УДК 629.7.05.001

А. А. Большаков, Д. Л. Лисицкий

УПРАВЛЕНИЕ ДВИЖЕНИЕМ МОБИЛЬНОГО РОБОТА

Введение

В настоящее время бурно развивается раздел робототехники, занимающийся созданием мобильных роботов. Однако создать роботов, уверенно перемещающихся даже по ровной поверхности, на которой имеются непреодолимые для них препятствия, пока не удалось по ряду причин, в том числе и из-за несовершенства алгоритмов управления. Поэтому разработка алгоритмов управления мобильными роботами является актуальной задачей.

Пусть необходимо, чтобы робот переместился из точки A в точку B горизонтальной плоскости (рис. 1) по траектории, минимально отклоняющейся от прямой, соединяющей эти точки (перемещение по прямой линии невозможно из-за препятствий произвольной формы). Для ясности дальнейших рассуждений предположим, что препятствие сплошное и несимметричное относительно прямой AB (рис. 1). Робот может огибать препятствие, обходя его с разных сторон, однако одно из направлений обхода будет требовать большего отклонения от прямой AB и, соответственно, должно быть исключено. Кроме того, в точке траектории, наиболее удаленной от прямой AB (точки C и C' на рис. 1), вектор скорости объекта должен быть ей параллелен и между роботом и препятствием должно оставаться некоторое заданное расстояние $Y_з$.

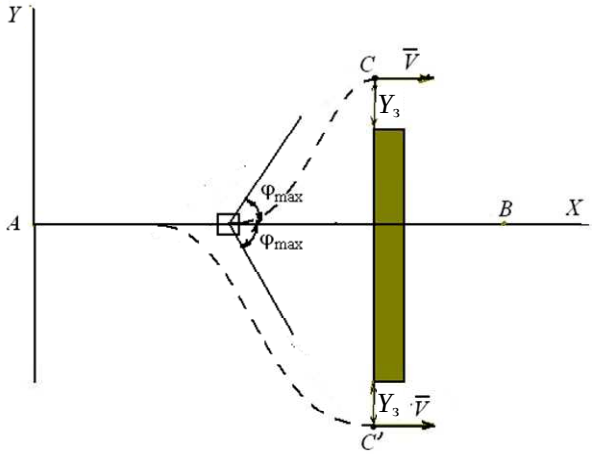

Рис. 1. Иллюстрация постановки задачи

Для обнаружения препятствий местности на корпусе робота устанавливается специальный датчик. Им может являться лазерный дальномер с лучом, сканирующим в горизонтальной плоскости (ЛДС), угол сканирования которого равен $2\varphi_{max}$, причем φ_{max} не меньше максимального угла, который может составлять траектория движения робота с прямой AB. Предположим, что ось симметрии сектора сканирования луча ЛДС параллельна прямой AB, а угол отклонения луча ЛДС от оси симметрии сектора – φ изменяется дискретно с интервалом $\Delta\varphi$ (рис. 1). Дальномер при каждом i-м измерении определяет угол наклона луча φ_i и расстояние до препятствия l_i, что позволяет вычислить координаты видимых дальномером точек препятствия в связанной с объектом системе координат. Всего за полупериод сканирования будет получена информация о расстоянии до $k = (2\varphi_{max} / \Delta\varphi) + 1$ точек препятствия. На корпусе робота установлены также два лазерных дальномера с неподвижными лучами, измеряющие расстояние от робота до препятствий в направлении перпендикулярном прямой AB.

Положение центра масс робота в неподвижной системе координат AXY (рис. 1), первые производные этих координат и ориентация робота в этой системе координат определяются с помощью комплексной инерциальной системы навигации и ориентации, корректируемой по показаниям дифференциальных *GPS*-датчиков.

Для реализации системы воспользуемся структурой (рис. 2), описанной в [1], которая позволяет разделить ее на две последовательно включенные подсистемы – «модель-вычислитель программного управления (ВПУ)» и «объект-регулятор» (рис. 3). Под моделью в этом случае понимается упрощенная математическая модель движения центра масс робота. Подсистема «модель-ВПУ» формирует программную траекторию, а подсистема «объект-регулятор» ее отслеживает.

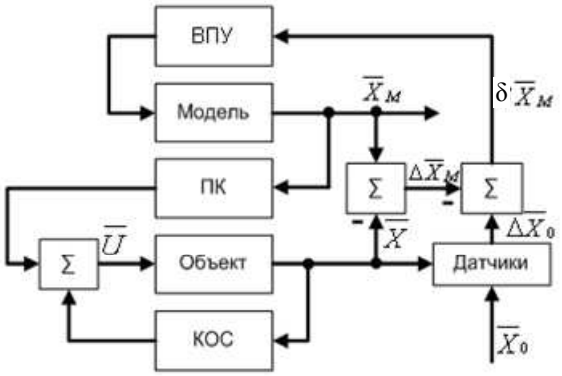

Рис. 2. Исходная структура САУ

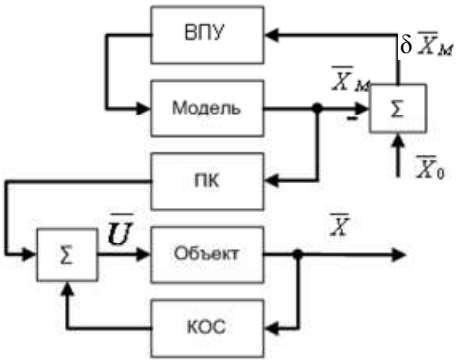

Рис. 3. Эквивалентная структура САУ

Предположим, что продольная составляющая скорости центра масс робота стабилизируется специальным регулятором и поэтому постоянна. Тогда упрощенную математическую модель движения центра масс робота можно записать как кинематические дифференциальные уравнения плоского движения материальной точки:

$$\frac{dY_M}{dt} = V_{yM}; \quad \frac{dV_{yM}}{dt} = gn_y; \quad \frac{dX_M}{dt} = V_{xM} = \text{const}, \tag{1}$$

где Y_M, X_M – координаты упрощенной модели объекта в системе координат AXY; V_{xM}, V_{yM} – соответственно продольная и поперечная составляющие скорости упрощенной модели; n_y – поперечная перегрузка модели; g – ускорение силы тяжести. В (1) X_M, Y_M, V_{xM}, V_{yM} – координаты; n_y – управление, которое далее будем называть программным управлением.

На координаты и управления упрощенной модели робота наложены ограничения:

$$-V_{y\max} \le V_{yM} \le V_{y\max}; \quad -n_{y\max} \le n_y \le n_{y\max}. \tag{2}$$

Первое из неравенств (2) ограничивает угол наклона программной траектории, второе – радиус кривизны траектории. Величины ограничений следует выбирать так, чтобы динамические свойства робота позволяли ему отслеживать «движение» модели.

Необходимо синтезировать алгоритм выбора программного управления n_y, который определяет направление обхода препятствия и переводит объект (1), (2) из произвольного начального состояния в точку, расположенную на расстоянии $Y_з$ от точки препятствия, наиболее удаленной от оси AX (рис. 1), с поперечной составляющей вектора скорости, равной нулю ($V_{yM} = 0$), таким образом, чтобы минимизировался функционал

$$I = \int_0^\infty Y_M\, dt \to \min. \tag{3}$$

Можно показать, что минимизация функционала (3) в рамках модели (1), (2) требует создания системы с максимальным быстродействием. Тогда, согласно принципу максимума Понтрягина и теореме об n-интервалах [2], оптимальное управление будет релейным и принимает только значения ($n_{y\max}, 0, -n_{y\max}$).

Для решения задачи будем использовать методику, описанную в [2], согласно которой алгоритм выбора оптимального управления синтезируется исходя из анализа положения изображающей точки относительно фазового портрета оптимального переходного процесса.

Введем новые переменные:

$$x_i = X_{\Pi i} - X_M;\ y_i = Y_{\Pi i} + Y_з \operatorname{sign}(Y_{\Pi i}) - Y_M;\ v_y = -V_{yM}, \tag{4}$$

где $X_{\Pi i}, Y_{\Pi i}$ – координаты i-й точки препятствия местности в неподвижной системе координат AXY.

Тогда система уравнений (1) примет вид:

$$\frac{dy_i}{dt} = v_y;\quad \frac{dv_y}{dt} = -gn_y;\quad \frac{dx_i}{dt} = -V_{xM} = \text{const}. \tag{5}$$

После введения новых переменных задача формально сводится к синтезу оптимального в смысле функционала (3) алгоритма выбора управления $n_y = (n_{y\max}, 0, -n_{y\max})$, перемещения изображающей точки из произвольного положения в фазовом пространстве $Oxyv_y$ в начало координат – точку O.

Фазовый портрет завершающего этапа такого переходного процесса приведен на рис. 4 (последние три интервала постоянных значений управления). Несложно видеть, что в начало координат можно попасть по двум траекториям – AO с управлением $n_y = -n_{y\max}$ и BO – с управлением $n_y = n_{y\max}$. На координату v_y, согласно обозначениям (4), также наложены ограничения:

$$-v_{y\max} \le v_y \le v_{y\max}, \tag{6}$$

где $|v_{y\max}| = |V_{y\max}|$, поэтому в точке A с координатами $(x_a, y_a, v_y = -v_{y\max})$ траектория AO переходит в траекторию AC, у которой $n_y = 0$, а $v_y = -v_{y\max} = \text{const}$. Аналогично траектория BO в точке B с координатами $(x_b, y_b, v_y = v_{y\max})$ переходит в траекторию BD, у которой $n_y = 0$, а $v_y = v_{y\max} = \text{const}$.

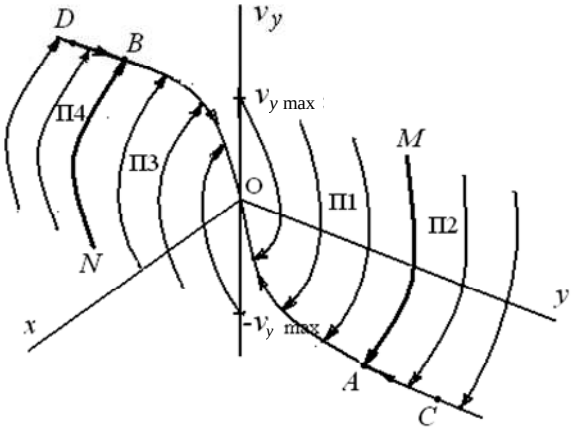

Рис. 4. Фазовый портрет системы

Приблизиться к траектории AO изображающая точка может только по поверхности П1 с управлением $n_y = n_{y\max}$, а к ее продолжению – траектории AC – по поверхности П2 тоже с управлением $n_y = n_{y\max}$. Границей между поверхностями П1 и П2 является траектория AM.

Приблизиться к траектории BO изображающая точка может только по поверхности П3 с управлением $n_y = -n_{y\max}$, а к ее продолжению – траектории BD – по поверхности П4 также с управлением $n_y = -n_{y\max}$. Границей между поверхностями П3 и П4 является траектория BN.

Для определения уравнений вышеперечисленных траекторий и поверхностей исключим из уравнений (5) время и проинтегрируем их в обратном направлении – от конца фазовых траекторий к началу при соответствующих постоянных значениях управления ($n_{y\max}, 0, -n_{y\max}$).

Получим уравнения:

– траектории *OA*:

$$y_{i\,OA} = y_i = v_y^2 /(2gn_{y\max}); \tag{7}$$

– поверхности П1:

$$y_{i\,\Pi1} = y_i = -\frac{v_y^2}{2gn_{y\max}} + \frac{(-x_i gn_{y\max} + V_{xM} v_y)^2}{4V_{xM}^2 gn_{y\max}}; \tag{8}$$

– граничной траектории *AM*:

$$y_{i\,AM} = y_i = (-v_y^2/2 + V_{y\max}^2)/(gn_{y\max}); \tag{9}$$

– поверхности П2:

$$y_{i\,\Pi2} = y_i = -\frac{v_y^2}{2gn_{y\max}} + \frac{V_{y\max}}{V_{x\text{м}}}\left(x_i - \frac{V_{xM} v_y}{gn_{y\max}} - \frac{V_{xM} V_{y\max}}{gn_{y\max}}\right); \tag{10}$$

– траектории *OB*:

$$y_{i\,OB} = y_i = -v_y^2 /(2gn_{y\max}); \tag{11}$$

– поверхности П3:

$$y_{i\,\Pi3} = y_i = \frac{v_y^2}{2gn_{y\max}} - \frac{(x_i gn_{y\max} + V_{xM} v_y)^2}{4V_{xM}^2 gn_{y\max}}; \tag{12}$$

– граничной траектории *BN*:

$$y_{i\,BN} = y_i = (v_y^2/2 - V_{y\max}^2)/(gn_{y\max}); \tag{13}$$

– поверхности П4:

$$y_{i\,\Pi4} = y_i = \frac{v_y^2}{2gn_{y\max}} - \frac{V_{y\max}}{V_{xM}}\left(x_i + \frac{V_{xM} v_y}{gn_{y\max}} - \frac{V_{xM} V_{y\max}}{gn_{y\max}}\right). \tag{14}$$

Программная траектория будет оптимальна в смысле функционала (3), если она совпадает с прямой *AB*, за исключением необходимых отклонений по объезду препятствий. Алгоритм должен анализировать все точки препятствия для определения момента начала маневра огибания препятствия, однако сам маневр должен начинаться в момент, когда потенциально опасные точки появляются как в положительной, так и отрицательной частях плоскости *AXY*. Причем программная траектория должна огибать сторону препятствия, которая меньше отклоняется от оси *AX*.

Любая *i*-я точка препятствия при $Y_{\Pi i} > 0$ является потенциально опасной для объекта, если изображающая точка попала на поверхности П1, П2, траекторию *OA* или пересекла их. Соответственно, при $Y_{\Pi i} < 0$ признаком потенциальной опасности *i*-й точки препятствия является попадание изображающей точки на поверхности П3, П4, траекторию *OB* или их пересечение.

При попадании изображающей точки на одну из поверхностей или линий переключения, приводящих ее в начало координат (в соответствии со знаком $Y_{\Pi i}$), программное управление должно выбираться равным управлению, для которого получены эта поверхность или линия. Если изображающая точка не попала ни на одну из линий или поверхностей переключения, программное управление должно выбираться таким, чтобы оно переводило модель к траектории *AB* за минимальное время.

В соответствии с изложенным подходом сформирован алгоритм выбора управления на каждом шаге интегрирования уравнений (1), однако для стабилизации объекта на прямой AB использован регулятор с зоной линейности при малых отклонениях.

1. Выбор программного управления, стабилизирующего объект на траектории AB, осуществляется регулятором с законом управления, учитывающим ограничения (2):

$$H1 = c_1(Y_M - (Y_\Pi + Y_з \operatorname{sign}(Y_\Pi))) + c_2 V_{yM}; \tag{15}$$

$$H2 = \begin{cases} H1, \text{ если } |H1| < n_{y\max}; \\ n_{y\max}\operatorname{sign}(H1), \text{ если } |H1| \ge n_{y\max}; \end{cases} \tag{16}$$

$$n_{yp} = \begin{cases} H2, \text{ если } (|V_{yM}| < V_{y\max}) \vee (H2 > 0 \wedge V_{yM} < V_{y\max}) \vee (H2 < 0 \wedge V_{yM} > -V_{y\max}); \\ 0, \text{ в остальных случаях,} \end{cases} \tag{17}$$

где c_1, c_2 – постоянные коэффициенты; $H1$, $H2$ – промежуточные переменные; n_{yp} – выходной сигнал регулятора; Y_Π – координата препятствия в системе координат AXY в месте нахождения объекта.

2. Для каждого положения луча дальномера выбирается программное управление:

Если $Y_{\Pi i} \ge 0$, то

$$H_{3i} = \begin{cases} y_{i\Pi1}, \text{ если } y_i - y_{iAM} < 0; \\ y_{i\Pi2}, \text{ если } y_i - y_{iAM} \ge 0. \end{cases} \tag{18}$$

$$n_{y1i} = \begin{cases} n_{y\max}, \text{если} [(y_i - H_{3i} \ge 0) \wedge (V_{yM} < 0)] \vee \\ \vee [(y_i - H_{3i} \ge 0) \wedge (V_{yM} \ge 0) \wedge (y_i - y_{iOA} > 0) \wedge (V_{yM} < V_{y\max})]; \\ 0, \text{если} [(y_i - H_{3i} < 0) \wedge (V_{yM} \le -V_{y\max})] \vee \\ \vee [(y_i - H_{3i} \ge 0) \wedge (V_{yM} \ge 0) \wedge (y_i - y_{iOA} > 0) \wedge (V_{yM} \ge V_{y\max})]; \\ -n_{y\max}, \text{если} [(y_i - H_{3i} < 0) \wedge (V_{yM} > -V_{y\max})] \vee \\ \vee [(y_i - H_{3i} \ge 0) \wedge (V_{yM} \ge 0) \wedge (y_i - y_{iOA} \le 0)]. \end{cases} \tag{19}$$

Если $Y_{\Pi i} < 0$, то

$$H_{4i} = \begin{cases} y_{i\Pi3}, \text{ если } y_i - y_{iBN} \ge 0; \\ y_{i\Pi4}, \text{ если } y_i - y_{iBN} < 0. \end{cases} \tag{20}$$

$$n_{y2i} = \begin{cases} -n_{y\max}, \text{если} [(y_i - H_{4i} \le 0) \wedge (V_{yM} > 0)] \vee \\ \vee [(y_i - H_{4i} \le 0) \wedge (V_{yM} \le 0) \wedge (y_i - y_{iOB} < 0) \wedge (V_{yM} > -V_{y\max})]; \\ 0, \text{если} [(y_i - H_{4i} > 0) \wedge (V_{yM} \ge V_{y\max})] \vee \\ \vee [(y_i - H_{4i} \le 0) \wedge (V_{yM} \le 0) \wedge (y_i - y_{iOB} < 0) \wedge (V_{yM} \le -V_{y\max})]; \\ -n_{y\max}, \text{если} [(y_i - H_{4i} > 0) \wedge (V_{yM} < V_{y\max})] \vee \\ \vee [(y_i - H_{4i} \le 0) \wedge (V_{yM} \le 0) \wedge (y_i - y_{iOB} \ge 0)], \end{cases} \tag{21}$$

где $i = \overline{0, k}$; H_{3i}, H_{4i} – промежуточные переменные.

Каждому значению i соответствует упорядоченная тройка $\{n_{y1i}, Y_{\Pi i}, h_{1i}\}$, где $h_{1i} = (y_i - H_{3i})$, если $Y_{\Pi i} \ge 0$, и $\{n_{y2i}, Y_{\Pi i}, h_{2i}\}$, где $h_{2i} = (y_i - H_{4i})$, если $Y_{\Pi i} < 0$.

3. Если отсутствуют тройки, в которых $Y_{\Pi i} \ge 0$, то формируется тройка $\{n_{y1} = n_{y\max}, Y_{\Pi 1} = 100, h_1 = 1\}$ и пропускаются пп. 5, 7 алгоритма.

4. Если отсутствуют тройки, в которых $Y_{\Pi i} < 0$, то формируется тройка $\{n_{y2} = -n_{y\max}, Y_{\Pi 1} = -100, h_1 = -1\}$ и пропускаются пп. 6, 8 алгоритма.

5. Из всех троек, соответствующих $Y_{\Pi i} \geq 0$, отбираем те, у которых $n_{y1i} > -n_{y\max}$. Если таких пар нет, то формируем тройку $\{n_{y1} = -n_{y\max}, Y_{\Pi 1} = 0, h_1 = -1\}$ и пропускаем п. 7 алгоритма.

6. Из всех троек, соответствующих $Y_{\Pi i} < 0$, отбираем те, у которых $n_{y2i} < n_{y\max}$. Если таких пар нет, то формируем тройку $\{n_{y2} = n_{y\max}, Y_{\Pi 2} = 0, h_2 = 1\}$ и пропускаем п. 8 алгоритма.

7. Из оставшихся троек, соответствующих $Y_{\Pi i} \geq 0$, отбираем ту, у которой $Y_{\Pi i}$ максимально. Обозначим параметры этой группы $\{n_{y1}, Y_{\Pi 1}, h_1\}$.

8. Из оставшихся троек, соответствующих $Y_{\Pi i} < 0$, отбираем ту, у которой $Y_{\Pi i}$ минимально. Обозначим параметры этой группы $\{n_{y2}, Y_{\Pi 2}, h_2\}$.

9. Выбираем программное управление:

$$n_y = \begin{cases} n_{yp}, \text{ если } h_1 < 0 \vee h_2 > 0; \\ n_{y1}, \text{ если } h_1 \geq 0 \wedge |Y_{\Pi 1}| < |Y_{\Pi 2}|; \\ n_{y2}, \text{ если } h_2 \leq 0] \wedge |Y_{\Pi 1}| \geq |Y_{\Pi 2}|. \end{cases} \quad (22)$$

В результате интегрирования уравнений (1) с выбранным значением управления получаем численное значение координат модели Y_M, V_{yM}, которые и определяют значение программной траектории в каждый момент времени. Координату X_M модели принимаем равной координате $X_{\text{р}}$ центра масс робота.

Работоспособность и эффективность предложенного алгоритма проверялись методом математического моделирования. Для этого разработана программа на языке СИ++. При моделировании использовалась линеаризованная математическая модель плоского движения гипотетического четырехколесного робота шестого порядка [3] с регулятором, обеспечивающим отслеживание программной траектории синтезированным методом АКОР.

Один из результатов моделирования представлен на рис. 5, где показана только та часть препятствия, которую объезжал робот. Программная траектория на этом графике показана сплошной линией, траектория движения центра масс робота – пунктирной.

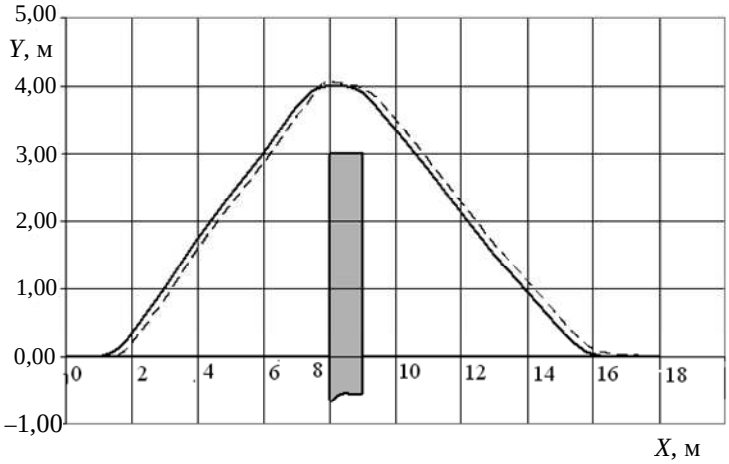

Рис. 5. Результаты моделирования объезда препятствия

Из результатов моделирования следует, что программная траектория соответствует предъявляемым требованиям. Робот правильно выбирал направление объезда препятствия, отклонение траектории движения центра масс робота от программной траектории несущественно и не превышало 0,2 м.

Заключение

Предложенный подход и алгоритм формирования программной траектории позволяют эффективно управлять движением робота. Несмотря на кажущуюся сложность, алгоритм управления может быть легко реализован в бортовом компьютере мобильного робота и может использоваться в качестве основной части общего алгоритма при создании системы управления. При этом предложенный алгоритм формирования программной траектории может послужить в качестве базового при создании алгоритмов, позволяющих мобильному роботу проезжать между препятствиями местности, объезжать группу препятствий, а также может быть положен в основу гибкой системы управления движением робота.

СПИСОК ЛИТЕРАТУРЫ

1. *Лисицкий Д. Л.* Выбор структуры системы автоматического управления траекторным движением мобильного робота // Вестн. Сарат. гос. техн. ун-та. – 2009. – № 4 (43). – С. 108–110.
2. *Лернер А. Я., Розенман Е. А.* Оптимальное управление. – М.: Энергия, 1970. – 360 с.
3. *Буданов В. М., Девянин Е. А.* О движении колесных роботов // Прикладная математика и механика. – 2003. – Т. 67, вып. 2. – С. 244–255.

Статья поступила в редакцию 17.11.2010

MOTION CONTROL OF MOBILE ROBOT

A. A. Bolshakov, D. L. Lisitsky

The article discusses creation of algorithms for systems of automatic control of mobile robot. The problem is solved within the framework of the structure that allows to separate it into two consistently included subsystems. One of the subsystems is named "the model of a computer software control". It generates a program trajectory. The second subsystem is named "the object-regulator". It controls this program trajectory. The article presents a simplified motion model of the center of mass of robot and the constraints imposed on its coordinates and controls. The article presents a synthesized algorithm for choice of program control over the phase portrait of the system by analyzing the position of the image point relatively to the surface and the switching line by the criterion of minimizing the deviation of the program trajectory from the straight line, that is connecting the start and the end points of the route. The efficiency of the proposed approach and the performance of the synthesized algorithm are tested by mathematical modeling.

Key words: robot, obstacles, program trajectory, model, algorithm, control.