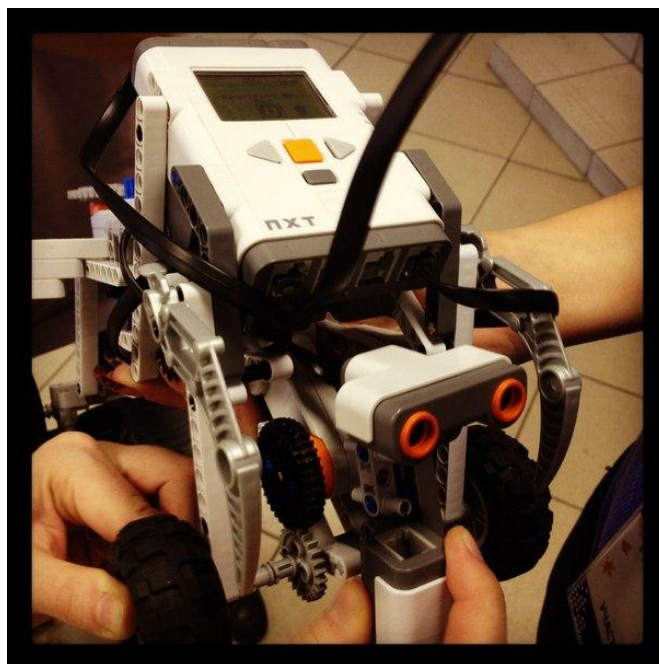


Рисунок 2 – Конструкция робота

Один из двигателей отвечает за движение робота, второй двигатель – за угол поворота.

Алгоритм

Возможные алгоритмы движения представлены на рисунке 3.

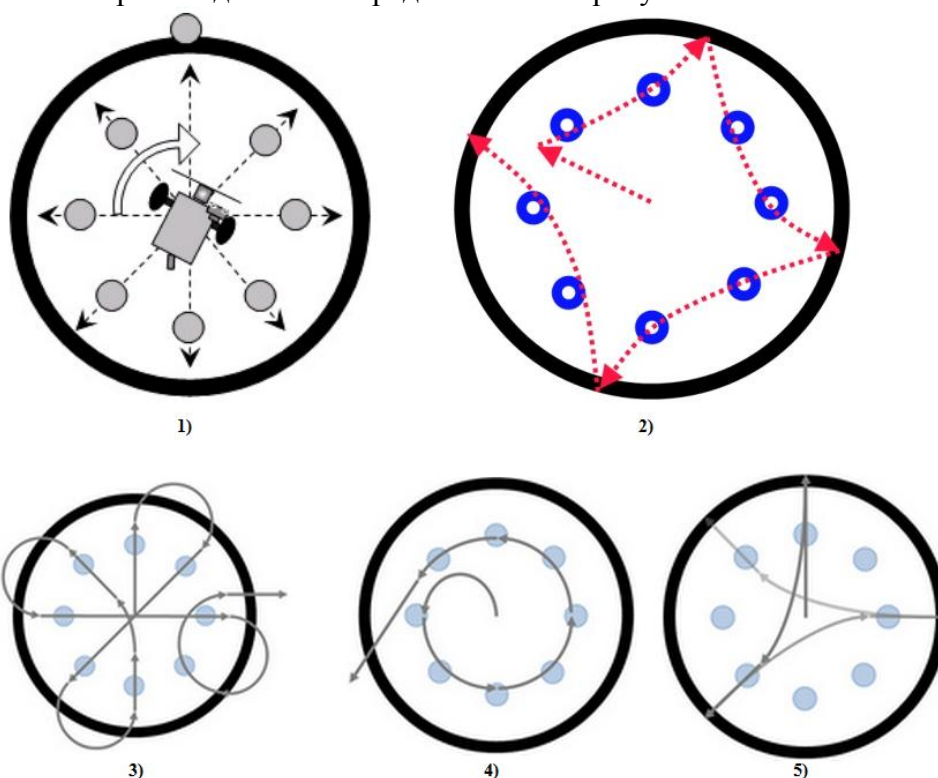


Рисунок 3 – Алгоритмы движения робота

Алгоритм 1 заключается в поиске и выталкивание кеглей с возвращением в центр ринга.

Алгоритм 2: робот выталкивает по две кегли за один раз, не возвращаясь в центр круга.

Алгоритм 3: вытолкнув кеглю за пределы ринга, робот разворачивается и сбивает 2 кегли, затем снова разворачивается и снова сбивает 2 кегли и так до тех пор, пока кегель в ринге не останется.

Алгоритм 4: робот движется по радиусу, сбивая все кегли.

Алгоритм 5: двигаясь вперед, робот выталкивает кеглю, затем двигается по дуге назад, выталкивая другую.

Эти алгоритмы движения робота были проанализированы по двум основным характеристикам:

- количество кеглей, которые можно выбить;
- время, требуемое на выполнение задания.

В результате анализа был выбран 4 алгоритм, потому что двигаясь согласно этому алгоритму затрачивается минимальное количество времени на выполнение задания. Алгоритм 4 адаптированный под конструкцию нашего робота представлен на рисунке 4.

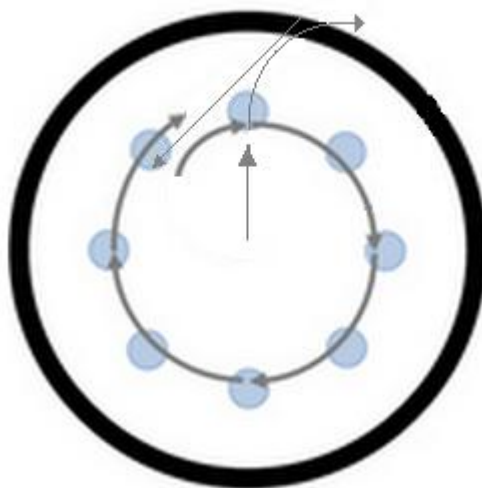


Рисунок 4 – Адаптированный алгоритм 4

Программирование робота в среде Bricx Command Center

NXC это язык программирования, созданный Джоном Хансеном, который был изначально разработан для Lego-роботов.

Bricx Command Center (BricxCC) поможет вам писать программы, загружать их в робота, запускать и останавливать их, просматривать флэш-память NXT, преобразовывать звуковые файлы для использования на роботе и многое другое. BricxCC работает по большей части как текстовый редактор, но с некоторыми дополнительными возможностями.

Листинг программы для работы робота по алгоритму 4 приведен в приложение А.

Тестирование

Во время написания программы проводились тестирование, по результатам которых осуществлялось редактирование листинга программы.

Также в качестве теста робот участвовал в «Осеннем РобоМарафоне».

По результатам тестов были выявлены достоинства и недостатки нашей модели.

Достоинства:

- 1) Оригинальность конструкции робота;
- 2) Правильная работа алгоритма.

Недостатки:

- 1) Большие габариты;
- 2) В зависимости от заряда батареи менялась мощность, подаваемая на двигатели;
- 3) Минимальное изменение в конструкции влияло на «поведение» робота.

Проанализировав полученные данные, приняли решение по модернизации робота.

Реконструкция робота

При реконструкции робота был устранен важный недостаток. Конструкция была упрощена и были уменьшены размеры робота. Также была добавлена вспомогательная часть к корпусу робота – щит. На рисунке 5 представлена реконструированная модель робота.

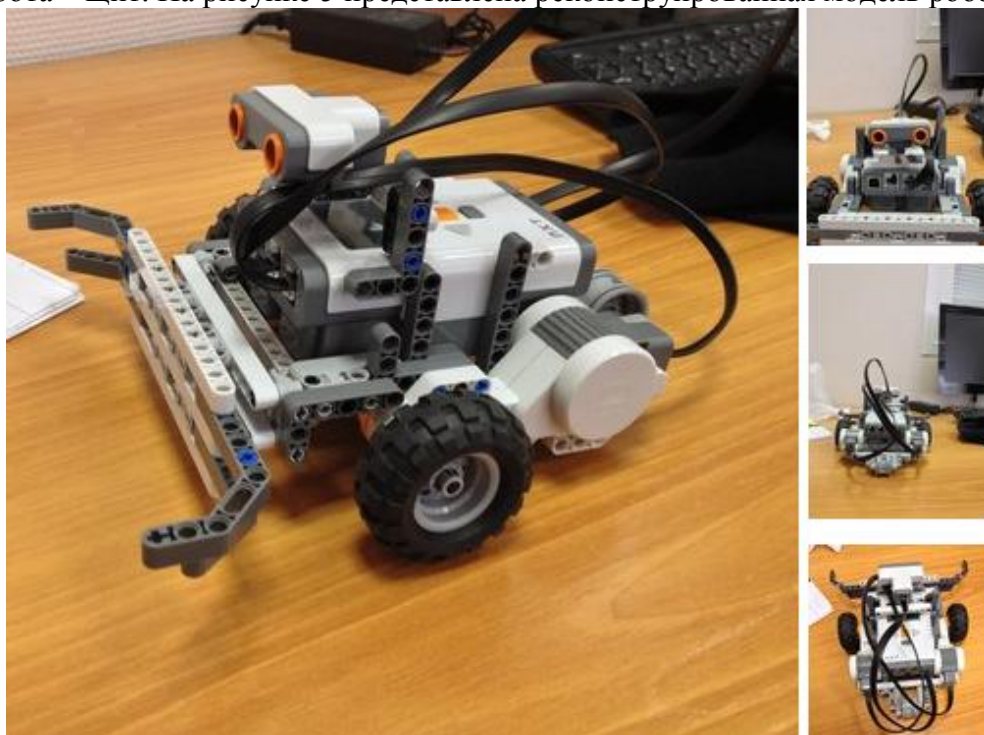


Рисунок 5 – Реконструированная модель робота

Корректировка программы

Так как конструкция была изменена, то необходимо было скорректировать программу управления роботом. Листинг скорректированной программы приведен в приложении Б. Сейчас робот находится на этапе повторного тестирования.

Приложение А

Листинг программы для работы робота по алгоритму 4.

```
task main ()
{
  SetSensorLowspeed(S4);
  SetSensorLight(S1);

  while(SensorUS(S4)> 10)
  {
    OnFwd(OUT_A, 50);
  }
  OnFwd(OUT_B, 55);
  OnFwd(OUT_A, 100);
  Wait(350);
  Off(OUT_B);
  Wait(500);
  Off(OUT_A);
  OnRev(OUT_B,55);
  Wait(450);
  Off(OUT_B);
  OnRev(OUT_A,75);
  Wait(2000);
  Off(OUT_A);
  OnFwd(OUT_B,50);
  Wait(550);
  Off(OUT_B);
  OnFwd(OUT_A, 100);
  Wait(10000);
while(Sensor(S1) > 40)
{
  OnFwd(OUT_A, 100);
  Wait(10000);
  Off(OUT_A);
}
  OnFwd(OUT_B,50);
  Wait(200);
  Off(OUT_B);
  OnFwd(OUT_A, 100);
  Wait(1000);
  OnRev(OUT_B,50);
  Wait(200);
  Off(OUT_B);
  PlayTone(1000, 300);
  Wait(300);
}
```

Приложение Б

Листинг скорректированной программы для измененной конструкции робота.

```
task main ()
{
    SetSensorLowspeed(S4);
    SetSensorLight(S1);

    while(SensorUS(S4)> 10)
    {
        OnFwd(OUT_AB, 50);
    }
    Off(OUT_AB);
    OnFwd(OUT_A,50);
    OnRev(OUT_B,50);
    Wait(600);
    OnFwd(OUT_A,100);
    OnFwd(OUT_B, 35);
    Wait(10000);
    while(Sensor(S1) > 40)
    {
        OnFwd(OUT_A, 100);
        Wait(10000);
        Off(OUT_A);
    }
    OnRev(OUT_AB,50);
    Wait(5000);
    Off(OUT_AB);
}
```