УДК 681.3.069 https://doi.org/10.26160/2618-7493-2020-3-18-20

ВНЕДРЕНИЕ МАТРИЦ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ДВИЖЕНИЯ РТС

Назаров А.А., Красило М.С.
Донской государственный технический университет, г.Ростов-на-Дону

Ключевые слова: робототехника, мехатронный объект, пульт управления, матрица.
Аннотация. Цель данной статьи – описание вспомогательного участка кода, который с легкостью поможет оператору управлять трехколесным роботом с помощью ПУ Xbox 360, а также ознакомить с читателем как это выглядит в программном коде.

THE INTRODUCTION OF MATRICES TO IMPROVE THE QUALITY OF MOVEMENT OF THE RTS

Nazarov A.A., Krasilo M.S.
Don state technical university, Rostov-on-Don

Keywords: robotics, mechatronic facility, control panel, matrix.
Abstract. The purpose of this article is to describe an auxiliary section of code that can easily help the operator control a three-wheeled robot using Xbox 360 controllers, as well as familiarize the reader with how it looks in program code.

В процессе работы над трехколесной платформой была поставлена задача – внедрить в программный код возможность нормального движения робота с помощью расчетной матрицы, по-другому называемой матрицей поворота. Но что мешает сделать движение без неё?

Дело в том, что без матрицы движение трехколесной платформы (рисунок 1) будет несколько некорректным, программный код не учитывает размеры от центра масс основания до колес, не учитывает это при повороте, поэтому было решено задумать о введении вспомогательного кода для улучшения качества движения.

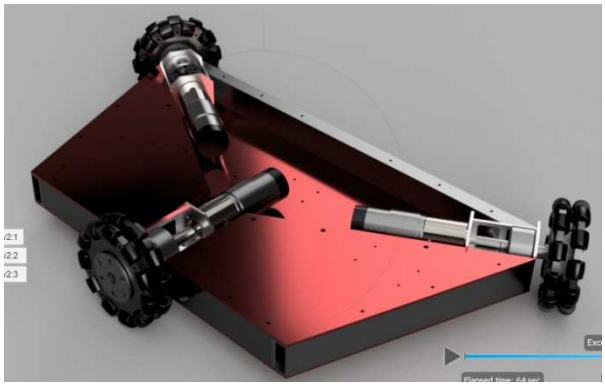

Рис. 1. Рендер трехколесной платформы

Для работы был выбран программный продукт EmBitz 1.11 а также микроконтроллер STM32f407. Для написания кода для движения выделены следующие файлы с кодом: 1) Communication.c (файл для приема данных с ПУ и их преобразоввания в дальнейшем); 2) Regulator.c (здесь будет фигурировать сама матрица); 3) Kinematics.c (здесь будет определяться матрица MRotSpeed, описание зачем она нужна будет далее).

На рисунке 2 приведено окно продукта EmBitz 1.11

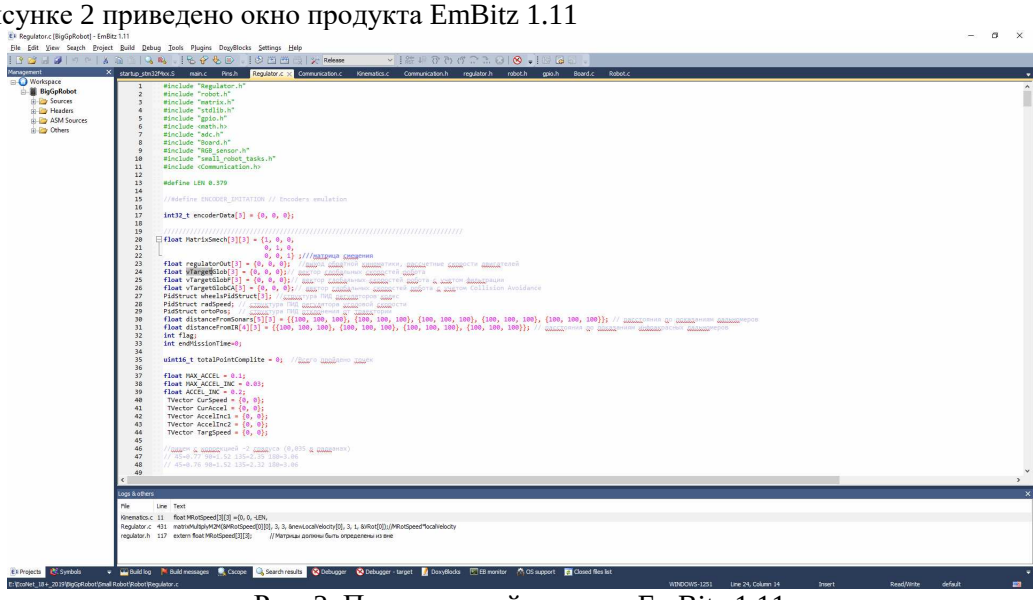

Рис. 2. Программный продукт EmBitz 1.11

Первое что было введено – введены матрицы MLineSpeed и MRotSpeed, который характеризуют соответственно скорость линейную, то есть движение по какой-то из осей и радиальная, характеризующая скорость поворота. Но это не окончательная скорость, необходимо преобразование с учетом смещения и учетом в локальной системе координат.

```
#define LEN 0.1232
#define LEN1 0.1232
 #define PHI_RAD 2.355
float MRotSpeed[3][3] ={0, 0, -LEN,
                        0, 0, -LEN,
                        0, 0, -LEN1};
float MLineSpeed[3][3] = {sinf(PHI_RAD), cosf(PHI_RAD), 0,
                        sinf(PHI_RAD), -cosf(PHI_RAD), 0,
                        -cosf(0), sinf(0), 0};
```

Переменная LEN и LEN1 характеризуют диаметры колес. Если возникает случай, когда основание платформы имеет форму равностороннего треугольника и все колеса равноудалены от центра, то нет необходимости выделять третье колесо как LEN1, но так как перед нами ситуация с равнобедренным треугольником, то это будет учитываться.

Далее переходим к написанию функции расчета скорости вращения двигателей:

```
void FunctionalRegulator(float *V_out)
{
float localVelocity[3];
float newLocalVelocity[3];
float Mrot[3][3] = {1, 0 , 0,
                    0, 1 , 0,
                    0, 0 , 1};
//матрица пересчета глобальных скоростей в локальные
float VLine[3], VRot[3], VSum[3][3], VMatr[3];
 float MaxMotSpeed;
float MaxLine,MaxRot;
if(flag == -1)
  {
matrixMultiplyM2M(&MatrixSmech[0][0], 3, 3, &localVelocity[0], 3, 1, &newLocalVelocity[0]);
  }
else
  {
        newLocalVelocity[0]=localVelocity[0];
        newLocalVelocity[1]=localVelocity[1];
        newLocalVelocity[2]=localVelocity[2];
  }
matrixMultiplyM2M(&MLineSpeed[0][0], 3, 3, &newLocalVelocity[0], 3,1,
&VLine[0]);//MLineSpeed*localVelocity
matrixMultiplyM2M(&MRotSpeed[0][0], 3, 3, &newLocalVelocity[0], 3, 1,
&VRot[0]);//MRotSpeed*localVelocity
matrixPlusMinus(&VLine[0], &VRot[0], 3, 1, 1, &VSum[0]);//MLineSpeed+MRotSpeed
c1 = 1;//ВНИМАНИЕ
matrixMultiplyS2M(&VSum[0], 3, 1, c1, &V_out[0]);
```

Функция matrixMultiplyM2M является упрощенной записью матричного умножения, чтобы уменьшить массивность программного кода. Функция matrixPlusMinus является упрощенной записью матричной суммы. c1 сигнализирует о начале передачи данных в V_out ,которые уже идут в расчет окончательной скорости с учетом регулятора.

Таким образом при движении стиком 3 на ПУ(рисунок 3) будет происходить движение платформы по введенным координатам, данные заходят в файл проекта Communication, где происходит преобразование координат.

```
x=(x-0)*(0.45-0)/(32767-0)+0;
y=(y-0)*(0.45-0)/(32767-0)+0;
```

Это нужно, потому что значение стика варьируется от -32767 до 32767, такие цифры подавать на двигатель нельзя, ибо в программном коде максимальная скорость сделана равной 0,45, чтобы успевать остановиться или повернуть робота.

Рис.3 Пу и назначенные кнопки

При нажатии триггеров 1 и 2 происходит поворот на месте, данные идут в матрицу радиальной скорости, и на выходе получаем скорость радиальной составляющей, в итоге робот будет поворачиваться стоя на месте.

Кнопка 4 была введена для изменения центра отсчета. Это нужно для того, чтобы поменять центр масс. Платформа изначально планировалась для перевозки другой платформы меньшего размера, это приведет к перебалансировке. В связи с этим нажатие кнопки 4 перемещает центр масс в центр маленького робота и при расчете скоростей это учитывается.

Таким образом, написание матрицы существенно повышает качество движения, с ними можно с легкостью взаимодействовать, а также вводить свои дополнительные условия, расширяя круг задач, которые может выполнять мобильная платформа.

Список литературы

1. Белоусов И.В. Матрицы и определители: учебное пособие по линейной алгебре. – Кишинев, 2006.
2. Литвиненко Н.А. Технология программирования на C++. Win32 API-приложения. – СПб.: БХВ-Петербург, 2010. – 288 с.
3. Подбельский В.В., Фомин С.С. Курс программирования на языке Си: учебник. – М.: ДМК Пресс, 2012. – 384 с.

Сведения об авторах:
Назаров Александр Александрович – студент, ДГТУ, г.Ростов-на-Дону;
Красило Михаил Сергеевич – студент , ДГТУ, г.Ростов-на-Дону.