

УПРАВЛЕНИЕ ДВУХКОЛЕСНОЙ ПЛАТФОРМОЙ-НОСИТЕЛЕМ ИЗМЕРИТЕЛЬНЫХ СРЕДСТВ

Е.А. Шушляпин, А.Е. Безуглая

Севастопольский государственный университет
г. Севастополь, ул. Университетская, 33
E-mail: биб@bk.ru
anna_bezuglaya@list.ru

В статье построены модель двухколесного робота и алгоритмы его терминального управления. Модель и алгоритмы управления верифицированы в среде Mathcad, а также на реальном оборудовании – колесном роботе ArcBotics Sparki. Разработанные модели при незначительной доработке описывают, в том числе, и автономные гусеничные платформы-носители измерительных средств

Введение. Двухколесные двухколейные роботы являются одним из подклассов мобильных роботов и отличаются от других подклассов отсутствием рулей и рулевых приводов. Повороты данного типа роботов осуществляются изменением скоростей вращения левого и правого колеса. Обычно такие платформы снабжаются третьей опорной точкой в виде шарика или маленького, свободно вращающегося, колеса. Весьма близки к данному типу платформ гусеничные платформы, где повороты также обеспечиваются разноразностным вращением левой и правой гусениц. Области применения гусеничных платформ широко известны в военном деле в виде машин для разминирования или извлечения взрывных устройств. В последнее время наметились и другие области использования подобных платформ. Одна из таких областей – носители измерительных приборов для использования при наземных, надводных и подводных измерениях. При этом, конечно, платформа должна принадлежать рассматриваемому классу. Примером задачи, где может применяться такая своеобразная платформа – задача очистки поверхности воды над ядерным реактором. После определенного

срока его эксплуатации приходится убирать накопившийся на поверхности мусор (листья, насекомых и т.п.), что во многих случаях (например, на Южно-Украинской АЭС), осуществляется людьми. Наличие колес по обоим бортам позволяет использовать платформу в виде катамарана и направлять мусор в промежуток между корпусами, где расположена ловушка. Ламинарные потоки, образующиеся от медленного вращения колес, вносят минимальные возмущения поверхности воды. Как измерительные платформы, данные устройства могут применяться в вулканологии и других областях измерений, связанных с вредными воздействиями. И, наконец, данная схема применяется в планетоходах.

Математическая модель. Ниже приведен вывод уравнений движения колесной платформы при следующих предположениях: динамика приводов колес не учитывается; эффект проскальзывания колес не учитывается. Похожая модель представлена Астапковичем А. М., начальником СКБ Санкт-Петербургского государственного университета аэрокосмического приборостроения, в статье [1]. В этой интернет-публикации, однако, имеется несоответствие кинематической схемы и полученных уравнений, а также не приведен их полный вывод.

Пусть x_1, x_2 – линейные перемещения выделенных точек первого и второго колеса робота; D_w – диаметр первого и второго колеса; ω_1, ω_2 – угловые скорости вращения первого и второго колеса.

Проекция на ось абсцисс «х» перемещающегося радиус-вектора R_c равна сумме движений от вращения колес с угловыми скоростями ω_1, ω_2 . Как известно, скорость линейного перемещения выделенной точки колеса диаметром

D_w равно $\frac{D_w}{2} \omega$. Скорость перемещения, вызванная вращением двух колес, равна $\frac{1}{2} \left(\frac{D_w}{2} \omega_1 + \frac{D_w}{2} \omega_2 \right)$. Множитель

1/2 учитывает то, что вращение обоих колес с одинаковой скоростью ω приводит к движению робота со скоростью ω .

В случае же разных скоростей точка, находящаяся посередине оси, жестко соединяющей колеса, будет перемещаться со средней скоростью от сложения скоростей ω_1 и ω_2 , т.е. с множителем 1/2. Проекция R_c на оси «х», «у» равны соответственно

$$\frac{D_w(\omega_1 + \omega_2)}{4} [\cos \varphi],$$

$$\frac{D_w(\omega_1 + \omega_2)}{4} [\sin \varphi],$$

где φ представляет собой угол траектории. Эти выражения являются скоростями линейных перемещений выделенных точек колес. Таким образом, первые два уравнения модели будут иметь в левых частях обозначения производных линейных перемещений точек x_1, x_2 , а правые части – приведенные выше выражения.

Что касается третьего уравнения для угла φ радиус-вектора R_0 , то для его вывода получим предварительно некоторые результаты. Очевидно, что при разных скоростях колес траектория должна иметь вид окружности, радиус R_0 которой зависит от соотношения скоростей. Получим величину этого радиуса. Пусть $\omega_1 > \omega_2$. Из-за большей скорости первого колеса (левого) робот будет поворачивать вправо. При этом левое колесо описывает окружность радиуса $R_1 = R_0 + \frac{L_r}{2}$ и длиной $2\pi R_1$ за время T . За это же время правое колесо описывает окружность радиуса $R_2 = R_0 - \frac{L_r}{2}$ и длиной $2\pi R_2$. Линейная скорость выделенной точки первого колеса равна $\frac{D_w}{2} \omega_1$. Путь, пройденный левым колесом за время T , равен $\frac{D_w}{2} \omega_1 \cdot T = 2\pi R_1$, а правым колесом – $\frac{D_w}{2} \omega_2 \cdot T = 2\pi R_2$. Получилась система двух линейных уравнений относительно двух неизвестных – T и R_0 .

Решим эту систему. Для этого в обоих уравнениях оставим справа $2\pi R_0$ и вычтем первое уравнение из второго.

$$\frac{D_w}{2} \omega_1 \cdot T - 2\pi \frac{L_r}{2} = 2\pi R_0,$$

$$\frac{D_w}{2} \omega_2 \cdot T + 2\pi \frac{L_r}{2} = 2\pi R_0.$$

Получим:

$$\frac{D_w}{2} \omega_2 \cdot T + \pi L_r - \frac{D_w}{2} \omega_1 \cdot T + \pi L_r = 0.$$

Или:

$$\frac{D_w}{2} (\omega_2 - \omega_1) \cdot T + 2\pi L_r = 0,$$

$$T = \frac{4\pi L_r}{D_w(\omega_1 - \omega_2)}. \quad (1)$$

Величину радиуса получим из второго уравнения:

$$\frac{D_w}{2} \omega_2 \cdot \frac{4\pi L_r}{D_w(\omega_1 - \omega_2)} + 2\pi \frac{L_r}{2} = 2\pi R_0,$$

$$\omega_2 \cdot \frac{L_r}{(\omega_1 - \omega_2)} + \frac{L_r}{2} = R_0,$$

$$R_0 = \left(\frac{\omega_2}{(\omega_1 - \omega_2)} + \frac{1}{2} \right) \cdot L_r. \quad (2)$$

Последнее выражение используем для верификации модели робота. Для $\omega_1 = 20$, $\omega_2 = 10$, $L_r = 0,5$ получаем $R_0 = 1,5 \cdot 0,5 = 0,75$.

Для вывода уравнения для угла φ следует учесть, что $\frac{d\varphi}{dt} = \Omega$, где $\Omega = \frac{2\pi}{T}$ – угловая скорость перемещения радиус-вектора R_0 (имеется в виду $\sin(\Omega \cdot t)$ – синусоидальное поведение координаты при траектории в виде окружности). При этом T – период, т.е. время (1), за которое робот описывает окружность вокруг некоторого центра. Таким образом, третье уравнение имеет вид:

$$\begin{aligned} \frac{d\varphi}{dt} = \Omega &= \frac{2\pi}{4\pi L_r} = \\ &= \frac{D_w(\omega_1 - \omega_2)}{2L_r}. \end{aligned} \quad (3)$$

Данное уравнение совпадает с третьим уравнением из интернет-публикации [1] с точностью до знака. Таким образом, кинематическая модель робота имеет вид:

$$\begin{aligned} \frac{dx_1}{dt} &= \frac{D_w(\omega_1 + \omega_2)}{4} [\cos \varphi], \\ \frac{dx_2}{dt} &= \frac{D_w(\omega_1 + \omega_2)}{4} [\sin \varphi], \\ \frac{d\varphi}{dt} &= \frac{D_w(\omega_1 - \omega_2)}{2L_r}. \end{aligned} \quad (4)$$

При моделировании (4) с численными данными, приведенными выше, получен фактический радиус $R_0 = 0,75$. При других численных данных верификация также успешна.

Система (4) представляет собой аффинную систему с двумя управляющими воздействиями ω_1, ω_2 , векторно-матричная запись которой имеет вид:

$$\frac{dx}{dt} = Ax + B(x)u, \quad (5)$$

где A – нулевая матрица, а компоненты матрицы $B = \{B_{ij}\}$, $i = 1, 2, 3$, $j = 1, 2$ следующие:

$$\begin{aligned} B_{11} = B_{12} &= \frac{D_w \cos(\varphi)}{4}, \\ B_{21} = B_{22} &= \frac{D_w \sin(\varphi)}{4}, \\ B_{31} = -B_{32} &= \frac{D_w}{2L_r}. \end{aligned} \quad (6)$$

В соответствии с обозначениями модели (5), векторы состояния и управления следующие:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \varphi \end{pmatrix}, \quad u = \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}.$$

Управление методом конечного состояния (МКС). Данный метод [2] представляет собой метод для отыскания управлений в терминальных задачах для нелинейных систем различных типов (дифференциальных вида (5), с переменными во времени запаздываниями, дискретных по времени и гибридных систем достаточно широкого класса). Применим МКС для приведения робота в заданное конечное состояние за заданное время. Критерий определим в виде квадратичной функции с терминальной целью, означающей приведение вектора состояния к заданным значениям x_1^*, x_2^*, x_3^* :

$$\begin{aligned} J(x(t_f)) &= F_1(x_1(t_f) - x_1^*)^2 + \\ &+ F_2(x_2(t_f) - x_2^*)^2 + \\ &+ F_3(x_3(t_f) - x_3^*)^2 \rightarrow J^* = 0. \end{aligned} \quad (7)$$

Формулы МКС применительно к модели вида (5) имеют вид:

$$\begin{aligned} u &= G^+ f_j, \quad G^+ = G^T (GG^T)^{-1}, \\ G &= \frac{\partial J(\bar{x}(t_f, t, x(t)))}{\partial \bar{x}} \times \\ &\times W(t_f, t, x(t)) B(t, x(t)), \\ \frac{d\bar{x}(\vartheta, t, x(t))}{d\vartheta} &= A \cdot \bar{x}(\vartheta, t, x(t)), \\ \frac{dW(\vartheta, t, x(t))}{d\vartheta} &= \frac{\partial A}{\partial \bar{x}}, \\ \vartheta &\in [t, t_f], \quad \bar{x}(t, t, x(t)) = x(t), \\ W(t, t, x(t)) &= I. \end{aligned} \quad (8)$$

Для модели (5) получаем:

$$\frac{d\bar{x}(\vartheta, t, x(t))}{d\vartheta} = 0,$$

следовательно, при начальных условиях $\bar{x}(t, t, x(t)) = x(t)$ получается, что

$\bar{x}(\mathcal{G}, t, x(t)) = x(t)$. Поскольку $A = 0$ и $\frac{\partial A}{\partial x} = 0$, то при начальных условиях в виде единичной матрицы $W(t, t, x(t)) = I$ получаем $W(\mathcal{G}, t, x(t)) = I$.

Тогда

$$G = \frac{\partial J(x(t))}{\partial x} \cdot I \cdot B(t, x(t)).$$

Подставляя конкретные выражения из (6) и (7), получаем:

$$G = (G_1 \quad G_2),$$

где

$$G_1 = 2 \begin{pmatrix} F_1(x_1(t) - x_1^*) \frac{D_w \cos(x_3(t))}{4} + \\ + F_2(x_2(t) - x_2^*) \frac{D_w \sin(x_3(t))}{4} + \\ + F_3(x_3(t) - x_3^*) \frac{D_w}{2L_r} \end{pmatrix},$$

$$G_2 = 2 \begin{pmatrix} F_1(x_1(t) - x_1^*) \frac{D_w \cos(x_3(t))}{4} + \\ + F_2(x_2(t) - x_2^*) \frac{D_w \sin(x_3(t))}{4} - \\ - F_3(x_3(t) - x_3^*) \frac{D_w}{2L_r} \end{pmatrix},$$

$$G^+ = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \begin{pmatrix} G_1 & G_2 \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}^{-1} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \cdot \frac{1}{G_1^2 + G_2^2}.$$

Потребуем, чтобы движение к заданной точке, т.е. при $J^* = 0$, происходило по экспоненте с постоянной времени T_u . Тогда

$$f_J = \frac{J^* - J(x(t))}{T_u} = -\frac{J(x(t))}{T_u}.$$

Ввиду громоздкости окончательного выражения для управления, выписывать

его не будем, так как при моделировании и реализации управления его все равно придется формировать по частям, так, как записано выше. Поскольку МКС-управление может оказаться нереализуемым на конкретном типе робота из-за слишком большой величины, введем ограничение на скорости:

$$\omega_i = \begin{cases} -\omega_m, & u_i < -\omega_m, \\ u_i, & -\omega_m \leq u_i \leq \omega_m, \\ \omega_m, & u_i > \omega_m, \end{cases}$$

$$i = 1, 2.$$

Управление линеаризационным методом (ЛМ) А. Исидори в модификации Вэй Ву. Постановка задачи для применения этого метода [3, 4] имеет вид критерия стабилизации

$$J(x(t)) = J^*$$

с дифференциальными ограничениями:

$$\frac{dx(t)}{dt} = \Phi(t, x(t)) + B(t, x(t))u(t),$$

$$t \in [t_0, \infty), \quad x(t_0) = x^0,$$

$$\dim x = (n \times 1), \quad \dim u = (r \times 1).$$

Для получения ЛМ-управления заданном желаемым экспоненциальном поведении критериальной функции времени $J(t) \equiv J(x(t))$ в виде дифференциального уравнения, решением которого является экспонента, стремящаяся к значению J^* с постоянной времени T :

$$\frac{dJ(t)}{dt} = \frac{J^* - J(t)}{T}. \quad (9)$$

Вычислим производную целевой функции критерия (7) с учетом уравнений (5):

$$\begin{aligned} \frac{dJ(t)}{dt} &= \frac{\partial J(x)}{\partial x} \cdot \frac{dx(t)}{dt} = \\ &= \frac{\partial J(x)}{\partial x} \cdot (\Phi(t, x(t)) + B(t, x(t))u(t)). \end{aligned} \quad (10)$$

Приравняем правые части (9) и (10) и выразим управление с помощью псевдообращения:

$$\begin{aligned} \frac{\partial J(x)}{\partial x} \cdot (\Phi(t, x(t)) + B(t, x(t))u(t)) &= \frac{J^* - J(t)}{T}, \\ \frac{\partial J(x)}{\partial x} B(t, x(t))u(t) &= \\ &= \frac{\partial J(x)}{\partial x} \Phi(t, x(t)) + \frac{J^* - J(t)}{T}, \\ u(t) &= G^+ \begin{pmatrix} -\frac{\partial J(x)}{\partial x} \Phi(t, x(t)) + \\ + \frac{J^* - J(t)}{T} \end{pmatrix}, \\ G^+ &= G^T (GG^T)^{-1}, \\ G &= \frac{\partial J(x)}{\partial x} B(t, x(t)). \end{aligned}$$

Для двухколесного робота с моделью (5, 6), где $\Phi(t, x(t)) \equiv 0$, получаем:

$$u(t) = G^+ \left(\frac{J^* - J(t)}{T} \right), \quad (11)$$

В данном случае G полностью совпадает с G для МКС-управления. В ЛМ матрица G может оказаться тождественно равной нулю, в результате чего нужно произвести преобразование Иисидори. Для двухколесного робота этого делать не надо, поскольку в данном случае $G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$ не равно тождественно нулю.

Таким образом, для двухколесного робота с моделью (5) МКС- и ЛМ-управления полностью совпадают.

Управление роботом ArcBotiks Sparki. Данный робот относится к классу двухколесных двухколейных минироботов. Он обладает следующим набором датчиков: шестью инфракрасными датчиками расстояния, ультразвуковым датчиком расстояния на управляемой поворачивающейся платформе, тремя световыми датчиками, трехосным магнитометром, трехосным акселеромет-

ром, компасом. Достоинством робота является его доступность (продается в интернет-магазинах) и низкая стоимость (около \$150). Имеющийся набор датчиков позволяет решать различные задачи управления движением (по заданной траектории, отслеживание контрастной линии, движение за световым пятном, движение по лабиринту), а также измерений (расстояний, напряженности магнитного поля, ускорений). Кроме этого, робот имеет вилочный захват и дисплей для отображения состояния датчиков и другой информации, формируемой управляющей программой. Программирование робота производится на специальной версии языка C++ посредством программной среды SparkiDuino. После отладки синтаксиса программы в SparkiDuino она передается на робот по USB-кабелю.

Для верификации алгоритма управления, изложенного выше, было произведено моделирование в среде Mathcad, а также разработана рабочая программа в среде SparkiDuino и реализована на реальном оборудовании. При этом пришлось учитывать специфику робота в части реализации управляющих воздействий и получения информации о текущем положении робота. Ниже указанные особенности реализации описаны подробнее.

Для робота ArcBotiks Sparki межцентровое расстояние $L_r = 8,5 \text{ sm} = 0,085 \text{ m}$, а диаметр колес $D_w = 5 \text{ sm} = 0,05 \text{ m}$.

Скорость вращения колеса задается параметром р3 функции

sparki.motorRotate(p1,p2,p3),

которая задает (в процентах) скорость вращения левого (p1=MOTOR_LEFT) или правого (p1=MOTOR_RIGHT) шаговых электродвигателей. Поскольку соответствие между скоростями ω_1, ω_2 с размерностями $[\omega] = c^{-1}$ и параметром р3 в документации для робота не указано, были выполнены эксперименты. В результате экспериментов установлено, что указанное соотношение отличается от линейного в области малых значений и может быть аппроксимировано полиномом четвертой степени вида

$$p3(\omega) = 0,721 + 197,686\omega - 273,529\omega^2 + 151,508\omega^3 - 27,882\omega^4.$$

Экспериментально проверено, что параметр $p3$ не должен превышать значение 200 при полной зарядке аккумуляторов. Параметр $p2=DIR_CW$ означает направление вращения по часовой стрелке, а $p2=DIR_CCW$ – против часовой стрелки. Для движения вперед нужно левому мотору назначать **DIR_CCW**, а правому – **DIR_CW**. При движении назад, что соответствует отрицательным значениям угловых скоростей, для левого колеса задаем **DIR_CW**, а для правого колеса – **DIR_CCW**. Под воздействием функции **sparki.motorRotate** робот движется в течение времени t , задаваемым функцией **delay(t)**, размещаемой в программе после **sparki.motorRotate**, где t задается в миллисекундах. Например, **delay(1000)** инициирует вращение колес в течение одной секунды.

Для реализации управления (8) (оно же (11)) необходимы время t_f , текущее время t , текущие значения состояния $x_1(t), x_2(t), x_3(t)$. Все эти величины роботом не измеряются.

Предлагается для реализации управления использовать математическую модель (5), (6) следующим образом.

1. После расчета ω_1, ω_2 по выражениям (10), (11) высчитываем скорость центра оси по формуле $\omega_c = \frac{\omega_1 + \omega_2}{2}$; при этом используется начальное время $t=0$, а также начальная точка $x(0)$, которую можно принять нулевой.

2. Определяем для контроля величину фактического перемещения центра оси Δx_c за заданное время $\Delta t = h$ из соотношения:

$$\Delta x_c = \omega_c \frac{D_w}{2} \cdot h.$$

3. Устанавливаем скорости вращения левого и правого моторов, задавая

им значения ω_1, ω_2 посредством функций

sparki.motorRotate(MOTOR_LEFT, DIR_CCW, p3(ω_1)) и **sparki.motorRotate(MOTOR_RIGHT, DIR_CCW, p3(ω_2))**.

4. Вращаем колеса робота в течение времени h посредством функции **delay(1000*h)**, задающей окончание времени вращения колес в миллисекундах.

5. Рассчитываем координаты нового центра оси робота по уравнениям (6) по схеме Эйлера:

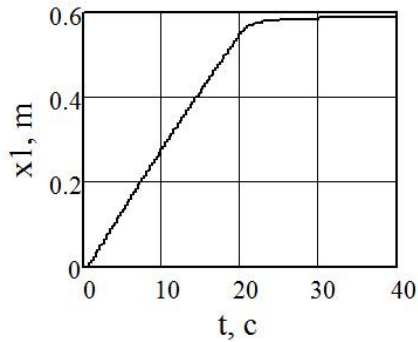
$$\begin{aligned} x_1(t + \Delta t) &= \\ &= x_1(t) + h \frac{D_w \omega_c}{2} [\cos x_3(t)], \\ x_2(t + \Delta t) &= \\ &= x_2(t) + h \frac{D_w \omega_c}{2} [\sin x_3(t)], \\ x_3(t + \Delta t) &= \\ &= x_3(t) + h \frac{D_w (\omega_1(t) - \omega_2(t))}{2L_r}. \end{aligned}$$

6. Продвигаем время $t = t + h$ и фиксируем в качестве новой начальной точки координаты нового центра оси робота.

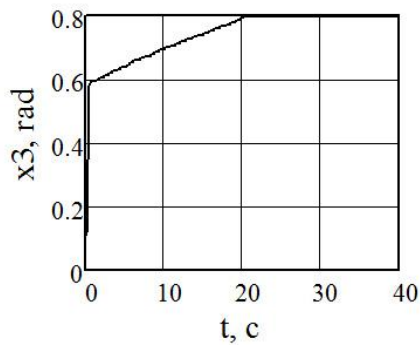
7. Управление роботом производим до тех пор, пока $t \leq t_f$.

Разработаны Mathcad-модель и управляющая C++-программа для робота, реализующие предложенный метод управления с критерием (7) для следующих параметров робота и алгоритма управления: $D_w=0,05$; $L_r=0,085$; $F_1=10$; $F_2=0$; $F_3=1$; $x_1^*=0,6$ m; $x_2^*=0,3$ m; $x_3^*=0,8$ m; $\omega_m=1,5$ c⁻¹; $h=0,1$ c; $T_u=0,01$ c; $t_f=40$ c.

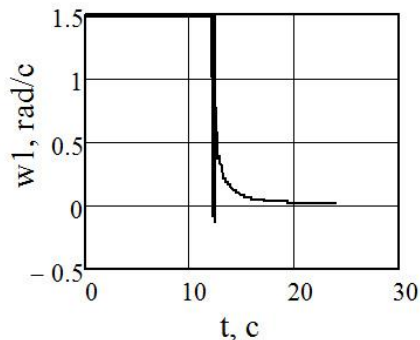
Результаты Mathcad-моделирования приведены на рисунках 1 – 4, где показаны графики зависимостей от времени управляемых координат x_1, x_3 и управляющих воздействий ω_1, ω_2 .



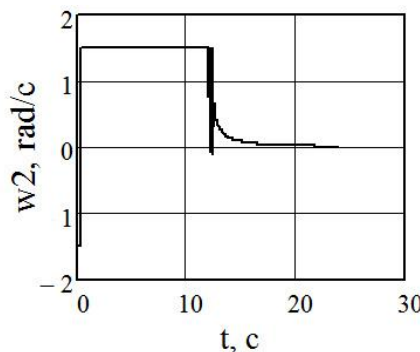
Р и с. 1. Зависимость от времени пути левого колеса



Р и с. 2. Зависимость от времени угла φ радиус-вектора



Р и с. 3. Зависимость от времени скорости левого колеса



Р и с. 4. Зависимость от времени скорости правого колеса

Как видно из приведенных графиков, желаемые значения по первой и третьей координатам (рис. 1, 2) достигаются с достаточно хорошей точностью, особенно по углу радиуса-вектора (напомним, что вторая координата ввиду $F_2 = 0$ является неуправляемой). При этом управления ω_1, ω_2 таковы, что в начальный момент времени $\omega_1 = \omega_m = 1,5$, $\omega_2 = -\omega_m = -1,5$ (см. рис.3, 4), т.е. колеса вращаются в противоположные стороны с максимальной скоростью, поворачивая робот на угол $\varphi = 0,6 \text{ c}^{-1}$, а далее устанавливаются скорости $\omega_1 = \omega_2 = \omega_m = 1,5$ до $t = 12 \text{ c}$. В оставшееся время скорости резко уменьшаются до нуля (рис. 3, 4).

Натурный эксперимент подтвердил результаты моделирования с небольшими отклонениями фактических значений от результатов моделирования (по положению – на 0,01 м, по углу – на 0,12 rad).

Закключение. В результате проведенных исследований показана возможность применения современных методов управления для решения задач управления двухколесными платформами как на уровне математического моделирования, так и при их реализации на реальном оборудовании.

СПИСОК ЛИТЕРАТУРЫ

1. *Астапкович А.М.* Виртуальный мобильный робот SOFA-2009 для исследований в области вычислительной робототехники URL: http://guap.ru/guap/skb/sofa2009_rus.doc (дата обращения: 13.10.2015)
2. *Шушляпин Е.А.* Управление нелинейными системами на основе прогноза конечного состояния неуправляемого движения. – Севастополь: СевНТУ, 2012. – 282 с.
3. *Isidori A.* Nonlinear Control Systems, 3rd Ed. – N.Y.: Springer Verlag, 1995. – 282 p.
4. *Wu Wei.* Lyapunov-based design procedures for a state-delay chemical process // 14th World Congress of IFAC, 1999. – N-7A-05.